Software for the Processing and Interpretation of Remotely sensed Image Time Series

USER'S MANUAL

Version: 1.5.2 - February 2018

Herman Eerens & Dominique Haesen (VITO)

Contents

Contents ... 2

Part I: General ... 8

1. Introduction ... 9

2. Image Formats ... 11

2.1. General Concepts ... 11

2.1.1. Image data ... 11

2.1.2. Annotation or “metadata” ... 12

2.1.3. Spectral-thematic annotation ... 12

2.1.3.1. Datatype & potential range ... 12

2.1.3.2. Low-endian vs. high-endian ... 13

2.1.3.3. Ordinal vs. categorical images & scaling of the values ... 13

2.1.3.4. Significant range & flags ... 14

2.1.4. Spatial-geographic annotation ... 14

2.1.4.1. Geodetic datums & datum transformations ... 14

2.1.4.2. Map projections ... 16

2.1.4.3. Image framing ... 17

2.1.5. Temporal and other annotation ... 19

2.2. The “modified ENVI” image format ... 19

2.2.1. Image files (*.img) ... 19

2.2.2. Annotation files (*.hdr) ... 20

2.2.2.1. Keywords VALUES and FLAGS ... 20

2.2.2.2. ENVI keywords for categorical images (classifications) ... 22

2.2.2.3. The ENVI method for the georeferencing of images ... 22

2.2.3. Metafiles (*.mta, *.var) ... 24

2.3. Miscellaneous elements ... 25

2.3.1. Processing of time series ... 25

2.3.2. The UNIflags system ... 25

2.3.3. SPx-files ... 26

Part II: Spirits ... 27

1. General concepts ... 28

1.1. Startup and Shutdown ... 28

1.1.1. Startup ... 28

1.1.2. Shutdown ... 29

1.2. Tools ... 30

1.2.1. Tool UIs ... 30

1.2.2. Tool File menu ... 31
1.2.3. Tool input and output files ... 31
1.3. Task execution ... 33
1.3.1. Worker thread ... 33
1.3.2. Tasks and subtasks ... 34
1.3.3. Task queue ... 34
1.3.3.1. Task manipulations ... 34
1.3.3.2. Task queue modes ... 36
1.3.4. Results queue ... 37
1.4. Time Series ... 38
1.4.1. Time Series UIs ... 38
1.4.2. Time Series Scenarios ... 39
1.4.2.1. General ... 39
1.4.2.2. Scenario selection ... 40
1.4.3. Time series file name specification ... 41
1.4.4. Time Series execution ... 43
2. Overview of the Spirits Functionalities ... 44
3. Tools and Time Series ... 46
3.1. Extract Band/ROI ... 46
3.2. IMG Thinning ... 48
3.3. IMG Resampling ... 50
3.4. Low Pass Filter ... 53
3.5. Mosaic ... 55
3.6. Reproject IMG ... 61
3.7. Scaling and Reclassification ... 70
3.7.1. General ... 70
3.7.2. SPS File: Scaling and Reclassification specification ... 72
3.8. Index ... 77
3.9. Band Calculator ... 82
3.10. Vegetation Health Index ... 108
3.11. Mask ... 111
3.12. Histogram ... 114
3.13. Frequency ... 117
3.14. ProdMax ... 119
3.15. Prod ... 121
3.16. Composit ... 124
3.17. Smooth ... 132

3.18. Cumulate ... 141
3.19. Pheno ... 146
3.20. Detect seasons ... 148
3.21. Progress of season ... 155
3.22. Cumulate over season ... 158
3.23. Cumulate full season ... 162
3.24. Crop coefficients IMG ... 166
3.25. Long Term Average ... 168
3.26. Difference ... 170
3.27. Standardized Precipitation Index ... 173
3.28. Similarity ... 177
3.29. Similarity To Yield ... 181
3.30. ISO Clustering ... 183
3.31. Convert ENVI from/to IDRISI ... 185
3.32. Convert ENVI from/to IDRISI Annotation ... 186
3.33. Convert ENVI to ArcVIEW/ArcGIS ... 187
3.34. Adapt LF to CRLF characters ... 188
3.35. Combine IMGs to 3D ENVI IMG ... 189
3.36. Descale IMG ... 190
3.37. Adapt HDR ... 192
3.38. Meteo to IMG ... 198
3.38.1. General ... 198
3.38.2. SPM File: Meteo conversion specification ... 201
3.39. Generic Import ... 207
3.40. Convert HDF5 to ENVI ... 223
3.41. Convert Periodicity ... 226
3.42. Flag VGT NDVI ... 228
3.43. Reproject SHP ... 231
3.44. Rasterize SHP ... 236
3.45. Area Fraction IMGs ... 239
3.46. Extract RUM ... 243
3.46.1. General ... 243
3.46.2. SPU File: Unmixing specification ... 245
3.46.3. RUM File: Regional Unmixed Means ... 252
3.47. RUM to Database ... 253
3.48. Reconvert RUM to IMG ... 255
4. RUM Statistics ... 258

4.1. RUM Database 258
4.1.1. Introduction 258
4.1.2. Sensors 259
4.1.3. Variables 263
4.1.4. Regions 266
4.1.5. Classes 272
4.1.6. RUM Datasets 278
4.1.7. RUM Browser 278
4.2. RUM Charts 280
4.2.1. Introduction 280
4.2.2. Datasets 282
4.2.2.1. Smoothing 282
4.2.2.2. Operations 283
4.2.2.3. Mode 290
4.2.2.4. Date range 291
4.2.3. RUM Charts Form and Panels 292
4.2.3.1. Views 292
4.2.3.2. Attributes 293
4.2.3.3. Legend 295
4.2.3.4. X-axis 297
4.2.3.5. Y-axis 301
4.2.3.6. Datasets 303
4.2.3.7. Bars 308
4.2.3.8. Table 312
4.2.4. RUM Charts Settings 313
4.2.5. RUM Charts File menu 313
4.3. RUM Matrix charts 314
4.3.1. Introduction 314
4.3.2. RUM Matrix charts Form and Panels 316
4.3.2.1. Views 316
4.3.2.2. Attributes 316
4.3.2.3. Legend 317
4.3.2.4. Colours 317
4.3.2.5. X-axis 320
4.3.2.6. Y-axis 320
4.3.3. RUM Matrix charts File menu 320
4.4. RUM Scatter charts 321

4.4.1. Introduction ... 321
4.4.2. RUM Scatter charts Form and Panels ... 323
4.4.2.1. Views ... 323
4.4.2.2. Attributes ... 323
4.4.2.3. Legend ... 323
4.4.2.4. X-axis and Y-axis ... 323
4.4.2.5. Datasets ... 324
4.4.3. RUM Scatter charts File menu ... 324
4.5. RUM Chart series tool ... 325
5. Quick Looks ... 329
5.1. Quick Look ... 329
5.1.1. Structure of a Quick Look ... 330
5.1.2. Quick Look Form and Panels ... 331
5.1.2.1. Canvas ... 331
5.1.2.2. Image ... 332
5.1.2.3. Image Background ... 333
5.1.2.4. Flags ... 334
5.1.2.5. Colours ... 335
5.1.2.6. Legend ... 339
5.1.2.7. Vectors ... 341
5.1.2.8. Pictures ... 342
5.1.2.9. Texts ... 343
5.1.2.10. View HDR ... 345
5.1.2.11. Image ROI selection. ... 346
5.1.3. Quick Look Settings ... 347
5.1.4. Quick Look File menu ... 347
5.2. Create Quick Look tool ... 348
5.3. Create ROI Quick Looks tool ... 351
6. External programs and commands ... 355
6.1. Command Line Tool ... 355
6.1.1. Goal ... 355
6.1.2. Tool ... 355
6.1.3. Examples ... 356
6.2. User Tools ... 364
6.2.1. Goal ... 364
6.2.2. Creating User Tools ... 366
6.2.3. Executing User Tools ... 367

6.2.4. User Tools parameters and UIs 368
6.2.5. User Tools examples 374
7. Miscellaneous 381
7.1. Rename files 381
7.1.2. Selecting the input files 381
7.1.3. Specifying the output files - Rename mode 381
7.1.4. Specifying the output files - Reformat dates mode 382
7.2. Create VAR/MTA files 384
7.3. Help 387
8. Spirits projects 390
8.1. Introduction 390
8.2. Selecting and creating projects 391
8.3. Project settings 392
Part III: Annexes 394
1. Spirits installation 395
2. Open Source Used in the SPIRITS Software 396
3. List of Abbreviations 402

Software for the Processing and Interpretation of Remotely sensed Image Time Series

USER'S MANUAL

Part I: General

1. Introduction

SPIRITS is a Windows-based software aiming at the analysis of remotely sensed earth observation data. Although it includes a wide range of general purpose functionalities, the focus lies on the processing of time series of images, derived from low resolution sensors such as SPOT-VEGETATION, NOAA-AVHRR, METOP-AVHRR, TERRA-MODIS, ENVISAT-MERIS and MSG-SEVIRI.

SPIRITS has been developed by VITO's remote sensing unit on behalf of (and sponsored by) the European Commission's Joint Research Centre (EC-JRC) in Ispra, Italy. The JRC-MARS group (Monitoring Agricultural ResourceS) continuously supplies the EC directorates with agro-statistical information on crop areas and yields for Europe and the major production areas of the world. This information is partly based on remote sensing images and it is used by the EC to adjust its agricultural interventions and food security policies.

SPIRITS forms the follow-up of another VITO software, called GLIMPSE (GLobal IMage Processing SoftwarE), which is a set of "command line driven" image processing routines, developed since 1990 in ANSI-C. GLIMPSE programs can be easily concatenated by means of scripting languages (DOS-batch, TCL, Python,...) to establish dedicated processing chains. At VITO's they are systematically used in this way for the automated processing of incoming satellite data. The SPIRITS development started in 2009 and primarily it only aimed at the setup of a Windows/JAVA-based Graphical User Interface (GUI) allowing a more convenient access of the GLIMPSE modules. However, gradually a number of new tools were incorporated without relationship to GLIMPSE.

In a technical sense, GLIMPSE and SPIRITS can be regarded as extensions to the widely spread commercial software IDL-ENVI, because both use the ENVI data format. Moreover, it is tacitly assumed that ENVI is available, be it only to display the generated images – a facility which is not provided by GLIMPSE or SPIRITS. However, both packages can be run perfectly on PCs which do not have ENVI installed. And if needed, image visualisation tools can be downloaded freely from the internet.

An exhaustive list of all SPIRITS functionalities will be provided in Part II - Overview of the Spirits Functionalities. For this introduction, they are summarized in three groups:

- Image processing routines allow to perform different operations in the following domains:
  - Spatial: resampling, thinning, filtering.
  - Thematic: rescaling, band combinations, masking, extraction of biophysical indicators, unsupervised classification.
  - Temporal: profile smoothing, compositing, detection of phenologies and anomalies, similarity analysis.
- Downstream analysis tools:
  - Generation of quicklook maps.
  - Extraction and management of databases with "regional unmixed means".
  - Graphical analysis of the database and creation of charts. The generated maps and charts can be ingested in agro-meteorological bulletins which provide assessments of the current crop conditions and yield forecasts.
- Other facilities:
  - Import/export of external image formats.
  - Rastering of vector files.
  - Option to setup new processing chains via so-called "user tools".
  - Project management.
  - Help functionalities & tutorial.

This User's Manual is subdivided in three parts. After this introduction, PART 1 continues with a discussion of the image formats, first in general terms, later focussing on the "modified ENVI format" adopted by GLIMPSE and SPIRITS. PART 2 covers the bulk of this manual and gives a full description of all the individual SPIRITS modules. The annexes in PART 3 give an overview of the software installation, of the included open source components and the used acronyms.

Related to this User's Manual the software also comes with a Tutorial which demonstrates the most important functionalities of SPIRITS by means of some practical exercises. However, these documents can not replace a SPIRITS training course. The target audience for this manual and software is assumed to have participated in a Spirits training.

This is not an end stage. Software development is still going on, both at the levels of GLIMPSE and SPIRITS, and new versions will be released in the near future.

The authors, November 2012

2. Image Formats

2.1. General Concepts

2.1.1. Image data

The image data as such (values per pixel) are generally stored in the most compact binary form. Compression might be used to reduce the size of the image files (Run-Length Coding, Quadtrees, ZIP,…). This approach is interesting for archiving, but not if the data still have to be processed because most often the compressed images can not directly be treated by the standard software.

As to the file organisation, images can be 2D ("single-band" or "flat") or 3D ("multi-band"). In the 3D case, different but spatially congruent image "layers" are stored together in the same file. The separate layers can for instance represent the spectral bands of a same registration or the different dates of a multitemporal image set. 3D image files can be organised following three different "interleave" types (see figure 2.1):

- BSQ or Band Sequential: first come all the data of layer 1, than those of layer 2, etc.
- BIL or Band-Interleaved per Line: first all the data of the 1st line/record, etc.
- BIP or Band-Interleaved per Pixel : first all the data of the 1st pixel, etc.

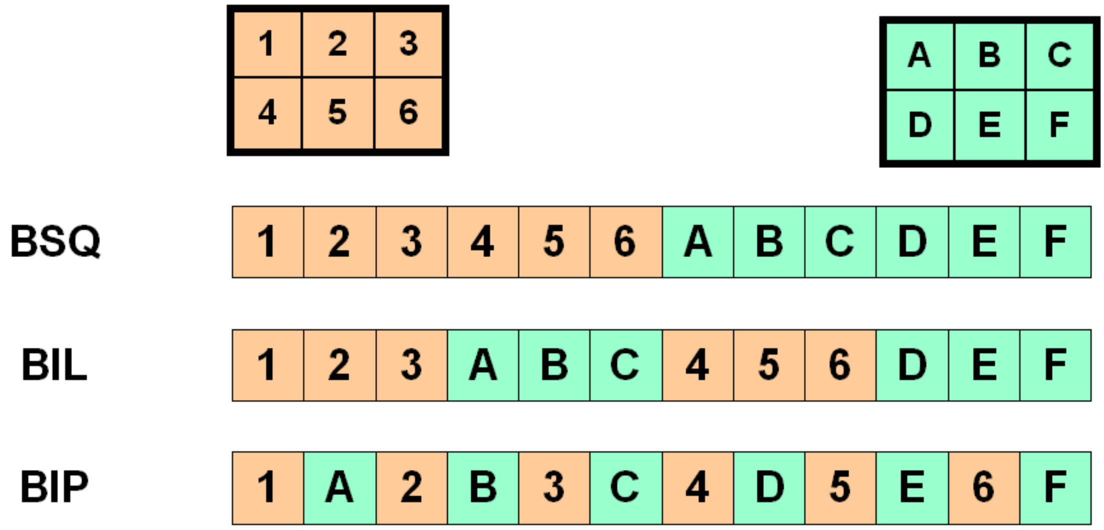


Figure 2.1*: On top two congruent 2D images (1 layer=1file). Below, both images are merged into a single 3D-image according to the three possible interleave types.*

The 3D-organisation is mostly used for small images, for instance to store RGB colour composites (JPG, TIF,…) to include in documents or posters. In the domain of remote sensing where huge data sets are the rule, they can offer serious drawbacks. For instance, the addition of a new layer to a 3D-file (or the removal of a layer) involves the copying of the entire file.

In practice, it is easier to work with 2D-images, where each layer is stored in a separate file. "Metafiles" are then used as an alternative for the 3D-organisation. These are small ASCII-files which only contain the names of all the 2D-images which belong together for a specific analysis. A metafile can be "opened" as if it were a single 3D image. Removal or addition of new layers only requires editing the image names in the metafile, not the copying of huge image data amounts.

2.1.2. Annotation or "metadata"

Each image has its own history and specifications. The "annotation" encompasses all the ancillary information needed for the correct interpretation of the image data. The annotation items can roughly be grouped in four categories (spectral, spatial, temporal, other) which are further detailed in the following paragraphs.

Another question concerns the place where the annotation is stored. At this level, two major approaches can be distinguished:

- *Annotation together with the image data*: The metadata are often saved together with the image data in a single file. In this case, the annotation items are stored – according to a specific format – in the "leader bytes" which precede the imagery. This approach is followed by many software packages of various domains: graphical (JPG, TIF, PNG), geographical (PCI, ERDAS-IMAGINE, …) and scientific (HDF, NETCDF,…). These "mixed formats" are often complex or ill-documented, which hampers software development. The main problem is that they are software-specific. For instance, a user of software A will mostly be able to open the images of software B, but they will first be converted to the A-format. Again, for the huge amounts of data, these conversions (copy operations) are contra-productive.
- *Annotation in separate ASCII-files*: This method stores the real data in a binary "image file" without leader/trailer bytes, and the annotation in a separate ASCII-TXT file with the same base name as the image. This approach is followed by IDL-ENVI (*.img/hdr), IDRISI for DOS (*.img/doc), IDRISI for WINDOWS (*.rst/rdc), ERDAS (*.img/ers), ArcView (*.bsq/hdr) and many others. For our purposes this is the most appropriate approach. The data transfer between different software packages is easy, because the image file is "common" and can remain unaltered (At most, the file name extension must be renamed). Of course, the ASCII-TXT annotation files are still software-specific, but in general the annotation files of different software packages can be mutually converted without major problems. Moreover these files are very small, so the conversion requires minimal disk space and time.

2.1.3. Spectral-thematic annotation

The spectral-thematic annotation is required for a correct interpretation of the image values.

2.1.3.1. Datatype & potential range

All the pixels in a given image follow the same datatype. This datatype determines the potential range of the pixel values but also the image file size. Table 2.1 lists the four most widely used datatypes in remote sensing.

Table 2.1: *Main datatypes and their potential ranges (BPP = Bytes-per-Pixel)*

DOMAIN	DATATYPE		BPP	DT_MIN	DT_MAX
Integers	BYTE	(unsigned)	1	0	255
	INTEGER	(signed)	2	-32 768	+32 767
	LONG	(signed)	4	-2 147 483 648	+2 147 483 647
Reals	FLOAT		4	-3.4 E+38	+3.4 E+38

The selection of a specific datatype depends on the following criteria:

- *Radiometric resolution and values range of the concerned image*: Raw satellite data (radiances) are mostly registered with 10 to 16 bit resolution and hence distributed in INTEGER format. However, the further processing involves a number of operations which degrade the intrinsic resolution. Especially the atmospheric correction introduces a lot of uncertainties, such that at the end only a limited number of discrete radiance levels remain. Such derived images can better be stored in the BYTE datatype. Another example concerns the raster conversion of a polygon map with the boundaries of 1000 parcels, each

labelled with a unique ID-number. Of course, this raster should be stored in INTEGER (1-1000 fits in the INTEGER potential range). If the number of parcels exceeds 32 767, the LONG datatype has to be used.

- *Minimization of disk storage capacity*: Obviously, any image can always be stored in FLOAT datatype. But this would lead to excessive disk space requirements. One should always select the smallest possible datatype (see BPP in the table). In practice, the vast majority of images derived from remotely sensed data can (and should be) stored in the most compact BYTE datatype.

2.1.3.2. Low-endian vs. high-endian

For the multi-byte datatypes INTEGER, LONG and FLOAT (BPP>1), the individual values ("words") can be stored in two ways:

- *Low-Endian*: The most significant byte comes first. This method is followed by computers with Motorola-processors (many UNIX-systems).
- *High-Endian*: The least significant byte comes first. This strategy is adopted by the family of INTEL-processors (Windows,...).

For instance, the 16-bit (2 bytes) INTEGER binary value "00000000 00000001" is interpreted as the decimal number 1 by a Low-Endian system, but as 256 by a High-Endian system. Obviously, mixing different approaches can lead to false interpretations. Many software packages foresee a "FLIP" or "BYTE SWAP" procedure which adapts the external data to the host system.

2.1.3.3. Ordinal vs. categorical images & scaling of the values

The **digital numbers V** stored in the binary images are mostly "**ordinal**". That means they are linearly related to a **physical variable Y** (radiance, reflectance, height above sea-level, population density…). Throughout this manual, this linear relationship will be expressed as follows:

$$Y = V_{int} + V_{slo}.V$$

For a correct interpretation of the values, both parameters of the linear equation (intercept V_{int}, slope V_{slo}) should be reminded – hence they form intrinsic part of the spectral annotation. However, it's interesting to note that this would not be the case if all images were stored in FLOAT datatype. Indeed, the FLOAT range is wide enough to allow for non-scaled storage of any variable (Y=V, or V_{int}=0, V_{slo}=1). The need for scaling arises because we mainly work with values compressed to the smallest possible datatype (mostly BYTE).

Other images might be "**categorical**": the digital numbers V only represent integer (and human-defined) ID-numbers. At this level, two variants can be distinguished:

- Classifications (of vegetation, soils,…): The number of classes is rather limited and mostly below 256, so these classified maps can be stored in the BYTE datatype. Classes can be spread all over the map.
- Rastered versions of vectorial object maps: The number of objects (countries, provinces, crop fields,...) can be huge so higher datatypes (INTEGER, LONG) are often needed. In this case, the pixels of a same object tend to cluster together.

For categorical images, the above scaling is irrelevant because one always has Y=V=ID, and hence V_{int}=0, V_{slo}=1. On the other hand, the annotation has to store the meaning of the ID-numbers, i.e. the "class legend" or "key".

Important remark: Many standard mathematical operations only make sense for ordinal images. In the example of a crop classification where V=1, 2 and 3 respectively indicate wheat, barley and maize, computing the mean of a group of adjacent pixels would be meaningless: if half of them is wheat (1) and the remainder maize (3), the result would be barley (mean=2). For categorical images, the modus/majority filter is more appropriate as a measure of central tendency.

2.1.3.4. Significant range & flags

Regarding their values, nearly all "real world" images show the following two features:

- The potential range of the selected datatype (see table 2.1) is not fully exploited. For instance, an INTEGER image with reflectances R might only have digital values V in the range 0 to 10000, with scaling R [%] = 0 + 0.01 x V. Or the values (class codes) in a BYTE crop classification might be restricted to the range V=1 to 10.
- Most images also contain one or more "flags", i.e. digital values V which represent special cases such as "no data", "sea", "cloud", "error", etc. Obviously, these flags should be interpreted and treated in a different way than the values in the significant range.

In other words, the "**potential range**", as defined by the selected datatype (see table 2.1), must be split in two parts:

- All digital values V in the **significant range**, symbolized as "**V_{lo} to V_{hi}**", should be treated in the normal way. For ordinal images the above scaling applies, while for categorical images they represent the ID-numbers of the concerned objects (classes, regions, fields,…).
- All values beyond the significant range (**$V<V_{lo}$ or $V>V_{hi}$**) should be treated as "**flags**", and the annotation should store the meaning of each flag.

It is important to avoid overlaps between flags and significant values. Two counterexamples:

- After the geometric correction, all satellite registrations show "empty" zones in the four corners, which should be flagged as "no data". But some providers distribute images which are flagged with V=0, while V=0 can also be a normal significant value.
- In the 10-daily composites of SPOT-VGT, the reflectance images can have V=0 for two reasons: the reflectance is zero (significant) or there is a flag (sea, error, ...).

2.1.4. Spatial-geographic annotation

Remotely sensed images normally cover parts of the earth surface. The spatial annotation is required to define the characteristics of the viewed zone, and to combine the imagery with external geographical information. At this level, three points must be discussed:

- The definition of the longitude and latitude of any point on earth. This belongs to the domains of physics and geodesy.
- The conversion of these spherical co-ordinates to planar co-ordinates x and y. This "map projection" is a purely mathematical operation.
- Images or "rasters" are rectangular matrices covering a certain zone in a given map projection. This involves new elements such as spatial resolution and framing.

2.1.4.1. Geodetic datums & datum transformations

A (global) geodetic datum comprises three elements:

- A 3D metric co-ordinate system with orthogonal axes XYZ, centred at the gravitational centre of the earth. Z corresponds with the polar rotational axis, and XY with the equatorial plane, the positive X pointing towards the conventional meridian (Greenwich).
- A mathematical or tabular function which describes the "geoid" in this XYZ-system, i.e. the real shape of the earth surface (or actually the irregular, equipotential surface with mean sea-level gravitation).
- Especially for cartographic purposes: a mathematical ellipsoid with semi-major and semi-minor axes a/b, which represents the best fit to the geoid. If a=b, the ellipsoid simplifies to a sphere.

When the XYZ-position of any point is known (e.g. via GPS), these metric co-ordinates can easily be converted to ellipsoidal analogues: longitude λ, latitude φ and ellipsoidal height h. Mathematically, XYZ and $\lambda\varphi h$ are interchangeable.

Although conceptually easy to understand, the exact definition of a geodetic datum is a very complicated task which involves questions such as: what is the shape of the earth, where is the polar axis, or even: how to measure longitude λ? Although one might believe there should only be one geodetic datum (the best one), for historical reasons there are plenty. Since about 1960, geodesists derive "global" datums from satellite measurements of XYZ-positions (style GPS). But since ancient history, "local datums" were inferred from $\lambda\varphi h$-measurements made over limited areas (countries). Obviously, all these attempts – since the New Times performed by national authorities – have yielded slightly different results. This can for instance be observed, when the maps of two neighbour countries are compared for an overlapping area near the territorial border. In general the λ/φ-values do not exactly fit, because both nations used different geodetic datums.

Nowadays, for cartographical purposes the WGS84 (World Geodetic System of 1984) functions as the basic reference. The WGS84-ellipsoid has dimensions a_0=6378137.0m, b_0=6356752.3m (the 0-suffix refers to WGS84). All GPS-measurements are primarily expressed as WGS84-XYZ co-ordinates.

In view of the conversion and harmonization of data from different datums, the relationship was revealed between the most relevant, existing datums and the reference WGS84 (suffix 0). Actually, in this way every datum (suffix d) is fully specified by 8 parameters:

- Datum shift parameters or Molodensky constants ΔX, ΔY, ΔZ. These express the linear translation along the three WGS84-axes between the centres of both datums (for X: $X_0 = X_d + \Delta X$).
- Three angles describing the rotation of the d-datum along the three WGS84-axes.
- The dimensions (a_d, b_d) of the datum ellipsoid.

Two types of datum transformations are available in practice:

- *Bursa-Wolff equations*: These account for all parameters and retrieve the best results.
- *Molodensky equations*: In practice it was observed that the three rotational angles are very small, at least for all datums of relatively recent origin. The simpler Molodensky equations thus only account for the translations and differences in a/b. The results are acceptable if one does not require meter-precision.

To convert a point's $\lambda\varphi h_1$-position in datum d_1 to $\lambda\varphi h_2$ in datum d_2, one thus has to perform the following steps:

- Convert $\lambda\varphi h_1$ to metric XYZ_1 (simple equations)
- Use Bursa-Wolff or Molodensky to convert XYZ_1 to XYZ_0 (valid for WGS84)
- Use the same equations in inverse mode to derive XYZ_2 in $datum_2$
- Convert XYZ_2 to ellipsoidal $\lambda\varphi h_2$.

Obviously, the above datum transformation also involves the ellipsoidal height h. In practice, most geographical data sets (images, vector files) only store planimetric co-ordinates (map-xy or $\lambda\varphi$). Even if the corresponding heights h are available in other files, most software is not able to combine them at once. As a consequence, the datum transformations are not error-free as they implicitly assume h=0. Another point is that the topographical maps (and rasterized DTMs) do not contain the ellipsoidal height h, but rather the orthometric height H. As outlined in figure 2.2, correct results can only be obtained if also the geoidal height N is known (h = H + N). Maps with geoidal height N are available for the main geodetic datums.

Remark: Below we will mostly use the symbols Lon/Lat instead of λ/φ.

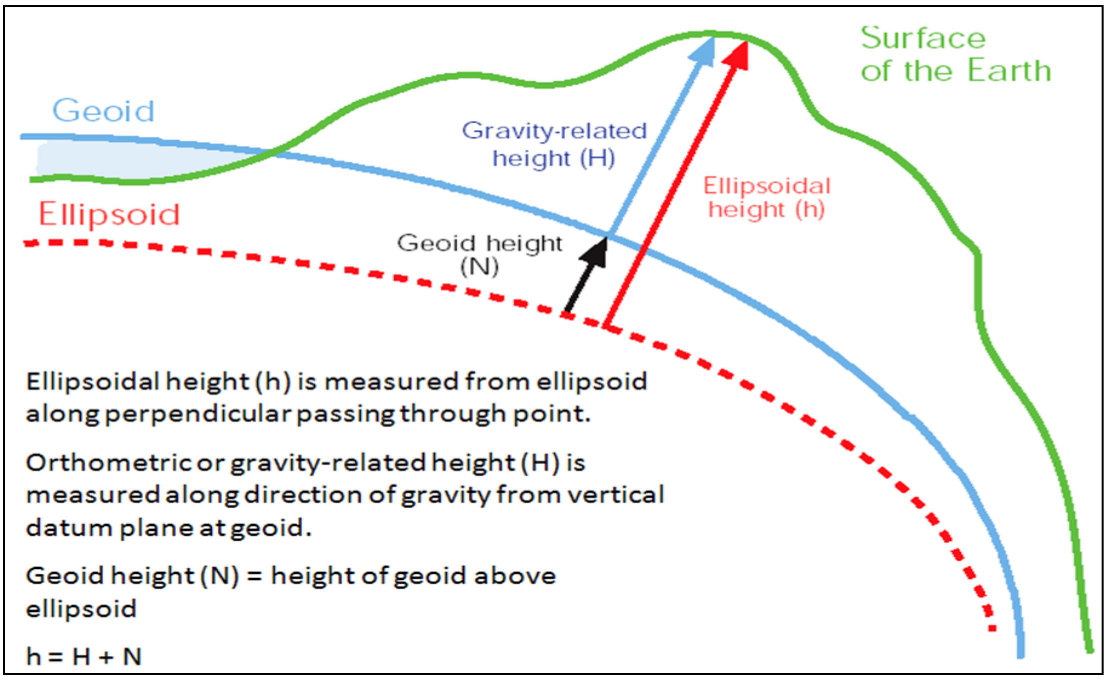


Figure 2.2*: Three definitions of "height" (from EUR 19575: Spatial reference systems for Europe, JRC, 2000).*

2.1.4.2. Map projections

The aim of any "map projection" is to transform the three-dimensional (ellipsoidal or spherical) Lon/Lat-values on a flat plane in terms of x/y co-ordinates. Mathematically, every projection consists of a set of two equations:

$$x = f_1(Lon,Lat, parameters) \qquad y = f_2(Lon,Lat, parameters)$$

In general, the projection equations are reversible, such that the original Lon/Lat can be retrieved back from the map co-ordinates x/y.

Remarks:

- Whereas geodesy has to account for observed physical data (gravity, geoid, ...) in order to define the geodetic datum which best fits the real world, map projection is a purely mathematical affair. Over the last centuries, the search for the "best" projection has dramatically rushed development in the domain of mathematics, with major contributions from Gauss, Euler, d'Alembert, Lambert and many others.
- For an excellent review of all issues related to map projections (mathematical, historical, cultural,...), see: Snyder J, 1987, *Map projections - a working manual*, USGS Professional Paper 1395.

Clear distinction should be made between:

- *Map projection families*: This is the mathematical definition of the functions f_1 and f_2. In practical life, the number of projection families is rather limited. To name some of the most important ones: Normal and Transverse Mercator, Conic Conform of Lambert, Conic Equal-Area of Albers, the three polar azimuthals (orthographic, stereographic, gnomonic) and the global ones (Sinusoidal, Mollweide, Interrupted Goode Homolosine).
- *Specific projections*: As indicated by the above equations, all projections require at least one parameter. As a consequence, the number of actual projections is endless. For instance, the Conic Equal-Area requires six parameters which can be varied in an endless way. Hence, a projection is only defined if its family is reported together with the full set of parameters.

Some of the most recurrent parameters are the following:

- *Projection ellipsoid (axes a/b)*: These two parameters are asked by all projection families. In general, one mostly selects the same values as used for the ellipsoid of the geodetic datum, on which the Lon/Lat-values were derived. But this is not an obligation. To give one simple counterexample: most global projections (Goode, Mollweide,…) use a sphere (a=b) for the projection, even for data defined in WGS84 (ellipsoid).
- *Longitude and Latitude of the origin (Lon_0, Lat_0)*: This point generally has minimal distortion and is mostly selected in the centre of the area to be mapped. It will form the "true origin" of the map xy-system.
- *False easting x_0 and false northing y_0*: To avoid negative co-ordinates, the true origin is often translated over map distances x_0, y_0 in a south-western direction. With this new, "false origin", all points in the region of interest now have positive xy-values.

The projection families are often classified according to different criteria. We only mention two of them:

- *Deformation*: The projection from spherical to planar co-ordinates unavoidably introduces some deformations at the level of the angles, areas and/or distances (scale). The latter can not be excluded for the entire map, but conformal projections retrieve correct directions in all points (at least over small distances), while equal-area (or "equivalent") maps have constant areas throughout. If an image is represented in an equal-area projection, all its pixels have the same area regardless their position.
- *Spherical vs. Ellipsoidal equations*: For many projection families two sets of equations are available (see Snyder). The spherical formulae hold for the case where the projection ellipsoid is a sphere (a=b). They are much simpler than their ellipsoidal equivalents. The spatial annotation should clearly specify which type of equations has been used.

Remarks:

- *Unprojected data (Geographic Lon/Lat)*: Of course, not all geographical information is projected. Many data sets (vectors and images) are still expressed in terms of Lon/Lat. For the digital analysis this does not raise any problem. For uniformity, one can state that in this case x=Lon and y=Lat. But this is also a kind of projection, known under the name "Plate Carré". Its general f_1/f_2-equations are simply: $x = C_x.Lon$ and $y = C_y.Lat$, with C_x and C_y two scale factors. When a vector or image file in Lon/Lat is plotted on paper or displayed on a screen (flat surfaces), we implicitly perform a Plate Carré projection.
- *Map Units*: There is a silent convention that only two units are allowed: metres for projected data (xy map co-ordinates) and decimal degrees for unprojected data (Lon/Lat). All other units (km, cm, radians,…) should be avoided.

2.1.4.3. Image framing

The combination of geodetic datum and projection fully defines the map system with planar xy co-ordinates. In case of unprojected data, we simply substitute x=Lon, y=Lat. Any xy-point in this map system is completely geo-referenced. For individual points ("vertices") and also for vectors (series of connected vertices), this information suffices. But for images (or "rasters"), the annotation has to deal with a third element: the framing. This is illustrated in figure 2.3, which shows the position of an image with 4 columns by 3 lines in a given map system (we now use XY instead of xy).

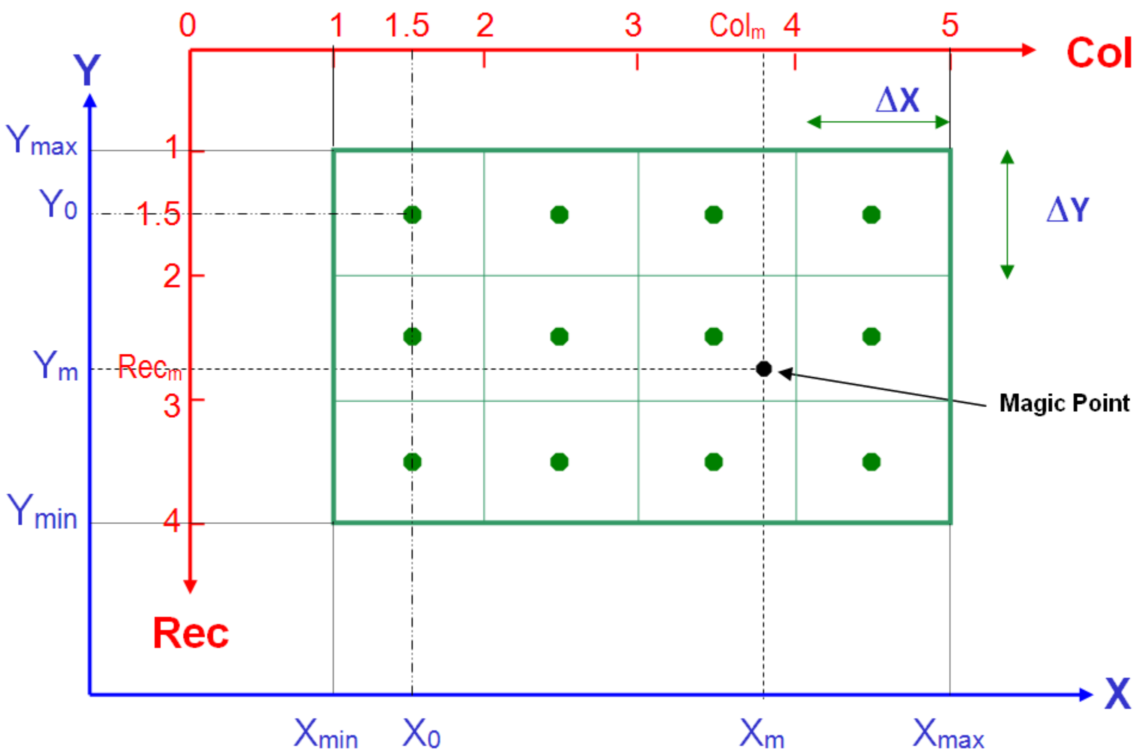


Figure 2.3: *Image framing (X_m/Y_m and Col_m/Rec_m are the map and image co-ordinates of the "magic point" as used in the definition of the framing of ENVI images)*

The framing defines the exact position and extension of the image and the pixel size (spatial resolution) in both directions. It is fully specified by six parameters. The IDRISI software for instance uses the extreme image edges in terms of map-XY (X_{min}, X_{max}, Y_{min}, Y_{max}) and the number of columns (N_{col}) and records or lines (N_{rec}). The resolutions are then derived via:

$$\Delta X = (X_{max} - X_{min}) / N_{col} \qquad \Delta Y = (Y_{max} - Y_{min}) / N_{rec}$$

But obviously, the same information can also be specified in other ways. For instance, in the ArcView software the framing must be specified via N_{col}, N_{rec}, ΔX, ΔY and X_0, Y_0. The latter two are the map co-ordinates of the centre of the topleft pixel. From this, the extreme image edges (X_{min}, X_{max}, Y_{min}, Y_{max}) can again be derived. For instance for X: $X_{min}=X_0 - \Delta X/2$ and $X_{max}=X_{min} + N_{col}.\Delta X$.

Actually, for images, we deal with a second co-ordinate system (Col, Rec), which is for instance defined as indicated in figure 2.3. Other definitions are possible but this is the one used by the ENVI software. Note that the Rec and Y-axes run in opposite ways. The crucial point is that the six image framing parameters are requested to define the parameters (intercept A, slope B) of the following two linear relationships:

$$Col = A_x + B_x.X \quad \text{and} \quad Rec = A_y + B_y.Y$$

And using the ENVI (Col/Rec) system:

- $B_x = N_{col} / (X_{max} - X_{min}) = 1/\Delta X$ and $A_x = 1 - B_x.X_{min}$.
- $B_y = N_{rec} / (Y_{min} - Y_{max}) = -1/\Delta Y$ and $A_y = 1 - B_y.Y_{max}$.

Both equations are again reversible such that the map X/Y can be derived as well for any Col/Rec-position. These equations are of crucial importance in any spatial image analysis (image display, map transformation, overlays, extraction of ROIs, …).

Figure 2.3 also illustrates the somewhat bizarre definition of the framing in ENVI. Instead of the classical six parameters, here the user has to specify eight values. In addition to the number of columns (N_{col}) and

records (N_{rec}) and the resolutions ΔX and ΔY, one also has to specify the map co-ordinates (X_m/Y_m) and corresponding image co-ordinates (Col_m/Rec_m) of any point in the XY-plane. This so-called “magic point” can be selected anywhere, even outside the image frame. These 8 parameters are also sufficient to define the above slopes and intercepts: $B_x=1/\Delta X$, $A_x=C_m-B_x.X_m$, $B_y=-1/\Delta Y$, $A_y=R_m-B_y.Y_m$. ENVI’s magic point concept provides a redundant degree of freedom. In practice the image framing is mostly defined using the topleft corner of the image ($X_m=X_{min}$, $Y_m=Y_{max}$, $Col_m=1.0$, $Rec_m=1.0$) as magic point, or the centre of the topleft pixel $X_m=X_0$, $Y_m=Y_0$, $Col_m=1.5$, $Rec_m=1.5$).

2.1.5. Temporal and other annotation

Images, especially satellite data, also have a specific timing. At this level, distinction should be made between:

- Individual registrations (raw or pre-processed): these have a registration date (and time).
- Composites (or “syntheses”) which combine the information of all images registered during a specific interval (mostly a day, dekad or month): these have a start date and a periodicity.

Finally, the annotation should also foresee place to store more textual, general information. Some examples: a title, comments, description of the applied procedures, name of the sensor (for satellite images), etc.

2.2. The “modified ENVI” image format

Based on the general concepts described above, this section expounds the basic image format used by SPIRITS and GLIMPSE. This format is called “modified ENVI” because it is ground on the ENVI principles though with some extensions and restrictions.

2.2.1. Image files (*.img)

GLIMPSE and SPIRITS only accept 2D flat, binary formatted images. Only the four most common datatypes are allowed: BYTE (unsigned), INTEGER (short, signed), LONG (integer, signed) and FLOAT (see table 2.1). And the image files must have extension *.img.

This definition thus forms a limited version of ENVI’s facilities, because ENVI images can have any extension, many more datatypes and a 3D layout.

The input images may have leader/trailer bytes, and (in most cases) they may follow the High-Endian or the Low-Endian system. However, the output images are always byte-swapped in accordance to the host system (INTEL-PC) and the leader/trailer bytes are removed. The size in bytes of each output image can thus be computed as: N_{col} x N_{rec} x BPP (BPP=bytes_per_pixel).

2.2.2. Annotation files (*.hdr)

The annotation of each image (X.img) must be stored in an associated ASCII-TXT file (X.hdr), with the same base name (X) as the image but extension *.hdr. Figure 2.4 shows an example. HDR-files must start with the text string "ENVI" in the first line. The further annotation items are labelled with “keywords”. In the HDR-file, these keywords may be listed in any order, as long as the first line contains “ENVI”. The ones used and/or recognized by GLIMPSE/SPIRITS are listed in table 2.2. Most of them are in the form “KEYWORD = *value*”. A minority follows the structure “KEYWORD = { *several comma-separated parameters* }”. In the latter case, the structure may be spread over several lines.

Remarks:

- Because the comma acts as delimiter, care should be taken to avoid the use of commas in string parameters. This error is quickly made for CLASS NAMES = { }. For instance, when a given class is named “orchards, vines and olive trees”, the comma should be replaced by another character.
- Some items may be omitted whenever redundant. For instance, for BYTE images (DATA TYPE=1), the BYTE ORDER is irrelevant and can be omitted. Similarly, INTERLEAVE is only requested for 3D-images (BANDS>1).

Actually, the “modified ENVI” format is a variation on the standard ENVI-format. In practice, two modifications were introduced:

- *Restriction*: ENVI HDR-files can contain more keywords than the ones listed in table 2.2. But for GLIMPSE/SPIRITS, some of them were suppressed, because they are redundant for our objectives. The only consequence is that GLIMPSE and SPIRITS do not “recognize” these omitted ENVI keywords and do not transfer them (whenever present in the input HDR-files) to the created output images.
- *Extension*: On the other hand, some new keywords were added. In table 2.2, these are marked in grey. The meaning and objective of these specific keywords was mostly explained in the previous section. Fortunately, the ENVI software passes these non-standard keywords from input to output header.

Some keywords are further described in the paragraphs below.

2.2.2.1. Keywords VALUES and FLAGS

The VALUES keyword specifies the spectral annotation items. The subdivision of the potential range in significant range (V_{lo}-V_{hi}) and flags was already discussed in §2.1.3.4. However, if for a certain image variable (say NDVI) the significant values range between 0 and 250, that does not mean these extremes effectively occur in all concerned images. De facto, the observed (significant) values might vary over a smaller range (say from 20 to 140). These observed extremes are stored in V_{min} and V_{max}. By definition, one always has: $V_{lo} \leq V_{min} \leq V_{max} \leq V_{hi}$. All GLIMPSE/SPIRITS modules track the actual values of V_{min} and V_{max} during the generation of the output images. But the values of V_{min}/V_{max} are only given for information, for instance to obtain a good “stretch” in the image display. The VALUES keyword also has to indicate the linear scaling of the digital numbers V to physical units Y ($Y=V_{int}+V_{slo}.V$). The scaling is only applicable to the values in the significant range (V_{lo}-V_{hi}).

For GLIMPSE/SPIRITS, the VALUES keyword is not mandatory. If it lacks for a given input image, the software uses the following defaults: V_{name}=”?”, V_{unit}=”?”, $V_{lo}=V_{min}=DT_{min}$, $V_{hi}=V_{max}=DT_{max}$, $V_{int}=0$, $V_{slo}=1$. DT_{min} and DT_{max} are the limits of the potential range of the concerned datatype (see table 2.1).

As mentioned, all digital values V beyond the significant range (V_{lo}-V_{hi}) are considered as flags and the involved pixels will be excluded from the normal computations. The FLAGS keyword is issued by the GLIMPSE/SPIRITS programs to explain the meaning of the flag values. It is only intended for information (output side), and at the input side it is never used for analytical purposes. Hence, at the input side it can always be omitted.

ENVI
description = {SPOT-VGT, S10-Synthesis}
samples = 274
lines = 224
bands = 1
header offset = 0
file type = ENVI Standard
data type = 1
interleave = bsq
map info = {bel72, 1, 1, 22000, 245000, 1000, 1000}
values = {NDVI-toc, -, 0, 250, 0, 202, -0.08, 0.004}
flags = {251=missing, 252=cloud, 253=snow, 254=sea, 255=back}
date = 20021121
days = 10
sensor type = SPOT-VEGETATION
comment = {BCGMS project}
program = {IMGcvt (V1002/1112)}

***Figure 2.4**: Example of a "modified ENVI" HDR-file. Non-ENVI keywords are marked in bold.*

***Table 2.2**: Contents of the HDR-files. The keywords are grouped per category.*
Non-ENVI keywords, added by GLIMPSE/SPIRITS, are marked in grey.

CAT	KEYWORD	DESCRIPTION
OTHER	Description = {…}	Textual info, general title
	Comment = {…}	More textual info
	Program = {…}	Name of program, which generated this IMG (+ version between brackets)
	Sensor type	E.g. SPOT-VGT, NOAA-AVHRR,… (only textual information)
3D	Bands	Nr. of image layers (for GLIMPSE/SPIRITS: normally Bands=1)
	Interleave	BSQ, BIL or BIP – Only requested for 3D-IMGs with Bands > 1
SPECTRAL	File type	"ENVI Standard" for ordinal IMGs, "ENVI classification" for categorical IMGs
	Header offset	Number of leader bytes in the IMG-file before the real image data
	Data type	1=BYTE, 2=INTEGER, 3=LONG, 4=FLOAT (see table 2.1)
	Byte order	0=least significant byte first (LSF), 1=most significant byte first (MSF)
	Values = { V_{name}, V_{unit}, V_{lo}, V_{hi}, V_{min}, V_{max}, V_{int}, V_{slo} }	Name of physical variable Y (e.g. reflectance, temperature, class,…) Dimension of physical variable Y (%, °C, kg/ha/day, -, …) Lowest/highest digital value of significant range (values beyond V_{lo}/V_{hi} are flags) Lowest/highest significant value which really occurs in this IMG NB: $V_{lo} \le V_{min} \le V_{max} \le V_{hi}$ Intercept/slope of linear scaling: physical Y = $V_{int} + V_{slo} . V$ NB: This scaling only applies to the significant range
	Classes	Nr. of classes, incl. unavoidable class 0. More correct: highest class_ID + 1
	Class names = {…}	For each class, starting with 0: class name (avoid commas!)
	Class lookup = {…}	For each class, starting with 0: R, G, B-values in range 0-255
	Flags = {…}	For each flag: "V=meaning" with V=digital value (only textual info!)
SPATIAL	Samples	Number of IMG columns (N_{col})
	Lines	Number of IMG records or lines (N_{rec})
	Map info = { Name ,Col_m, Rec_m ,X_m, Y_m ,ΔX, ΔY [,zone, N/S] [,datum] [,units=x] }	Projection_Name (must be entry in ENVI file Map_proj.txt) IMG Col/Rec co-ordinates of "Magic Point" (see figure 2.3) Map X/Y co-ordinates of same "Magic Point" X/Y pixel size in map-units Only for Projection_name=UTM: zone [1-60] and "North" or "South" Optional: geodetic datum (entry ENVI-file Datum.txt) map units: x="Metres" or "Degrees"
	Projection info = {…}	All specifications of map projection. Same form as used in ENVI-file Map_proj.txt.
TMP	Date	YYYYMMDD: IMG registration date, or start date for composite IMGs
	Days	Periodicity in days: 1, 10, 30,…; 0=unknown/irrelevant; -1=actual registration

Important remark:

Standard image processing software doesn't keep track of the scaling and doesn't distinguish flags from significant values. For a wide range of procedures this may lead to biased results. Just two examples:

- Standard software directly performs all computations on the digital values V, thereby omitting the scaling to physical units ($Y=V_{int}+V_{slo}.V$). Let's consider the example where we have two images with the reflectances R in the RED and NIR bands, and want to compute a new image with the Ratio Vegetation Index: RVI=NIR/RED. Standard packages directly apply this equation on the digital numbers V, hence $RVI_V=V_{NIR}/V_{RED}$. But obviously it's safer to use the physical units and to compute $RVI_Y=R_{NIR}/R_{RED}=(V_{int,\ NIR} + V_{slo,\ NIR}.V_{NIR})/\ (V_{int,\ RED} + V_{slo,\ RED}.V_{RED})$. All GLIMPSE/SPIRITS modules perform these scalings in the background, whenever needed. Clearly, both results are only in agreement if the two input images follow the same scaling – and if both intercepts V_{int} are zero. In practice, this is not always the case. For instance, we mostly store our RED and NIR reflectances in BYTE images with a different scaling adapted to their dynamical range (which is higher for the NIR). Numerous other examples can be given where the omission of the image scaling yields biased outcomes.
- Also the confusion between flags and significant values often leads to errors. Before any computation, GLIMPSE and SPIRITS always check the nature of the requested input values. Results are only retrieved if all of them belong to their proper significant range. If one or more inputs are flagged, the corresponding output pixel will be flagged as well. In such cases, the software always tries to maintain the same "flagging system" as used for the input images (which is not always possible).

2.2.2.2. ENVI keywords for categorical images (classifications)

Categorical images are distinguished in the HDR via "FILE TYPE = ENVI classification". These images must be BYTE, and the three class keywords must be completed. Names and colours must be provided via the keywords CLASS NAMES and CLASS LOOKUP for all class_IDs ranging consecutively from 0 up to the highest class_ID. Class 0 always represents "Not classified" or "background". For instance, if there are 3 classes, with non-consecutive IDs (or digital nrs. in the IMG) 1, 3 and 5, then CLASSES = 6, and CLASS NAMES = { not classified, class1, , class2, , class3}. Moreover CLASS LOOKUP = { … } must contain 18 comma-separated values (RGB-colours for 6 classes, inclusive the non-existing ones).

For compatibility with the GLIMPSE/SPIRITS modules, the VALUES-keyword should be completed as well. In the above example, one could have: VALUES={ class, -, 1, 5, 1, 5, 0, 1 }. Note that in this case, the scaling vanishes (Y=V=class_ID, $V_{int}=0$, $V_{slo}=1$), and that the significant range runs from $V_{lo}=1$ to $V_{hi}=5$. So the 0-class is considered as a flag. Optionally, this can be indicated explicitly via FLAGS={0=Not classified}.

2.2.2.3. The ENVI method for the georeferencing of images

The ENVI software comprises three important ASCII-TXT files (see figure 2.5):

- **Ellipse.txt** contains a list of all ellipsoids. For each, three items are provided: the *ellipse_name*, the semi-major axis a, and the semi-major axis b.
- **Datum.txt** lists all the geodetic datums, with five items per line: *datum_name*, *ellipse_name* and the three Molodensky constants ΔX, ΔY, ΔZ. *Ellipse_name* must be an entry in file Ellipse.txt.
- **Map_proj.txt** contains a list of all specific projection systems. The comment lines on top give for each projection family, the ID-number and the list of requested parameters. The following data lines provide for each projection the following items: the projection_ID (e.g. 3=Transverse Mercator), the actual values of the requested parameters, the *datum_name* (which again must be an entry in file Datum.txt) and finally the *Projection_Name* (a user-defined string).

The three files can be modified or extended as to the needs with any TXT-editor.

ELLIPSE.TXT

Clarke 1880,	6378249.1,	6356514.9
…		
International,	6378388.0,	6356911.9
…		
WGS 72,	6378135.0,	6356750.5
WGS 84,	6378137.0,	6356752.3

DATUM.TXT

Adindan,	Clarke 1880,	-166,	-15,	204
…				
European 1950,	International,	-87,	-96,	-120
…				
WGS-72,	**WGS 72**,	0,	0,	5
WGS-84,	WGS 84,	0,	0,	0

MAP_PROJ.TXT

```
; ENVI CUSTOMIZED PROJECTION FILE
; 3 - Transverse Mercator
;   a, b, lat0, lon0, x0, y0, k0, [datum], name
; 9 - Albers Conical Equal Area
;   a, b, lat0, lon0, x0, y0, sp1, sp2, [datum], name
; 11- Lambert Azimuthal Equal Area
;   a, b, lat0, lon0, x0, y0, [datum], name
;
3,     6378137.0, 6356752.3,   0.0, 129.00, 500000.0, 0.0, 0.999600,   WGS-84,    WGS84-UTM52N
9,     6378135.0, 6356750.5,  51.4, 22.65, 0.0, 0.0, 32.500000, 54.50,  WGS-72,    SpaceII (EU-NOAA)
11,    6378137.0, 6356752.3,  52.0, 10.00, 4321000, 3210000,           ETRS89,    INSPIRE-LAEA
```

***Figure 2.5**: Extracts from ENVI's three ancillary geo-referencing files*

Any specific image can be georeferenced via the HDR keyword MAP INFO = {*Projection_Name*, Col_m, Rec_m, X_m, Y_m, ΔX, ΔY}. *Projection_Name* must be an entry in the file Map_proj.TXT. Whenever needed, the system can find over there all information on the map system and geodetic datum, used for the concerned IMG.

The image framing (see figure 2.3) is further defined via keywords SAMPLES=N_{col}, LINES=N_{rec}, the two resolutions (ΔX, ΔY) and the co-ordinates of the "magic point" in map units (X_m, Y_m) and pixel units (Col_m, Rec_m). Optionally, the geodetic datum and the map units (degrees or meters) can also be included in the MAP INFO keyword.

MAP INFO = {…} can also contain three reserved *Projection_Name*'s, which are not declared in file Map_proj.txt:

- *"Geographic Lat/Lon"*: for unprojected IMGs, expressed in the Lon/Lat-system of the WGS84-datum. However, when the Lon/Lat-system is used with a different geodetic datum, it should be declared explicitly in Map_proj.txt, with a unique *Projection_Name*.
- *"UTM"* (Universal Transverse Mercator): In this case, the UTM-zone (1-60) and its position (North or South) must be given at the end of MAP INFO, after the first 7 mandatory items.
- *"Arbitrary"*: for "floating" images with unknown referencing. These can be simple pictures or raw satellite images (before the geometric correction). As to the framing, one can easily imagine fictive X/Y-axes which run in parallel to the Col/Rec-axes of the IMG. For instance: X_{min}=0, X_{max}=Ncol, Y_{min}=0, Y_{max}=N_{rec}, ΔX=ΔY=1. As an exception to the rule, in this case X and Y are expressed in pixel-units.

Finally, the keyword PROJECTION INFO = {...} allows to include the projection parameters, as specified in Map_proj.txt for the concerned system (*Projection_Name*), directly in the header file. This avoids situations where a given image cannot be treated correctly because its projection is not declared in the local version of Map_proj.txt.

2.2.3. Metafiles (*.mta, *.var)

As explained in §2.1.1, metafiles are a good alternative for the use of 3D-images. They are simple ASCII-files, basically with the names of the images which belong together for a certain application or analysis. SPIRITS/GLIMPSE have to deal with two different types of metafiles. Figure 2.6 shows an example holding for a classification exercise, which started from (500 x 500 pixel extracts of) the RED and NIR bands of two Landsat5-TM registrations. Instead of combining them into a single 3D-image file, it's more easy and flexible to keep them as separate image files and to declare them as an entity (for this particular analysis) by means of a metafile.

```
ENVI META FILE
File : D:\TESTS\clas\f1r.img
Bands: 1
Dims : 1-500,1-500
File : D:\TESTS\clas\f1n.img
Bands: 1
Dims : 1-500,1-500
File : D:\TESTS\clas\f2r.img
Bands: 1
Dims : 1-500,1-500
File : D:\TESTS\clas\f2n.img
Bands: 1
Dims : 1-500,1-500
```

```
Vu  CODE IMG NAME
11  R1   d:\tests\clas\f1R
12  N1   d:\tests\clas\f1N

21  R2   d:\tests\clas\f2R
22  N2   d:\tests\clas\f2N
```

Figure 2.6: *Metafiles for ENVI (*.mta, left) and GLIMPSE (*.var, right).*

The two types are the following:

- *ENVI metafiles*: The first line must contain the string "ENVI META FILE". Then follow 3 lines per image layer, indicating the image file name, the band to select (in case the IMG is 3D) and the image window to consider (mostly the entire image, so "Dims: 1-N_{col}, 1-N_{rec}". Although ENVI does not impose a fixed extension, SPIRITS/GLIMPSE labels all ENVI-metafiles with the reserved extension *.mta. MTA-files can be opened in ENVI as if the included layers formed a single 3D-image.
- *GLIMPSE metafiles*: These must have the fixed extension *.var (each individual image layer can indeed be considered as a "variable"). Each data line contains 3 information items for a specific image variable: a user-specified ID-number (Vu, greater than 0) in the 5 leftmost columns, the code of the variable (1-3 characters) in columns 6-10, and the image name (without extension) from column 11 onwards. Lines whose five leftmost characters do not contain a value greater than zero, are considered as comments and skipped. This is for instance the case for the top line in figure 2.6. The IDs and codes are mainly intended for the classification modules. In time series analyses, they become less relevant. Yet they may never be left blank.

Most SPIRITS/GLIMPSE-modules which produce multiple output images also retrieve an appropriate MTA and VAR-file. Metafiles can also be created manually with an editor, or with dedicated modules.

2.3. Miscellaneous elements

2.3.1. Processing of time series

Many SPIRITS/GLIMPSE programs such as compositing, smoothing and the similarity analysis, deal with series of input images. All these modules assume that the names of the images agree with the following generic template (omitting the fixed extensions *.img/hdr):

IMG-name = [P]date[S]

The prefix P and suffix S may be empty, and the prefix may also include a (complete or partial) data path, but the date must be specified according to one of the twelve formats listed in table 2.3. Hence, valid names are for instance: c:\METOP\ot_201012_NDVI (format 5: dekad 12 of 2010) or even simply "20100421" (format 1, no prefix/suffix). The "dekadal" system works as follows: the first two dekads of each month always count 10 days (01-10, 11-20), while the third one has a variable number of days (21→end_of_month).

This naming convention implies another important assumption: all the input images for a specific analysis are assumed to reside in the same data path. Oppositely, if a module generates multiple images, they are all stored in the same data path (folder).

***Table 2.3**: The twelve date formats supported by the time series modules.*

<table>
<tr><th>N</th><th>DATE FORMAT</th><th>MINIMAL PERIOD</th><th>EXPLANATION of SYMBOLS</th></tr>
<tr><td>1</td><td>YYYYMMDD</td><td rowspan="4">Day</td><td rowspan="12">YYYY = Year [1950 … 2049]
YY = Year [50=1950 … 49=2049]
MM = Month in year [01=Jan. … 12=Dec.]
m = Month in year [A=Jan. … L=Dec.]
TT = Dekad in year [01 … 36]
DD = Day in month [01 … 31]</td></tr>
<tr><td>2</td><td>YYMMDD</td></tr>
<tr><td>3</td><td>YYYYmDD</td></tr>
<tr><td>4</td><td>YYmDD</td></tr>
<tr><td>5</td><td>YYYYTT</td><td rowspan="2">Dekad</td></tr>
<tr><td>6</td><td>YYTT</td></tr>
<tr><td>7</td><td>YYYYMM</td><td rowspan="4">Month</td></tr>
<tr><td>8</td><td>YYMM</td></tr>
<tr><td>9</td><td>YYYYm</td></tr>
<tr><td>10</td><td>YYm</td></tr>
<tr><td>11</td><td>YYYY</td><td rowspan="2">Year</td></tr>
<tr><td>12</td><td>YY</td></tr>
</table>

Although most image sets fit into this generic naming convention, there are many exceptions. For instance, some composites comprise the textual name of the month in their names (Jan, Feb,...), and other follow bi-weekly time steps. SPIRITS provides a powerful renaming tool, which allows to adapt many data sets to our standards. Unforeseen periodicities (weekly, biweekly) can often be treated by "cheating" the program and declaring them as a daily series.

2.3.2. The UNIflags system

The important role of the "flags" has been stressed before. They are situated outside the significant range (V_{lo}-V_{hi}) of digital values and are used to label pixels with deviate measurements. In our project working at VITO, we maintain a so-called "UNIflags" scheme, which fixes the flag values and their meaning. The system is described in table 2.4. It only deals with BYTE and INTEGER images for the simple reason that all images are stored in these two data types (LONG and FLOAT are only used occasionally for dedicated purposes). For BYTE images, the flags are situated in the upper range of values (251-255), so the significant values (V_{lo}-V_{hi}) are restricted to the range 0-250. For INTEGER images, the significant range goes from V_{lo}=0 to V_{hi}=32767, while the flags run from -1 to -5.

Table 2.4: *The "UNIflags" system: standardized flags for BYTE and INTEGER images*

BYTE	INTEGER	MEANING / INTERPRETATION
0 – 250	**0 – 32767**	**Significant range with scaling: $Y = V_{int} + V_{slo}.V$**
251	-5	Missing value over land (data error, missing inputs, division by 0)
252	-4	Cloud over land
253	-3	Snow/Ice over land
254	-2	Sea/water
255	-1	Background (no information at all, mostly: no data)

UNIflags is an attempt to maintain the same flags for all images, created in subsequent processing steps. Unified flags simplify the interpretation of the screen displays, but also the creation of derived quick look maps, etc. The approach is presented here in advance, because some modules explicitly ask whether or not the input images follow this UNIflags system.

2.3.3. SPx-files

For some GLIMPSE programs, the list of input parameters is too long to fit on the command line (and to be practical). In that case, GLIMPSE only asks for the most relevant ones, while the others (those who mostly remain fixed) are to be specified in an ASCII-file with extension *.SP*x* (with *x*=any character). For instance, an SPS-file is needed for the image scaling, an SPC-file for the compositing, an SPU-file for the unmixing, etc. GLIMPSE users are supposed to create these files independently, using any text editor and possibly starting from the example files. But SPIRITS offers dedicated UIs for the generation of all SPx-files.

Software for the Processing and Interpretation of Remotely sensed Image Time Series

USER'S MANUAL

Part II: Spirits

1. General concepts

1.1. Startup and Shutdown

1.1.1. Startup

Spirits is started by executing the Spirits.jar executable in the Spirits installation directory (see Annexes: Spirits installation). This can be done by double-clicking the .jar file, by creating a (desktop) shortcut to the .jar file, or by creating a .bat file launching the Java VM with the .jar file as parameter.

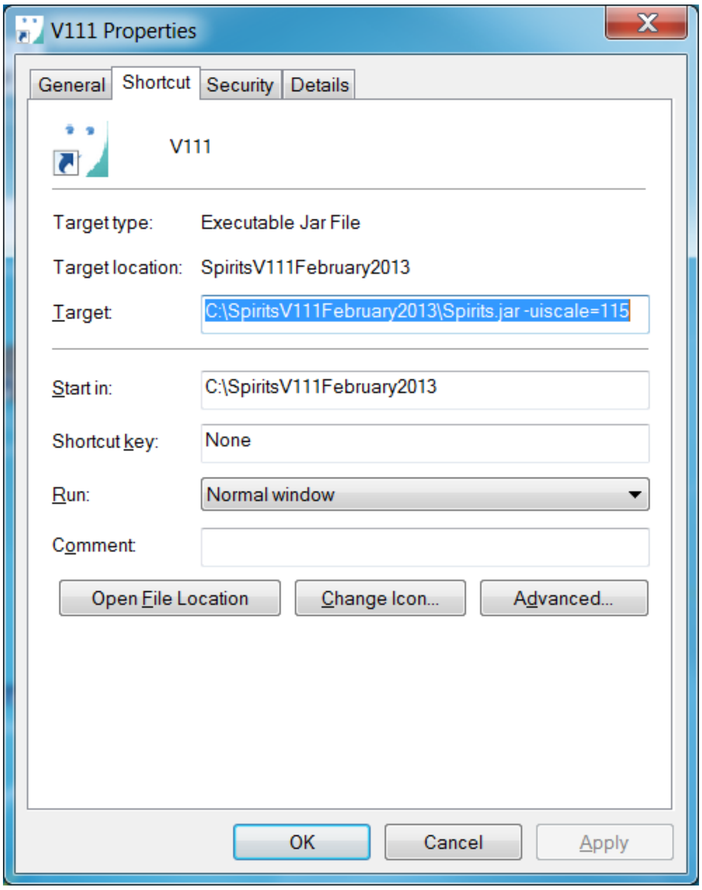
```
start javaw -jar Spirits.jar
```

example command line in a .bat file

Optionally a command line parameter (**uiscale**) can be passed to the .jar file causing all UI panels to rescale with respect to their default sizes (uiscale rescale value; an integer value in %). Typically this parameter would be specified in the desktop shortcut or in the .bat file used to start Spirits.

The parameter can be used on systems with non standard display settings (e.g. text sizes set to 125%) in which case the default sizes of some panels are too small to show all of their contents.

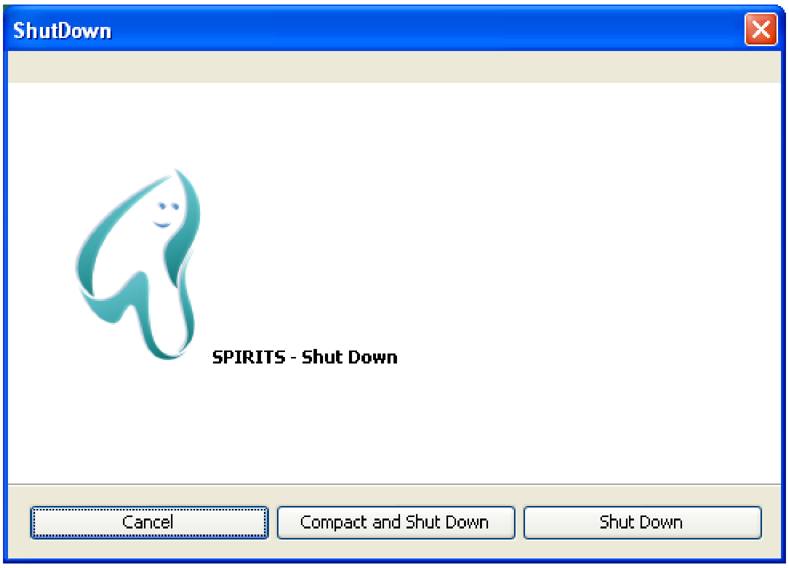
Syntax example: "…Spirits.jar **-uiscale=110**"



Example shortcut specifying "-uiscale=115"

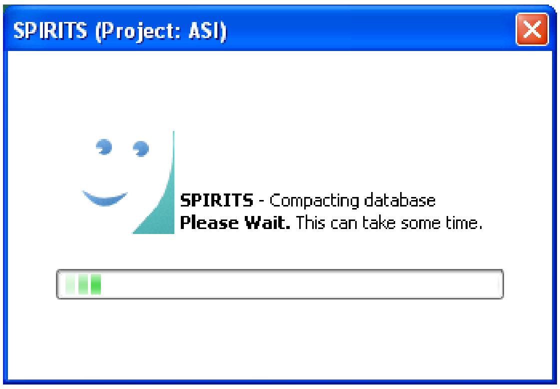
1.1.2. Shutdown

Upon closing the application, a confirmation dialog will appear. This dialog allows to "Cancel" or continue the "Shut Down" process.



It also gives an additional option "Compact and Shut Down". This option should be used periodically, especially when lots of inserts, updates or deletes have been performed on the project database (see RUM Statistics).

Please beware that the "compacting" process can take some seconds up to several hours, depending on the size of the database and the number of changes made since the last time it has been compacted. Also be aware that during this process additional disk space is required about the size of the actual database.

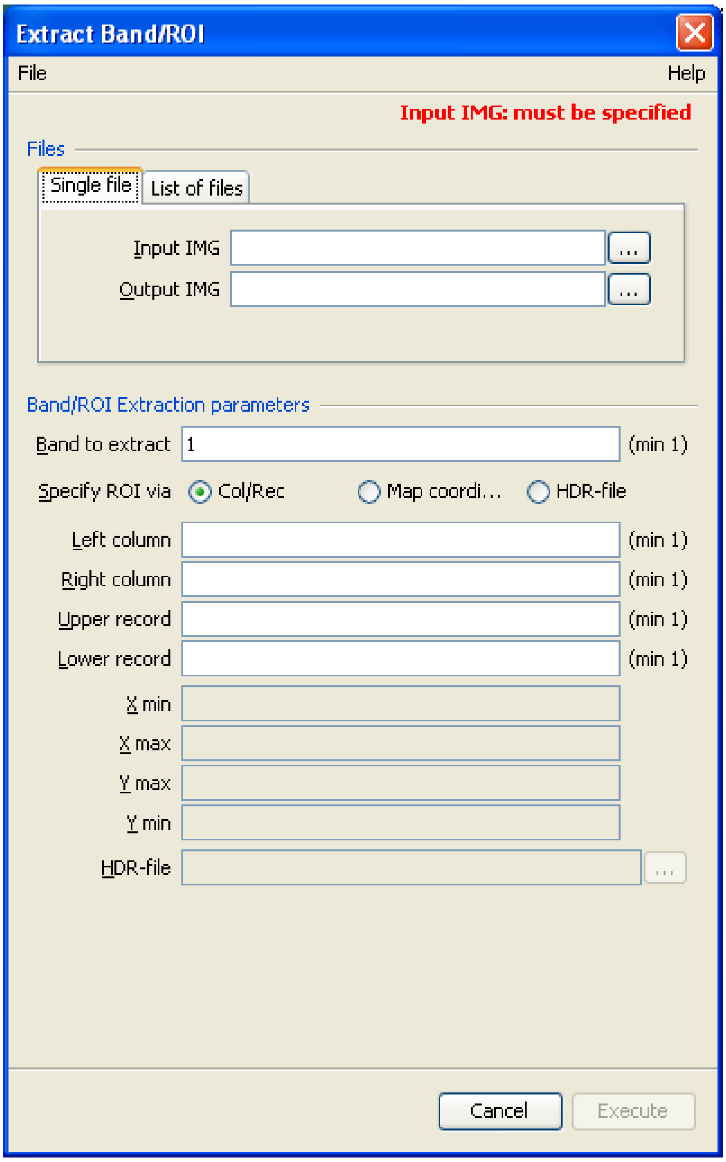


1.2. Tools

1.2.1. Tool UIs

Most Tool UIs in Spirits share a common structure:

- the frame title shows the name of the tool;
- the menu bar contains a File item, and if available an Help item;
- the zone where possible error messages are displayed;
- the user input panel - typically consists of two major parts:
    - a top part where input and output files can be specified;
    - a bottom part where more specific parameters can be specified;
- the action buttons for the tool, typically this would be a Cancel and an Execute button.



Tool title

Menu bar with File and Help items

Error messages

Actual user input panel

input/output files

tool parameters

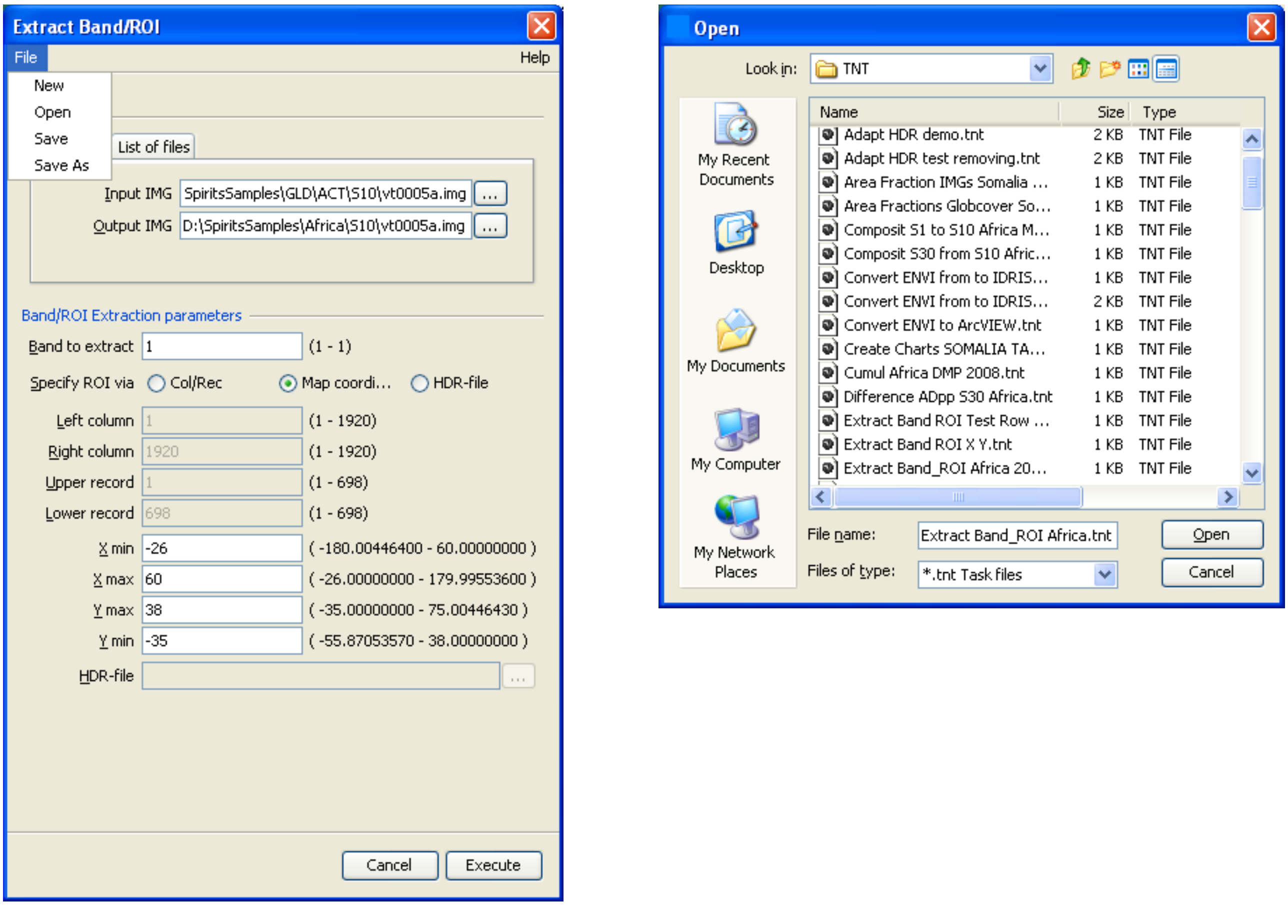
Action buttons

Example of a typical Spirits tool UI

1.2.2. Tool File menu

The Tools File menu allows to save and restore all the parameters filled out by the user, so they can be reused and modified.

The Tools parameters are stored in files with a "tnt" extension. They are called TNT files for short.

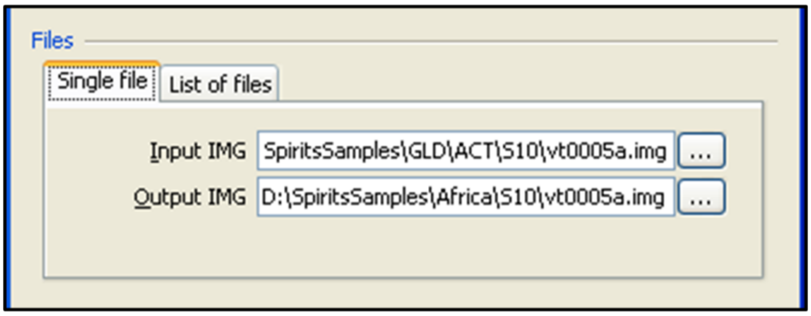


Tools File menu

Remark: the default location (directory) for TNT files can be specified per Spirits project in the Project settings.

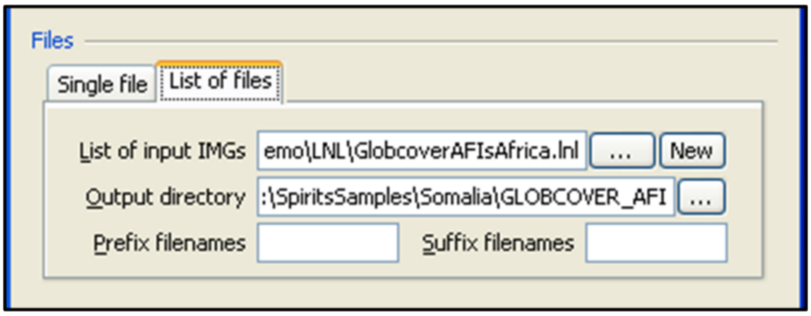
1.2.3. Tool input and output files

Most tools operate on files, therefore their UIs will contain fields where input and output files can be filled out or chosen via a file chooser at the top of the user input panel.



single input / output file

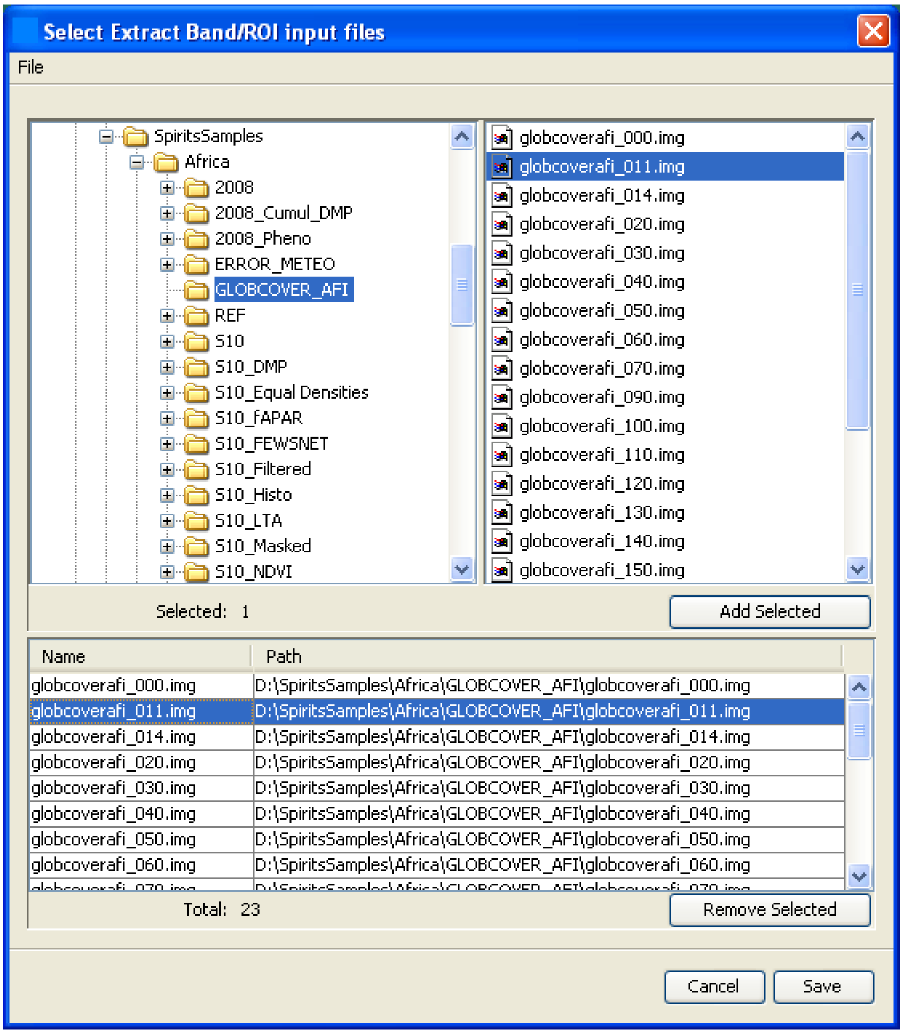
Some tools will offer a "List of files" option. This allows to submit the tool to be executed over a list of input files at once.



list of files

In this case the actual input files to be processed by the tool, are specified via a "meta file" containing a list of input files. These meta files use the "lnl" extension and are called LNL files for short.

Existing LNL files can be chosen via a file chooser. Via the "New" function new LNL files can be created: a panel will appear (LNL editor) allowing the user to browse through the file system and select the files to be added to the list. The resulting list can then be saved as an LNL file. The LNL editor can also open an existing LNL file, and add or remove files in its list.



LNL editor

When working with an input list of files, the output files are specified via

- a directory where the output files should be stored;
- an optional prefix;
- an optional suffix.

For each input file, the file name of the corresponding output file will then be formatted as:

PrefixInputfilenameSuffix.Extension of which the extension is determined by the tool.

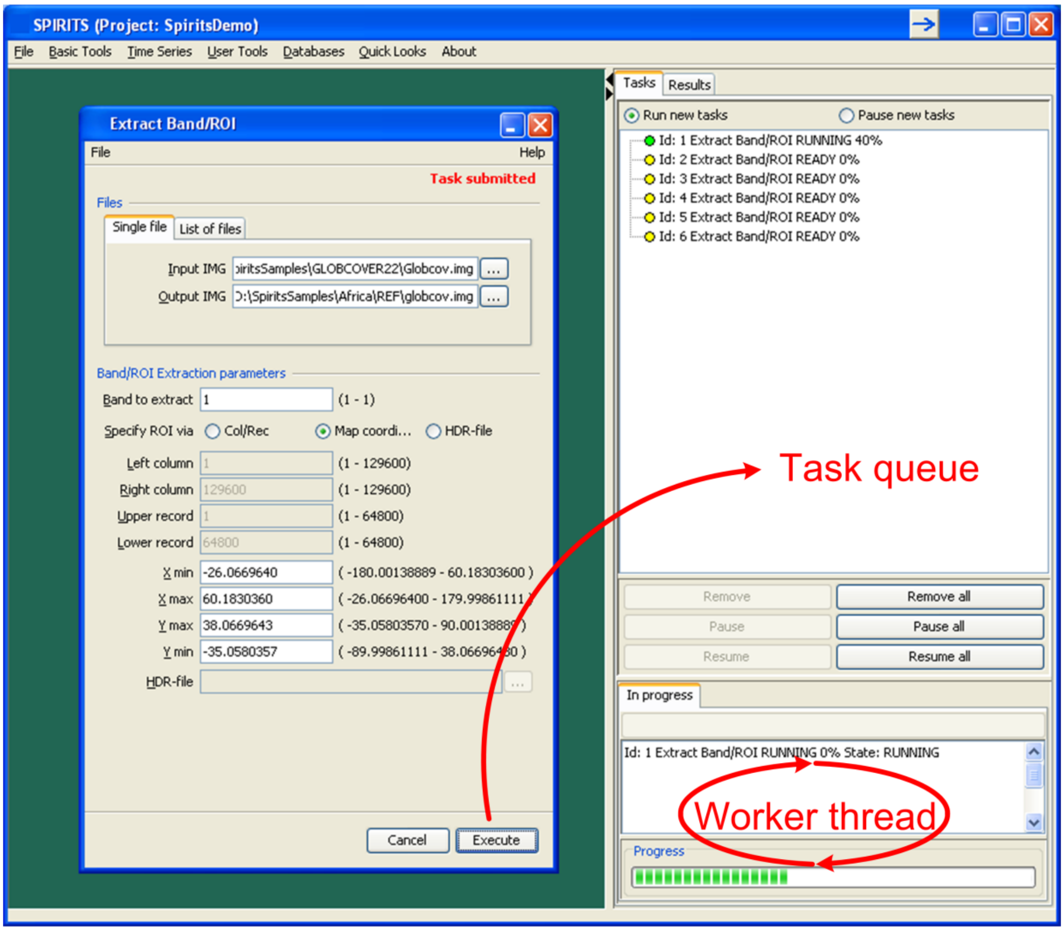
Remark: LNL files can typically be used for collections of related files which have no temporal aspect. An example would be the collection of area fraction images (AFIs) on a global or continental level, which can be used repetitively to create AFIs for smaller regions of interest (ROI's), such as countries or other administrative regions.

1.3. Task execution

1.3.1. Worker thread

When the parameters of a tool have been filled out, it can be submitted for processing via the Execute action button.

Processing does not start immediately: a task is created and pushed in the Task queue where it will wait for its turn to be processed by a separate worker thread.



1.3.2. Tasks and subtasks

When a tool is submitted for processing, it creates a task which will be pushed in the task queue.

A task can be a single task, typically in case of a basic tool operating on a single input file.

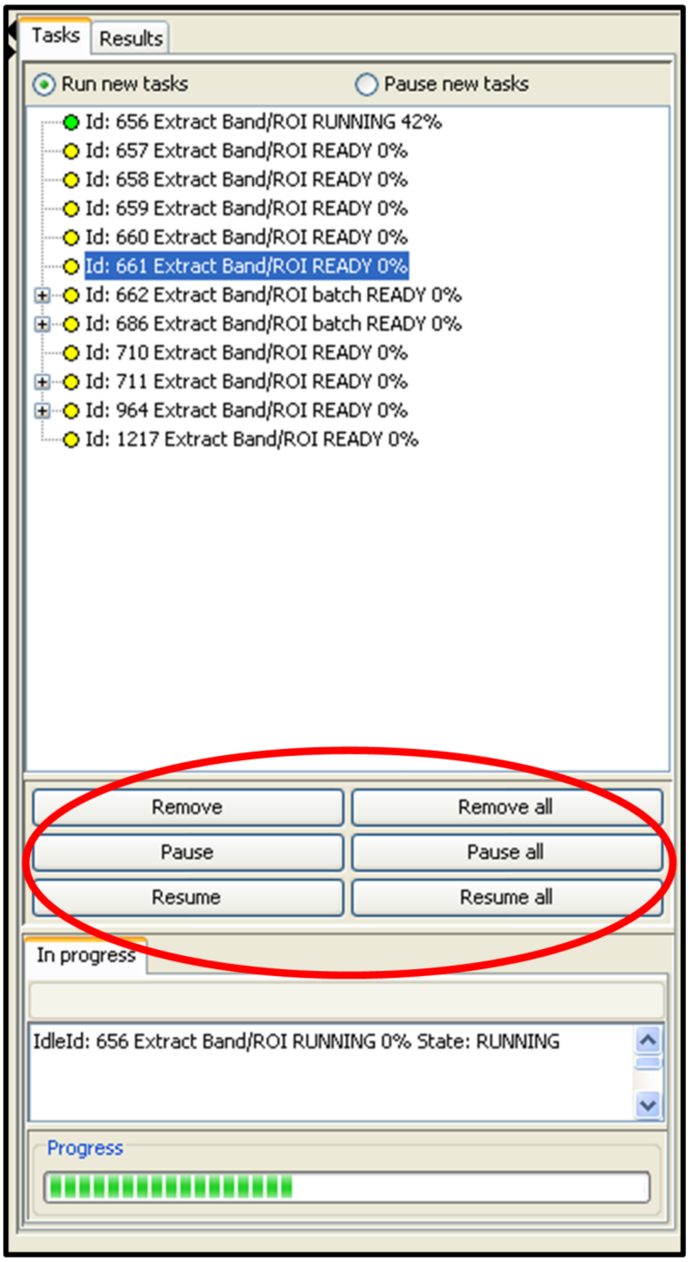
A task can also be a batch task, consisting of a series of subtasks. Examples of batch tasks are:

- a basic tool operating on a list of files;
- a time series tool operating on a time series of input files;
- tools combining multiple discrete steps.

1.3.3. Task queue

1.3.3.1. Task manipulations

Tasks which are waiting in the task queue to be processed can be manipulated.



Remove

remove the selected task from the queue

Remove All

remove all tasks from the queue

Pause

pause the selected task. A paused task will not be processed and stays in the queue until it is removed or resumed

Pause All

pause all tasks in the queue

Resume

resume the selected task

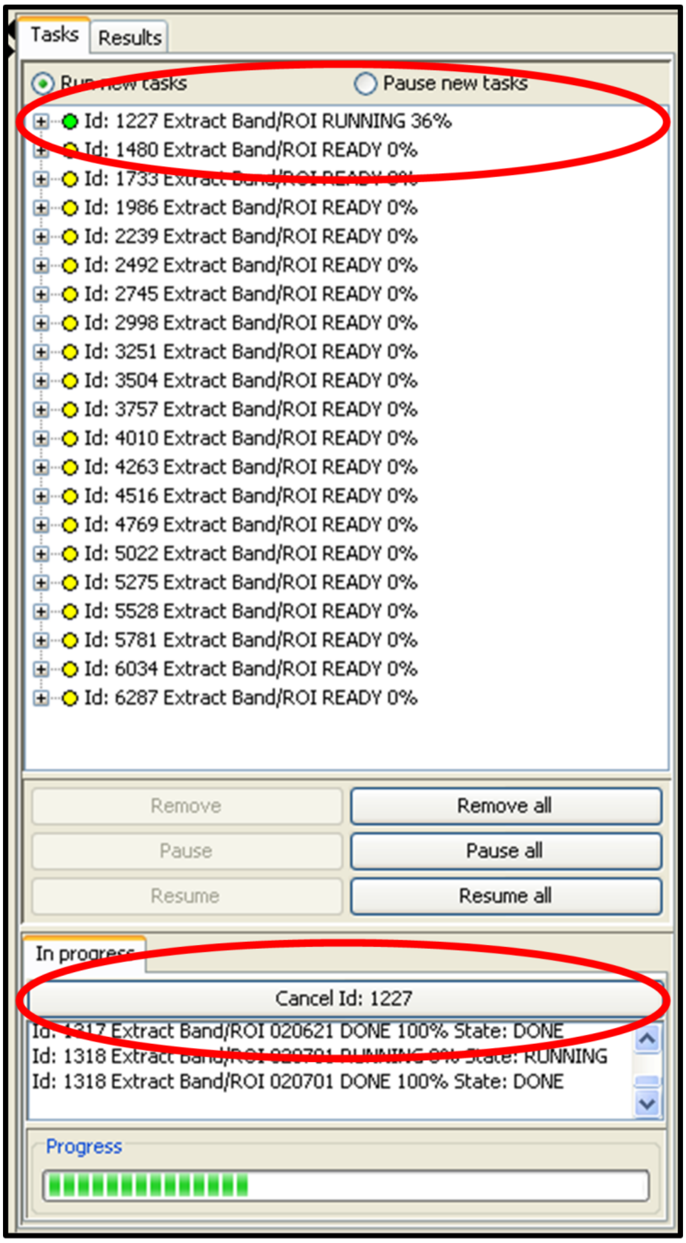
Resume All

resume all (paused) tasks in the queue.

Task Queue

The Remove, Pause and Resume operations are not applicable on tasks which are being processed. They work only on tasks waiting in the task queue to be processed. In the case of batch tasks (tasks consisting of a series of subtasks) they can only be applied on the top level task, not on individual subtasks.

Besides these operations on waiting tasks, there is also a Cancel operation, but only for batch tasks.



Task Queue

In general, tasks which are being processed cannot be interrupted.

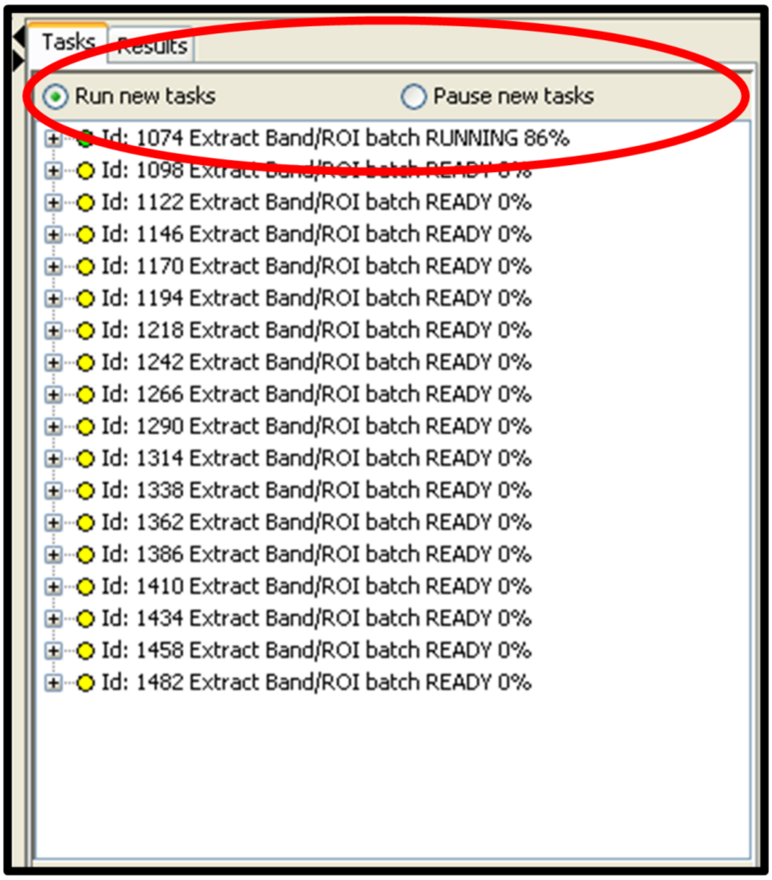
However, in the case of batch tasks the top level task can be requested to cancel.

The processing will then continue until the active (leaf) subtask ends, the remaining subtasks will not be started.

1.3.3.2. Task queue modes

The task queue itself has two states or modes:

- "Run new tasks" - default
- "Pause new tasks"



Task Queue modes

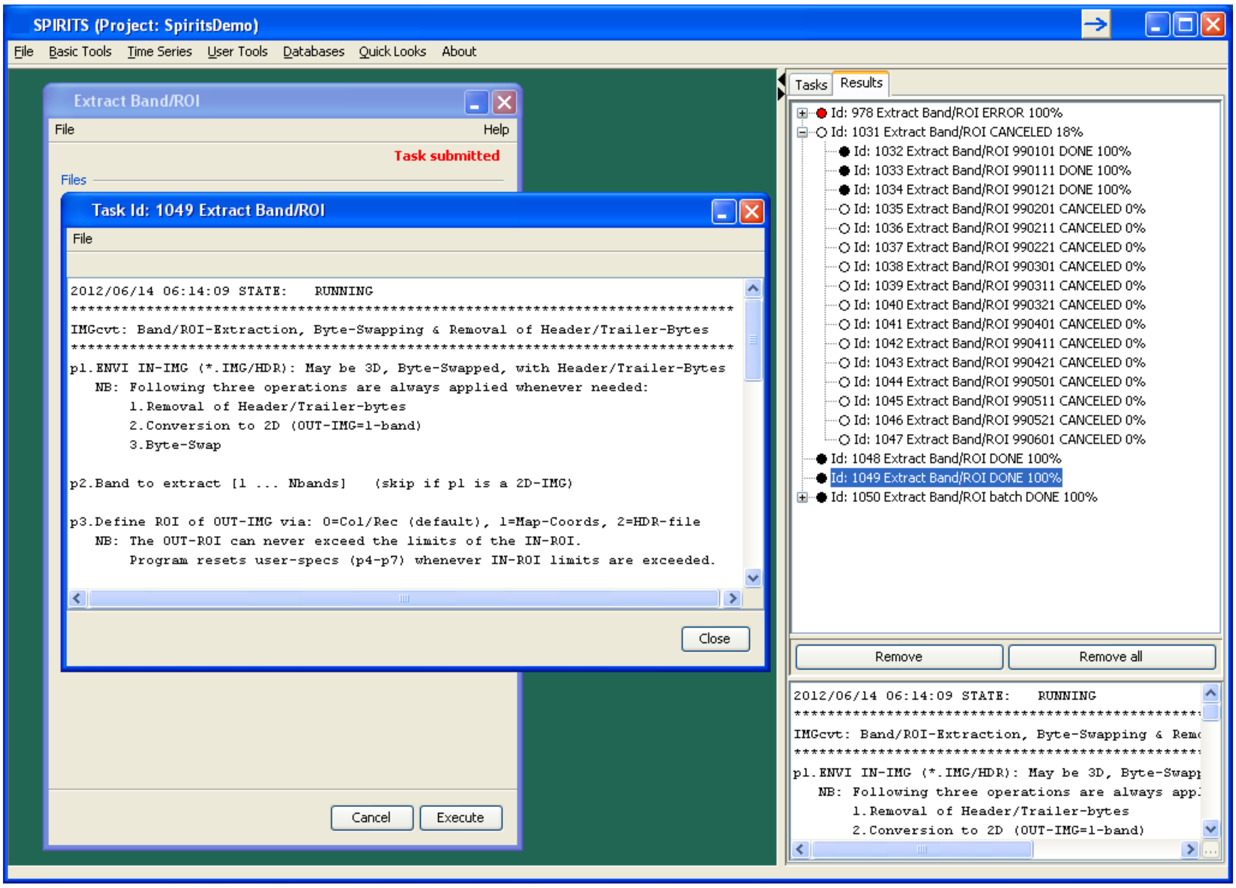
In the "Run new tasks" mode, new tasks entering the queue will be scheduled to be executed as soon as possible. Tasks enter the queue in the "READY" state, meaning ready to be processed.

In case of an empty queue this would mean a task entering the queue would start to be processed immediately.

In the "Pause new tasks" mode, new tasks will enter the queue in the "PAUSED" state. They will not be executed until they are resumed explicitly. This mode could be used for example to prepare a set of tasks, and resume them to be processed overnight.

1.3.4. Results queue

Upon completion, tasks leave the task queue and enter the results queue where they can be inspected.



The bottom half of the results queue shows the task log of the selected task.

By double-clicking a task, a separate panel will appear showing the selected tasks log. Detail information generated by the underlying module, e.g. error messages, can be inspected there.

Upon selecting another task in the results queue, the contents of the panel will be updated accordingly.

Via the file menu of the panel, this content can be saved in an ASCII text file.

The Remove button allows the deletion of the log of a selected task, the Remove all button clears the complete results queue.

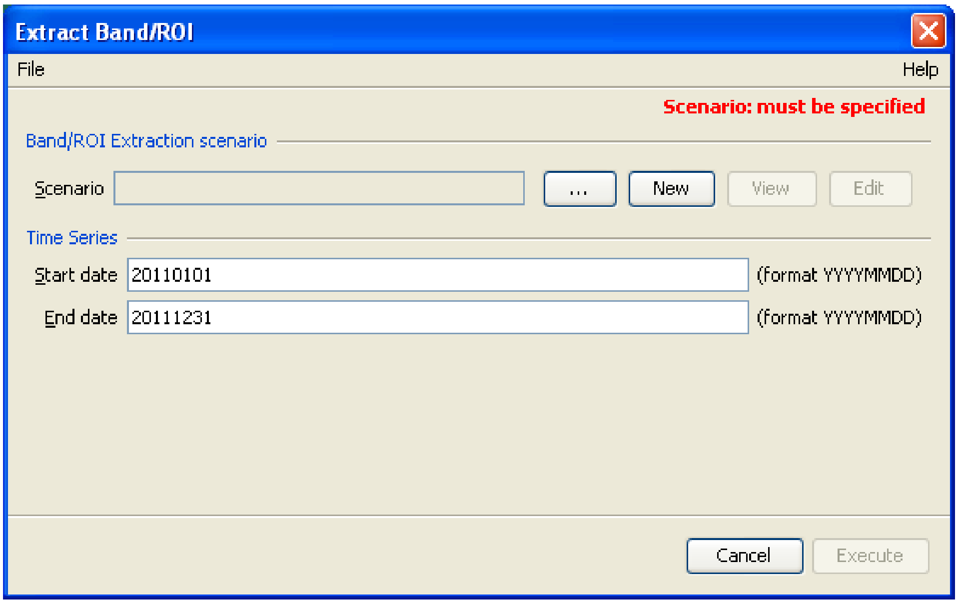
1.4. Time Series

1.4.1. Time Series UIs

Time Series tools allow the creation of tasks over time series of input files.

The parameters of a time series tool will typically be:

- the start date;
- the end date;
- all other parameters, requested by the specific tool. In most cases these specific parameters will be specified separately in a so-called "scenario".



Scenario containing specific parameters

Start date

End date

example of a typical time series UI

The start an end date parameters are specified in a ***YYYYMMDD*** format, YYYY being the four-digit year, MM the two-digit month and DD the two-digit day in the month.

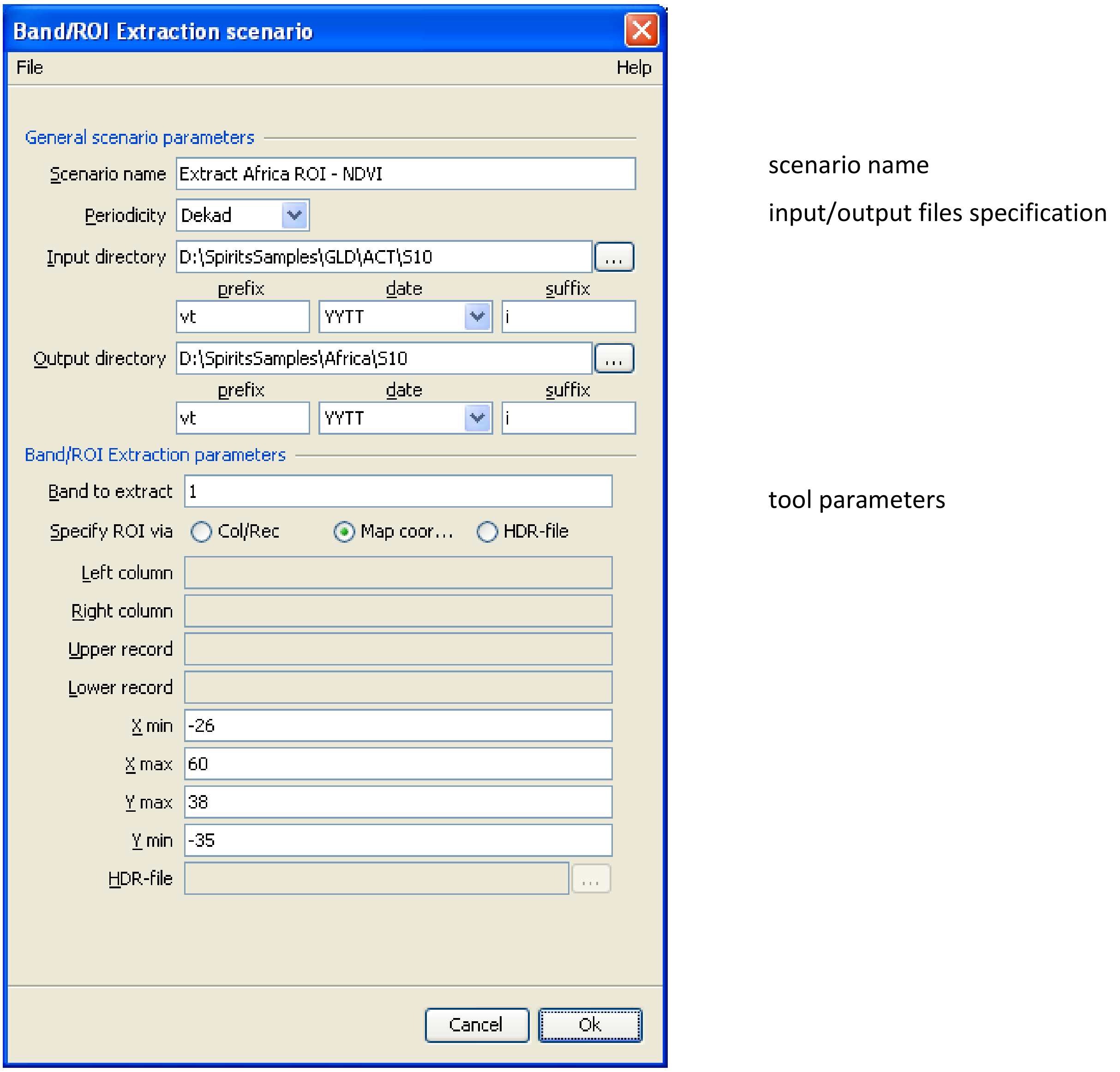
The scenario parameter will show the name of the selected scenario. Scenarios can be created in situ via the New action button, or an existing scenario can be chosen via a chooser. The selected scenario can be inspected via the View action button, or modified via the Edit action button.

The File menu functions in the same way as for the other tools, allowing the entered tool parameters to be saved in a TNT file which can be reused later.

1.4.2. Time Series Scenarios

1.4.2.1. General

The parameters, specific for a time series tool, are grouped into a scenario. Scenarios can be created via scenario UIs which resemble those of their corresponding tools, apart from the scenario name and the specification of input and output files.

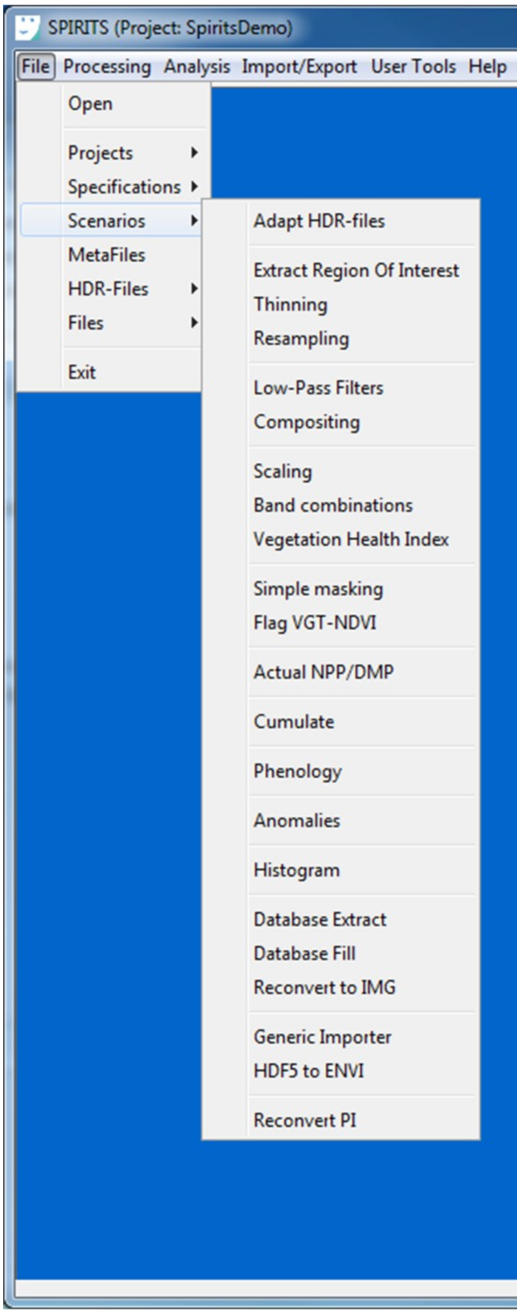


example of a typical scenario UI

The File menu allows the scenario to be saved, or to restore a scenario saved earlier, so it can be modified and reused.

Scenarios are stored in files using a "sns" extension. They are called SNS files for short.

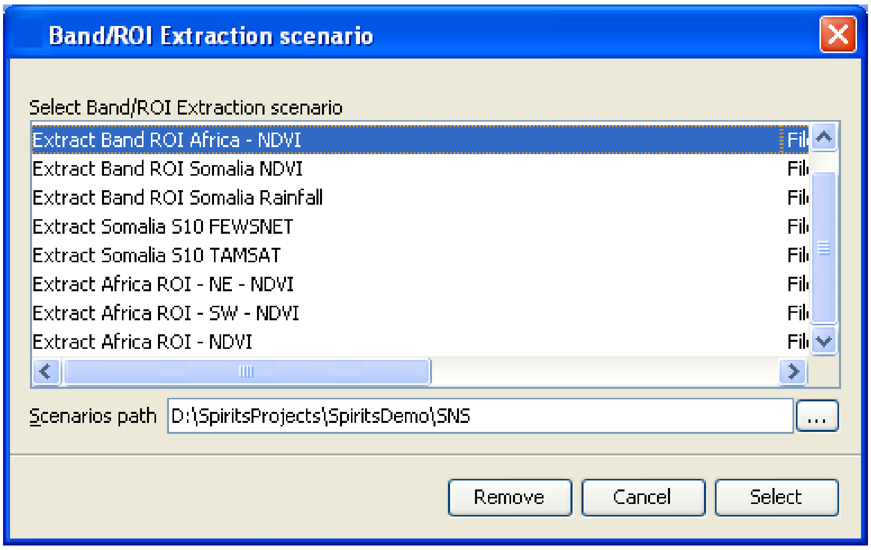
Scenarios can be created via the New action button in a time series UI, or from the File menu of the Spirits application window.



1.4.2.2. Scenario selection

The first scenario parameter will typically be its name. In practice, scenarios will be re-used or adapted frequently. Therefore it is advisable to assign a well chosen name to each scenario, with a clear description which makes it easily recognisable.

In the time series UIs, a scenario will be selected via a chooser which shows the names of the scenarios available for the specific time series tool.



example scenario selection

The scenario selection panel shows all scenarios available for the specific time series tool, from the directory specified in its search path. This search path defaults to the scenario files directory, specified in the Project settings, but it can be changed to select scenarios from another directory.

1.4.3. Time series file name specification

The input and output files which can be used in scenarios in particular and in time series in general need to be periodic and their names need to follow a file naming convention.

- the supported periodicities are Day, Dekad, Month and Year.
- the file naming convention is: ***PrefixDateformatSuffix.Extension***

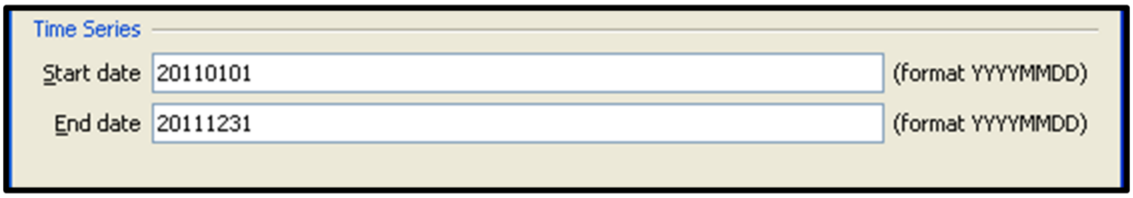
In the scenario or time series UIs, the file name specifications are defined by selecting the periodicity and specifying values for Prefix, Dateformat and Suffix. Extensions are mostly fixed in the context of the tool.

The Dateformats supported are specified in the table below:

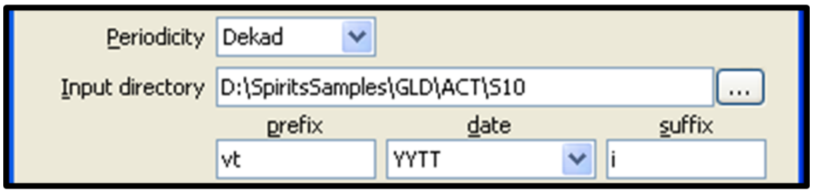
Date format	Minimal Periodicity	Explanation of terms			
YYYYMMDD	Day	YYYY	year	[1950	...2049]
YYMMDD		YY	year	[50 (=1950)	...49 (= 2049)]
YYYYmDD		MM	month in year	[01	...12]
YYmDD		m	month in year	[A (= january)	...L (= december)]
YYYYTT	Dekad	TT	dekad in year	[01	...36]
YYTT		DD	day in month	[01	...31]
YYYYMM	Month				
YYMM					
YYYYm					
YYm					
YYYY	Year				
YY					

Remark: The YYYY values from 1950 till 1964 are used by the Long Term Average tool to code historical IMGs representing averages from multi-annual IMGs sets. E.g. 1962 is used as code for IMGs containing the mean values from multi-annual IMGs sets. These 'coded' IMGs themselves are used in other tools, e.g. the Difference tool. Therefore it is recommended not the use these YYYY values for other purposes.

The input and output files are determined by executing a loop from the time series start date till its end date, with an interval according to its periodicity, and coding the date obtained from the loop in the Dateformat specified.



time series start and end date parameters



scenario/time series file name specification

```
D:\SpiritsSamples\GLD\ACT\S10\vt1101i.img
D:\SpiritsSamples\GLD\ACT\S10\vt1102i.img
D:\SpiritsSamples\GLD\ACT\S10\vt1103i.img
...
D:\SpiritsSamples\GLD\ACT\S10\vt1136i.img
```

resulting files

example: time series/scenario file name specification

Remark: input files which do not meet the naming convention, but do include date information in their filenames, can be copied and renamed via the Rename files utility.

1.4.4. Time Series execution

When the parameters of a time series tool have been filled out, it can be submitted for processing via the Execute action button.

As for other tools, a task will be created and sent to the task queue to be processed.

The time series task itself will comprise a series of subtasks, one for each time-step determined by the start date, end date and periodicity parameters.

The actual start date considered, will be the start date specified by the user, rounded downward according to the periodicity specified. E.g. In case of Dekad periodicity, start dates specified between YYYYMM01 and YYYYMM10 will yield an actual start date YYYYMM01, representing dekad 1 in year YYYY, month MM.

The loop will run from this actual start date, in steps according to the periodicity specified. E.g. in case of Dekad periodicity, this would be the dates YYYYMM01, YYYYMM11, YYYYMM21, YYYY(MM+1)01, ...

The actual last date considered, will be the one determined by the steps, as described above, which does not exceed the end date specified by the user. E.g. In case of Dekad periodicity, end dates specified between YYYYMM21 and YYYYMM31 will yield an actual end date YYYYMM21, representing the last dekad in year YYYY, month MM.

2. Overview of the Spirits Functionalities

The Spirits menu system comprises three levels. All the menu entries are listed in the table below. For each endpoint or module, a brief description is included as well. More details can be found in the concerned sections of this User's Manual. The Level 3 entries starting with an asterisk (*) can be run in the standard mode (single tool) and in the time series mode. Of course the menu structure might be adapted in future versions of the software.

LEVEL 1	LEVEL 2	LEVEL3	FUNCTIONALITY
FILE	Open		Select and open Spirits specific files (*tnt, *.sns,...)
	PROJECTS	Select Define	Define/select Spirits projects
	SPECIFICATIONS	SPC, SPM, SPP, SPS, SPU	Create/edit SPX-files needed by some Glimpse modules
	SCENARIOS	List of Time Series modules	Create/edit SNS-files needed by the Spirits time series
	METAFILES		Create ASCII-files with list of images: VAR-files for Glimpse, MTA-files for ENVI
	HDR-FILES	View/edit * Adapt	View/edit/adapt HDR-files with image annotation
	FILES	Rename	Generic file renamer
	EXIT		Quit the program
PROCESSING	SPATIAL	* Extract region of interest	Extract ROI, but also band selection, byte-swapping,...
		* Thinning	Degrade the resolution of an image, using different filter types
		* Resampling	Modify the framing (extension/resolution) of an image
		* Low-Pass Filters	Low-pass, smoothing filters, using a moving window
		* Mosaicing	Mosaicing of IMGs covering different zones
		Compositing	Generic spatio-temporal compositing
		Area Fraction IMGs	Create LowRes AFIs from a HighRes classification
		* Reproject	Reproject images
	THEMATIC	* Scaling	Rescaling, reclassification, stretching, modification of data type
		* Band combinations	Different combinations of 2 or 3 IN-images
		* Band calculator	Algebraic operations on IN-images
		* Vegetation Health Index	Kogan's VHI, with some specific facilities
		* Simple masking	Flag an image with the info from a mask image
		* Flag VGT-NDVI	Specific for the NDVI of SPOT-VGT, using the status mask
		Maximum NPP/DMP	Monteith approach applied on LowRes Meteo-IMGs
		* Actual NPP/DMP	Actual = MaxNPP/DMP * fAPAR
		Clustering	Non-supervised classification with modified ISOclus algorithm
	TEMPORAL	Smoothing	Clean time series of NDVI, fAPAR,... with modified Swets algorithm
		Frequency analysis	Count frequency of an "event" in a series of IN-images
		Compositing	Generic spatio-temporal compositing (see SPATIAL)
		* Cumulate	Cumulate/average between two fixed dates
		Time statistics	Derive IMGs with Min/Mean/Max over a series of IN-images
		* Phenology	Define start/end of seasons (SOS/EOS) from a dekadal IN-series
		Progress of season	Relative progress of current date between SOS and EOS
		Cumulate over season	Cumulate/average between (pixel-specific) SOS and EOS (or today)
		Long-term statistics	Derive "historical images" from a multi-annual image set
		* Anomalies	Difference of actual IMG vs. IMG of historical/previous year
		SPI	Standardized Precipitation Index
		Similarity analysis	Define "most similar year" from multi-annual image set
		Sim-based yield assessment	Similarity based yield assessment

LEVEL 1	LEVEL 2	LEVEL3	FUNCTIONALITY
ANALYSIS	IMAGES	* Histogram	Derive ASCII HIS-file with Image histogram
	MAPS	Create templates	Setup & layout: image type, legend, vectors, logos,...
		* Map series	Generation of quicklook maps by application on a series of images
		* ROI series	Generation of quicklook maps by application on a series of ROI's
	DATABASE	Sensors	Preliminary definition of the basic database components
		Variables	
		Regions	
		Classes	
		* Extract	Extract ASCII RUM-files with "Regional Unmixed Means"
		* Fill	Ingest the RUM-values in the database
		Browse	Browse the database & elementary analysis tools
		* Reconvert	Reconvert ASCII RUM-files to images
	CHARTS	Chart template	Setup & layout: chart type, axes, colours, ...
		Matrix template	Setup & layout for matrix chart: data, axes, colours, ...
		Scatter template	Setup & layout for scatter chart: data, axes, colours, ...
		Chart series	Generation of a series of charts
IMPORT EXPORT	IMPORT	* Generic importer	GDAL based conversion of many external formats to ENVI
		* HDF5	Convert HDF5 dataset to D ENVI-IMG
		* ENVI-3D	Extract 1 band from a 3D ENVI-IMG
		* IDRISI image	Convert IDRISI image to ENVI
		* IDRISI annotation	Convert IDRISI DOC-file to ENVI HDR-file
		* Meteodata	Convert ASCII-files with meteodata to IMG-format
		Convert periodicity	Convert frequencies of time series to S1/S10/S30
		LF2CRLF	Replace UNIX-LF (ASCII 10) delimiters by Windows CRLF (13-10)
	EXPORT	ENVI-3D	Combine a set of 2D IMGs into a single 3D IMG (BSG, BIL, BIP)
		* ArcGIS	Convert ENVI IMG to format of ArcGIS
		* IDRISI image	Convert ENVI IMG to IDRISI format
		* IDRISI annotation	Convert ENVI HDR-file to IDRISI DOC-file
		* Descale IMG	Convert IMG to float datatype and remove scaling
	PI-COMPRESSION	Create PI-Template	Template to extract specific ROI from compressed Pseudo-IMGs
		* Reconvert PI	Extract ROI in normal format from global Pseudo-IMGs
	VECTORS	Rasterize SHP-file	Generate raster version of an ESRI Shape-file
		Reproject SHP file	Reproject ESRI Shape-file
USER TOOLS	COMMAND		Execute command line
	DEFINE TOOL		Define the program(s) to run and their parameters
	EXECUTE TOOL		Execute the tool on a specific data set
HELP	MANUAL		Access the User's Manual
	TUTORIAL		Access the Tutorial (if present)
	ABOUT SPIRITS		Brief info on developments, funding, disclaimers, etc.
	System		Brief info on Java and Spirits versions and Memory state.

3. Tools and Time Series

3.1. Extract Band/ROI

Goal

Extract a “Region of Interest” (ROI) from an IMG. If the input IMG has multiple bands (3D), one of the bands must be selected.

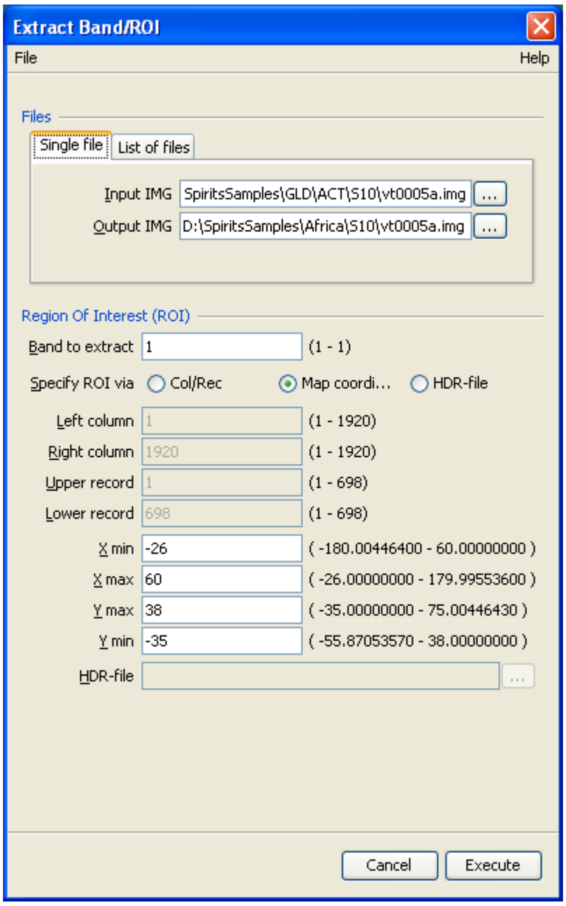
Parameters

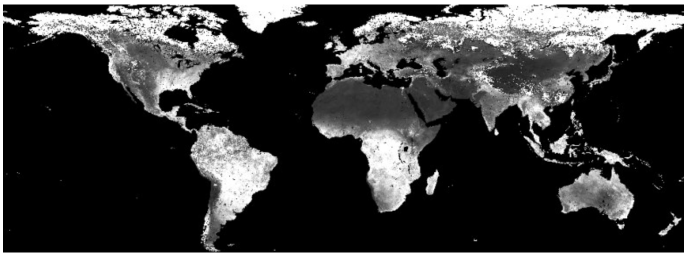
- the IMG band to extract from;
- the ROI to be extracted.

The ROI can be specified three ways:

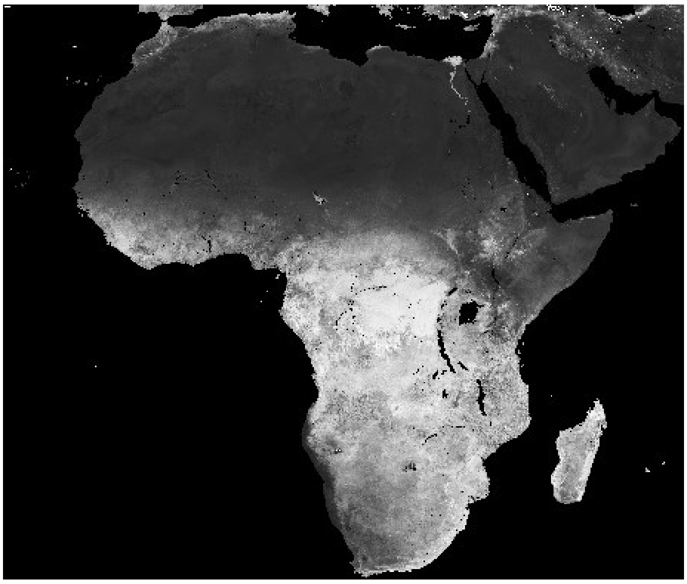
- in terms of IMG coordinates (Columns/Records);
- in terms of Map coordinates (X/Y or Lon/Lat);
- in terms of Map coordinates using the ROI-limits specified in an existing HDR file.

Tool





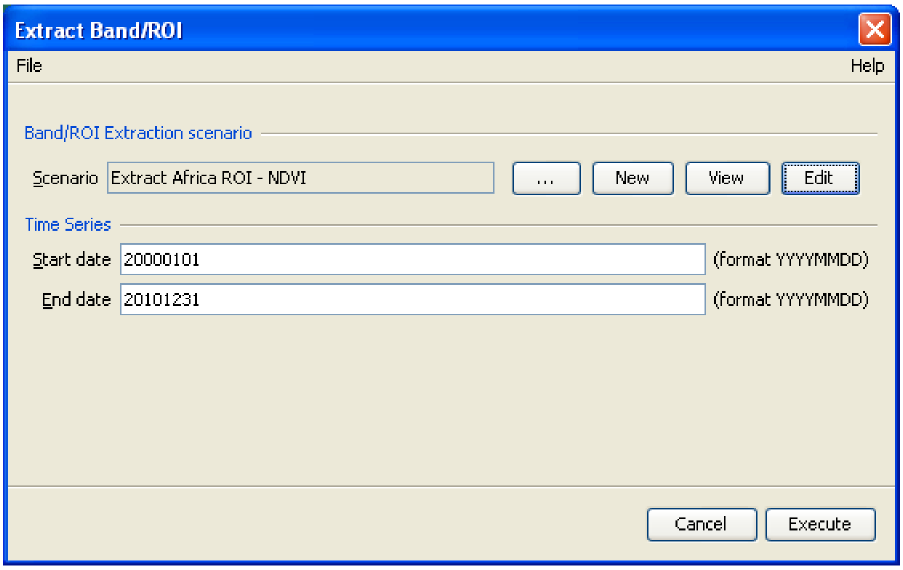
input IMG



Extract Band/ROI Tool example

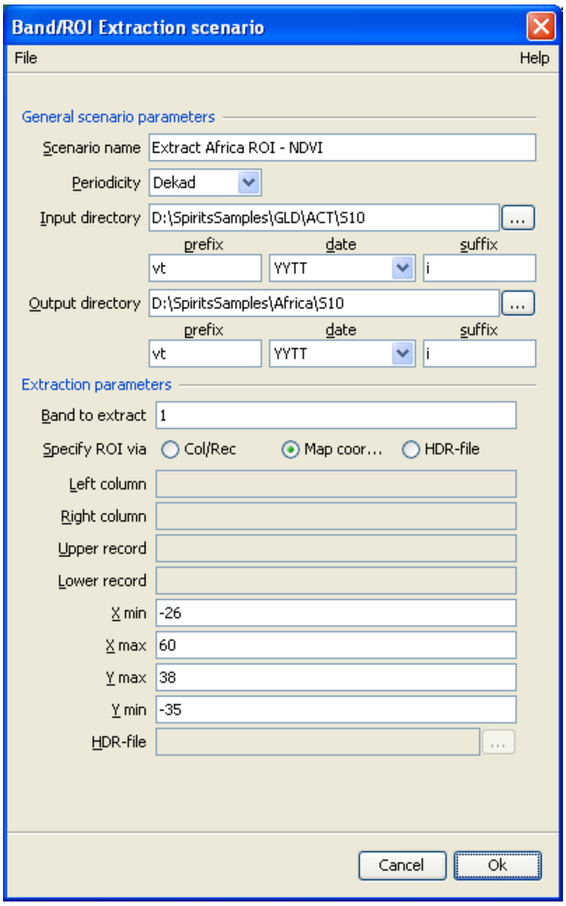
output IMG

Time Series



Extract Band/ROI Time Series example

Scenario



Extract Band/ROI Scenario example

3.2. IMG Thinning

Goal

Degrade the resolution of an IMG.

Parameters

- the ROI to be extracted;
- the thinning window size;
- the thinning filter type and its specific parameters.

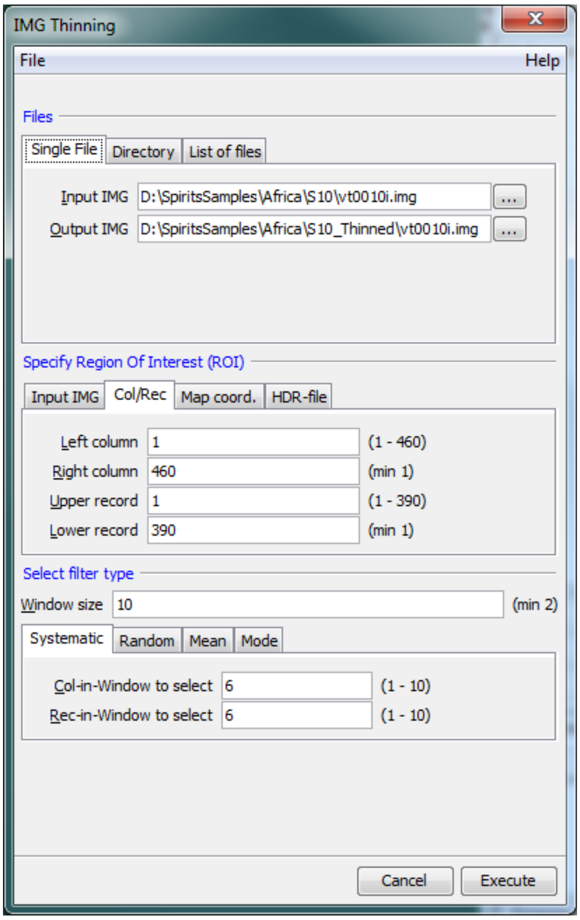
The ROI can be specified four ways:

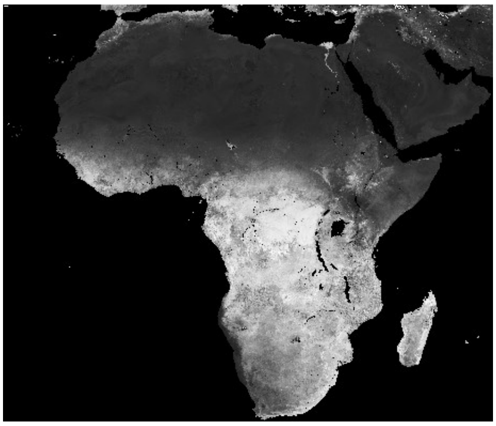
- from input IMG. The ROI is kept as is;
- in terms of IMG coordinates (Columns/Records);
- in terms of Map coordinates (X/Y or Lon/Lat);
- in terms of Map coordinates using the ROI-limits specified in an existing HDR file.

Four filter types are available:

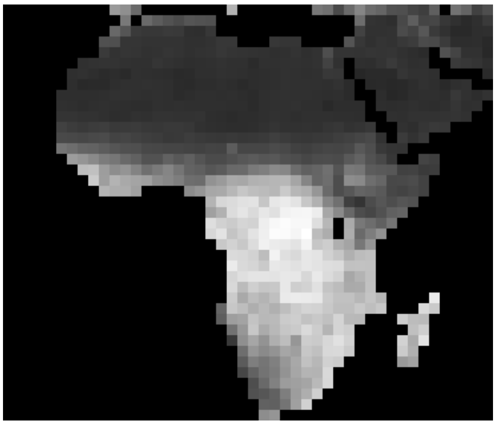
- Systematic;
- Random;
- Mean;
- Modus.

Tool





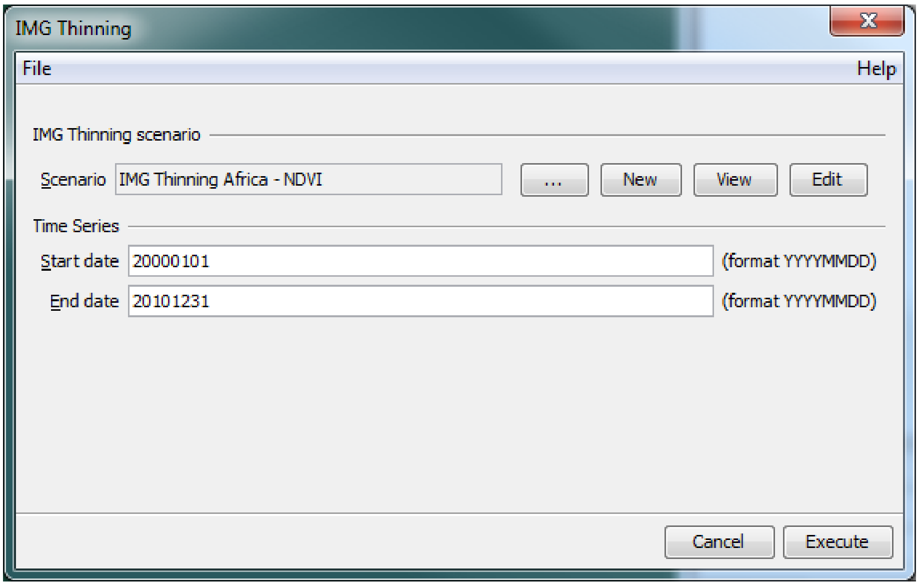
input IMG (460 x 390)



IMG Thinning Tool example

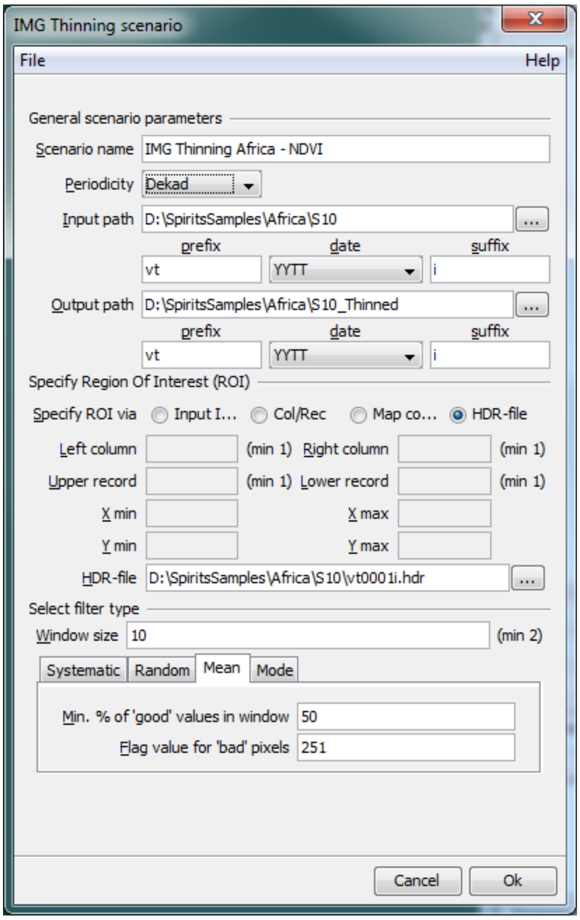
output IMG (46 x 39)

Time Series



IMG Thinning Time Series example

Scenario



IMG Thinning Scenario example

3.3. IMG Resampling

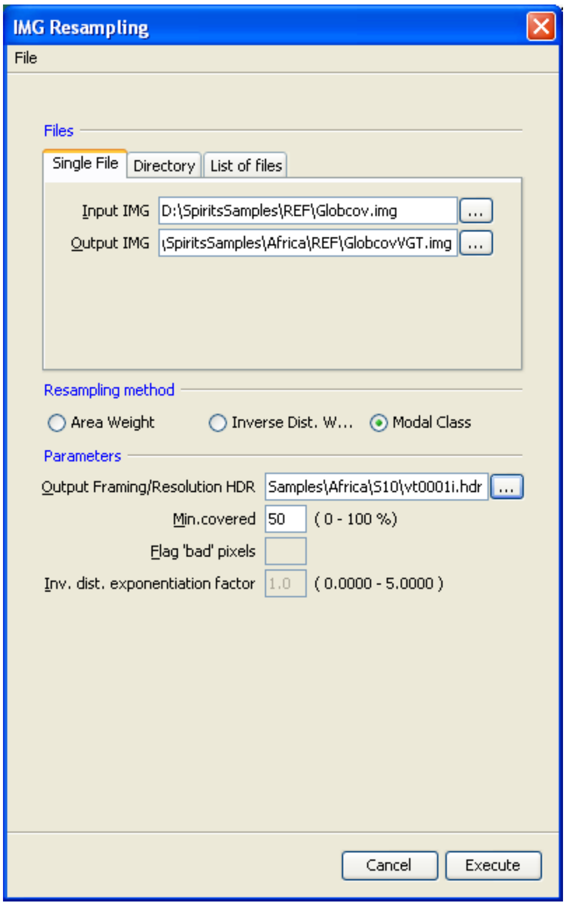
Goal

Create resampled IMG with modified resolution and/or framing. Remark: the map projection is not changed.

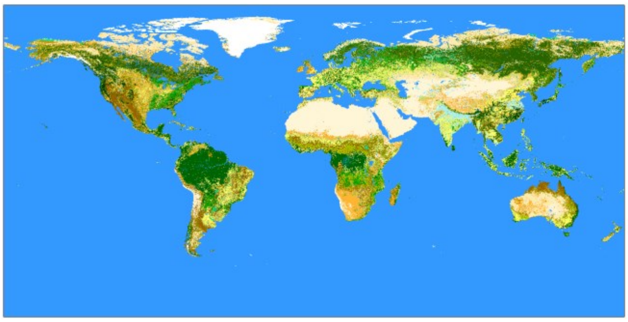
Parameters

- the input resampling method:
  - Area Weighted Mean Resampling (for ordinal images);
  - Inverse Distance Weighted Mean Resampling (for ordinal images);
  - Modal (predominant) class (for classification images - for this case the input IMG must be byte-type).
- a HDR file specifying the framing and resolution of the output IMG(s);
- the minimum % of an output pixel area which must be covered by good input values to be retained;
- the flag value to be used for output pixels not covered by the input IMG, or with their covered area below the specified minimum. For the modal method (classification) this flag is fixed at "0".
- only for the Inverse Distance Weighted Mean Resampling method: the exponentiation factor (0.0 - 5.0) to apply to the inverse distances, with 0.0 meaning no weighting, simple mean till 5.0, meaning very sharp distance weighting.

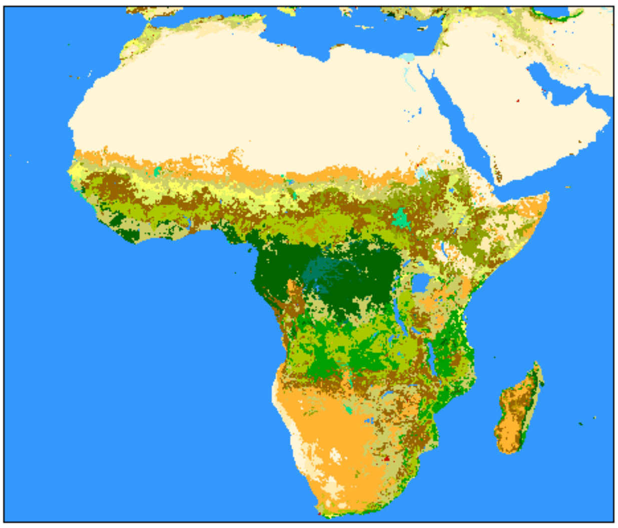
Tool



IMG Resampling Tool example



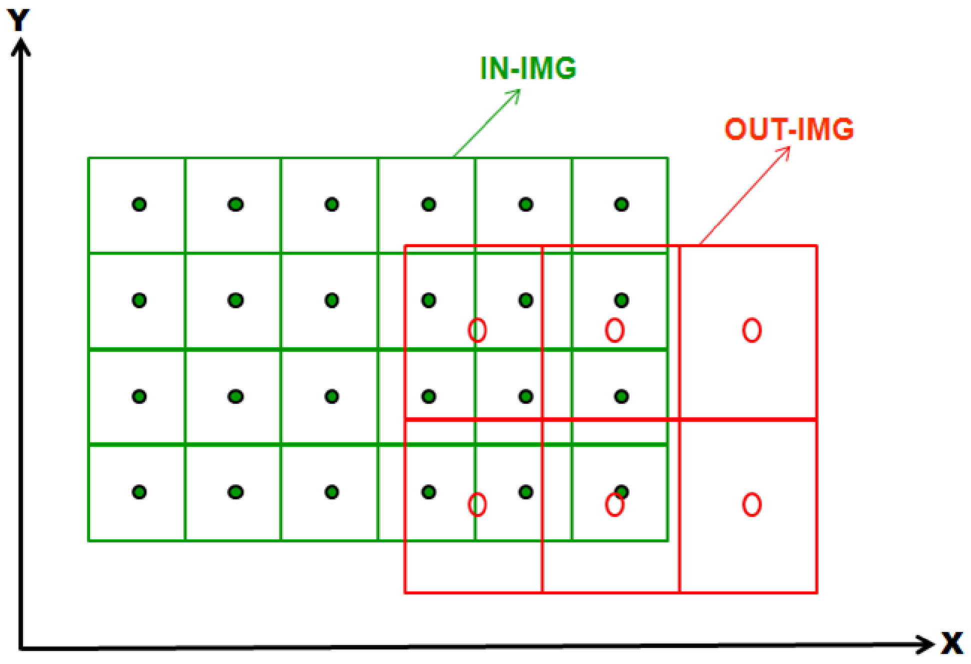
input IMG: map info = {Geographic Lat/Lon, 1.5, 1.5, -180, 90, 2.7778e-003, 2.7778e-003}



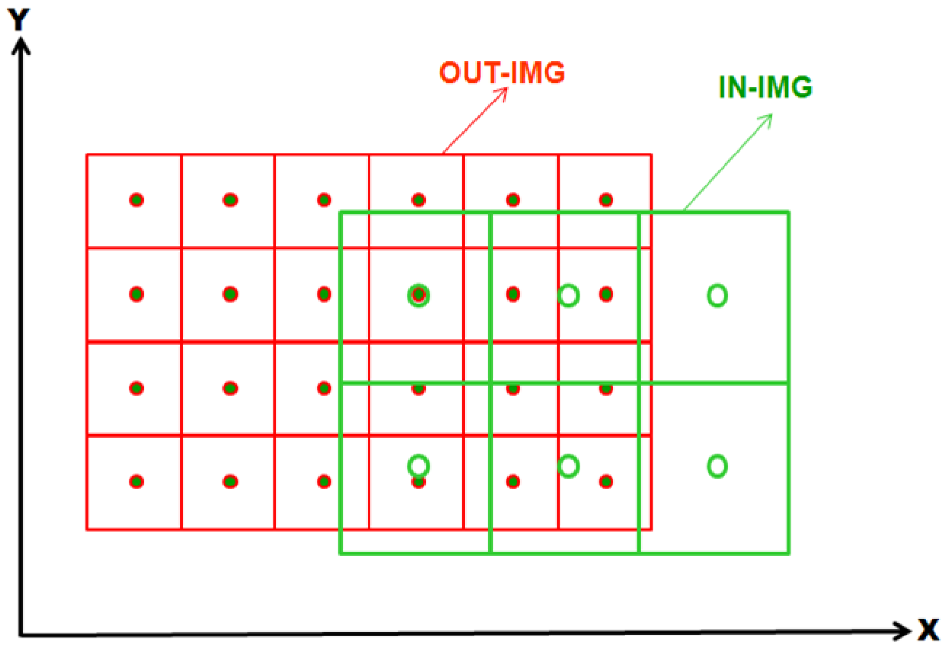
output IMG: map info = {Geographic Lat/Lon, 1, 1, -26.066964, 38.0669643, 0.1875, 0.1875}

Remarks:

- the map system of the specified HDR must be the same as for the input IMG;
- the resolution of the output IMG (specified via the HDR) may be lower or higher than the resolution of the input IMG;

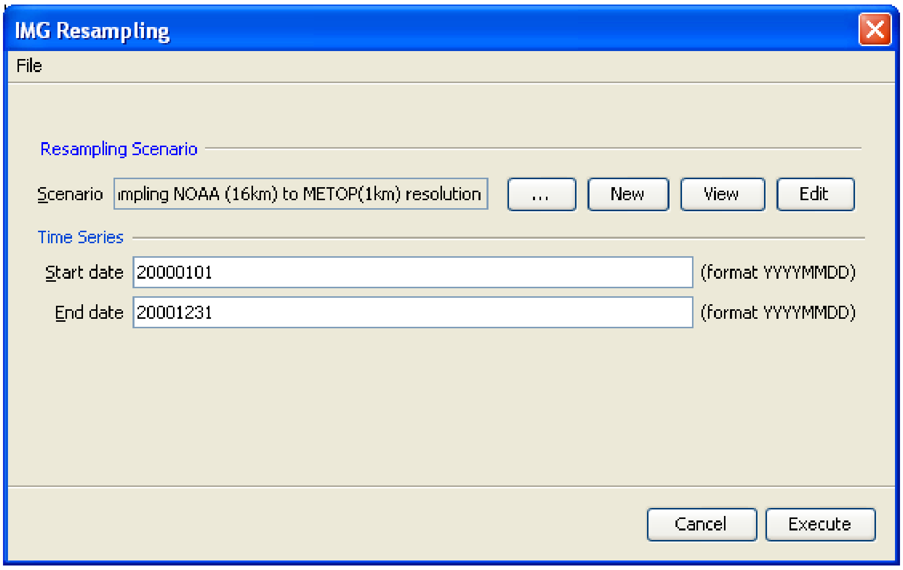


Resampling from higher to lower resolution



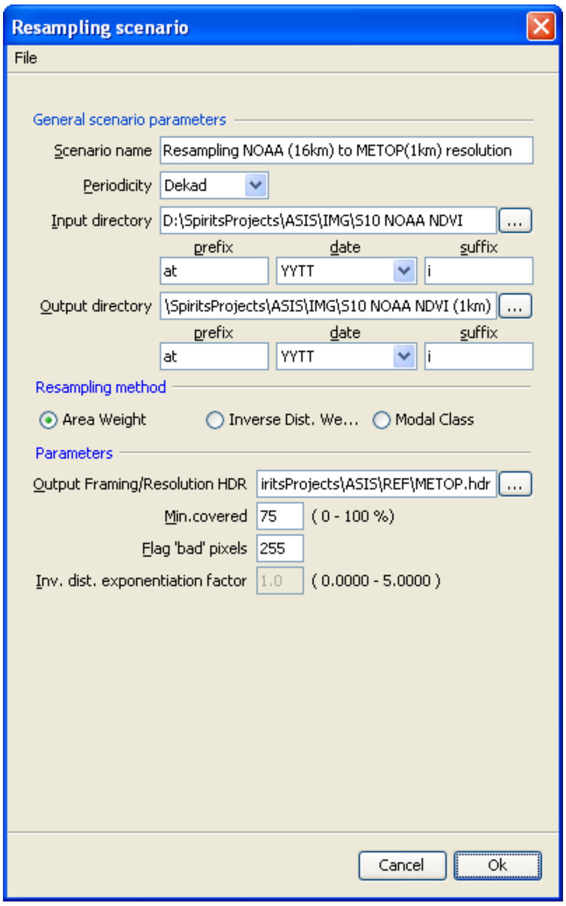
Resampling from lower to higher resolution

Time Series



IMG Resampling Time Series example

Scenario



IMG Resampling Scenario example

3.4. Low Pass Filter

Goal

Apply a low-pass smoothing spatial filter on an IMG, using a moving window.

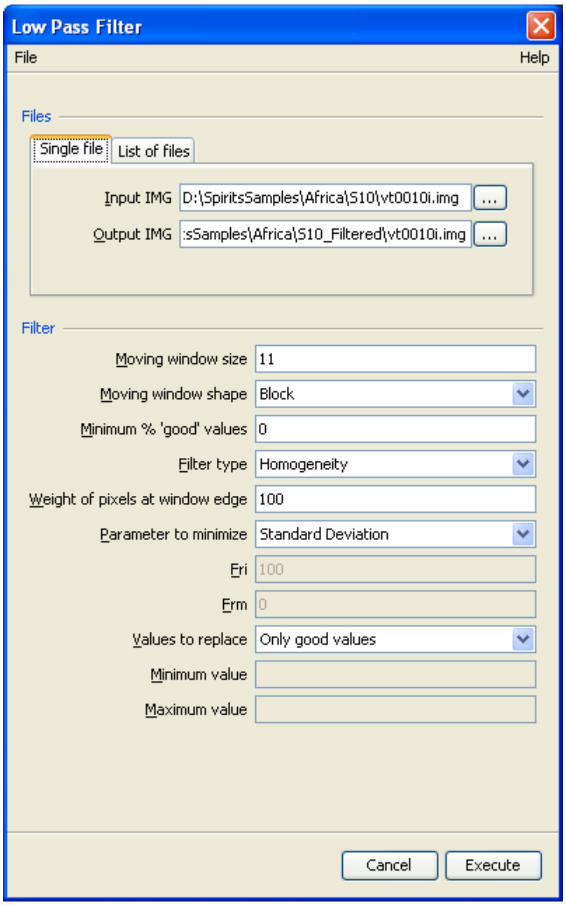
Parameters

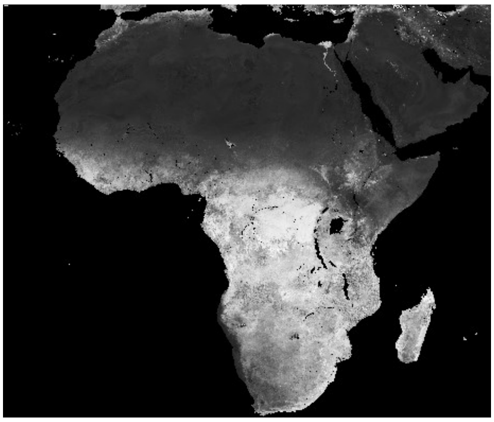
- the size of the moving window (must be an odd value);
- the shape of the moving window (block or circle - ignored if filter type is Homogeneity);
- the minimum percentage of non-flagged values per window;
- the filter type and its specific parameters.

Six types are available:

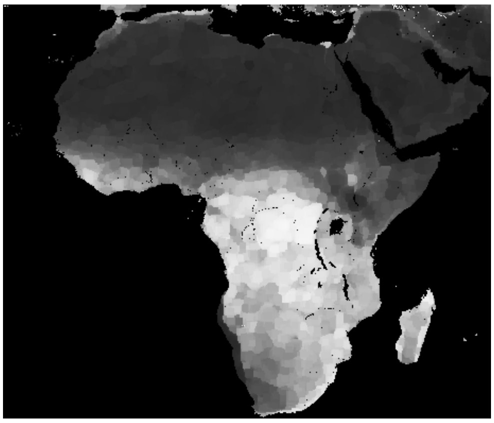
- Mean;
- Median;
- Minimum;
- Maximum;
- Homogeneity;
- Mode.

Tool





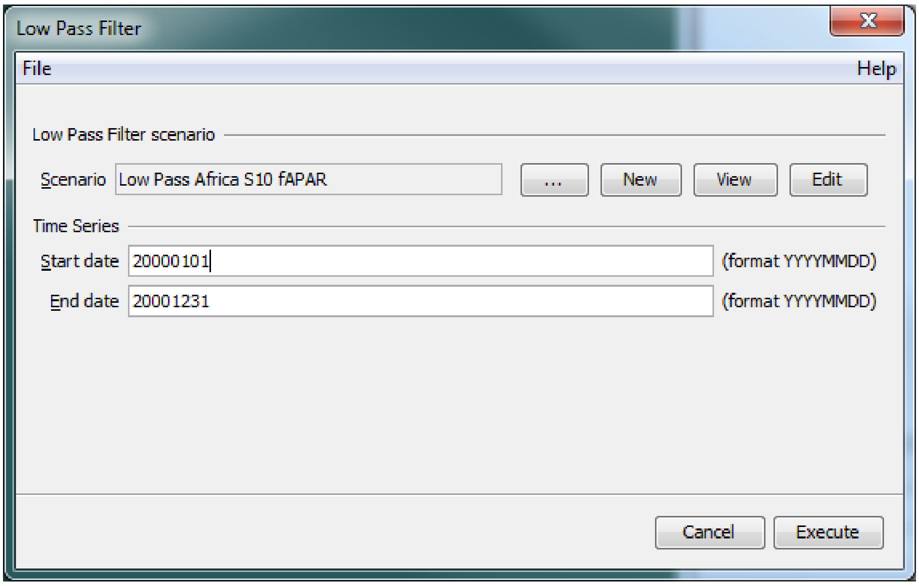
input IMG



Low Pass Filter Tool example

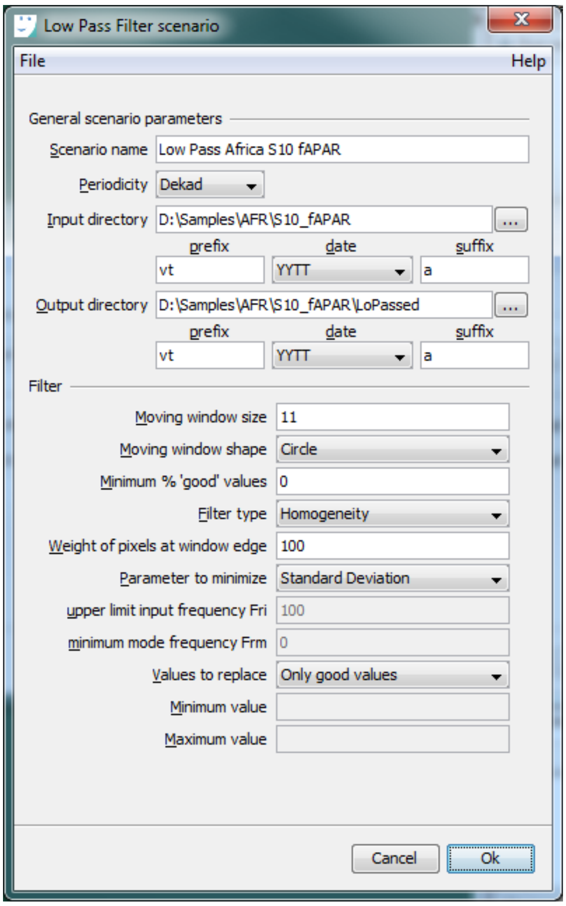
output IMG

Time Series



Low Pass Filter Time Series example

Scenario



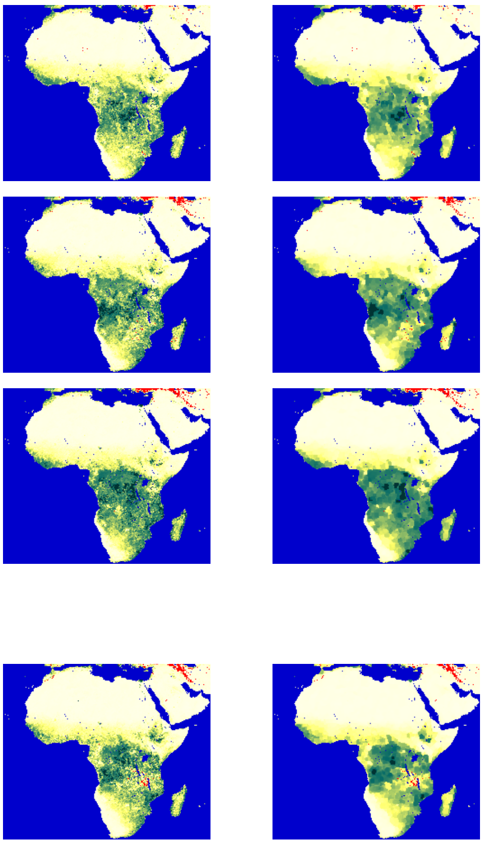
Low Pass Filter Scenario example

dekad 1 2000

dekad 2 2000

dekad 3 2000

dekad 36 2000



input IMGs output IMGs

Exampe: filtering fAPAR time series

3.5. Mosaic

Goal

Mosaicing of a collection of IMGs covering different zones.

The IMGs must have identical spatial resolution, but may have different spatial extensions (typically tiles) and can overlap.

Parameters

- the collection of input IMGs. These can be specified:
  - via a VAR file containing the filenames of the input IMGs;
  - via the directory containing the input IMGs and a filename pattern (mix of constant characters with '*' and '?' wildcards);
  - via an LNL file containing the filenames of the input IMGs;
- whether or not to allow variations of the spectral coherence of the input IMGs. All input IMGs datatype, scaling (Vint/Vslo) and Classes (if applicable) must always be identical. They must also have identical spatial resolution, without sub-pixel shifts. Variations in spectral features Vlo/Vhi, Vname/Vunit and Flags can be allowed or forbidden;
- the output ROI. The ROI of the output IMG can optionally be specified via a HDR file. In case no explicit output ROI is specified, the output region will be the bounding box covering all input IMGs;
- the compositing rule to be applied in case of overlapping pixels. This can be:
  - the minimum
  - the maximum or
  - the mean value of all (non flagged) pixel values
- a mask IMG. Optionally a mask IMG (must be BYTE type) can be specified (typically Land/Sea). In case no mask is specified the output IMG will contain only a "missing value" flag (see further). In case a mask is specified, pixels with mask values [1-255] will be considered as land, and will be treated normally, whereas pixels with mask value 0 will be considered as sea, and will be flagged as such in the output IMG. A mask IMG can only be specified in case the output ROI has been specified. This mask IMG must be spatially congruent with the ROI. Typically the same IMG would be used for both the mask and ROI.

Remark: detailed features of the output IMG:

- As a general rule, the mosaic inherits all the spectral features of the IN-IMGs:
- Datatype, scaling (Vint/Vslo) and number of classes (mostly 0) must always be identical anyway.
- If the IN-IMGs are classifications (classes > 0), the class names and colours are taken over from the first IN-IMG and assumed to be identical for the others.
- Significant range (Vlo/Vhi) and the name and units of the concerned image variable (Vname, Vunit): If variations do occur and if they are allowed, the solution is as follows:
  - For Vname, Vunit, the longest string is searched amongst all IN-IMGs and used for the OUT-mosaic.
  - For Vlo the lowest value is retained, for Vhi the highest one.
- The output flags are fixed in function of the datatype and whether or not a mask IMG is used. Without such mask, the mosaic will only contain one single "missing" flag. If this flag overlaps with the significant

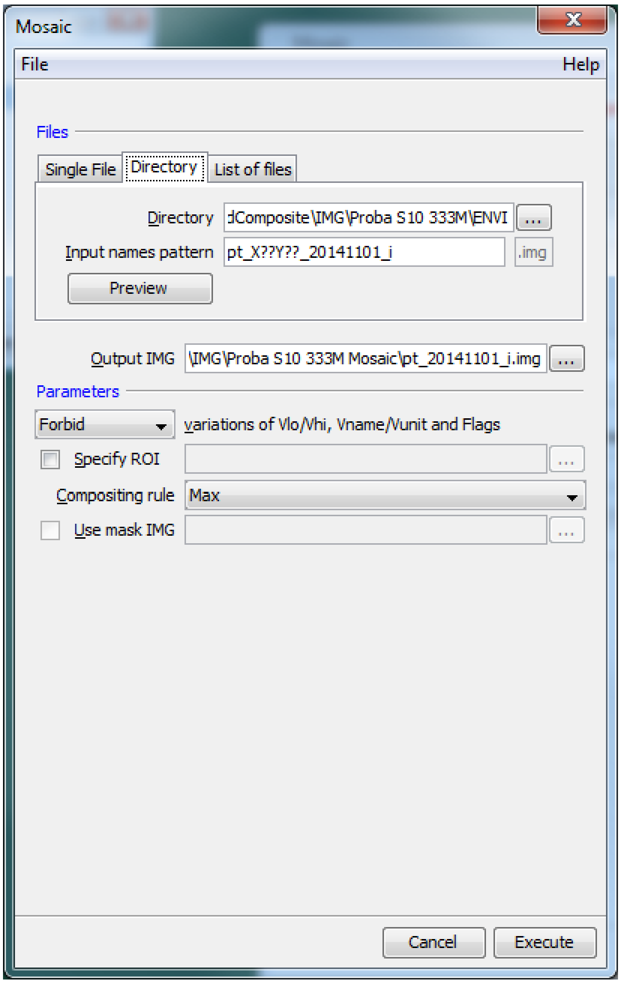
range Vlo-Vhi, this range is adapted (for the mosaic). Examples: If Vhi=255 (BYTE) it is reset to 254, if Vlo=-32768 (SHORT) it is reset to -32767, etc. This measure may yield some saturation in the resulting IMG.

- If a mask IMG is specified, the mosaic will contain two flags (“sea” and “missing over land”) conform to the UNIflags system.

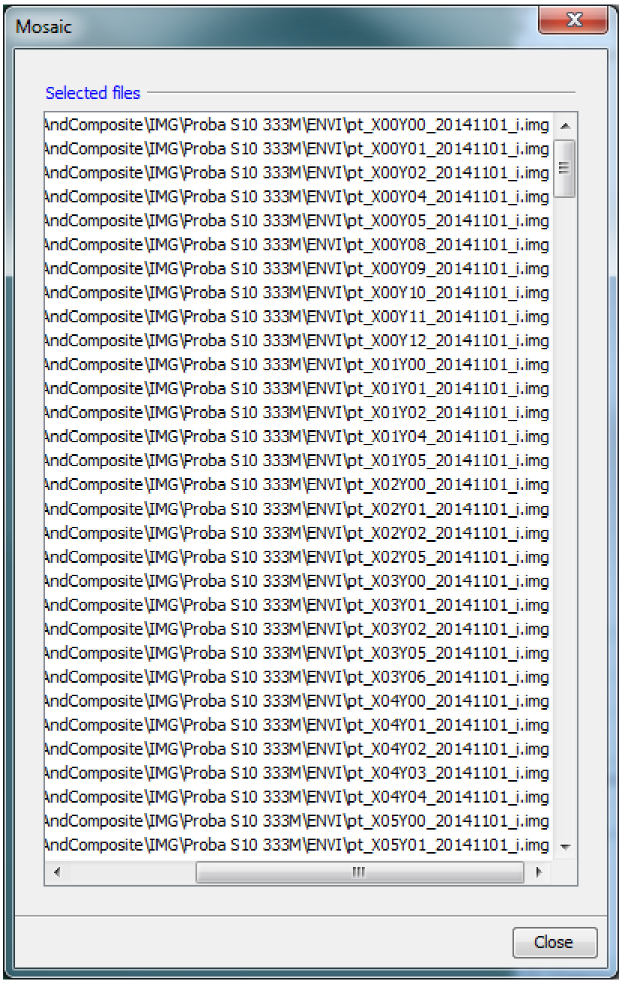
LAND/SEA MASK	FLAG	BYTE	SHORT	LONG & FLOAT
none	Missing	255	-32768	-1000005
	Possible adaptations	Vhi = 254	Vlo = -32767	Vlo = -1000000
specified	Sea	254	-2	-1000002
	Missing over Land	251	-5	-1000005
	Possible adaptations	Vhi = 250	Vlo = 0	Vlo = -1000000

Flags and possible Vlo/Vhi adaptions of the resulting mosaic IMG

Tool



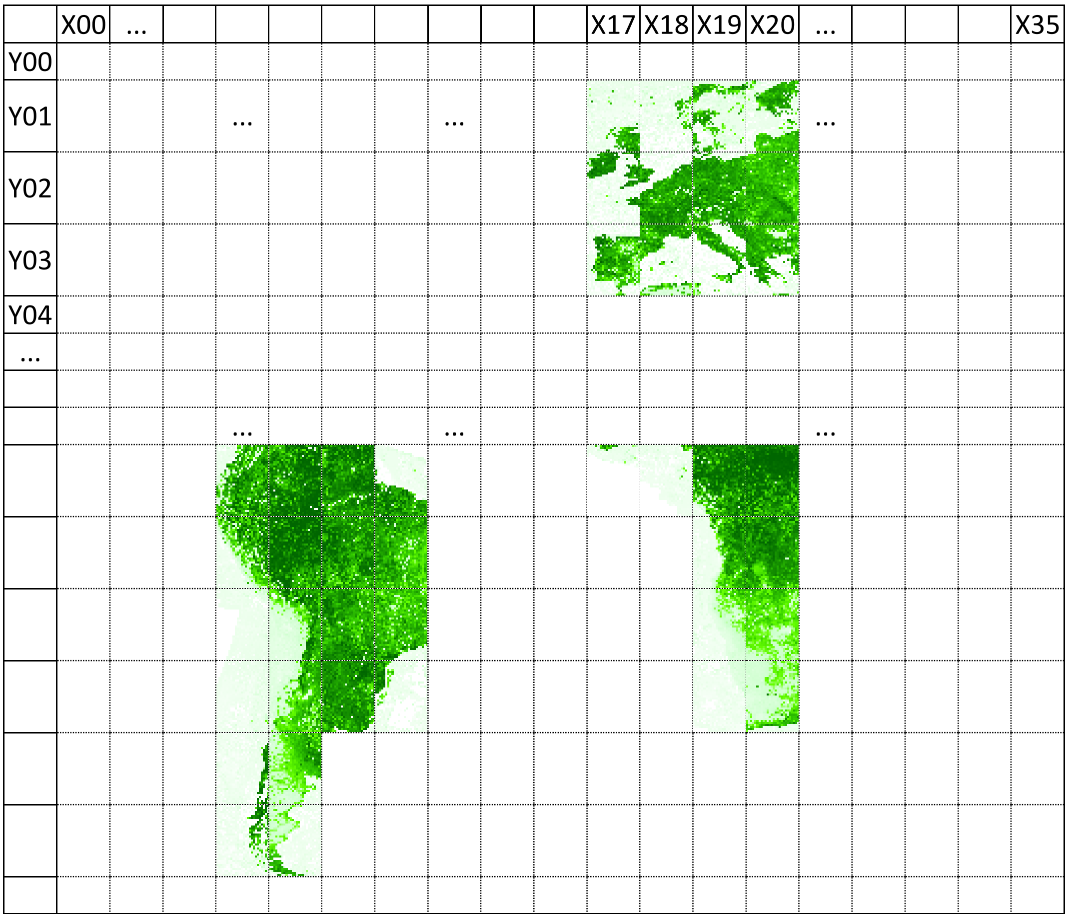
Mosaic Tool example



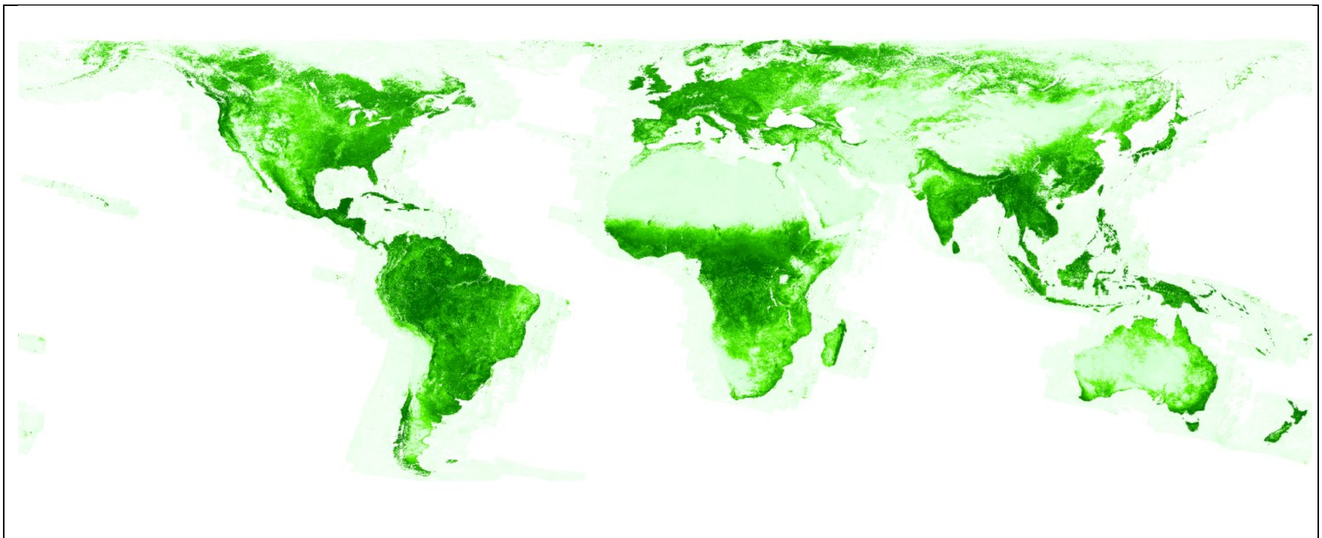
Mosaic Tool - Preview selected files

Proba-V tiles mosaic example

Proba-V tiles are mapped on a grid with in X-direction tiles from 00-35, in Y-direction 00-13. These coordinates are used in the tiles filenames. By using a wildcard pattern "...X**??**Y**??**.." as in screenshot above, the tiles can be selected to be mosaiced.



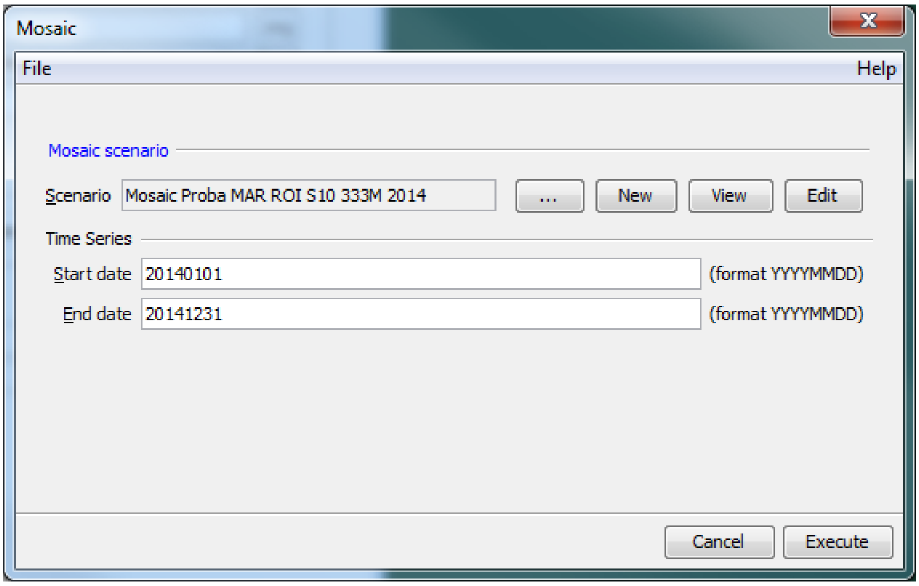
Proba-V Tiles



Resulting mosaic

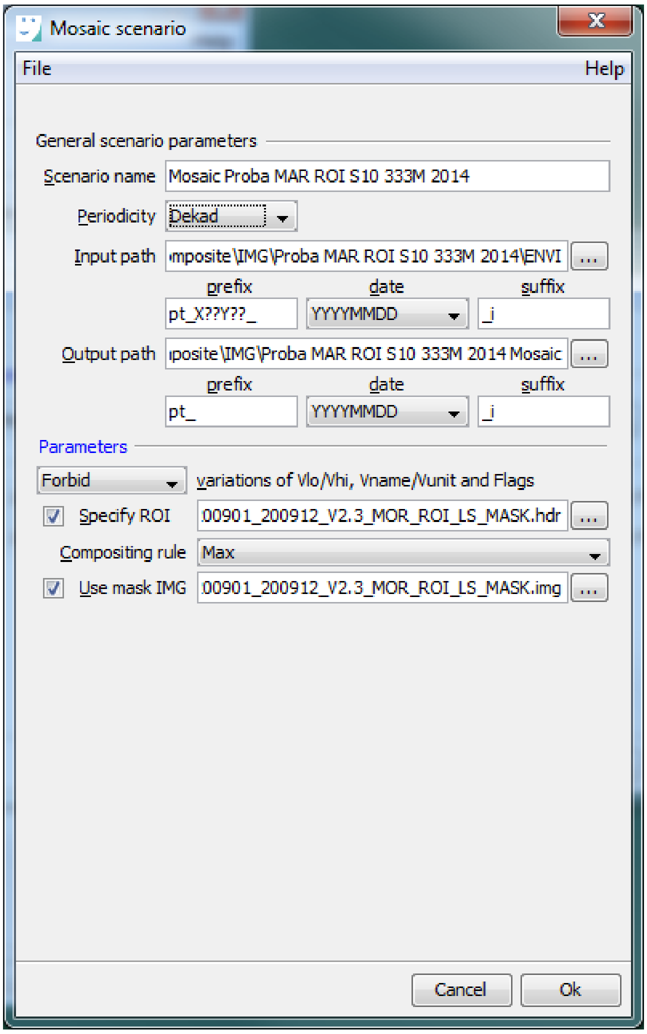
Time Series

The Mosaic time series tool creates a time series of mosaics. For each time slot in the time series, a collection of IMGs is selected, which will be mosaiced.



Mosaic Time Series example

Scenario



Mosaic Scenario example

The mosaicing scenario supports the use of wildcards ('*' and '?') in the prefix and suffix fields specifying the input files.

The time series creates a loop between the dates specified in the time series tool, with a step according to the periodicity specified in the scenario.

For each of these dates, all input files matching the filename pattern specified by the combination of the prefix (with wildcards), the suffix (with wildcards) and the date formatted according to the date format selection in the scenario, are grouped as collection of input IMGs for the mosaic, for that date.

Proba-V Morocco tiles mosaic time series example

In this example, the input path contains for each dekad in 2014, the four Proba-V tiles (coded X16Y03, X17Y03, X16Y04 and X17Y04) covering the bounding box of Morocco. The filenames have been coded (during an HDF5 Import step) as ***pt_XxxYyy_YYYYMMDD_i***.img/hdr.

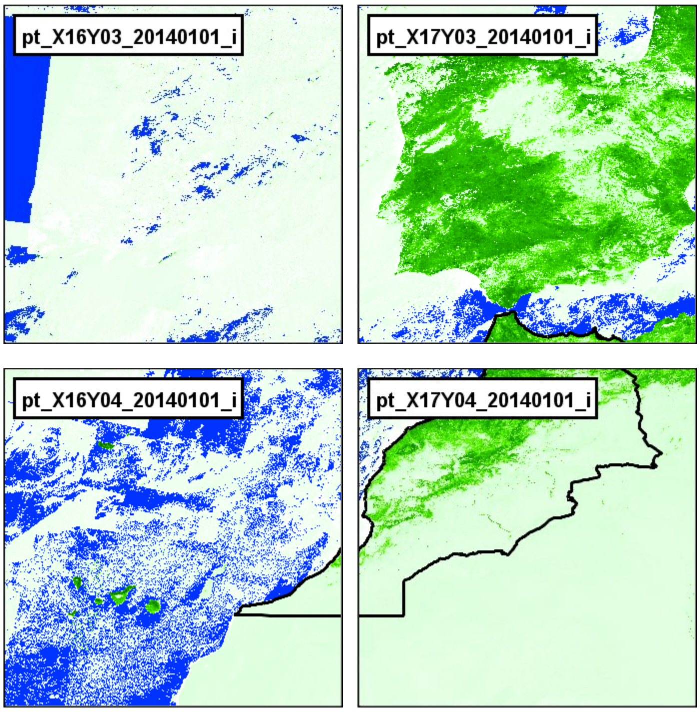
The time series creates a loop over the periodic (dekadel) dates between start and end date.

These dates are formatted according the selected data format, here ***YYYYMMDD***.

For each of these dates, a mosaic is created for the files matching the specified pattern, here via prefix: ***pt_X??Y??_*** and suffix: ***_i***. E.g:

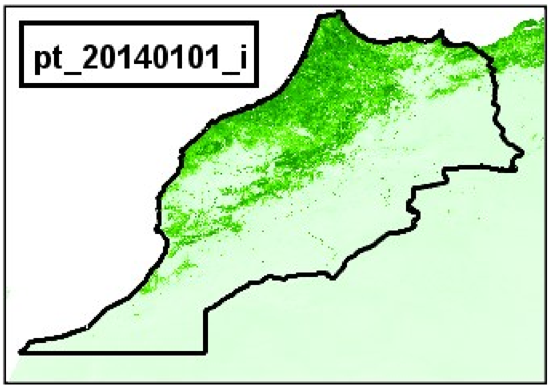
- for date 2014 01 01, the files
  - pt_X16Y03_20140101_i,
  - pt_X16Y04_20140101_i,
  - pt_X17Y03_20140101_i,
  - pt_X17Y04_20140101_i will be found in the input path – and mosaiced;
- for date 2014 01 11, the next dekad, the files
  - pt_X16Y03_20140111_i,
  - pt_X16Y04_20140111_i,
  - pt_X17Y03_20140111_i,
  - pt_X17Y04_20140111_i will be found in the input path – and mosaiced;

...



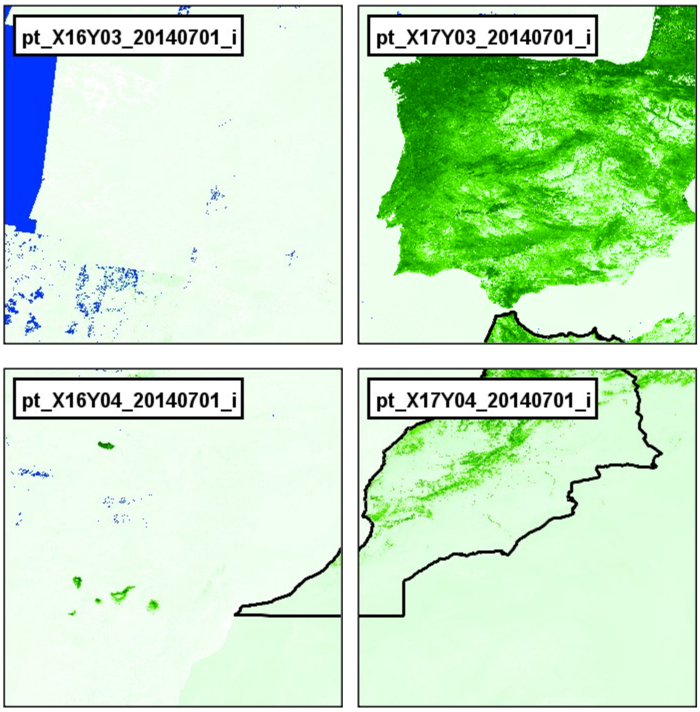
Tiles X16Y03, X17Y03, X16Y04 and X17Y04

Date 2014 01 01



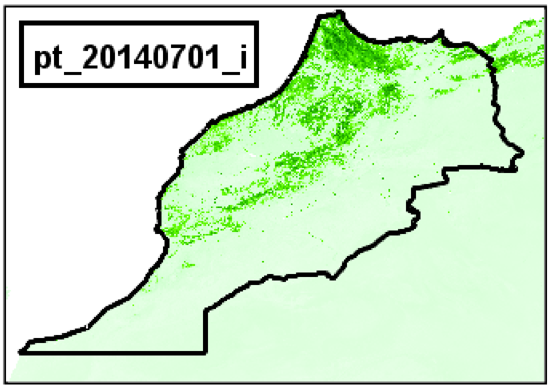
Mosaic X??Y?? for Date 2014 01 01

…



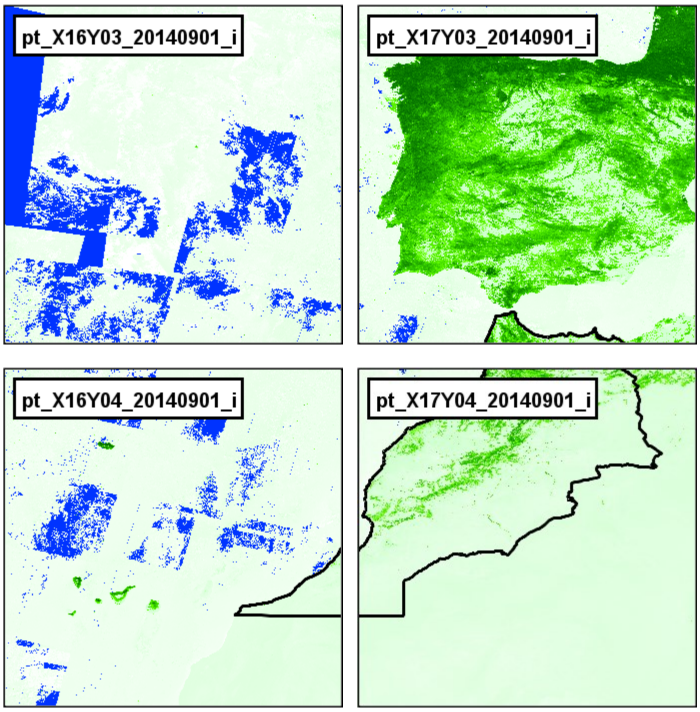
Tiles X16Y03, X17Y03, X16Y04 and X17Y04

Date 2014 07 01



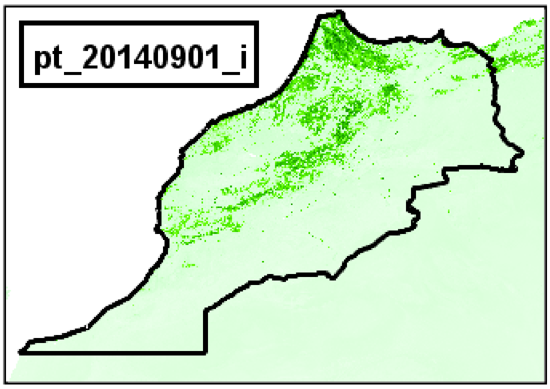
Mosaic X??Y?? for Date 2014 07 01

…



Tiles X16Y03, X17Y03, X16Y04 and X17Y04

Date 2014 09 01



Mosaic X??Y?? for Date 2014 09 01

…

3.6. Reproject IMG

Goal

Project/re-project ENVI IMGs.

Besides the actual image (re-)projection, the output framing and resolution can be specified. Also the Adapt HDR tool is integrated to modify the resulting output IMG HDR if required.

Parameters

The tool is based on the gdalwarp utility from the GDAL utilities (Geospatial Data Abstraction Library from the Open Source Geospatial Foundation).

The gdalwarp syntaxes used are as follows:

```
gdalwarp [-s_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
         [-t_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
         -dstnodata flag
         -of ENVI
         -overwrite
         input_img
         output_img

gdalwarp [-s_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
         [-t_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
         -te xmin ymin xmax ymax
         [-tr xres yres]
         -dstnodata flag
         -of ENVI
         -overwrite
         input_img
         output_img

gdalwarp [-s_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
         [-t_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
         -te xmin ymin xmax ymax
         [-ts cols rows]
         -dstnodata flag
         -of ENVI
         -overwrite
         input_img
         output_img
```

These parameters are captured in the UI as follows:

Projection panel:

- the input IMG coordinate system ("Spatial Reference Set" – SRS) can be specified via:
  - the metadata (in the HDR file) of the input IMG itself (if present);
  - an EPSG code (for instance EPSG:27700 is the British National Grid);
  - a file containing the SRS in OpenGis Well Known Text format (OGC-WKT);
  - a file containing the SRS in ESRI Well Know Text format (ESRI-WKT).

  these last two are ASCII files containing an SRS description in OGC or ESRI Well Know Text format. OGC files will often use extension *.PRF while ESRI files use *.PRJ. The tool allows *.PRF, *.PRJ and *.WKT extensions.

- the SRS for the output IMG can be specified via similar options.

In case the “Input file” option is selected for the output SRS, it means there will be no actual reprojection as such, since the input and output images will have the same SRS. This configuration can be used for example to add the “projection info” and “coordinate system string” entries to the output IMG HDR.

- the value for the “background flag” of the output IMG. This value will be used to flag output IMG areas for which no corresponding area is present in the input IMG.

Framing panel:

- the output framing can be specified:
  - by default – meaning GDAL itself will choose the framing and resolution of the output IMG;
  - via X/Y coordinates (to be specified according to the **output** SRS);
  - via an existing HDR file (assumed to have an SRS **identical** to that of the **output** IMG).

In case output framing via X/Y coordinates is selected, following parameters apply:

- the bounding box of the output IMG is to be specified via X min, X max, Y max, Y min. These coordinates need to be specified in output coordinates (according to the selected output SRS);
- the resolution of the output IMG. This can be specified:
  - by default - meaning GDAL itself will choose the resolution of the output IMG;
  - by specifying the X and Y resolution explicitly. These resolutions must be specified in output units (according to the selected output SRS);
  - by specifying the number of columns (X-direction) and rows (Y-direction) in the output IMG.

In case output framing via an existing HDR is selected, the bounding box (X min, X max, Y max, Y min) and resolution (X res, Y res) values are calculated from the values found in the “map info” entry of this HDR, and passed to gdalwarp as in previous case.

Remark: in some cases (especially with projection systems which are not globally valid) omitting an explicit output framing specification can yield unpredictable results.

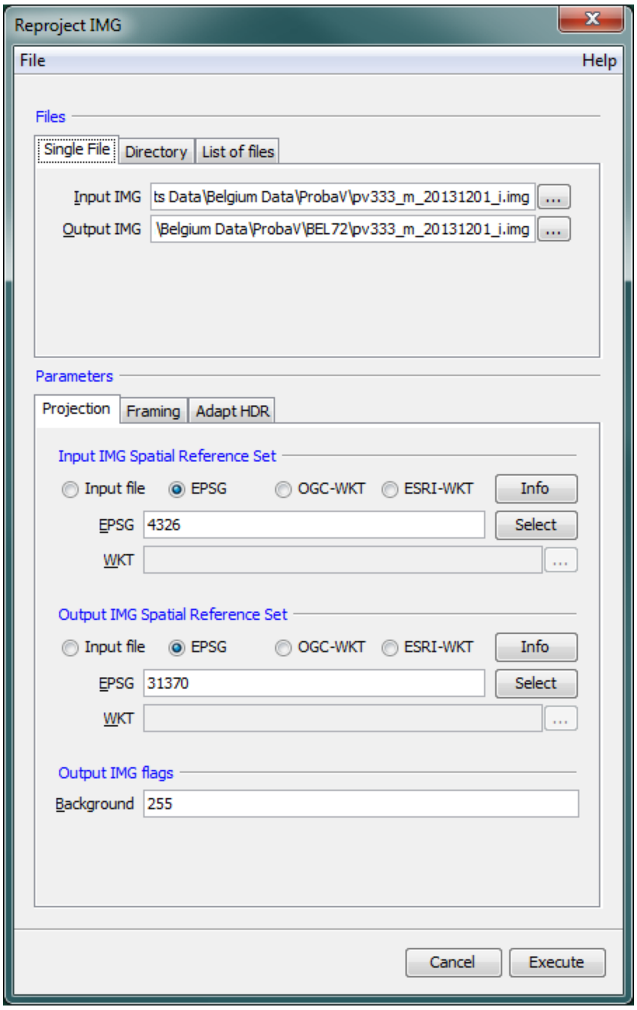
Adapt HDR:

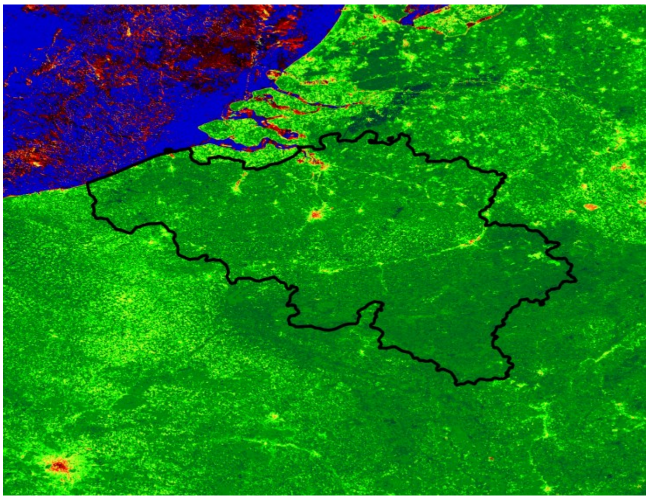
In the output HDR, all entries related to spatial information (the samples, lines, map info, projection info, and coordinate system string entries) originate from gdalwarp.

The background flag, as specified on the Projection panel, will be added to the flags entry of the input HDR.

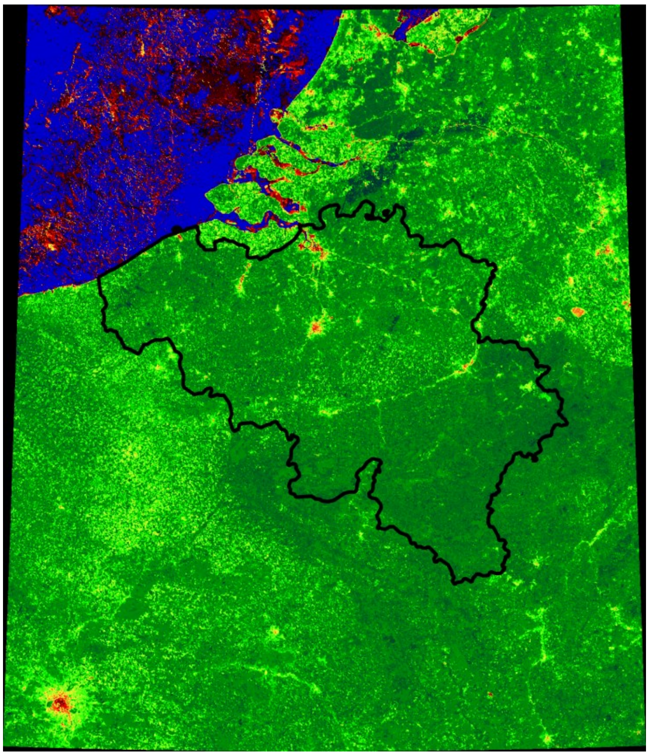
Other info, where present, is copied from the input HDR, and can be overridden via the Adapt HDR panel (see Adapt HDR tool).

Tool





input IMG (EPSG 4326 - “WGS 84” – Proba-V NDVI)



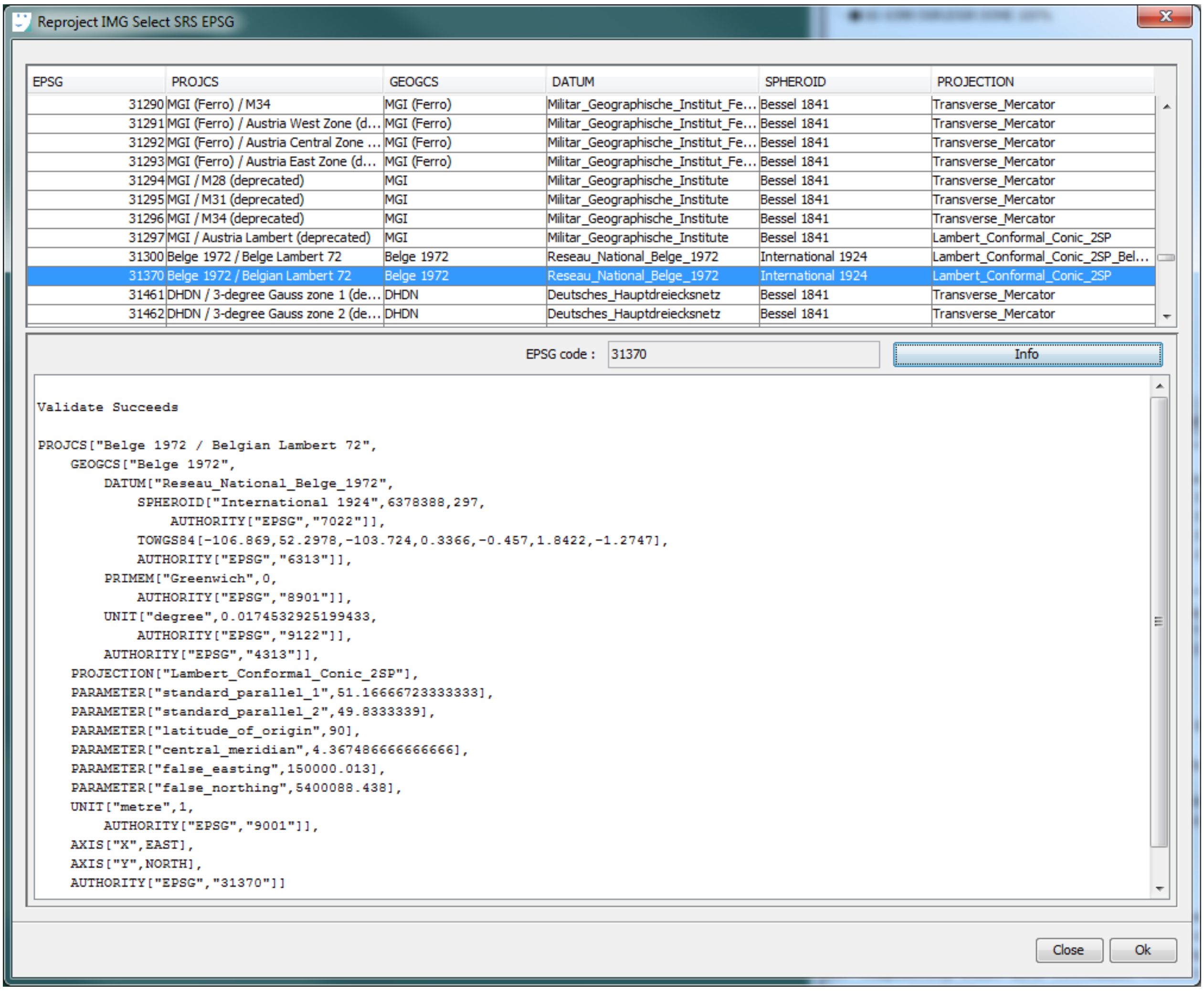
Reproject IMG Tool example output IMG (EPSG 31370 - “Bel 72” – Proba-V NDVI)

Projection panel - Select buttons:

Via the “Select” buttons the EPSG codes, know by GDAL, can be selected.

At the upper part of the selection panel the names of the projected coordinate system, the geographic coordinate system, the datum, spheroid and the projection type are listed for the available EPSG codes.

More detailed information about a selected EPSG code can be obtained via the Info button: the information is displayed at the bottom part of the panel in Well Known Text format (OGC type).



EPSG selection panel

Remarks:

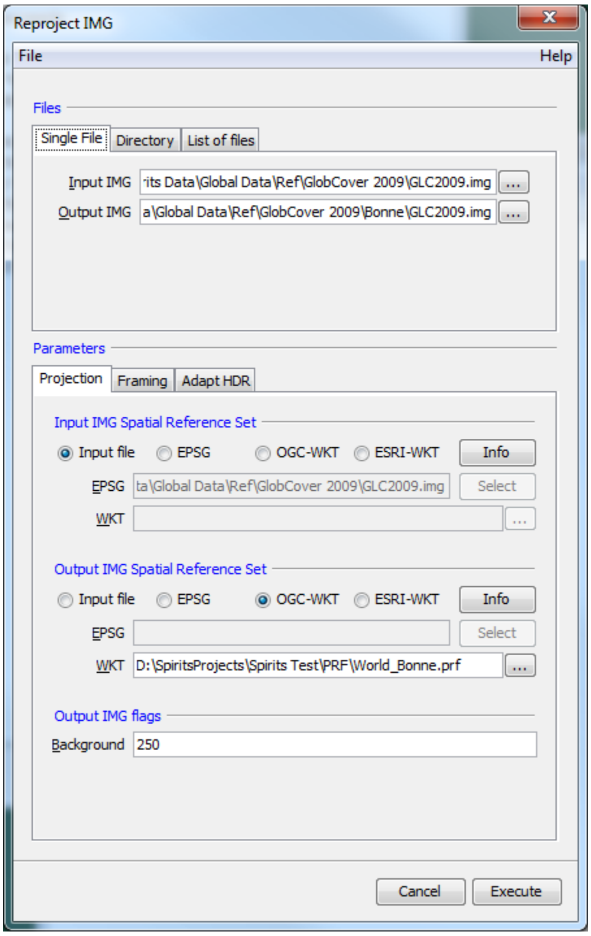
- The WKT text shown in the EPSG selection panel can be used to create a (customised) PRF file in OGC Well Known Text format, which can be used as SRS specification via the OGC-WKT selection.

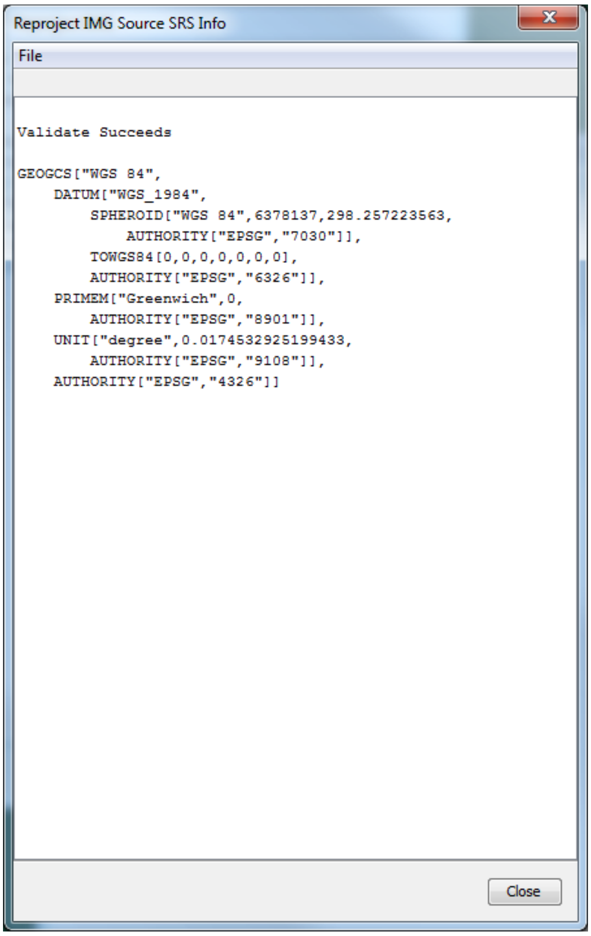
- The list of available EPSG codes originates from a Spirits system file (gdalepsgsrs.cfg) which lives in the Spirits installation directory. This file itself is has been compiled from the GDAL gcs.csv and pcs.csv data files and should not be modified. In case it is removed by accident, an attempt will be made to regenerate it automatically, but this process can take some time.

Projection panel - Info buttons:

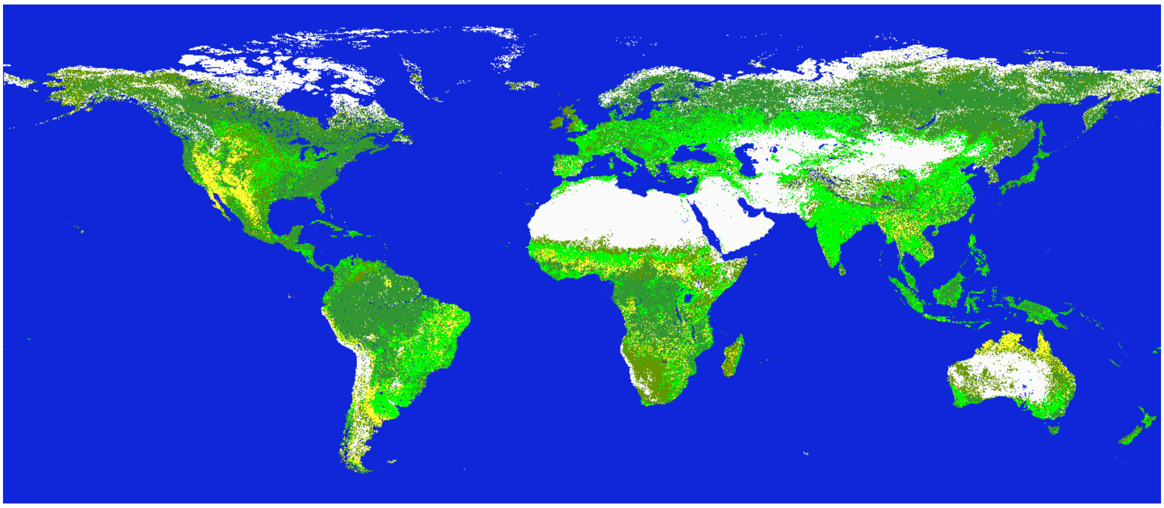
Via the “Info” buttons, the selected input and output SRS can be validated and examined. Subpanels will show the validation result and detailed information in Well Known Text format (OGC type).

Once these panels are activated, they can remain on the screen while selecting other SRS sources and input modes. In case the “Input file” option is selected for the SRS type, they relate to the file specified in the “Single File” subpanel –if any-, not to files selected in the “Directory” or “List of files” subpanels.





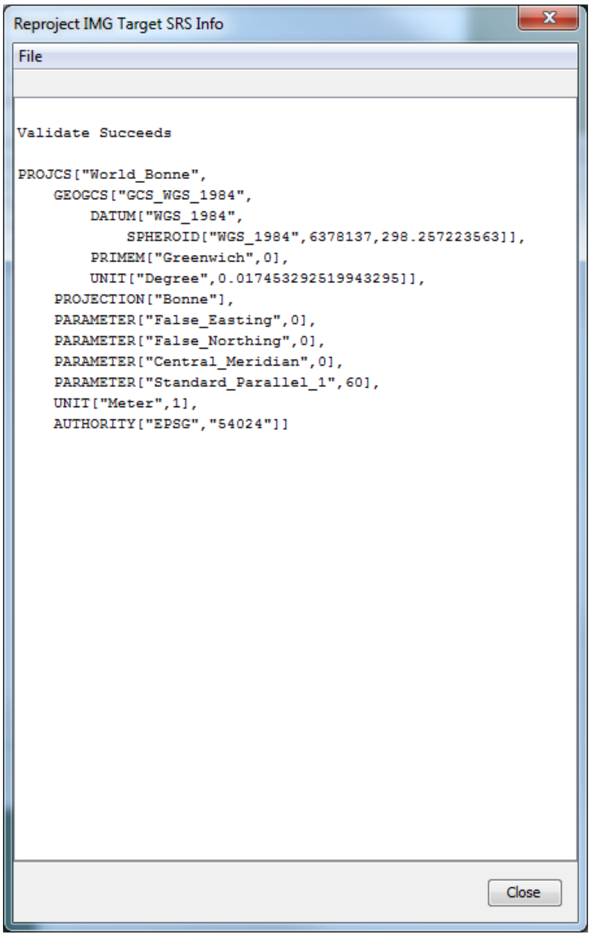
Reproject IMG Tool example Source (input) SRS Info panel

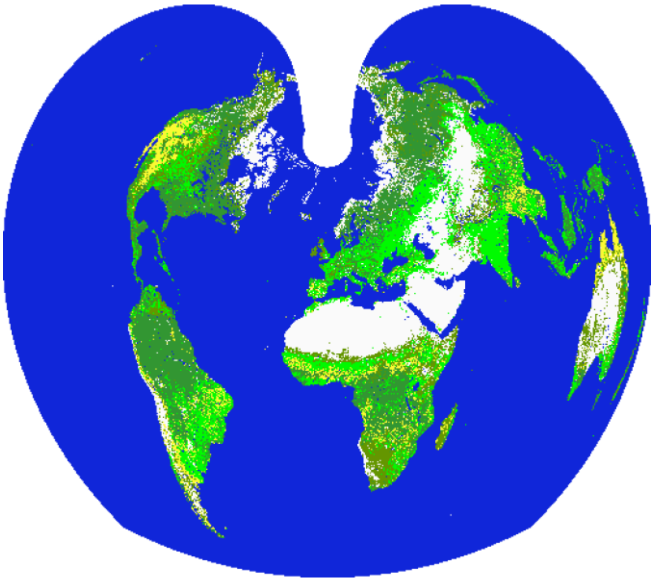


input IMG (GLC2009)

```
PROJCS["World_Bonne",
    GEOGCS["GCS_WGS_1984",
        DATUM["WGS_1984",
            SPHEROID["WGS_1984",6378137,298.257223563]],
        PRIMEM["Greenwich",0],
        UNIT["Degree",0.017453292519943295]],
    PROJECTION["Bonne"],
    PARAMETER["False_Easting",0],
    PARAMETER["False_Northing",0],
    PARAMETER["Central_Meridian",0],
    PARAMETER["Standard_Parallel_1",60],
    UNIT["Meter",1],
    AUTHORITY["EPSG","54024"]]
```

content of the selected “World_Bonne.prf” file





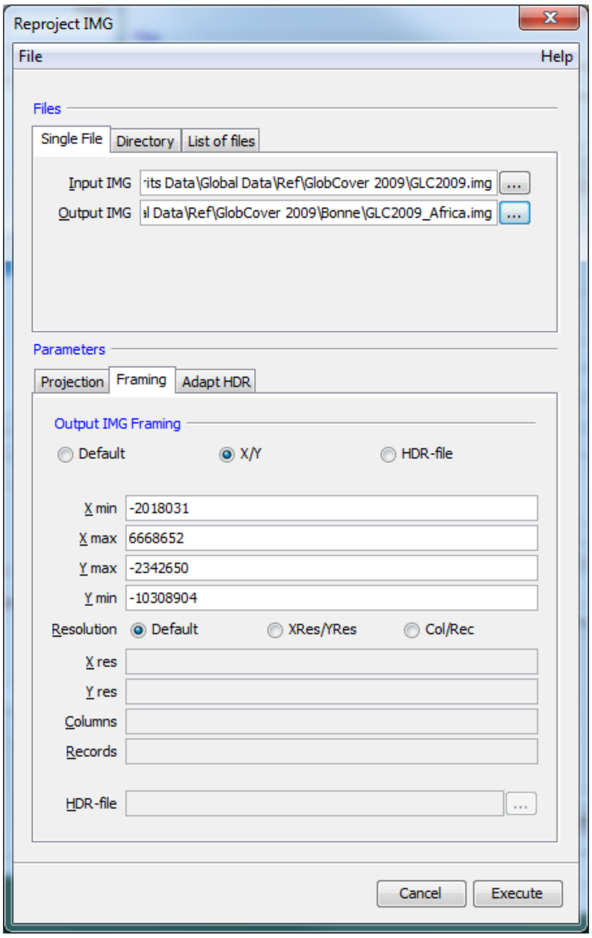
Target (output) SRD Info panel output IMG (GLC2009-Bonne) – no framing specified

```
map info = {Bonne, 1, 1, -13040395.8, 9240378.09, 24938.9969, 24938.9969}
coordinate system string =
{PROJCS["World_Bonne",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_198
4",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.01745329251994329
5]],PROJECTION["Bonne"],PARAMETER["False_Easting",0],PARAMETER["False_Northing
",0],PARAMETER["Central_Meridian",0],PARAMETER["Standard_Parallel_1",60],UNIT["M
eter",1]]}
```

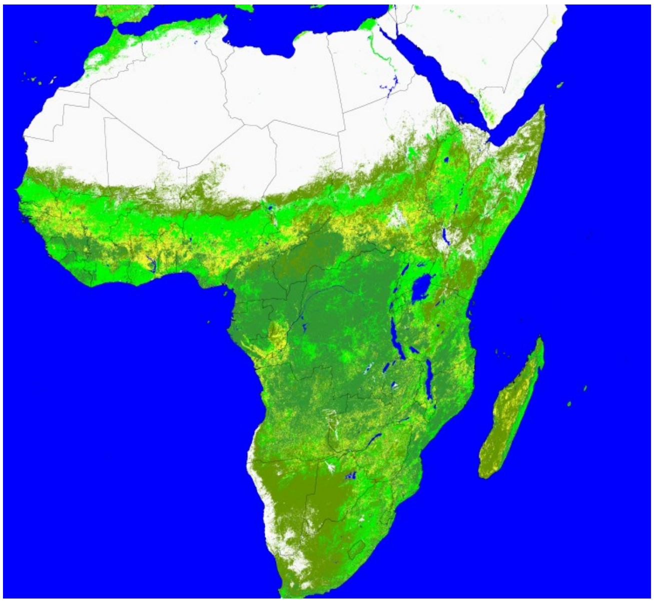
snippet from the output IMG HDR

example - framing

using the same parameters as in previous example, but with additional framing specification:



Reproject IMG Tool example - framing



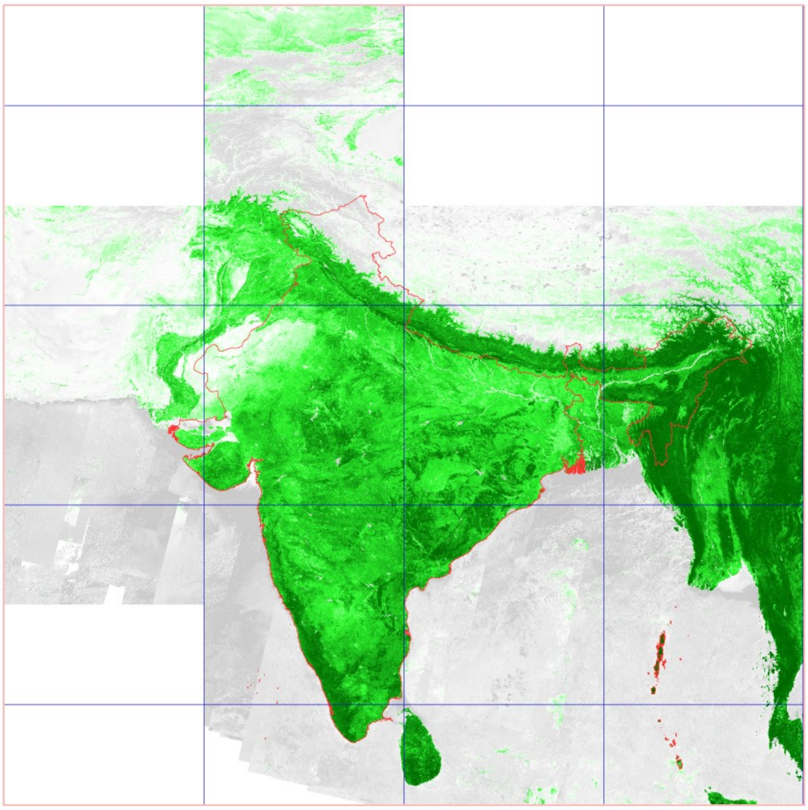
output IMG (GLC2009-Bonne – framing Africa)

Remark:

the validation and the information retrieval for an SRS is based on the gdalsrsinfo utility from the GDAL utilities. The gdalsrsinfo syntaxes used are as follows:

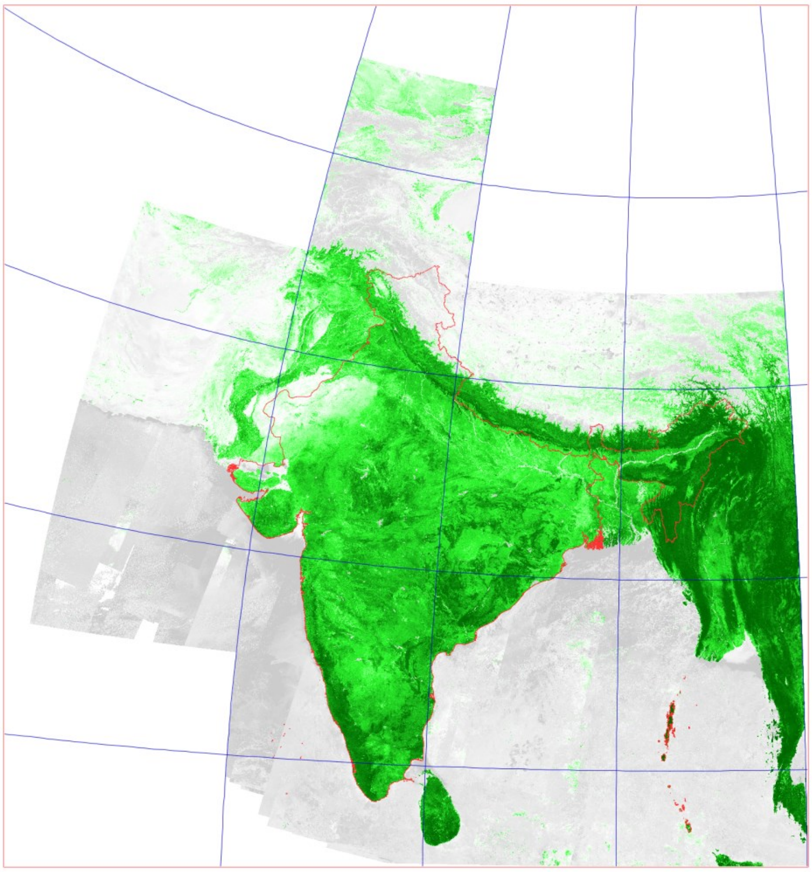
```
gdalsrsinfo -p -V -o -wkt input_img

gdalsrsinfo -p -V -o -wkt EPSG:epsg

gdalsrsinfo -p -V -o -wkt ogc_wkt_file

gdalsrsinfo -p -V -o -wkt ESRI::esri_wkt_file
```

example: Proba V 333M tiles composited and projected



probav_s30_ndvi_20131201_india

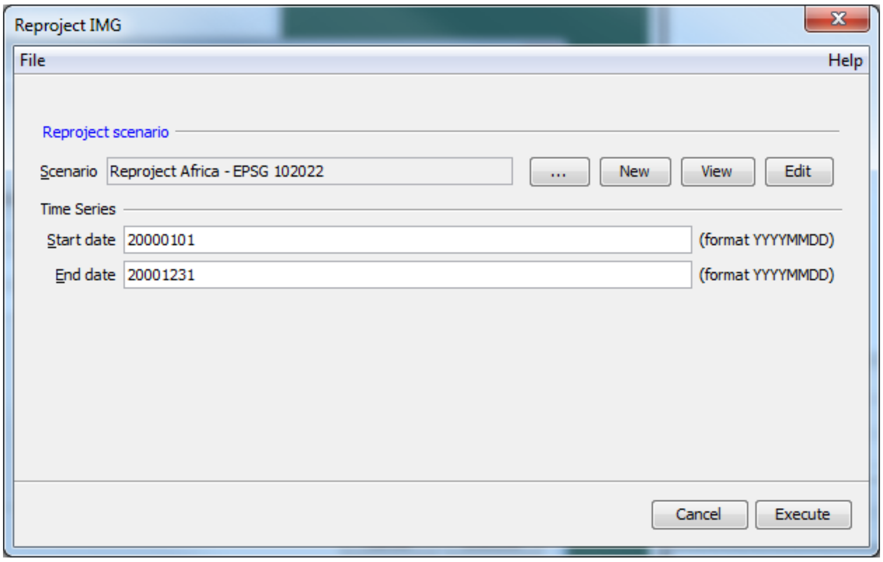
composite:
temporal S10 to S30
spatial to India ROI



probav_s30_ndvi_20131201_india_UTM46N

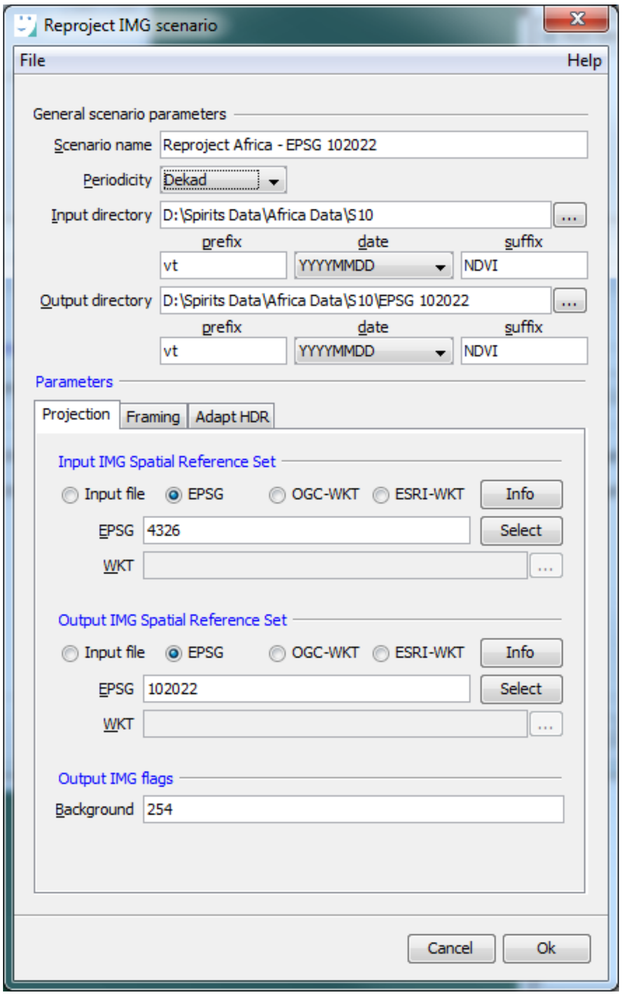
projection to UTM 46 North
(EPSG 32646)

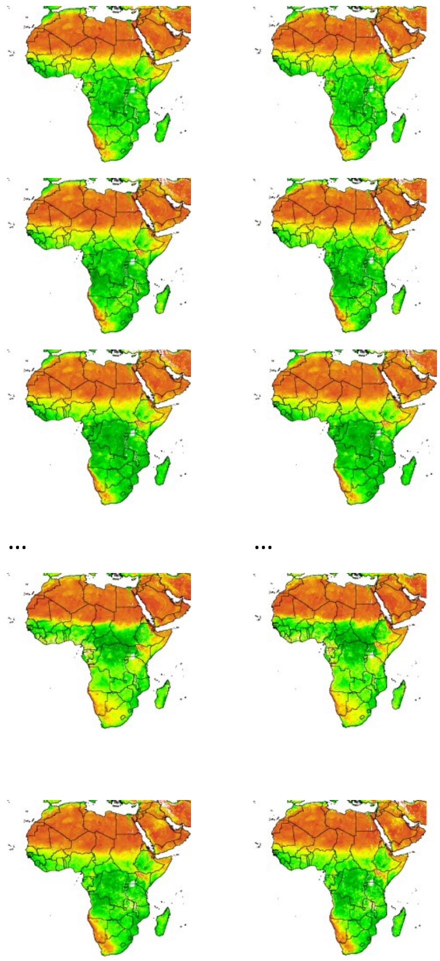
Time Series



Reproject IMG Time Series example

Scenario





EPSG 4326 EPSG 102022

Reproject IMG Scenario example SPOT NDVI S10 year 2000

3.7. Scaling and Reclassification

3.7.1. General

Goal

Apply spectral conversions: pixel based modification of the values in an IMG.

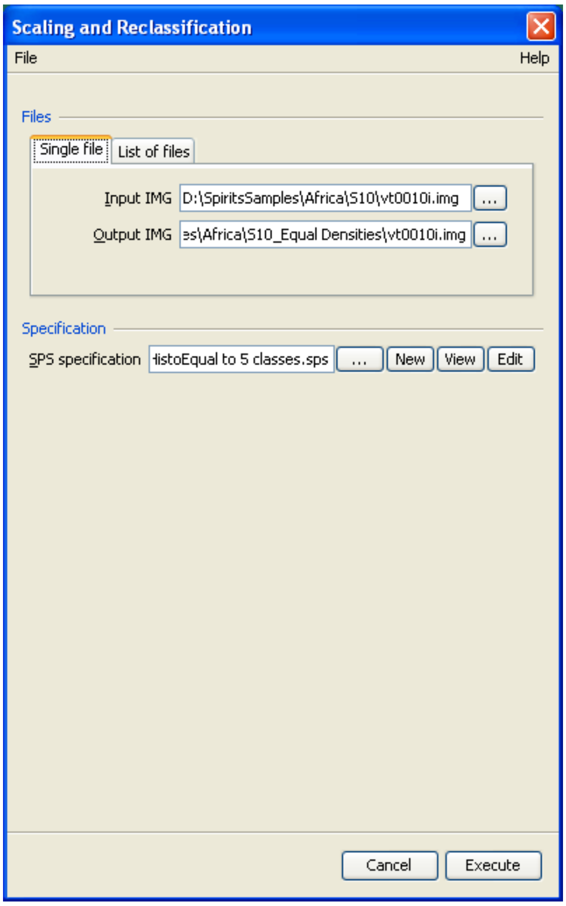
The following spectral conversions can be applied simultaneously:

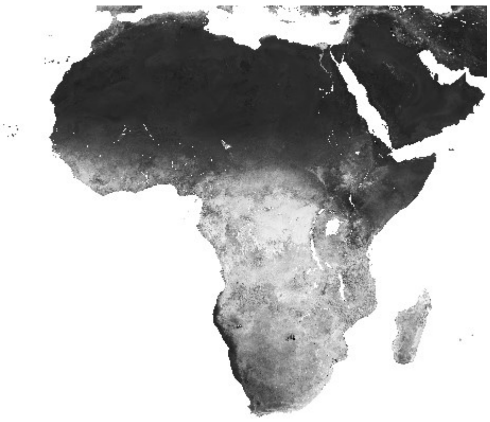
- Modification of data type;
- Straight copy of IN- to OUT-values;
- Reset ranges of IN-values to constant OUT-value (e.g. flags);
- Reclassification;
- Level slicing;
- Linear Min/Max scaling (with or without saturation);
- Histogram equalisation.

Parameters

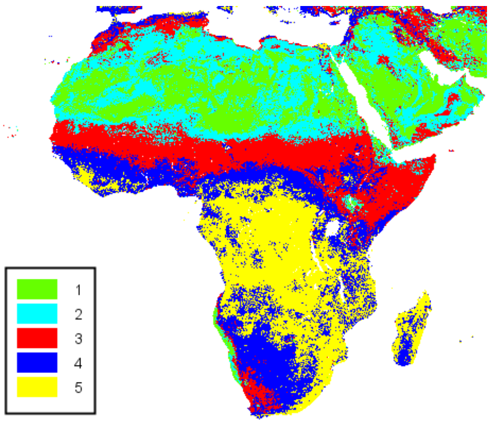
The Scaling and Reclassification Tool only needs one single parameter: the name of an SPS file which contains all the information for the scaling and reclassification process in a specific format. It is described in the following paragraph.

Tool





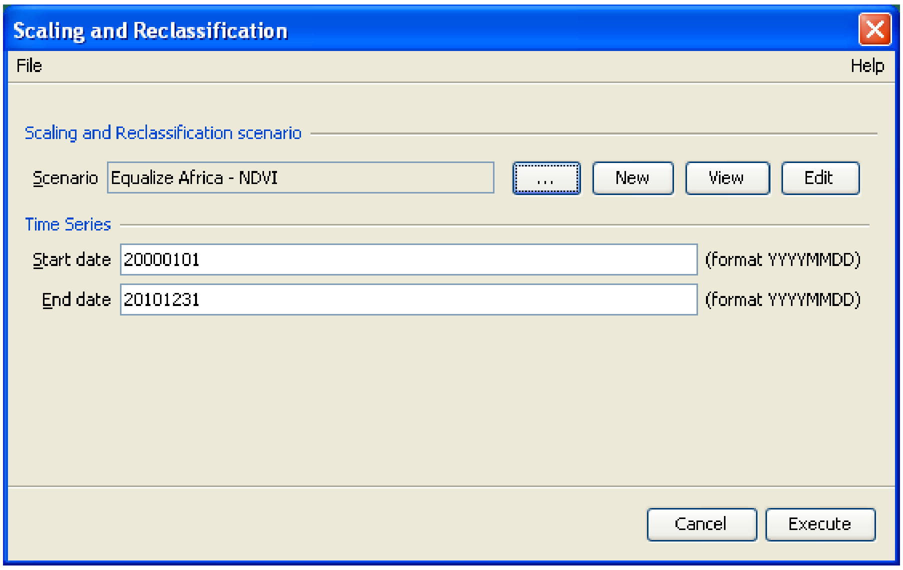
input IMG



Scaling and Reclassification Tool example

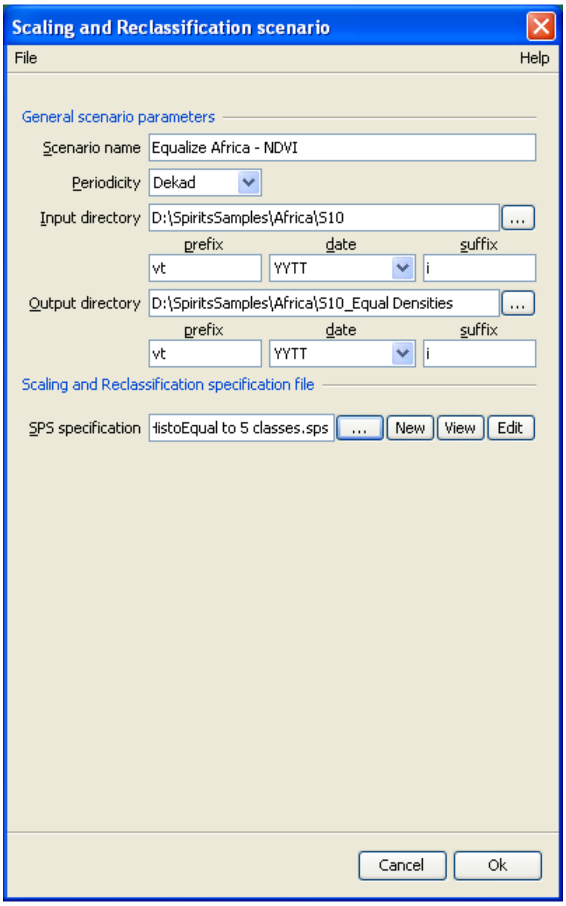
output IMG

Time Series



Scaling and Reclassification Time Series example

Scenario



Scaling and Reclassification Scenario example

3.7.2. SPS File: Scaling and Reclassification specification

Description

An SPS file, Scaling and Reclassification specification file, contains all specifications for the Scaling and Reclassification tool and time series. These files can be created and edited with the SPS editor.

Parameters

- the data type of the input-IMG: Byte(8 bit, unsigned), Integer (16 bit, signed), Long (32 bit, signed) or Float (32 bit);
- the data type of the output-IMG: Byte(8 bit, unsigned), Integer (16 bit, signed), Long (32 bit, signed) or Float (32 bit);
- a histogram file (*.HIS - optional);
- a HDR file (optional) containing the spectral information (values, flags, classes) of the HDR of the output IMG.
- a description (optional) to be used as value for the description keyword in the HDR file of the generated IMG.
- The slices must be listed in a consecutive way, and in increasing order of values. The ranges of subsequent slices cannot overlap. The total range of all slices combined will cover the entire potential range of the data type of the input IMG. Slices have the following parameters:
  - the minimum or threshold value of the slice. Each slice except the first one requires this value which specifies the lower limit value of the slice. For the first slice the lower limit will always be the potential minimum of the data type of the input IMG. The range of a slice will start from and including this minimum threshold, and end, excluding at the threshold value of the next slice. The last slice will always end at the potential maximum of the data type of the input IMG;
  - the operation to be performed on the slice, and the additional parameters specific to the operation.

Five different scaling operations can be selected:

- straight copy of the values of the input IMG. This operation does not require further information;
- replacement of the values of the input IMG by a constant value. One more parameter is needed: the value of this constant (V1);
- linear scaling of the values in the slice range to an output range. The limits of this output range (two values V1, V2) must be specified;
- histogram equalisation of the values in the slices range towards an output range. The output range (two values (V1, V2) must be specified;
- linear scaling with saturation at both edges. The output range (two values V1, V2) must be specified, as well as the lower/upper histogram percentiles where the linear scaling should start (two values, P1, P2, expressed as percentages). From the histogram, the input values (I1, I2) corresponding with these percentiles (P1, P2) are derived. The slice is then split in 3 sub-slices:
  - the lower part: input values ≤ I1 are copied to V1 (same as 'straight copy');
  - the middle part: input values between I1 and I2 are scaled to V1 -V2 (same as 'linear scaling');
  - the upper part: input values ≥ I2 are copied to V2 (same as 'straight copy').

Remarks:

- As an exception to the rule, this program only deals with "raw digital values" and not with rescaled physical values.
- Obviously, the specified output values (V1, V2) should fit into the potential range of the chosen output data type. For instance, if an image has to be scaled to the BYTE data type, V1 and V2 may not exceed the range 0-255. Otherwise, the program issues an error message.

The HDR of the output IMG inherits most characteristics of the input HDR:

- spatial features: samples, lines, map info
- temporal features: days, date
- other elements: sensor, description

However, the spectral features must be adapted. This is achieved in three levels:

- By default, the spectral parameters are reset to the most primitive values:
  - the data type is set as specified in the SPS file for the output IMG
  - all class info is removed
  - all flags are removed the values entry is reset to {"?", "?", Vlo, Vhi, Vmin, Vmax, 0, 1} with Vlo/Vhi the extreme values of the potential range of the data type of the output IMG (e.g. Vlo=0 and Vhi=255 for BYTE), and Vmin/Vmax as tracked during the creation of the output IMG.
- When an external HDR is specified in the SPS file, then its description and spectral items (values, flags, class info) are copied. Vmin/Vmax are tracked within the limits of Vlo/Vhi.
- Finally, if the SPS includes a HDR description, this overrides the previous description values.

Additionally the Adapt HDR tool can be used afterwards to adjust the output HDR.

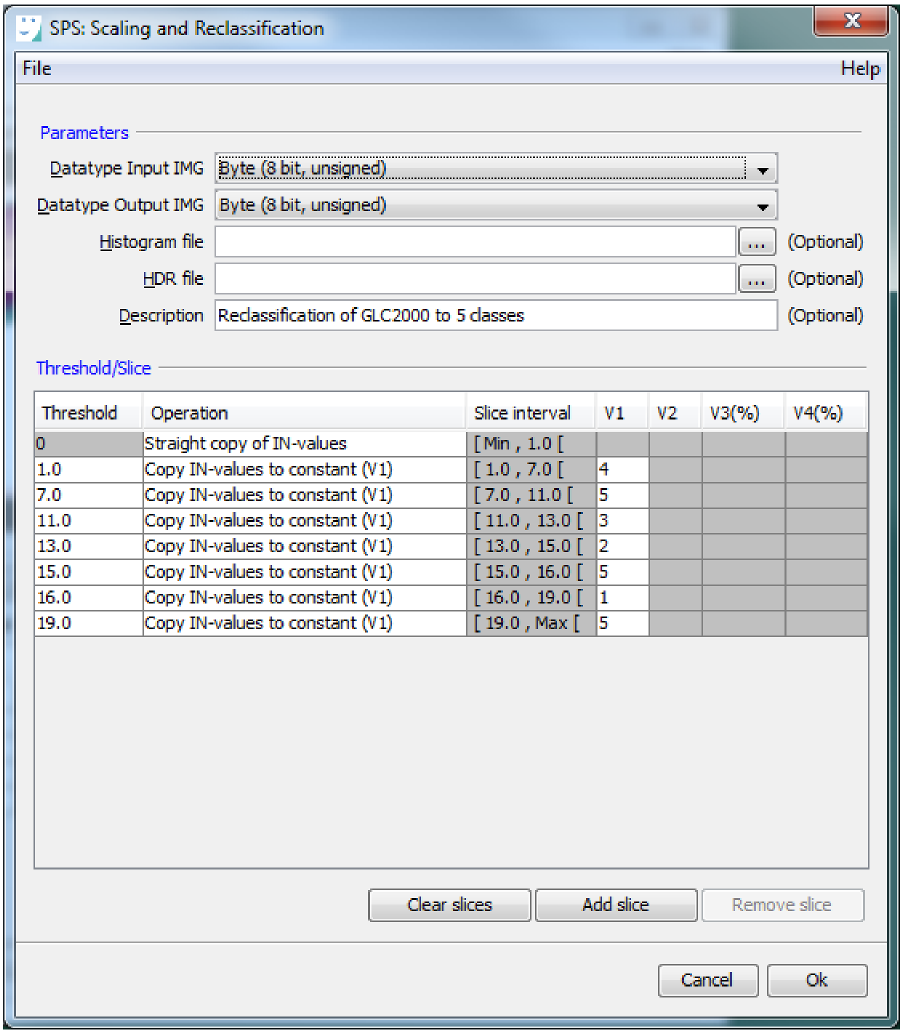
For the histogram equalisation and linear scaling with saturation the histogram of the input IMG must be computed over the input-range of the concerned slice. Therefore the original digital values must be 'binned' or grouped into a number of bins. The maximum number of bins is 1000. The actual number of used bins (Nbin) is determined as follows:

- When the data type of the input-IMG is Float: Nbin = 1000;
- For the other data types a bin width of 1 is assumed and Nbin is first assessed as: Nbin = I2 - I1 + 1 (I1 and I2 are the limits of the inputs slice). When Nbin > 1000, Nbin is reset to its maximum (1000) and the bin width is enlarged correspondingly.

The histogram can also be saved as a ASCII text file with extension *.his. To this goal the file name must be specified in the SPS file.

Any positive number of slices can be defined, each with a specific operation. However, there can only be one histogram equalisation or linear scaling operation in the list.

SPS Editor



The general parameters can be filled out directly in the SPS-editor panel.

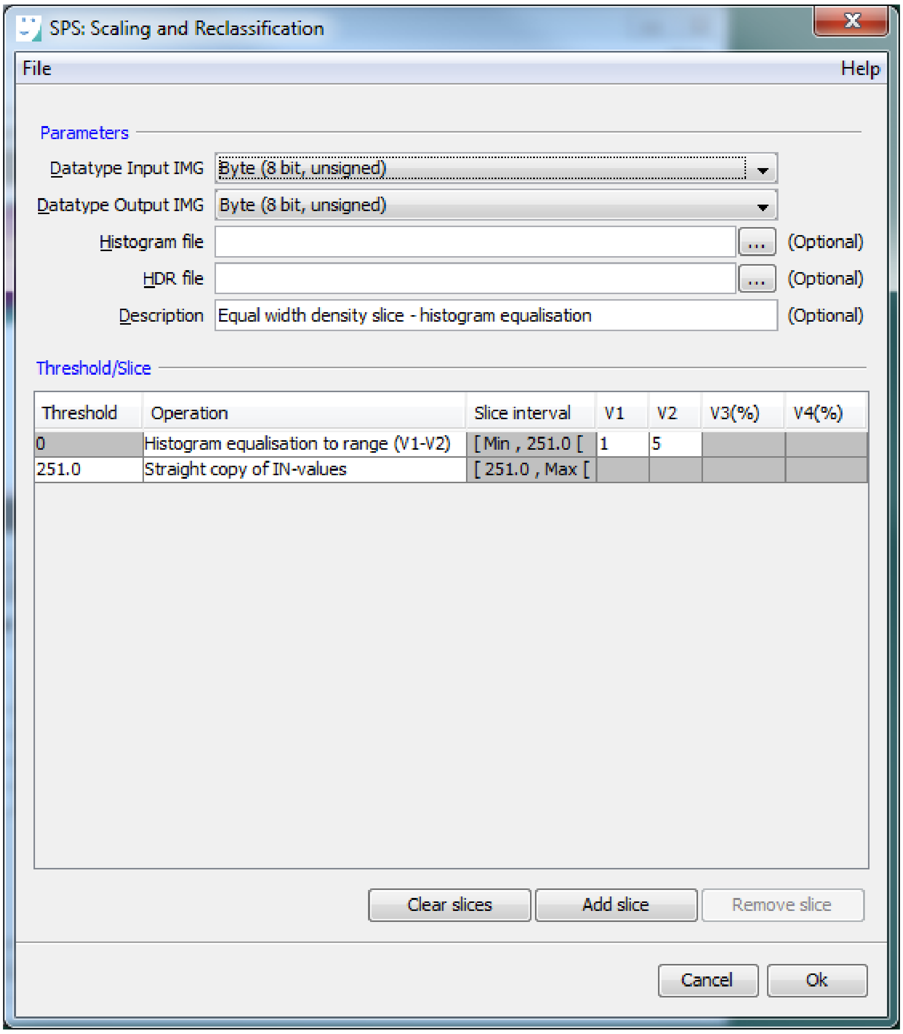
At the bottom of the panel is a list which shows all the slices and their specific parameters.

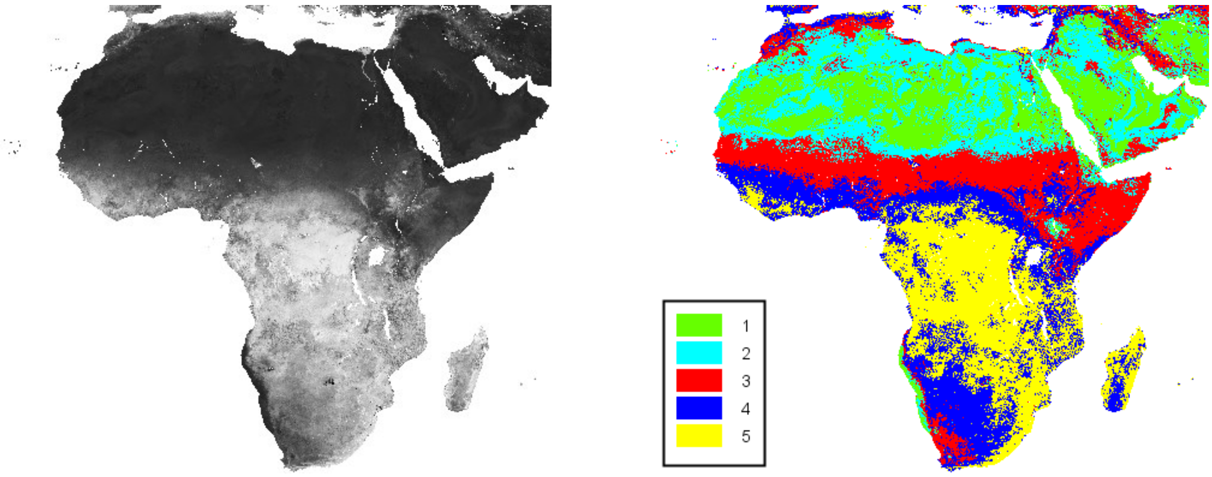
Via the action buttons slices can be added, or removed.

The slices are automatically (re-) ordered according their thresholds.

Equal width density slice example

example: equalize IMG to five classes (values 1..5) with equal pixel amounts.

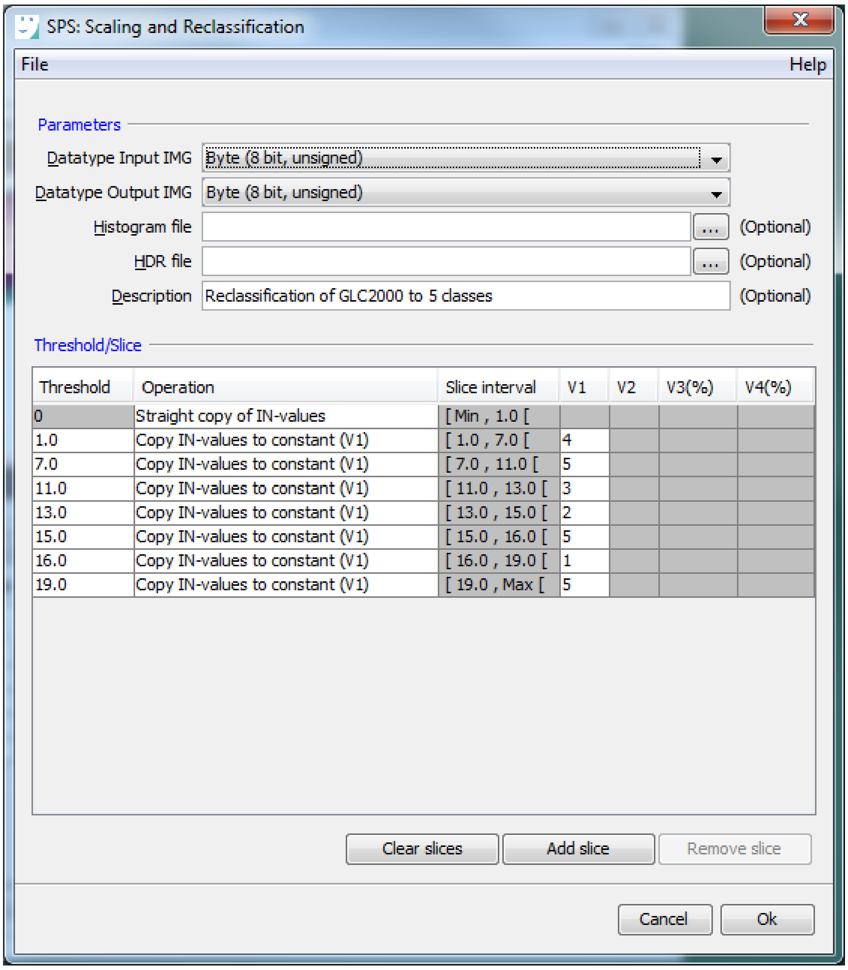




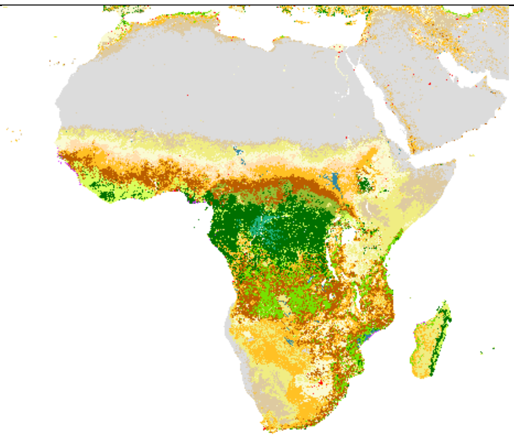
Scaling and Reclassification: input IMG output IMG

Reclassification example

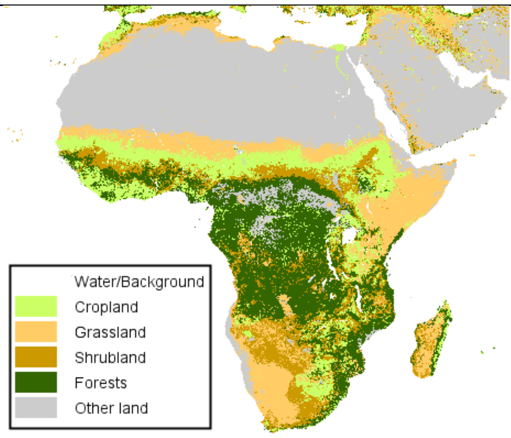
example: reclassification of GLC2000 to 5 classes.



Reclassification from:	**to:**
GLC2000 classes 16-18	class 1 : Cropland
GLC2000 classes 13-14	class 2 : Grassland
GLC2000 classes 11-12	class 3 : Shrubland
GLC2000 classes 1- 6	class 4 : Forests
GLC2000 other classes	class 5 : Other



Scaling and Reclassification: input IMG



output IMG

3.8. Index

Goal

Compute various kinds of "index" images, derived from two or three IN-IMGs. Twelve different operations are available, allowing to derive a wide range of vegetation indices (NDVI, NDWI, ...), anomalies (VCI,...), colour indices, etc.

Parameters

- the input IMG files (two or three);
- the Index to compute: twelve operations are available:
    - Y1+Y2
    - (Y1+Y2)/2
    - Y1-Y2
    - Y1/Y2
    - (Y1-Y2)/Y1
    - (Y1-Y2)/Y2
    - (Y1-Y2)/(Y1+Y2)
    - Y1+Y2+Y3
    - (Y1+Y2+Y3)/3
    - Y1/(Y1+Y2+Y3)
    - (Y1-Y2)/Y3
    - (Y1-Y3)/(Y2-Y3)

  Reminder: the Y values are the physical values from the input IMGs according to:
  Physical value = Vint + Vslo * Digital IMG value
  with Vint and Vslo the intercept and slope specified in the values entry in the IMG HDR:
  values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo }.
  Reminder: only digital IMG values in the [Vlo, Vhi] range will be considered, values outside this range are regarded as flags.

- the method to deal with negative input values: three options are available: Keep them as such, Reset them to zero or Flag them in the output IMG.
- whether to use the absolute value of the denominator (if any).

  Example: suppose (Ycurrent-Yprevious)/Yprevious is used to compute a relative difference between images of successive time periods, where Yprevious < Ycurrent < 0, the index would be negative, suggesting a declining trend while it is actually positive (Yprevious < Ycurrent). By taking the absolute value of the denominator, this bias is corrected.

- the data type of the output IMG: Byte(8 bit, unsigned), Integer (16 bit, signed), Long (32 bit, signed) or Float (32 bit);
- a multiplication factor (default 1). The (physical) index value is the result of the operation multiplied by this factor;
- the potential minimum and maximum (physical) index values.
- the minimum and maximum digital values corresponding to the potential minimum and maximum (physical) index values.

Remark: The potential minimum and maximum physical index values and the minimum and maximum digital values together define the scaling of the output IMG. The output IMG HDR will contain the values entry:

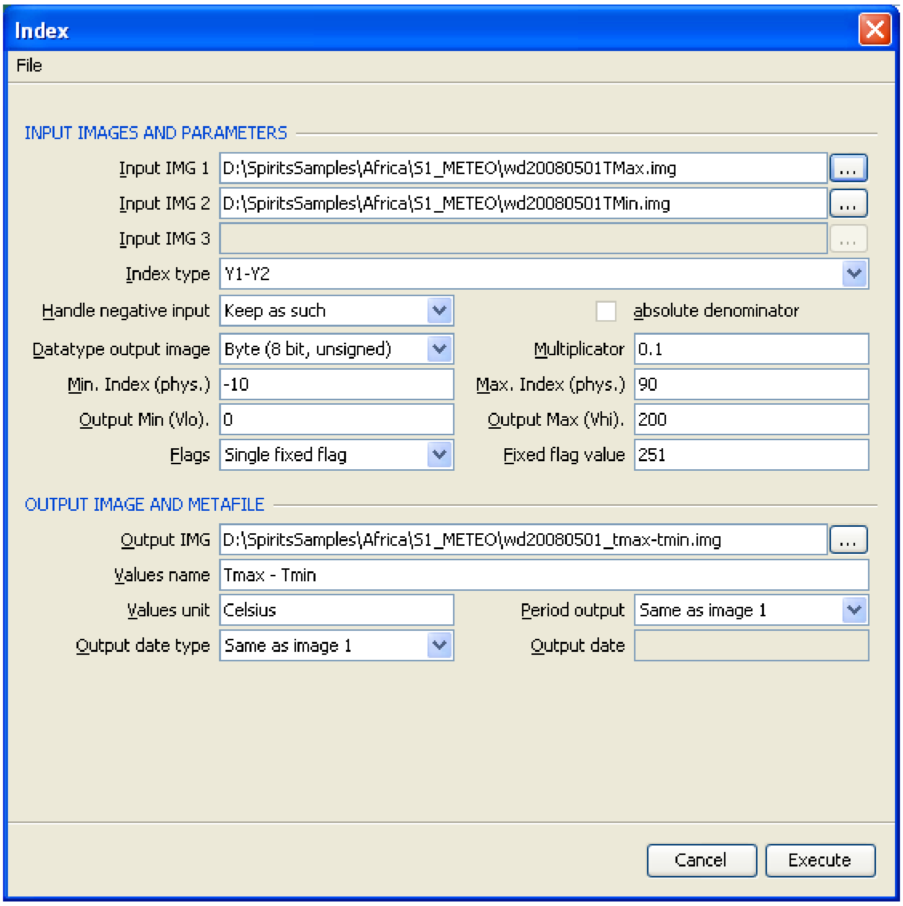
values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo } with
Vlo = the minimum digital value specified
Vhi = the maximum digital value specified
Vslo = (maximum physical index specified -minimum physical index specified) / (Vhi - Vlo)
Vint = minimum physical index specified - Vslo * Vlo

- the output IMGs flags type:
  - if input IMGs are UNI-Flagged, the output IMG can also be UNI-Flagged;
  - otherwise a single flag will be used in the output IMG. The value of this flag must be specified.

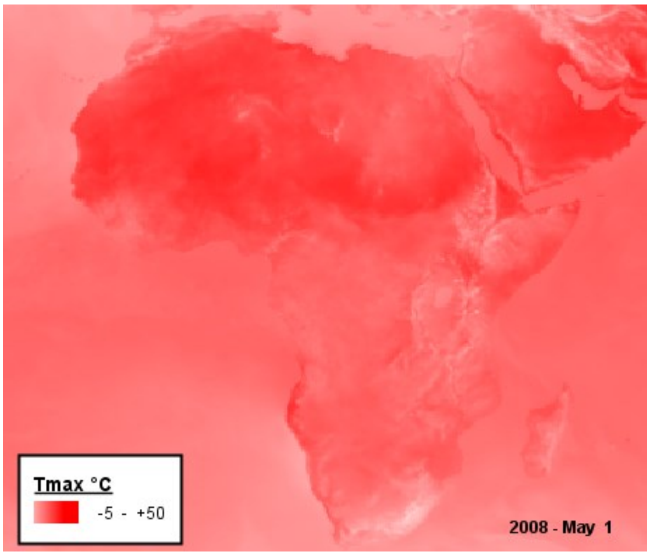
Remark: the output IMG will be flagged where one (or more) of the input IMGs is flagged, where an operation divides by zero and in case the "flag negative inputs" was selected where negative input occurs.

- the output IMG file;
- the values name to be used in the output IMG HDR values entry (optional);
- the values unit to be used in the output IMG HDR values entry (optional);
- the periodicity and date to be used used in the output IMG HDR days and date entries.

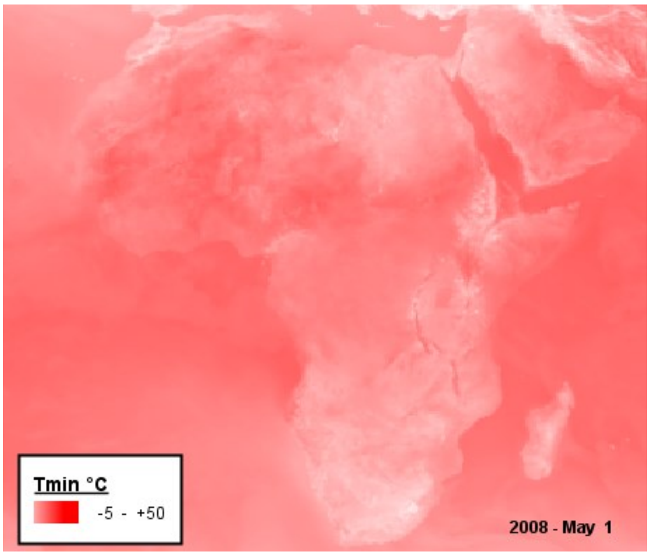
Tool



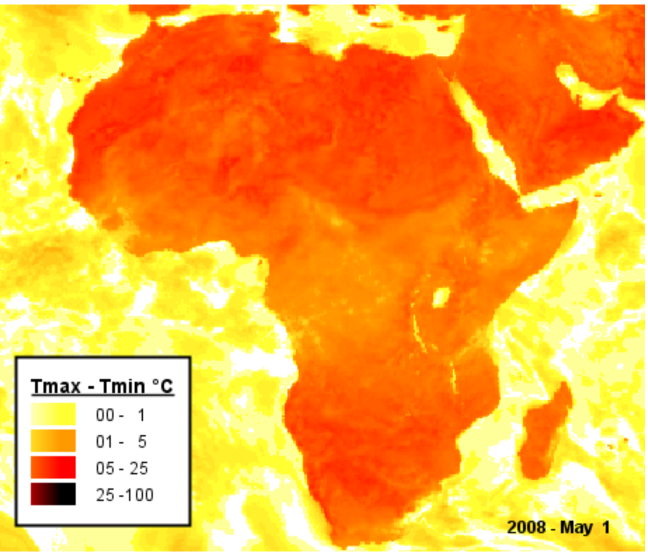
Index Tool example



Input IMG 1(Tmax)



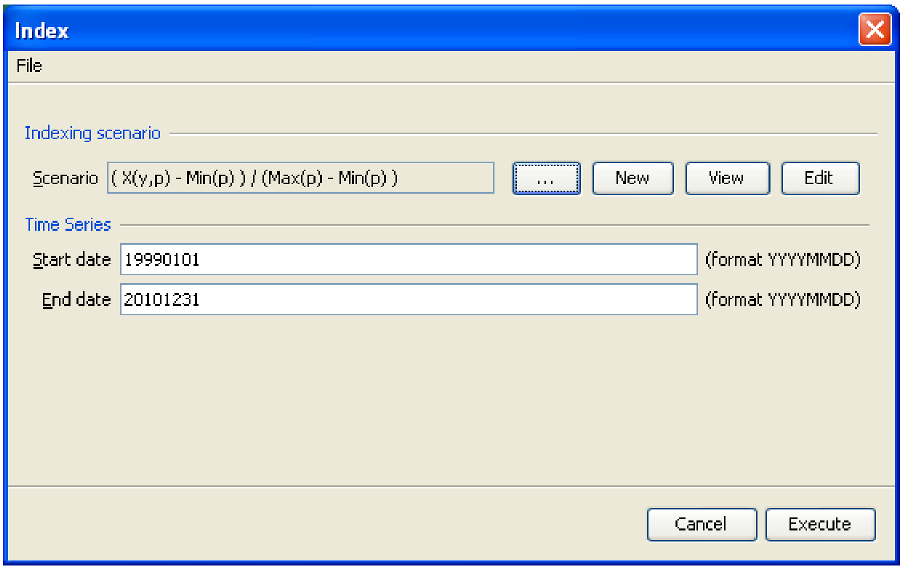
Input IMG 2 (Tmin)



Output IMG: Y1-Y2 Index (Tmax - Tmin)

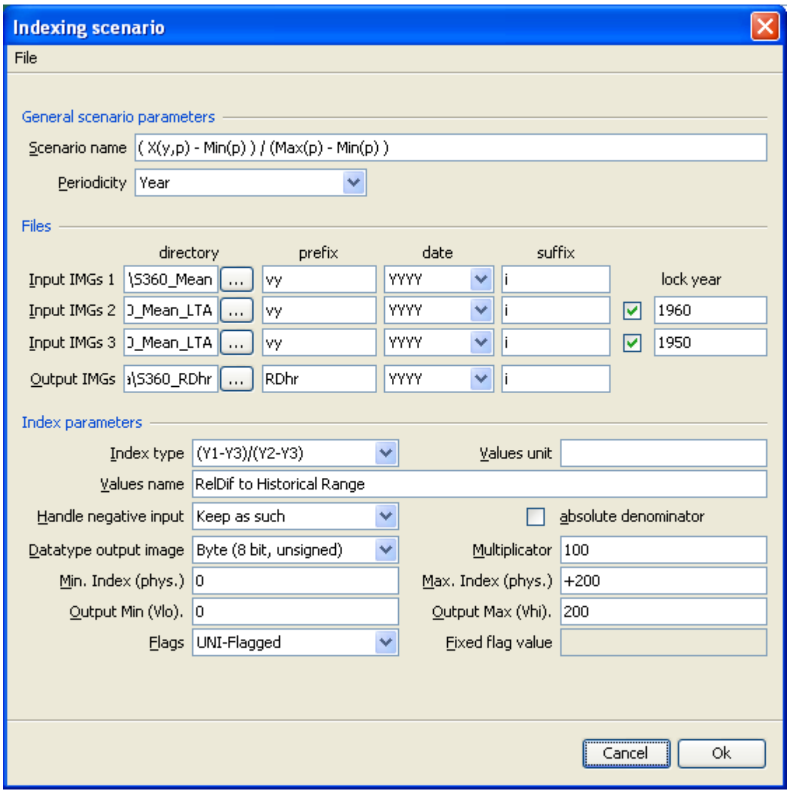
Index Tool example

Time Series



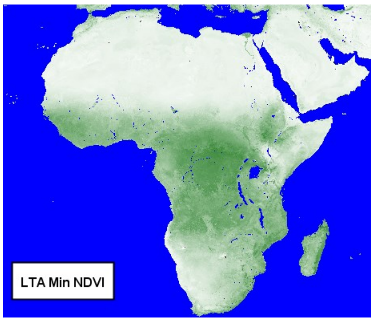
Index Time Series example

Scenario

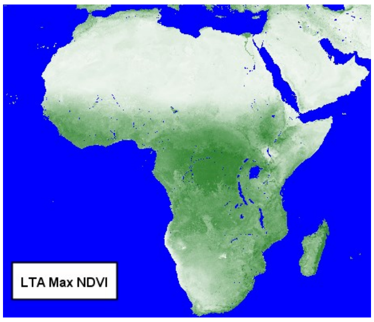


Index Scenario example

Remark: the lock year option allows specification of a constant YYYY value for the second and third IMG of the operation. This enables the use of operations against some reference year, e.g. against the historical IMGs computed by the Long Term Average tool, as in the example.

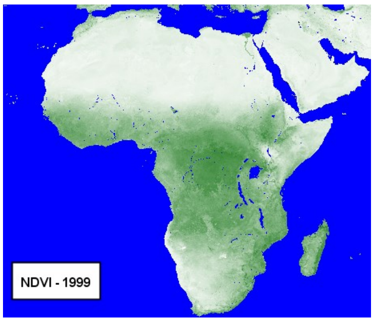


input IMG 3
historical Minimum (calculated with the Long Term Average tool)

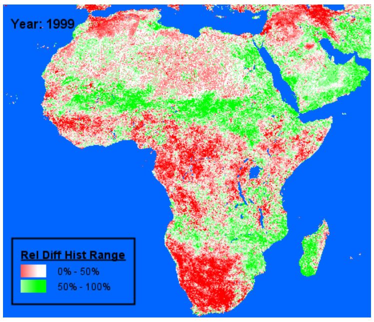


input IMG 2
historical Maximum (calculated with the Long Term Average tool)

Index (Y1 -Y3) / (Y2-Y3) :
(NDVI(y) - NDVI(min)) / (NDVI(max) - NDVI(min))

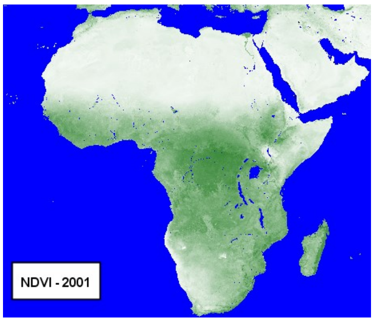


input IMG 1
(NDVI 1999)

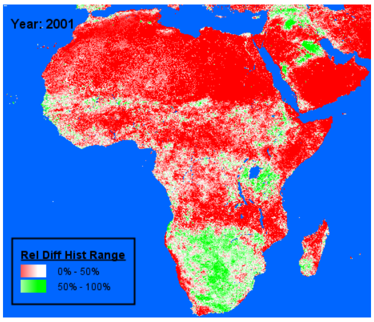


output IMG
Index 1999

...

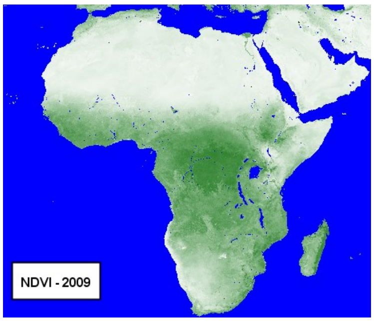


input IMG 1
(NDVI 2001)

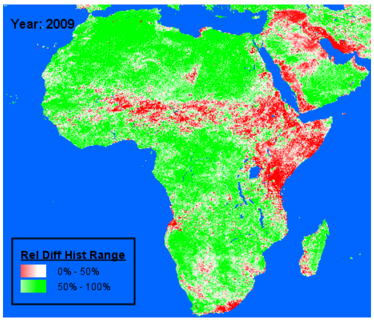


output IMG
Index 2001

...



input IMG 1
(NDVI 2009)



output IMG
Index 2009

input IMGs series

Index output IMGs

3.9. Band Calculator

Goal

Create output images by performing algebraic operations (per pixel) on a set of specified input images (bands) and constants. Besides a collection of standard algebraic operators (addition, multiplication, ...), some bit-oriented operators, boolean operators and a conditional operator are available.

Parameters

- the input IMG files with their specific parameters, whose values will be the variables in the expression;
- the constant values used in the expression;
- the algebraic expression or formula to be applied;
- the output IMG file with its specific parameters.

3.9.1. Overview

Input IMGs

A number of input IMGs (actually bands) can be specified. The pixel values of these can then be used as the values for the variables in the expression. Each input yields two variables, indicated in the expression as X(i) and Y(i), i being the index of the variable. The X(i) variables contain the digital values as they are stored in the IMG band, the Y(i) variables represent the physical values, which are obtained by (de-)scaling the digital values.

Constants

A number of constant values can be specified to be used in the expression. These will be indicated as C(i), i being the index of the constant.

Expression

The expression used by the Band Calculator consists of a sequence of operators, acting on operands.

The operands can be the input IMG values (X(i), Y(i)), the constants values (C(i)) or the intermediate result from previous operators.

The operators can be selected from a set of standard algebraic operators (addition, multiplication, ...), bit-oriented operators (bitwise AND, OR, ...), boolean operators (AND, OR, ...) and a conditional operator.

The Band Calculator uses the so called Reverse Polish notation (RPN), a mathematical notation in which an operator follows its operands, as opposite to the traditional infix notation which puts its operators in-between the operands.

traditional Infix	***1 + 2***	operator “+” between the operands
Reverse Polish notation RPN	***1 2 ADD***	operator “ADD” following its operands

RPN allows solving problems in a way that mimics how people learn to do math on paper.

1	1
2	2
+ ----	ADD
3	3
“math on paper”	typical RPN calculator

RPN removes the need for parentheses, operator-precedence and associativity (what happens when an operand is surrounded by operators of the same precedence) rules, which are needed in infix. The greatest –if not only- advantage of the infix notation, is its familiarity, however, once learned, RPN simplifies calculations, especially in case of non-trivial expressions. Typical RPN implementations will “visualize” their expression in some stack.

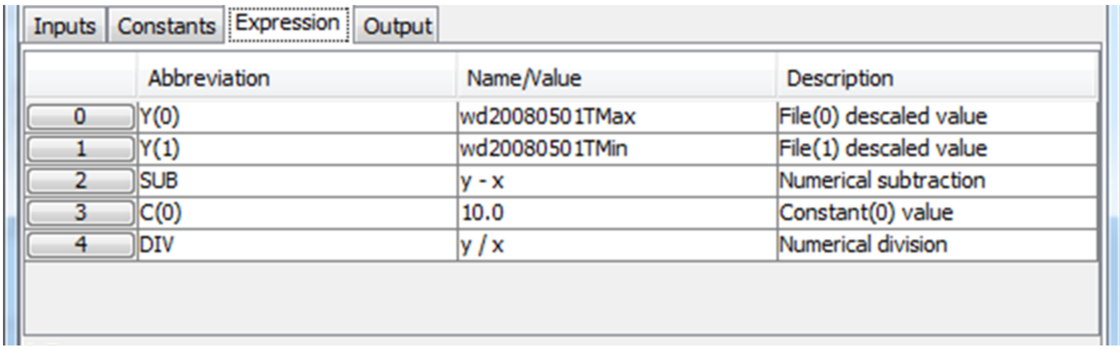
RPN operators act on ('consume') their operands, thereby producing a resulting operand. This operand can be the overall result of the expression, or act as input operand for a next operator. We distinguish unary, binary and ternary operators, which respectively consume one, two or three operands.

X	operand	result is a new operand (with value sqr(x))
SQR	unary operator	

X1	operand	result is a new operand (with value X1 + X2)
X2	operand	
ADD	binary operator	

X1	operand	result operand value (X1-X2)	result operand value (X1-X2)/(X1+X2)
X2	operand		
SUB	binary operator		
X1	operand	result operand value (X1+X2)	
X2	operand		
ADD	binary operator		
DIV	binary operator		

X1	operand	result operand R value TRUE or FALSE	result operand value X1 in case R is TRUE, X2 in case R is FALSE
X2	operand		
GT	binary operator		
X1		operand	
X2		operand	
IIF	ternary operator		



	Abbreviation	Name/Value	Description
0	Y(0)	wd20080501TMax	File(0) descaled value
1	Y(1)	wd20080501TMin	File(1) descaled value
2	SUB	y - x	Numerical subtraction
3	C(0)	10.0	Constant(0) value
4	DIV	y / x	Numerical division

example Band Calculator expression

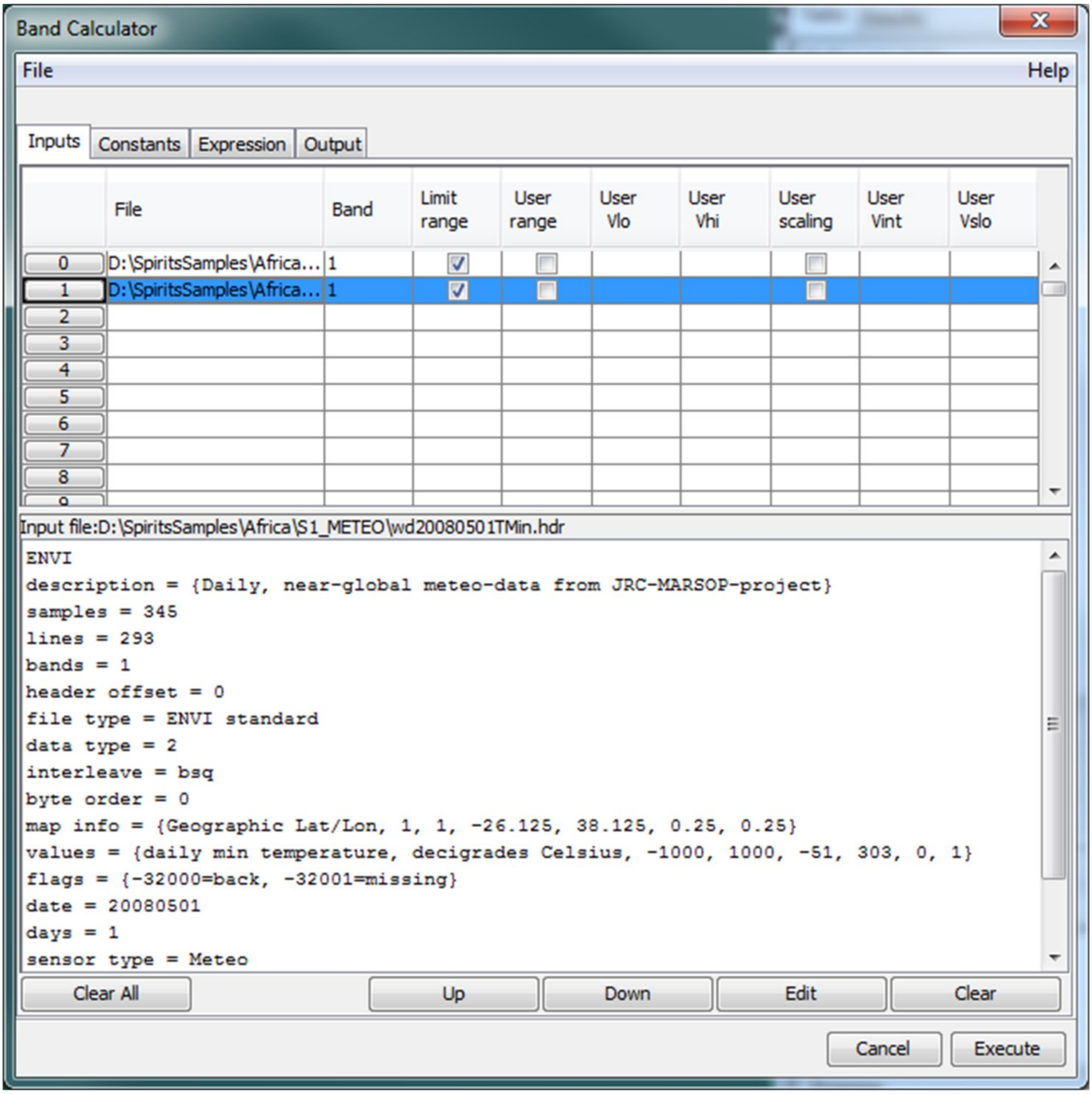
Output

The output file must be specified, together with its basic parameters (valid range, scaling,...). Furthermore two mandatory flags must be specified:

- "invalid input": will be used in case the pixel of one or more input files is out of its valid range (flagged);
- "invalid result": will be used in case the result of the calculation is considered invalid, e.g. divisions by zero, or values outside the specific range.

3.9.2. Band Calculator Tool

3.9.2.1. Input IMGs and parameters



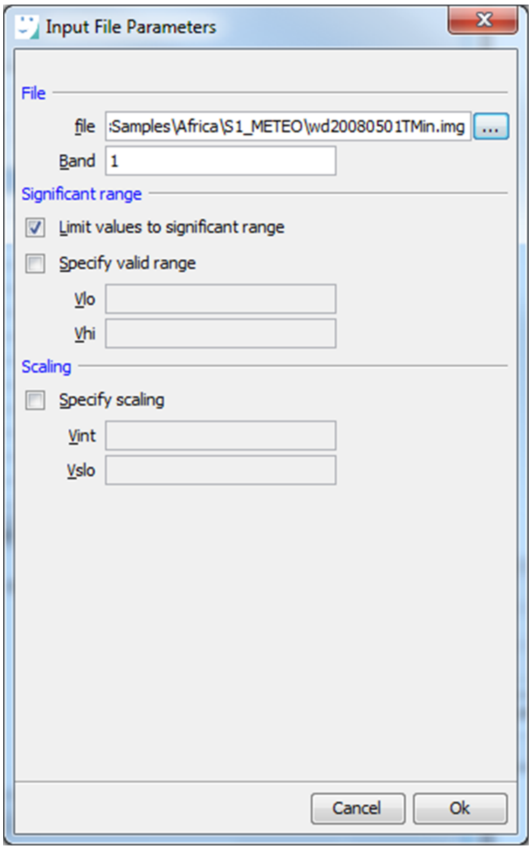
Band Calculator Tool - Input IMGs panel

The inputs panel contains a list of input IMGs and parameters.

- entries can be added or modified (double click selected entry or select entry and "Edit" button);
- the selected entry or the complete list can be cleared ("Clear" or "Clear All" buttons);
- the selected entry can be moved up and down in the list ("Up" and "Down" buttons). Its position in the list is relevant as it determines its index.

Each input yields two variables: entry with index ***i*** will be used as variables ***X(i)*** and ***Y(i)*** in the expression.

- X(i) : the digital value as stored in the IMG band;
- Y(i) : the physical value of the entity represented by the IMG band;
- their relation: ***Physical value Y(i) = Vint + Vslo * Digital value X(i)***



Band Calculator Tool - Input IMG parameters panel

Significant range

The input values can be limited to a significant range. In that case the calculation will be restricted to those pixels with (digital) values in this range. Pixels with values outside this range will be flagged in the output file. The significant range [Vlo, Vhi] would typically be available in the values entry of the HDR file (*values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo }*). They can also be specified directly by the user, thereby overriding the HDR values if present.

In case the "limit values" option is selected, Vlo and Vhi must be available, either from the HDR or from direct user specification.

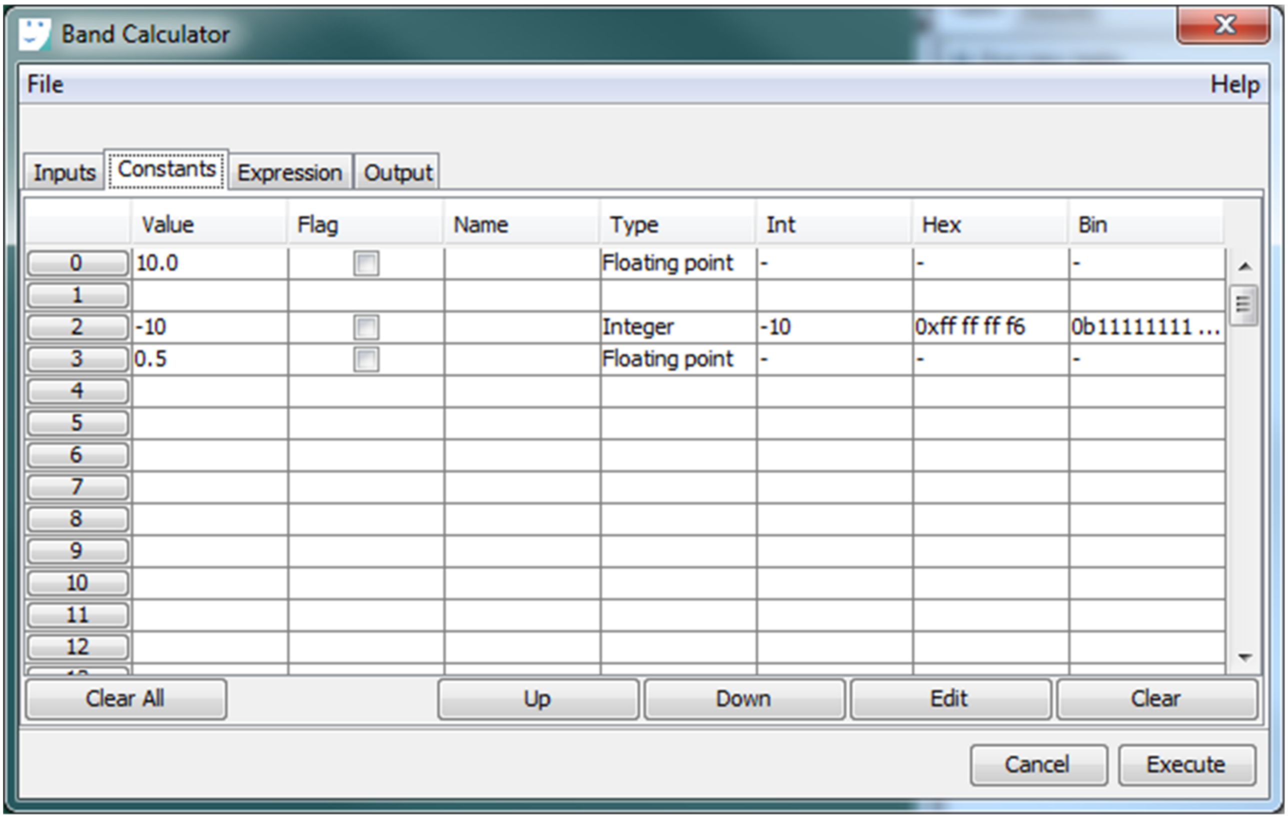
In most cases (calculations of indexes), one would choose to limit the inputs to their significant range. However, allowing values outside the valid range makes sense for example in cases where one wants to manipulate flag values directly.

Scaling

The scaling parameters Vint (intercept) and Vlso (slope), needed for Y(i), would typically be available in the values entry of the HDR file (*values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo }*). They can also be specified directly by the user, thereby overriding the HDR values if present.

In any case, Vint and Vslo must be available for each input IMG.

3.9.2.2. Constants

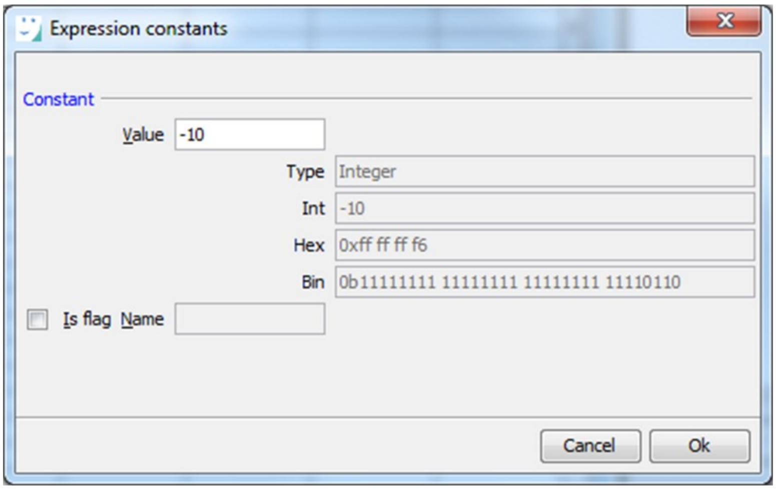


Band Calculator Tool - Constants panel

The constants panel contains a list of constant values, their parameters and properties.

- entries can be added or modified (double click selected entry or select entry and "Edit" button);
- the selected entry or the complete list can be cleared ("Clear" or "Clear All" buttons);
- the selected entry can be moved up and down in the list ("Up" and "Down" buttons). Its position in the list is relevant as it determines its index

These constants (values) can be used in the expression, entry with index ***i*** will be indicated as ***C(i)***.



Band Calculator Tool - Constant parameters panel

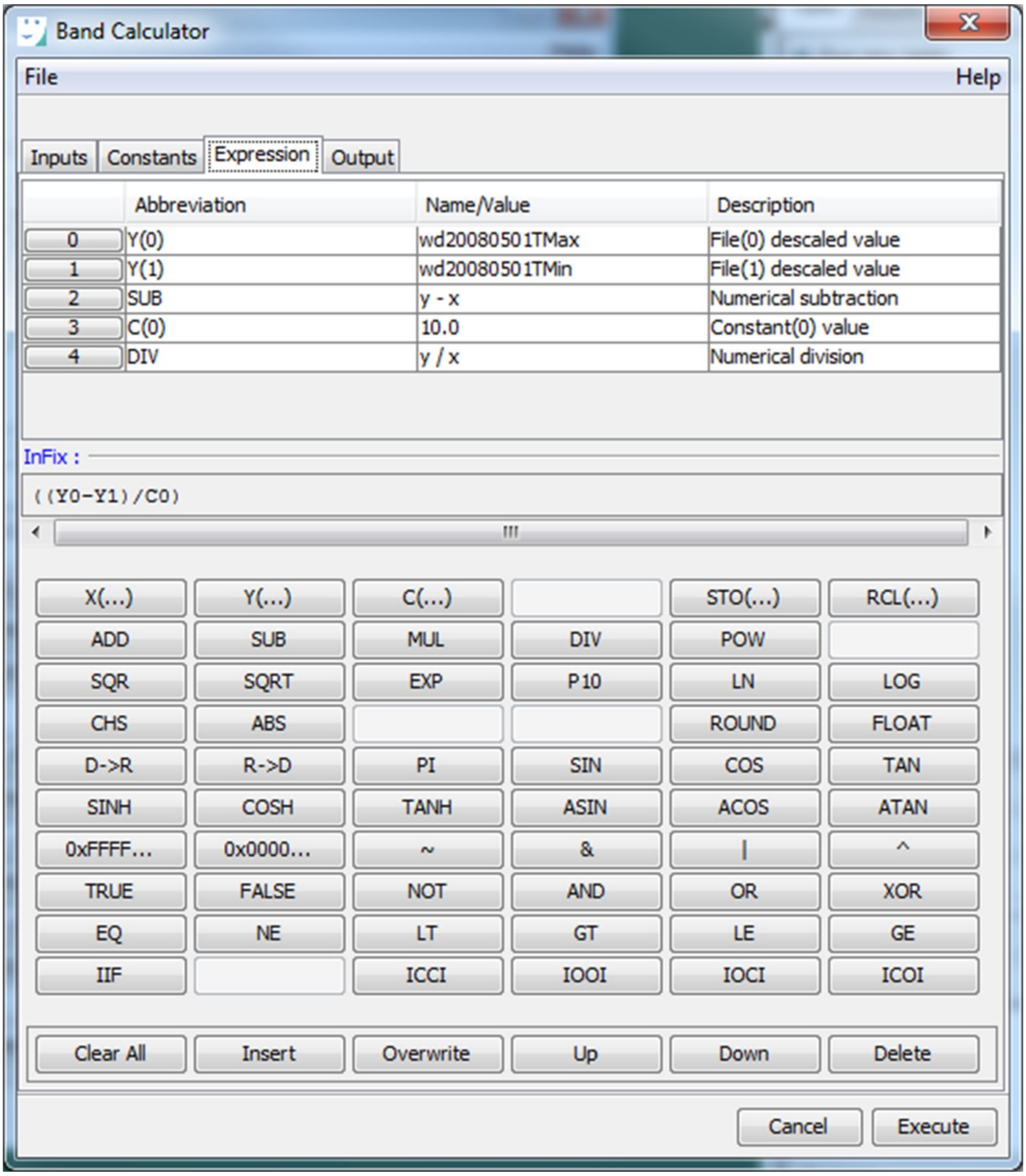
Integer values can be entered decimal, hexadecimal (starting with 0x) or binary (starting with 0b). To 'force' a value being treated as floating point one can add '.0'. When the value has been specified, additional information is available on the panel: whether it is considered as integer or floating point type, and for integer types the decimal, hexadecimal and binary representations of the value are displayed.

Its type is important for the implicit type conversions during calculation (e.g. adding two integers gives an integer result, whereas adding an integer and a floating point gives a floating point result.).

The hexadecimal and binary representations can be handy in case of bitwise operations.

If the "Is flag" option is selected, the value can be given a name, and the 'value=name' will be written in the flags entry of the output HDR.

3.9.2.3. Expression

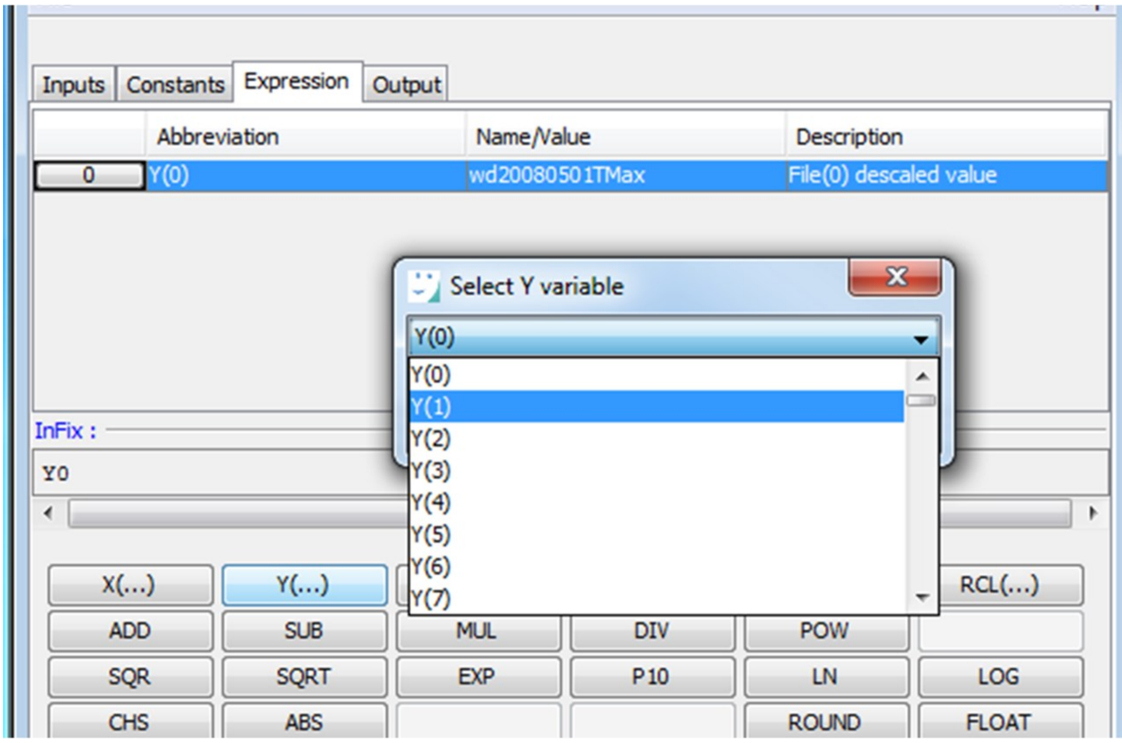


Band Calculator Tool - Expression panel

The expression panel contains the operation stack, a list of operands (X(i), Y(i) and C(i)) and operators representing the expression in RPN notation.

- operands and operators can be appended to the stack via the array of buttons mimicking the keypad of a calculator. Operators have their own button, X(i), Y(i) and C(i) operands can be selected via the X(...), Y(...) and C(...) buttons, which popup a panel allowing the selection of the operand with the specific index.;
- the selected entry can be moved up and down in the list ("Up" and "Down" buttons);
- the selected entry or the complete stack can be cleared ("Delete" or "Clear All" buttons);
- when pressing the "Insert" button, the next operand/operator will take the place of the selected entry, and shift the stack downwards;
- when pressing the "Overwrite" button, the next operand/operator will replace the selected entry.

Below the stack, as an aid, the expression is also displayed in its InFix form.



Band Calculator Tool - Expression pane - operand selection

The Band Calculator offers a set operators:

ADD	Numerical addition	y + x
SUB	Numerical subtraction	y - x
MUL	Numerical multiplication	y * x
DIV	Numerical division	y / x
POW	Numerical power	y ^ x

SQR	Numerical square	x ^ 2
SQRT	Numerical square root	x ^ (1/2)
EXP	Numerical Euler's ***e*** power	e^x
P10	Numerical 10 power	10 ^ x
LN	Numerical natural log (base ***e***)	ln(x)
LOG	Numerical base 10 logarithm	log(x)

<table>
<tr><td>CHS</td><td>Numerical change sign</td><td>+/- x</td></tr>
<tr><td>ABS</td><td>Numerical absolute value</td><td>abs(x)</td></tr>
<tr><td>ROUND</td><td>Numerical conversion float to integer</td><td>round(x)</td></tr>
<tr><td>FLOAT</td><td>Numerical conversion integer to float</td><td>float(x)</td></tr>
<tr><td>D->R</td><td>Numerical conversion Degrees to Radians</td><td>rad(x)</td></tr>
<tr><td>R->D</td><td>Numerical conversion Radians to Degrees</td><td>deg(x)</td></tr>
<tr><td>SIN</td><td>Numerical sine(Radians)</td><td>sin(x)</td></tr>
<tr><td>COS</td><td>Numerical cosine (Radians)</td><td>cos(x)</td></tr>
<tr><td>TAN</td><td>Numerical tangent(Radians)</td><td>tan(x)</td></tr>
<tr><td>ASIN</td><td>Numerical arc sine(Radians)</td><td>asin(x)</td></tr>
<tr><td>ACOS</td><td>Numerical arc cosine(Radians)</td><td>acos(x)</td></tr>
<tr><td>ATAN</td><td>Numerical arc tangent(Radians)</td><td>atan(x)</td></tr>
<tr><td>SINH</td><td>Numerical hyperbolic sine(Radians)</td><td>sinh(x)</td></tr>
<tr><td>COSH</td><td>Numerical hyperbolic cosine(Radians)</td><td>cosh(x)</td></tr>
<tr><td>TANH</td><td>Numerical hyperbolic tangent(Radians)</td><td>tanh(x)</td></tr>
<tr><td>~</td><td>Bitwise (integer) NOT operator</td><td>~ x</td></tr>
<tr><td>|</td><td>Bitwise (integer) OR operator</td><td>y | x</td></tr>
<tr><td>&</td><td>Bitwise (integer) AND operator</td><td>y & x</td></tr>
<tr><td>^</td><td>Bitwise (integer) XOR operator</td><td>y ^ x</td></tr>
<tr><td>NOT</td><td>Boolean NOT operator</td><td>(NOT p)</td></tr>
<tr><td>OR</td><td>Boolean OR operator</td><td>(q OR p)</td></tr>
<tr><td>AND</td><td>Boolean AND operator</td><td>(q AND p)</td></tr>
<tr><td>XOR</td><td>Boolean XOR operator</td><td>(q XOR p)</td></tr>
<tr><td>EQ</td><td>Numeric Equal condition</td><td>(y == x)</td></tr>
<tr><td>NE</td><td>Numeric Not Equal condition</td><td>(y != x)</td></tr>
<tr><td>LT</td><td>Numeric Less Then condition</td><td>(y < x)</td></tr>
<tr><td>GT</td><td>Numeric Greater Then condition</td><td>(y > x)</td></tr>
<tr><td>LE</td><td>Numeric Less Or Equal Then condition</td><td>(y <= x)</td></tr>
<tr><td>GE</td><td>Numeric Greater Or Equal Then condition</td><td>(y => x)</td></tr>
<tr><td>ICCI</td><td>In closed interval condition</td><td>(y <= z <= x)</td></tr>
<tr><td>IOOI</td><td>In open interval condition</td><td>(y < z < x)</td></tr>
<tr><td>IOCI</td><td>In left-open right-closed interval condition</td><td>(y < z <= x)</td></tr>
<tr><td>ICOI</td><td>In left-closed right-open interval condition</td><td>(y <= z < x)</td></tr>
<tr><td>IIF</td><td>Conditional operator</td><td>If z Then y Else x</td></tr>
<tr><td>STO</td><td>Store operator</td><td>Copy current operand to memory location i</td></tr>
<tr><td>RCL</td><td>Recall operator</td><td>Recall stored operand from memory location i</td></tr>
</table>

examples - simple expressions

X(0)	digital value as-is	copies the input band
Y(0)	physical value	descales the input band
Y(0) Y(1) ADD		adds the (physical values of the) input bands V = Y0 + Y1
C(0) Y(0) MUL C(1) ADD		linear transformation of the (physical values of the) input band : V = C1 + C0 * Y0

examples - conditional operator IIF

The IIF operator implements an in-line "if-then-else" structure. It needs three operands: a boolean operand and the two alternative result operands:

boolean operand 'true' case operand 'false' case operand IIF	"if" the boolean operands value is true "then" the result will be the 'true' case operand "else" the resulting operand will be the 'false' case operand

The 'true' and 'false" case operands, can be 'any' operand:

- an input variable X(i), Y(i);
- a constant C(i);
- the operand resulting from a single operation
- the operand resulting from an expression

boolean operand X(0) C(0) IIF		"if' the boolean operands value is true "then" result will be the X(0) input value "else" result will be the C(0) constant value
boolean operand		"if' the boolean operands value is true
Y(0) SQR	'true' case operand as result from a single operation	"then" result will be the operand resulting from the operation
C(0)	'false' case operand	"else" result will be the C(0) constant value
IIF		
boolean operand		"if' the boolean operands value is true
C(0) Y(0) MUL C(1) ADD	'true' case operand as result from an expression	"then" result will be the operand resulting from the expression
C(0)	'false' case operand	"else" result will be the C(0) constant value
IIF		

examples - boolean operators and operands

A boolean operand is an operand with value either true or false. Boolean operands are the result of conditions or boolean operators. Since it is the purpose of the calculator to create output images, a boolean operand should never be the "end result" of an expression, their primary goal is to facilitate the use of the IIF conditional operator.

Numerical binary compare conditions (EQ, NE, LT, GT, LE, GE) allow comparison between values, e.g. :

X(0)	compares the X(0) input value with the C(0) constant value,
C(0)	result is the "TRUE" operand in case X0 < C0
LT	result is the "FALSE" operand otherwise

Numerical ternary range or interval conditions (ICCI, IOOI, IOCI, ICOI) allow range checking, e.g. :

X(0)	
C(0)	checks if the X(0) input value is in closed interval [C(0), C(1)]
C(1)	result is the "TRUE" operand in case C0 <= X0 <= C1
ICCI	result is the "FALSE" operand otherwise

remark: the mnemonics indicate the type of interval:

- CC: left-closed, right-closed; ICCI: In Closed Closed Interval;
- OO: left-open, right-open; IOOI: In Open Open Interval;
- OC: left-open, right-closed; IOCI: In Open Closed Interval;
- CO: left-closed, right-open; ICOI: In Closed Open Interval.

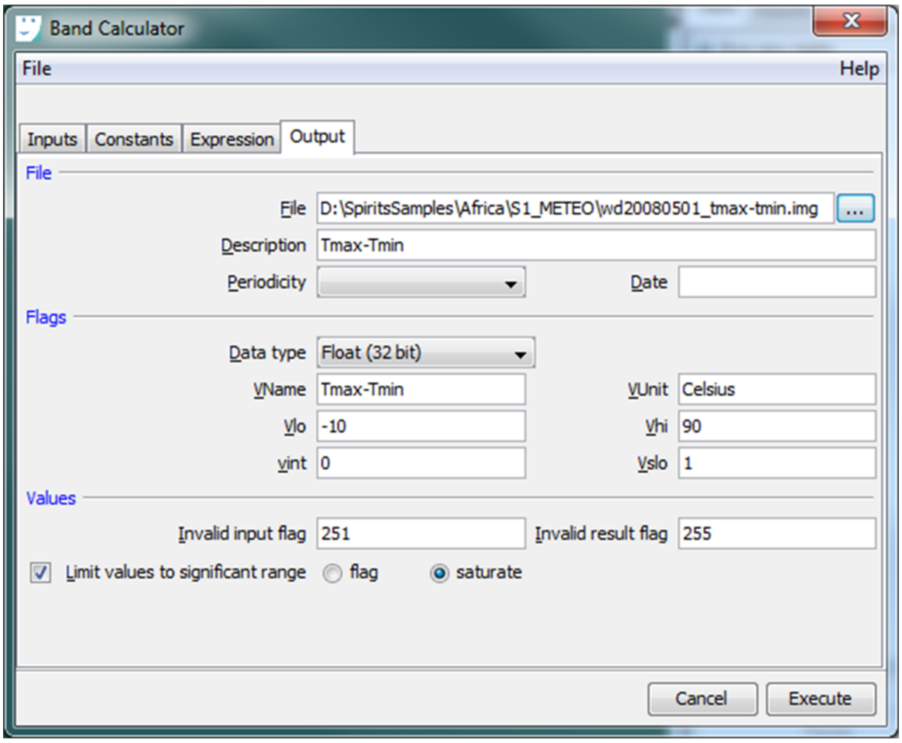
Boolean operators perform boolean operations (NOT, AND, OR, XOR) on boolean operands, e.g.

X(0)	compares the X(0) input value with the C(0) constant value,
C(0)	result is the "TRUE" operand in case X0 >= C0, "FALSE" otherwise
GE	=> this result is the first boolean operand for the AND operation
X(0)	compares the X(0) input value with the C(1) constant value,
C(1)	result is the "TRUE" operand in case X0 <= C1, "FALSE" otherwise
LE	=> this result is the second boolean operand for the AND operation
AND	TRUE if both boolean operands are true,
	hence in this case : if C0 <= X0 <= C1

example:

Y(0)	boolean operand	
Y(1)	TRUE if Y(0) > Y(1)	
GT	FALSE otherwise	
Y(0)	'true' case operand	
Y(1)	'false' case operand	
IIF		would result in the maximum of Y(0) and Y(1)

3.9.2.4. Output IMG and parameters



Band Calculator Tool - Output panel

The output panel contains the parameters specifying the output file.

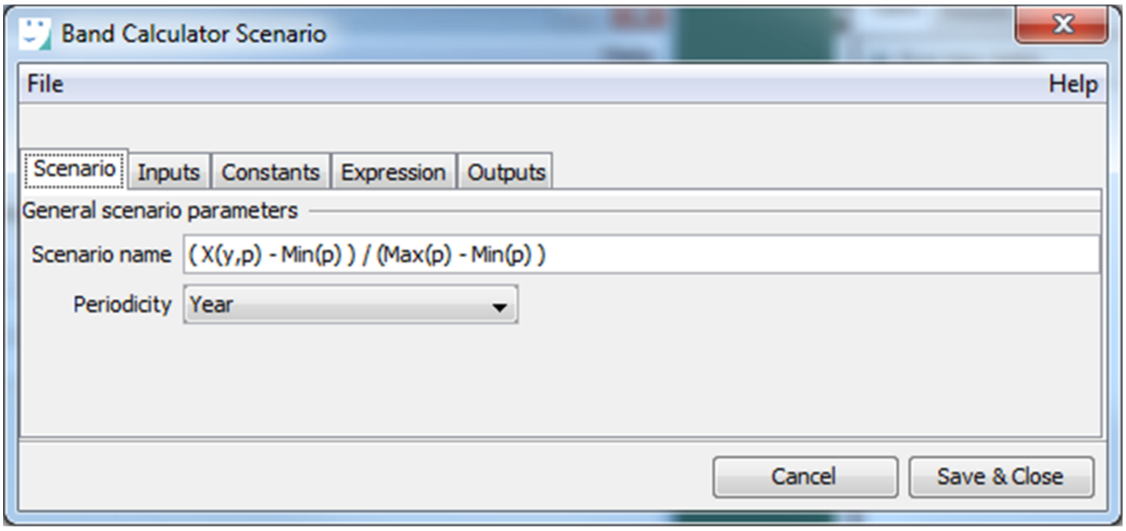
- the output IMG file;
- optional: the description, periodicity and date (YYYYMMDD) to be used in the output IMG HDR file;
- the data type of the output file. It is the responsibility of the user to make sure the output values fit in the selected datatype;
- optional: the name and unit of the results, to be used in the values entry of the output IMG HDR file;
- the significant range [Vlo, Vhi] and scaling (Vint, Vslo) of the results, to be used in the values entry of the output IMG HDR file;
- the value for the "Invalid input flag". This flag will be used for pixels where one (or more) of the inputs has its "limit values" option selected, and its value falls outside its significant range;
- the value for the "Invalid result flag". This flag will be used for pixels where the calculation fails (e.g. divisions by 0) and in case the "limit values" option for the output is selected, for pixels where the result falls outside the specified output significant range;
- the option whether or not to limit the resulting values to their significant range. If this option is selected, there are two options:
  - "flag": the pixels will be flagged with the "Invalid result flag" value;
  - "saturate": results < Vlo will be replaced by Vlo, results > Vhi will be replaced by Vhi.

In most cases (calculations of indexes), one would choose to limit the outputs to their significant range. However, allowing results outside the valid range makes sense for example in cases where one wants to create output flags via the expression itself.

3.9.3. Band Calculator Scenario and Time Series

The expression and constant panels of the Band Calculator Scenario are the same as those of the Tool. The Input and Output tabs however are specific, and an addition "Scenario" tab is present.

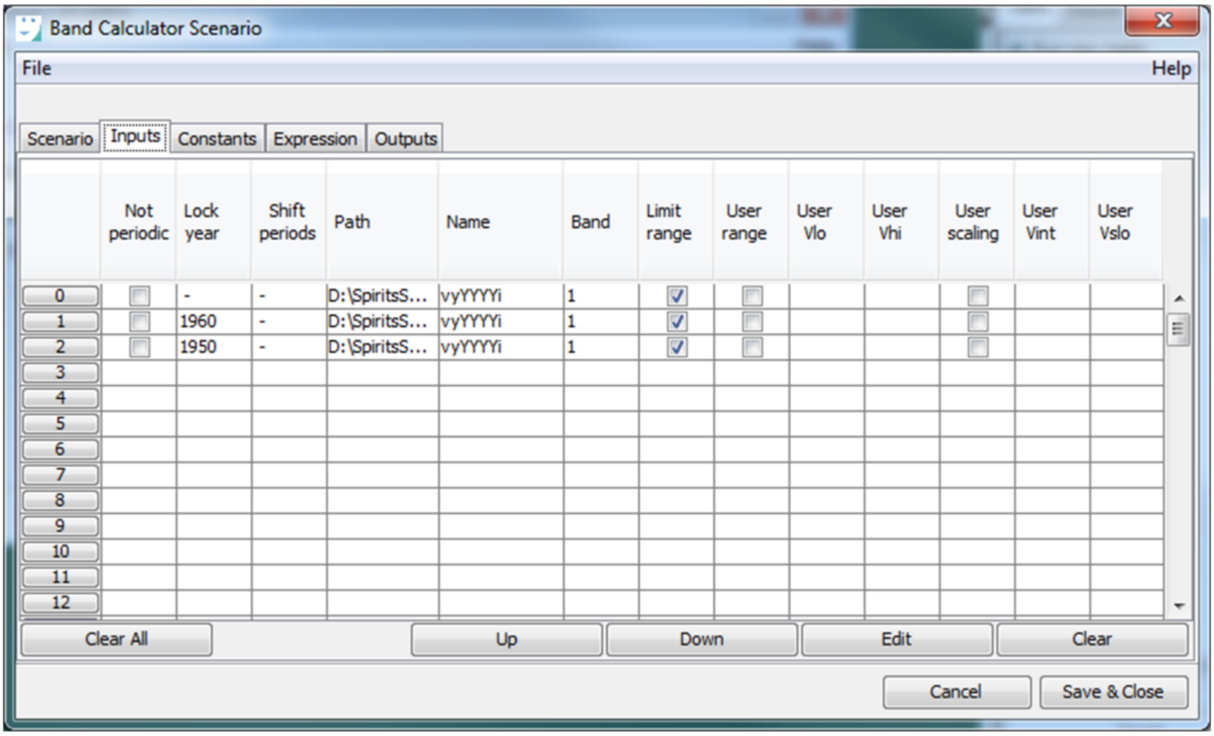
3.9.3.1. Scenario parameters



Band Calculator Scenario - Scenario panel

The scenario panel contains the basic scenario parameters; its name and periodicity.

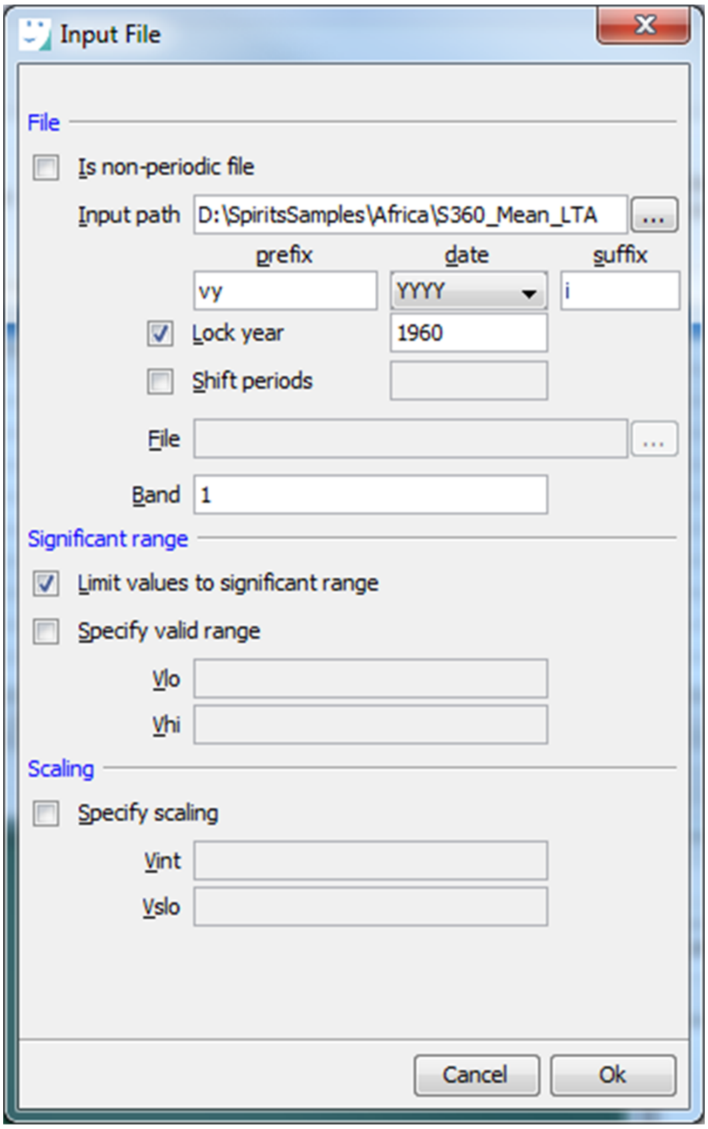
3.9.3.2. Scenario Input IMGs and parameters



Band Calculator Scenario - Input IMGs panel

The inputs panel and the relation between its entries and the variables ***X(i)*** and ***Y(i)*** in the expression are identical with the Tool.

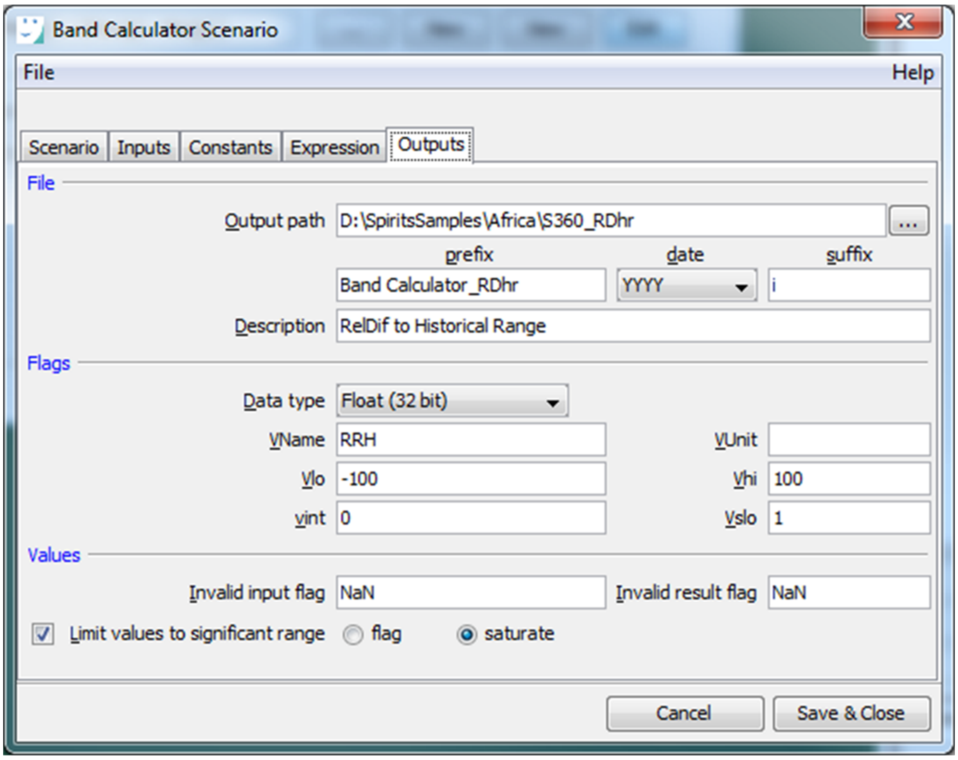
Also the significant range and scaling parameters of the actual entries are the same as for the tool. However there are some additional parameters.



Band Calculator Scenario - Input IMG parameters panel

- The input files can be periodic or non-periodic. In a time series we expect at least one periodic series of files. Non periodic files would typically be mask images e.g. Land/Sea masks.
- Periodic files have to be specified by their path, and file name prefix, date format and suffix.
- Periodic files have the "lock year" option. This allows specification of a constant YYYY value, enables the use of operations against some reference year, e.g. against the historical IMGs computed by the Long Term Average tool.
- Periodic files also have the "shift periods" option, which allows to specify a shift value relative to the period in the steps of the time series. This enables calculations against previous or following files in the same series, e.g. absolute or relative difference to previous period as in the Difference tool.

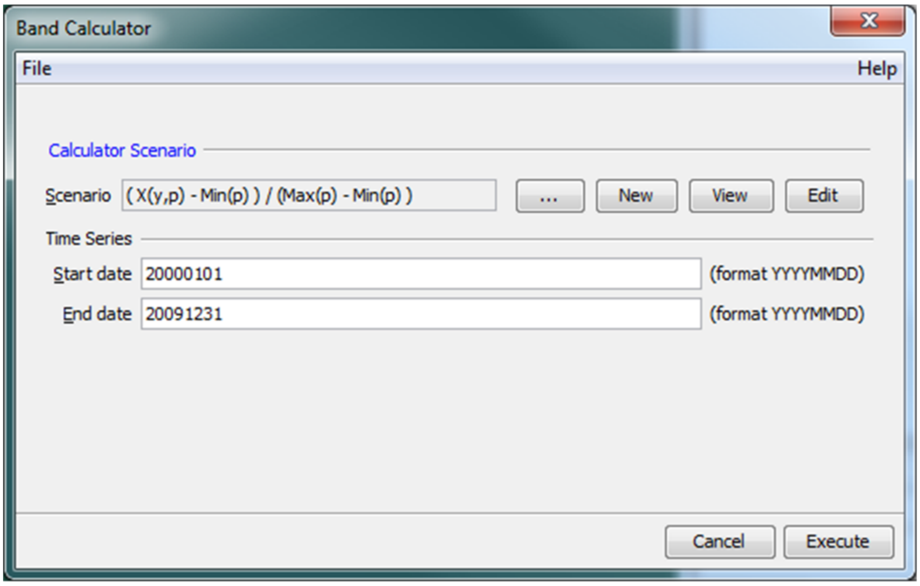
3.9.3.3. Scenario Output IMG and parameters



Band Calculator Scenario - Output panel

The (periodic) output files have to be specified by their path, and file name prefix, date format and suffix. For the rest the panel is identical with the Tool.

3.9.3.4. Time Series

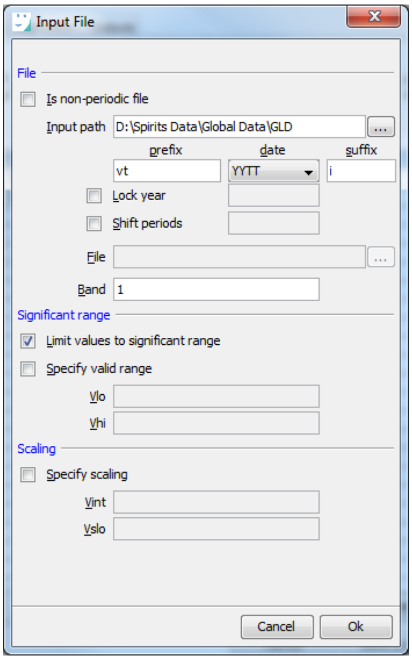


Band Calculator Time Series

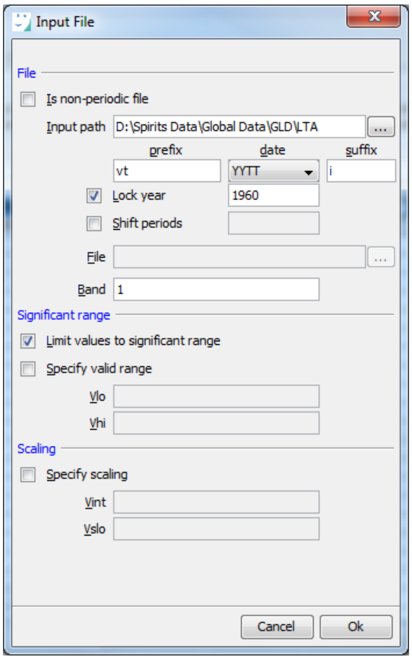
3.9.4. Examples

3.9.4.1. Example - Index

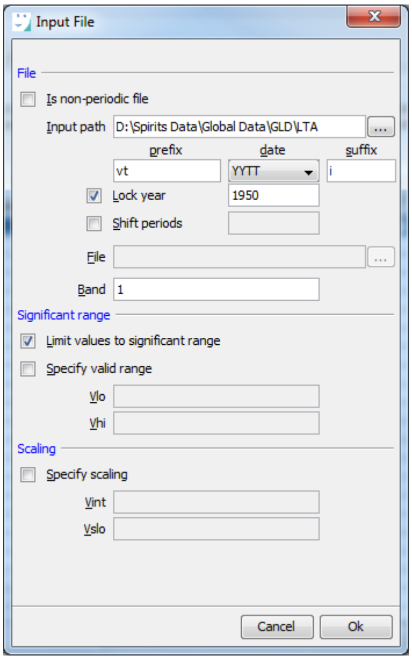
The Band Calculator can be used as an alternative for the Index tool operations. In this example the VCI will be calculated: VCI = NDVI - NDVI(min) / NDVI(max) - NDVI(min). By using the "Lock year" option of the Input IMGs in the scenario we can use the LTA max (indicated by year 1960) and LTA min (indicated by year 1950) images created by the Long Term Average Tool for the Time Series.



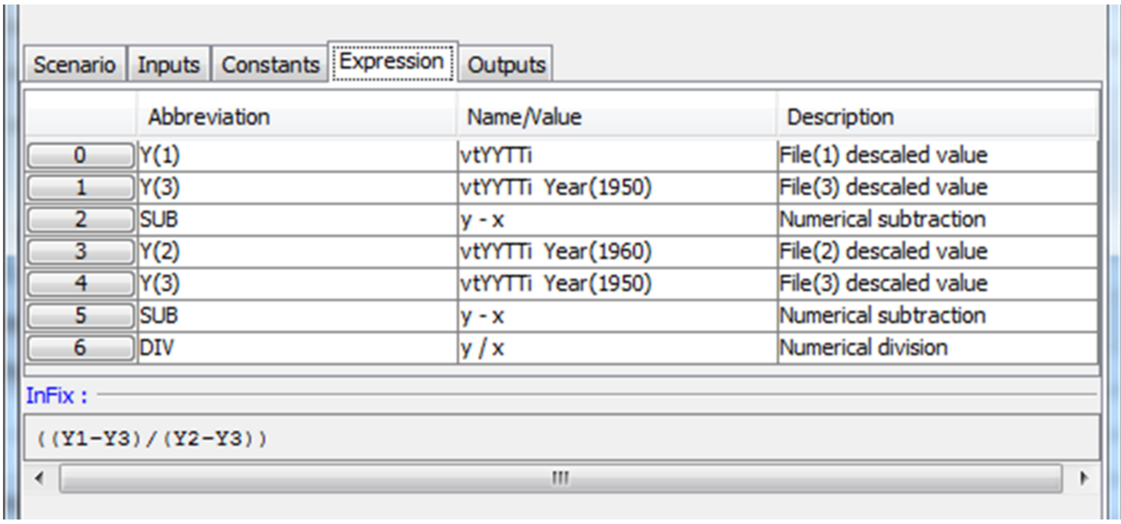
input IMG (1) periodic



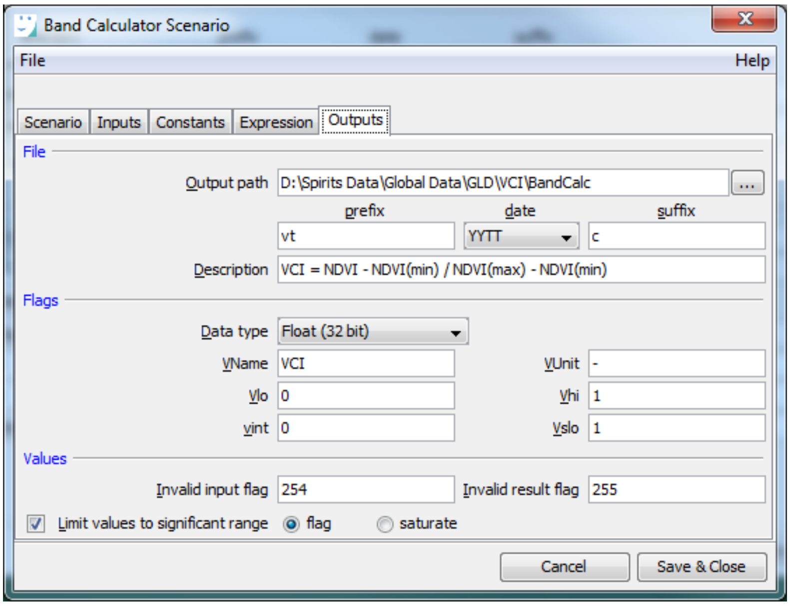
input IMG (2) periodic- LTA max



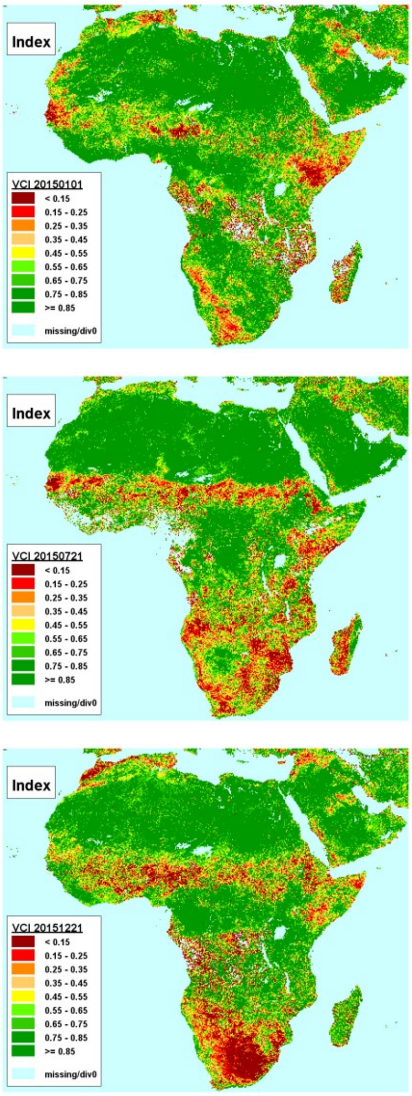
input IMG (3) periodic- LTA min



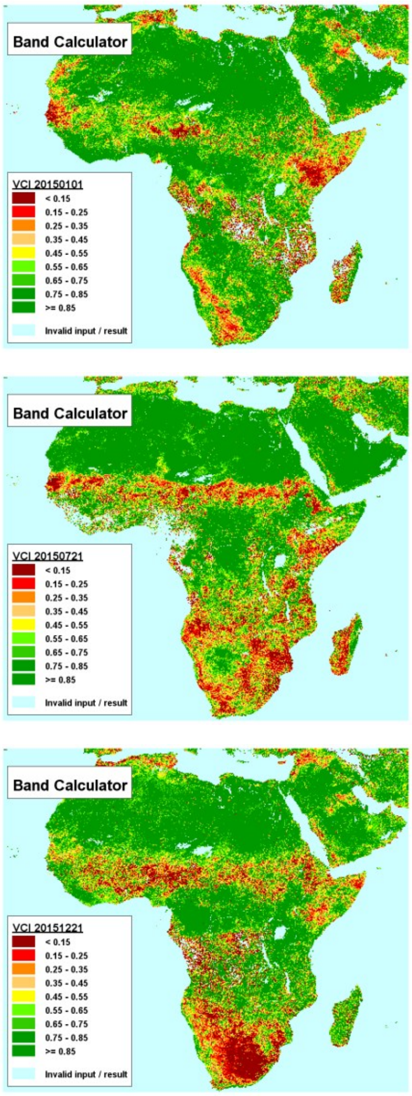
Band Calculator Expression - NDVI - NDVI(min) / NDVI(max) - NDVI(min)



Band Calculator Output



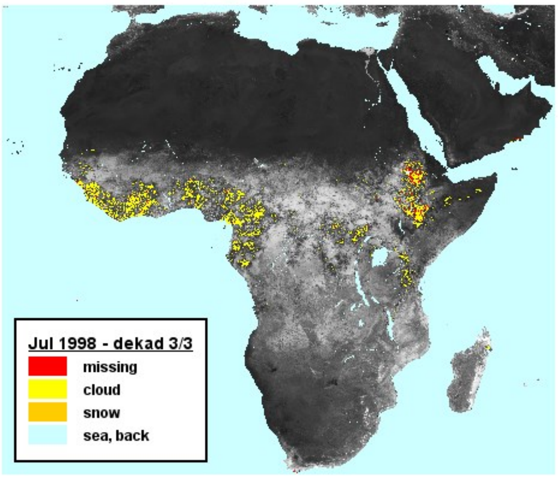
VCI - via Index



VCI - via Band Calculator

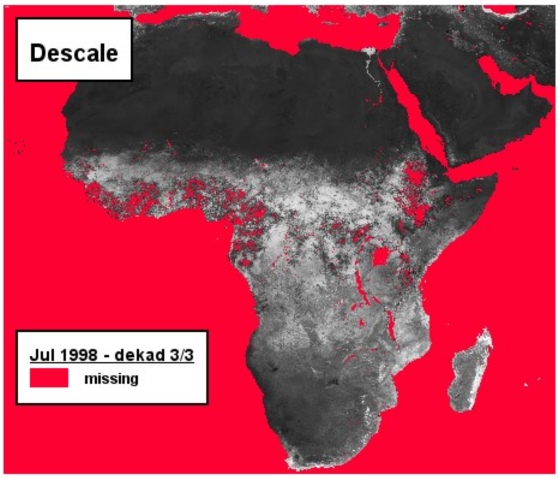
3.9.4.2. Example - Descale

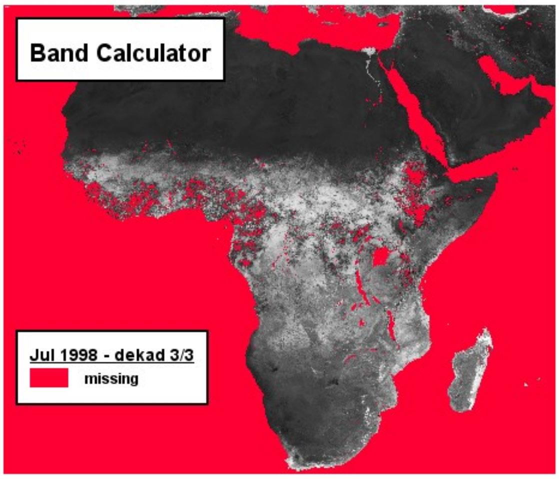
The Band Calculator can be used as an alternative for the Descale IMG tool. The input uses the 'limit range' option. The expression is simply Y(0). The output IMG Data type is Float, its scaling Vint = 0 and Vslo =1.

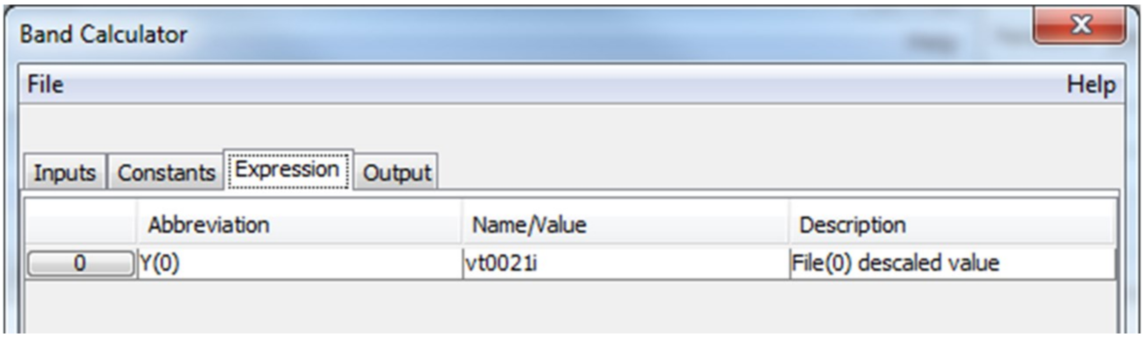


In this specific example we find from the input IMG values entry, {NDVI-toc, -, 0, 250, 0, 250, -0.08, 0.004}, its physical range [Vlo, Vhi] :
[-0.08 + 0.004 *0, -0.08 + 0.004 * 250] or [-0.08, 0.92].

Since the expression is simply Y(0), this is also our output range.







Band Calculator expression for "Descale"

The problem however - with the Descale IMG tool- is that all flag information is lost, that is to say all pixels beyond [0,250] are flagged as "Invalid input". For example, there is no longer a distinction between "missing" and "sea" pixels.

3.9.4.3. Example - Descale, keeping the flags

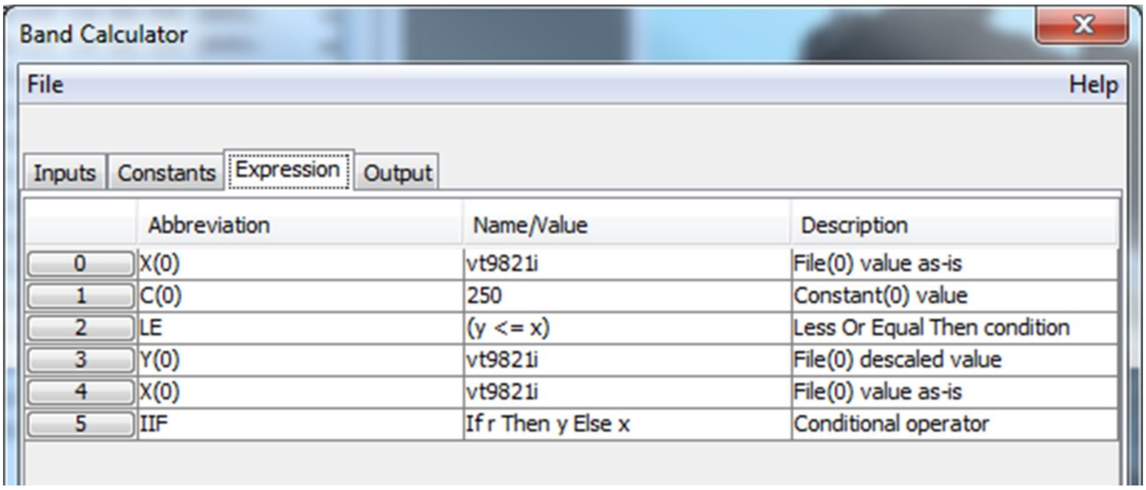
By NOT selecting the "limit values" of the input IMG, we allow the values beyond its significant range (which would be the input IMG flags) to enter the expression, where we can treat them separately.

In this specific example we find from the input IMG values entry, {NDVI-toc, -, 0, 250, 0, 250, -0.08, 0.004}, its digital range [0,250]. Since it is a byte-type IMG there cannot be any values less than 0, so we only need to distinguish the digital values in the significant range [0,250], which we want de-scaled, and the values [251,255] which are flags values.

Since physical range of the descaled values in the output IMG [Vlo, Vhi] is [-0.08, 0.92], there is no overlap between these de-scaled values and the 'original' flag values, thus we can keep them as they are.

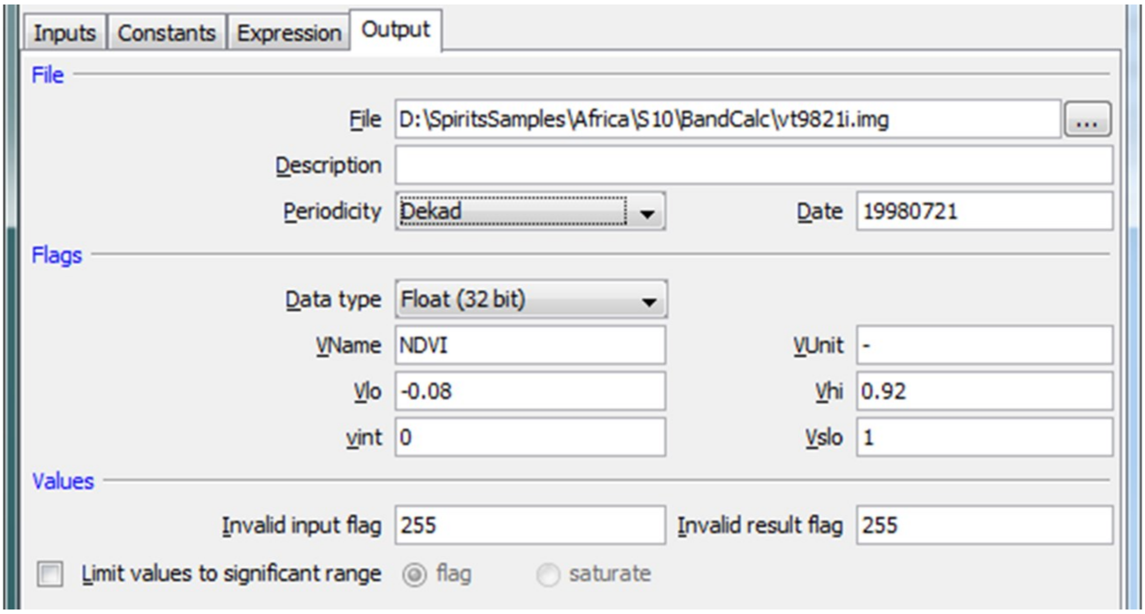
The expression then becomes:

X(0)	digital value 0-255 of the input IMG	boolean operand for conditional (IIF) operator
C(0)	constant 250 - limits the significant range of the input IMG	
LE	compare operator: - TRUE if the digital value is in its significant range, - FALSE in the flags range	
Y(0)	descaled value of the input IMG	'true' case operand for IIF operator
X(0)	digital value as-is of the input IMG	'false' case operand for IIF operator
IIF	conditional operator: - if the boolean operator is true, the first option (Y0) is selected - the descaled value - otherwise the second option (X) is selected - the flag value as-is;	



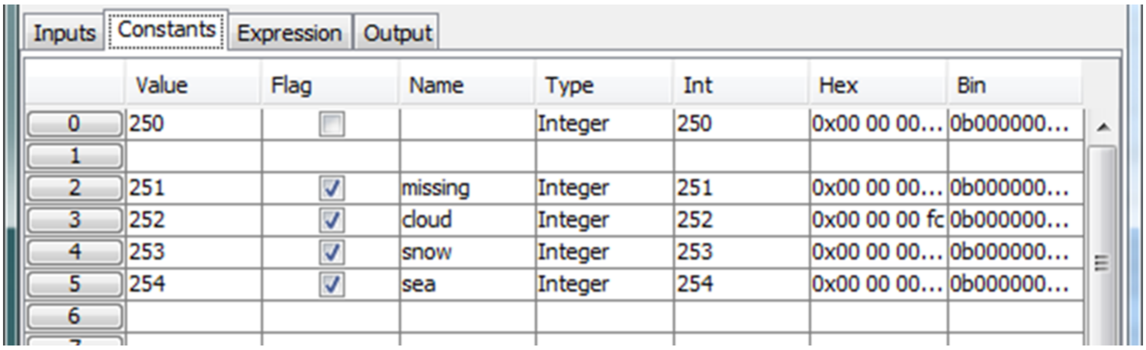
Band Calculator expression for "Descale" keeping the input flags.

In the output IMG HDR we want a meaningful values entry, thus we need to specify its significant range as [-0.08, 0.92]. But since the flag values (251,..255) travel through the expression too, and we want to keep them, we may NOT select the "limit values to significant range" option for the output values.

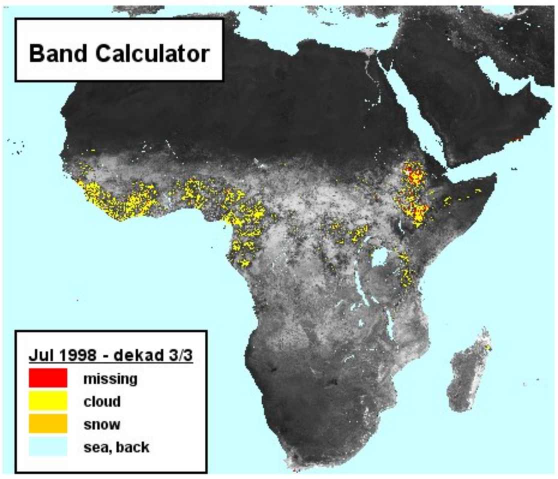


Band Calculator output for "Descale" keeping the input flags.

To get the actual list of flags in the output IMG HDR, it suffices to declare them as constants, with the "Is flag" option selected.



Band Calculator constants for "Descale" keeping the input flags.

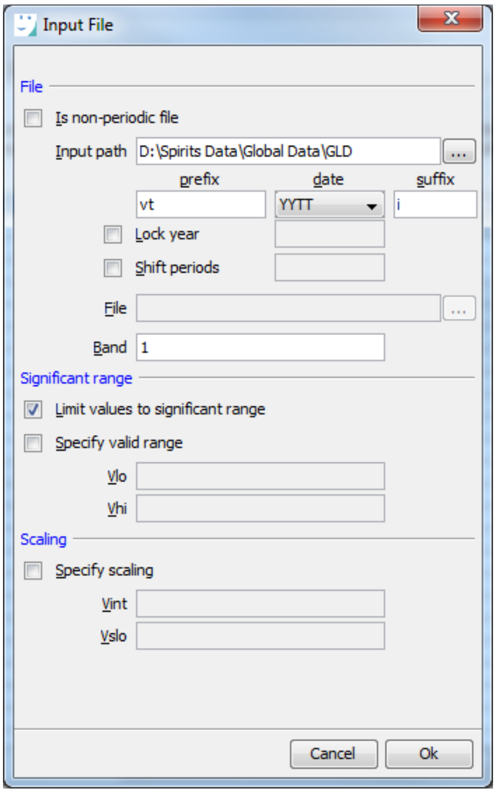


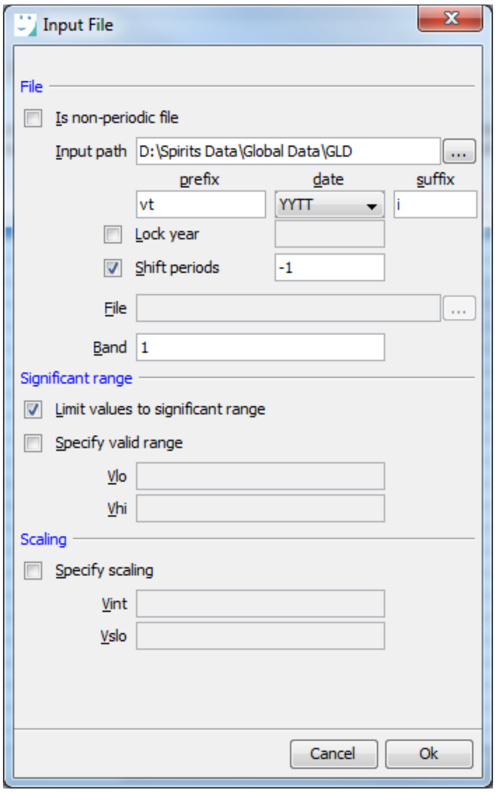
HDR:

```
...
data type = 4
values = {NDVI, -, -0.08, 0.92, -0.08, 0.92, 0, 1}
flags = {251=missing, 252=cloud, 253=snow, 254=sea,
255=Invalid input, 255=Invalid result}
...
```

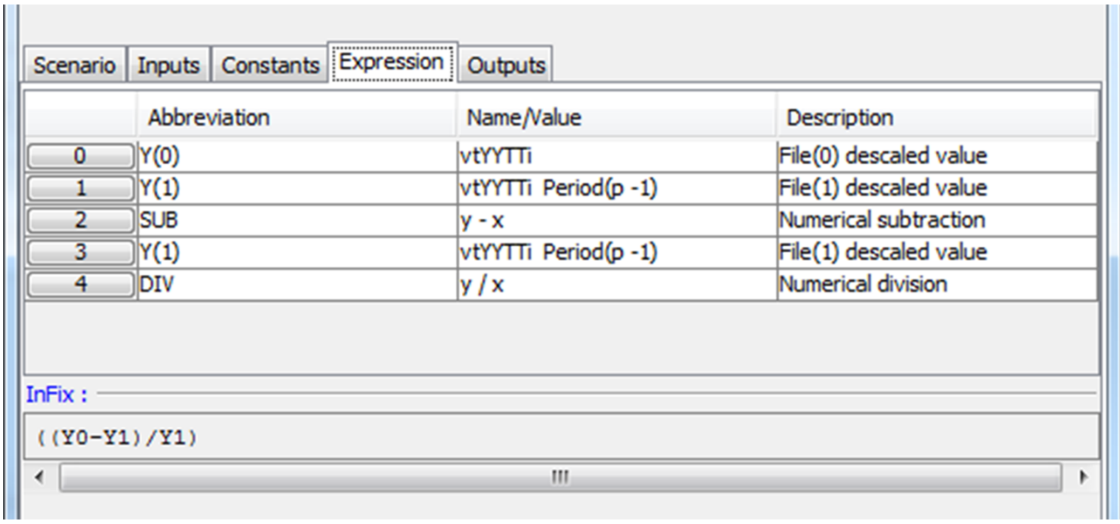
3.9.4.4. Example - Difference

The Band Calculator can be used as an alternative for some of the Difference tool operations. By using the "shift periods" option of the Input IMGs in the scenario, we can calculate the "Relative difference to previous period" (RDpp = [Y(p) - Y(p-1)] / y(p-1)).

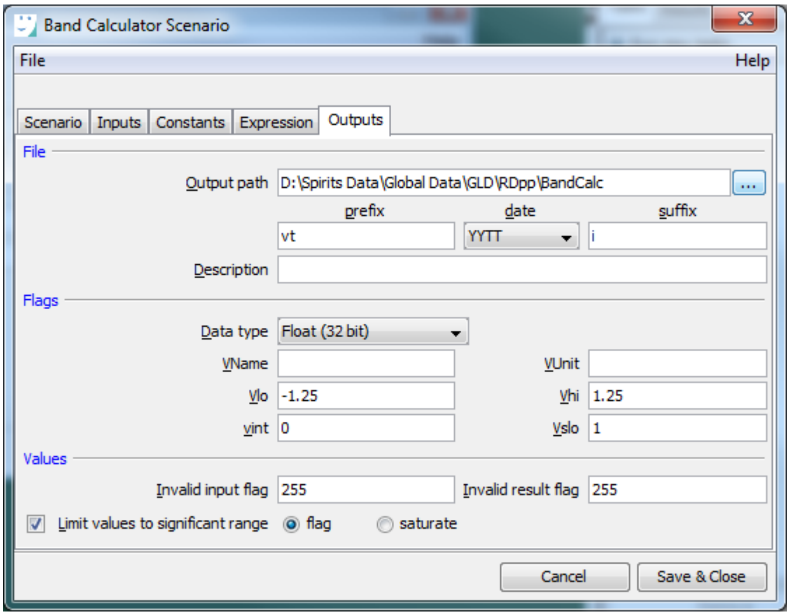




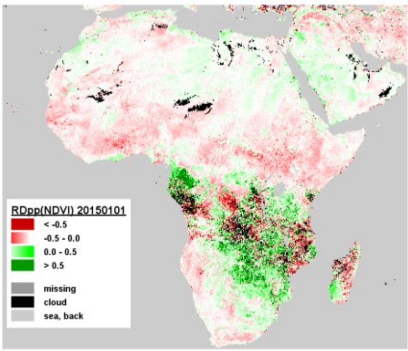
Band Calculator Scenario - input IMG (0) periodic - input IMG (1) periodic, previous period

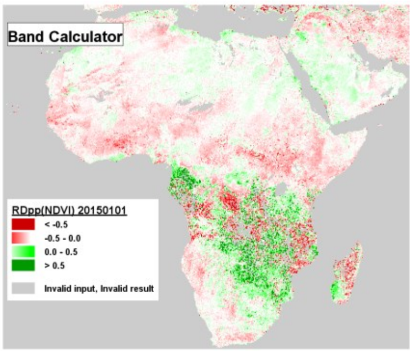


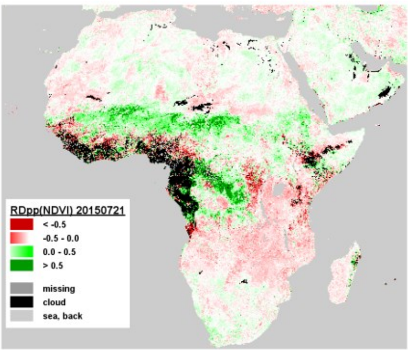
Band Calculator Expression - Relative difference to previous period

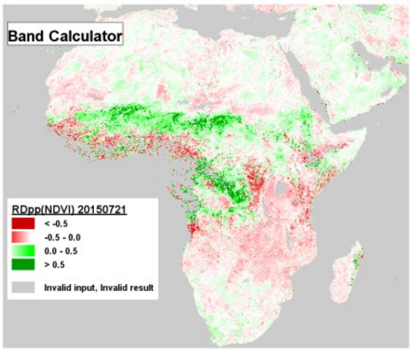


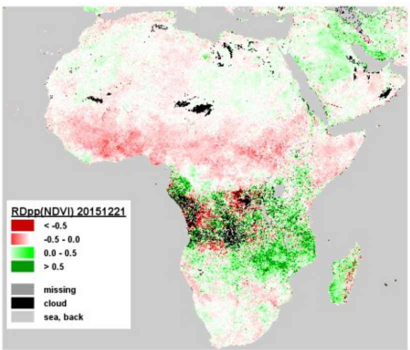
Band Calculator Output

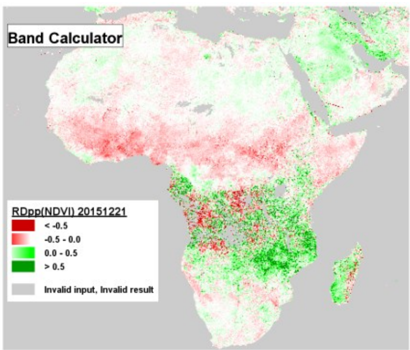












As can be seen in the resulting images, the significant result values from Differ and the Band Calculator are the same, however, the different input flags are all mapped to the single "Invalid input" flag.

3.9.4.5. Example - Difference, keeping the flags

By NOT selecting the "limit values" of the input IMG, we allow the values beyond their significant range (which would be the input IMG flags) to enter the expression, where we can treat them separately.

In this specific example we find from the input IMG values entry, {NDVI-toc, -, 0, 250, 0, 250, -0.08, 0.004}, its digital range [0,250]. Since it is a byte-type IMG there cannot be any values less than 0, so we only need to distinguish the digital values in the significant range [0,250], and the values [251,255] which are flags values. However, since there are two input files, we need to decide what we want for each case, e.g:

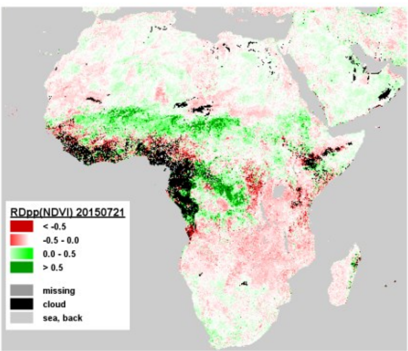
- in case the values are in significant range in BOTH files, we want the RDpp calculation as result;
- else in case the values of IMG(0) (period p) are flagged, we'll use the IMG(0) flag value;
- else in case the values of IMG(1) (period p-1) are flagged, we'll use the IMG(1) flag value;

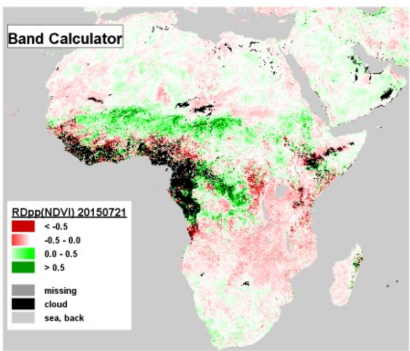
Since physical range of the descaled values in the output IMG [Vlo, Vhi] is [-1.25, 1.25], there is no overlap between these RDpp values and the 'original' flag values, thus we can keep these as they are.

The expression needs a nested condition:

X(0)	digital value 0-255 input IMG(0)		boolean operand for outer conditional (IIF) operator
C(0)	const 250 - limits range of IMG(0)		
GT	compare operator: - TRUE if IMG(0) is some flagged, - FALSE if IMG(0) value is significant		
X(0)	flagged value of the input IMG(0)		'true' case operand for outer conditional (IIF)
X(1)	digital value 0-255 input IMG(1)	boolean operand for inner conditional (IIF) operator	'false' case operand for outer conditional (IIF)
C(0)	const 250 - limits range of IMG(1)		
GT	compare operator: - TRUE if IMG(1) is some flagged, - FALSE if IMG(1) value is significant		
X(1)	flagged value of the input IMG(1)	'true' case operand for inner conditional (IIF)	
Y(0)		actual RDpp calculation 'false' case operand for inner conditional IIF	
Y(1)			
SUB			
Y(1)			
DIV			
IIF	(inner) conditional operator	=> result inner IFF	
IIF	(outer) conditional operator		=> result outer IFF

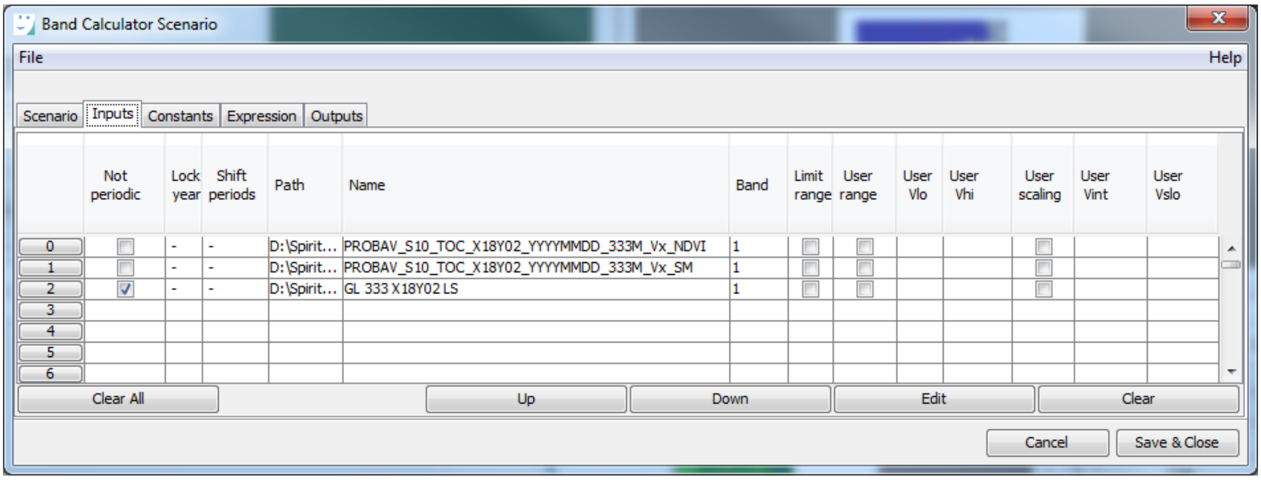
And as in the Descale example, since the flag values (251,..255) travel through the expression too, and we want to keep them, we may NOT select the "limit values to significant range" option for the output values, and to get the actual list of flags in the output IMG HDR, we declare them as constants, with the "Is flag" option selected.





3.9.4.6. Example - Masking

By combining bitwise and boolean operations, the Band Calculator can be used for masking operations such as applying status masks or land/sea masks. In this example a (Proba-V 333M) status mask and a land/sea mask will be applied on raw NDVI data.



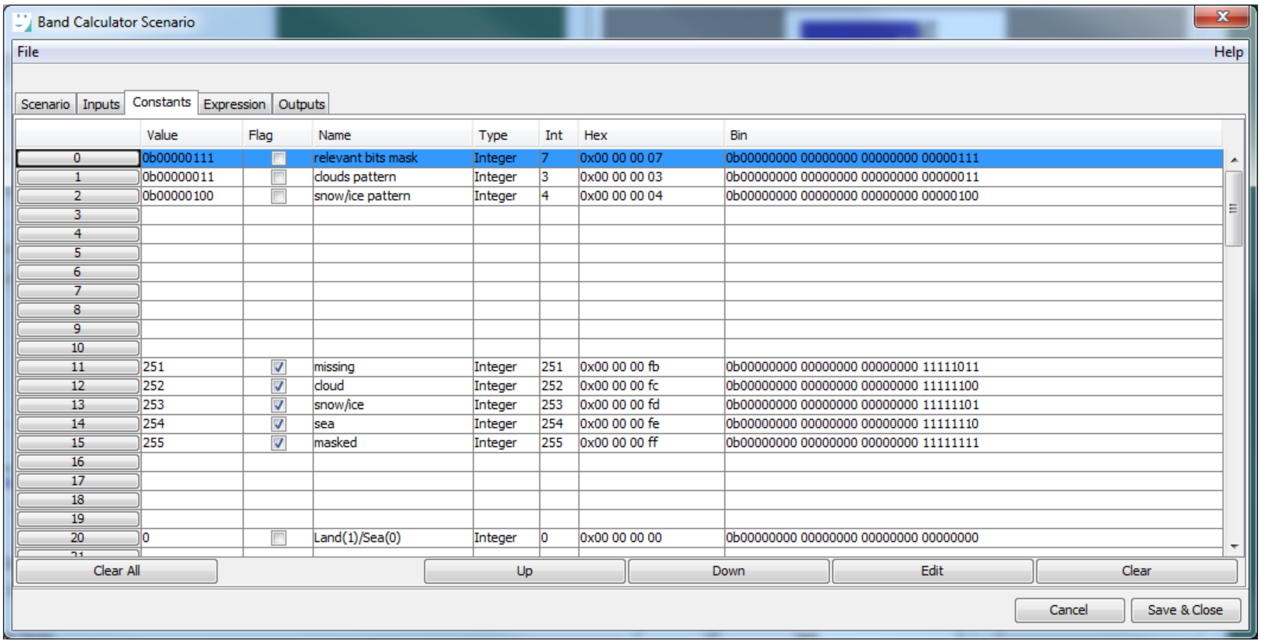
Band Calculator Inputs:

(0) : the raw NDVI data
(1) : the status mask
(2) : a land /sea mask

From the status mask, we'll only consider the three least significant bits, which are defined as follows: 000 = clear, 010 = undefined, 011 = cloud, 100 = ice/snow, 001 = shadow.

The land/sea mask has only two values: 0 = sea, 1 = land.

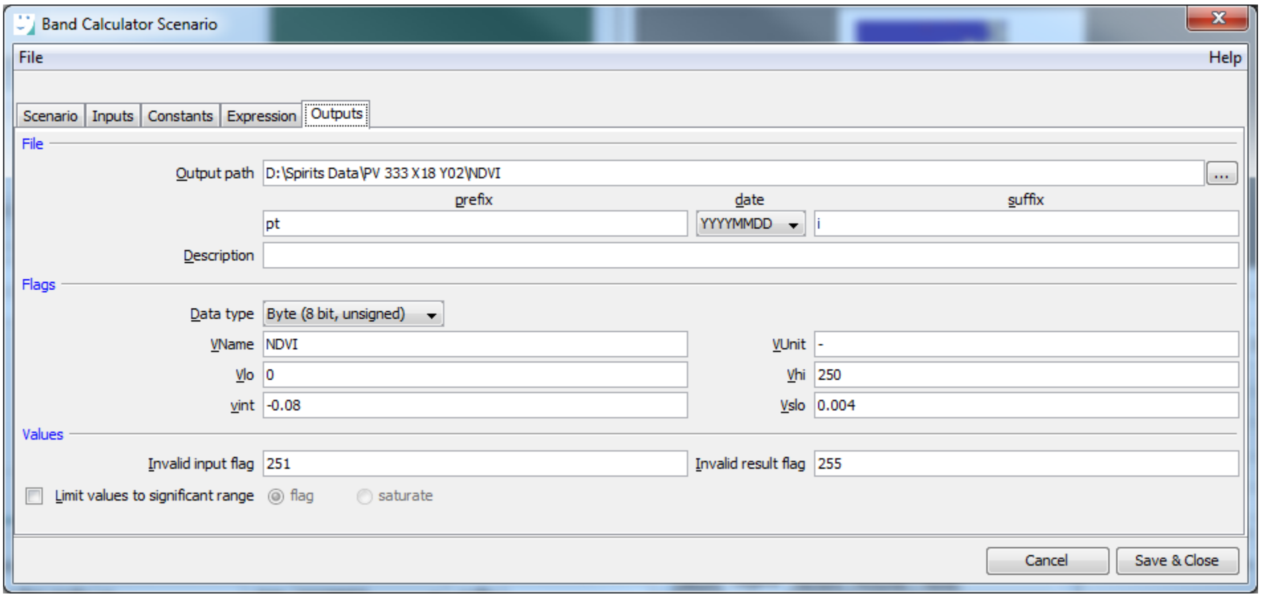
The resulting IMG should contain the original NDVI data only if the status mask indicates clear sky. Sea, clouds and snow/ice should be flagged explicitly. For this purpose we need constants for masks and flags:



Band Calculator Constants:

(0)-(2) : binary mask and values
(11-15) : output flags
(20) : land/sea mask "sea" value

Since we're not manipulating the NDVI values, the output characteristics (Vlo, Vhi, Vint, Vslo) are those from the NDVI input file.



Band Calculator Output

	Abbreviation	Name/Value	Description
0	X(2)	GL 333 X18Y02 LS	File(2) value as-is
1	C(20)	0 - Land(1)/Sea(0)	Constant(20) value
2	EQ	(y == x)	Numeric Equal condition
3	C(14)	254 - flag: sea	Constant(14) value (Flag)
4	X(1)	PROBAV_S10_TOC_X18Y02_YYYYMMDD_333M_Vx_SM	File(1) value as-is
5	C(0)	7 - relevant bits mask	Constant(0) value
6	&	y & x	Bitwise (integer) AND operator
7	C(1)	3 - clouds pattern	Constant(1) value
8	EQ	(y == x)	Numeric Equal condition
9	C(12)	252 - flag: cloud	Constant(12) value (Flag)
10	X(1)	PROBAV_S10_TOC_X18Y02_YYYYMMDD_333M_Vx_SM	File(1) value as-is
11	C(0)	7 - relevant bits mask	Constant(0) value
12	&	y & x	Bitwise (integer) AND operator
13	C(2)	4 - snow/ice pattern	Constant(2) value
14	EQ	(y == x)	Numeric Equal condition
15	C(13)	253 - flag: snow/ice	Constant(13) value (Flag)
16	X(1)	PROBAV_S10_TOC_X18Y02_YYYYMMDD_333M_Vx_SM	File(1) value as-is
17	C(0)	7 - relevant bits mask	Constant(0) value
18	&	y & x	Bitwise (integer) AND operator
19	0x00000000	0x00	Integer(0x00000000) constant
20	NE	(y != x)	Numeric Not Equal condition
21	C(15)	255 - flag: masked	Constant(15) value (Flag)
22	X(0)	PROBAV_S10_TOC_X18Y02_YYYYMMDD_333M_Vx_NDVI	File(0) value as-is
23	IIF	If z Then y Else x	Conditional operator
24	IIF	If z Then y Else x	Conditional operator
25	IIF	If z Then y Else x	Conditional operator
26	IIF	If z Then y Else x	Conditional operator

Band Calculator Expression

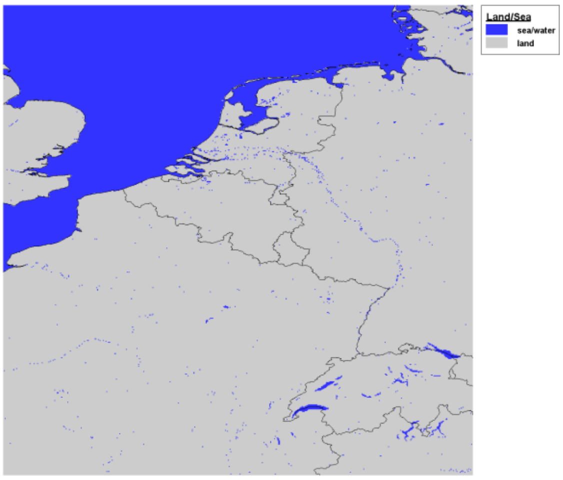
step	description
0 - 2 and 26 (IIF)	compares the land/sea image with its sea value
3	If so, the pixel is flagged with the flag 254=sea
	else
4 - 8 and 25 (IIF)	masks the status mask image to obtain the three least significant bits and compares the result with the clouds pattern.
9	If so, the pixel is flagged with the flag 252=cloud
	else
10 -14 and 24(IIF)	masks the status mask image to obtain the three least significant bits and compares the result with the snow/ice pattern.
15	If so, the pixel is flagged with the flag 253=snow/ice
	else
16 - 20 and 23 (IIF)	masks the status mask image to obtain the three least significant bits and compares the result with 0x00 indicating clear sky.
21	If NOT so, the pixel is flagged with the flag 255
	else
22	the NDVI value as-is from the ndvi image is passed
23 (IIF)	
24 (IIF)	
25 (IIF)	
26 (IIF)	

Examining the expression we see the sequence needed to mask the relevant bits from the status mask occurs three times. We can simplify this by using the memory operators STO/RCL

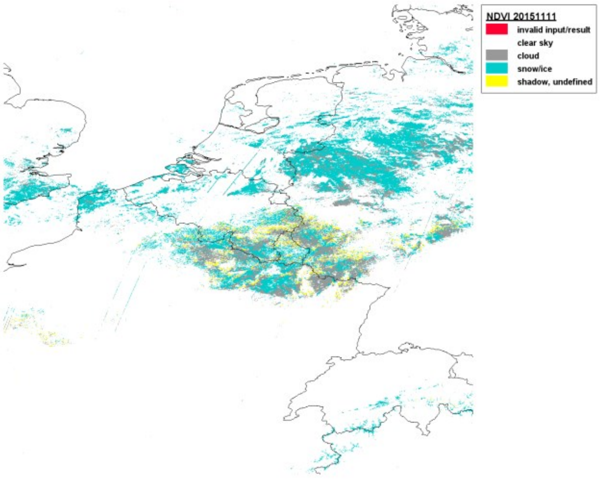
Scenario | Inputs | Constants | Expression | Outputs

	Abbreviation	Name/Value	Description
0	X(2)	GL 333 X18Y02 LS	File(2) value as-is
1	C(20)	0 - Land(1)/Sea(0)	Constant(20) value
2	EQ	(y == x)	Numeric Equal condition
3	C(14)	254 - flag: sea	Constant(14) value (Flag)
4	X(1)	PROBAV_S10_TOC_X18Y02_YYYYMMDD_333M_Vx_SM	File(1) value as-is
5	C(0)	7 - relevant bits mask	Constant(0) value
6	&	y & x	Bitwise (integer) AND operator
7	STO(0)	store x to location(0)	Copy current operand to location(0)
8	C(1)	3 - clouds pattern	Constant(1) value
9	EQ	(y == x)	Numeric Equal condition
10	C(12)	252 - flag: cloud	Constant(12) value (Flag)
11	RCL(0)	recall location(0) as x	Recall stored operand from location(0)
12	C(2)	4 - snow/ice pattern	Constant(2) value
13	EQ	(y == x)	Numeric Equal condition
14	C(13)	253 - flag: snow/ice	Constant(13) value (Flag)
15	RCL(0)	recall location(0) as x	Recall stored operand from location(0)
16	0x00000000	0x00	Integer(0x00000000) constant
17	NE	(y != x)	Numeric Not Equal condition
18	C(15)	255 - flag: masked	Constant(15) value (Flag)
19	X(0)	PROBAV_S10_TOC_X18Y02_YYYYMMDD_333M_Vx_NDVI	File(0) value as-is
20	IIF	If z Then y Else x	Conditional operator
21	IIF	If z Then y Else x	Conditional operator
22	IIF	If z Then y Else x	Conditional operator
23	IIF	If z Then y Else x	Conditional operator

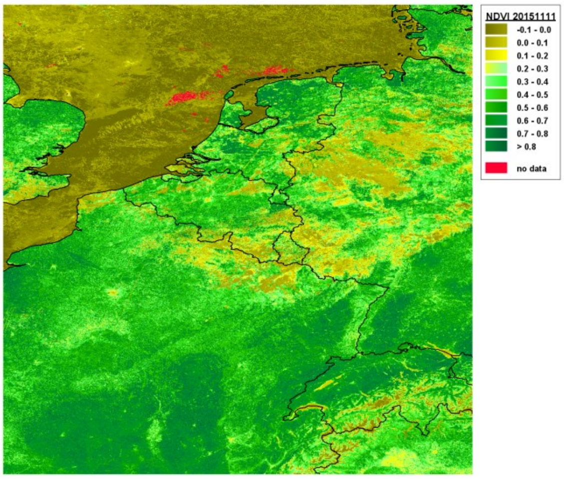
Band Calculator Expression - using STO/RCL



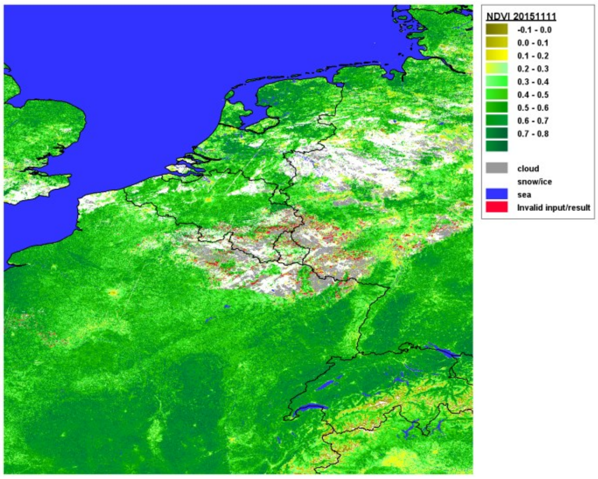
land/sea mask



status mask



raw NDVI IMG



flagged NDVI IMG

3.10. Vegetation Health Index

Goal

Compute Vegetation Health Index (VHI) IMG from Vegetation Condition Index (VCI) and Temperature Condition Index (TCI) IMG.

Parameters

- the VCI IMG;

  Remark: starting from NDVI IMGs,

  - VCI, defined as (NDVI-NDVImin)/(NDVImax-NDVImin) can be calculated using the Difference Tool;
  - the historical NDVImax and NDVImin, can be calculated using the Long Term Average Tool.

- the TCI IMG;

  Remark: starting from TI (Temperature Index) IMGs (LST, BT4,...),

  - TCI1, defined as (T-Tmin)/(Tmax-Tmin) can be calculated using the Difference Tool;
  - the historical Tmax and Tmin, can be calculated using the Long Term Average Tool.

- the TCI type. Specifies the TCI IMG used:

  - TCI1 = (T-Tmin)/(Tmax-Tmin) (as can be calculated by the Difference Tool);
  - TCI0 = (Tmax-T)/(Tmax-Tmin) = 1 - TCI1

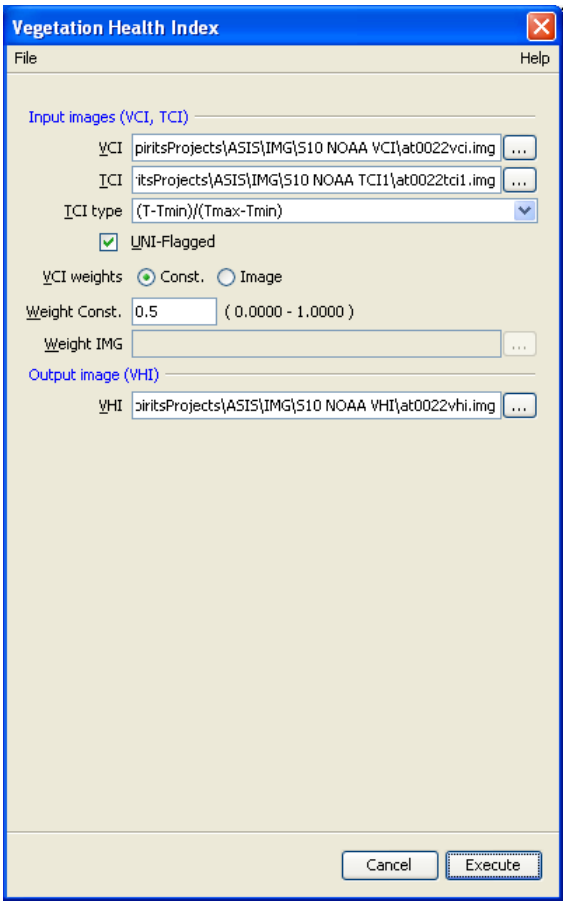
- whether or not the input IMGs are UNI-Flagged;

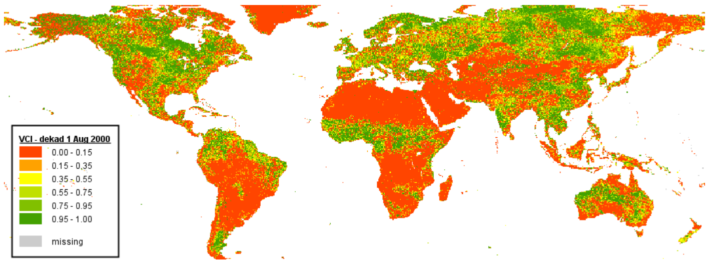
  - if input IMGs are UNI-Flagged, the output IMG can also be UNI-Flagged;
  - otherwise a single flag will be used in the output IMG.

- the VCI weight(s): a constant factor or an IMG file containing the weights per pixel:
  VHI = weight.VCI + (1-weight).TCI0
  The weight(s) can be specified as:

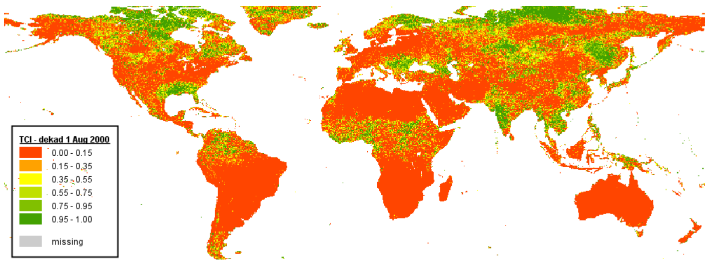
  - a constant factor. In this case the factor must be specified (0-1);
  - an IMG file containing the weight per pixel. In this case the IMG must be specified.

Tool



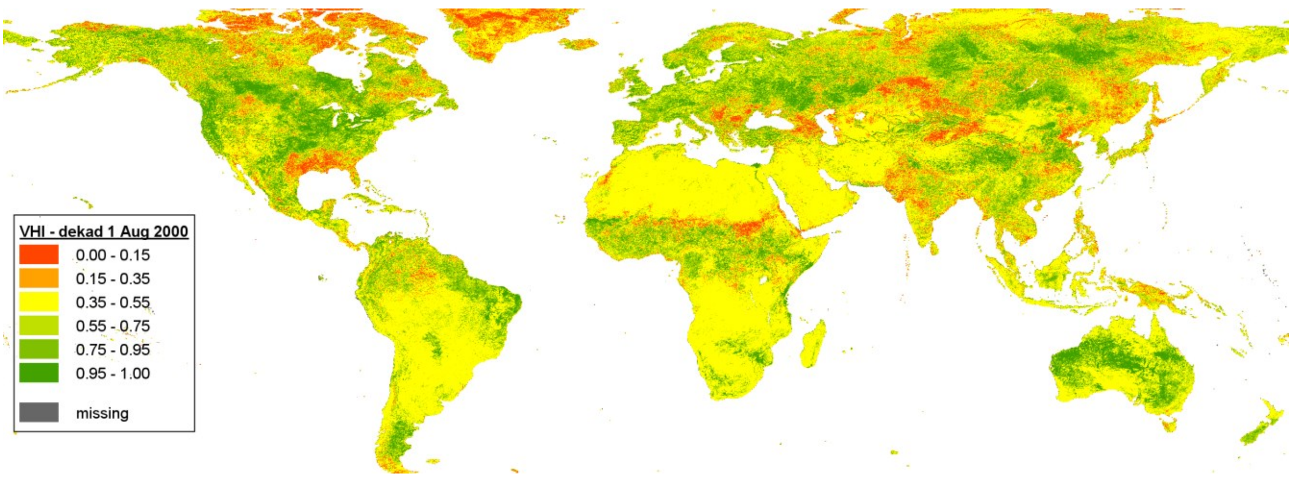


input VCI IMG



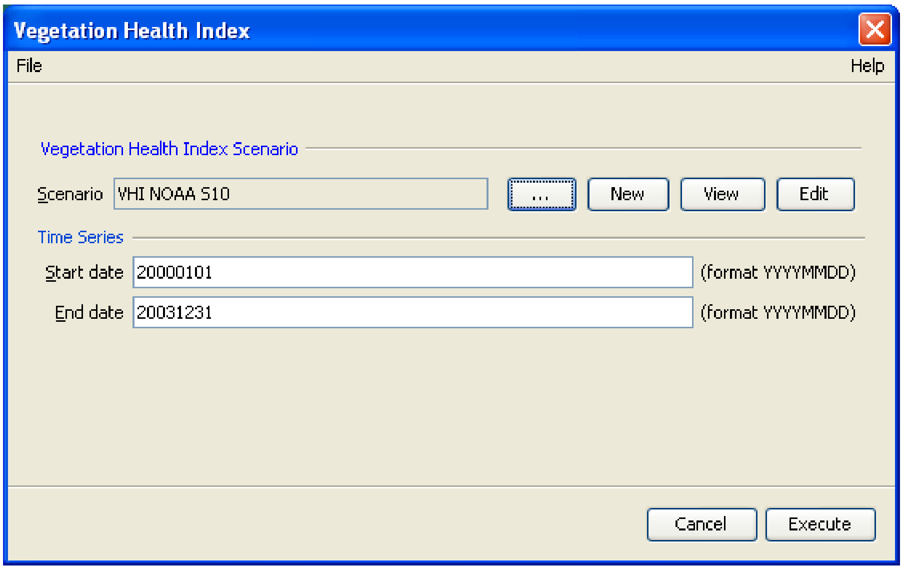
input TCI IMG

Vegetation Health Index Tool example



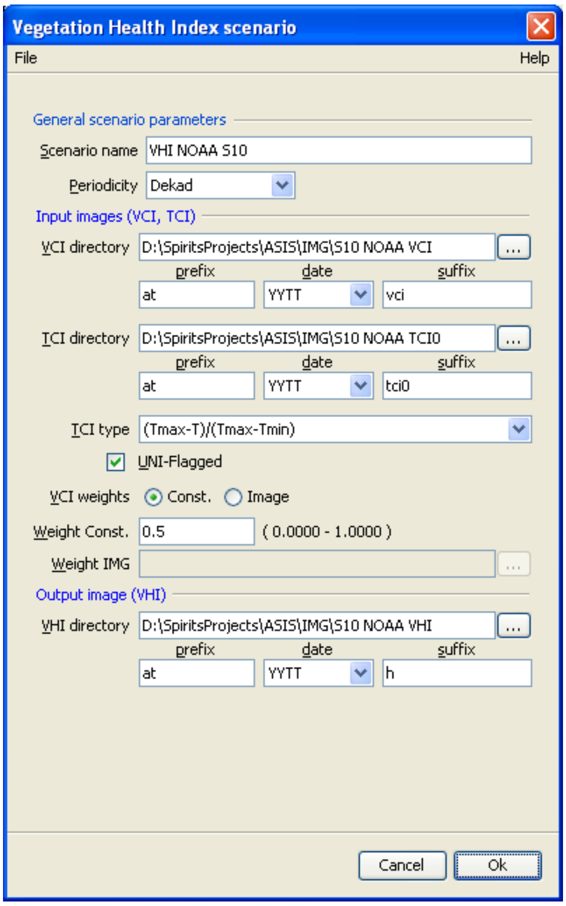
output VHI IMG

Time Series



Vegetation Health Index Time Series example

Scenario



Vegetation Health Index Scenario example

3.11. Mask

Goal

Flag an IMG with a mask IMG. All pixels with mask values in a rage to be specified, are flagged with an output flag value to be specified.

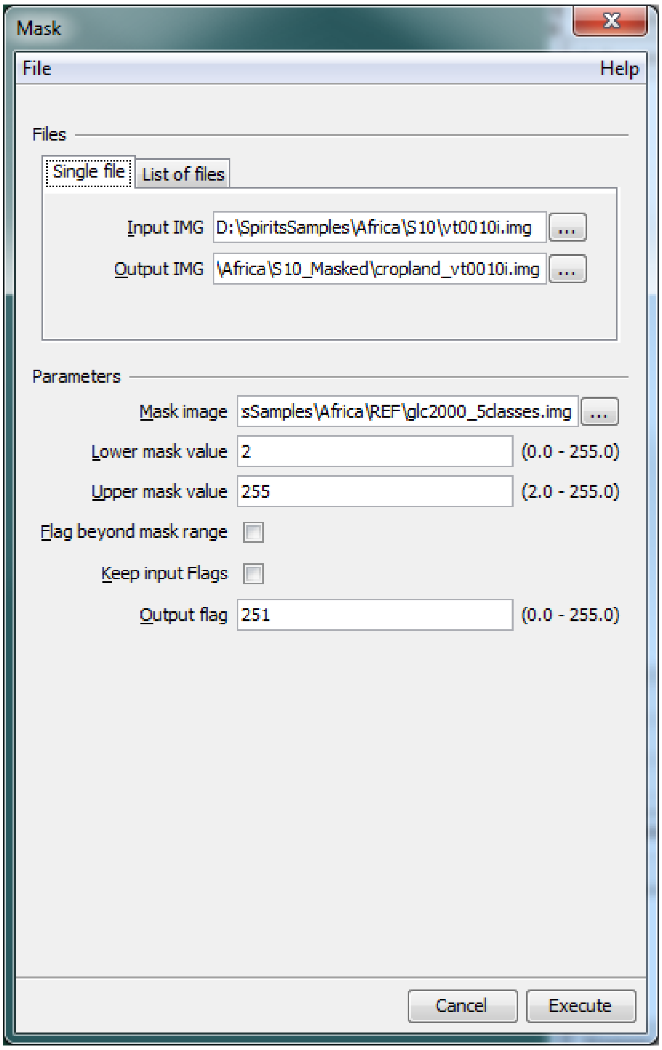
Parameters

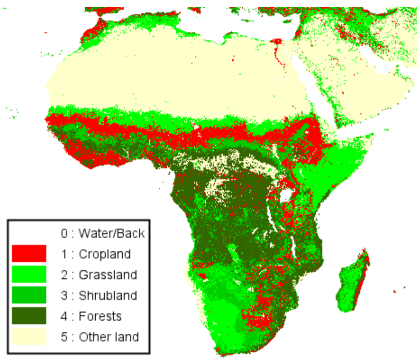
- the mask IMG.
- the lower mask value. (Must be in the range of the mask IMG data type).
- the upper mask value. (Must be in the range of the mask IMG data type).
- whether to flag pixels with mask values in the specified range, or beyond the specified range.
- whether to keep the input IMG flags or treat (flag) them as 'normal' pixels in case they are masked.
- the output flag. (Must be in the range of the input IMG data type).

The input IMG is copied to the output IMG, but the pixels for which the mask IMG has values between (or beyond) the lower and upper values specified, will be flagged with the output flag value specified.

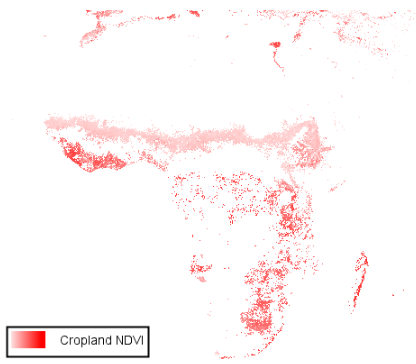
Tool

input IMG





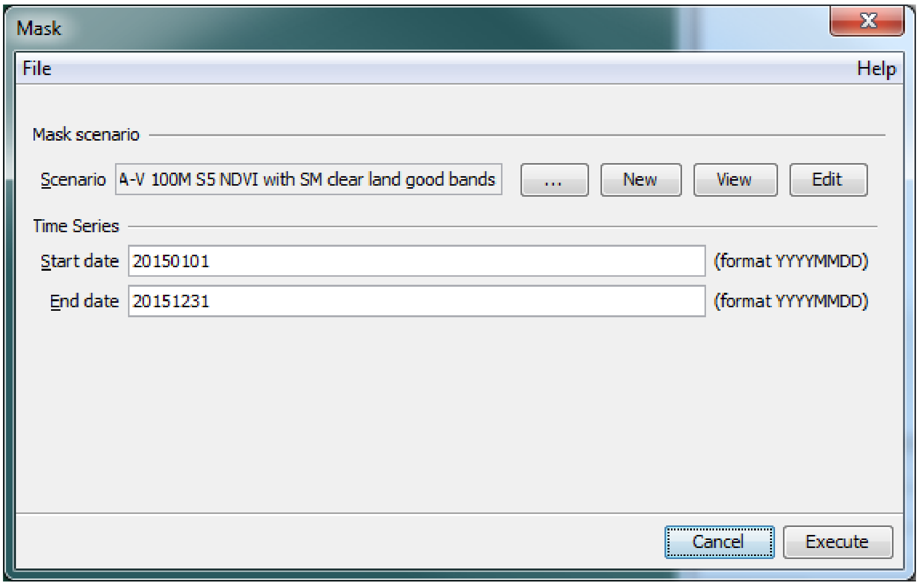
mask IMG



Mask Tool example

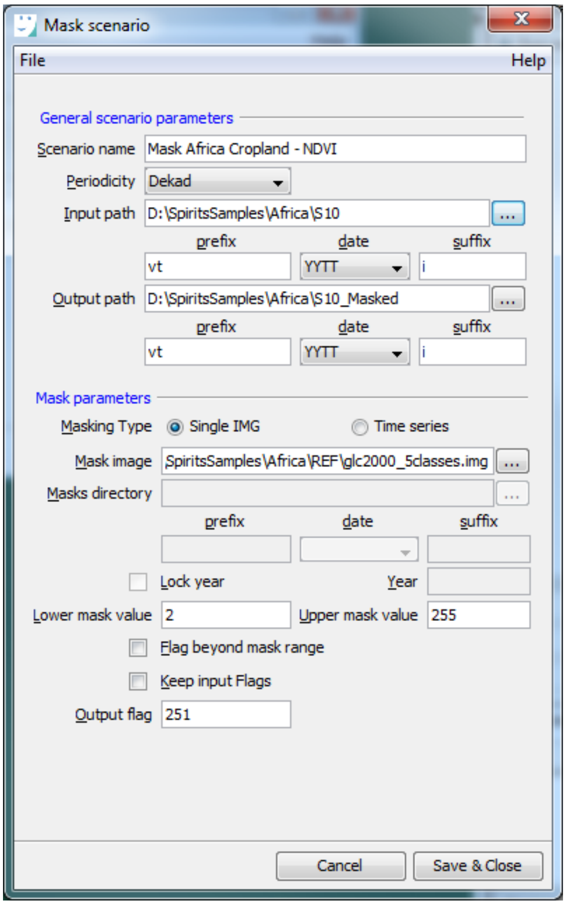
output IMG

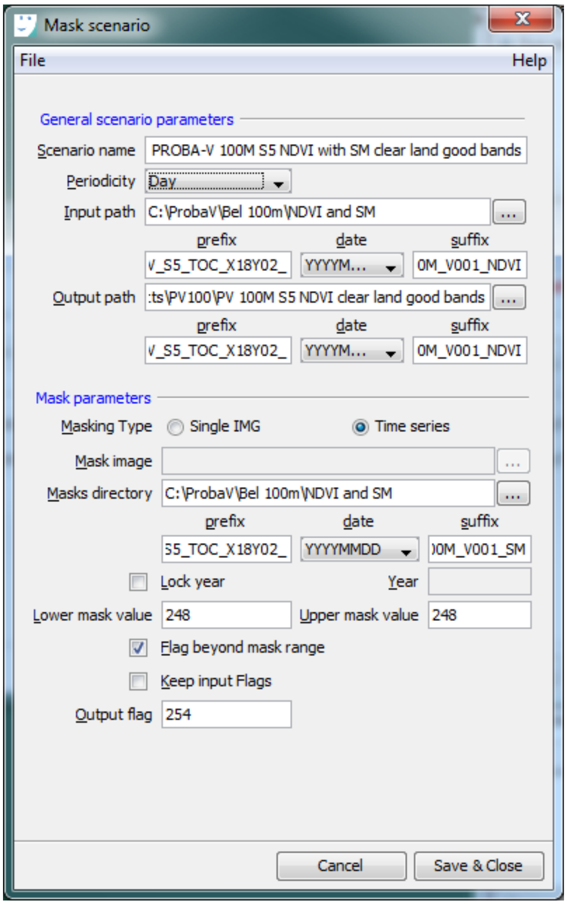
Time Series



Mask Time Series example

Scenario



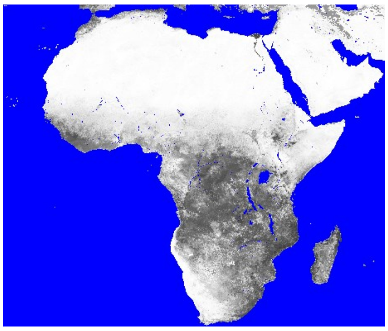


Mask Scenario example - static mask Mask Scenario example - dynamic mask

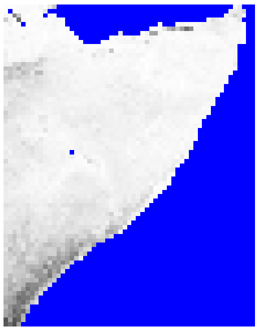
Remark: The Mask Time Series allows a constant mask IMG or a time series of mask IMGs. The lock year option allows specification of a constant YYYY value for the mask IMGs. This enables the use of some reference year, e.g. historical IMGs computed by the Long Term Average tool.

Masking an administrative region example

example: limit an IMG to the pixels belonging to an administrative region.



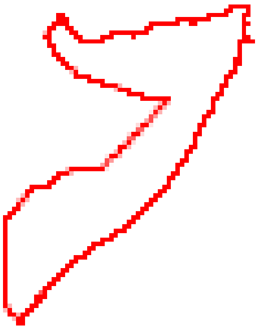
=>



fAPAR IMG

extracted ROI

Extract Band/ROI tool: extract Somalia ROI

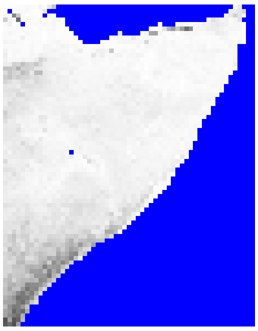


=>

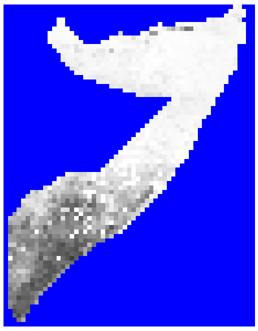
administrative region Somalia SHP

rasterized SHP

Rasterize SHP tool: raster Somalia administrative region SHP to mask IMG



=>



input IMG

mask IMG

output IMG

Mask tool: mask out non-Somalia pixels

3.12. Histogram

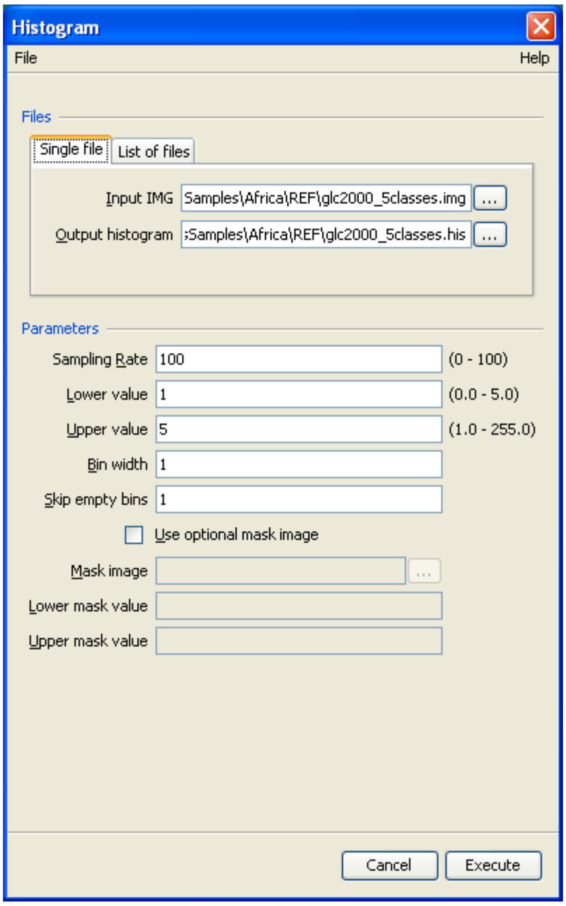
Goal

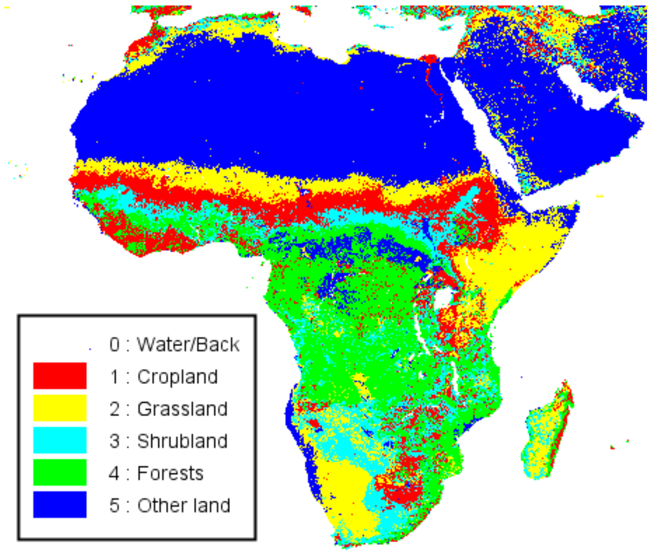
Compute ASCII file (*.HIS) with the histogram (and additional info) of an IMG.

Parameters

- the image sampling rate (percentage of the pixels of the input IMG to be sampled: 1%-100%);
- the image values range to be included (digital value);
- the histogram bin width;
- the threshold to skip empty bins (1 = skip all empty bins, 0 = keep all bins, including empty ones);
- an optional mask IMG and mask range: Pixels with mask values beyond this range are excluded.

Tool





Histogram Tool example input IMG

BIN	V	Y	Npix	Npix%	CUM%
1	1	Cropland	11090	12.86438	12.86438
2	2	Grassland	13927	16.15530	29.01969
3	3	Shrubland	9752	11.31231	40.33199
4	4	Forests	15428	17.89646	58.22845
5	5	Other land	36010	41.77155	100.00000

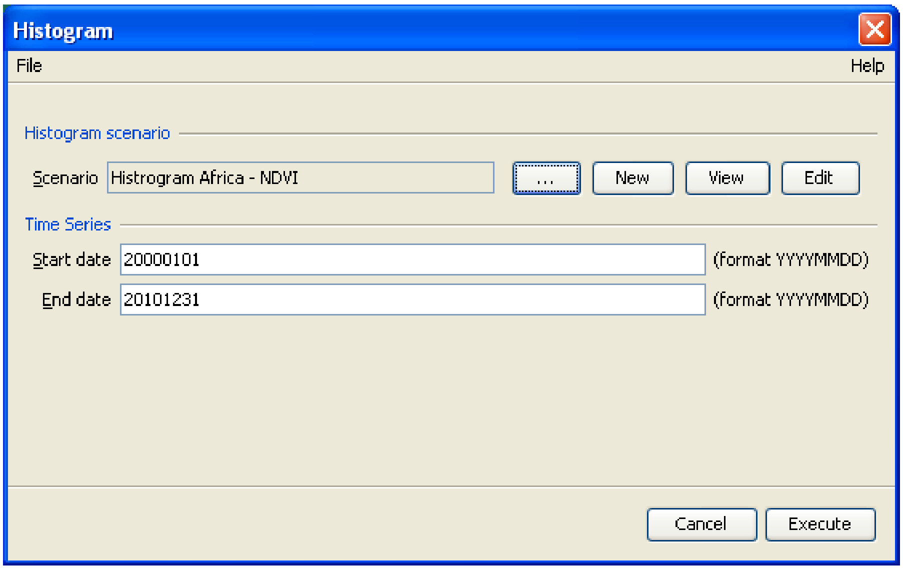
extract Histogram Tool result

Complete HIS file

```
***************
IMAGE HISTOGRAM - Program HISTO.EXE (V1011/1009)
***************
NB: V = Digital Numbers in IN-IMG (potential range defined by Data Type)
    Y = Physical units (eg. NDVI [-], DMP [kgDM/ha/day], classes [-], ....)
    B = Bins of Digital Nrs., used for Histogram (defined by Bin Width)

Output File          : D:\SpiritsSamples\Africa\REF\glc2000_5classes.his
Input Image          : d:\spiritssamples\africa\ref\glc2000_5classes.img
Data Type            : 1
Physical Y-Units     : GLC2000-derived-classes [-]
Classes              : 6
Phys. Scaling V->Y   : Y = 0 + 1 * V   (Only valid over range Vlo ... Vhi)
Flags                : 0=water/background
Significant Vlo/hi   : 1  ...  5
V-range in HISTO     : 1  ...  5

Width of V-bins      : 1
Number of V-Bins     : 5
Bin Scaling V->B     : B = floor(-0.5 + 1 * V)    [0  ...  5]

Background Mask      : none

 Pixels              :          Npix        Npix%
-Total Image         :        179400    100.00000
-Sampled for HIS     :        179400    100.00000
-Observed in HIS     :         86207     48.05295   (48.05295% of Sample)

Number of Modes      : 1
Modal Frequency      : 36010  (41.772% of total observations)
1st Modal V-Value    : 5
Mean V-Value         : 3.59555
St. Deviation of V   : 1.47353
Observed V-range     : 1  ...  5
Idem from Bins       : 0.5  ...  5.5

 NOTES:
-Statistics (Mode/Mean/Min/Max/St.Dev) derived from V-data over Observed Range
 Beware: If this exceeds the significant range Vlo-Vhi,
         it also includes Flags, and Statistics might be wrong!
-The Modes are Derived from Binned V-Data, and hence might be slightly biased
-In the table below, V represents the lowest V-Value of each Bin
                    Y the corresponding physical value or class
-The table only covers the observed V-range (1 - 5)
-BIN counting starts at 1 (not at 0).
 Series of 1 or more subsequent zero-bins are skipped

----------------------------------------------------------------------------------
     BIN            V                           Y         Npix       Npix%       CUM%
----------------------------------------------------------------------------------
      1             1                    Cropland        11090    12.86438    12.86438
      2             2                   Grassland        13927    16.15530    29.01969
      3             3                   Shrubland         9752    11.31231    40.33199
      4             4                     Forests        15428    17.89646    58.22845
      5             5                  Other land        36010    41.77155   100.00000
```

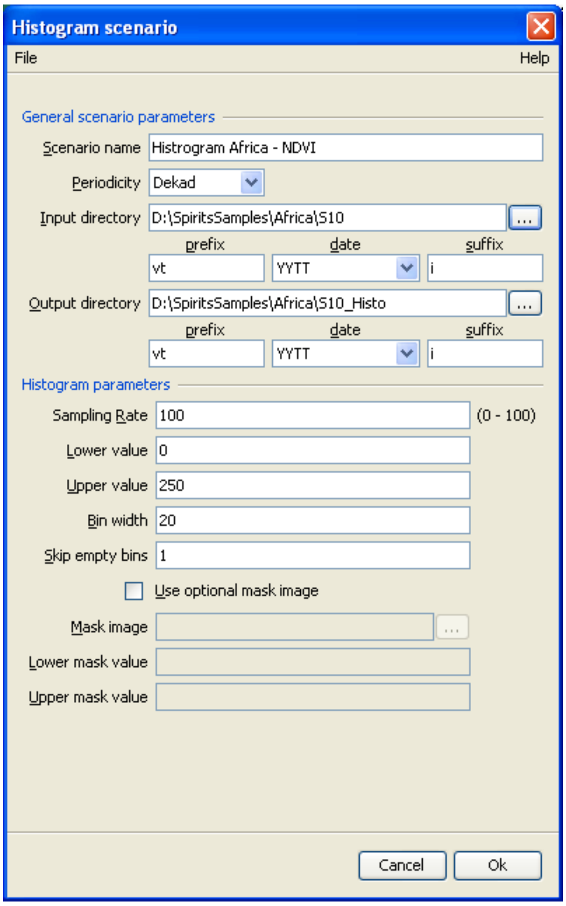
output HIS

Time Series



Histogram Time Series example

Scenario



Histogram Scenario example

3.13. Frequency

Goal

Count the frequency of an event in a series of IMGs.

Parameters

- the VAR file specifying the series of IMGs to inspect;

- the upper and lower limits of the range to consider as event criterium in the counting process;
- the specification of these upper and lower values:
  - to be considered digital values, being the raw digital IMG values;
  - to be considered physical values, according to ***Physical value = Vint + Vslo * Digital IMG value*** with Vint and Vslo the intercept and slope specified in the values entry in the IMG HDRs.
- whether to consider as event IMG values ***in*** the specified range or those ***beyond*** the specified range;

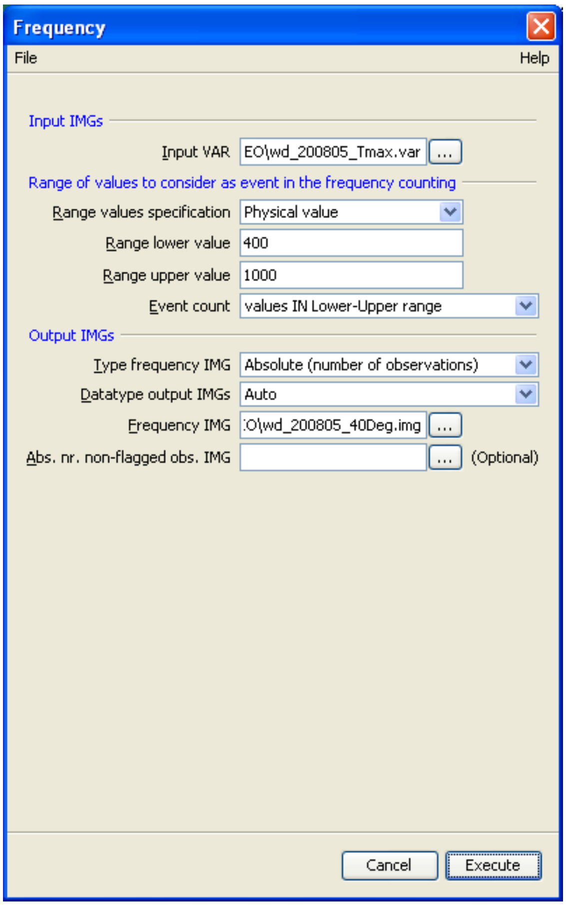
- the contents type of the output IMG containing the events frequency:
  - absolute frequency (the number of observations);
  - relative frequency (%) - relative to the number of non-flagged observations;
  - relative frequency(%) - relative to the total number of observations.
- the data type of the output IMGs. Automatic, Byte or Integer. In case Automatic is selected, the program will search for the best solution (Byte or Integer);
- the output IMG containing the events frequency;
- the output IMG containing the (absolute) frequency of non-flagged observations (optional).

Remarks:

- Absolute frequencies will be chopped at 254 for Byte output IMGs, at 32767 for Integer output IMGs;
- Relative frequencies will be scaled between 0 and 200 for Byte output IMGs, between 0 and 10000 for Integer output IMGs;

- Pixels flagged in all input IMGs will be flagged in the output IMGs:
  - the output IMG containing the events frequency will use flag value 255 for Byte output IMGs, -1 for Integer output IMGs;
  - the (optional) output IMG containing the frequency of non-flagged observations will always use flag value 0;

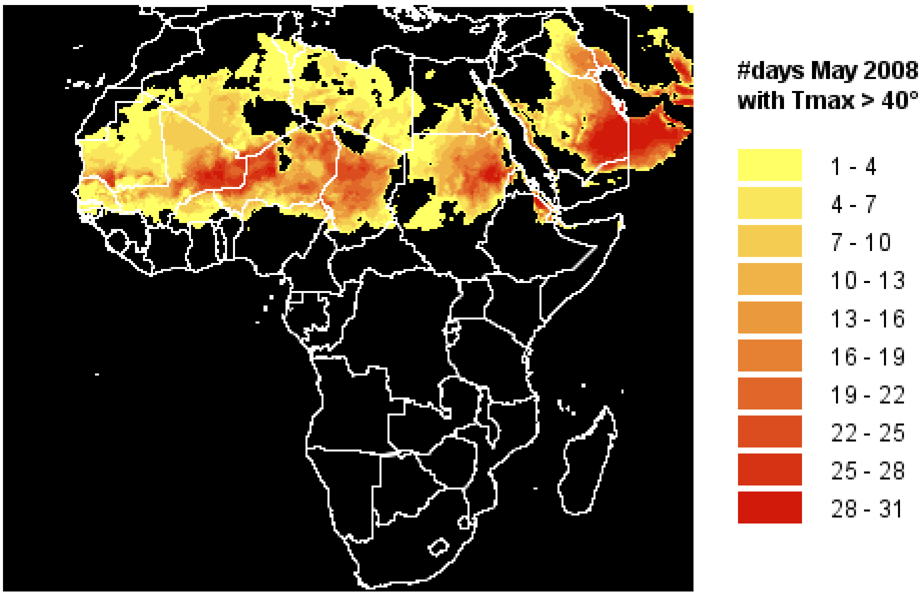
Tool



Frequency Tool example

Count days with max temperature above some limit example

example: using a VAR file containing the daily maximum temperature IMGs in a month, the number of days for which the temperature exceeds some threshold in that month can be counted.



3.14. ProdMax

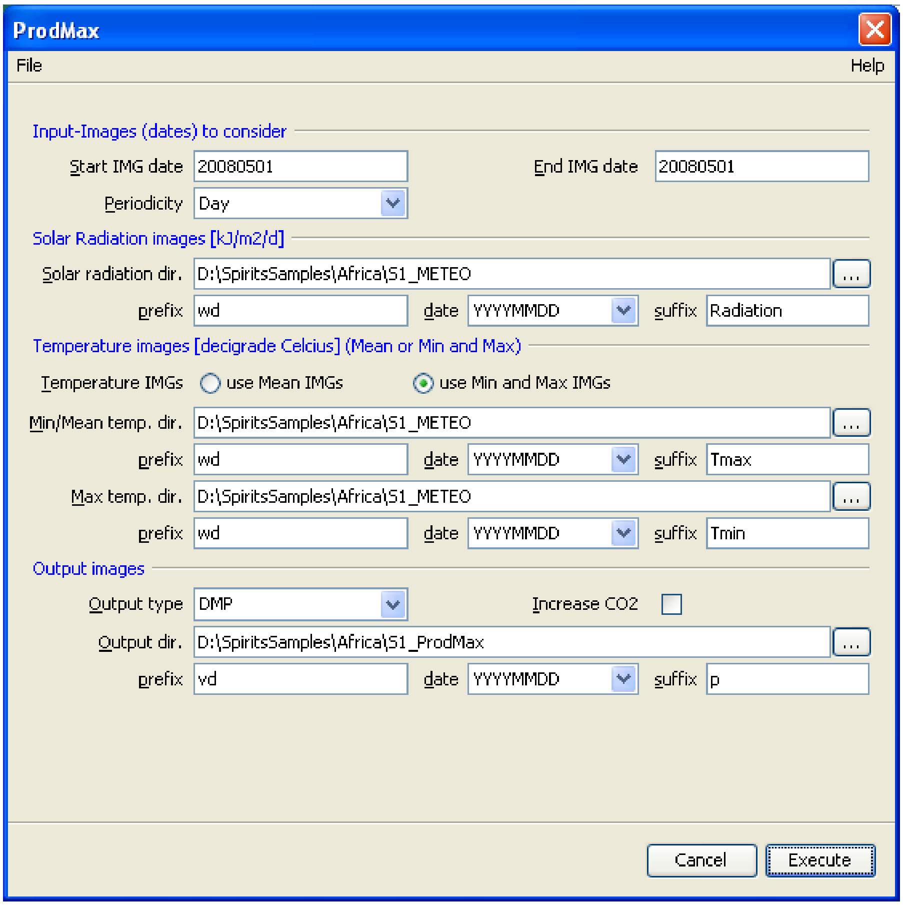
Goal

Compute potential maximum DMP/NPP (Dry Matter Productivity/Net Primary Productivity: Remote Sensing vegetation-indicators derived with the Monteith-approach) IMGs from meteo data. All IN-IMGs must have data type 16-bit integer, scaling must be Vint=0/Vslo=1, Radiation must be in kJ/m2/day, temperatures in decigrade Celsius.

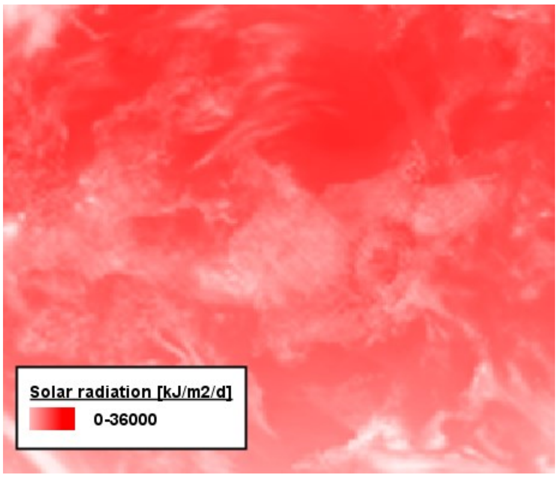
Parameters

- the first date of the input IMGs to consider;
- the last date of the input IMGs to consider;
- the periodicity of the input IMGs;
- the location and filename structure of the input IMGs containing the solar radiation (in kJ/m2/d);
- the location and filename structure of the input IMGs containing the temperature (in decigrade (one tenth of a grade) Celsius). There are two possibilities: using the mean temperatures or using the minimum and maximum temperatures;
- the content of the output IMGs: DMP or NPP;
- the location and filename structure of the output IMGs.

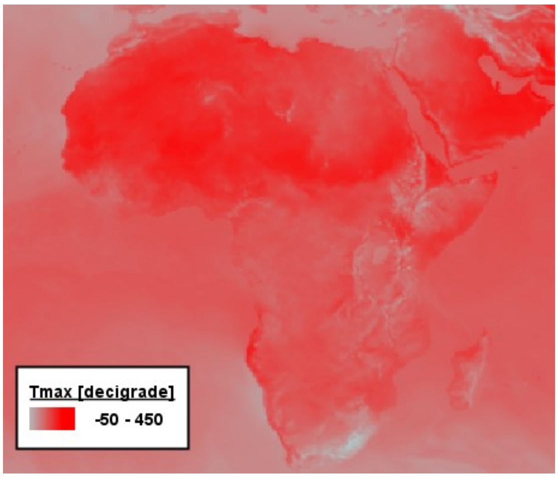
Tool



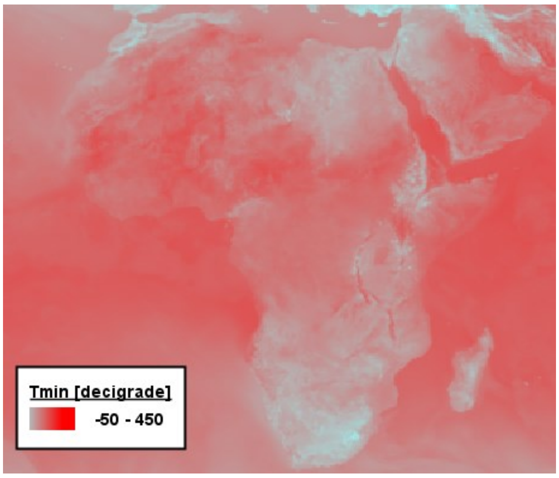
ProdMax Tool example



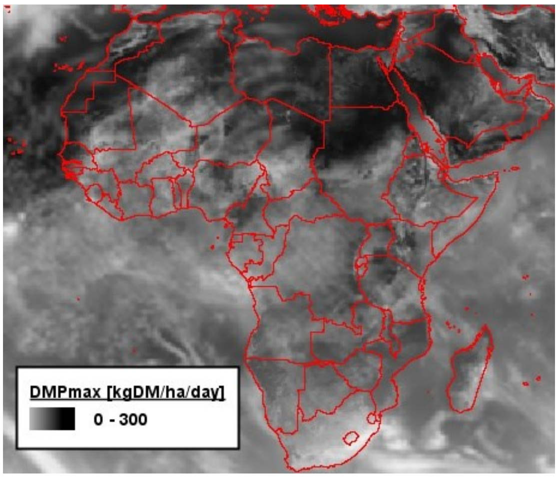
input Solar radiation IMG



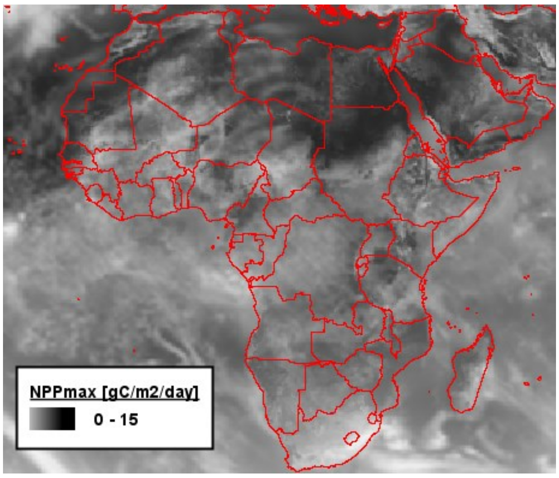
input Maximum temperature IMG



input Minimum temperature IMG



output Max DMP IMG



output Max NPP IMG

ProdMax Tool example

3.15. Prod

Goal

Compute DMP/NPP (Dry Matter Productivity/Net Primary Productivity) IMGs using the Monteith approach.

Parameters

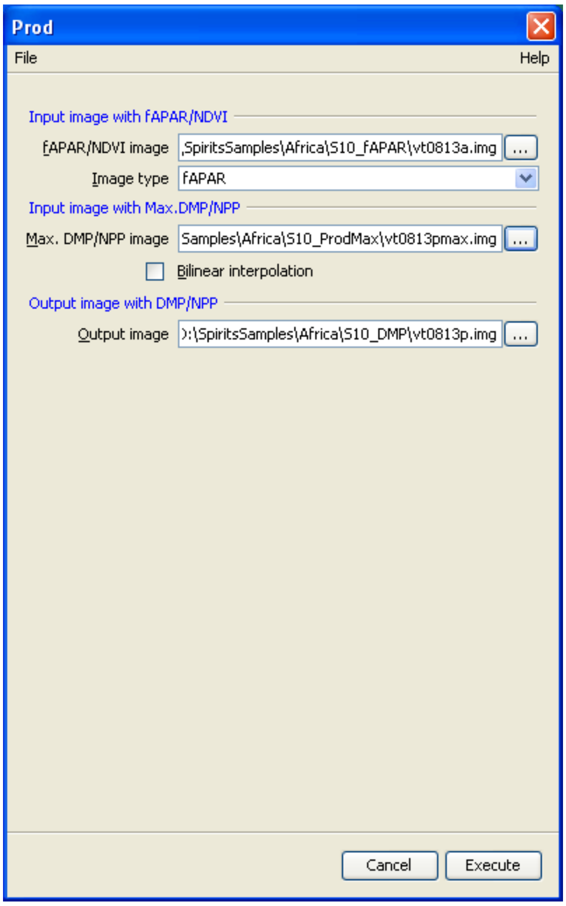
- the input IMG needed to define fAPAR;
- the actual contents of this IMG. This can be:
  - fAPAR;
  - AVHRR - NDVI (NOAA sensor);
  - VGT-NDVI (SPOT sensor).

In case AVHRR - NDVI is selected, fAPAR will be calculated as fAPAR = -0.269 + 1.68.NDVI.

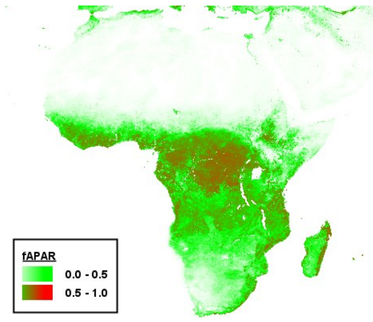
In case VGT-NDVI is selected, fAPAR will be calculated as fAPAR = -0.247 + 1.54.NDVI.

- the IMG containing the potential maximum DMP/NPP;
- the output IMG.

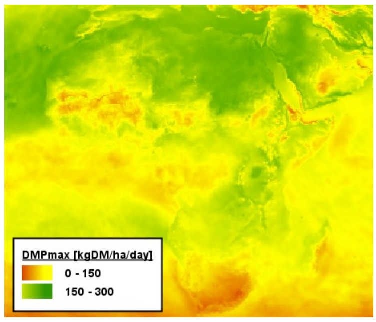
Tool



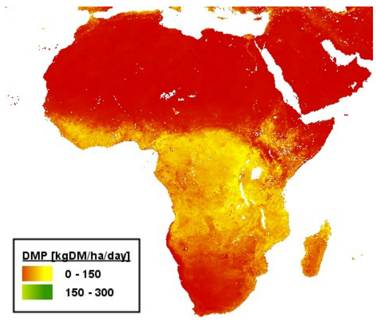
Prod Tool example



input IMG: fAPAR

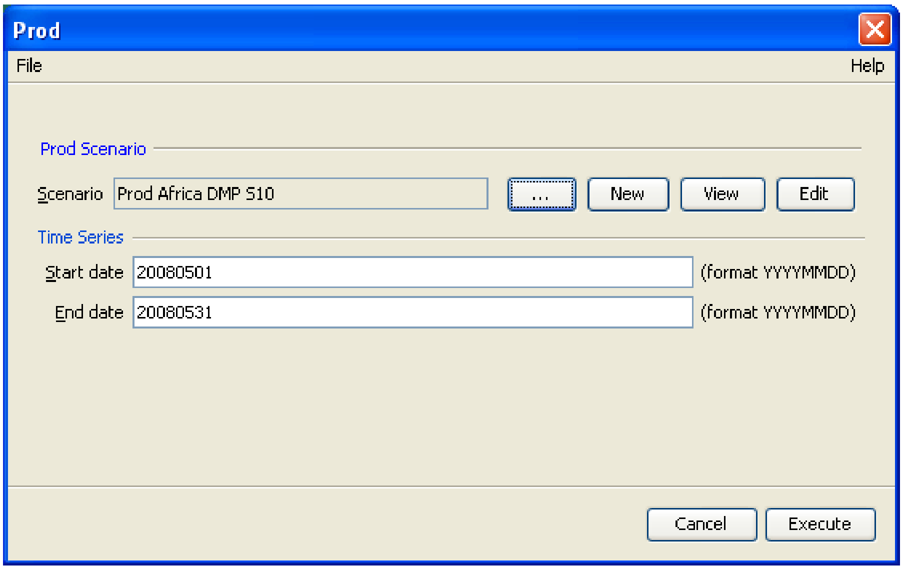


input IMG: max DMP



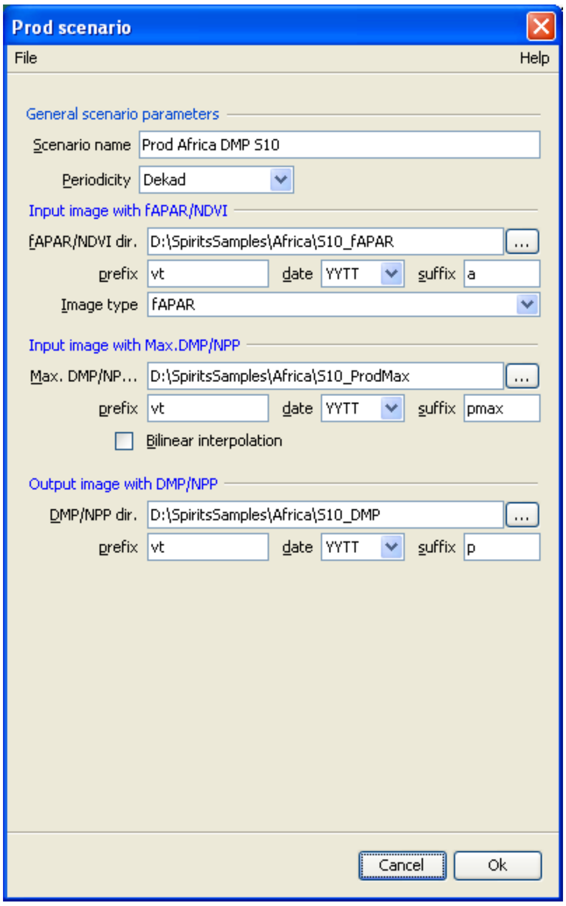
output IMG: DMP

Time Series



Prod Time Series example

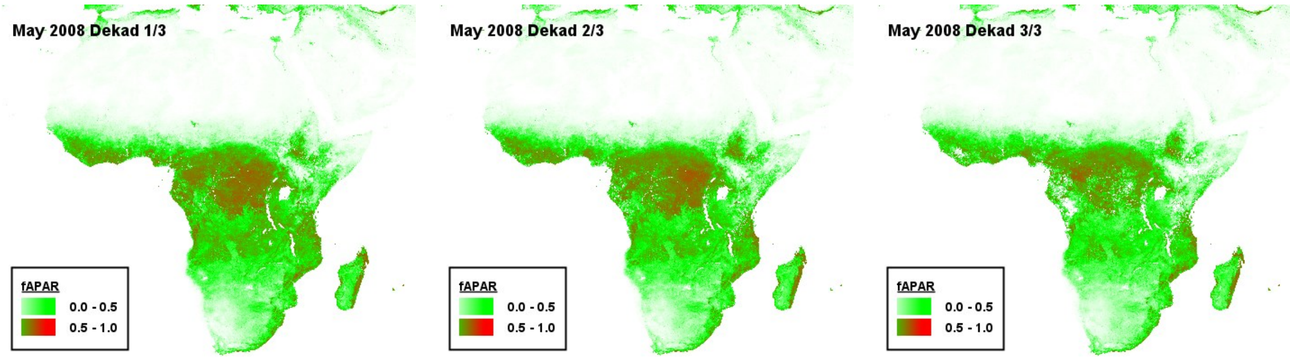
Scenario



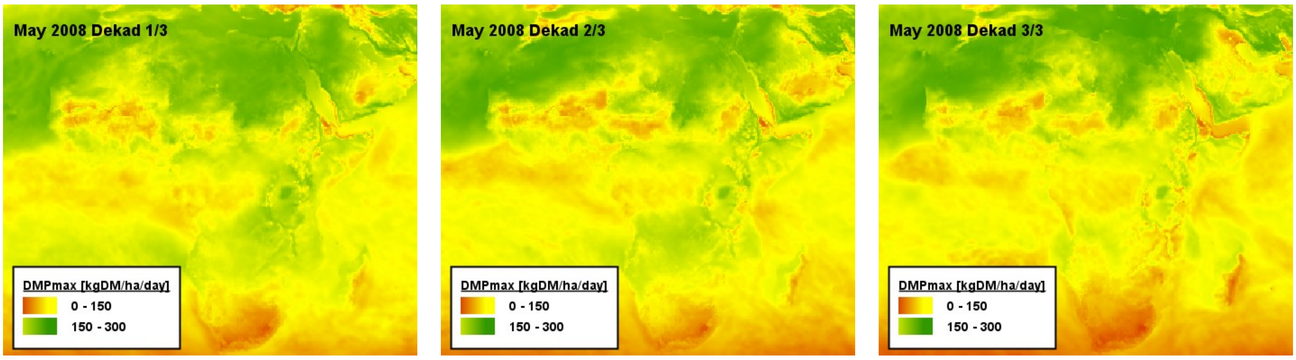
Prod Scenario example

Prod Time Series example

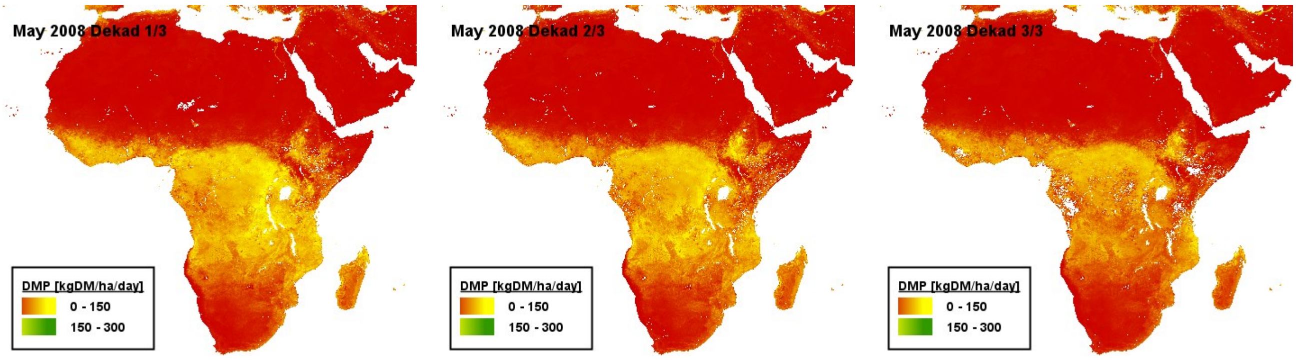
example: calculation of dekadel DMP's in May 2008.



input IMGs: fAPAR May 2008, Dekads 1, 2 and 3



input IMGs: max DMP (from ProdMax tool) May 2008, Dekads 1, 2 and 3



output IMGs: DMP May 2008, Dekads 1, 2 and 3

3.16. Composit

Goal

Calculate spatio temporal composites.

Based on all "registrations" found in a specified period, containing at least a partial coverage of the specified ROI, a single output registration is created: the "composite".

Remark: Composit is a more complex variant of Mosaic:

- it can be used in different modes:
  - Pure spatial compositing (as Mosaic; e.g. stitching of tiles);
  - Pure temporal compositing (e.g. calculate monthly composites from dekadel images);
  - Spatio-temporal compositing.
- it offers additional possibilities (e.g. use of a bitmask (status mask) and a threshold layer as elimination criteria and selection/creation of multiple IN/OUT slave layers);
- it requires a more complex configuration than Mosaic. Especially it discerns “registrations” from “layers”. “Layers” are image variables such as NDVI, reflectances, status mask information, ... Each “Registration” is a set of “layer” images, belonging to a same registration date (or period) and covering the same spatial zone. Different registrations may cover different areas (Africa, Asia,...), but they always comprise the same set of layers.

Parameters

The tool needs two groups of parameters:

- general parameters, (e.g. in- and output data location, ...);
- specifications (e.g. the criteria and the layers to be used).

general parameters

- the location of the input IMGs;
- the location of the output IMGs (composites);
- the output IMGs periodicity (or compositing period);
- the start date (for a single composite) or the start dates for the first and last composite in a series .

specifications

general output specifications:

- “OUTtiltle”: an (optional) description to be used as value for the description item in the HDR files of the generated IMGs. If left blank, the value for the description item from the “OUThdr” is used;
- “OUThdr”: the HDR-file with full geo-referencing (samples, lines, map info, ...) of the target zone to be covered by the composit;
- “OUTmask”: (optional) Land/Sea mask IMG. Must be BYTE (0=sea, 1-255=land) and spatially congruent with “OUThdr”. If not specified, all pixels are treated and the composites contain one single flag. If OUTmask is specified, the sea pixels are skipped and there are two flags (sea, missing over land).

input IMGs specifications:

The input IMGs filenames need to follow a file naming convention:

- filenames must be formatted as either ***pdmse***.img/hdr or ***psmde***.img/hdr;
- ***p***, ***m***, **e**: the prefix, middle and end parts of the filenames. Each of these is optional. (In the tool, these parts will be specified by combinations of constant characters and "?" wildcards);
- ***d***: the date part of the filenames, discerning the different registration dates;
- ***s***: the suffix part of the filenames, discerning the different layers;

In the tool they will be specified via the “INpattern”, “INdateFMT” and “INsuf” parameters.

- “INpattern”: input IMGs filename pattern:
  - the pattern is a combination of (optional) constant characters, (optional) “?” wildcards (any character) and the (mandatory) “%D” and “%S” placeholders;
  - the placeholders “%D” and “%S” must be literally present;
  - %D is the “date field” placeholder, and will be replaced by the dates, formatted according to the “INdateFMT” parameter;
  - %S is the file name suffix placeholder, and will be replaced by the suffixes specifying the input layers according to the “INsuf” parameter;
- “INdateFMT”: the date format of the input IMGs filenames (YYYYMMDD, YYMMDD, ...YYTT,...). The dates, determined by the start dates and periodicity, specified in the general parameters, and formatted according to this selected date format, will substitute the “%D” placeholder in the input IMGs filenames pattern;
- “INsuf”: a list of suffixes for the filenames for each of the input layers. Specified as comma separated list of suffixes. These suffixes distinguish the different input layers, their values will substitute the “%S” placeholder in the input IMGs filename pattern;
- “INsevere”: whether or not to allow variations of the spectral coherence of the input IMGs of the same layer. For all input IMGs of the same layer, the datatype, the scaling (Vint/Vslo) and Classes (if applicable) must be identical. They must also have identical spatial resolution, without sub-pixel shifts. Variations in spectral features Vlo/Vhi, Vname/Vunit and Flags can be allowed or forbidden;

selection:

- “selINsuf”: the suffix of the (first) selection IMG. This suffix must correspond with one of the entries in the “INsuf” list;
- “SELrule”: the selection criterium for this (first) selection: Mean, Minimum or Maximum;

two-strata selection

In case Min/Max compositing criteria are used, a second selection layer, with its own criterium (Min/Max) can be specified. An additional IMG is needed which selects per-pixel between both selection layers/criteria.

- “SELMask”: stratification IMG to discern the two “strata”:
  - the IMG must be BYTE and spatially congruent with the target zone specified in “OUThdr”.
  - all pixels with SELmask=0 belong to the first stratum, the others (values > 0) the second stratum.
  - typically one would use different selections for land and sea pixels, in that case the “OUTmask” IMG could be used as “SELMask” IMG too.

- “selINsuf2”: the suffix of the (second) selection IMG. This suffix must correspond with one of the entries in the “INsuf” list;
- “SELrule2”: the selection criterium for the (second) selection: Minimum or Maximum;

Remark: typical application would be to apply MAX-NDVI compositing over land, MIN-NIR over sea.

output IMGs specifications:

The output IMGs filenames need to follow a file naming convention:

- filenames will be formatted as ***PoDoSo***.img/hdr;
- ***Po***: the output IMGs prefix;
- ***Do***: the date part of the filenames, discerning the different composite dates;
- ***So***: the suffix part of the filenames, discerning the different composited layers;

In the tool they will be specified via the “OUTprefix”, “OUTdateFMT”, “OUTsuf”, “OUTsufNG” and “OUTsufID” parameters.

- “OUTprefix”: the prefix of the output IMGs file names;
- “OUTdateFMT”: the date format of the output IMGs filenames (YYYYMMDD, YYMMDD, ...YYTT,...). The selected format must be compatible with the periodicity, specified in the general parameters;
- “OUTsuf”: a list of suffixes for the filenames for each of required output layers. Specified as comma separated list of suffixes. This list must match the “INsuf” list: for each entry in the “INsuf” list (specifying an input layer) there **must** be an entry in the “OUTsuf” list. In case an output is required for the corresponding input layer, an actual suffix must be specified, in case no output is required for the corresponding input layer, the entry must be left blank. Practically this means that the “OUTsuf” and “INsuf” lists must have same number of comma's.
- “OUTsufNG”: (optional) suffix for additional output IMG containing the number of "good" values, available for the composite;
- “OUTsufID”: (optional and only in case of Min/Max criteria) suffix for additional output IMG containing the ID of the selected registration for the composite. This IMG is an ENVI-classification with the registrations dates as class names;

Remark: when using the **Mean** compositing criterium:

- there must be (only) one selection layer (often NDVI) ;
- optionally there can be bitmask and threshold rejection layers;
- hence there will minimum one and maximum three input layers (suffixes in “INsuf”);
- there must be (only) one output layer;
- hence “OUTsuf” will contain minimum one and maximum three entries (suffixes), matching the “INsuf” entries. In any case, exact one (of the) entrie(s) will not be blank;

example:

	date layer (suffix ‘NDVI’)	bitmask (suffix ‘SM’) layer specified		bitmask (suffix ‘SM’) and theshold (suffix ‘VZA’) layers specified		
INsuf	NDVI	NDVI, SM	SM, NDVI	NDVI, SM, VZA	SM, NDVI, VZA	SM, VZA, NDVI
OUTsuf	i_mean	i_mean**,**	**,**i_mean	i_mean**, ,**	**,**i_mean**,**	**, ,**i_mean

optional eliminations via a bitmask IMG:

Optional pixels from a registration can be rejected via a bitmask layer (typically status mask IMGs). The IMGs (one per registration) must be BYTE.

- "BITsuf": the suffix of the bitmask IMGs. Must correspond with one of the entries in the "INsuf" list;
- "BITval": test value. The concerned registration pixel is rejected if the logical AND operation between the BITval and BITMASK values > 0;
- "BITsevere": (optional and only in case of Min/Max criteria) Rejected observations can be allowed to appear in the composit, in case all other observations are invalid;

example: assume a status mask IMG with bit 0=1 for cloud shadow, bit 1=1 for clouds and bit 4=1 for snow, then (decimal) BITval = 19 (or binary '0010 011') will reject all these cases.

optional eliminations via a threshold IMG:

Optional pixels from a registration can be rejected via a threshold layer (typically View Zenith Angle IMGs, to discard observations with VZA beyond certain thresholds (say 30°)). The IMGs (one per registration) may be BYTE, SHORT, LONG or FLOAT.

- "THRsuf": the suffix of the threshold IMGs. Must correspond with one of the entries in the "INsuf" list;
- "THRlo" and "THRhi": threshold limits. The concerned registration pixel is rejected if its value in the threshold IMG, scaled to its physical units (Vint+Vslo*V) , is less than THRlo or greater than THRhi;
- "THRsevere": (optional and only in case of Min/Max criteria) Rejected observations can be allowed to appear in the composit, in case all other observations are invalid;

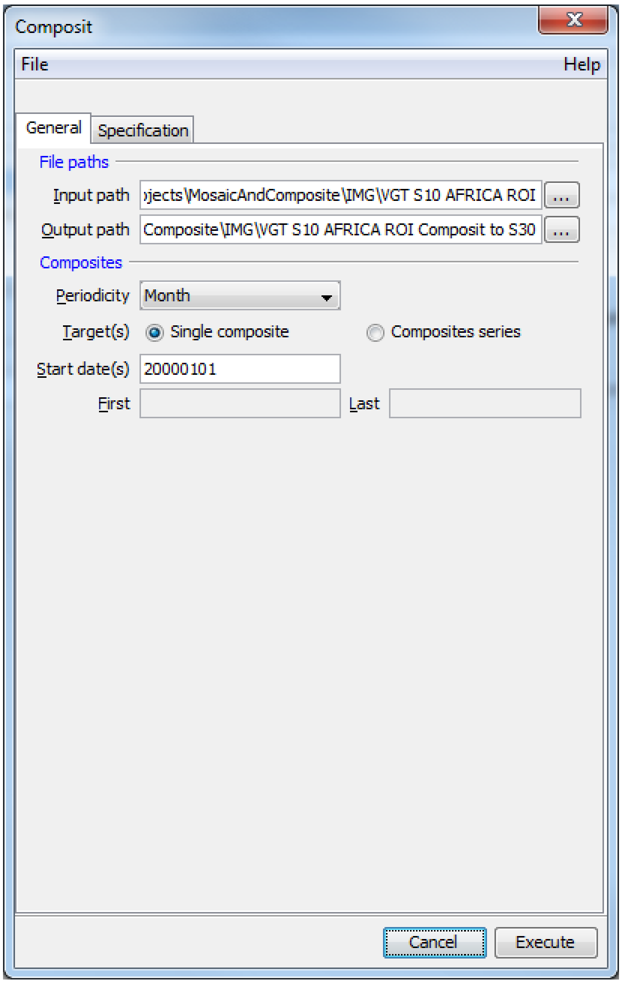
Remark: detailed features of the output IMGs:

- Basically, each output IMG inherits all spectral features of the corresponding input IMGs, belonging to the concerned "layer".
- Datatype, scaling (Vint/Vslo) and number of classes (mostly 0) are fixed. If the concerned layer is a classification (classes>0), the class names and colours are taken from the first input IMG.
- If variations occur (and are allowed) in significant range (Vlo/Vhi) and the name and units of the concerned image variable (Vname, Vunit):
  - For Vname, Vunit, the longest string is searched amongst all input IMGs (of the concerned layer) and used for the output IMG.
  - For Vlo the lowest value is retained, for Vhi the highest one.
- The output flags are fixed (per layer) in function of the datatype and whether a land/sea mask is used:
  - Without such mask the output IMGs will contain one single "missing" flag. If this flag overlaps with the significant range Vlo-Vhi, the latter is adapted. Examples: If Vhi=255 (BYTE) it is reset to 254, if Vlo=-32768 (SHORT) it is reset to -32767, etc. This measure may yield some saturation. For the BYTE example, it is not excluded that some output pixels initially have value 255 (a valid input value if Vhi=255). But in the created image, the program resets these values to Vhi=254 to avoid confusion with the flag 255.
  - If a land/sea mask is specified, the output IMGs will contain two flags ("sea" and "missing over land") and they will conform to the UNIflags system. Here too, the significant range of the generated output-IMGs is adapted if the flags make part of it.

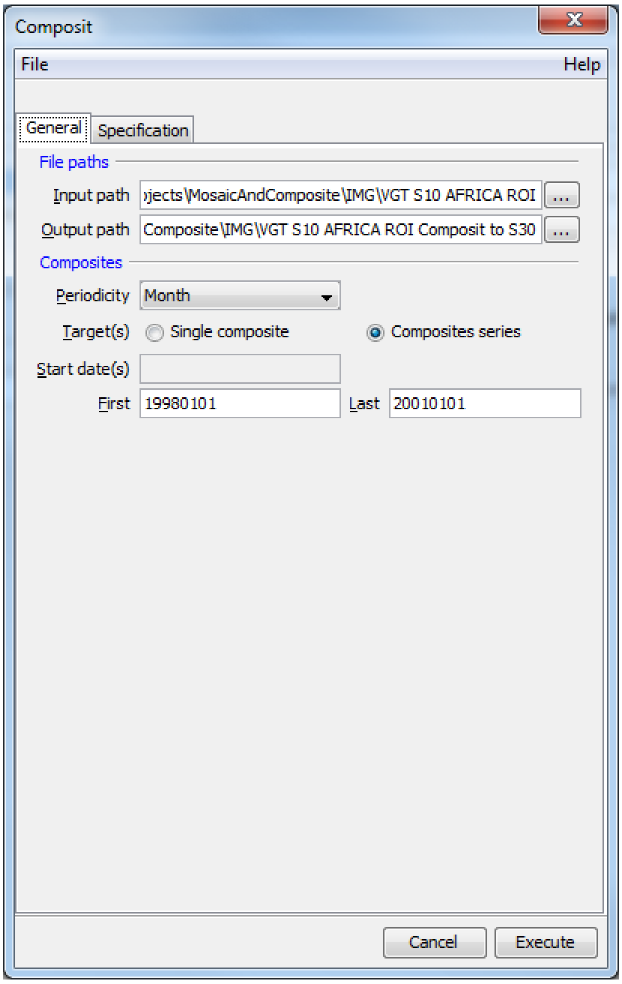
LAND/SEA MASK	FLAG	BYTE	SHORT	LONG & FLOAT
none	Missing	255	-32768	-1000005
	Possible adaptations	Vhi = 254	Vlo = -32767	Vlo = -1000000
specified	Sea	254	-2	-1000002
	Missing over Land	251	-5	-1000005
	Possible adaptations	Vhi = 250	Vlo = 0	Vlo = -1000000

Flags and possible Vlo/Vhi adaptions of the resulting composites

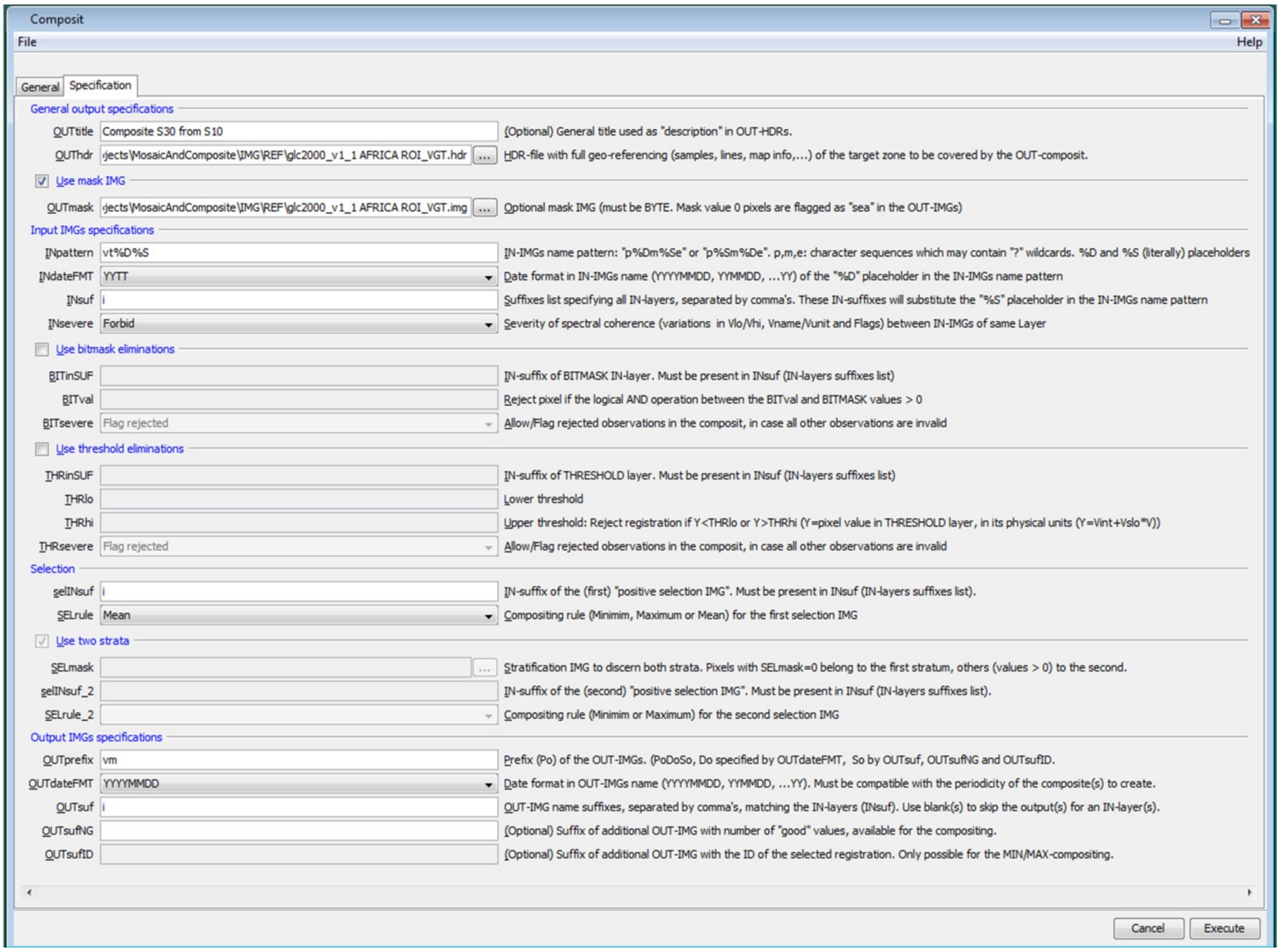
Tool



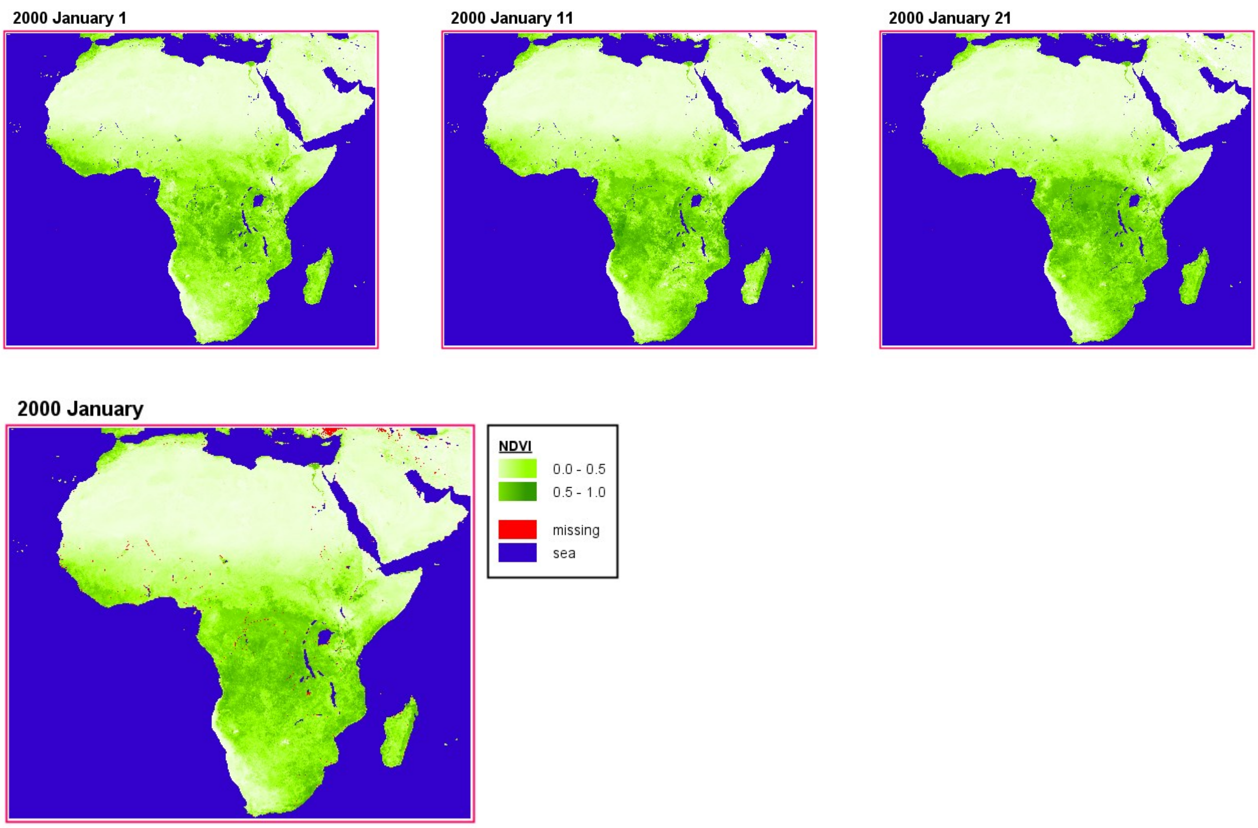
Composit Tool example:
general parameters
single composite



Composit Tool example:
general parameters
composites series



Composit Tool: specification example



Composit Tool example: composite dekadel to monthly NDVI.

Proba-V Morocco tiles spatio-temporal compositing example

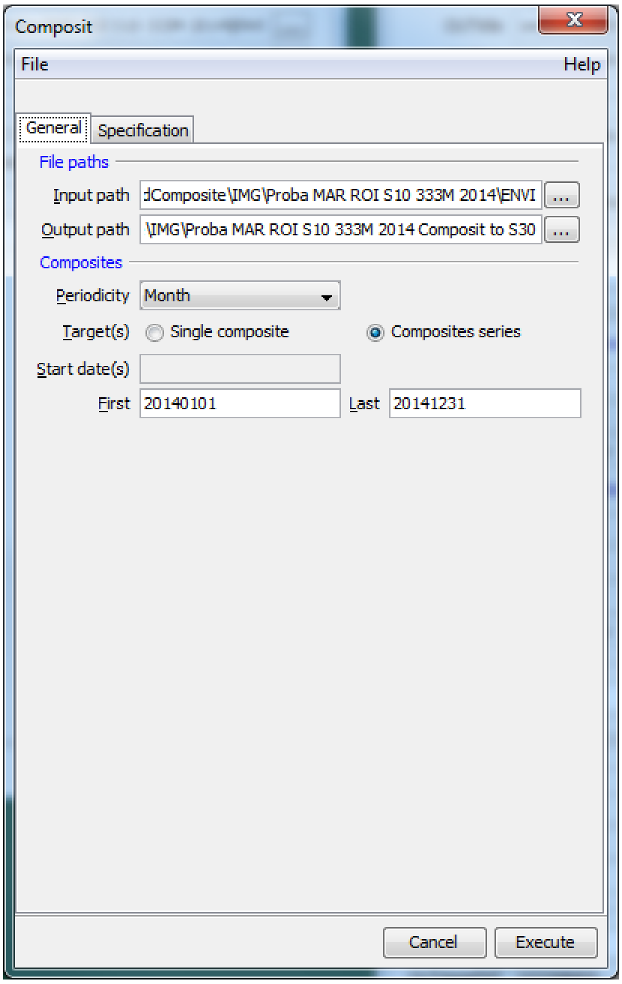
In this example, the input path contains for each dekad in 2014, the four Proba-V tiles (coded X16Y03, X17Y03, X16Y04 and X17Y04) covering the bounding box of Morocco. The filenames have been coded (during an HDF5 Import step) as ***pt_XxxYyy_YYYYMMDD_i***.img/hdr.

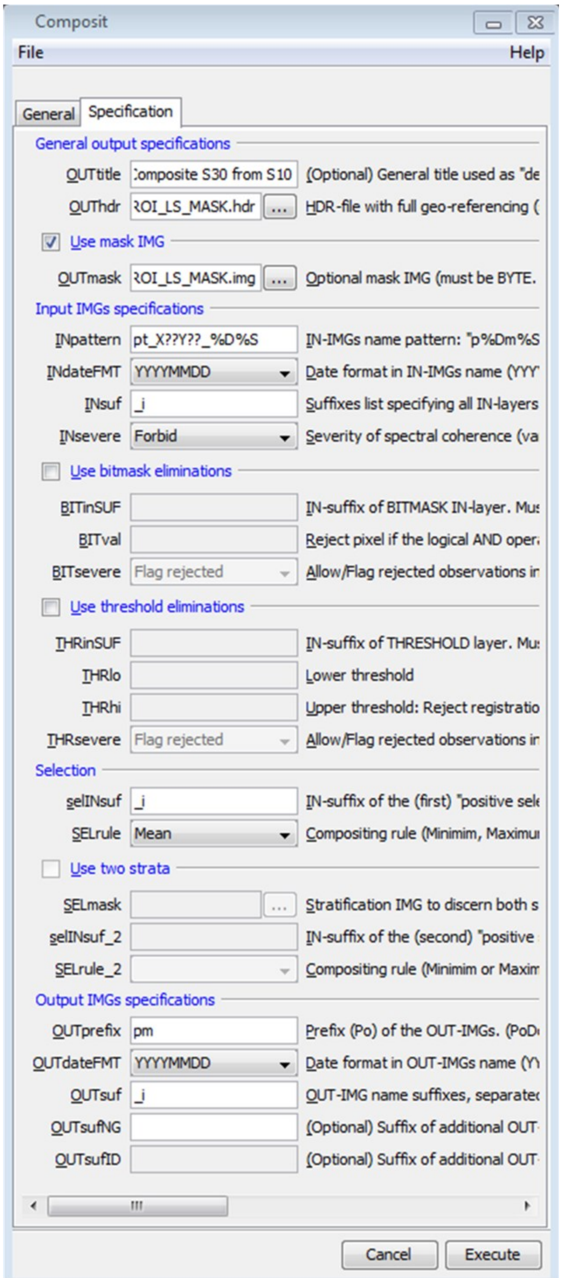
The series creates a loop over the periodic (monthly) start dates from the first till the last composite.

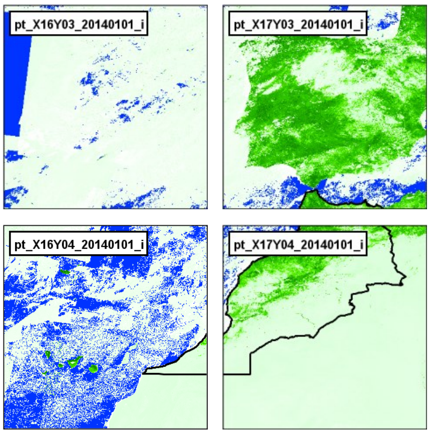
For each of these dates, a composite is created for the files matching the specified pattern, here ***pt_X??Y??_%D%S***, with %D formatted as ***YYYYMMDD*** and %S replaced by ***_i***. Via the ***?*** wildcards the different tiles (X16Y03, X17Y03, X16Y04 and X17Y04) will be selected.

- e.g. for date 2014 01 01, the files
  - pt_X16Y03_20140101_i, pt_X16Y04_20140101_i, pt_X17Y03_20140101_i, pt_X17Y04_20140101_i
  - pt_X16Y03_20140111_i, pt_X16Y04_20140111_i, pt_X17Y03_20140111_i, pt_X17Y04_20140111_i
  - pt_X16Y03_20140121_i, pt_X16Y04_20140121_i, pt_X17Y03_20140121_i, pt_X17Y04_20140121_i

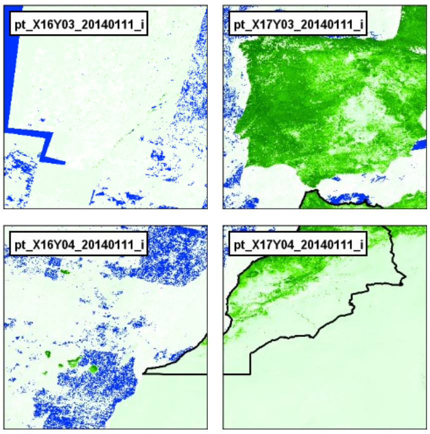
  will be found in the input path – and composited to pm20140101_i;



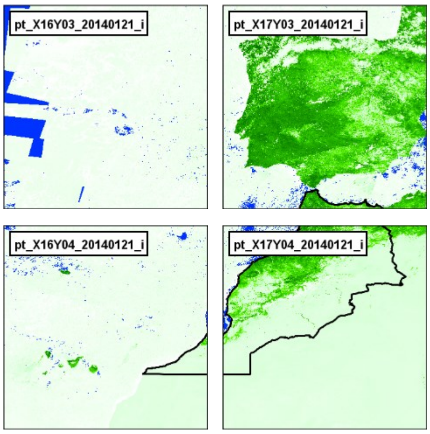




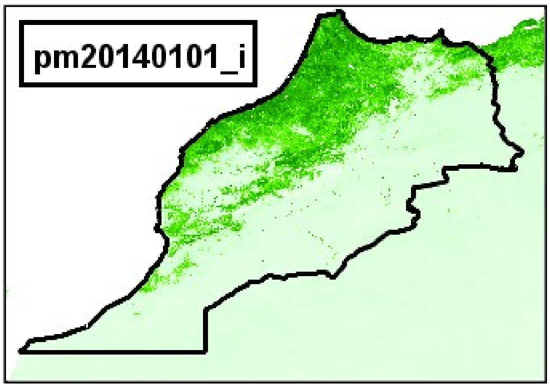
tiles January dekad 1



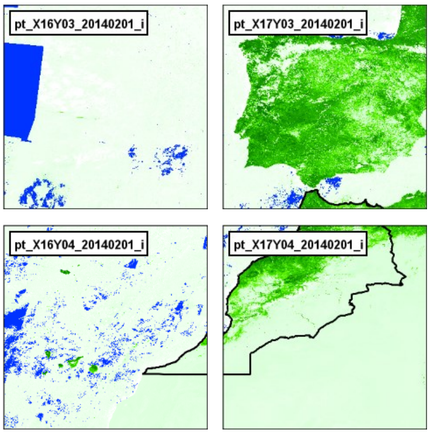
tiles January dekad 2



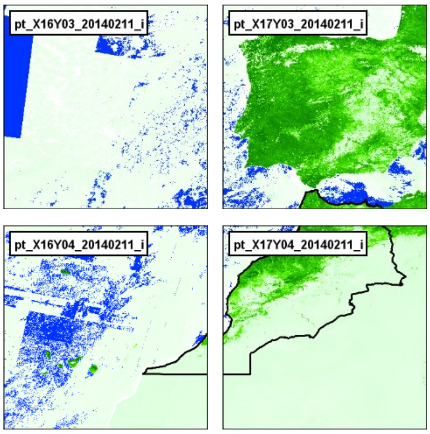
tiles January dekad 3



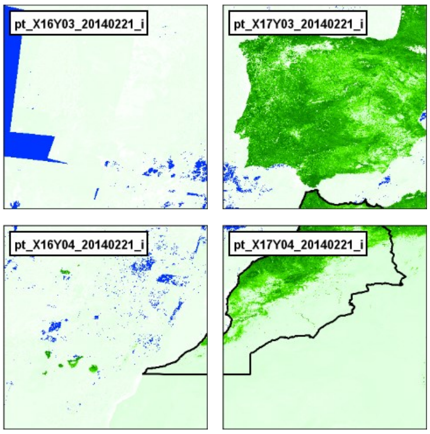
spatio-temporal composite January



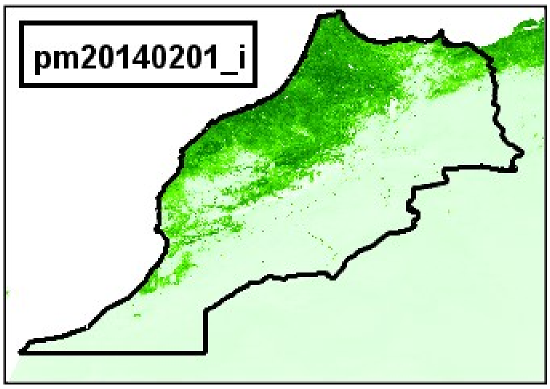
tiles February dekad 1



tiles February dekad 2



tiles February dekad 3



spatio-temporal composite February

...

3.17. Smooth

Goal

Smoothing of time series of images (vegetation indices (VI) such as NDVI and SAVI or biophysical state variables such as fAPAR, fCover or LAI).

In general, these series are still perturbed by noise due to missing values, data errors and especially clouds. The tool allows to generate an image series with cleaned values. For this “profile cleaning”, four different smoothing algorithms are available: MEAN (Interpolate missing values & apply Running Mean Filter RMF), BISE (Best Index Slope Extraction), SWETS and WHITTAKER.

Parameters

The tool needs two groups of parameters:

- general parameters which mainly describe the in- and output data (location, filenames, periodicity, properties of the input and output IMGs,...).
- specifications selecting the algorithm to use, together with its specific parameters and parameters for the preliminary and final adjustments of the series.

general parameters

- the start and end dates of the entire input IMGs series (including the optional IMGs -historical or those from previous year- that will be used to replace missing IMGs at the edges of the series);
- the location and filename structure of the actual input IMGs. The IMGs must have byte data type (HDR entry data type = 1);
- the series periodicity (Day, Dekad and Month are supported);
- the maximum number of consecutive missing actual IMGs allowed in the centre of the series;
- whether or not images from previous or historical year should be used to replace lacking input images at the edges of the series. If so, their location and filename structure must be specified;
- the amount of periods to copy from the profile end to before the profile start, and from the start to the end. This can only be used in case the input series covers exactly one or more full years;
- the start and end dates of the output IMGs series;
- the location and filename structure of the output IMGs;
- the lowest physical value for cloudfree land pixels. In case this parameter is left blank, the value will default to the minimum value according the the HDR file: Vint + Vslo * Vlo. Observations below this value are reset to missing values;
- the maximum number of missing values per pixel profile, in % of the total number of input IMGs. The output pixels for observations above this value will be flagged as “missing over land”;
- whether or not extrapolation is allowed at the edges of the series;
- the method to handle flags:
    - in case the input images are UNI-Flagged, their flags (251-255) can be copied to the outputs;
    - otherwise only two flags will be used (251=missing over land, 254 = water);
- whether or not to create the VAR (and MTA) metafile, containing the output IMGs names. If so, its location and filename must be specified;

smoothing algorithm parameters

The actual smoothing process is executed in three steps, each requiring their own parameters.

- the Pre-Smooth step - to eliminate dubious observations;
- the actual Smooth step - applying the selected algorithm (MEAN, BISE, SWETS or WHITTAKER);
- the Post-smooth step - adapting or resetting dubious resulting values.

Pre-Smooth:

- YmaxDip: The first test eliminates all local minima, whose absolute difference to both nearest-by values on both sides surpasses this threshold value. If one of both neighbour values is missing, only the other, good one is tested. If both are lacking, the test is skipped.
- PreMaxTop: The second test withdraws all local maxima, if both neighbours on both sides have "good" values and if the mean of the absolute differences to these two neighbours exceeds this threshold. Because in general the higher VI-values are the most reliable ones (that's also the rationale behind max-NDVI compositing rule), PreMaxTop should be given a very high value (say at least 0.4 for NDVI). If lower, also logical and significant maxima could be withdrawn. The sole objective of this test is to detect exuberant upward jumps, which are mostly due to errors in the pre-processing of the data.
- PreMaxGap: The third test deals with the treatment of longer gaps of missing values. These gaps can be reset to lowest physical value for cloudfree land pixels (specified in the general parameters) if they last longer than PreMaxGap IN-periods. This test targets boreal regions where long periods of missing values can occur in winter;
- PreMaxGapMsk: a snowmask (byte IMG) can be specified (optionally) to finetune the PreMaxGap test. For all pixels where this mask is greater than zero (mid-latitude to boreal regions), the PreMaxGap approach is applied. wherever the mask value is zero (equatorial zones, where gaps can appear due to persistent cloudiness), PreMaxGap is reset to zero.

Smooth:

The four smoothing algorithms discussed below take as input the adapted vector Yh(Ni), in which a number of points are labelled as "missing" by the pre-smooth step, and create a vector Ys(Ni) with smoothed values.

The program starts by defining the sequential ID of the first and last "good" (non-flagged) observations in vector Yh(Ni), which are labelled below as I1 and I2.

In case extrapolation is not allowed at the edges of the series (general parameters) the smoothing will be restricted to the central part of the profile (from I1 to I2) and the tails (if any) will remain flagged. Otherwise suitable Ys-estimates will also be defined for the missing IN-values in the profile tails, as a consequence, the smoothed vector Ys(Ni) will only contain "good" (non-flagged) values.

- MEAN method:
  - MUinterpol: if selected, the missing values in the centre of vector Yh(I1 to I2) are first replaced by linear interpolation. Otherwise interpolation is skipped and the series centre still may contain missing values.
  - MUrmf: The Yh profile is smoothed by means of a running mean filter RMF with a sliding window whose length (MUrmf) is defined in terms of days. If 'MUrmf' is set to 0 the smoothing is skipped, meaning the resulting vector Ys(Ni) is a copy of Yh(Ni) – previously interpolated or not via MUinterpol.

- BISE method:

The original BISE procedure is described in: Viovy N, Arino O & Belward A, 1992, The Best Index Slope Extraction (BISE): a method for reducing noise in NDVI time-series, International Journal of Remote Sensing, Vol. 13: 1585-1590. The implemented version is only loosely inspired by this paper and strongly deviates from the original. Only dekadel series are supported.

- BiWinDekads: Length of Sliding Window in DEKADS: the procedure tries to find the best compromise between two opposite goals: conservation of the good measurements and detection/elimination of suspect values. It begins with the search of a measurement which is certainly reliable, and not contaminated by clouds or snow. De facto that is the maximal value in the vector Yh(Ni). Beginning from this "starting point", the 'BiWinDekads' following dekads are examined towards the right/end of the series. Within this group of values, the program searches the two dekads which respectively have the highest value and the highest slope with respect to the starting point. Amongst these two – most often they occur simultaneously – the dekad nearest to the starting point is retained. This dekad is again considered "reliable", its value is copied to Ys, and it will serve as new starting point for the following iteration. In other words, all measurements (here: 1 to 'BiWinDekads'-1 at the most) between the starting point and the retained dekad are considered unreliable and hence they remain rejected (flagged). This process is repeated until the end of the time series is reached. The rationale of this method is that clouds can be very persistent, but that within a period of 'BiWinDekads' at least one clear-sky registration may be expected. Once the rightward search is finished, the procedure is repeated in a leftward way, beginning again from the maximum value until the start of the series is reached.
- BiMaxDif: Max. allowed VI-change per Dekad: a number of unreliable measurements might have slipped through the net, and after the above elimination step there often remain some abrupt and unlikely dips in the profile. This is a tricky matter, because these local minima can also be due to relevant phenomena on the ground (for example, the harvest of an important crop). However, the objective is to eliminate only those dips where the abrupt descent and rise of the concerned values surpasses the potential growth dynamics of the vegetation. The retained points in the profile (vector Ys) are scanned and searched for the dekads with local minima. For each such dekad, the slopes (dVI/dDekads) are computed w.r.t. the two retained neighbours (which by definition have higher values). If the mean of the absolute value of both slopes exceeds the 'BiMaxDif' threshold the concerned dekad is eliminated (flagged).
- BiIterate: selection whether or not to repeat step above; the elimination of unreliable values until no Dips remains

Based on the remaining good and reliable observations, all intermediate, missing values in the centre of the profile (vector Ys(I1 to I2)) are finally replaced by simple linear interpolation.

- SWETS method:

This procedure is based on the following paper: Swets, D.L, Reed, B.C., Rowland, J.D., Marko, S.E., 1999. A weighted least-squares approach to temporal NDVI smoothing. In: Proceedings of the 1999 ASPRS Annual Conference, Portland, Oregon, pp. 526-536.

Initially All missing values in the profile centre, Yh(I1 to I2), are first replaced by linearly interpolated values. Each value is classified into one of five types (slope, min, max, plane, edge) and depending on this type, each point is a given a weight (normally: max > plane > slope/edge > min).

- SwWmax: weight local maxima;
- SwWmin: weight local minima;
- SwWplane: weight planes (same value as 2 neighbours);
- SwWslope: weight for all others in regular profile;
- SwWedge: left/right edge point in profile;

Distinction is made between “regression and combination windows”, with lengths defined in terms of days:

- SwWinR: length in days of regression window
- SwWinC: length in days of combination window

All observations or points in the central part of the series (I1 to I2) are then treated as follows:

Apply a weighted linear regression to the observations in the regression window (central observation + SwWinR/(days per period) at left, SwWinR/(days per period) at right);

Then apply the found equation (Y=A+B.I) to all the I-points in the combination window, thereby obtaining Y-estimates for these points.

This way, for each I-point (SwWinC/(days per period)) Y-estimates are obtained, derived from the regressions applied on the neighbouring series of length SwWinR/(days per period).

The final smoothed value Ys(I) of the concerned point will be the mean of these estimates.

Optionally, the calibration step (definition of A, B) can be run iteratively and in each iteration possible outliers are removed using a CHI^2-test. The regression constants A and B are only defined on the final set of retained values. This step is selected and specified via the SwCI parameter:

- SwCI: Confidence Interval in %, for outlier correction with CHI2-test. "0" skips this test.

- WHITTAKER method:

The procedure is described in the following paper: Atzberger C & Eilers P (1999), Evaluating the effectiveness of smoothing algorithms in the absence of ground reference measurements, International Journal of Remote Sensing, Vol. 32, No. 13, 10 July 2011, 3689–3709. The implementation is based on code, kindly provided by co-author Clement Atzberger (BOKU-Vienna).

- WhKappa: The method ingests input vector Yh(Ni) data and then generates a smoothed vector Ys(Ni) by minimizing the function Q=S+ WhKappa.R. The “smoothness” S (or rather the lack of it) is defined as the sum of squares between vectors Ys and Yh. The “roughness” R is computed as the sum of differences between each Ys-value and its one or two neighbours on both sides. Parameter WhKappa gives relative weights to both factors. Higher values will yield smoother output profiles, largely deviating from the original Yh-inputs. Oppositely, lower values retrieve less smoothed profiles which more closely fit the original data. In practice, function Q is minimized by solving a matrix equation

which also retrieves the final output vector Ys(Ni). Missing values are replaced by logical and “good” (non-flagged) estimates, based on the entire set of input values;

- WhMethod: the program foresees two variants: the roughness R can be computed as the sum of the first or second order differences between each Ys-value and its neighbours;
- WhNiter: the smoothing is applied iteratively WhNiter times. Iterative application can be deactivated by selecting "1";

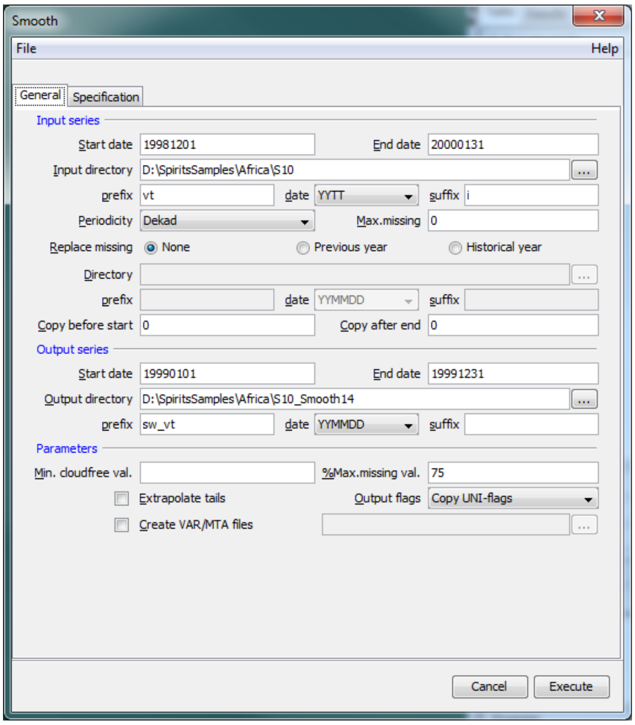
Two types of iterative smoothing can be chosen: (1) smoothing is iteratively applied to fit the upper envelope of the time series under the assumption that atmospheric perturbations result in sudden drops of the VI. At the first iteration the original time series is smoothed. At each subsequent iteration, the smoothing algorithm is applied to an updated time series where all observed values that are smaller than the curve fitted in the previous iteration, are replaced by their fitted values (this is called “upper envelope”). (2) iterative smoothing without upper envelope adaptation: iteration n simply starts with smoothed results of iteration n-1.

- WhMETiter: selection of the type of iterative smoothing.

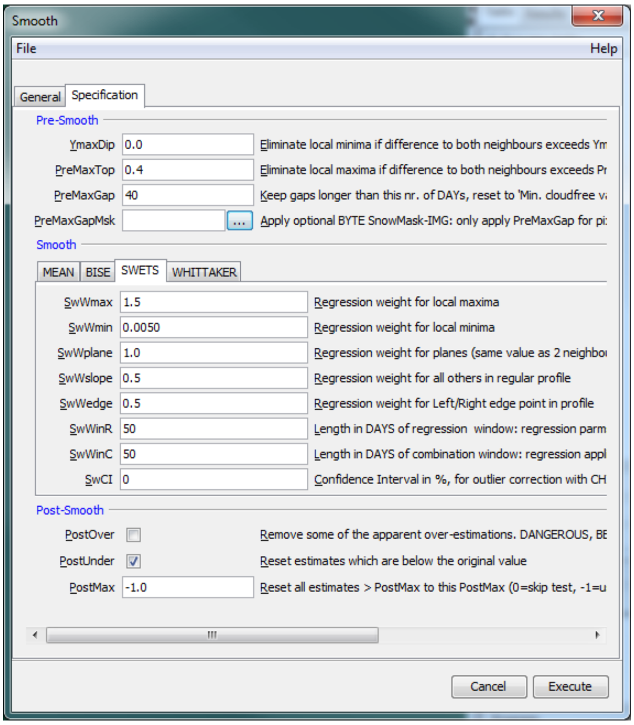
Post -Smooth

- PostOver: if selected each resulting value Ys(I) is reset to the mean Ys of its neighbours (I-1, I+1), whenever this smoothed Ys(I) appears to be higher than its own original value Yh(I) and than the ones of the two neighbours: Yh(I-1) and Yh(I+1). This is only be applied in case SWETS or MEAN with MUinterpol=1 have been selected, otherwise it is automatically skipped;
- PostUnder: if selected each initial Ys(I) estimate is compared to its original input value Yh(I) – possibly adapted via the Pre-Smooth operations or later interpolations. If Ys(I) is lower than Yh(I), it is reset to this Yh(I). (Typically used for cases where one can assume that the highest observations are the most reliable ones.);
- PostMax: Via PostMax, cases where the smoothing algorithm yields unexpectedly high Ys values, can be re-adapted. Three possibilities:
    - in case PostMax=0: the test is skipped;
    - in case PostMax=-1: the maximum of all original Yh(I)-values is searched for each pixel, and higher values are reset to this maximum;
    - in case PostMax=any maximum physical value: higher Ys(I)-values are reset to this maximum.

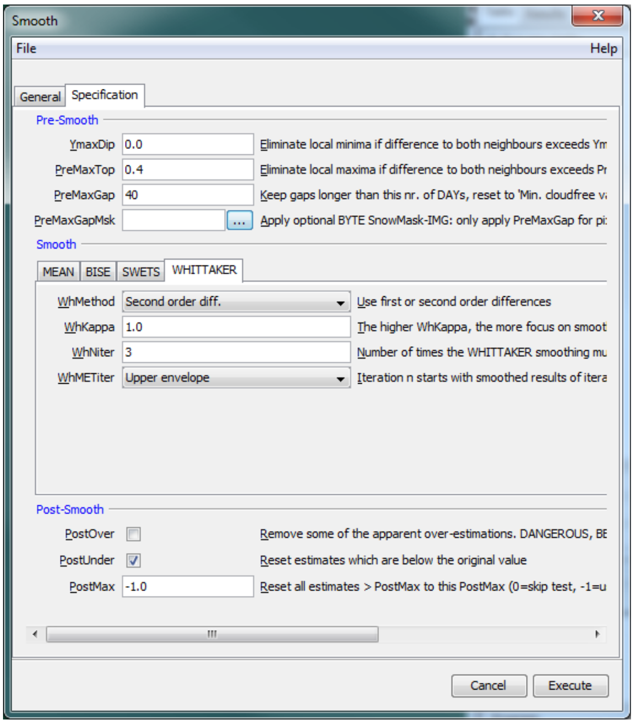
Tool



Smooth tool - general parameters example



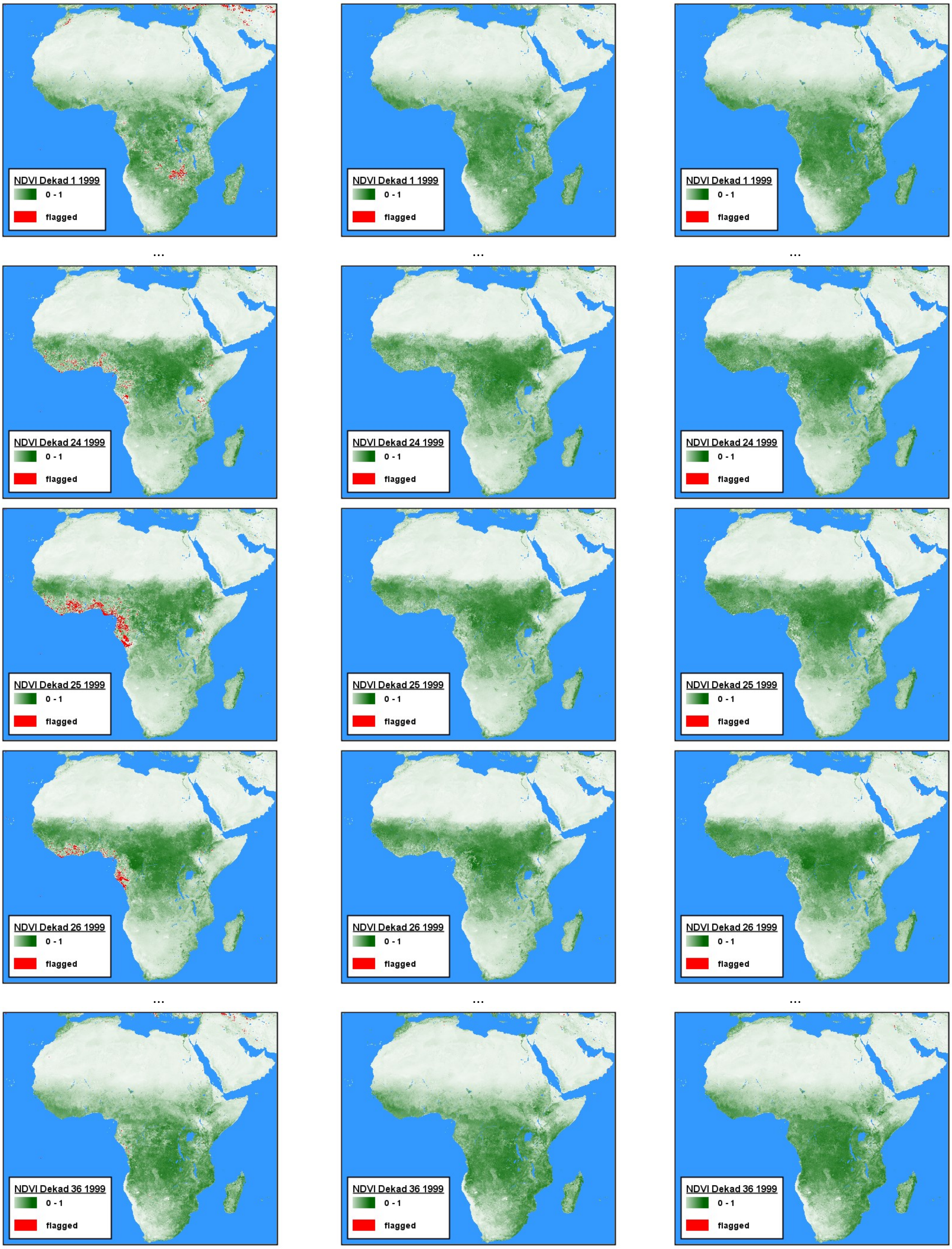
Smooth tool - specifications example (SWETS - algorithm selected)



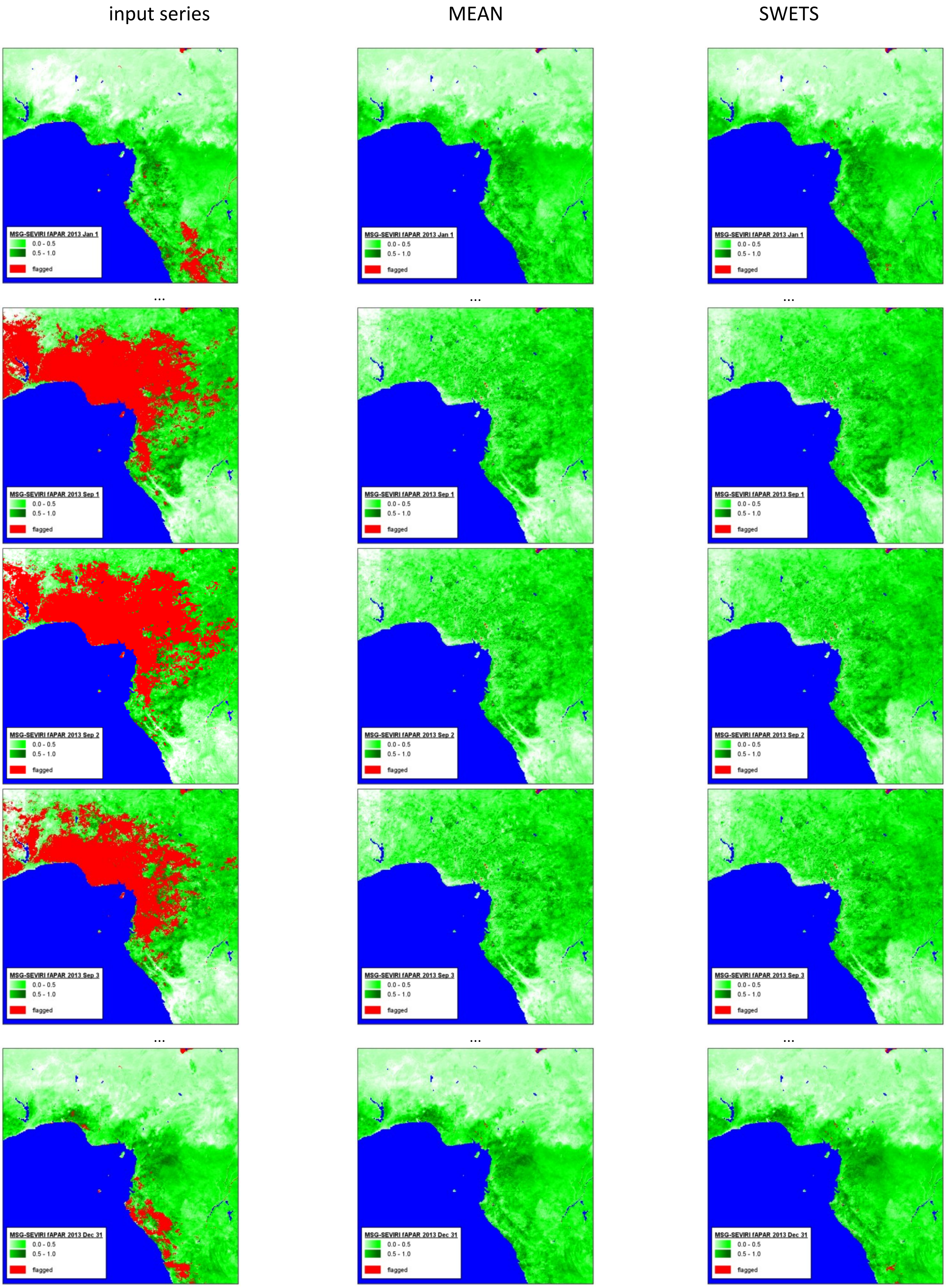
Smooth tool - specifications example (WHITTAKER - algorithm selected)

Example: smoothing SPOT-VGT NDVI S10 Africa

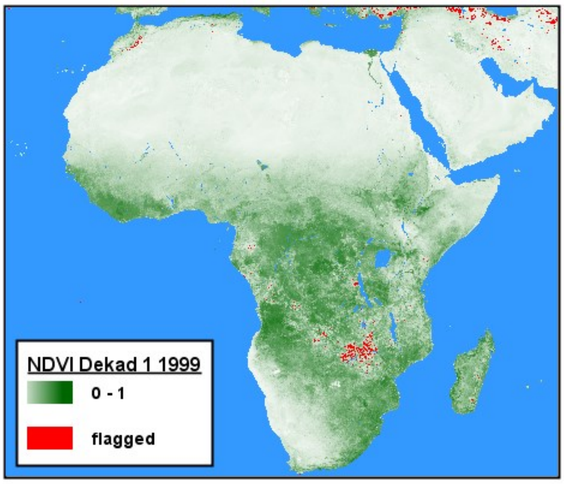
input series SWETS BISE



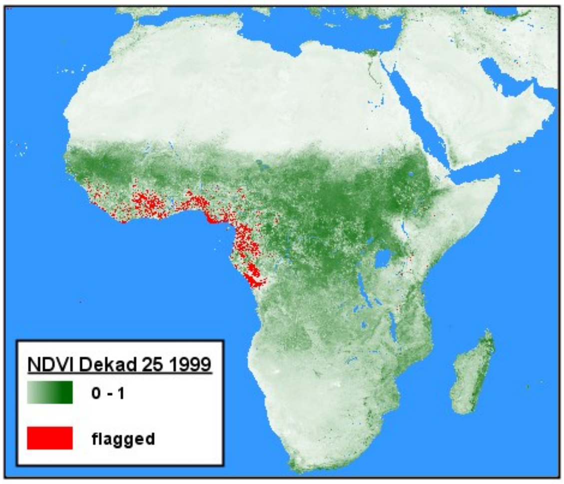
Example: smoothing MSG-SEVIRI-fAPAR S1 Gulf of Guinea



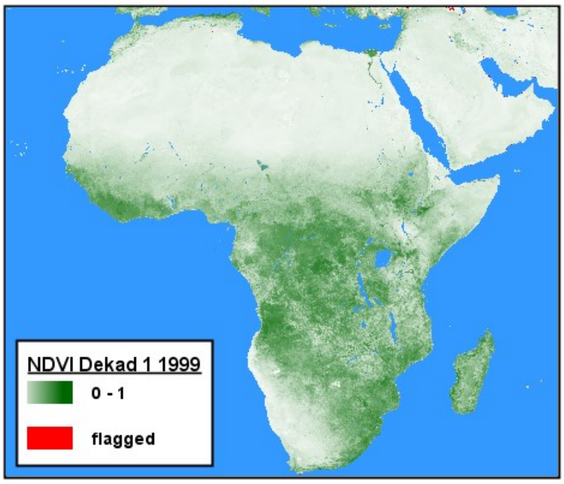
Example: smoothing SPOT-VGT NDVI S10 Africa - Whittaker



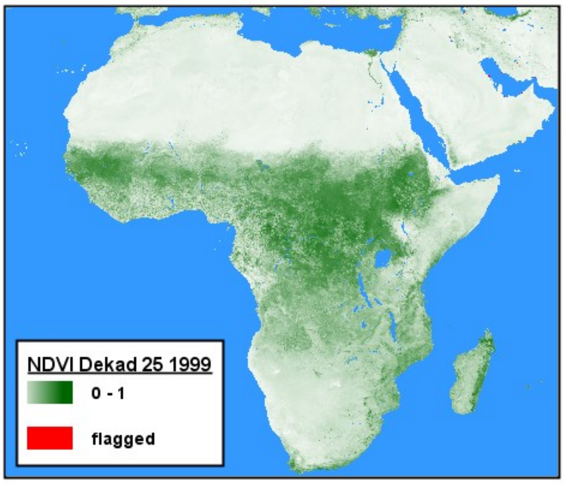
…



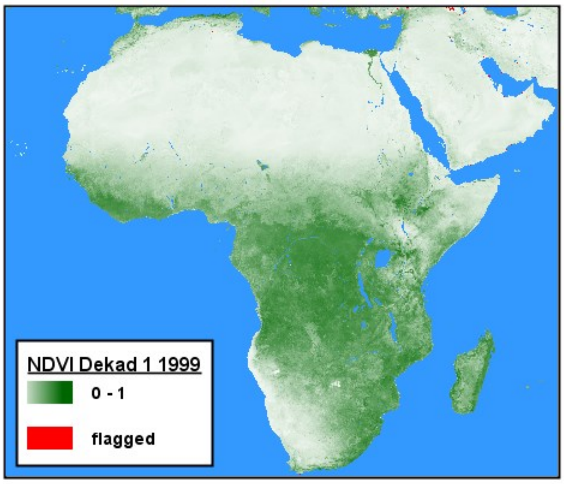
input series



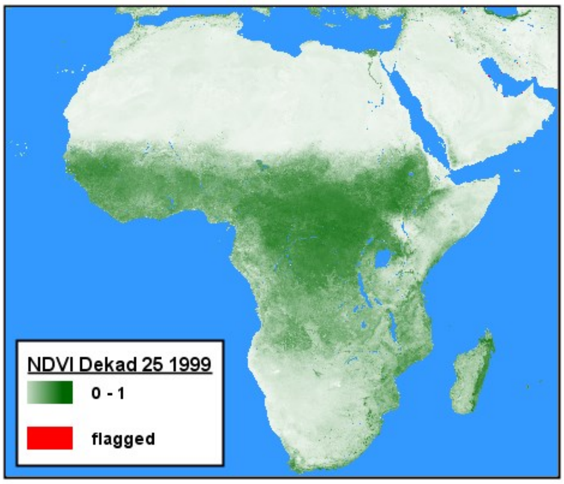
…



WhMethod = second order diff, WhMETiter= upper envelope, WhKappa = 0.05, WhNiter = 1



…



WhMethod = second order diff, WhMETiter= upper envelope, WhKappa = 20, WhNiter = 5

3.18. Cumulate

Goal

Compute sum/mean IMGs from a time series of IMGs.

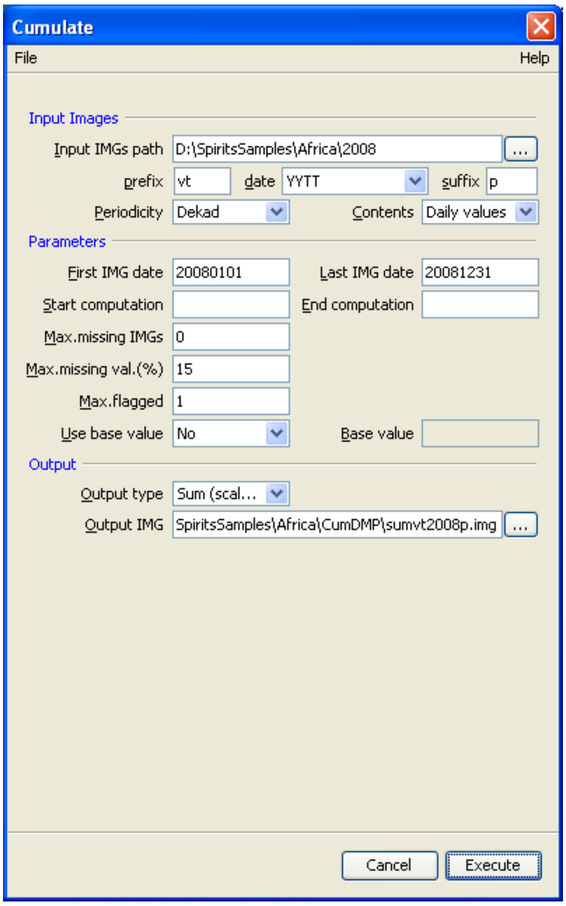
Parameters

- the location of the input IMGs;
- the filename structure of the input IMGs;
- the periodicity of the input IMGs;
- the content type of the input IMGs, this can be:
  - Periodic sums (Dekadal or monthly [X/dekad, X/month]);
  - Daily values (possibly means per period [X/day]). Results are weighted as to variable nr. of days in the periods (dekad/month).
- the dates of the first and last input IMG,
  these can include tails to obtain better extrapolation at start/end;
- the (optional) start and end date of the actual computation,
  these dates default to the first and last IMG dates;
- the maximum number of consecutively missing IMGs in the series;
- the maximum percentage of missing or flagged pixel values in the series. Pixels above this maximum will be flagged;
- the maximum. nr. of flagged observations at both ends of the series;
- whether to use a (physical) base value (***YBase***) and how to use it:
  - no base value is used;
  - using a base value as a lower limit: all values below the base value are reset to this value;
  - using a base value to calculate differences (***Ydiff = Y-Ybase***, or 0 in case Y<Ybase). This is typically used for temperature sums);
- the output type:
  - mean value. The data type, scaling and flags of the output IMG will be those from the input IMGs;
  - cumulative sum. The data type and flags of the output IMG will be those from the input IMGs. Scaling will be applied (so the sum will 'fit' in the original data type);
  - de-scaled cumulative sum. The output IMG will have float data type, no scaling and will use fixed flags.
- the output IMG location and filename;

Remarks:

- All input IMGs should have the same datatype;
- All input IMGs should have the same significant range (Vlo-Vhi entries in the HDR);
- All input IMGs should have the same scaling (Vint, Vslo entries in the HDR);
- All input IMGs should have the same flags.

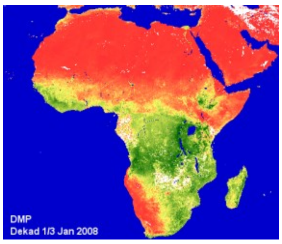
Tool



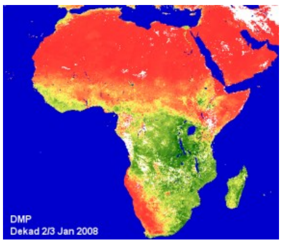
Cumulate Tool example

Cumulative example

example: compute cumulative sum of DMP over a year.

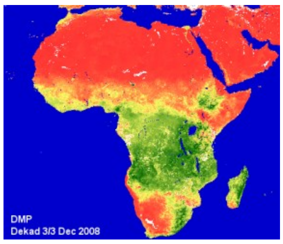


Dekad 1

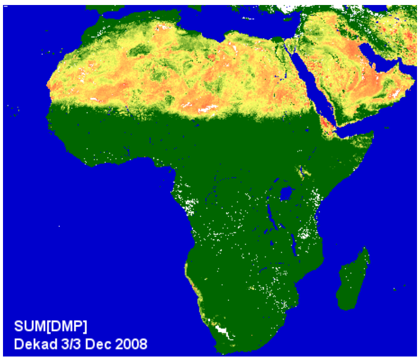


Dekad 2

...

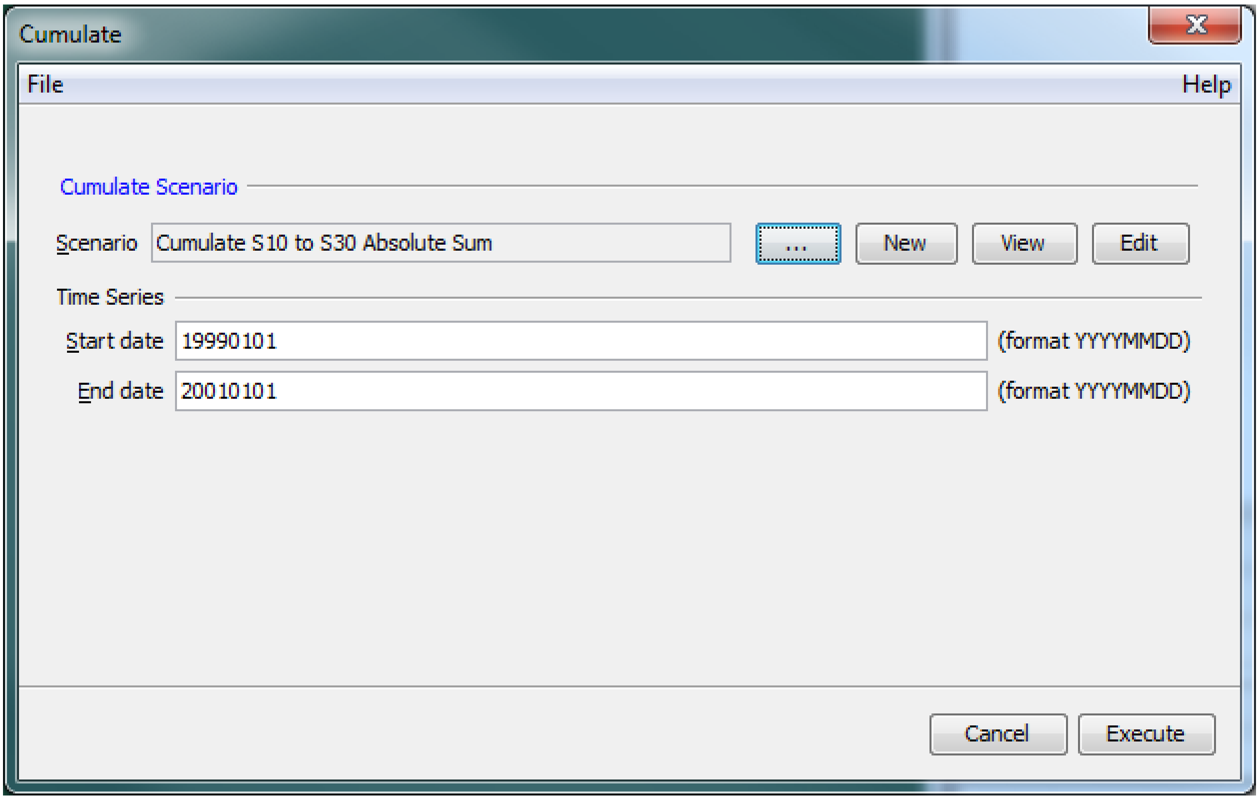


Dekad 36



Sum Dekads 1 -36

Time Series



Cumulate Time Series example

Scenario

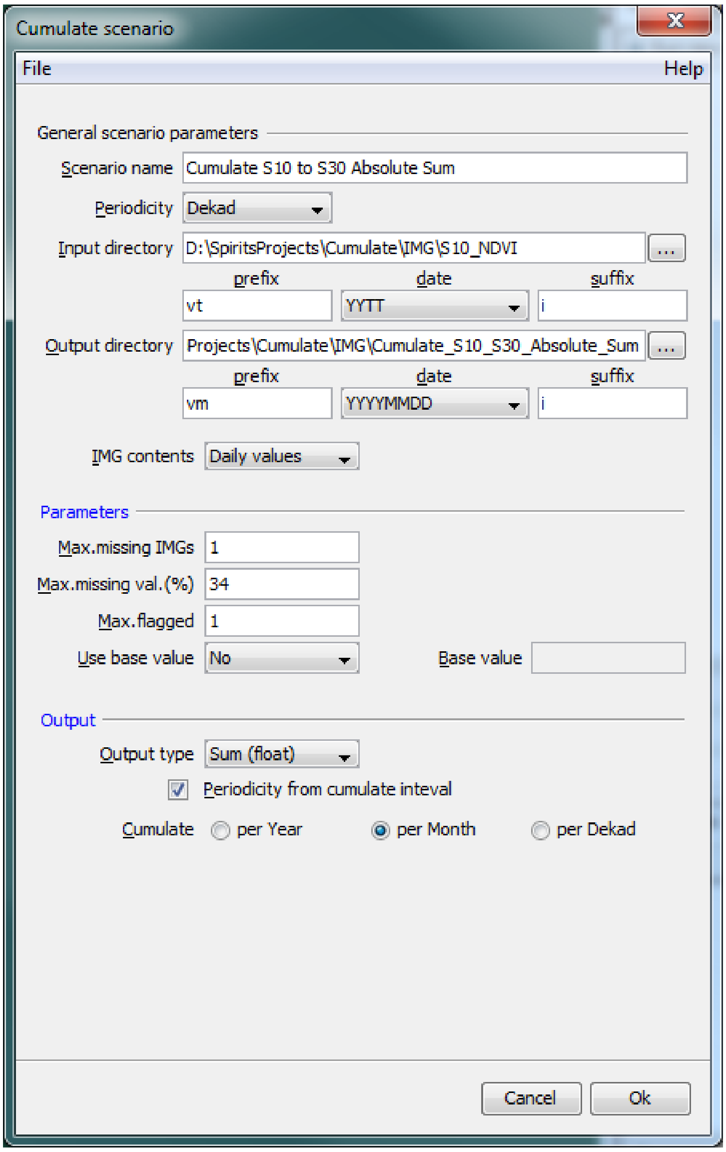
As for most other scenarios/time series the periodicity parameter of the Cumulate scenario refers to the periodicity of the input IMGs.

However, in contrast with most other scenarios/time series, the resulting IMGs can have a periodicity and date which is different from those of the input IMGs. This is determined via the “Periodicity from cumulate interval” option:

- In case the “Periodicity from cumulate interval” option is not selected, the time series behaves just as the Cumulate tool:
  - the periodicity of the output IMGs is that of the input IMGs;
  - the date of an output IMG is that of the last input IMG in the cumulate interval.

- In case the “Periodicity from cumulate interval” option is selected:
  - the periodicity of the of the output IMGs is determined by the selected cumulate interval (thus Year, Month or Dekad);
  - the date of an output IMG is that of the first input IMG expected in the cumulate interval.

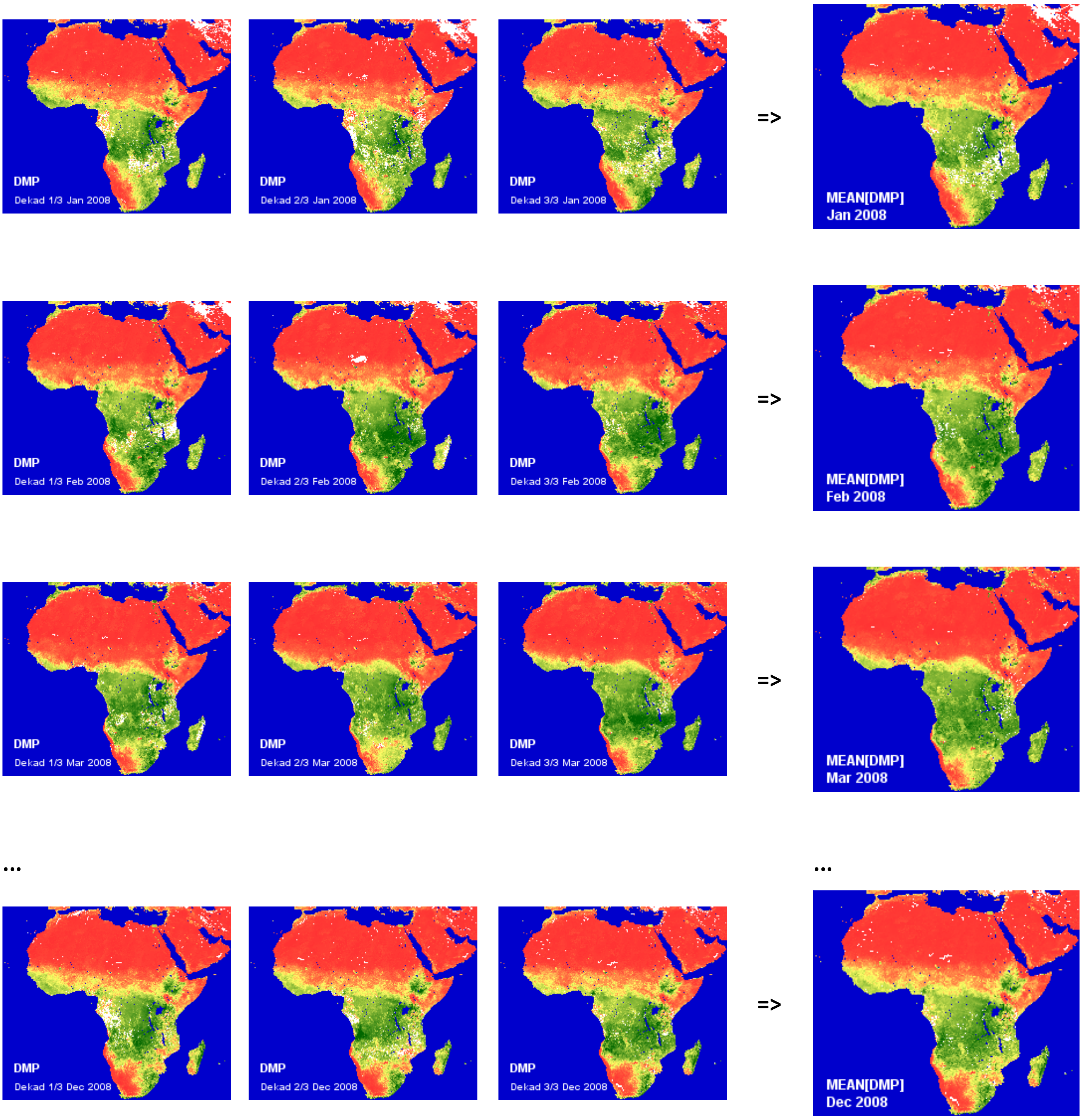
The output IMGs date will reflect in the IMG filename as well as in the date entry in its HDR file.



Cumulate Scenario example

Averaging example

example: calculate monthly averages in 2008 from dekad IMGs



Dekad 1/3 Dekad 2/3 Dekad 3/3 mean

input IMGs series Cumulate output IMGs

3.19. Pheno

Goal

Extract phenological parameters from a series of periodic IMGs. Twelve different phenological parameters can be extracted.

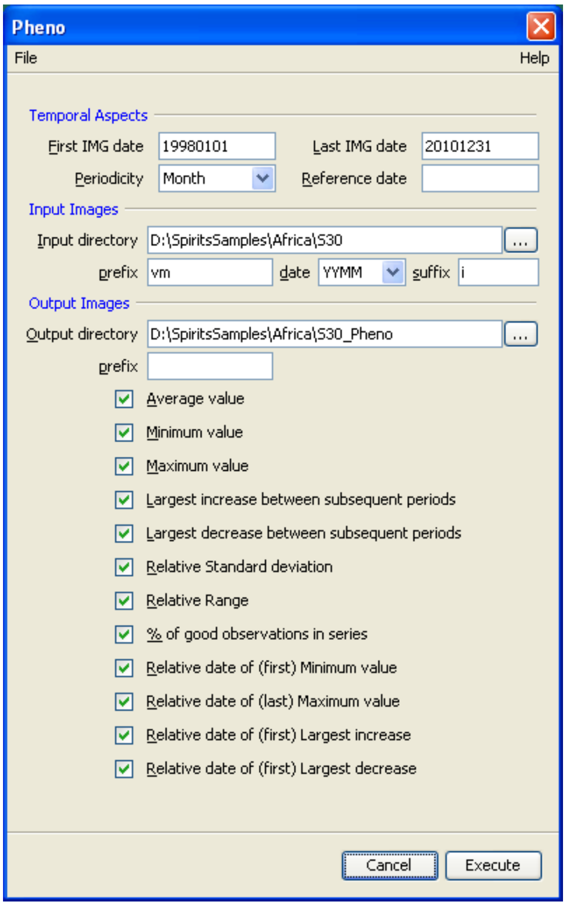
Parameters

- the periodicity of the input IMGs;
- the date of the first input IMG;
- the date of the last input IMG;
- the reference date: start date of the time frame [reference date-last input IMG date] against which the relative date IMGs ("Dmn", "Dmx", "Dup", "Ddn", see below) will be calculated. If left blank the date of the first input IMG is used as reference date;
- the location and filename structure of the input IMGs;
- the location and filename prefix of the output IMGs;
- the requested output IMGs. The filename structure of the output IMGs is as follows:

Lo\PoD1_D2Si_So.img. (example C:\Temp\vt0801_0836i_Vav.img)

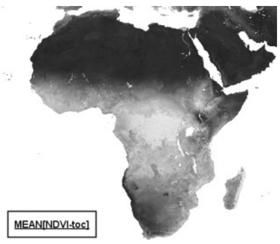
Lo	output IMGs directory	
Po	output IMGs prefix	
D1	date of the first input IMG, using the input IMGs date format	
D2	date of the last input IMG, using the input IMGs date format	
Si	input IMGs suffix	
So	output IMGs suffix - coding the different phenological parameters	
	Vav	Average value (or Mean)
	Vmn	Minimum value
	Vmx	Maximum value
		Vav, Vmn and Vmx have the same data type and scaling as the input IMGs
	Aup	Largest increase (angle) between subsequent periods
	Adn	Largest decrease (angle) between subsequent periods
		Aup and Adn use angles (-90 to +90 degrees) scaled to BYTE (0-180)
	Rsd	Relative Standard deviation (with N as denominator, not N-1)
	Rrg	Relative Range (Maximum - Minimum)
		Rsd and Rrg use % of potential range (Yhi-Ylo), scaled to BYTE (0-200)
	Pok	% of good observations in series (not flagged for clouds/snow/...)
		Pok uses % of potential observations, scaled to BYTE (0-200)
	Dmn	Relative date of (first) Vmn
	Dmx	Relative date of (last) Vmx
	Dup	Relative date of (first) Aup
	Ddn	Relative date of (last) Adn
		Rel. position in time-frame between the reference date and the last IMG date, scaled to BYTE (0-200)

Tool

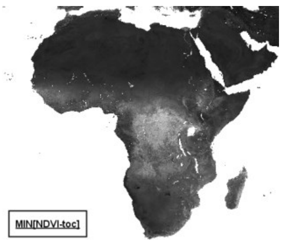


Pheno Tool example

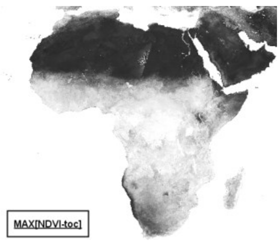
input IMG series(April 1998 - November 2009)



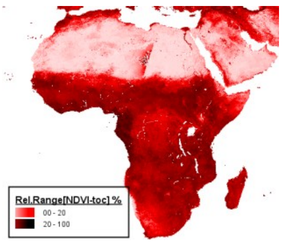
Vav (average)



Vmn (min)



Vmx (max)



Rrg (rel range)

3.20. Detect seasons

Goal

Derive phenological parameter IMGs from a dekadal time-series of IMGs.

The program inspects a time series of dekadal images with any type of Vegetation Index VI (NDVI, fAPAR,...). For a given “target year”, it detects (always per pixel) the number of green seasons Ns (Ns = 0, 1 or at most 2), and for each such cycle s the dates of the start (SOS), the maximum (MOS) and the end (EOS) of the season. Optionally some other information can be retrieved such as the VI-values at these three breakpoints, the season lengths, etc.

Parameters

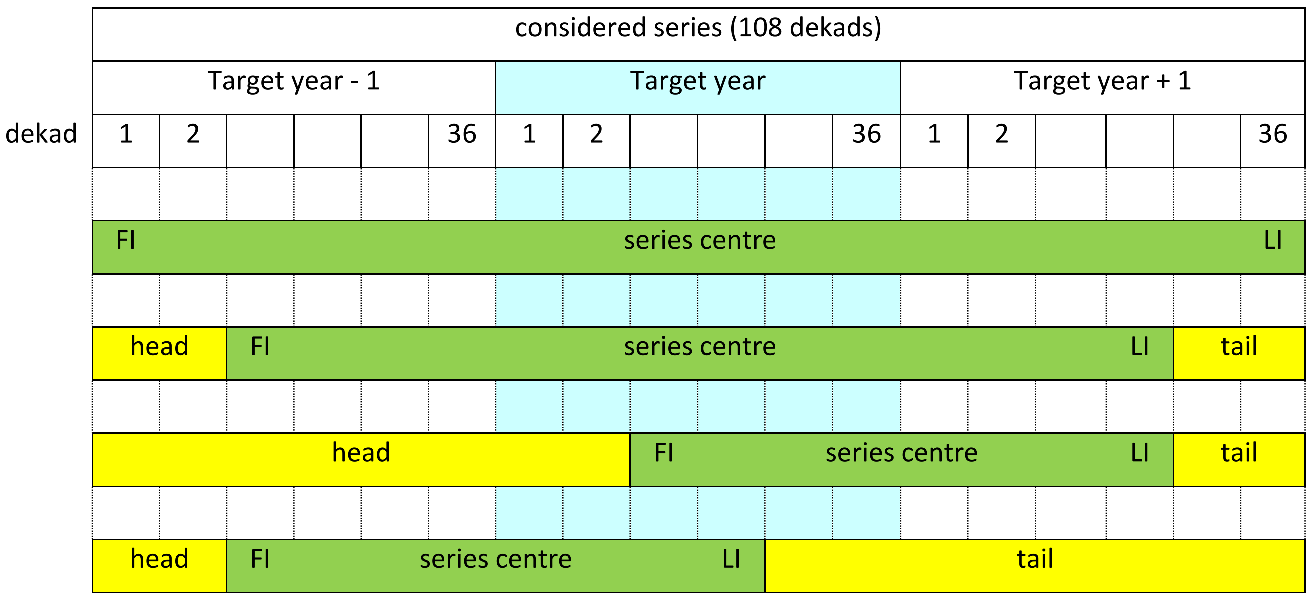
The tool can operate on actual IMGs or on historical IMGs. Operating on actual IMGs, outputs can be created for a single target year, or for a series of target years.

Parameters in case of using actual images:

- the target year(s): the year(s) the phenological parameters are to be derived for;
- the location and filename structure of the input IMGs (NDVI, SAVI, ...);
- the maximum number of consecutively missing IMGs in the series centre;
- whether or not to replace missing IMGs at the series edges (head and tail) with historical IMGs, and if so: the location and filename structure of the historical IMGs;

For a given target year, the tool considers a series of three civil years (108 dekads) with the specified target year in the middle.

The specified location is searched for files. The first (earliest in the considered series) IMG found (FI) determines the start of the series centre and the left edge (head) of the series). The last (latest in the considered series) IMG found (LI) determines the end of the series centre and the right edge (tail) of the series.



Following limitations apply:

- ○ the input IMGs must be spatially congruent with the same map info, framing, etc.
- ○ the input IMGs must be of BYTE type, and have the same range (Vlo-Vhi) and scaling (Vint, Vslo).
- ○ there must be at least 36 actual IMGs available in the series centre (target year);
- ○ there must be at least one IMG available in the target year, thus the first IMG (FI) must belong to the target year, or the preceding year and the last IMG (LI) must belong to the target year or the successive year;
- ○ in case heads or tails exist (centre contains less than 108 IMGs), they must be replaced with historical IMGs. Remark: the replacement of missing IMGs applies only in the series edges (head and tail areas). In the series centre the IMGs cannot be replaced.

Parameters in case of using only historical IMGs:

- the location and filename structure of the historical IMGs (none of these may be lacking);

Parameters for both cases:

- the maximum. percentage (0-100, typical 15) of missing values in 1 pixel-profile;
- whether or not to apply a mask, and if so:
  - ○ the location and filename structure of the mask IMG. This must be a byte-type IMG;
  - ○ the lower and upper (digital) mask values to include;
- the location and filename structure of output IMGs containing the phenological parameters.
- the parameters for the phenology algorithms (table below). Remarks:
  - ○ these parameters must be adapted to the nature of the concerned variable (NDVI [-],fAPAR[-], fAPAR [%], ...). The example values hold for NDVI.
  - ○ Y* (Ymin, Ymax, Y): indicate the physical values of the concerned variable
  - ○ T* (Tmin, Tmax, T): indicate the dekad [1-108] in which these occur

```
===============================================================================
Preliminary elimination of pixels without seasonality
(Ymax and Ymin are computed over the 36 dekads [37-72] of the central year)
FEN0Max  = 0.180   If Ymax        < FEN0Max => No seasonality (deserts)
FEN0Min  = 0.750   If Ymin        > FEN0Min => No seasonality (evergreen)
FEN0Rng  = 0.075   If (Ymax-Ymin) < FEN0Rng => No seasonality (variability too low)

-------------------------------------------------------------------------------
The Yi-profiles of the remaining pixels are smoothed with a weighted, running mean
filter (RMF)
=> Smoothed curve Ys.
FENrmf   = 2       Half length of the RMF (2=best, 0=skip filtering)
FENw     = 4       All dekadal values get default weight W=1,
                   but the extreme values (MIN/MAX) get weight FENw (best FENw=4)
                   NB: Tsos/Tmos and Teos (for both seasons) are derived from this
                       smoothed Ys, but the corresponding Y-values are extracted from
                       the original Yi-values
```

```
-------------------------------------------------------------------------------
Each smoothed Ys-profile is further decomposed in "segments",
i.e. pairs of MIN-MAX or MAX-MIN.
Each cycle or season comprises two subsequent segments:
the rising MIN1-MAX plus the descending MAX-MIN2. But even after the RMF-smoothing,
pixel profiles can still show many irrelevant cycles.
Via five subsequent tests (all executed iteratively), the irrelevant cycles are
searched and eliminated (always in subsequent pairs MIN-MAX or MAX-MIN).

FENdY    = 0.025   Test1: Eliminate segment if absolute value of
                          its Y-difference  < FENdY
FENdT    = 10             AND TOGETHER      if interval between
                          extremes Tmin-Tmax < FENdT (dekads)
FENmax   = 0.000   Test2: Eliminate all maxima with Ymax < FENmax.
                          Also eliminate the highest neighbouring minimum.
                          NB: Best skip this test via FENmax = 0.0.
                   Test3: The overall Ymin and Ymax are searched (over 3 years)
                          and a threshold is defined:
                          Yt = Ymin + FENratio * (Ymax-Ymin)
FENratio = 0.250          Eliminate all maxima with Y < Yt. Also eliminate
                          the highest neighbouring minimum.
                          Set FENratio=0 to skip this test.
FENmaxDt = 6       Test4: If interval between 2 maxima   < FENmaxDt (in dekads),
                          remove intermediate minimum and lowest maximum.
FENextDt = 3       Test5: If interval between 2 extremes < FENextDt (in dekads),
                          remove entire segment.

-------------------------------------------------------------------------------
At this stage and for most pixels, only one or two maxima (thus seasons) remain within
the central year. But exceptionally there can be more: three or even four. In that
case, the program searches and removes the least relevant maxima (+ highest
neighbouring minimum) until at the end only 2 seasons remain. For each season, the Area
is computed below the profile between MIN1-MAX-MIN2. At each iteration, the season with
smallest area is eliminated.
NB: No parameters are needed for this step.

-------------------------------------------------------------------------------
Two ratios with values between 0.0 and 1.0 (Ymin1/2 are the minima
before/after the concerned maximum Ymax)

FENsos   = 0.15    Tsos is defined as the dekad between Tmin1 and Tmax
                   where the profile cuts the
                   value Y = Ymin1 + FENsos * (Ymax - Ymin1).
FENeos   = 0.35    Teos is defined as the dekad between Tmax  and Tmin2
                   where the profile cuts the value
                   Y = Ymin2 + FENeos * (Ymax - Ymin2).
```

The file names or the output IMGs are formatted as "***BasenameSuffix.img***", using the specified base name, with different suffixes.

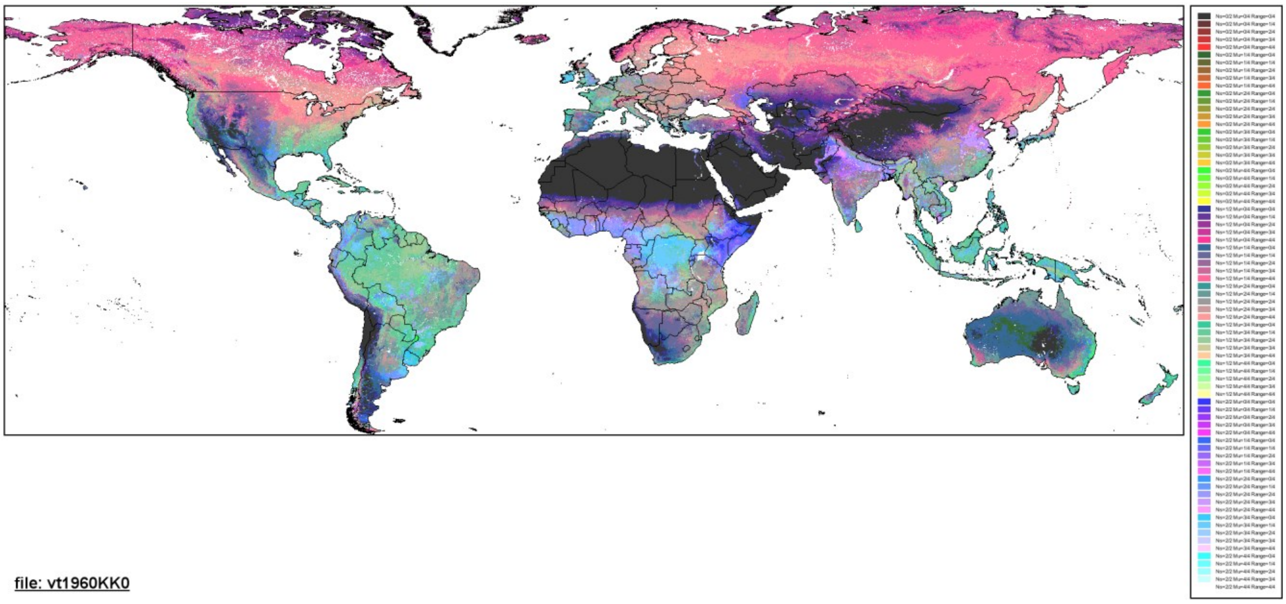
In each case following output IMGs are created:

- 6 IMGs with dekad of SOS/MOS/EOS for 2 seasons (suffixes s1/s2, m1/m2, e1/e2)
- 2 Metafiles Basename.VAR/MTA with lists of all created OUT-IMGs
- 1 TXT-file Basename.TXT with lists of problematic pixels.

Optional IMGs (and their parameters - table below) can be selected:

- ○ (3 IMGs) overall annual features (Ya0, Yr0, kk0),
- ○ (2 IMGs) for 2 seasons the Y-value at MOS (Ym1/Ym2).
- ○ (4 IMGs) for 2 seasons the Y-value at SOS and EOS (Ys1/Ys2, Ye1/Ye2)
- ○ (2 IMGs) for 2 seasons the season length (L1/L2)
- ○ (2 IMGs) for 2 seasons the mean Y-value (Ya1/Ya2)
- ○ (3 IMGs) season-dependent annual features (LT0, LR0, SI0)

```
------------------------------------------------------------------------------
FENYsm   = 0      Values in OUT-IMGs with YsS/YmS/YeS/YaS (Season S=1/2) are based on
                  original Y-values or on smoothed versions.
FENlDEK  = 1      Express season Lengths (L1, L2, LT0) as % of the year or in dekads.

------------------------------------------------------------------------------
Classes k in the optional OUT-classification (KK0) are defined as:
k=(100 * Ns) + (10 * Kmu) + Krg:
 - Ns  [0-2] = Seasons in year.
 - Kmu [0-4] = INTEGER[(Ya0 - MUlo)/ MUdelt], with Ya0 = Annual mean Y-value.
 - Krg [0-4] = INTEGER[(Yr0 - RGlo)/ RGdelt], with Yr0 = Annual Y-range (Ymax-Ymin).
If Kmu<0 or Kmu>4, it is reset to range [0-4].
Idem for Krg. Hence, the range of k-classes is limited to 0-244 (byte).The maximum
of 244 occurs if Ns=2, Kmu=4 and Krg=4.

MUlo   = 0
MUdelt = 0.2
RGlo   = 0
RGdelt = 0.15
```



example classification IMG (kk0)

Overview output IMGs

Scope	Suffix	Digital Nrs. V		$Y=V_{int} + V_{slo}.V$		Physical values Y			Y-CONTENTS
		V_{lo}	V_{hi}	V_{int}	V_{slo}	Y_{lo}	Y_{hi}	Y_{unit}	
36 dekads in target year	Ya0	*	*	*	*	*	*	*	Annual mean VI
	Yr0	*	*	0	*	*	*	*	Annual VI-range (Y_{max}-Y_{min})
	kk0	0	244	0	1	0	244	*	Classif. ~ Ns, Ya0, Yr0
Dates of season 1	s1	1	108	-36	1	1	72	dekad	Season 1: dekad of s=SOS, m=MOS, e=EOS Y=1: dekad 1 of target year
	m1	1	108	-36	1	1	72		
	e1	1	108	-36	1	1	72		
	L1	0	36	0	1	0	36	dekads	Season 1: Length
Dates of season 2	s2	1	108	-36	1	1	72	dekad	Season 2: dekad of s=SOS, m=MOS, e=EOS Y=1: dekad 1 of target year
	m2	1	108	-36	1	1	72		
	e2	1	108	-36	1	1	72		
	L2	0	36	0	1	0	36	dekads	Season 2: Length
Values of season 1	Ys1	*	*	*	*	*	*	*	Season 1: Y-value at s=SOS, m=MOS, e=EOS, a=mean over season
	Ym1	*	*	*	*	*	*	*	
	Ye1	*	*	*	*	*	*	*	
	Ya1	*	*	*	*	*	*	*	
Values of season 2	Ys2	*	*	*	*	*	*	*	Season 2: Y-value at s=SOS, m=MOS, e=EOS, a=mean over season
	Ym2	*	*	*	*	*	*	*	
	Ye2	*	*	*	*	*	*	*	
	Ya2	*	*	*	*	*	*	*	
Both seasons in target year	LT0	0	36	0	1	0	36	dekads	Length of seasons 1 + 2
	LR0	0	200	0	0.5	0	100	%	Length Ratio: L1/(L1+L2)
	SI0	0	250	0	1.0	0	250	%	Seasonality Index (**)

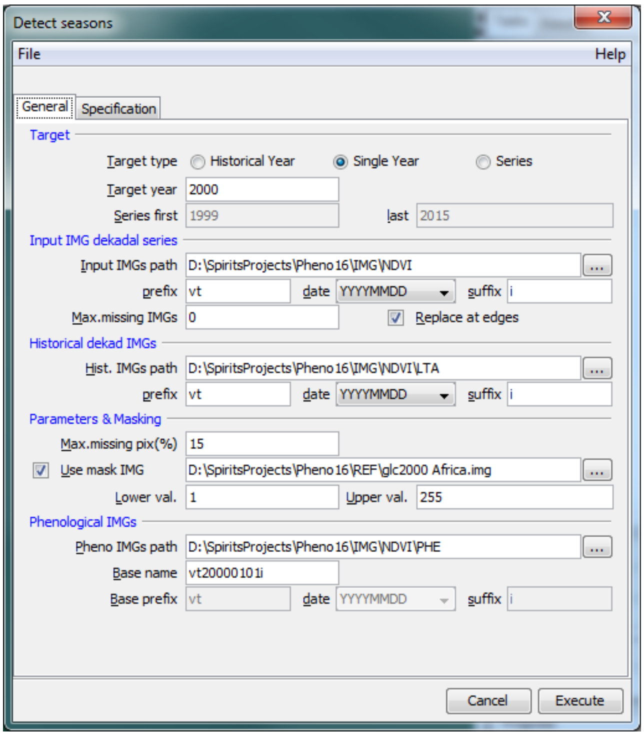
(*) inherited from input VI IMGs)

(**) Seasonality Index = 100 * (Ygs – Ya0) / Ya0 [%], with Ya0 the mean annual VI, and Ygs the mean VI in the one or two seasons (both restricted to the 36 dekads of the central year). By definition: Ygs=>Ya0.

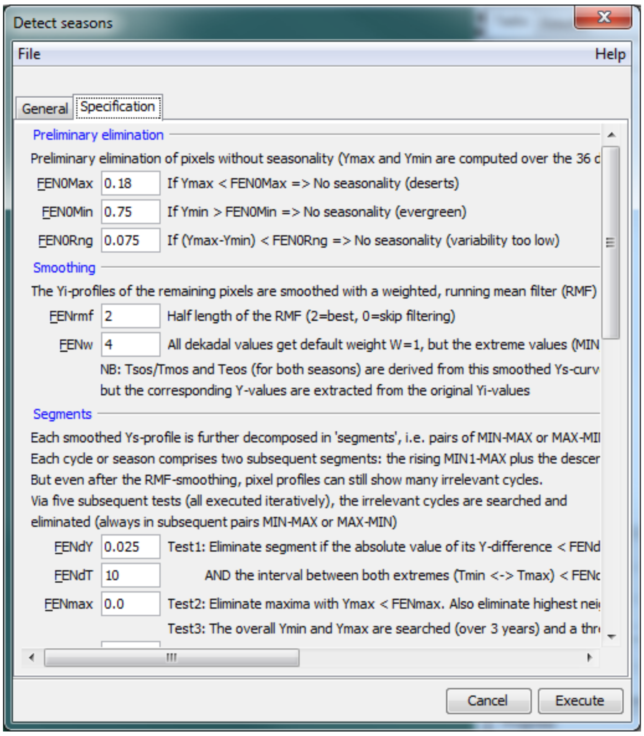
Overview output IMGs - flags

FLAG	MEANING
255	Water (all 108 input values are flagged)
254	Masked land pixel (only if a mask is used, see program parameters 12-14)
253	Insufficient non-flagged input values (see program parameter 10)
252	Processing error (not in Ya0, Yr0 and kk0)
251	Pixel without any seasonality (deserts, evergreen forests,... Not in Ya0, Yr0 and kk0). For all IMGs related to the second season: if this season 2 is absent.

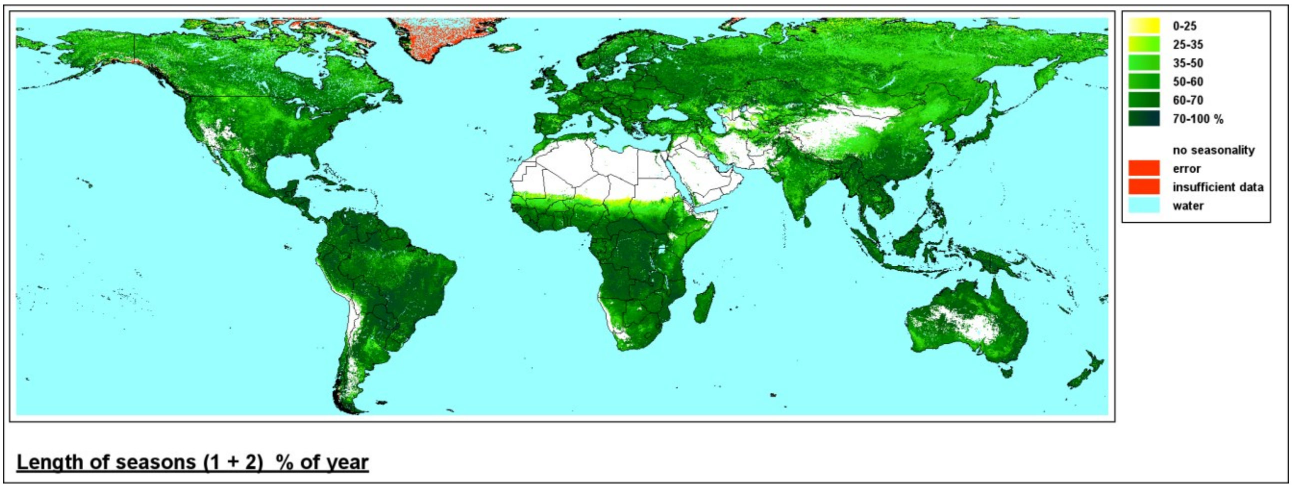
Tool



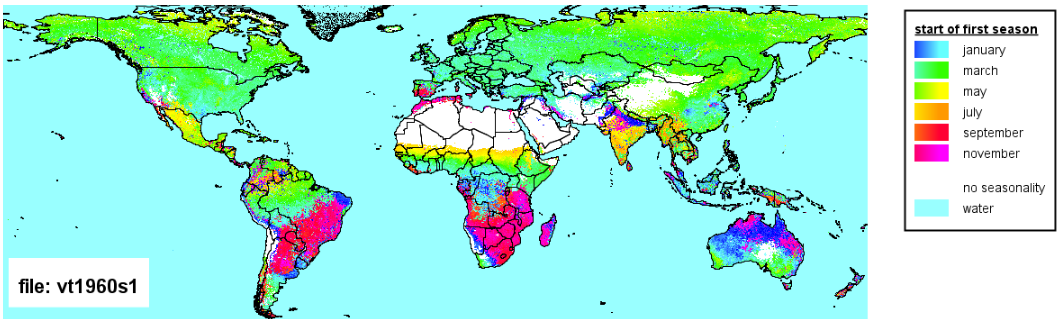
Detect seasons Tool example general parameters



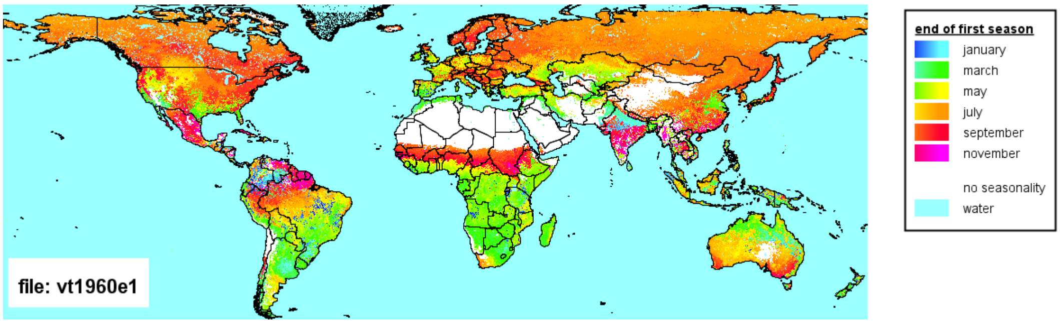
Detect seasons Tool example specifications



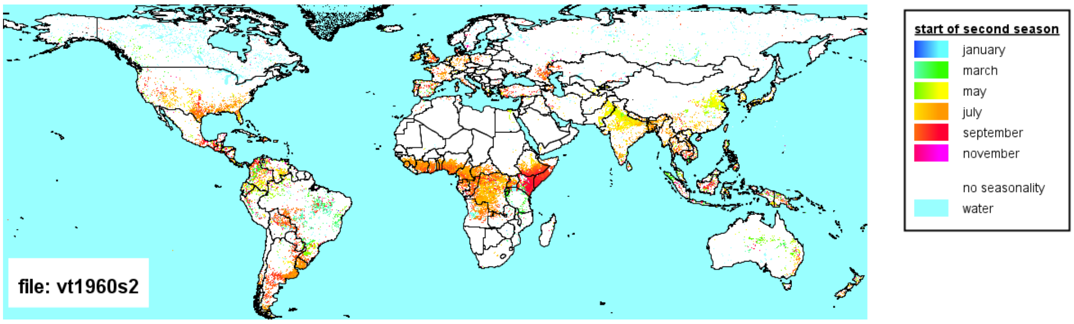
example: green season(s) length IMG (LT0)



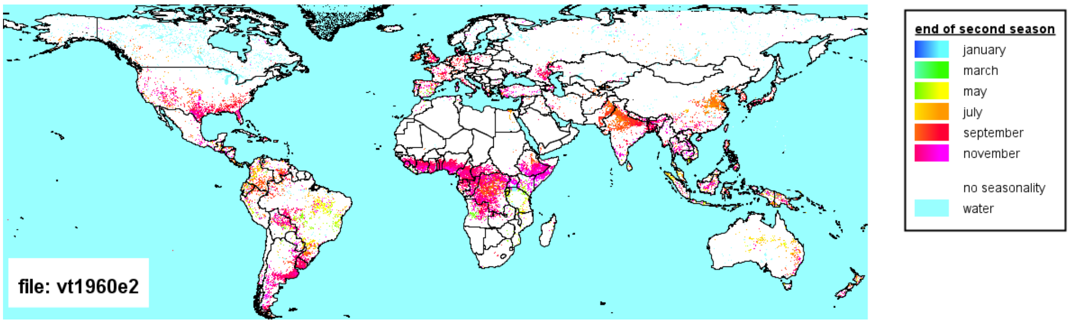
SOS 1 IMG (s1)



EOS 1 IMS(e1)



SOS 2 IMG(s2)



EOS 2 IMG(e2)

example: start and end of season IMGs

3.21. Progress of season

Goal

Calculate the progress of a season, per dekad, over the complete season.

The tool operates on the phenological IMGs (SOS/EOS) created by the "detect seasons" tool.

A set of phenological IMGs (SOS/EOS) represent phenological parameters either:

- of a season of the specific target year for which they were calculated;
- or of a season assumed to be constant and applicable for any year, in case they were calculated from Long-Term Averages (LTA).

In case the phenological IMGs have been derived from LTA-IMGs, the tool creates 36 output IMGs representing the progress for each dekad in the fictive (LTA) year 1962.

In case the phenological IMGs have been derived for and from a specific target year, to obtain the progress over the complete season three successive civil years have to be considered, with the target year in the middle. The tool creates 72 dekadal output IMGs representing the progress in:

- the last 18 dekads of the year before the target year;
- the 36 dekads of the target year;
- the first 18 dekads of the year succeeding the target year.

The "detect seasons" tool is capable of creating phenological IMGs for a series of target years. From these, the progress IMGs can be also be calculated for a series of target years at once.

Parameters

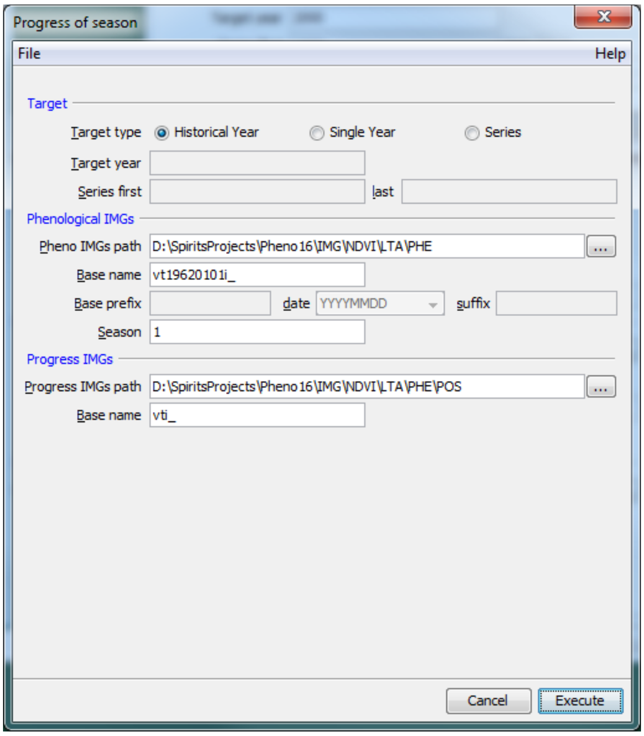
- the target year(s), the year(s) the progress IMGs have to be created for;
- the location and filename structure of the SOS and EOS IMGs as created by the Detect seasons Tool.
  - in case progress has to be calculated for LTA or single target year phenological IMGs, their base name has to be specified. The actual SOS and EOS IMG names used will then be ***BasenameSs*** and ***BasenameEs***, with *s* the season number (1 or 2).
  - In case a series of target years is selected, these IMGs are assumed to have a structured base name: basename = ***PrefixDateformatSuffix***. The actual SOS and EOS IMG names used will then be ***PrefixDateformatSuffixSs*** and ***PrefixDateformatSuffixEs***, with *s* the season number (1 or 2), for each target year this year formatted according to the specified ***Dateformat.***
- the season number (s = 1 or 2)
- the location and base name of the progress IMGs

The progress IMGs contain the relative progress in percentages for the specified season up to the IMG date. They are named as ***BasenameXXXX_YYYYTT_sS.img/hdr***, with

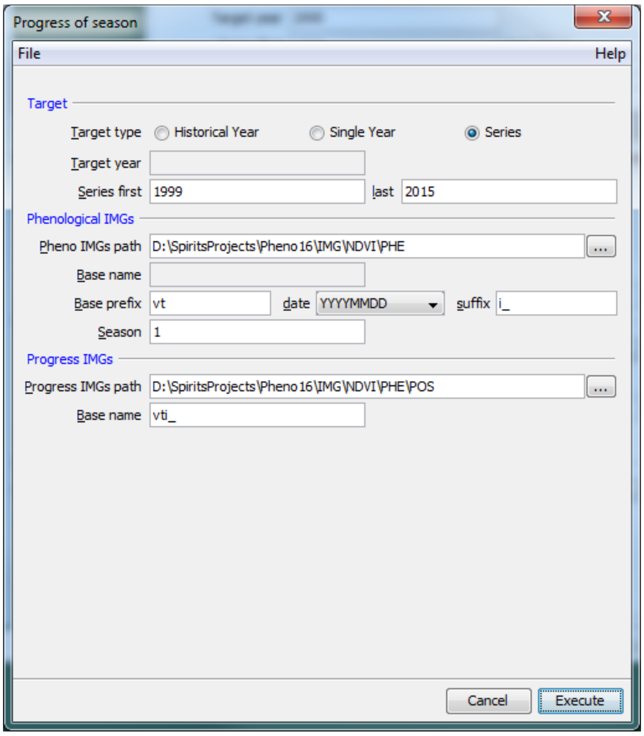
- *Basename* as specified;
- *XXXX* the seasons target year;
- *YYYYTT* year and dekad (1..36) of the IMG (from year XXXX-1 dekad 19, till year XXXX+1 decad 18)
- *S* the season number

Remark: the target year specified by the user must match the date found in the HDRs of the SOS/EOS files. In case of LTA, there should be no date entry in the SOS/EOS HDR files. In that case, the (fictive) year 1962 is used for the output filenames.

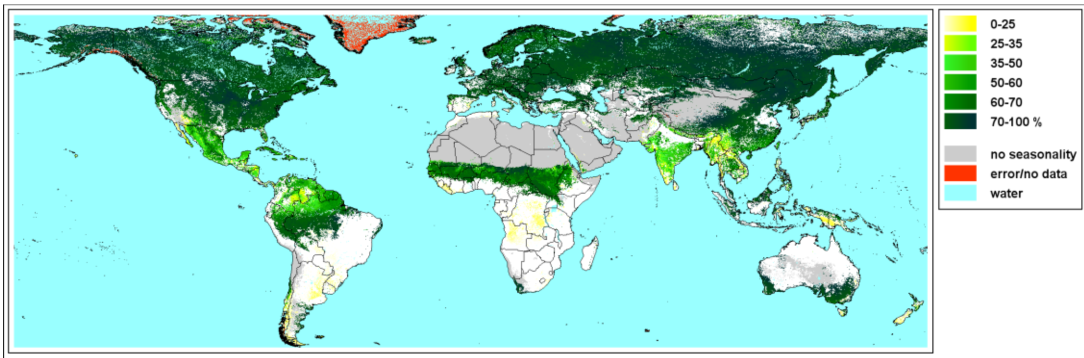
Tool



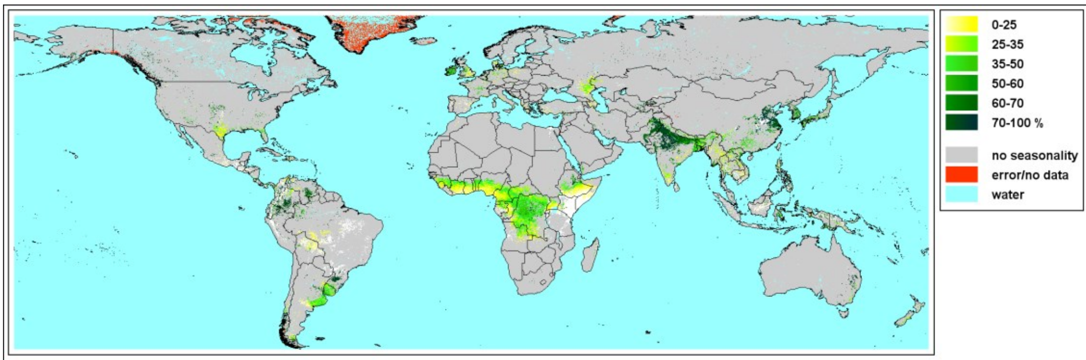
Progress of season Tool example using Historical (LTA) pheno files



Progress of season Tool example for a series of pheno files

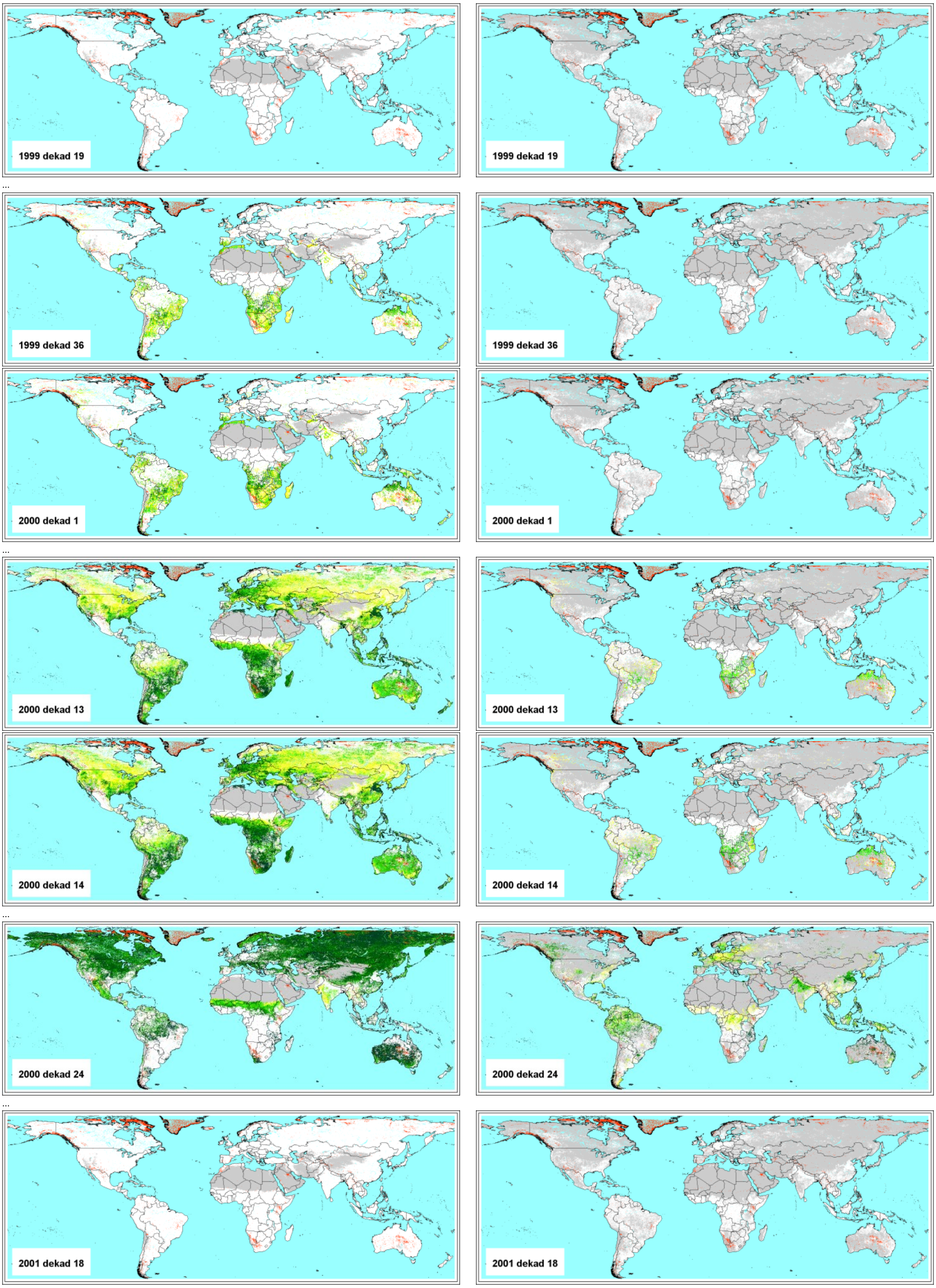


Progress of first season (%) - October dekad 1



Progress of second season (%) - October dekad 1

example: progress of season - LTA October first dekad.



example progress of season- target year 2000

first season second season

3.22. Cumulate over season

Goal

Temporal integration over the growing season: compute an image with per pixel the temporal mean (or sum) of any VI-series from the SOS to the specified output date (or EOS).

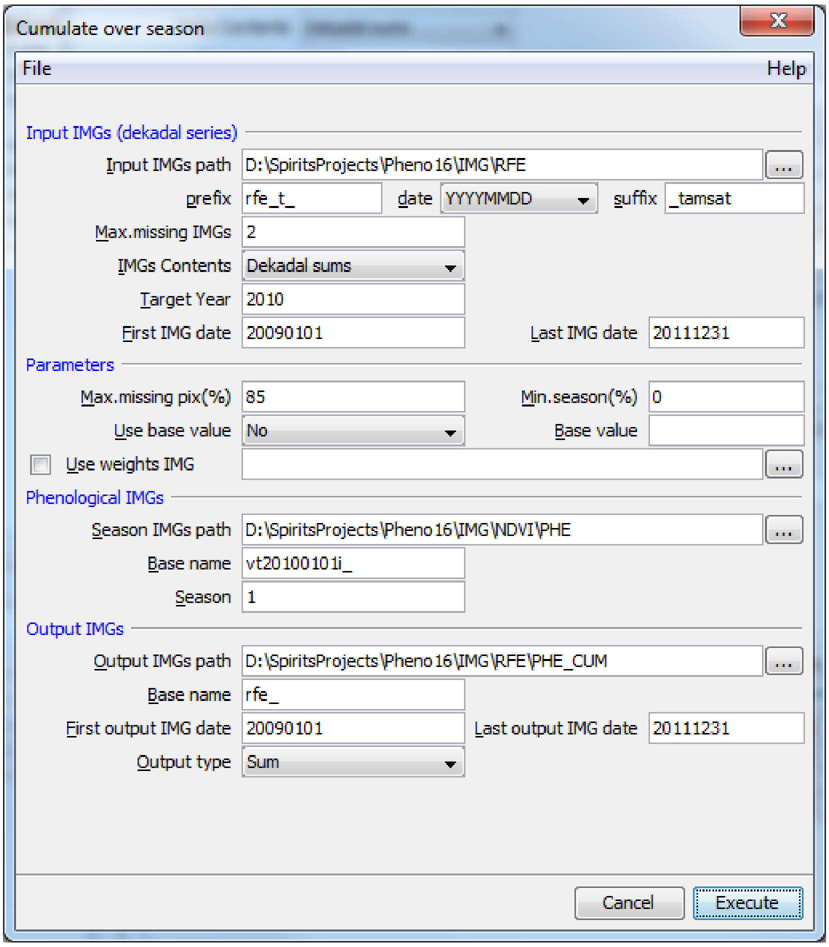
Parameters

- the location and filename structure of the input IMGs.
- the maximum number of consecutively missing IMGs in the series;
- the content type of the input IMGs, this can be:
  - daily values
  - dekadal sums
- the target (or 'central') year of the season;
- the date of the first input IMG. This date must belong to the (civil) year preceding the target year;
- the date of the last input IMG. The last date to be included in the calculated sum/mean. This date must belong to the target year, or its successor.
- the maximum percentage of missing or flagged pixel values in the series. Pixels above this maximum will be flagged;
- the minimum percentage of the full season length to be covered by the series. Pixels below this minimum will be flagged;
- whether to use a (physical) base value (***YBase***) and how to use it:
  - no base value is used;
  - using a base value as a lower limit: all values below the base value are reset to this value;
  - using a base value to calculate differences (***Ydiff = Y-Ybase***, or 0 in case Y<Ybase). This is typically used for temperature sums);
- in case a base value is used its actual (physical) value must be specified;
- whether to apply a crop coefficients IMG (Kc classification);
- the location and base name of the phenological (SOS/MOS and EOS) IMGs (as created by the "detect seasons" tool) for the specified target year (or LTA);
- the season number (s = 1 or 2)
- the location and base name of the output IMG(s);
- the (first) output IMG date;
- optionally the last output IMG date. In this case a series of output IMGs will be created for each dekad between the first and last date. These dates are limited as follows:
  First input IMG date <= First output IMG date <= Last output IMG date <= Last input IMG date;
- the output IMGs type (mean value or cumulative sum).

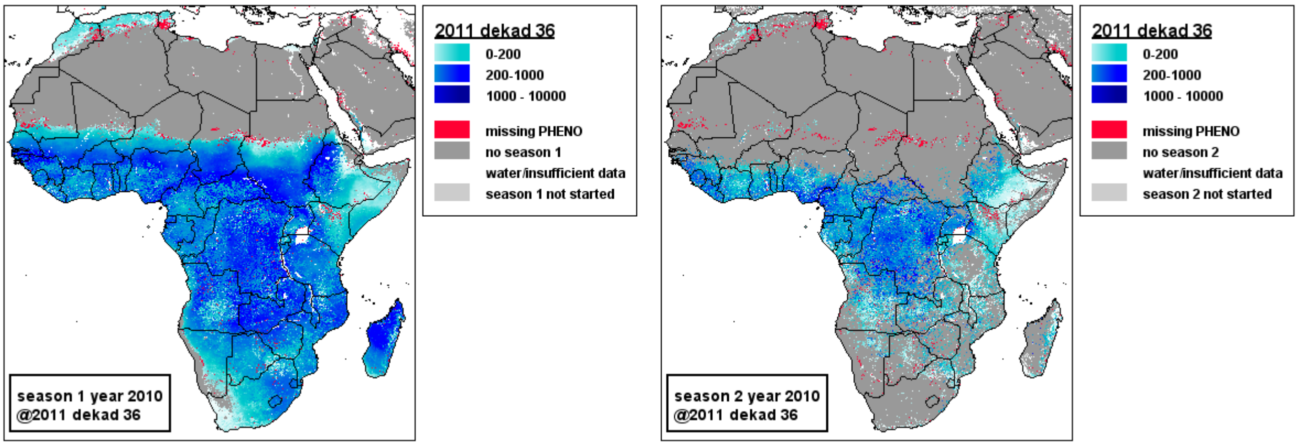
The output IMG(s) contain the sum or mean values of the input IMGs from the start of the season (or the first input IMG if that comes later) till the output date (or till the end of the season if that comes sooner). They are named as ***BasenameXXXX_YYYYTT_sS.img/hdr***, with

- *Basename* as specified;
- *XXXX* the seasons target year;
- *YYYYTT* year and dekad (1..36) of the output IMG (minimum: year XXXX-1 dekad 1, maximum: year XXXX+1 dekad 36)
- *S* the season number

Tool



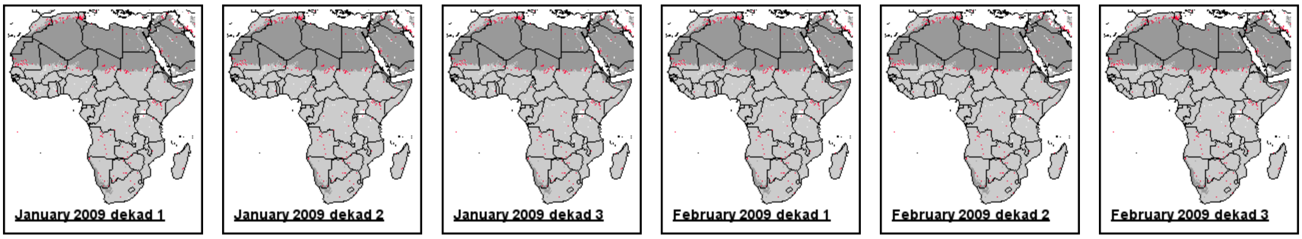
Cumulate over season Tool example



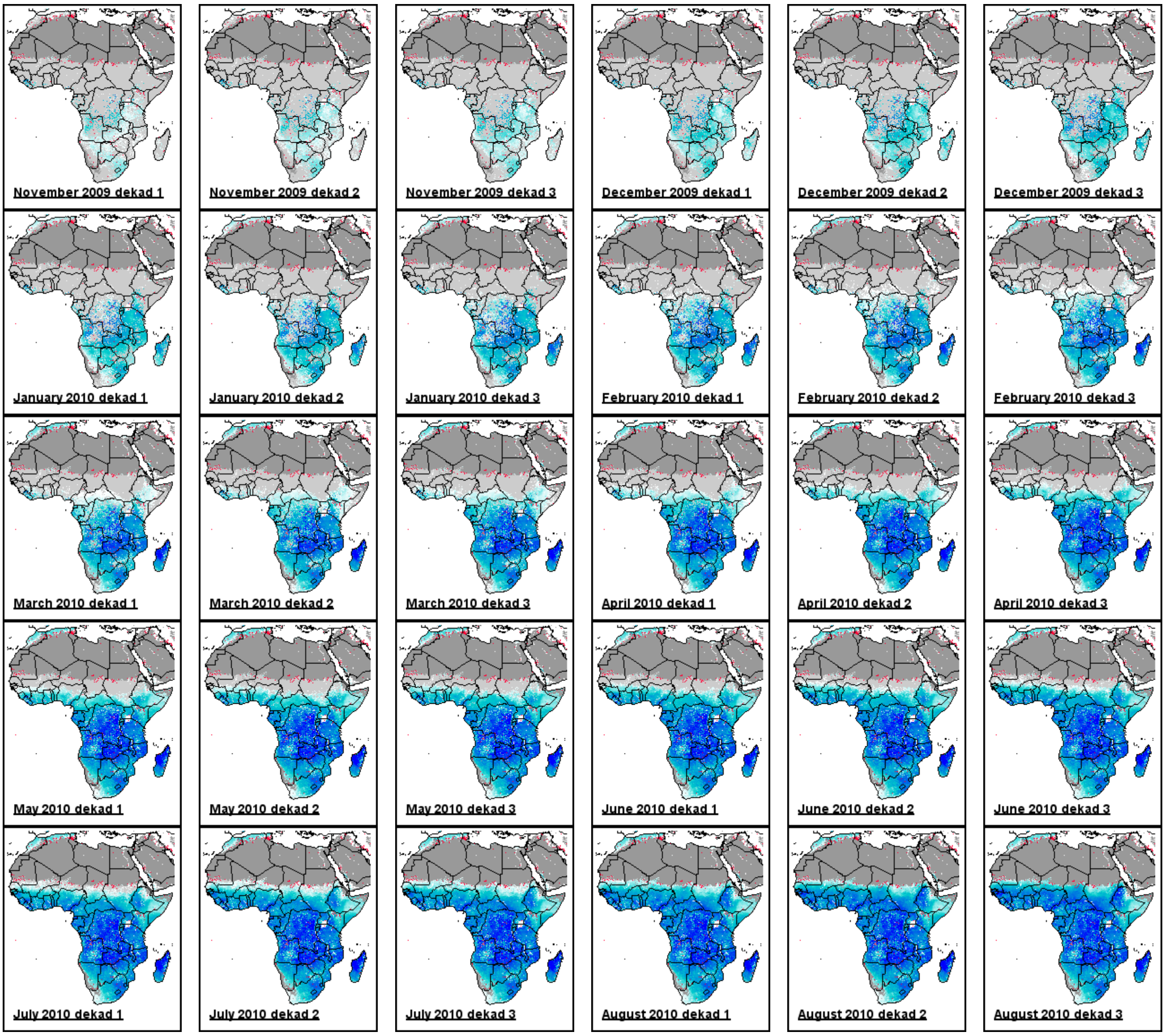
Cumulate over full season - RFE

season 1 of 2010 (sum 20090101 - 20111231) season 2 of 2010 (sum 20090101 - 20111231)

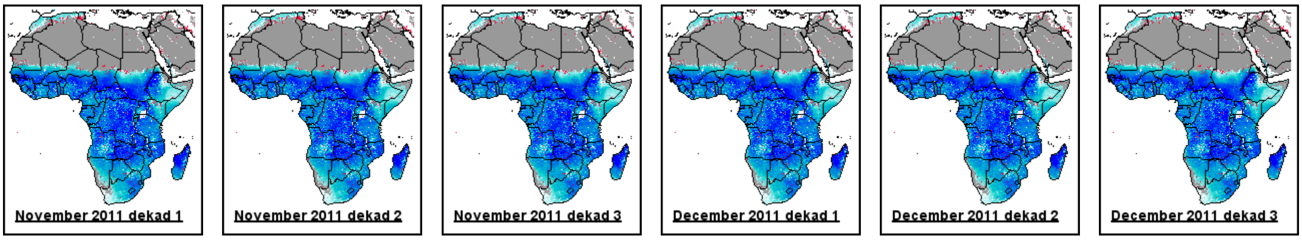
example: RFE dekadel cumulates, from January 2009 till December 2011, for season 1 of 2010.



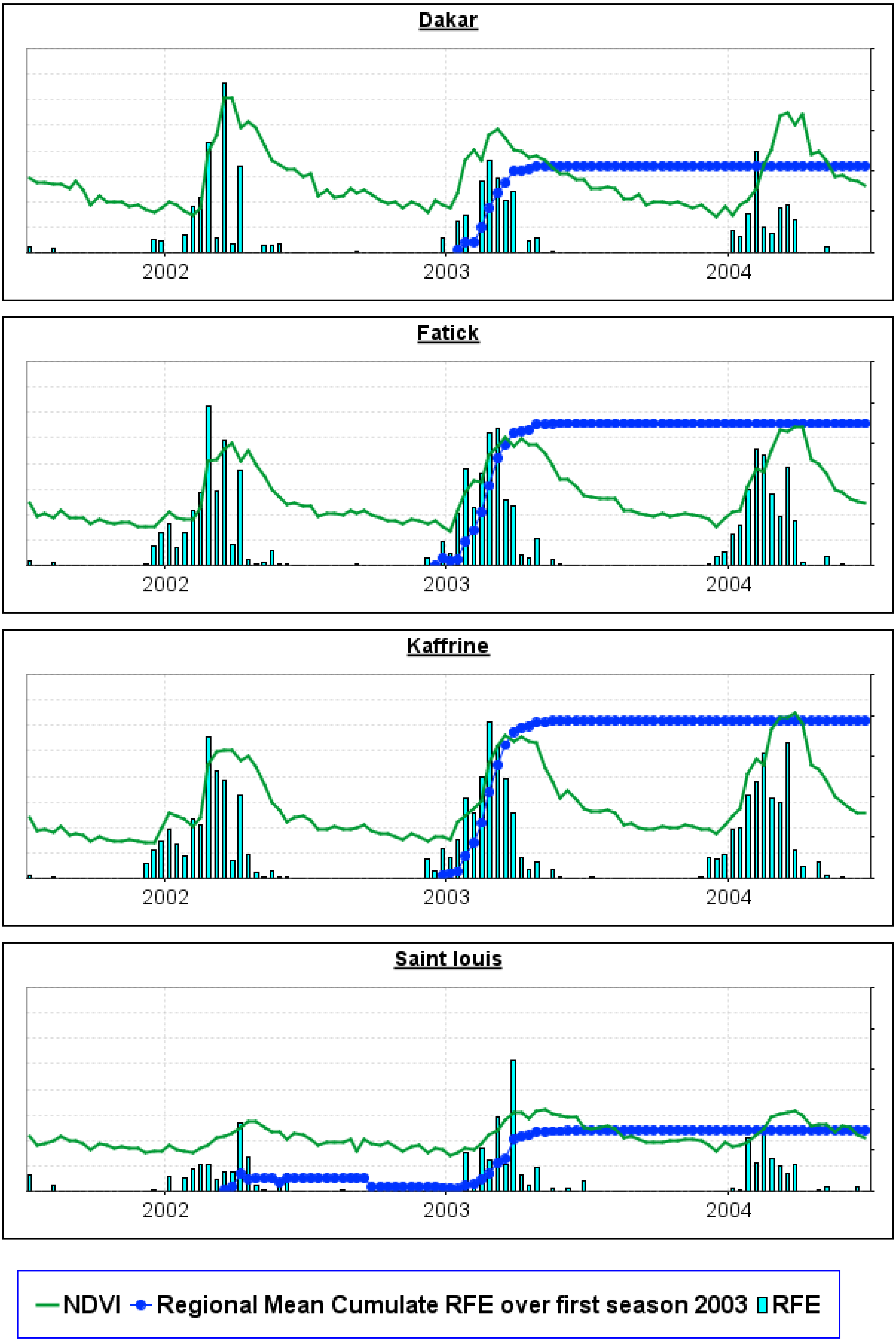
...



...



example: regional mean of RFE dekadel cumulates, from January 2002 till December 2004, for season 1 of 2003, for cropland in Senegal regions.



3.23. Cumulate full season

Goal

Temporal integration over growing seasons: compute yearly series of images with per pixel the temporal mean (or sum) of any VI-series from the SOS to EOS.

Parameters

- the location and filename structure of the input IMGs.
- the maximum number of consecutively missing IMGs in the series;
- the content type of the input IMGs, this can be:
  - daily values
  - dekadal sums
- the date of the first input IMG. This date must precede the first target year;
- the date of the last input IMG. This date must be in or beyond the last target year.
- the maximum percentage of missing or flagged pixel values in the series. Pixels above this maximum will be flagged;
- the minimum percentage of the full season length to be covered by the series. Pixels below this minimum will be flagged;
- whether to use a (physical) base value (***YBase***) and how to use it:
  - no base value is used;
  - using a base value as a lower limit: all values below the base value are reset to this value;
  - using a base value to calculate differences (***Ydiff = Y-Ybase***, or 0 in case Y<Ybase). This is typically used for temperature sums);
- in case a base value is used its actual (physical) value must be specified;
- whether to apply a crop coefficients IMG (Kc classification);
- the location and filename structure (or base name) of the phenological (SOS/MOS and EOS) IMGs (as created by the "detect seasons" tool) for the specified target years (or LTA);
- the season number (s = 1 or 2)
- the location and the base filename structure of the output IMG(s);
- the first target year;
- optionally the last target year;
- the output IMGs type (mean value or cumulative sum).

The output IMG(s) contain the sum or mean values of the input IMGs from the start of the season (or the first input IMG if that comes later) till the end of the season (or last input IMG if that comes sooner). They are named as ***PrefixDateformatSuffixSeasonno.img/hdr***, with

- *Prefix and Suffix* as specified;
- *Dateformat the target year, formatted as specified*;
- *Seasonno* the season number

For each target year in the series, the "cumulate over season" is executed:

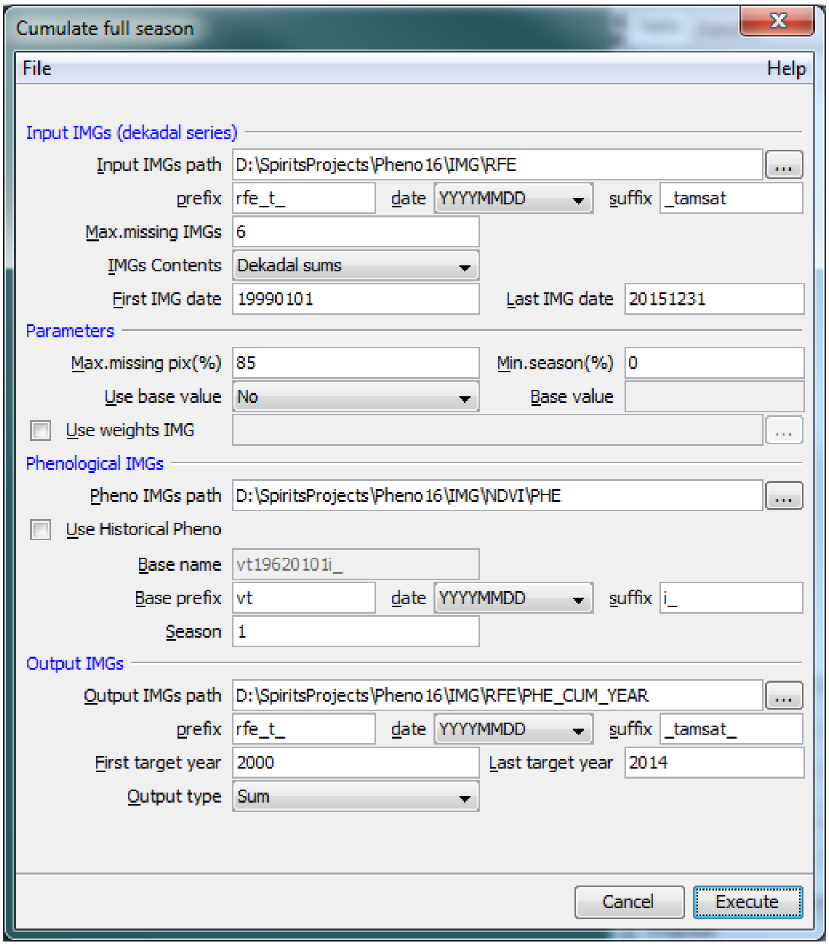
- with an input IMGs series from January 1st of the year preceding the target year, till December 31 of the year following the target year;
- with an output IMG date on December 31 of the year following the target year.

Hence this resulting IMG contains the cumulative value of the index specified, for the complete season.

The header of this output IMG is then modified:
- the periodicity is set to annual ("DAYS=360")
- the date is set to the target date ("DATE=YYYY0101" with YYYY the target date).

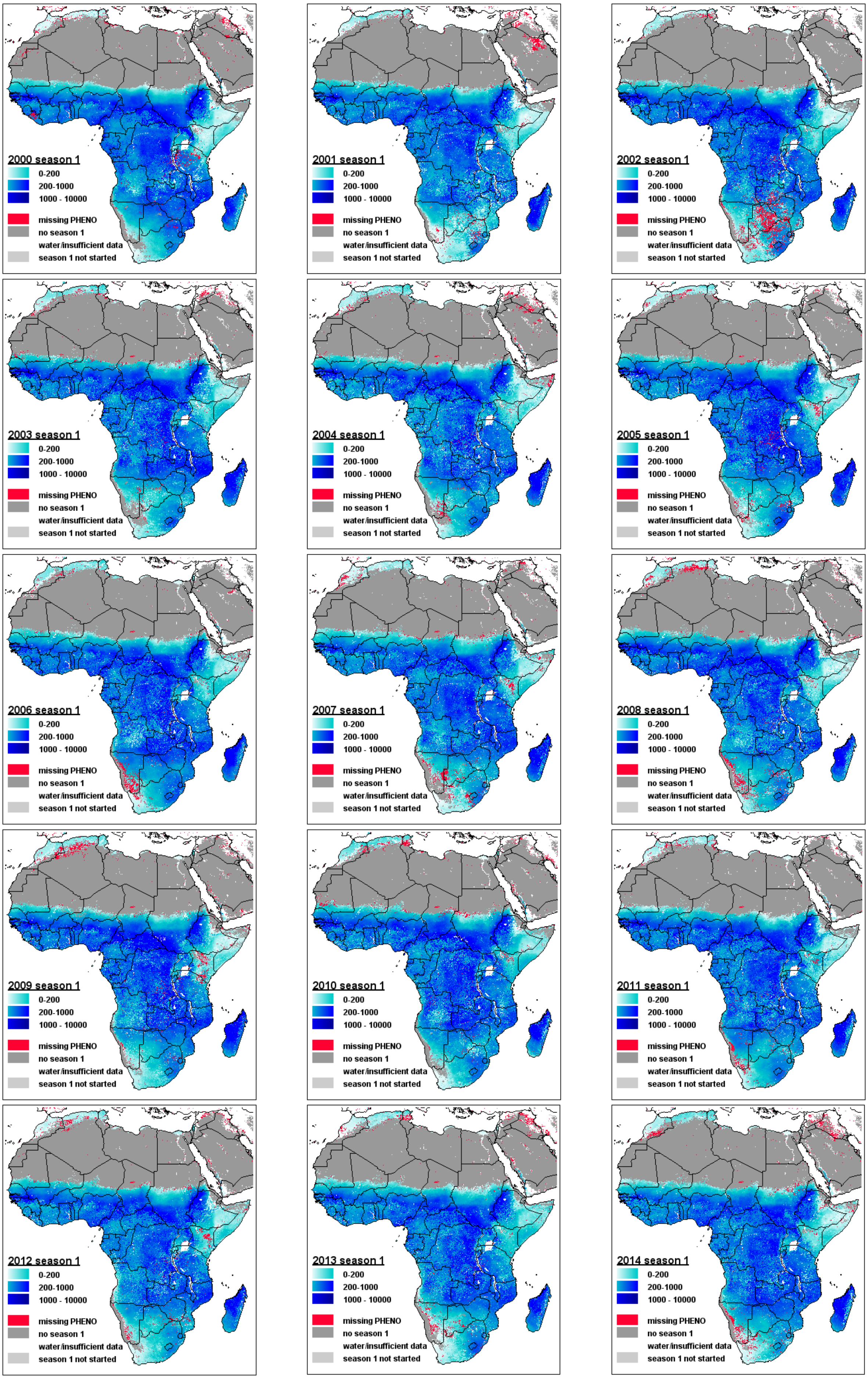
Hence the resulting IMGs can be used as a yearly time series of IMGs, containing the cumulative values of the index specified, for the complete season of the IMGs year.

Tool



Cumulate full season Tool example

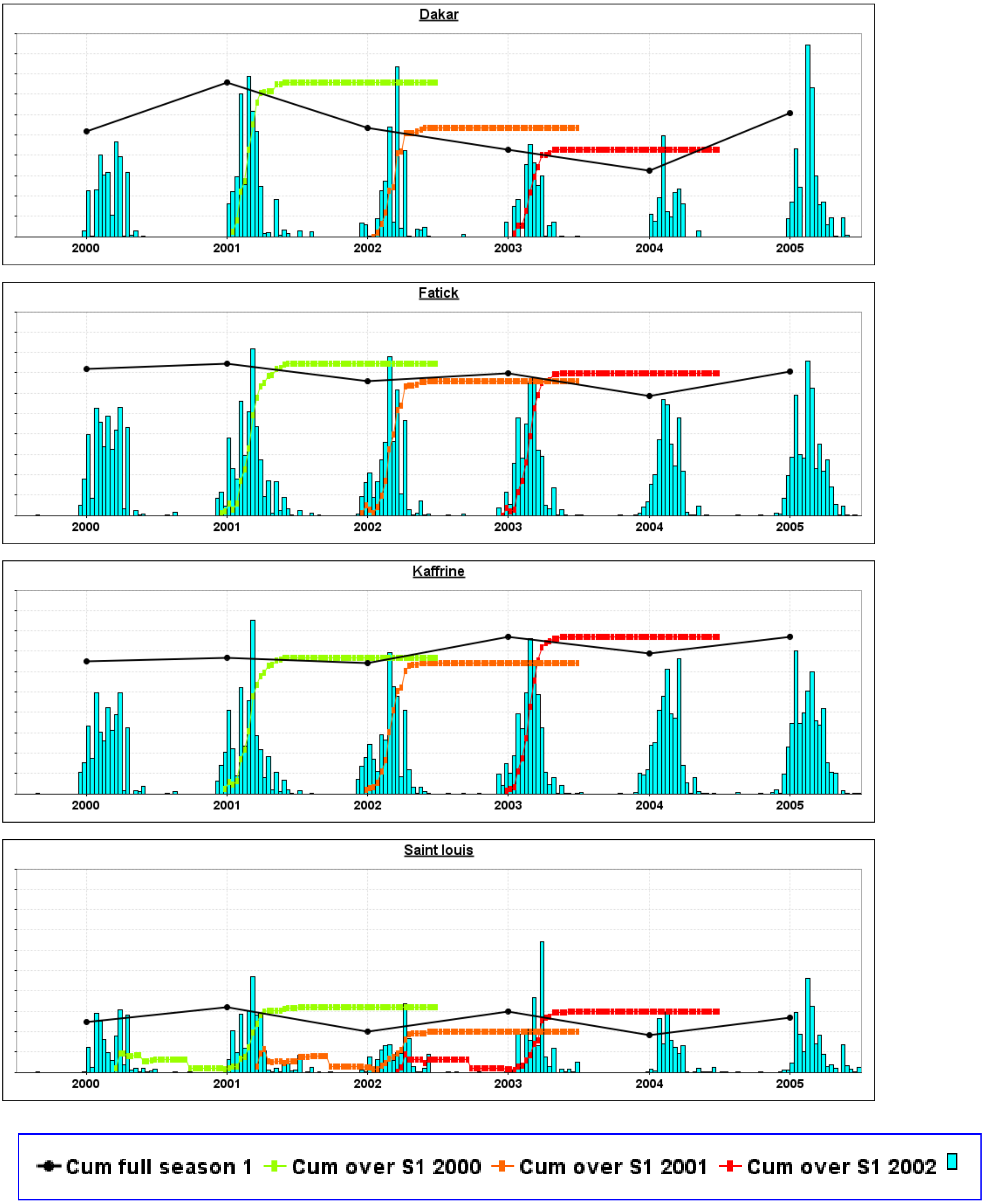
example: RFE full season cumulates.



Cumulate full season Tool example: RFE series first season of years 2000 - 2014

example: regional mean of RFE full season cumulates for season 1, for cropland in Senegal regions.

To illustrate the difference between the "cumulate over season" and "cumulate full season" outputs, the dekadal "cumulate over season" results are added for the years 2000 (green), 2001 (orange) and 2002 (red). The "full season" values (black) are identical to the values obtained at the saturation point of the "cumulate over season" values.



3.24. Crop coefficients IMG

This tool enables the creation of custom crop coefficient (Kc) files, which can be used by the "cumulate over season" and "cumulate full season" tools allowing them to weight pixel profiles.

Parameters

Kc types can be defined and assigned to regions and/or classes.

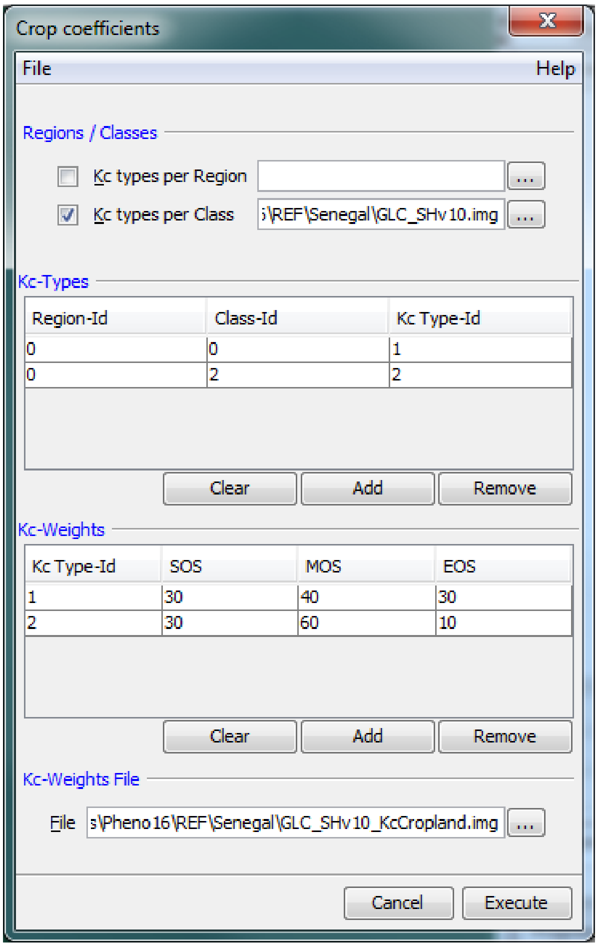
- whether or not Kc-types will be specified per region, and if so, a regions IMG;
- whether or not Kc-types will be specified per class, and if so, a land use IMG;
- per region x class combination a reference Kc-type Id;

Following rules apply:

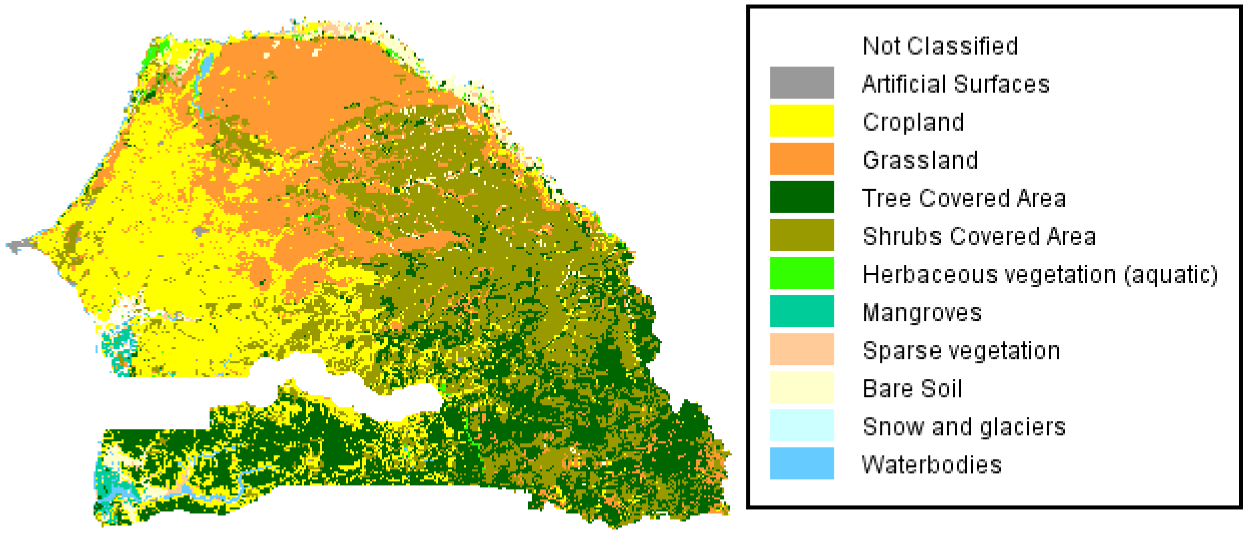
- in case "Kc-types per region" is not selected, the values of column 1 (REGION_ID) are skipped.
- in case " Kc-types per class" is not selected, the values of column 2 (CLASS_ID) are skipped.
- in case both are selected:
    - if Region_ID=0 & Class_ID=K : Class K follows the Kc-type in all regions;
    - if Region_ID=R & Class_ID=0 : Region R follows the Kc-type for all classes;
    - if Region_ID=0 & Class_ID=0 : All regions and classes follow the Kc-type;
    - Later records with same Region_ID/Class_ID overwrite earlier records;
    - Non-mentioned (Region x Class)-combinations get the default Kc-type 0 (weights 100,100,100).

The Kc types are then specified by means of weights at the three phenological SOS, MOS and EOS stages. Kc-values at intermediate dates will be linearly interpolated from the neighbouring values.

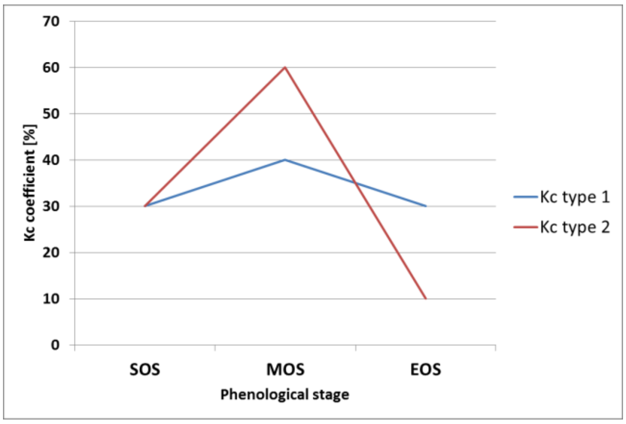
Tool



Crop coefficients Tool example

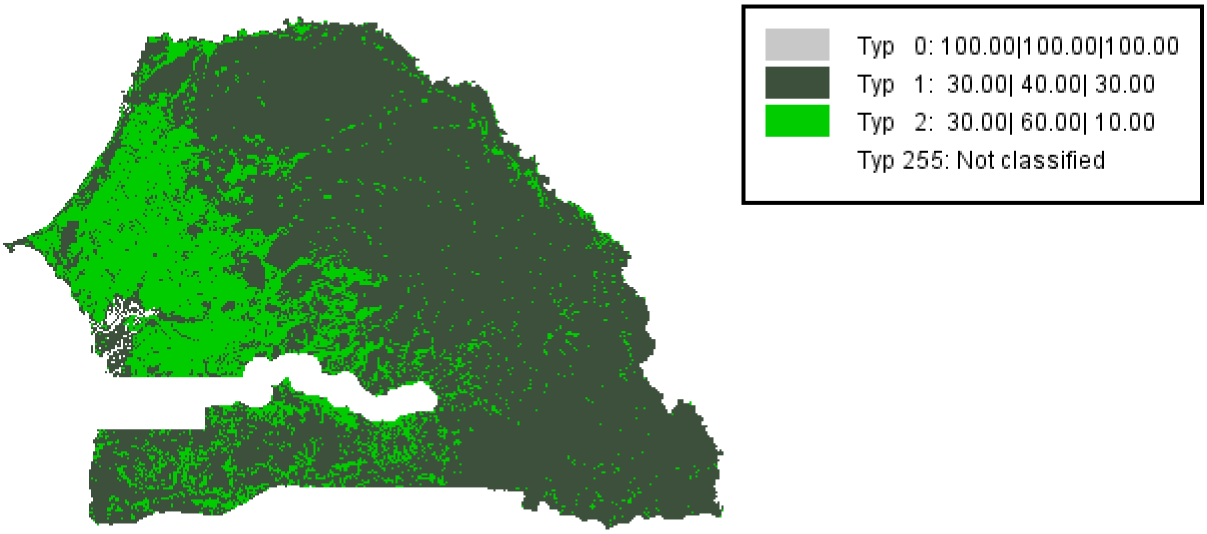


example: Senegal land use classification.



type 1 :30,40,30

type 2 : 30,60,10 (emphasizing the vegetative stage)



example: crop coefficients (Kc) IMG

Type 2 -> cropland
Type 1 -> other

3.25. Long Term Average

Goal

Compute historical IMGs (Long Term Averages) from a multi-annual IMGs set.

Parameters

- Specification of the input IMGs:
  - input IMGs directory;
  - input IMGs file names specification (PiDiSi);
  - input IMGs periodicity;
- First and Last year to be considered in the computations;
- Specification of the output IMGs:
  - output IMG directory;
  - output IMG file names specification (PoDoSo);
- Start and End period to compute output IMGs for ;
- Selection of the output IMG types to be computed.

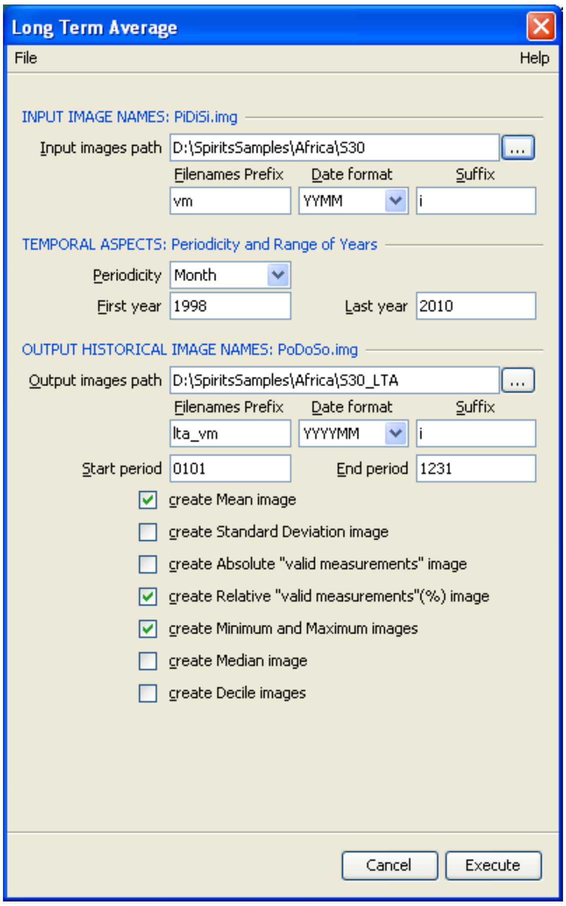
The output IMG types are coded by using the year in their file names, (example: vm**1962**01i.img).

The year codes are used as follows:

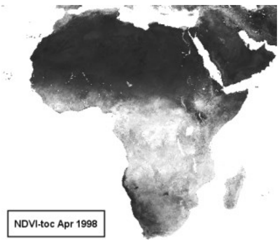
year	output IMG content	year	output IMG content
1950	minimum (0% decile)	1951	10% decile
1960	maximum (100% decile)	1952	20% decile
1961	Number of valid measurements	1953	30% decile
1962	Mean	1954	40% decile
1963	Standard Deviation	1955	50% decile
1964	% of valid measurements	1956	60% decile
		1957	70% decile
		1958	80% decile
		1959	90% decile

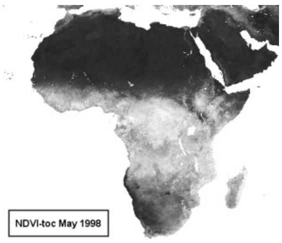
Remark: Other tools (e.g. Smooth, Difference) can use historical IMGs as input files. They identify the historical IMGs via this year code.

Tool

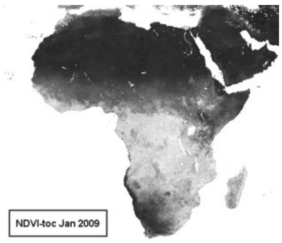


Long Term Average Tool example

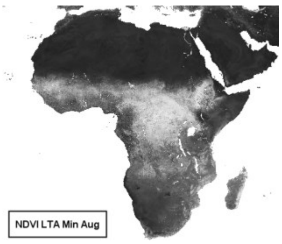




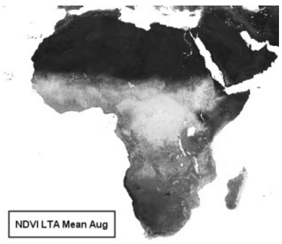
…



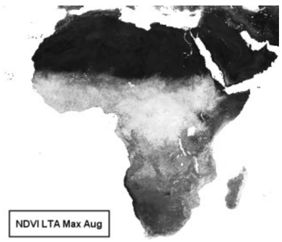
input IMG series(April 1998 - November 2009)



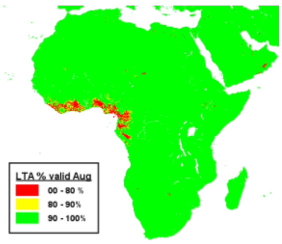
LTA Minimum



LTA-Mean



LTA-Maximum



LTA-valid measurements

3.26. Difference

Goal

Compute difference images for anomaly assessment. Remark: only for byte-type IMGs.

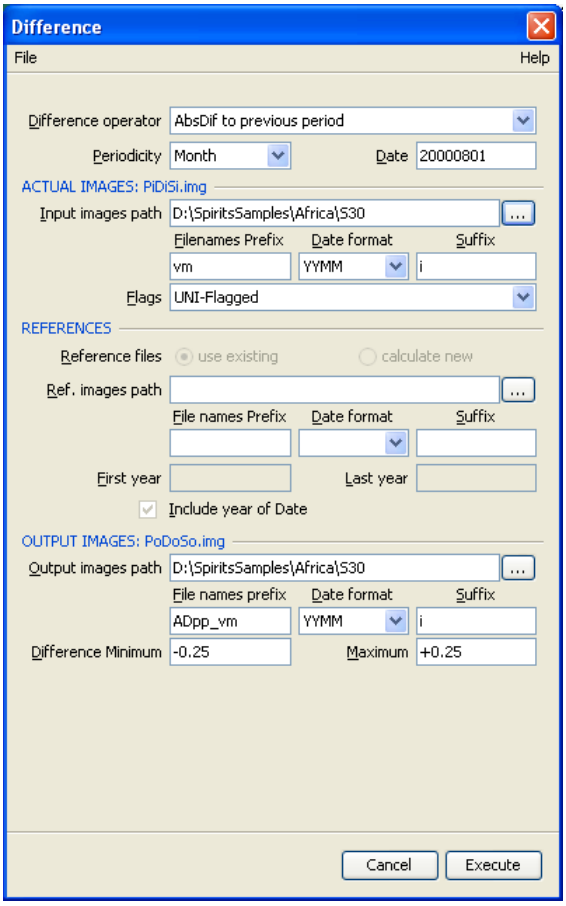
Parameters

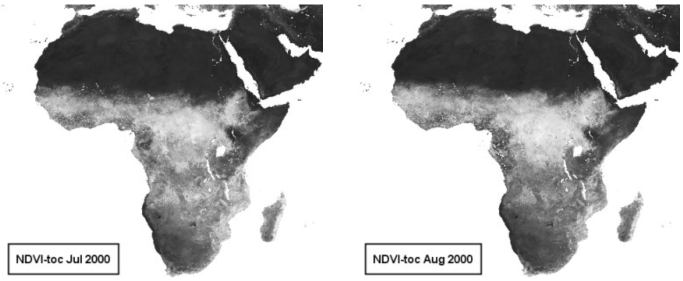
- Difference operator. Available operators are:

AbsDif to prev period	ADpp(y,p)	=	X(y,p)	-	X(y, p-1)
RelDif to prev period	RDpp(y,p)	=	[X(y,p)	-	X(y, p-1)]/X(y, p-1)
AbsDif to prev year	ADpy(y,p)	=	X(y,p)	-	X(y-1, p)
RelDif to prev year	RDpy(y,p)	=	[X(y,p)	-	X(y-1, p)]/X(y-1, p)
AbsDif to hist median	ADhm(y,p)	=	X(y,p)	-	MEDIAN(p)
RelDif to hist median	RDhm(y,p)	=	[X(y,p)	-	MEDIAN(p)]/MEDIAN(p)
AbsDif to hist average	ADha(y,p)	=	X(y,p)	-	MEAN(p)
RelDif to hist average	RDha(y,p)	=	[X(y,p)	-	MEAN(p)]/MEAN(p)
Standardized diff.	SDh(y,p)	=	[X(y,p)	-	MEAN(p)] / StDEV(p)
Relative range	RRh(y,p)	=	[X(y,p)	-	MIN(p)] / [MAX(p) - MIN(p)]
Historical probability	HPh(y,p)	=	Prob. of X(y,p) in hist. distribution		
Classified hist.prob.	CPh(y,p)	=	HPh(y,p) in 11 classes: 0-10%...90-100%		
Historical rank	HRh(y,p)	=	Rank of X(y,p) in hist. distribution		
Classified hist.rank	CRh(y,p)	=	HRh(y,p) in 11 classes: 0-10%...90-100%		

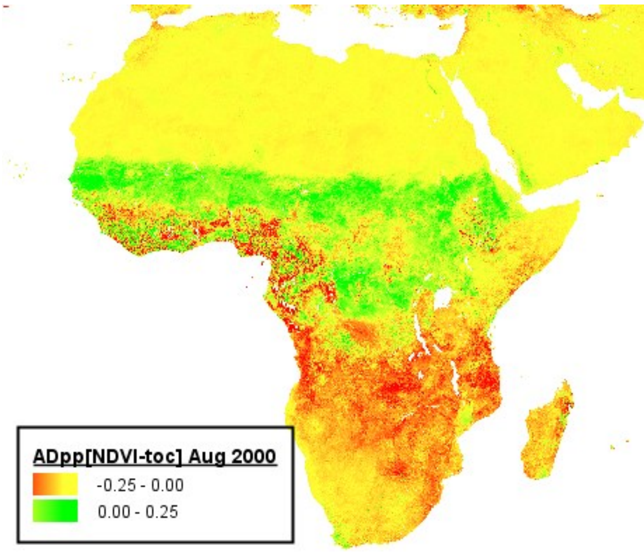
- Periodicity of the input IMGs;
- Date of the output IMG;
- Specification of the input IMGs:
  - input IMGs directory;
  - input IMGs file names specification (PiDiSi);
  - input IMGs flags type (if input IMGs are UNI-Flagged, the output IMG will also be UNI-Flagged, otherwise a single flag will be used in the output IMG);
- Specification of the reference files:
  - mandatory for operations using historical IMGs;
  - in case these files are available (calculated by the Long Term Average tool), their specification is needed (directory and file names specification PrDrSr);
  - in case these are to be calculated intermediately, the parameters for the calculation are required (First year, Last year and whether to include the year of the output IMG date in the calculation);
- Specification of the output IMG:
  - output IMG directory;
  - output IMG file name specification (PoDoSo);
- Minimum and maximum values of the difference result (in physical units). Only in case an Absolute Difference operator is selected. In the output IMG, this minimum difference value will be rescaled to digital value 0, the maximum difference value to 250.

Tool





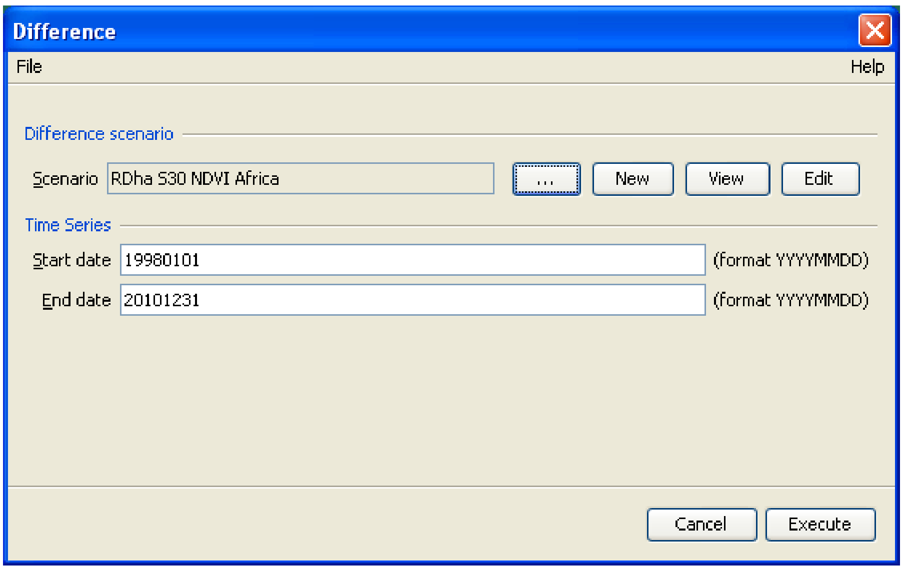
input IMG



Difference Tool example

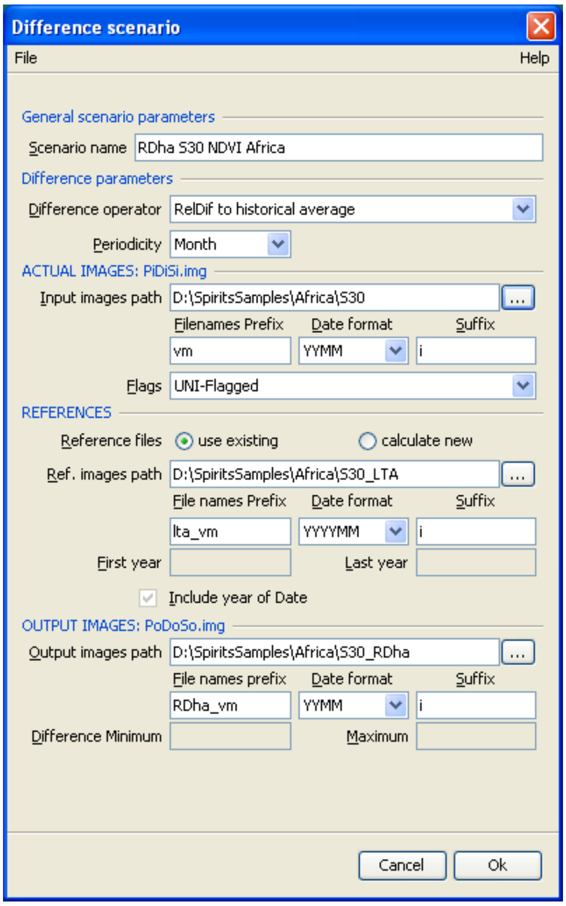
output IMG

Time Series

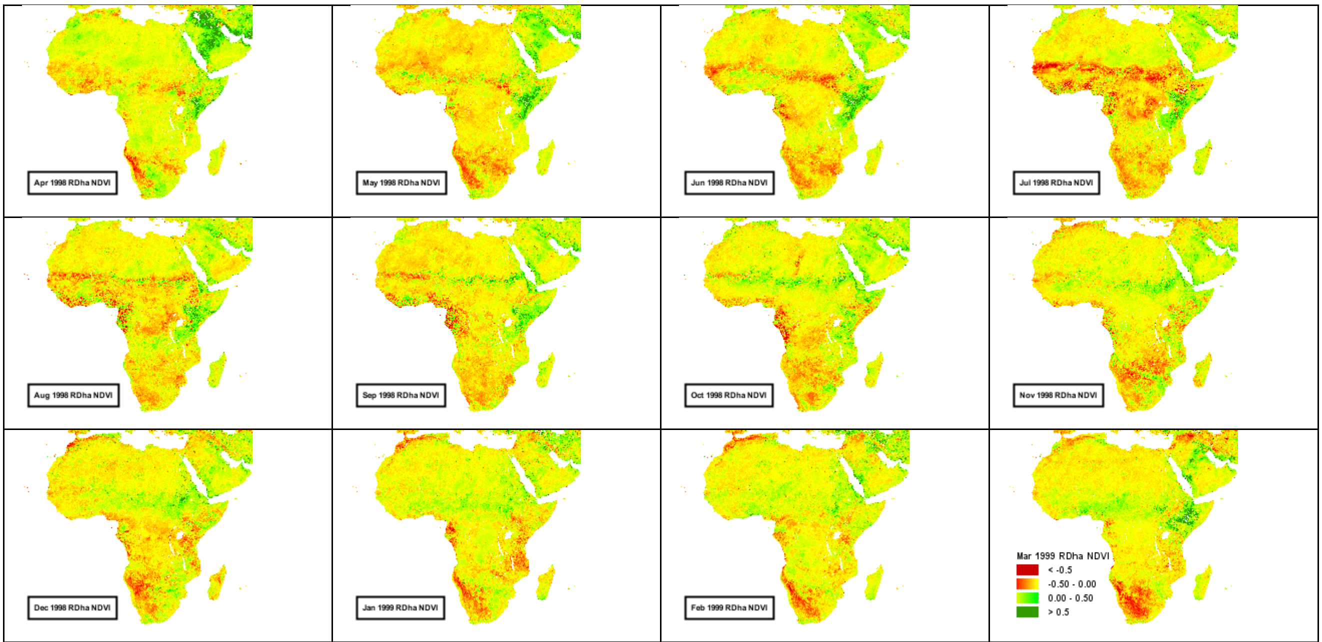


Difference Time Series example

Scenario



Difference Scenario example



Difference Time Series example: RDha NDVI from April 1998 - March 1999

3.27. Standardized Precipitation Index

Goal

Compute Standardized Precipitation Index (SPI) images for anomaly assessment, from a multi-annual IMGs set. Typically used for precipitation IMGs.

Parameters

- specification of the input IMGs:
  - location, filename structure and periodicity of the input IMGs;
  - first and last year to be considered in the computations;
- specification of the output target(s):
  - whether IMGs are to be computed for one specific period, or for each period in the target year(s);
  - whether IMGs are to be computed for one specific target year, or for a series of years;
  - in case a specific period/year is selected, this period/year must be specified;
  - in case a series of years is selected the first and last year of the series must be specified.
- the accumulation length (periods):

During the first stage of the algorithm, the calibration phase, the distribution of the input values is approximated via a gamma function. To determine this gamma function:

  - In case accumulation length 1 (default) is selected, only values for target period P are considered;
  - In case an accumulation length l > 1 is selected, values from the interval [P, P+L-1] are averaged;

- in case an accumulation length >1 is selected: the minimum % of 'good' values in the accumulation interval (below this minimum the concerned output pixel will be flagged);
- the minimum (physical) input value Ymin. Input values below this minimum are considered as 'missing'. If Ymin is not specified, the default value Ylo = Vint + Vslo * Vlo (from the HDR) will be used;
- in case Ymin < 0: the method to handle negative input values:
  - Reset: set to zero;
  - Flag: skip and label as 'missing values';
  - Shift: all values are shifted with offset = -Ymin.
- the minimum number of 'good' observations (years) per period to allow calibration. At least three values are required to determine a gamma function. For pixels with less 'good' observations the concerned output pixel will be flagged.
- whether or not to include the observations of the target year itself in the calibration;
- the location and filename structure of the output IMG(s):

In the second stage of the algorithm, the Extrapolation phase:

  - for each annual Y-value in the input series, its cumulative probability p(Y) is defined (using its gamma-approximation), which situates this Y-value within its historical range. For instance p(Y)=0%, 50% and 100% respectively correspond with the minimum, median and maximum values, ever observed;
  - this p(Y) is considered as the cumulative probability holding for a normal Gaussian distribution p(z) with mean =0 and standard deviation =1. By means of the inverse Gauss function, the cumulative p(Y) or p(z) is translated into a z-value. This standard z-score is the final SPI.

- in case IMGs are to be computed for a specific period in a specific year: a base name for the output file(s) has to be specified;
- in all other cases: a prefix, date format and suffix have to be specified. These which will be used to generate appropriate base names for the output files;
- selection (at least one) of the outputs to generate:
  - SPI IMG: containing ordinal SPI-values as described above;
  - CLS IMG: a classification file containing classified SPI-values (11 classes-byte type);
  - CUM IMG: containing the cumulative probabilities (byte type);

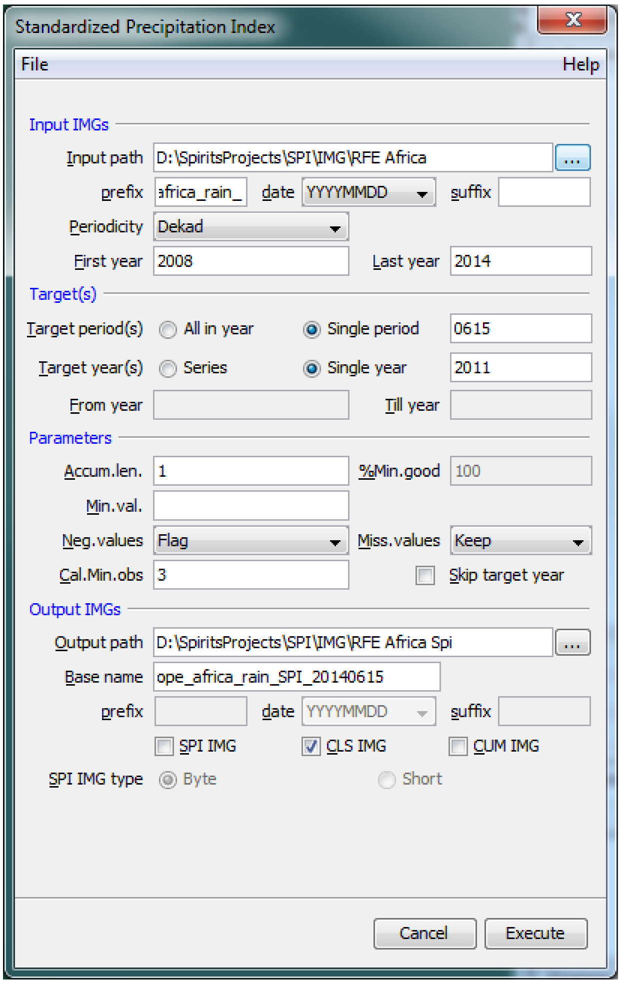
The file name structure of the output images is as follows:

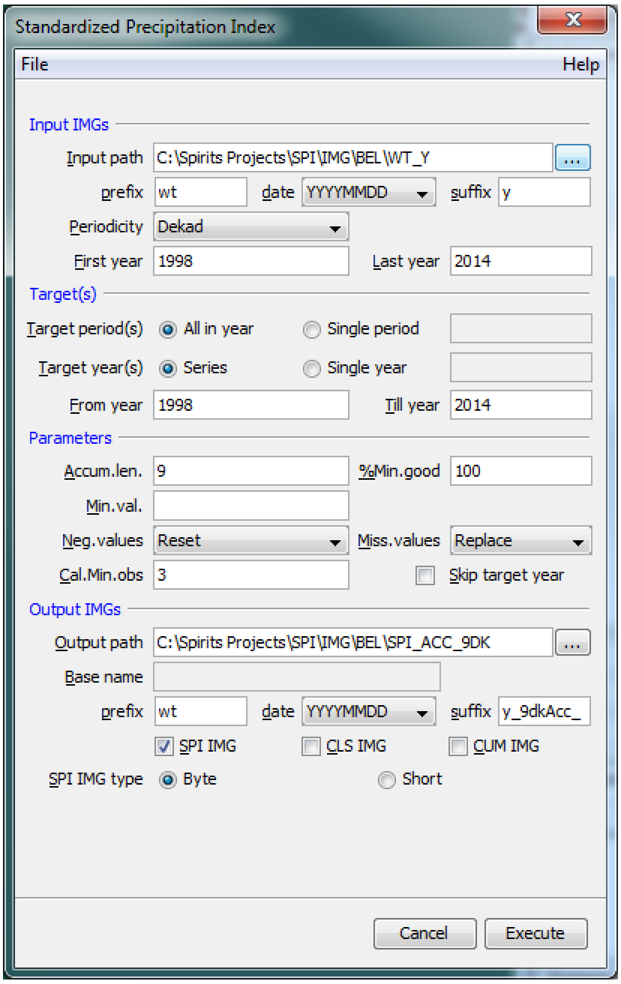
Lo\BoTo.img/hdr			
Lo	output IMGs directory		
Bo	output IMGs base name:		
	as-specified	for outputs for a specific period in a specific year	
	PoDoSo	for outputs over multiple periods and/or multiple years ***Po****: prefix* ***Do:*** *formatted date* ***So****: suffix*	
To	output IMGs suffix:		
	spi	ordinal SPI-values	
	cls	classified SPI-values	
	cum	cumulative probabilities	

- in case the SPI IMG is selected, its output type can be selected:

BYTE	output type:	byte (8 bits, ENVI type 1)	
	scaling:	SPI = -2.5 + 0.020 * digital value (Vint = -2.5, Vslo = 0.020)	
	minimum value:	digital (Vlo) 0	physical (SPI lo) -2.5
	maximum value:	digital (Vhi) 250	physical (SPI hi) -2.5
SHORT	output type:	short (16 bits, ENVI type 2)	
	scaling:	SPI = 0.001 * digital value (Vint = 0.0, Vslo = 0.001)	
	minimum value:	digital (Vlo) -10000	physical (SPI lo) -10.0
	maximum value:	digital (Vhi) +10000	physical (SPI hi) -10.0

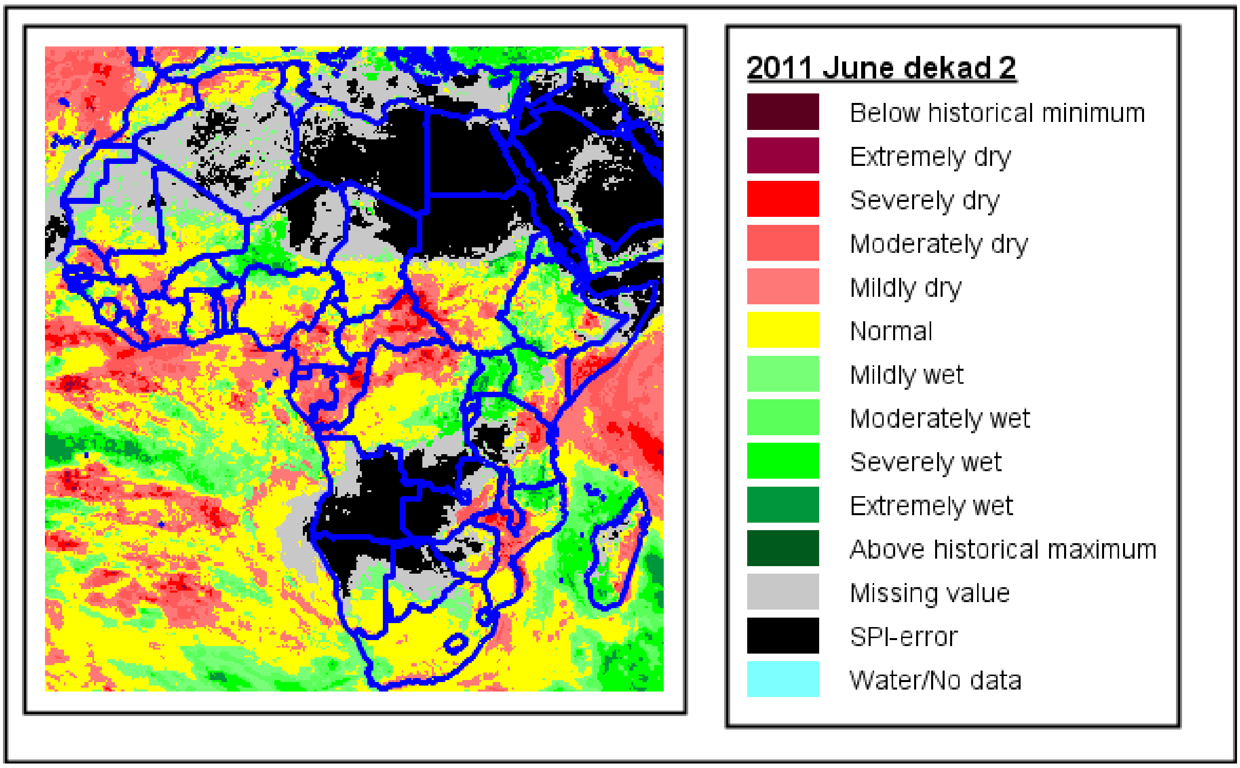
Tool





Standardized Precipitation Index Tool example

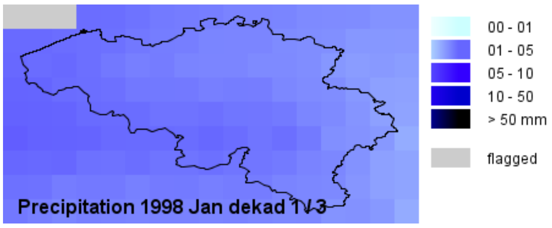
configured for single target period configured for all periods over a series of years

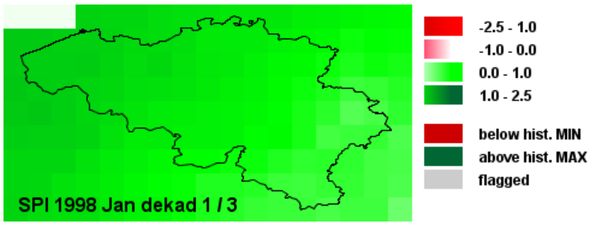


example CLS IMG (classified SPI-values)

Example

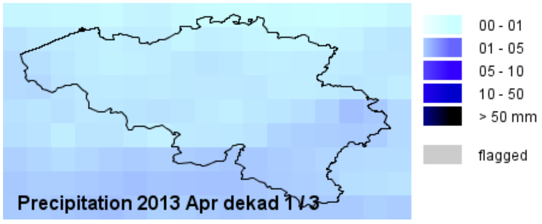
Belgium rainfall 1998 - 2014

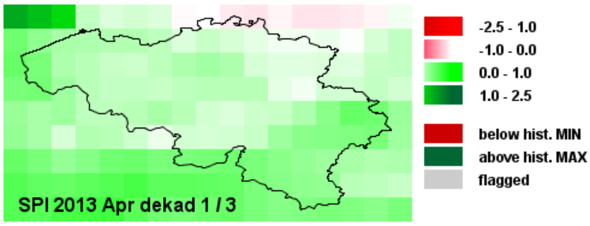


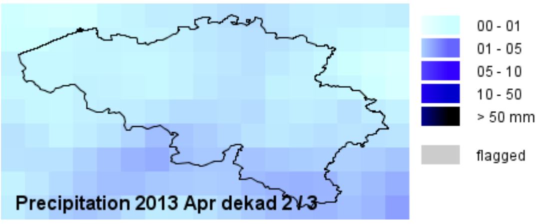


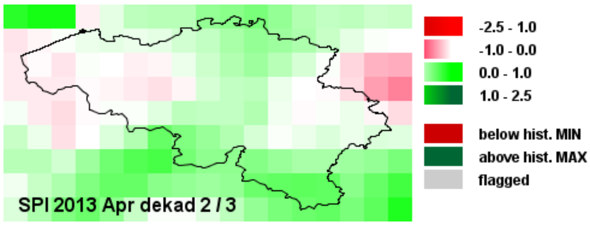
...

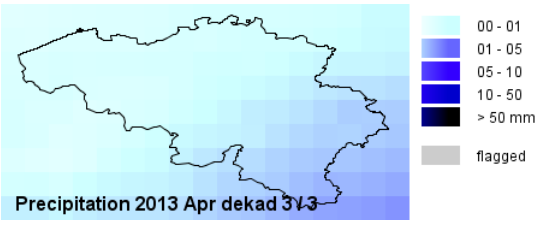
...

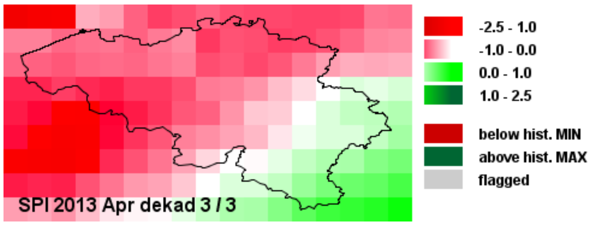


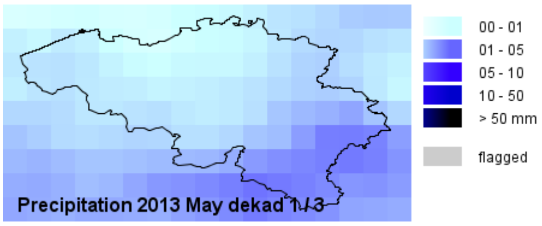


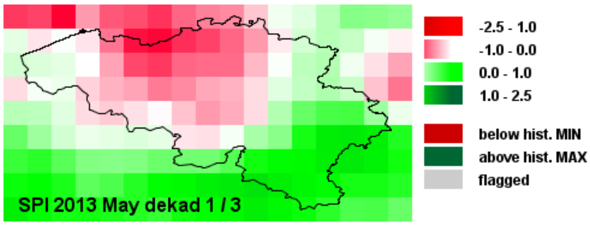


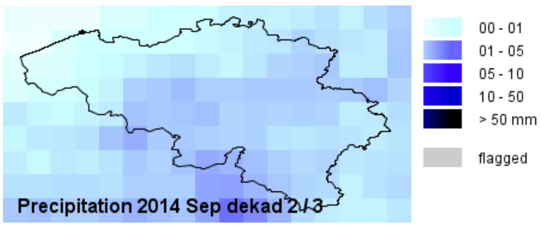


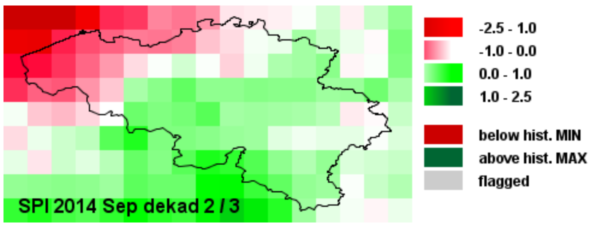












...

...

3.28. Similarity

Goal

Compute an IMG with per pixel the most similar year from a multi-annual IMGs set for a target year.

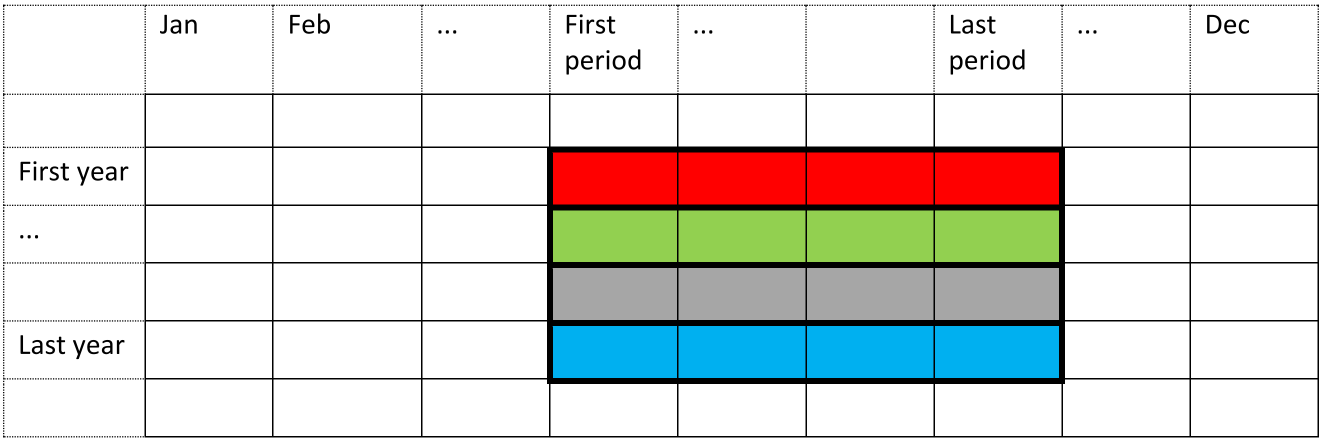
Parameters

- the location and filename structure of the input IMGs;
- the periodicity of the input IMGs;
- the input IMGs flags type;
- the first and last (calendar) year (in the series) to consider;
- the first and last period to consider (specifying the "seasons" from first period till last period);

The seasons which are considered are specified by the first and last period parameters. Depending on the relative position of the first and last period, a season contains the periods of a single calendar year or those of two successive calendar years (including the civil new-year).

The actual periods which are considered are specified by the first/last year and first/last period parameters. These periods must span at least three successive years.

seasons in a single calendar year / seasons including new-year example



- the last season year: the (calendar) year the last season is considered to belong to:
  - in case first period <= last period (seasons belong to single calendar years) this can be the last (calendar) year considered in the series, or the last (calendar) year considered in the series + 1;
  - in case first period > last period (seasons include the civil new-year) this can be the last year, or the last year - 1.
- the target season year(s): the year(s) to compute the most similar year for. Outputs can be created for a 'Single' target year, or a 'Series' of years. The target season year follows the convention specified by the last season year, therefore:
  - in case first period <= last period (seasons belong to single calendar years), and the last season is considered to belong to the last (calendar) year in the series, the target season has to be chosen between the first and the last (calendar) year in the series;
  - in case first period <= last period (seasons belong to single calendar years), and the last season is considered to belong to the last year + 1, the target season has to be chosen between the first year + 1 and the last year + 1;
  - in case first period > last period (seasons include the civil new-year), and the last season is considered to belong to the last year - 1, the target season has to be chosen between the first year and the last year - 1;
  - in case first period > last period (seasons include the civil new-year), and the last season is considered to belong to the last year, the target season has to be chosen between the first year + 1 and the last year;

last season year and target season year example:

	Jan	Feb		**Apr**	...	...	**Oct**	...	Dec
2000									
...									
2009									
2010									

Example: Consider an monthly input IMG series from October (first period) 2000 (first year) till April (last period) 2010 (last year). Seasons start in October, and last till April of the next calendar year.

The last season (from October 2009 till April 2010) can be labelled as 'the 2009 season' or 'the 2010 season'.

In case the last season is chosen to be 'the 2009 season', the target season year has to be chosen between 2000 and 2009.

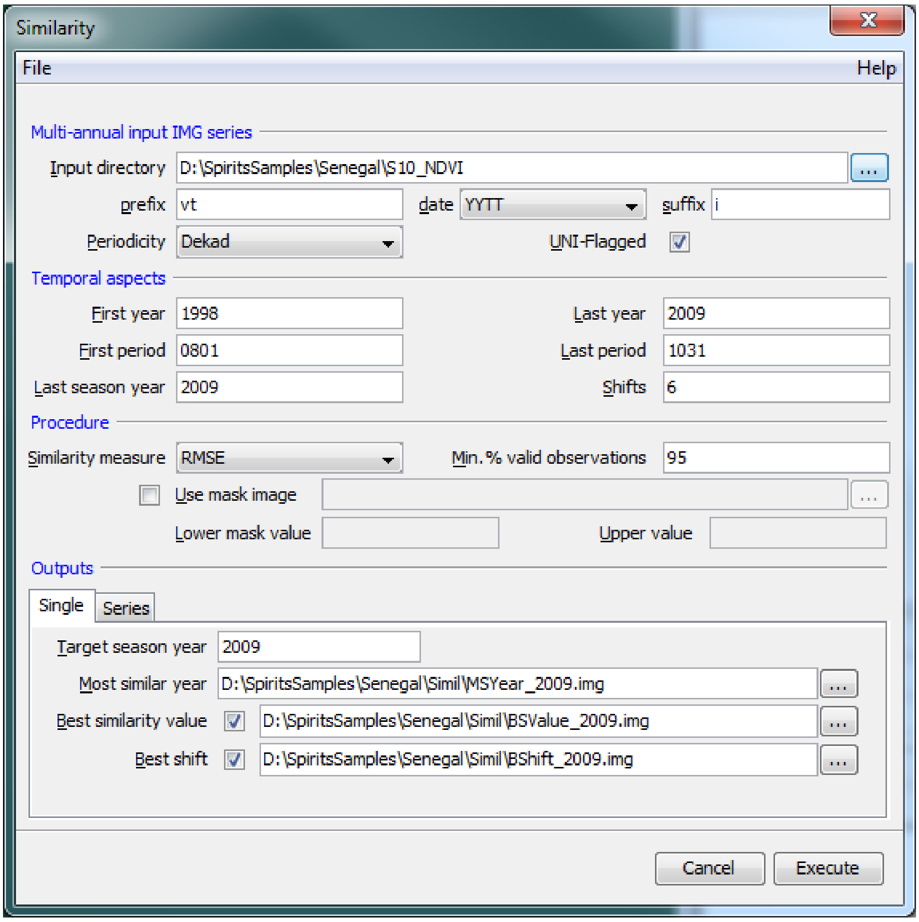
In case the last season is chosen to be 'the 2010 season', the target season year has to be chosen between 2001 and 2010.

- the number of periods the input seasons are allowed to shift in both directions for the comparison, to take into account annual differences in phenology.
- the similarity measure. Available measures are:
  - MAD: Mean Absolute Deviation;
  - RMSE: Root Mean Squared Error;
  - R - correlation coefficient.
- the minimum % of valid observations;
- an optional mask IMG and mask range (upper and lower digital value): Pixels with mask values beyond this range are excluded. This enables the analysis to be limited to a subset of pixels, e.g. croplands;
- the output images:
  - the IMG containing the most similar years (offsets from 1950). Remark in case these images are to be used in the "Similarity To Yield" tool, the ***PrefixDateformatSuffix*** file naming convention, with Dateformat being YYYY or YY must be used;
  - optional, the IMG containing the best similarity values (lowest MAD/RMSE or highest R value);
  - optional, the IMG containing the best shift-values (periods, relative to the most similar year).

Remark: In case of yearly input IMGs:

- the last period does not apply, and is reset to the first period;
- extending and shifting the input series is not applicable;
- the output IMG containing shift values is not applicable;

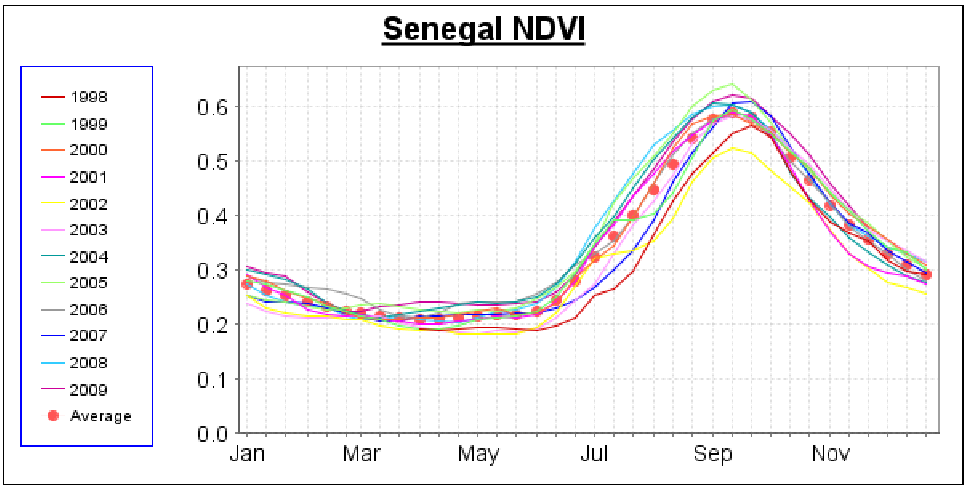
Tool



Similarity Tool example

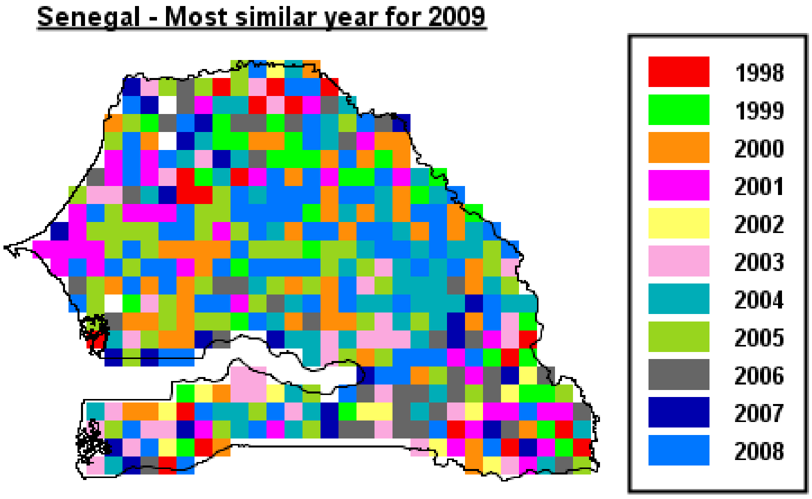
Similarity example

example: compute most similar year and shifts for Senegal 2009 from IMGs from 1998 till 2009.

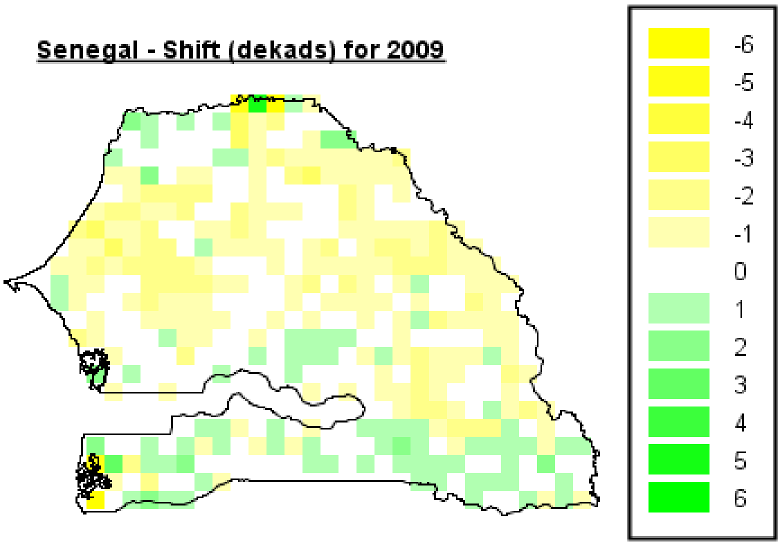


Based on the average NDVI in the graph, the critical period for the similarity analysis, in this example, is chosen to be August - October.

RUM Chart Senegal NDVI (overall mean)



Most similar year IMG.



Best shift IMG.

3.29. Similarity To Yield

Goal

Convert "most similar year" IMGs (created by the Similarity tool) into IMGs with estimated yields.

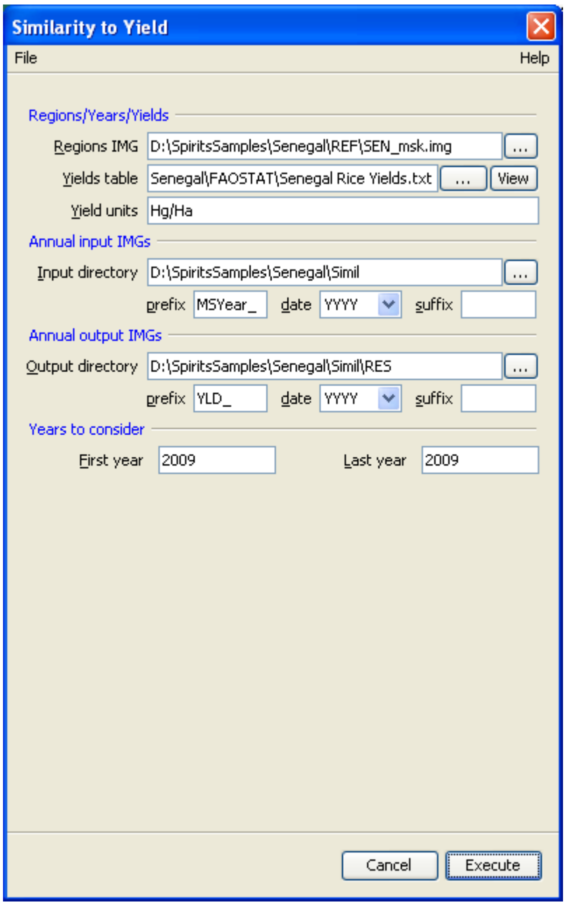
Parameters

- the regions IMG containing per pixel an integer region-ID;
- the yield values per region per year. These values must be provided by a comma separated ASCII file, with per line the region-ID (corresponding with the regions IMG), the year (YYYY) and the yield value for this region/year. Additional following columns may be present and will be ignored. The sorting of the table is irrelevant;
- the yield units to be used in the output IMG HDRs values entry;
- the location and filename structure of the "most similar year" input IMGs created by the Similarity tool;
- the location and filename structure of the output IMGs containing the yield estimates;

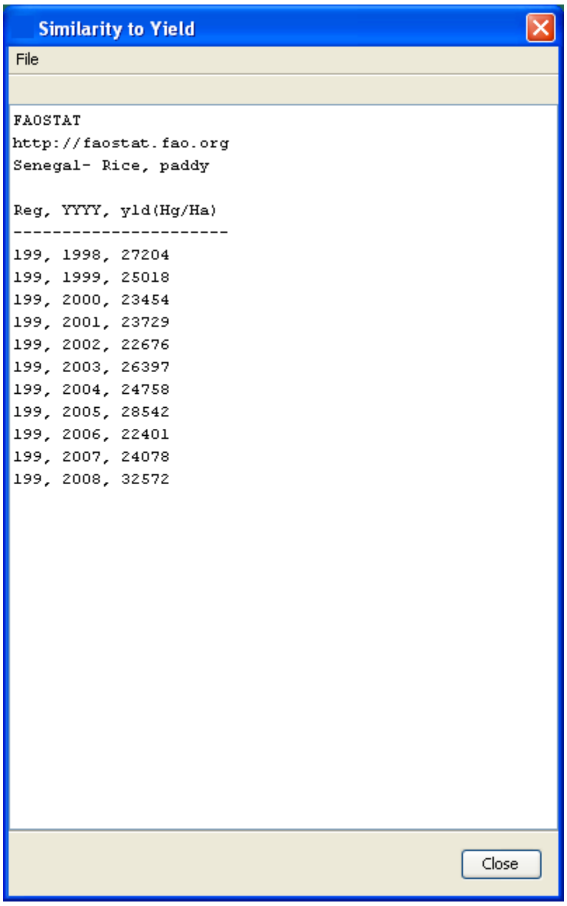
The tool requires both input and output IMGs to follow the ***PrefixDateformatSuffix*** file naming convention, with Dateformat being YYYY or YY; the images generated by the Similarity tool can have any name (in 'Single' mode) or any DateFormat (in 'Series' mode), but as it concerns a typical “annual” time series analysis, it is most logical to include (only) the (target) year in the image names.

- the start and end year to of the input IMGs to process (the years for which the Similarity tool was executed as “target years”).

Tool



Similarity to Yield Tool example



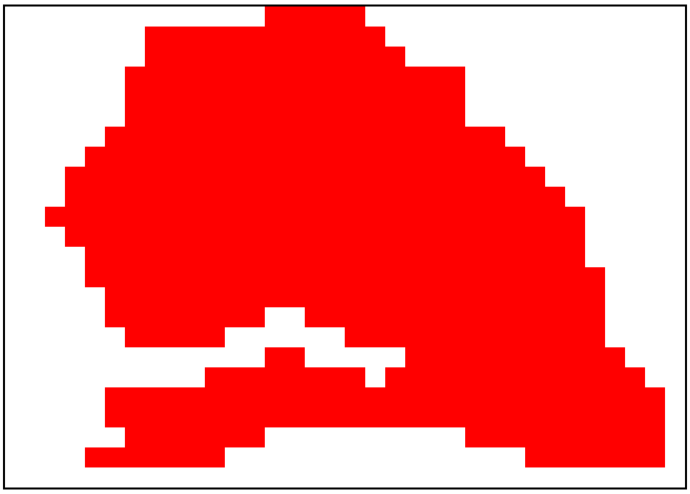
Similarity to Yield view panel example

Similarity To Yield example

example: compute Rice yield estimates for 2009 in Senegal, based on the most similar year IMG from the Similarity example.

In the example only one 'region' is used, being the whole of Senegal. The regions IMG used is actually a mask IMG, containing value 199 for all 'Senegal' pixels.

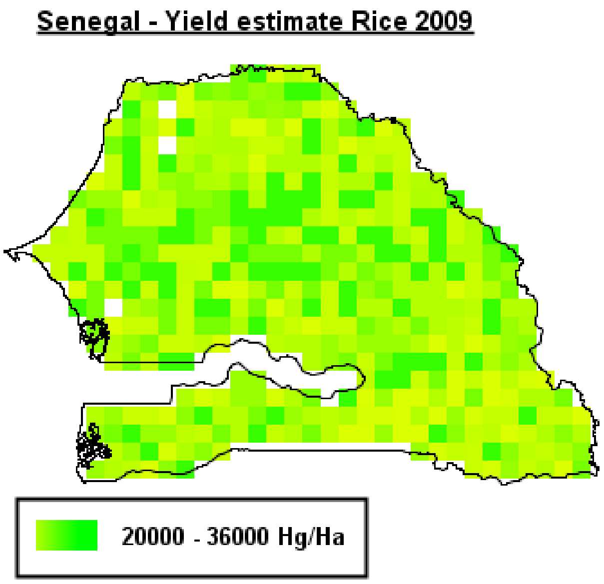
The yields table is based on values downloaded from FAOSTAT (Statistics Division of the Food and Agriculture organization of the United Nations ref. http://faostat.fao.org.).



Senegal mask used as regions IMG

```
FAOSTAT
http://faostat.fao.org
Senegal- Rice, paddy

Reg, YYYY, yld(Hg/Ha)
----------------------
199, 1998, 27204
199, 1999, 25018
199, 2000, 23454
199, 2001, 23729
199, 2002, 22676
199, 2003, 26397
199, 2004, 24758
199, 2005, 28542
199, 2006, 22401
199, 2007, 24078
199, 2008, 32572
```

yields file



3.30. ISO Clustering

Goal

Create classification IMGs using ISODATA unsupervised classification (clustering) and maximum likelihood methods.

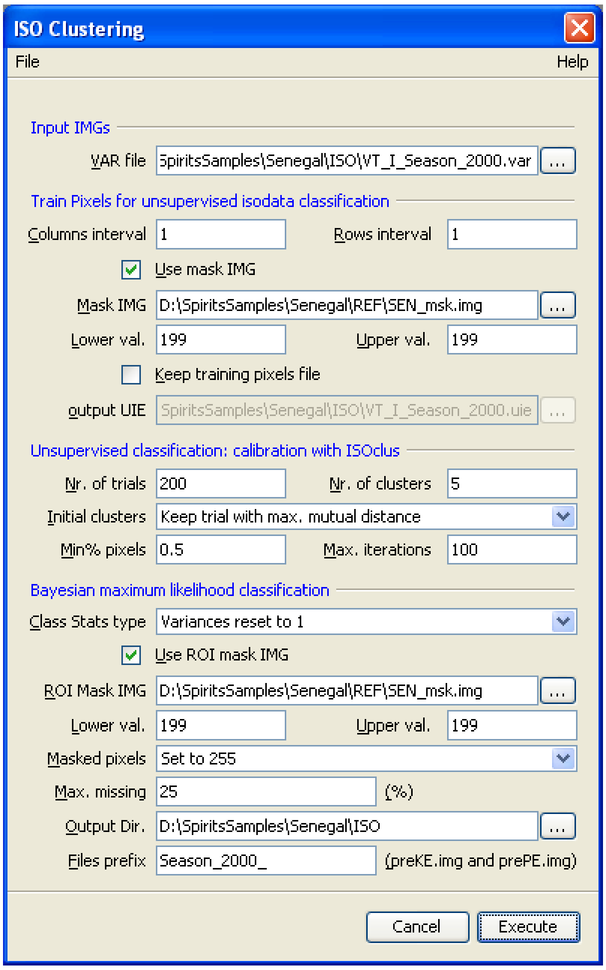
Parameters

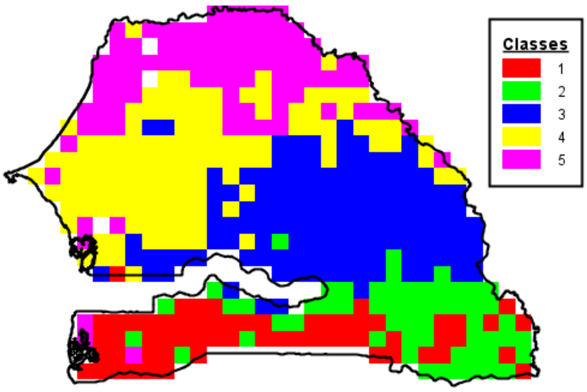
- the VAR file specifying the series of IMGs to be classified. Typically this VAR file contains the NDVI IMGs over the growing season of the concerned region. The IMGs must have byte data type (HDR entry data type = 1);
- the columns (N) and rows (M) intervals, specifying the train pixels. The pixels of each N-th column of each M-th row of the IMGs will be selected as train pixel.
- an optional mask IMG and mask range: Pixels with mask values beyond this range are excluded.

- the number of trials. The maximum number of trails is the available number of train pixels / 2 - with an absolute maximum of 10000;
- the desired number of clusters. The actual resulting number of clusters can be minimal 2 less or maximal 2 more than the desired number of clusters;
- the method to define initial cluster centra. During each trial a random set of pixels (one for each desired cluster) is selected. Options are:
  - to keep the trail with maximum mutual distance;
  - to keep the trail with minimum within-cluster variance.
- the maximum number of iterations to obtain the desired number of clusters;
- the minimum % pixels (of the total) per class;

- the statistics type to use. Options are:
  - with variances reset to 1;
  - with original variances kept.
- an optional mask IMG and mask range specifying the ROI. This mask must byte-type.
- the maximum percentage of missing values. Pixels with more missing values will be flagged (flag=254);
- the output directory where the resulting IMGs should be stored;
- the prefix (or base name) of these files. The names of the actual files created will be:
  - prefixKE.IMG : an IMG containing the estimated classes (hard classification file);
  - prefixPE.IMG: an IMG containing the post probabilities for the estimated classes.

Tool





ISO Clustering Tool example output classification file

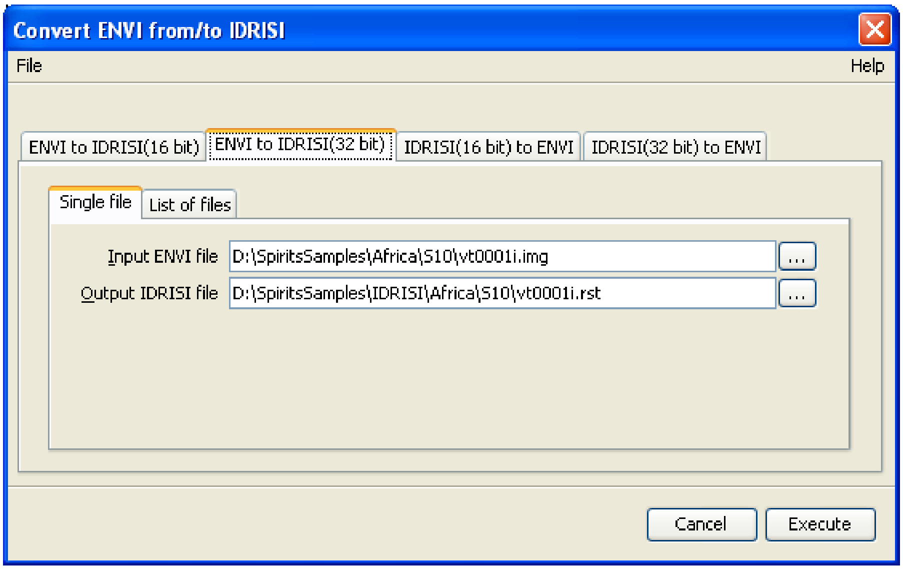
3.31. Convert ENVI from/to IDRISI

Goal

File conversions between images in ENVI and IDRISI raster formats:

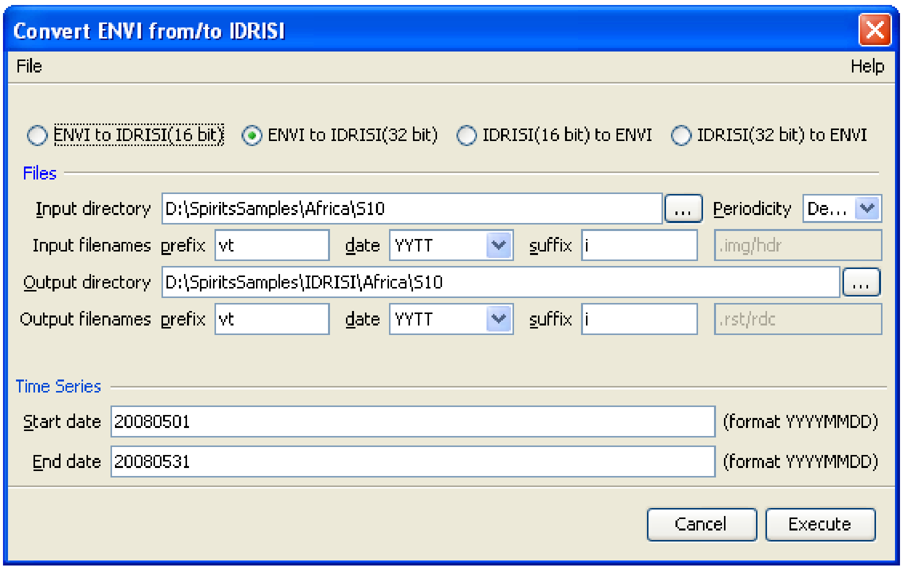
- from/to ENVI IMG/HDR format to/from IDRISI IMG/DOC format;
- from/to ENVI IMG/HDR format to/from IDRISI RST/RDC format;

Tool



Convert ENVI from/to IDRISI Tool example

Time Series



Convert ENVI from/to IDRISI Time Series example

Remark:

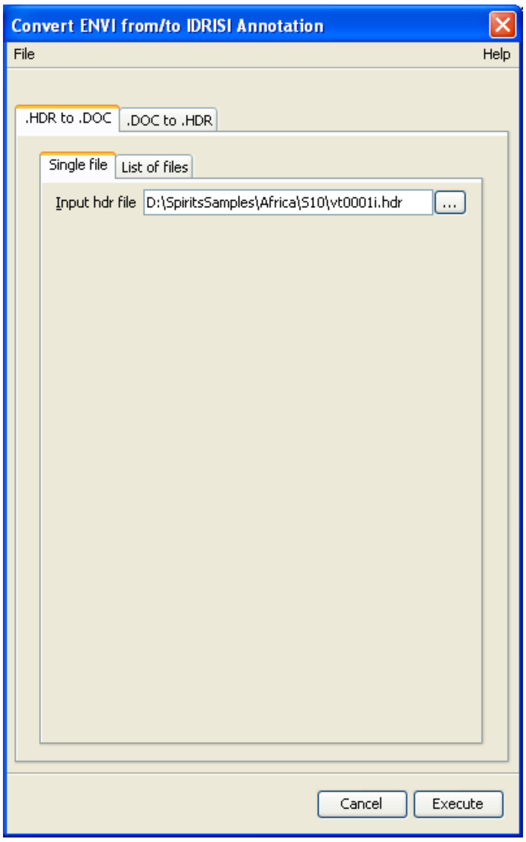
due to the simplicity of the parameters required, this time series does not use a scenario but hosts the parameters directly.

3.32. Convert ENVI from/to IDRISI Annotation

Goal

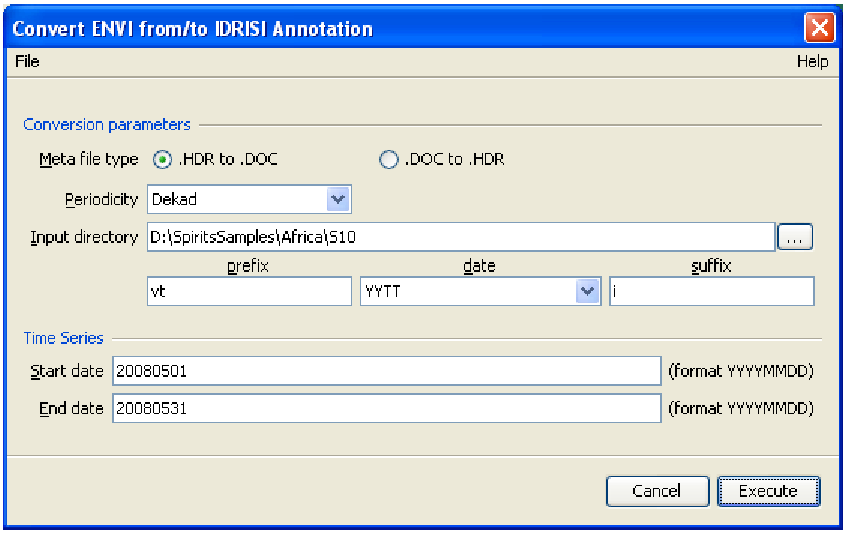
File conversions between the annotation files (HDR and DOC) of images in ENVI and IDRISI IMG format. The DOC (HDR) file is created at the same locations, with the same name as the input HDR (DOC) file.

Tool



Convert Annotation Tool example

Time Series



Remark:

due to the simplicity of the parameters required, this time series does not use a scenario but hosts the parameters directly.

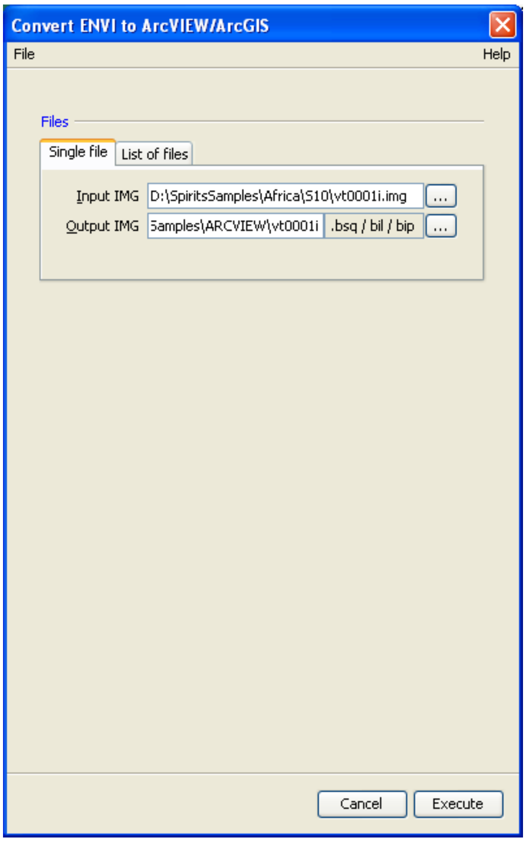
Convert Annotation Time Series example

3.33. Convert ENVI to ArcVIEW/ArcGIS

Goal

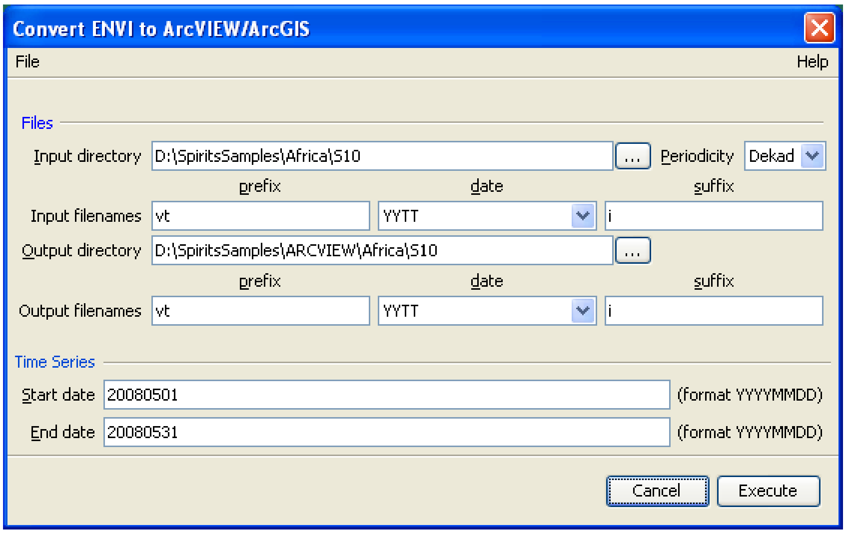
File conversion from (single or multiband) images in ENVI to ArcVIEW/ArcGIS raster format (BSQ/BIL/BIP).

Tool



Convert ENVI to ArcVIEW/ArcGIS Tool example

Time Series



Remark:

due to the simplicity of the parameters required, this time series does not use a scenario but hosts the parameters directly.

Convert ENVI to ArcVIEW/ArcGIS Time Series example

3.34. Adapt LF to CRLF characters

Goal

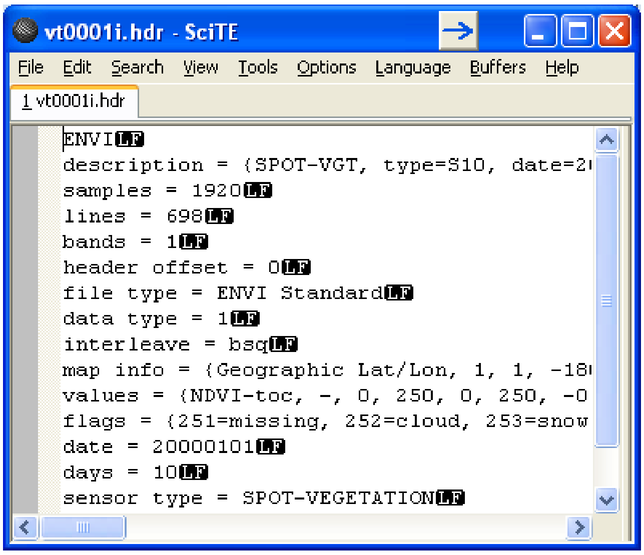
In-Situ replacement of all LF characters (UNIX End-Of-Line) by CRLF (Windows/DOS) in ASCII files

Remark: should be used ONLY on ASCII text files.

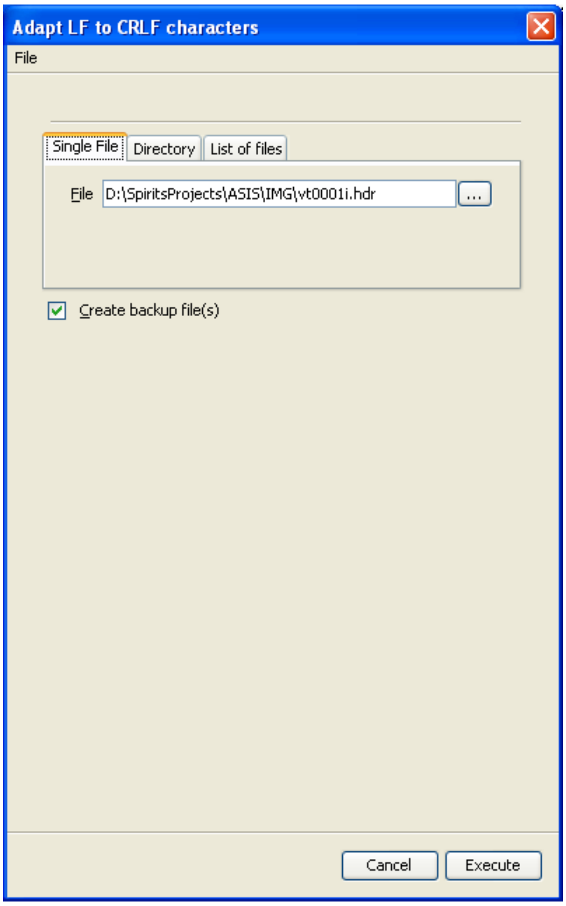
Parameters

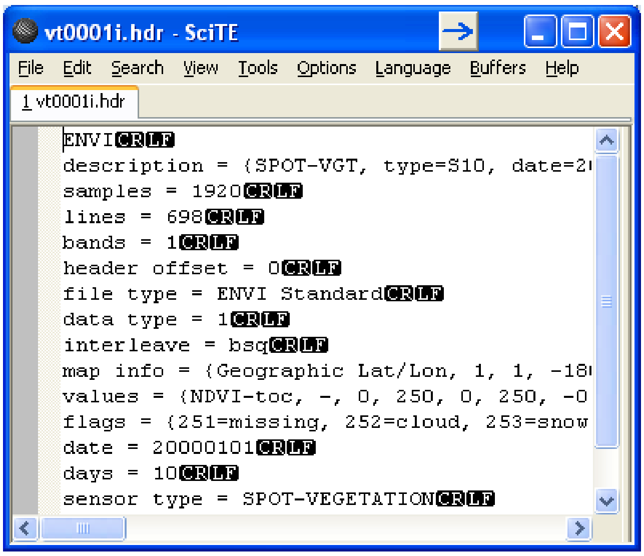
- the file(s) to be adapted;
- whether to backup the files prior to modifying them. Backup files will have the same file names as the original ones with an additional .BAK extension. In case a .BAK file already exists, the program issues an error message and leaves the original file intact.

Tool



original file





Extract Band/ROI Tool example

adapted file

3.35. Combine IMGs to 3D ENVI IMG

Goal

Combine a set of (2D) IMGs into a 3D ENVI IMG.

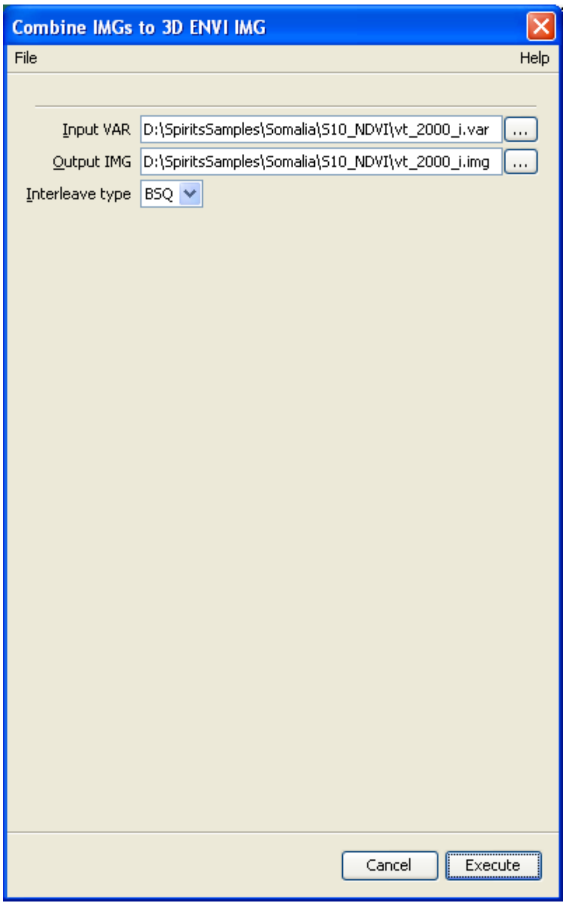
Parameters

- the VAR file specifying the IMGs to combine. All IMGs must be 2-D and have identical data type and spatial features;
- the output IMG;
- the band interleave type for the output 3-D ENVI IMG:
  - BSQ (band sequential);
  - BIL (band interleaved per line);
  - BIP (band interleaved per pixel).

Remarks:

- for BIL/BIP, the number of IMGs in the VAR file is limited to 2045;
- for BIP, the product (samples x bands x bytes per pixel) is limited to 2 147 483 647.

Tool



Combine IMGs to 3D ENVI IMG Tool example

3.36. Descale IMG

Goal

Descale image to float datatype and remove scaling.

The scaling factors (***Vint***, ***Vslo***) in the result image are (0,1) thus its physical values are equal to its digital values (Physical value = ***Vint*** + ***Vslo*** * Digital IMG value).

Optionally a GeoTIFF formatted copy of the descaled IMG can be created.

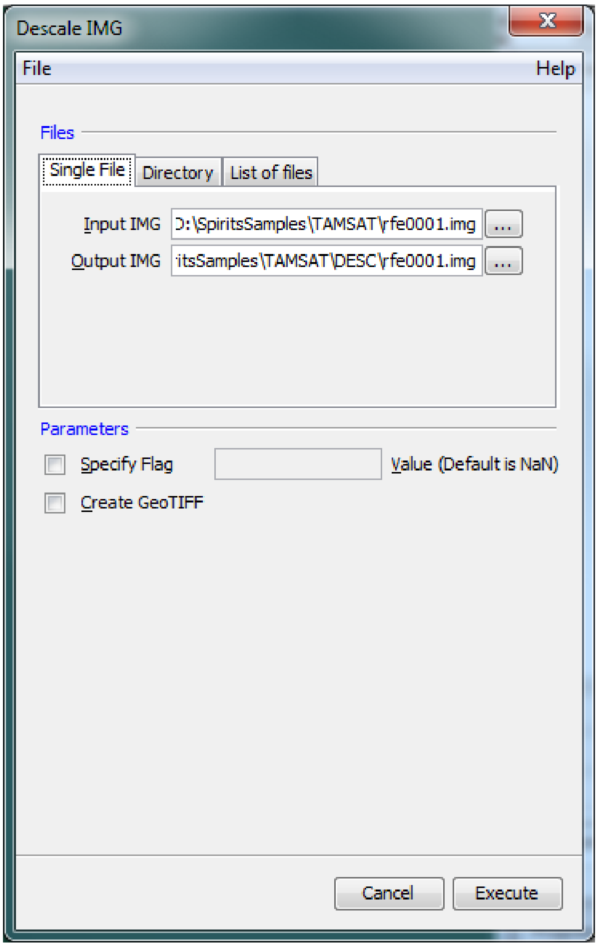
Parameters

- The flag value to be used. Default value is NaN (Not a Number).

Remarks:

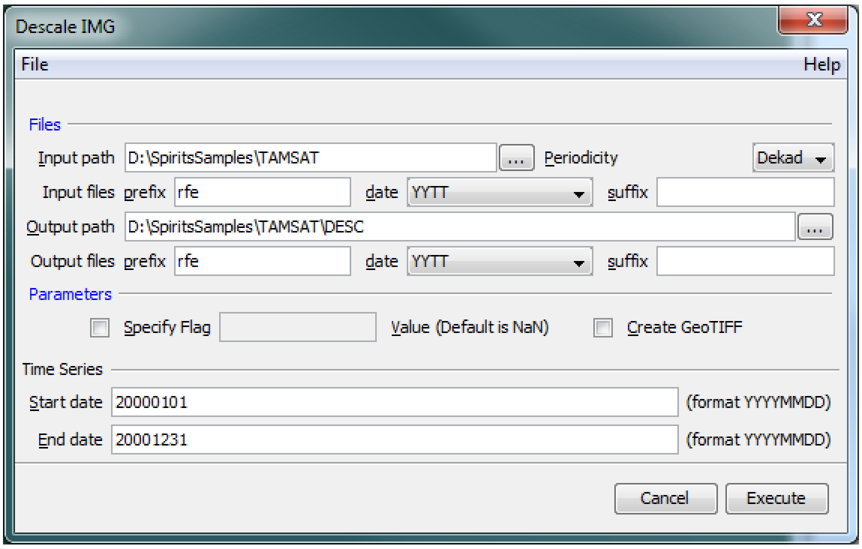
- o the flag value -if specified- may not overlap the images significant range [Vlo - Vhi].
- o the GeoTIFF file name will be that of the output file, prefixed with "TIFF_", and have extension ".tiff".

Tool



Descale IMG Tool example

Time Series



Descale IMG Time Series example

Remark:

due to the simplicity of the parameters required, this time series does not use a scenario but hosts the parameters directly.

3.37. Adapt HDR

Goal

Adapt an existing HDR file.

Parameters

The new values or values-set for the HDR items (keys) to be adapted. Some items can be removed (for standard items this means their values will be cleared, non-standard items are removed completely). In case of the "Flags" item, the item can be removed completely, the values-set can replaced completely, or additional values can be added to the original set.

The HDR items this tool can handle are described below. Keys marked with * are non-standard keywords.:

- Description

	meaning:	General title
	HDR key:	description
	example:	Africa ROI extracted from GLD
	result:	description = {Africa ROI extracted from GLD}

- Comment

	meaning:	Additional comments
*	HDR key:	comment
	example:	created for spirits manual
	result:	comment = { created for spirits manual}

- Sensor

	meaning:	Sensor name
	HDR key:	sensor type
	example:	SPOT-VEGETATION
	result:	sensor type = SPOT-VEGETATION

- Offset

	meaning:	Header Offset (leading bytes before actual data)
	HDR key:	header offset
	example:	0
	result:	header offset = 0

- Samples

meaning: number of columns in IMG

HDR key: samples

example:	460
result:	samples = 460

- Lines

meaning: number of rows in IMG

HDR key: lines

example:	390
result:	lines = 390

- Map Info, Map System, Magic Column, Magic Record, Magic X, Magic Y, X-resolution, Y-resolution

meaning: Map System: projection name

Magic Column, Magic Record: column and record of the Magic point

Magic X, Magic Y: map coordinates-X/Y (or Lon/Lat) of the magic point

X-resolution, Y-resolution: pixel sizes in X/Y-units

Map Info: the complete set of values.

HDR key: map info

map info = { Map System,
Magic Column , Magic Record,
Magic X , Magic Y,
X-resolution, Y-resolution }

example:	Map Info	any_sys, 1, 1, 100, 100, 10, 10
result:		map info = {any_sys, 1, 1, 100, 100, 10, 10}
example:	Magic Column	1.5
	Magic Record	1.5
	Magic X	105
	Magic Y	95
result:		map info = { any_sys, 1.5, 1.5, 105, 95, 10, 10 }

Remark: via "Map Info" one can adapt the complete set of values at once, via " Map System", "Magic Column"... "Y-resolution" one can adapt the individual values. In case both options are used, the individual values will overwrite the ones from the set.

- Bands

meaning: number of bands in IMG

HDR key: bands

example:	1
result:	bands = 1

- Interleave

meaning: organisation of the bands IMG file: BSQ (Band Sequential), BIL (Band-Interleaved per Line) or BIP (Band-Interleaved per Pixel). Only relevant in case bands > 1.

HDR key: interleave

example:	BSQ
result:	interleave = bsq

- Data type

meaning: data type of the IMG: Byte (8 bit, unsigned), Integer (16 bit, signed), Long (32 bit, signed) and Float (32 bit)

HDR key: data type

example:	Byte
result:	data type = 1

- Byte Order

meaning: byte-order (endianness) in the IMG: High or Low Endian

HDR key: byte order

example:	0
result:	byte order = 0

- Yname, Yunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo

meaning: Yname, Yunit: name and unit of physical variable Y

Vlo, Vhi: significant range in digital units V. Flags are situated beyond this range.

Vmin, Vmax: observed range in digital units V. (Vlo <=Vmin <= Vmax <=Vhi).

Vint, Vslo: scaling of physical variable Y: Y = Vint + Vslo x V

* HDR key: values (values = { Yname, Yunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo})

example:	Yname	daily max temperature
	Yunit	decigrades Celsius
result:		values = { daily max temperature, decigrades Celsius,...}
example:	Vlo	-1000
	Vhi	+1000
result:		values = {...,-1000, 1000,...}
example:	Vint	0
	Vslo	1
result:		values = { ..., 0, 1}

- Flags

  meaning:

  * HDR key:

example:	Flags	251=missing, 252=cloud
result:		flags = {251=missing, 252=cloud}
example:	Add Flag	253=snow
result:		flags = {251=missing, 252=cloud, 253=snow}

- Classes HDR

  meaning: all classes - related items are copied from the referenced HDR file

  HDR key: classes
  class names
  class lookup

```
example:  D:\SpiritsSamples\GLD\REF\glc2000.hdr

result:   classes = 24
          class names = {
            Water/Background,
          ...
             Artificial surfaces and associated areas,
             No data (small islands)
          }
          class lookup = {
            255,255,255,
          ...
            255,  0,  0,
            255,255,255
          }
```

- Date

  meaning: registration date or start date for composite IMGs (YYYYMMDD format).

  * HDR key: date

example:	20010101
result:	date = 20010101

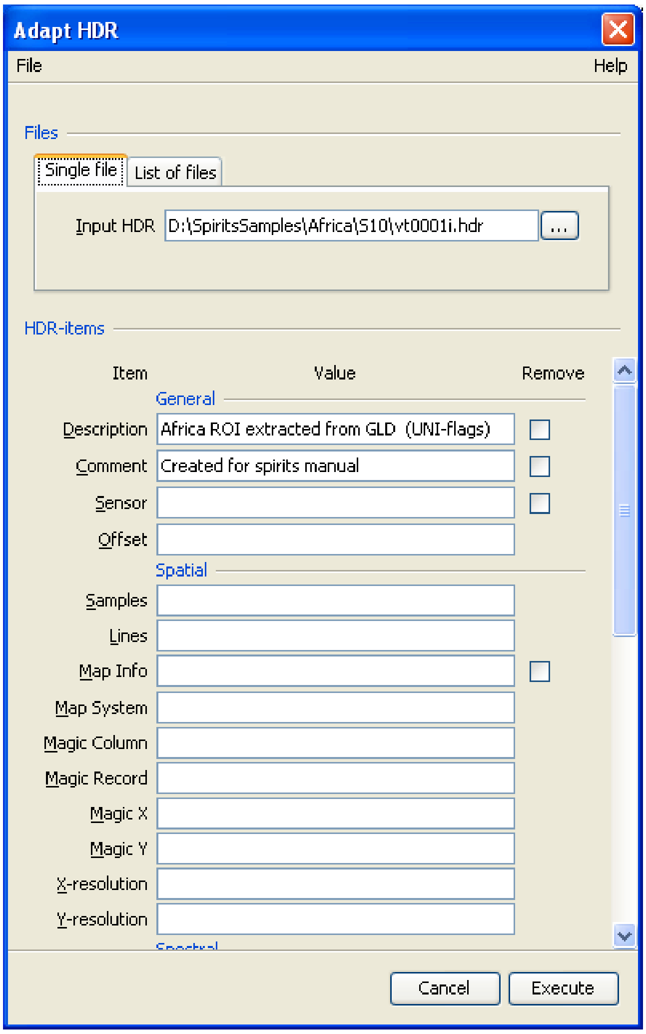
- Periodicity

  meaning: Periodicity in days. 1,10,30,... 0=unknown, -1= actual registration

  * HDR key: days

example:	Dekad
result:	days = 10

Tool

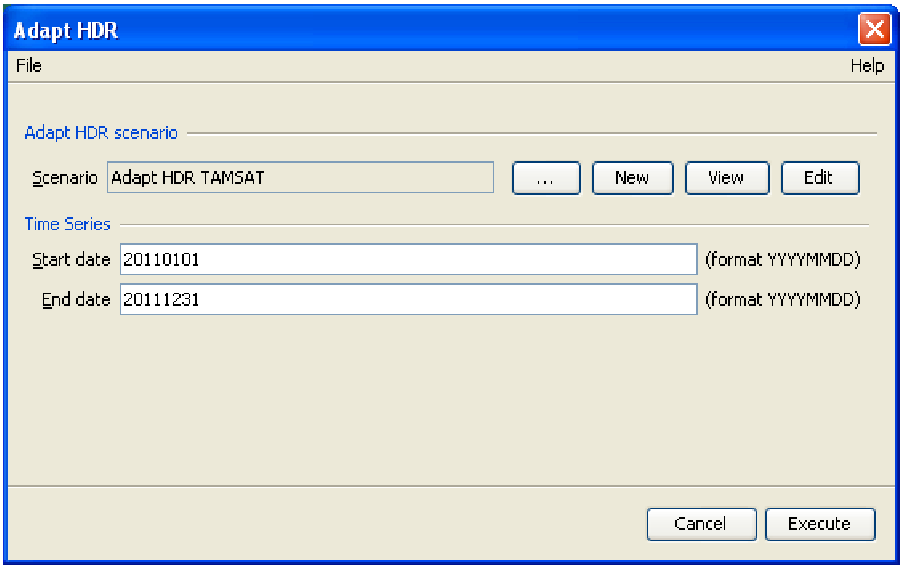


Adapt HDR Tool example

```
ENVI
description = {Africa ROI extracted from GLD (UNI-flags)}
samples = 460
lines = 390
bands = 1
header offset = 0
file type = ENVI Standard
data type = 1
interleave = bsq
map info = {Geographic Lat/Lon, 1, 1, -26.066964, 38.0669643, 0.1875, 0.1875}
values = {NDVI-toc, -, 0, 250, 0, 250, -0.08, 0.004}
flags = {251=missing, 252=cloud, 253=snow, 254=sea, 255=back}
date = 20000101
days = 10
sensor type = SPOT-VEGETATION
comment = {Created for spirits manual}
program = {IMGcvt.exe (V1002/1009) + HDRadapt.exe (V912)}
```

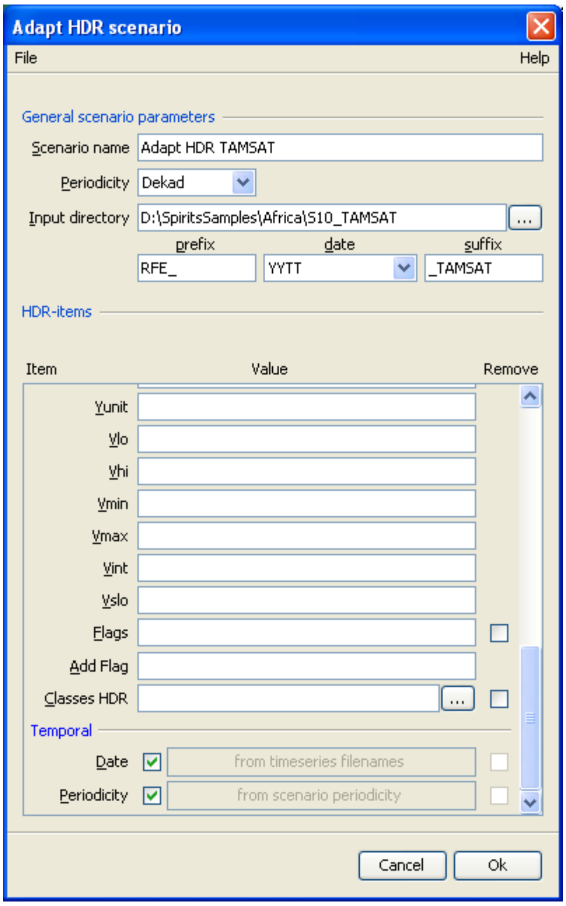
output HDR

Time Series



Extract Band/ROI Time Series example

Scenario



Adapt HDR Scenario example

Remark:

for Adapt HDR time series, the date and days values are determined via the file names and periodicity.

3.38. Meteo to IMG

3.38.1. General

Goal

Extract and convert spatially gridded meteorological data from ASCII tables to IMGs.

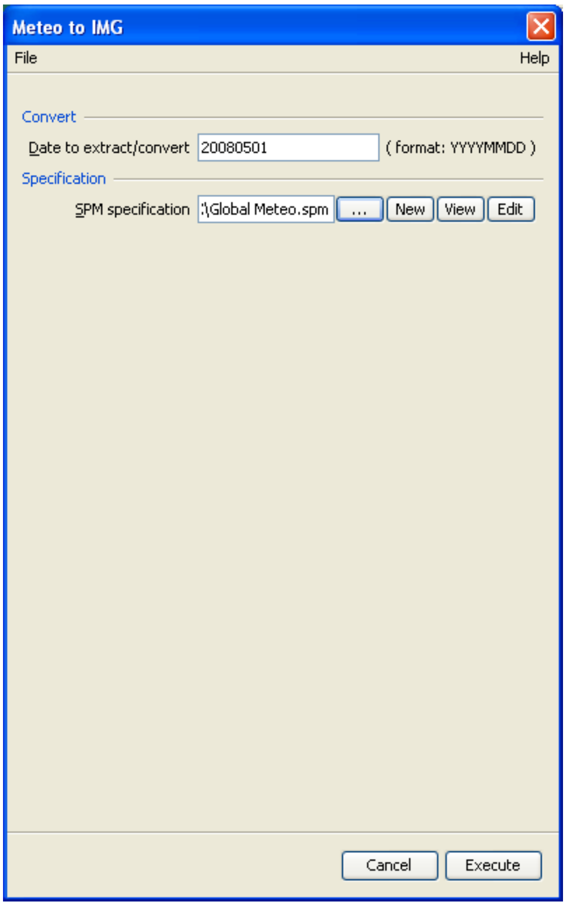
Parameters

- the date of the data to be extracted/converted. This date has dual usage:
  - it is part of the specification of the file name of the input file that will be converted;
  - it defines the date of the data that will be extracted. In case the input file contains a YYYYMMDD field, it may comprise data from different dates, but the extracted data will be limited to those lines for which the YYYYMMDD field matches the date parameter specified.
- an SPM file, Meteo conversion specification file, containing the detail specifications for the conversion.

Remarks:

- The input files need to follow the ***PrefixDateformatSuffix.Extension*** file naming convention;
- The input files must be ASCII tables with per line all meteo-information of one date and location.
  - Subsequent values represent fixed labels and variable meteo values (temperature, precipitation,...). They must be separated by a clear delimiter (mostly comma or semicolon).
  - A date field, in the format YYYYMMDD, is optional. If not present, all data are assumed to be of the date specified by the date parameter.

Tool

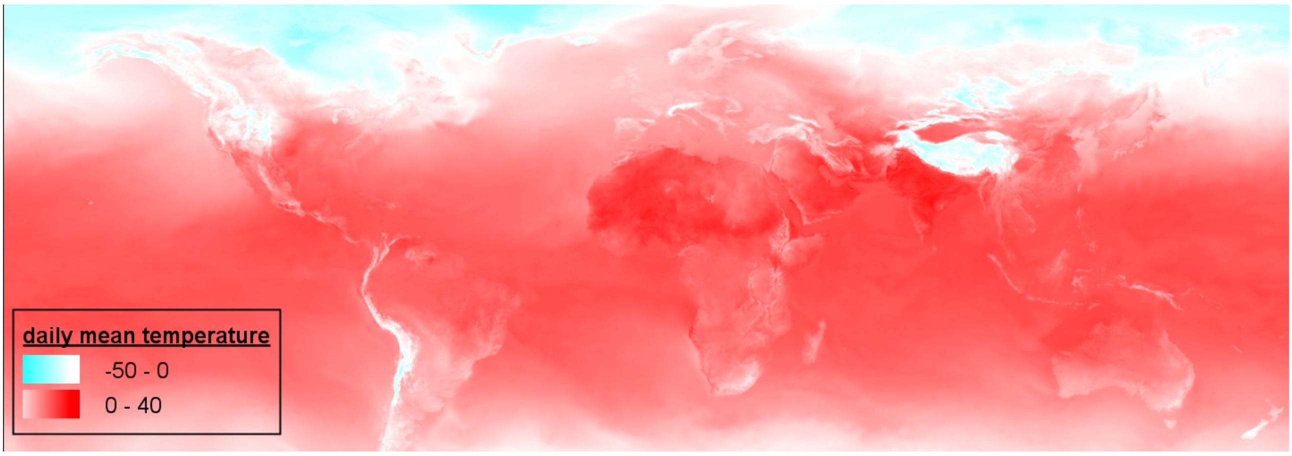


Meteo to IMG example

```
75.00,-179.75,20080501,-12.82,-12.22,-13.46,0.12,1.13,0.95,0.60,20578.3
75.00,-179.50,20080501,-12.85,-12.25,-13.52,0.12,1.12,0.94,0.60,20578.3
75.00,-179.25,20080501,-12.88,-12.28,-13.58,0.12,1.12,0.94,0.60,20578.3
75.00,-179.00,20080501,-12.94,-12.31,-13.65,0.12,1.11,0.93,0.60,20578.3
75.00,-178.75,20080501,-12.97,-12.34,-13.71,0.12,1.10,0.93,0.60,20578.3
75.00,-178.50,20080501,-13.00,-12.41,-13.77,0.12,1.10,0.92,0.60,20578.3
75.00,-178.25,20080501,-13.07,-12.44,-13.83,0.12,1.09,0.91,0.60,20578.3
75.00,-178.00,20080501,-13.10,-12.50,-13.93,0.00,1.09,0.91,0.60,20611.1
75.00,-177.75,20080501,-13.16,-12.56,-13.99,0.00,1.08,0.90,0.60,20643.8
...
-50.00,178.25,20080501,6.68,6.87,6.38,0.49,0.36,0.28,0.99,3997.7
-50.00,178.50,20080501,6.72,6.91,6.45,0.37,0.36,0.27,0.99,4128.8
-50.00,178.75,20080501,6.75,6.94,6.51,0.37,0.34,0.26,0.97,4210.7
-50.00,179.00,20080501,6.75,7.00,6.57,0.24,0.31,0.23,0.94,4259.8
-50.00,179.25,20080501,6.75,7.00,6.67,0.12,0.27,0.18,0.90,4292.6
-50.00,179.50,20080501,6.78,7.00,6.76,0.12,0.25,0.16,0.88,4309.0
-50.00,179.75,20080501,6.81,6.97,6.88,0.12,0.22,0.13,0.85,4276.2
-50.00,180.00,20080501,6.81,6.94,7.07,0.12,0.13,0.05,0.78,4161.5
```

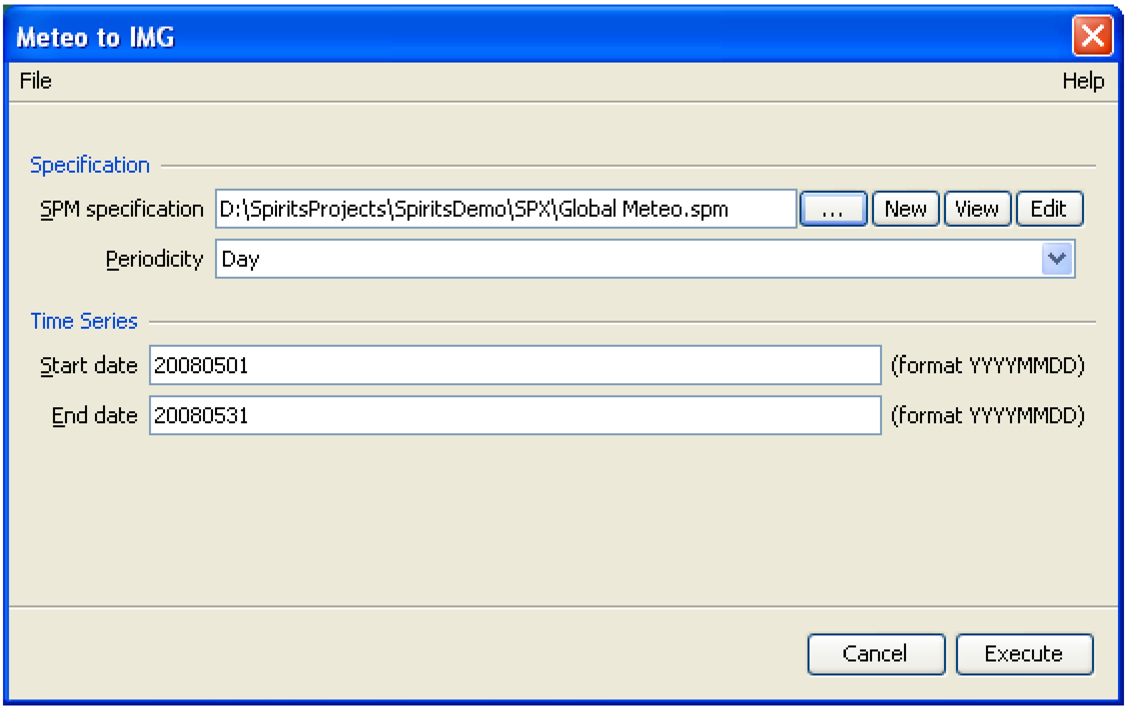
```
Fields:

01    Latitude                (decimal degrees)
02    Longitude               (decimal degrees)
03    Date                    (yyyymmdd)
04    Mean temperature        (Celsius)
05    Maximum temperature     (Celsius)
06    Minimum temperature     (Celsius)
07    Precipitation           (mm)
08    E0, bare soil           (mm)
09    ES0, over water         (mm)
10    ET0, Penmann-Monteith   (mm)
11    Global radiation        (kJ.m-2.d-1)
```

example: meteo data file for 20080501



example: extracted mean temperature, converted to IMG

Time Series



Meteo to IMG Time Series example

Remark:

since there are so few parameters needed apart from the SPM file, the Meteo to IMG Time Series tool does not use a separate scenario.

The only additional parameter needed for the time series is the Periodicity, which can be filled out directly in the UI of the Meteo to IMG Time Series tool.

3.38.2. SPM File: Meteo conversion specification

Description

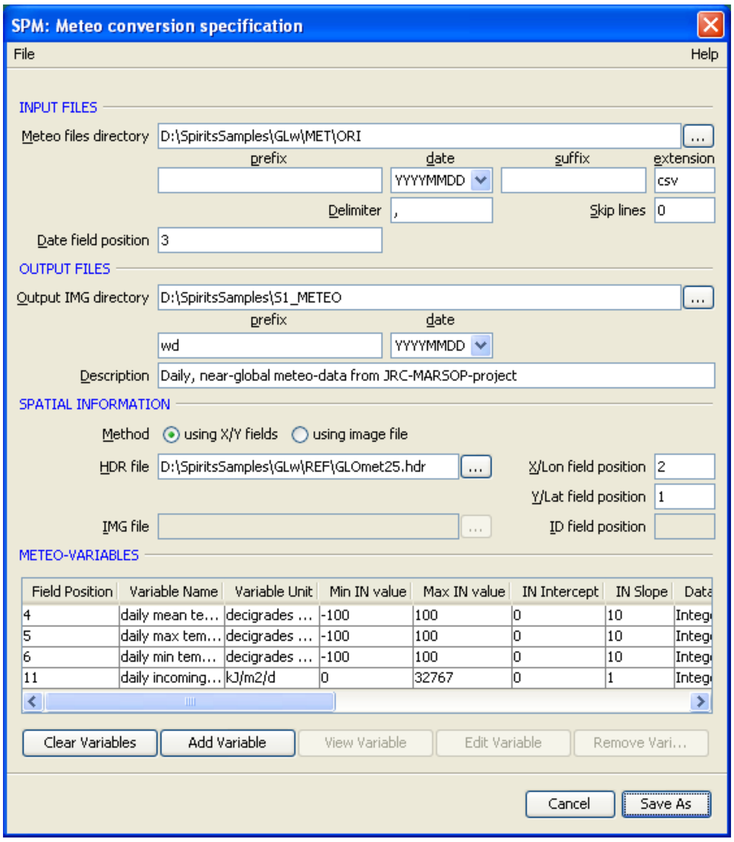
An SPM file, Meteo conversion specification file, contains the detail specifications for the Meteo to IMG tool and time series. These files can be created and edited with the SPM-editor.

Parameters

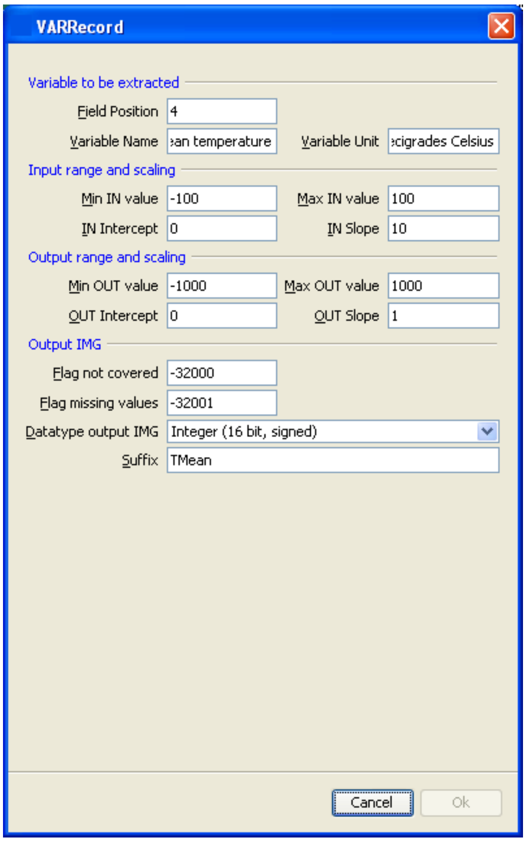
- the location and filename structure of the (ASCII) input file(s) containing the meteorological data tables;
- the delimiter used to separate the fields in these file(s);
- files might contain leading lines describing the file(s) contents, the Skip lines field allows to specify a number of lines to be ignored at the start of the file(s);
- the position of the date field (optional). .In case the input file(s) contains a YYYYMMDD field, they may comprise data from different dates, but the extracted data will be limited to those lines for which the YYYYMMDD field matches the date parameter specified in the tool (the same date coded in the file(s) name).
- the location and filename structure of the output IMGs. Since the tool can extract multiple meteo variable at one, thus create multiple IMG files. Each of these will share the same output directory, date format specification and prefix. The file name suffix however needs to be specified for each extracted meteo variable separately;
- a description (optional). This description will be used as value for the description key in the HDR files of the generated IMGs.
- the method by which the spatial information for the input data can be determined. There are two methods supported:
  - the data table contains X and Y fields directly containing the Lon/Lat or X/Y coordinates of the centre of the cell. In this case the position of these fields need to be specified. Additionally a HDR file is needed which expresses the geo-referencing of the grid cells (datum, map projection, framing). The same HDR can also be used to restrict the extension of the generated IMGs (e.g. extract only China from global meteo files).
  - the data table contains an ID field, containing a unique ID-number for each cell. In this case the position of this ID field need to be specified. Additionally an intermediate “Meteo-ID” IMG must be provided which shows the location of each cell. This image can be used to restrict the analysis to a limited zone. And the generated images can immediately be created in a different map system, if the Meteo-ID image is projected to that system (e.g. extract meteo-images for China in Albers Equal Area from global meteo-data which are originally gridded in the Lon/Lat system).
- for each variable to be extracted additional parameters are needed:
  - the position of the field containing the variable;
  - the name of the variable (to be used in the values item in the HDR files of the generated IMG);
  - the unit of the variable (to be used in the values item in the HDR files of the generated IMG);
  - the minimum and maximum value of the variable as read from the file. Values beyond this range will be flagged as missing values;
  - the intercept and slope for the variable as read from the file, determining the physical input value according to Physical value = intercept + slope * (value read from the file). Normally intercept will be 0, slope will be 1. These can be used in case the output values should be expressed in different units, example: in case the input file contains a variable in degrees Celsius, and the output file needs this variable to be expressed in decigrades Celsius, it can be converted by specifying a slope=10;
  - the data type of the generated IMG (Byte, Integer, Long, Float);

- ○ the intercept and slope for the variable as written in the generated IMG, according to Physical value = intercept + slope * (value in the generated IMG). These can be used to scale the physical value of the variable into the data type of the generated IMG;
- ○ the minimum and maximum value of the variable as written in the generated IMG. Values beyond this range will be chopped to this minimum or maximum value;
- ○ the file name suffix for the generated IMG.

Editor



SPM Editor - main panel example



SPM Editor - variables subpanel example

The parameters common for all variables to be extracted can be filled out directly in the SPM-editor panel.

At the bottom of the panel is a list showing for each variable to be extracted its specific parameters.

Via the action buttons variables can be added, removed, edited or inspected: a new panel appears handling the specific parameters of the selected variable.

FOODSEC Meteodata example

In this example the average temperature (TAV), the maximum temperature (TMAX), the minimum temperature (TMIN), the precipitation sum (RRR) and the global radiation sum (RAD) are extracted from a FOODSEC Meteodata DBF file for the first dekad of 2011 of the Africa region.

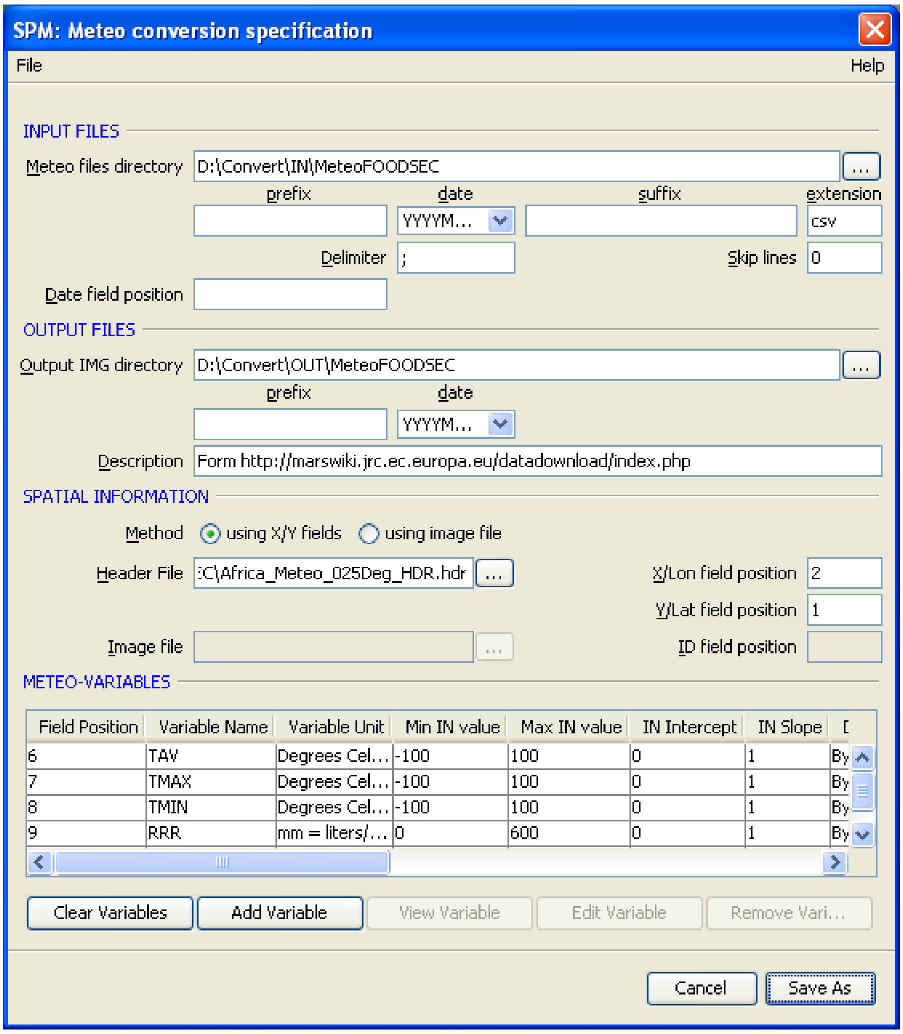
Initially this file has been converted to an ACII table, and renamed to meet the file name date format requirements. (In this example: 20120101.csv.)

The FOODSEC Meteodata documentation informs that the file has a 0.25 degree resolution, and contains following fields:

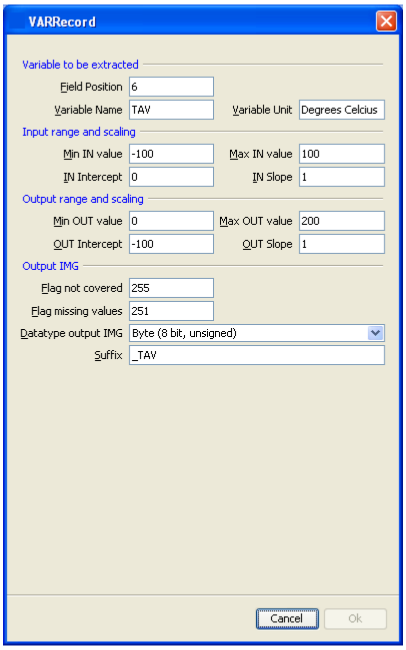
```
Meteorological variables:
* TAV - average temperature - (C)
* TMAX - maximum temperature - (C)
* TMIN - minimum temperature - (C)
* RRR - precipitation sum - (mm = liters/m2)
* E0 - evapo-transpiration sum (over water) - (mm = liters/m2)
* ES0 - evapo-transpiration sum (bare soil) - (mm = liters/m2)
* ET0 - evapo-transpiration sum (Penman-Monteith) - (mm = liters/m2)
* RAD - global radiation sum - (kJ/m2 per dekad)
* SDAV - average snow depth - (cm)
* SDMIN - minimum snow depth - (cm)
* SDMAX - maximum snow depth - (cm)
* CWB - climatic water balance - (mm = liters/m2)
* FFAV - average wind speed - (m/s) (just for ERA INTERIM + OPE)
* VAPAV - avearge water vapour pressure - (hPa) (just for ERA INTERIM + OPE)
Addictional fields:
* LATITUDE - (Deg.decDeg)
* LONGITUDE - (Deg.decDeg)
* YEAR - (yyyy)
* MONTH - (mm)
* DEKAD - [1-2-3]
* ACQDATE - (dd/mm/yyyy)
```

A minimal HDR file according to the FOODSEC Meteodata projection/resolution and the desired ROI is created:

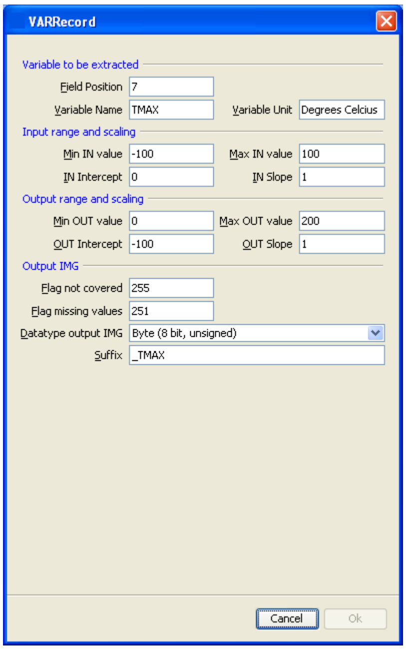
```
ENVI
samples = 344
lines = 292
bands = 1
data type = 1
map info = {Geographic Lat/Lon, 1, 1, -26, 38, 0.25, 0.25}
```



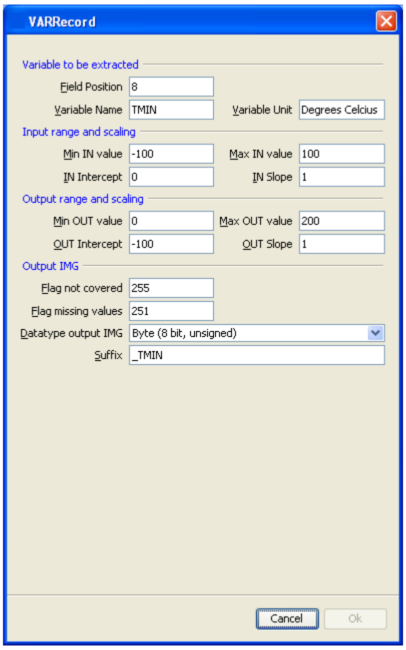
example: SPM to import and convert FOODSEC Meteodata



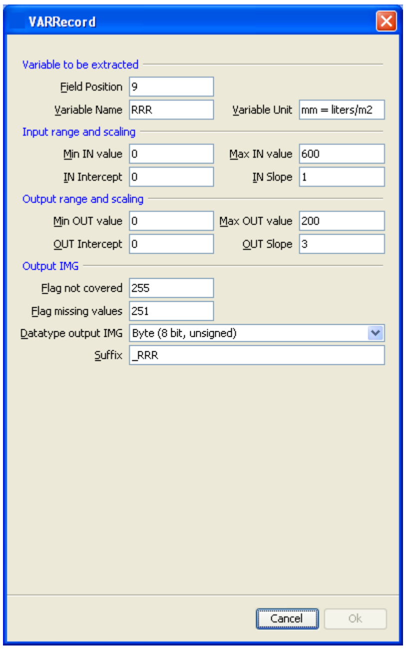
specification for 'TAV' variable



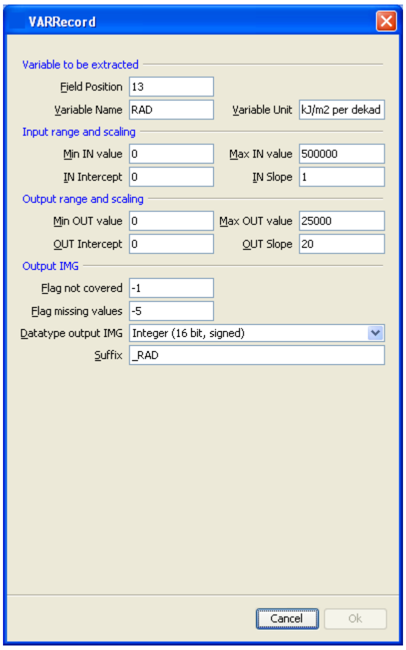
specification for 'TMAX' variable



specification for 'TMIN' variable



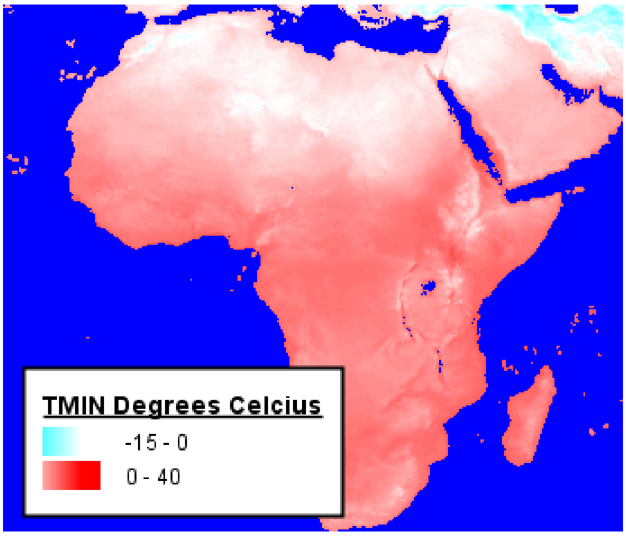
specification for 'RRR' variable



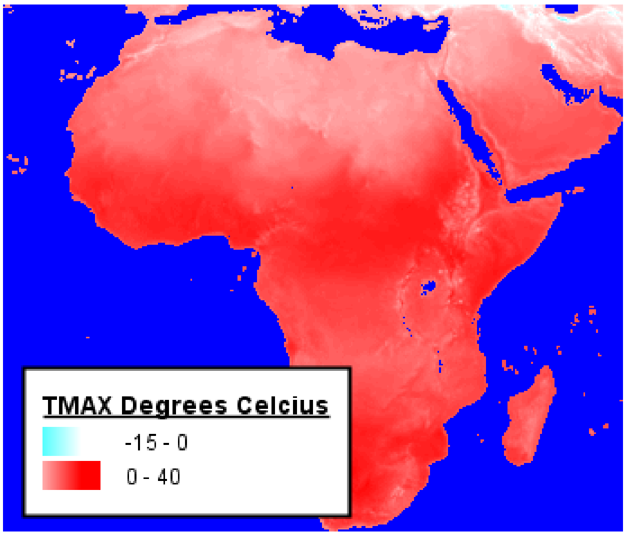
specification for 'RAD' variable

example: Meto to IMG toolfor FOODSEC Meteodata

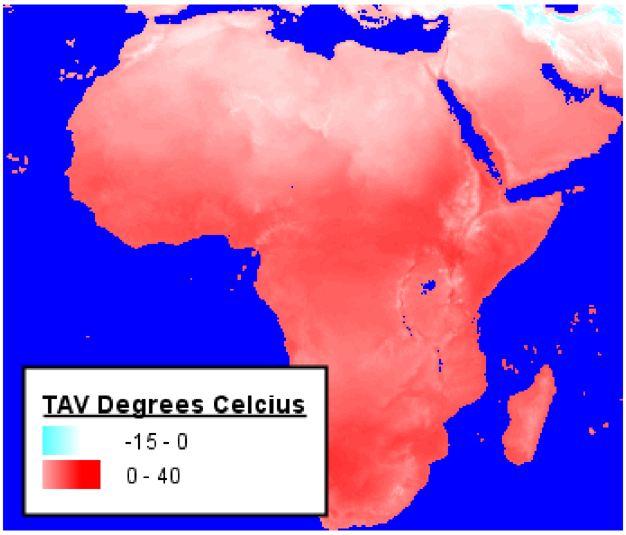
Results:



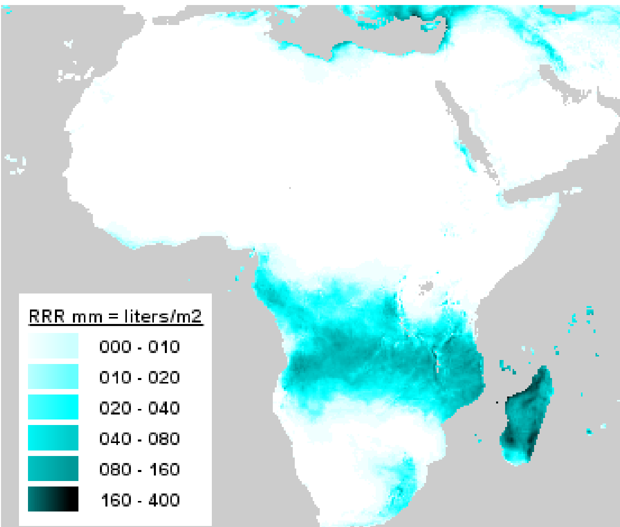
minimum temperature2012 Jan dekad 1



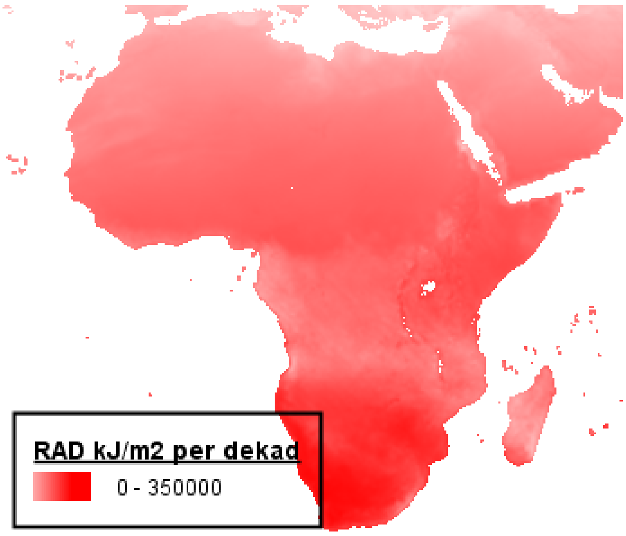
maximum temperature2012 Jan dekad 1



average temperature 2012 Jan dekad 1



precipitation sum 2012 Jan dekad 1



global radiation sum 2012 Jan dekad 1

3.39. Generic Import

Goal

File conversions from images in various formats such as GeoTIFF (*.tif) NetCDF (*.nc) and Hierarchical Data Format (*.hdf) to ENVI raster format.

Besides the actual format conversion additional (optional) tools are integrated, enabling retrieval of information, extracting an ROI, scaling/reclassification and adapting the resulting IMG HDR in one tool.

Parameters

The format conversion is based on the gdal_translate utility from GDAL (Geospatial Data Abstraction Library from the Open Source Geospatial Foundation).

The gdal_translate syntax used is as follows:

```
gdal_translate -ot {Byte, Int16, Int32, Float32} -b band -of ENVI
               source_dataset destination_dataset
```

Only a single band of a single dataset can be extracted per run.

In case the input file does not contain sub-datasets, it will function directly as source dataset.

In case the input file does contain sub-datasets (e.g. Proba V HDF5 files, or Copernicus NetCDF files), the source dataset must be further specified via a prefix and suffix: GDAL will typically identify sub-datasets via the format ***PREFIX:full_filename:SUFFIX***.

example: Copernicus NetCDF NDVI:
HDF5:C:\Data\c_gls_NDVI_200001010000_GLOBE_VGT_V2.2.1.nc:***//NDVI***

example: Proba V - HDF5:
HDF5:C:\Data\PROBAV_S10_TOC_X20Y03_20140601_1KM_V101.HDF5:***//LEVEL3/NDVI/NDVI***

The selected band, the sub-dataset and the output data type parameters are captured in the top part of the UI as:

- the band to extract;
- whether or not a sub-dataset must be selected, and if so, its prefix and suffix
- the conversion data type: Byte(8 bit, unsigned), Integer (16 bit, signed), Long (32 bit, signed) or Float (32 bit).

The parameters of the additional steps are those of the corresponding tools (Extract Band/ROI, Scaling and Reclassification and Adapt HDR). Beware: the selected conversion data type relates to the gdal_translate step. In case an additional Scaling and Reclassification step is selected, its input datatype must match the selected conversion data type.

Remark:

The gdal_translate utility supports a number of different formats. For each of these formats different limitations apply. Therefore, whether a certain format can be translated into IMG/HDR, and which tags will be available in the created HDR file, depends on the specific format.

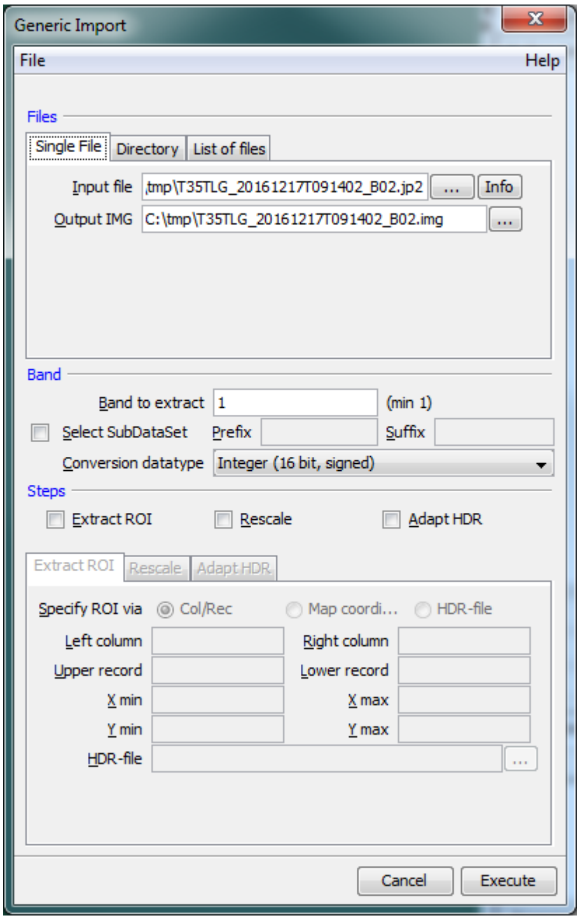
To give the user an indication if gdal_translate recognises the input file, whether it contains sub datasets and other meta data, there is the "Info" button on the panel (only active in case of "Single File"). This button opens a window which shows the result of another GDAL utility: gdalinfo. In case gdal recognises the current input file, the metadata from the file (and optionally of the selected sub dataset) will appear in this window. The content of this info-window can be saved as a text file via its File menu.

The gdalinfo syntax used is as follows:

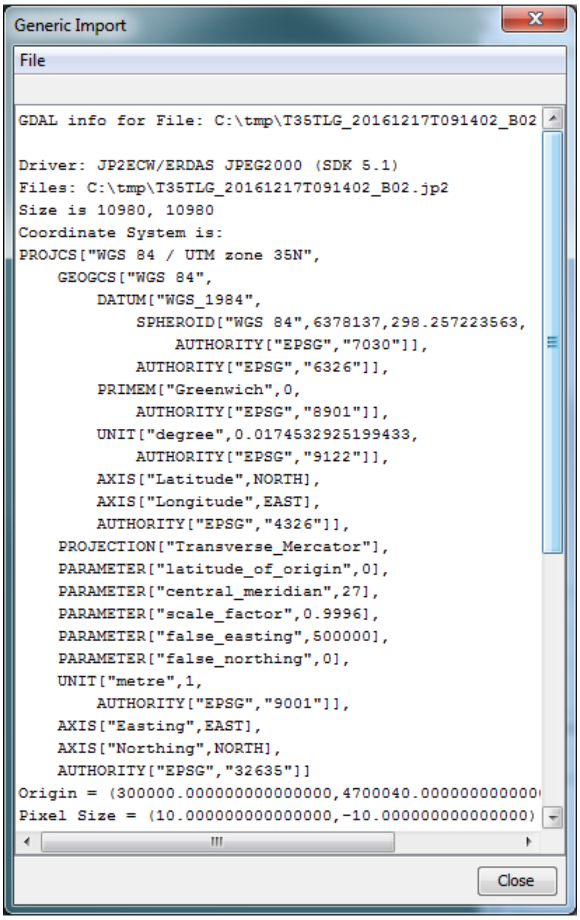
```
gdalinfo source_dataset
```

In case a sub-datasets is specified, gdalinfo is actually executed twice:
1) with source_dataset : ***input_filename***
2) with source_dataset: ***prefix: input_filename:suffix***

Tool

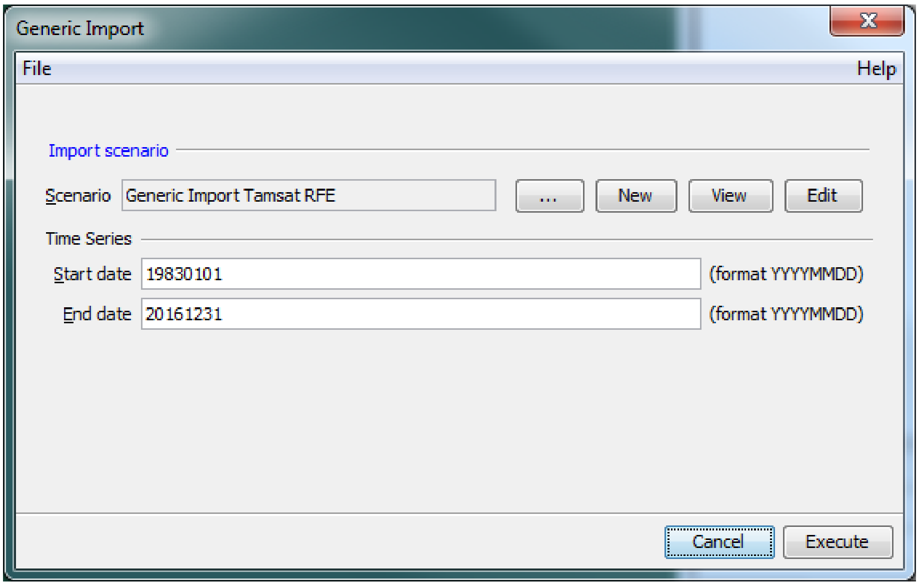


Generic Import Tool example



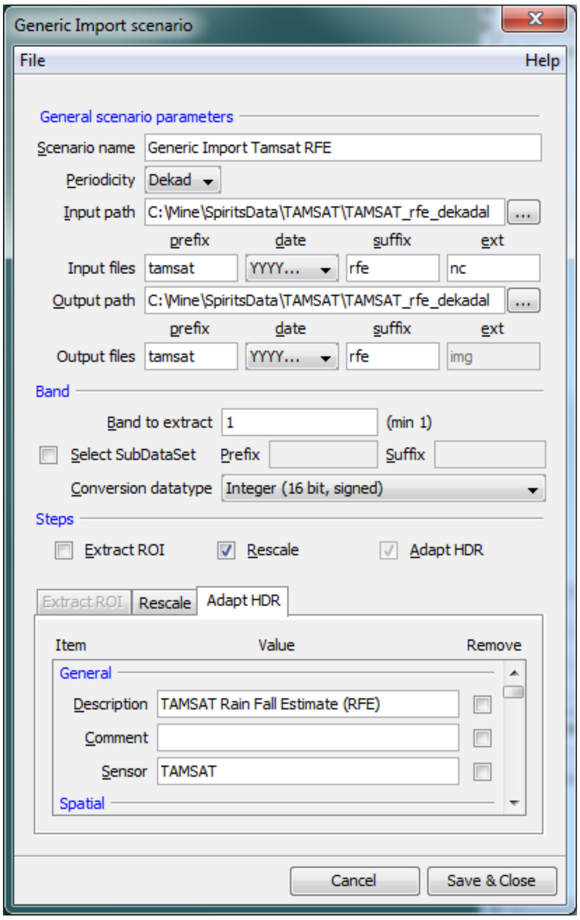
Generic Import Tool info panel example

Time Series



Generic Import Time Series example

Scenario



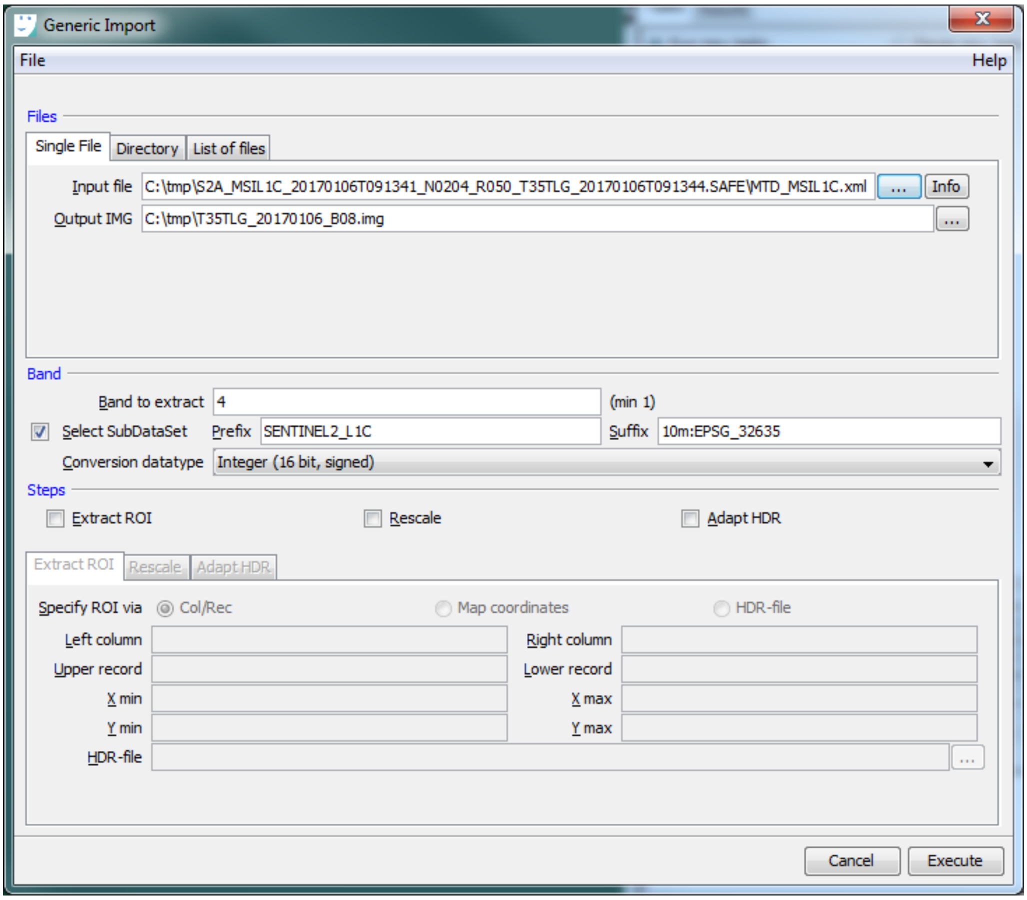
Generic Import Scenario example

Remark: since Scenario's do not specify one explicit file, there is no "Info" button. Typically one would perform some single-file experiments with the Tool anyway before setting up an actual scenario.

Import Sentinel 2 L1C example

Sentinel 2 Level 1C data can be downloaded e.g. from the Copernicus Open Access Hub (https://scihub.copernicus.eu/). This data comes in a .zip archive containing a complex folder structure of directories and files with meta data, vectors, JPEG2000 images,...of which the format and specifications can be found at the Copernicus site. To obtain the actual bands, one can navigate in this structure, locate the relevant JPEG2000 images, and convert them.

Alternatively, GDAL has a dedicated Sentinel 2 driver, capable of interpreting the xml file at the root of the structure, and allowing the data to be approached as bands in sub-datasets.



example: import B08 band from Sentinel 2 L1C tile via MTD_MSIL1C.xml metafile

```
GDAL info for File:
C:\tmp\S2A_MSIL1C_20170106T091341_N0204_R050_T35TLG_20170106T091344.SAFE\MTD_MSIL1C.xml

Driver: SENTINEL2/Sentinel 2
...
Metadata:
  CLOUD_COVERAGE_ASSESSMENT=99.66830000000002
  SPECIAL_VALUE_SATURATED=65535
...
Subdatasets:
SUBDATASET_1_NAME=SENTINEL2_L1C:C:\tmp\S2A_MSIL1C_20170106T091341_N0204_R050_T35TLG_20170
106T091344.SAFE\MTD_MSIL1C.xml:10m:EPSG_32635
SUBDATASET_1_DESC=Bands B2, B3, B4, B8 with 10m resolution, UTM 35N
SUBDATASET_2_NAME=SENTINEL2_L1C:C:\tmp\S2A_MSIL1C_20170106T091341_N0204_R050_T35TLG_20170
106T091344.SAFE\MTD_MSIL1C.xml:20m:EPSG_32635
SUBDATASET_2_DESC=Bands B5, B6, B7, B8A, B11, B12 with 20m resolution, UTM 35N
...
```

part of the info Panel contents

via the info panel the sub-dataset prefix and suffix can be found:

```
SUBDATASET_1_NAME=SENTINEL2_L1C:C:\tmp\S2A_...44.SAFE\MTD_MSIL1C.xml:10m:EPSG_32635
```

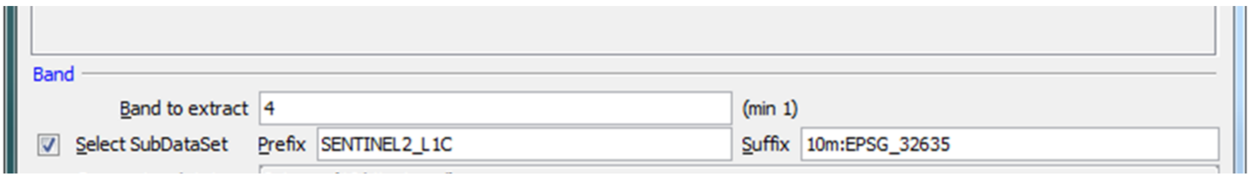
once these are filled out in the tool, the specific information of the selected sub-dataset also becomes available in the info panel:

```
...
GDAL info for SubDataset:
SENTINEL2_L1C:C:\tmp\S2A_MSIL1C_20170106T091341_N0204_R050_T35TLG_20170106T091344.SAFE\MT
D_MSIL1C.xml:10m:EPSG_32635

Driver: SENTINEL2/Sentinel 2
...
Size is 10980, 10980
Coordinate System is:
PROJCS["WGS 84 / UTM zone 35N",
    GEOGCS["WGS 84",
...
Band 1 Block=128x128 Type=UInt16, ColorInterp=Red
  Description = B4, central wavelength 665 nm
...
Band 2 Block=128x128 Type=UInt16, ColorInterp=Green
  Description = B3, central wavelength 560 nm
...
Band 3 Block=128x128 Type=UInt16, ColorInterp=Blue
  Description = B2, central wavelength 490 nm
...
Band 4 Block=128x128 Type=UInt16, ColorInterp=Undefined
  Description = B8, central wavelength 842 nm
  Overviews: 5490x5490, 2745x2745, 1372x1372, 686x686, 343x343, 171x171
  Metadata:
    BANDNAME=B8
    BANDWIDTH=115
    BANDWIDTH_UNIT=nm
    SOLAR_IRRADIANCE=1036.39
    SOLAR_IRRADIANCE_UNIT=W/m2/um
    WAVELENGTH=842
    WAVELENGTH_UNIT=nm
  Image Structure Metadata:
    NBITS=15
```

part of the info Panel contents

eventually, the bands information shows that to obtain the B8 or 842 nm band, we need to select band 4 in the selected sub-dataset:

```
Band 4 Block=128x128 Type=UInt16, ColorInterp=Undefined
  Description = B8, central wavelength 842 nm
  Overviews: 5490x5490, 2745x2745, 1372x1372, 686x686, 343x343, 171x171
```

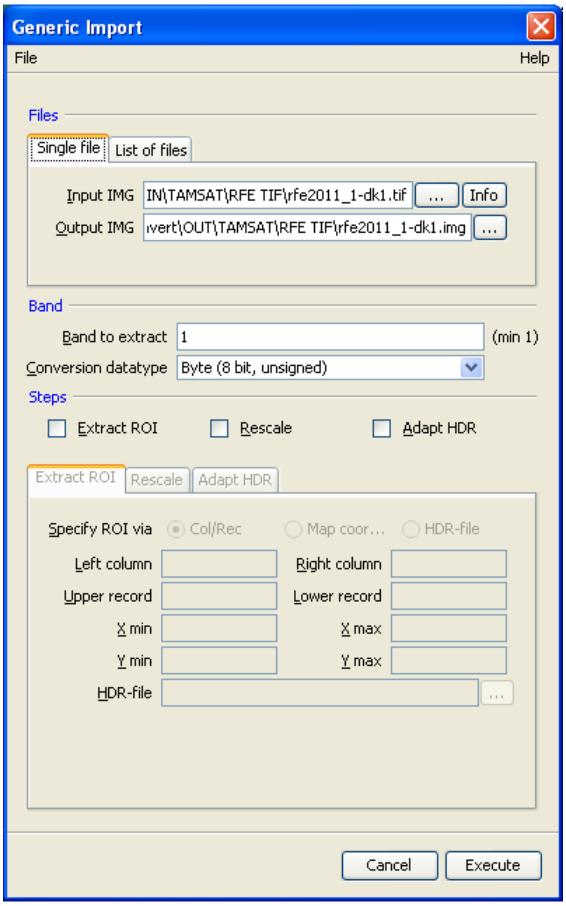


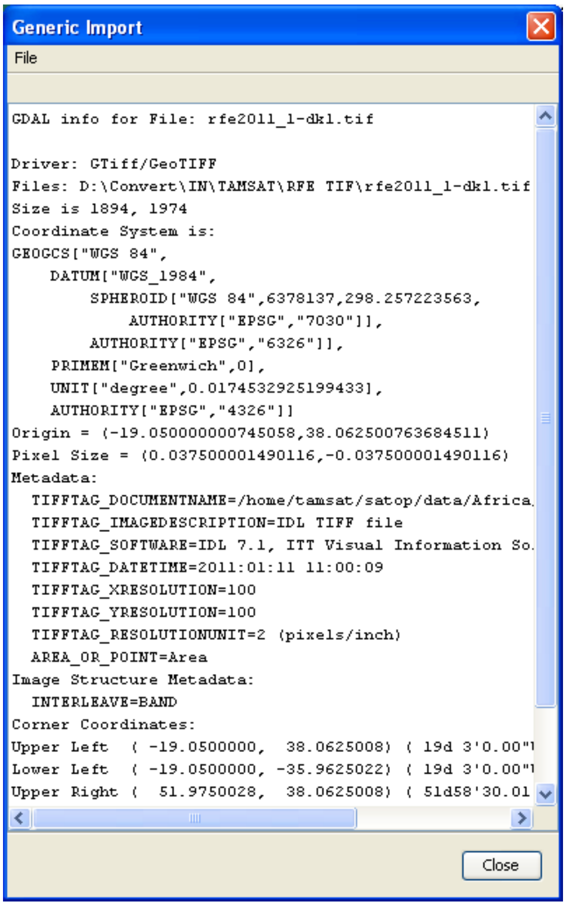
example: import B08 band from Sentinel 2 L1C via

band ***4*** of sub-dataset indicated

with prefix ***SENTINEL2_L1C*** and suffix ***10m:EPSG_32635***

Import TAMSAT RFE GeoTIFF example





example: import TAMSAT RFE GeoTIFF file info Panel

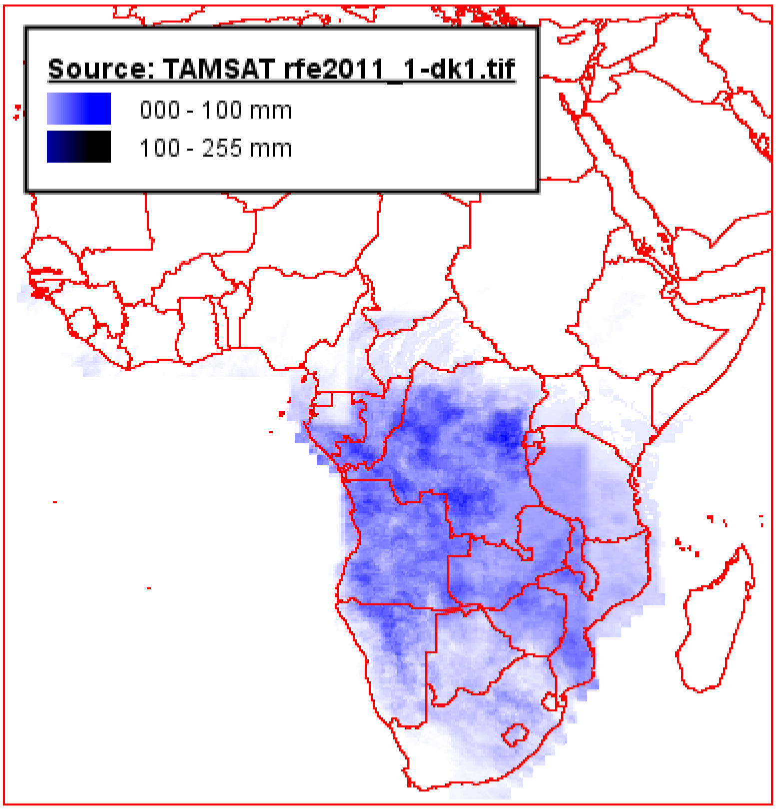
The information obtained from the Info Panel suggests:

- the file format is recognized (thus can be converted);
- the coordinate system could be retrieved (thus the HDR will contain a map info entry).

```
Driver: GTiff/GeoTIFF
Files: D:\Convert\IN\TAMSAT\RFE TIF\rfe2011_1-dk1.tif
Size is 1894, 1974
Coordinate System is:
GEOGCS["WGS 84",
    DATUM["WGS_1984",
        SPHEROID["WGS 84",6378137,298.257223563,
            AUTHORITY["EPSG","7030"]],
        AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433],
    AUTHORITY["EPSG","4326"]]
Origin = (-19.050000000745058,38.062500763684511)
...
Band 1 Block=1894x4 Type=Byte, ColorInterp=Gray
```

part of the info Panel contents

Result:



imported IMG from TAMSAT GeoTIFF rfe2011_1-dk1.tif

```
ENVI
description = {
D:\Convert\OUT\TAMSAT\RFE TIF\rfe2011_1-dk1.img}
samples = 1894
lines    = 1974
bands    = 1
header offset = 0
file type = ENVI Standard
data type = 1
interleave = bsq
byte order = 0
map info = {Geographic Lat/Lon, 1, 1, -19.0500000007451, 38.0625007636845,
0.0375000014901161, 0.0375000014901161,WGS-84}
band names = {Band 1}
```

imported IMG HDR

Remark: for further processing (e.g. Extract RUM) the imported IMG HDR should be adapted to include e.g. the values, flags, date and days entries. This could be done directly via the integrated Adapt HDR option or afterwards by the Adapt HDR tool, the View HDR utility or any ASCII editor.

Import TAMSAT RFE NetCDF example

The information obtained from the Info Panel for the rfe2012_01-dk1.nc NetCDF file shows:

- the file format is recognized (thus can be converted);
- the coordinate system could not be retrieved;
- the file has an integer data type (Int16) and uses -99 for no data flag ;
- coordinates and resolution are available (latmin, latmax, ... lonres);

```
Driver: netCDF/Network Common Data Format
Files: D:\Convert\IN\TAMSAT\RFE NC\rfe2012_01-dk1.nc
Size is 1894, 1974
Coordinate System is `'
Metadata:
  NC_GLOBAL#title=TAMSAT Rain Fall Estimate (RFE)
  NC_GLOBAL#institution=TAMSAT Research Group, Meteorology Department,
University of Reading, UK
...
  NC_GLOBAL#latmin=-35.9625
  NC_GLOBAL#latmax=38.025
  NC_GLOBAL#lonmin=-19.0125
  NC_GLOBAL#lonmax=51.975
  NC_GLOBAL#latres=0.0375
  NC_GLOBAL#lonres=0.0375
  rfe#units=mm
...
  lat#long_name=latitude
  lat#standard_name=latitude
  lat#units=degrees_north
  lat#axis=Y
  lon#long_name=longitude
  lon#standard_name=longitude
  lon#units=degrees_east
  lon#axis=X
  time#long_name=time
  time#units=days since 2012-01-01 0:0:0
  time#day_begins=06:15
...
Band 1 Block=1894x1 Type=Int16, ColorInterp=Undefined
  NoData Value=-99
...
```

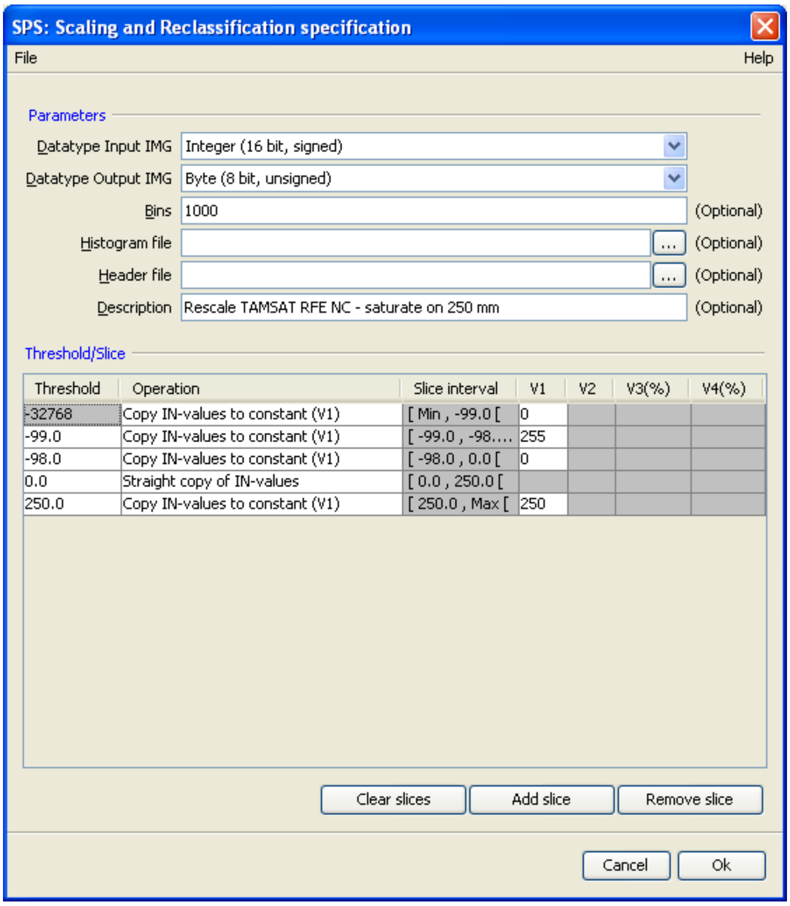
part of the info Panel contents

Executing the tool with only the translation step produces an IMG with a 'bare' HDR

```
ENVI
samples = 1894
lines   = 1974
bands   = 1
header offset = 0
file type = ENVI Standard
data type = 2
interleave = bsq
byte order = 0
```

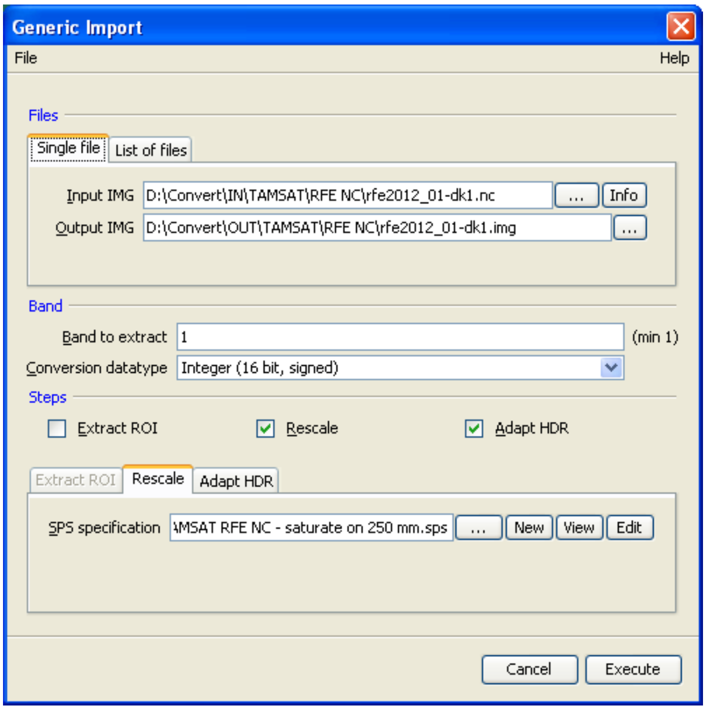
To enable further processing the Adapt HDR step is needed. In this example the data will be also be rescaled to byte type.

For the rescale step an SPS is needed:



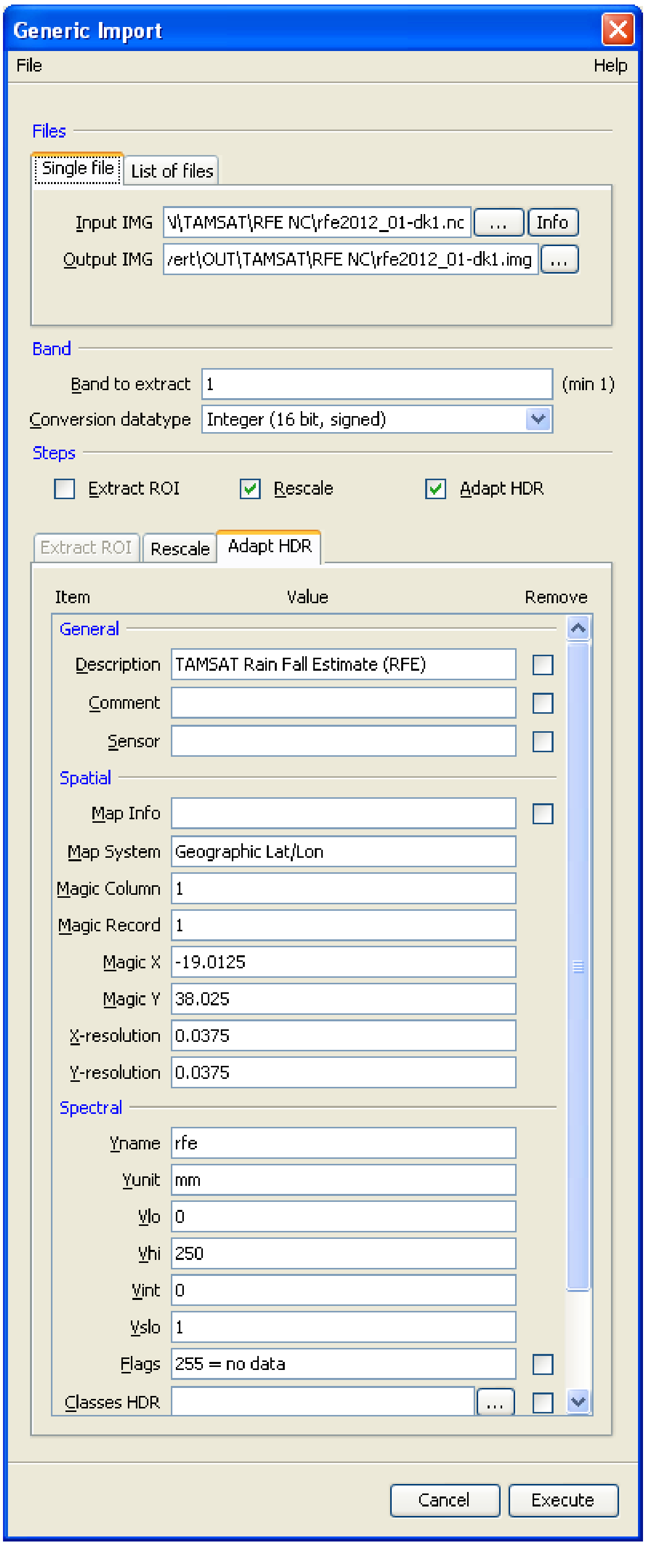
- all negative values except the 'no data' value (-99) are rescaled to 0;
- the -99 flag is scaled to 255;
- positive values up to 249 as kept as they are;
- positive values above 249 are rescaled to 250

SPS file used in the Rescaling step



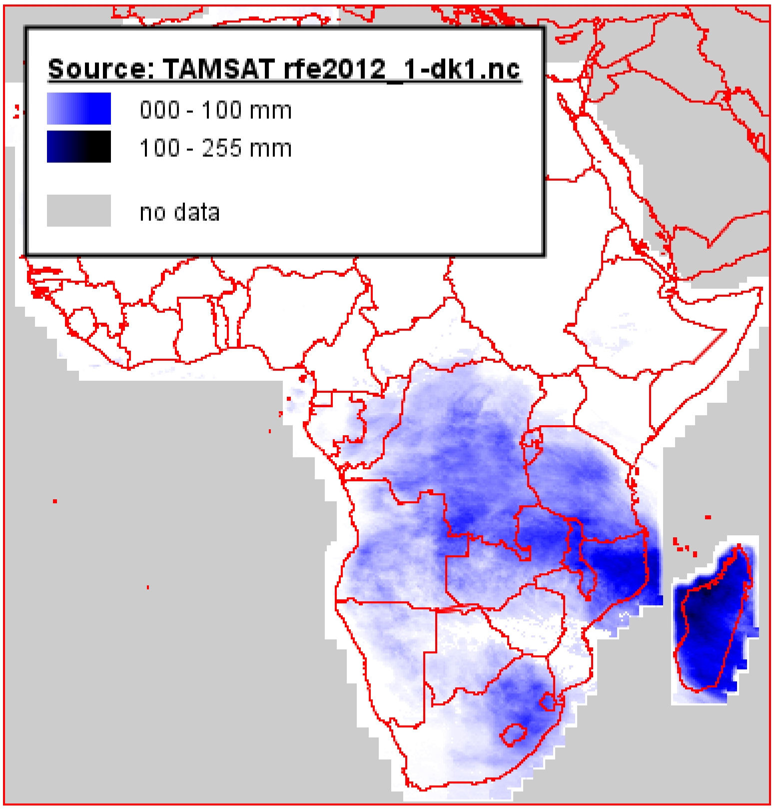
Generic Import tool - Rescaling panel

In the Adapt HDR step the map info, the values and the flags entries are filled out according to information obtained from the Info Panel and accounting for the effects of the rescale step.



Generic Import tool - Adapt HDR panel

Result:

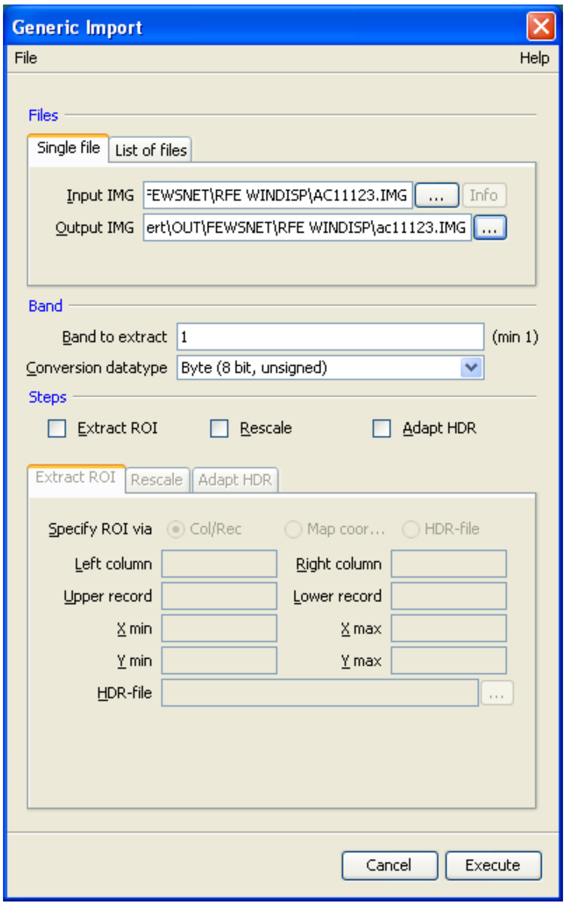


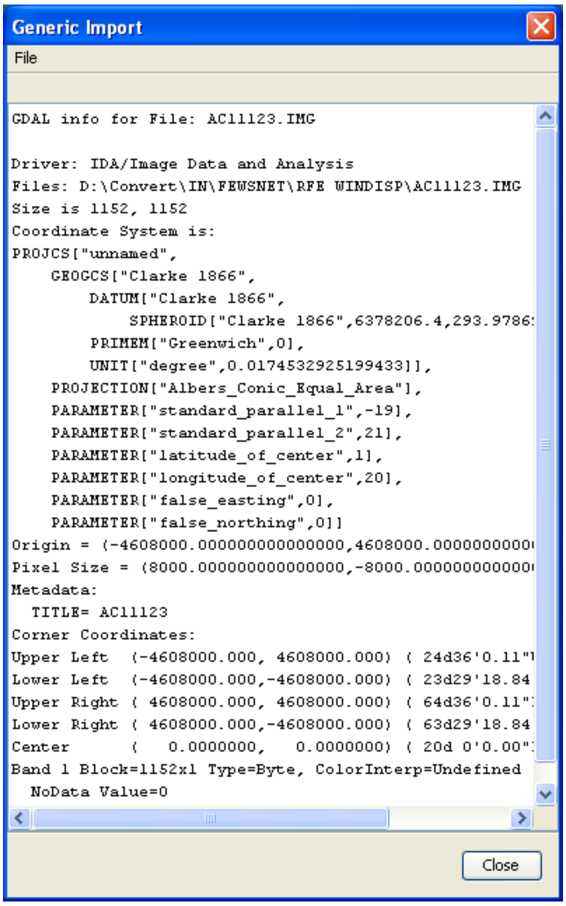
imported IMG from TAMSAT NetCDF rfe2012_1-dk1.nc

```
ENVI
description = {TAMSAT Rain Fall Estimate (RFE)}
samples = 1894
lines = 1974
bands = 1
header offset = 0
file type = ENVI standard
data type = 1
interleave = bsq
map info = {Geographic Lat/Lon, 1, 1, -19.0125, 38.025, 0.0375, 0.0375}
values = {rfe, mm, 0, 250, 0, 250, 0, 1}
flags = {255 = no data}
program = {IMGscale.exe (V912) + HDRadapt.exe (V912)}
```

imported IMG HDR

Import FEWS NET RFE Windisp example





example: import FEWSNET RFE Windisp file info Panel

The information obtained from the Info Panel suggests:

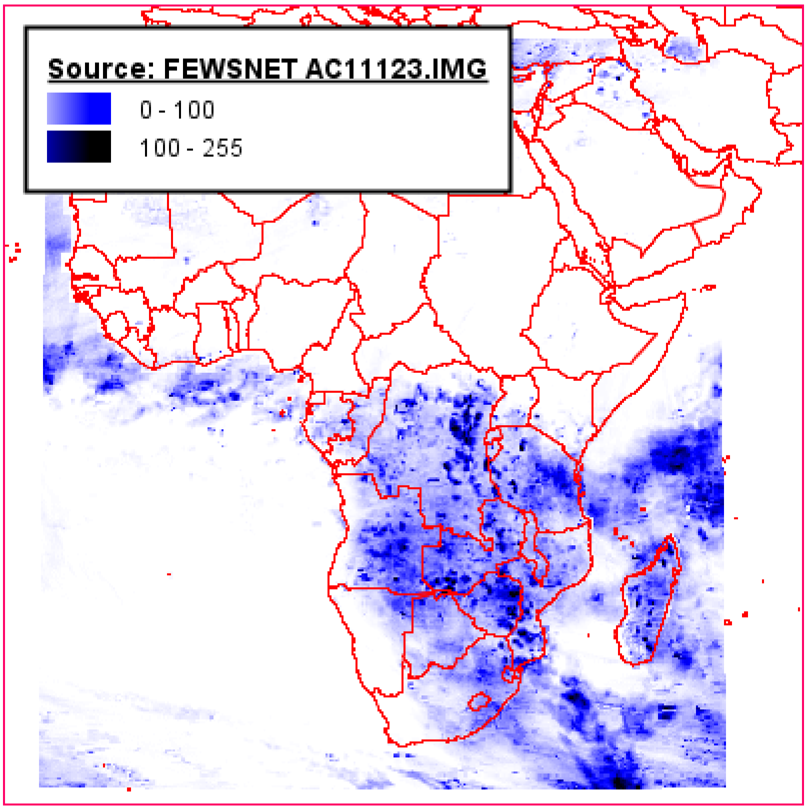
- the file format is recognized (thus can be converted);
- the coordinate system could be retrieved (thus the HDR will contain a map info entry).

```
Driver: IDA/Image Data and Analysis
Files: D:\Convert\IN\FEWSNET\RFE WINDISP\AC11123.IMG
Size is 1152, 1152
Coordinate System is:
PROJCS["unnamed",
    GEOGCS["Clarke 1866",
        DATUM["Clarke 1866",
            SPHEROID["Clarke 1866",6378206.4,293.9786982138966]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433]],
    PROJECTION["Albers_Conic_Equal_Area"],
    PARAMETER["standard_parallel_1",-19],
    PARAMETER["standard_parallel_2",21],
    PARAMETER["latitude_of_center",1],
    PARAMETER["longitude_of_center",20],
    PARAMETER["false_easting",0],
    PARAMETER["false_northing",0]]
Origin = (-4608000.000000000000000,4608000.000000000000000)
Pixel Size = (8000.000000000000000,-8000.000000000000000)
```

```
Metadata:
  TITLE= AC11123
Corner Coordinates:
Upper Left  (-4608000.000, 4608000.000) ( 24d36'0.11"W, 43d42'41.48"N)
Lower Left  (-4608000.000,-4608000.000) ( 23d29'18.84"W, 42d20'13.06"S)
Upper Right ( 4608000.000, 4608000.000) ( 64d36'0.11"E, 43d42'41.48"N)
Lower Right ( 4608000.000,-4608000.000) ( 63d29'18.84"E, 42d20'13.06"S)
Center      (    0.0000000,    0.0000000) ( 20d 0'0.00"E,  1d 0'0.00"N)
Band 1 Block=1152x1 Type=Byte, ColorInterp=Undefined
  NoData Value=0
```

part of the info Panel contents

Result:



imported IMG from FEWSNET Windisp AC11123.IMG

```
ENVI
description = {
D:\Convert\OUT\FEWSNET\RFE WINDISP\ac11123.IMG}
samples = 1152
lines   = 1152
bands   = 1
header offset = 0
file type = ENVI Standard
data type = 1
interleave = bsq
byte order = 0
map info = {Albers Conical Equal Area, 1, 1, -4608000, 4608000, 8000, 8000}
projection info = {9, 6378206.4, 6356510.248412312, 1, 20, 0, 0, -19, 21,
Albers Conical Equal Area}
band names = {Band 1}
```

imported IMG HDR

The information obtained from the Info Panel for the 20110501_DMP.HDF file shows:

- the file format is recognized (thus can be converted);
- the coordinate system could not be retrieved;
- the file has an integer data type (Int16);

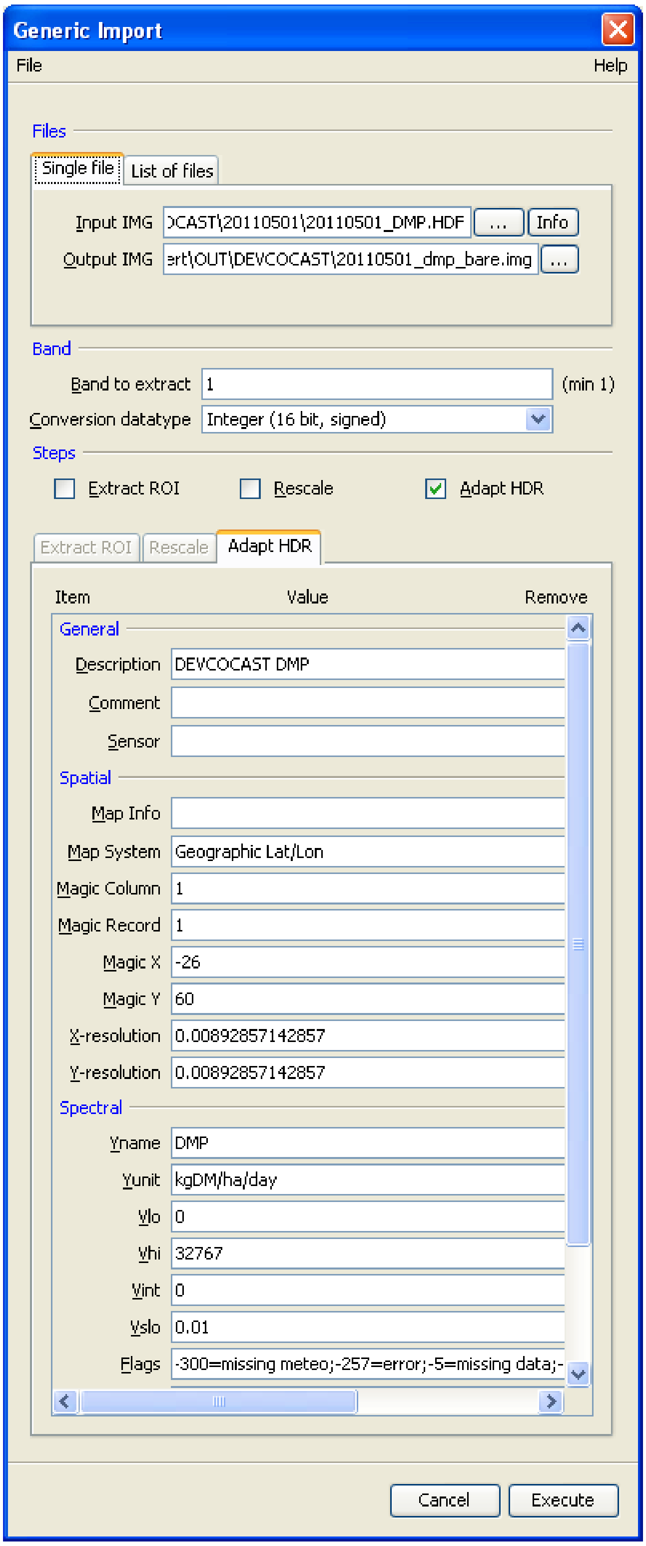
```
GDAL info for File: 20110501_DMP.HDF

Driver: HDF4Image/HDF4 Dataset
Files: D:\Convert\IN\DEVCOCAST\20110501\20110501_DMP.HDF
Size is 9633, 8177
Coordinate System is `'
Corner Coordinates:
Upper Left  (    0.0,    0.0)
Lower Left  (    0.0, 8177.0)
Upper Right ( 9633.0,    0.0)
Lower Right ( 9633.0, 8177.0)
Center      ( 4816.5, 4088.5)
Band 1 Block=9633x10 Type=Int16, ColorInterp=Gray
```

part of the info Panel contents

Executing the tool with only the translation step produces an IMG with a 'bare' HDR

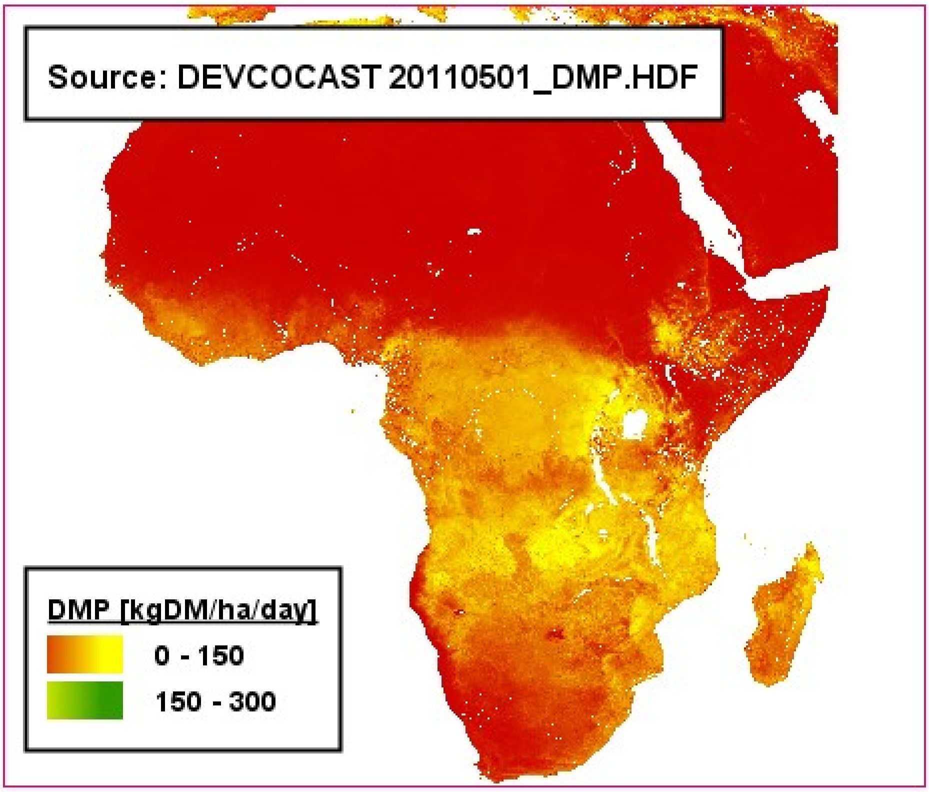
```
ENVI
samples = 9633
lines   = 8177
bands   = 1
header offset = 0
file type = ENVI Standard
data type = 2
interleave = bsq
byte order = 0
```

To enable further processing the Adapt HDR step is needed, filling out the map info, the values and the flags entries according to information obtained from the XML file (20110501_DMP_PROD_DESCR.XML) accompanying the 20110501_DMP.HDF file.



Generic Import tool - Adapt HDR panel

Result:



imported IMG from DevCoCast HDF4 20110501.HDF

```
ENVI
description = {DEVCOCAST DMP}
samples = 9633
lines = 8177
bands = 1
header offset = 0
file type = ENVI Standard
data type = 2
interleave = bsq
byte order = 0
map info = {arbitrary, 1, 1, -26, 60, 0.00892857143, 0.00892857143}
values = {DMP, kgDM/ha/day, 0, 32767, 0, 32767, 0, 0.01}
flags = {-300=missing meteo;-257=error;-5=missing data;-4=cloud;-3=snow;-
2=sea;-1=back}
program = {HDRadapt.exe (V912)}
```

imported IMG HDR

3.40. Convert HDF5 to ENVI

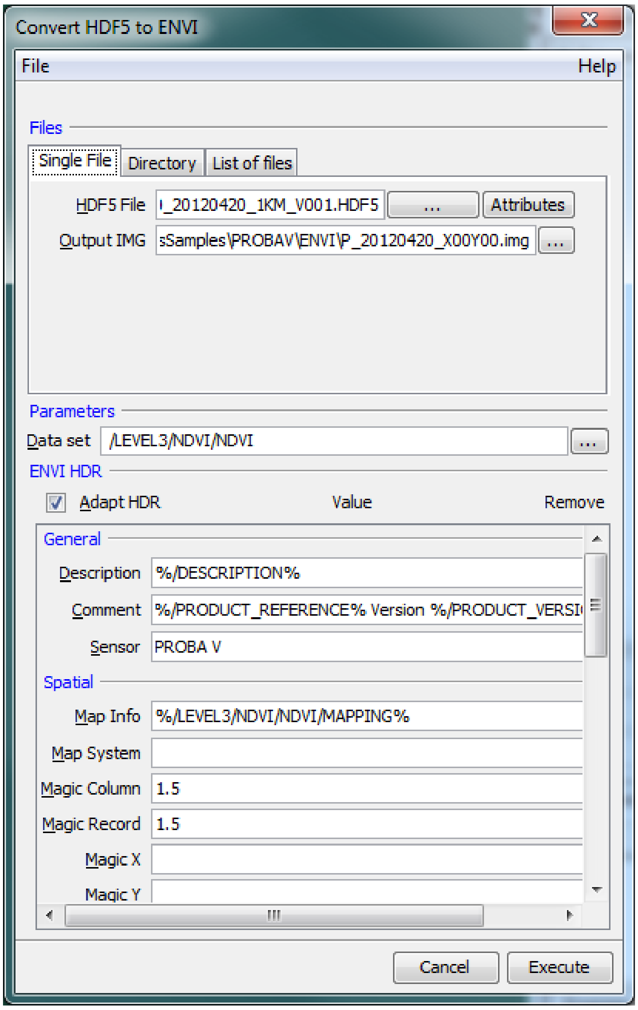
Goal

Extract and convert datasets from HDF5 files to ENVI IMGs.

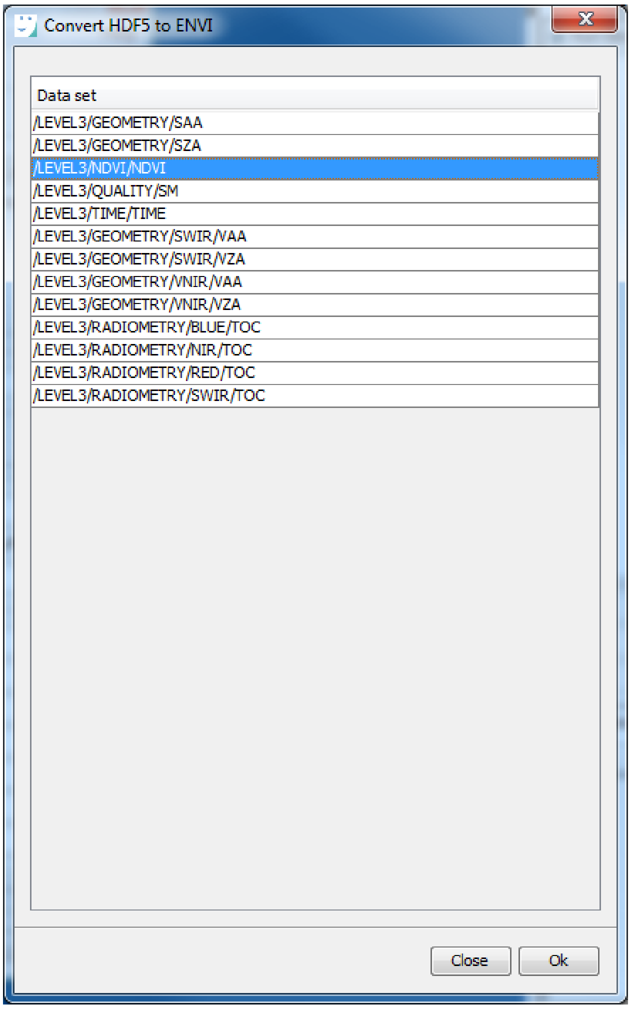
Parameters

- the HDF5 file(s) to extract the dataset from;
- the full name of the dataset to be extracted;
- option to modify/complete the generated HDR file as an additional step. The parameters of this step are those of the Adapt HDR tool.

Tool



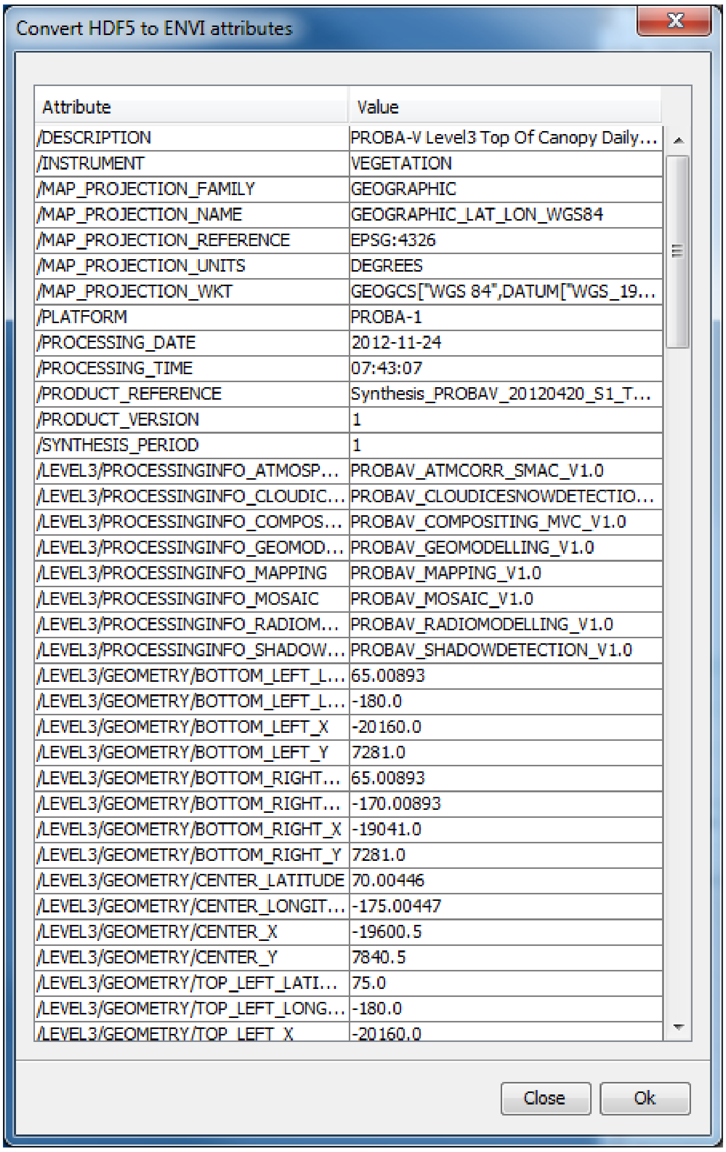
Convert HDF5 to ENVI Tool example



HDF5 Dataset selection panel

In "Single File" input mode, the dataset name can be selected via the button next to the "Data set" parameter field. An additional panel appears showing the names of the datasets found in the selected HDF5 file. In other input modes, the dataset name is considered a string, its presence in the files will only be checked at runtime.

Also only in "Single File" input mode, the attributes found in the selected HDF5 file can be examined via the "Attributes" button.



HDF5 file attributes panel

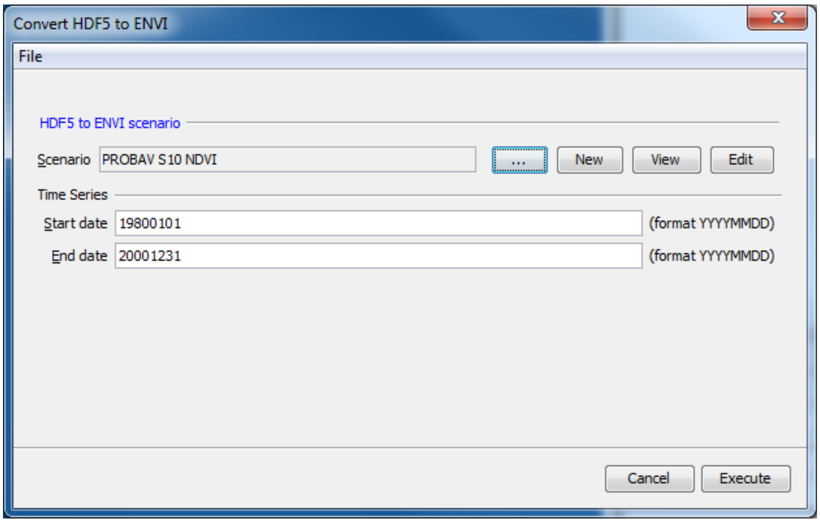
Remark:

- the attributes from the HDF5 file can be used as parameters in the Adapt HDR step. Most fields in the Adapt HDR panel will accept a parameterized string: a mixture of constant text and parameters representing the values of the attributes from the HDF5 file. These parameters are specified by enclosing the full attribute name (as shown in the panel) with "%" signs. e.g. :

parameterized string	result (used during Adapt HDR execution)
`%/INSTRUMENT%`	`VEGETATION`
`Image from %/PLATFORM% on %/PROCESSING_DATE%`	`Image from PROBA-1 on 2012-11-24`

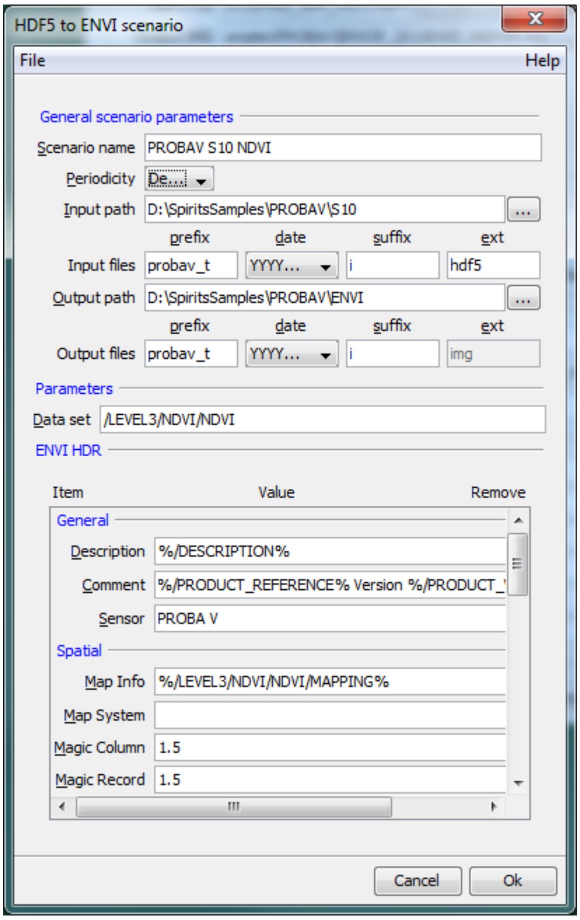
- once the attributes panel is activated it can remain on the screen while selecting other input modes. Its contents however relates to the file specified in the "Single File" subpanel, not to files selected in the "Directory" or "List of files" subpanels.

Time Series



Convert HDF5 to ENVI Time Series example

Scenario



Convert HDF5 to ENVI Scenario example

3.41. Convert Periodicity

Goal

Convert time series of images (vegetation indices (VI) such as NDVI and SAVI or biophysical state variables such as fAPAR, fCover or LAI) with any frequency between 1 and 30 days to daily, dekadel or monthly series.

Parameters

The tool is based on the same algorithms as the Smooth tool, hence apart from the specifications of the input data periodicity, the parameters -as far as applicable- are the same as those for the Smooth tool.

specification of the input periodicity

- the number of days per period: the input periodicity can range between 1 and 30 days;

Basic assumption is that every annual series restarts at January 1. The start day of every next period (or composite) is then simply obtained by adding as this number of days to the previous start date, until the end of the year is surpassed.

- account for leap years;

Leap years can raise problems. Normally, the 29th of February is accounted for in the enumeration of the time series, but this particular day can be skipped as if it wouldn't exist.

- periods per year;

The last period in the year mostly comprises a deviate number of days. And as shown in table below, for most periodicities two possible solutions are equally possible. Via number of periods per year, the number of days in the last period can be derived, for normal and leap years.

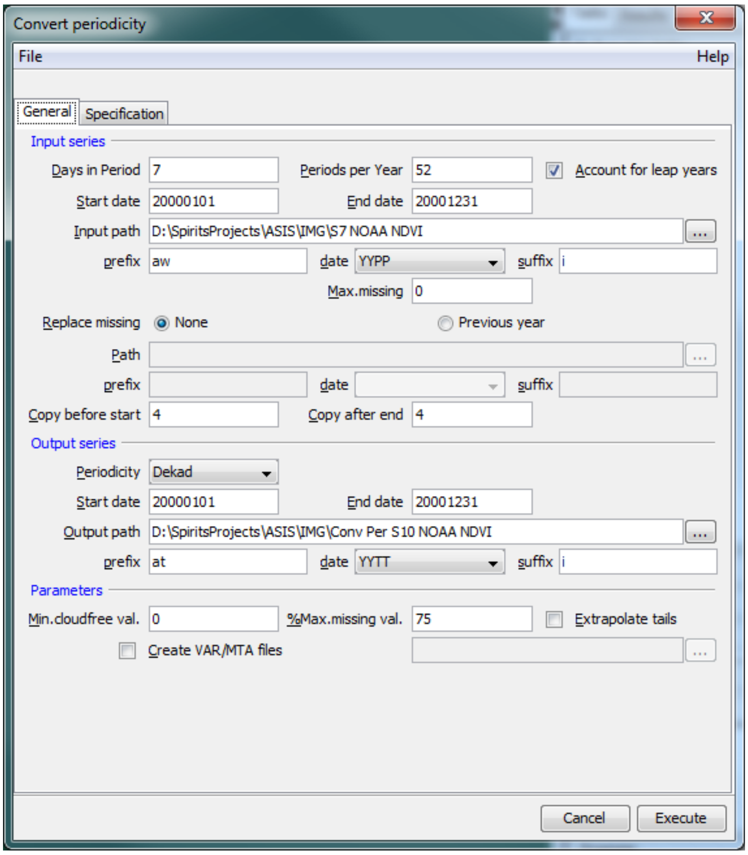
Per	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
NpY1	**365**	182	121	91	**73**	60	52	45	40	**36**	33	30	28	26	24	22	21	20	19	18	17	16	15	15	14	14	13	13	12	**12**
Dl1	**1**	3	5	5	**5**	11	8	13	14	**11**	13	17	14	15	20	29	25	23	23	25	29	35	43	29	40	27	41	29	46	**31**
NpY2	**365**	183	122	92	**73**	61	53	46	41	**36**	34	31	29	27	25	23	22	21	20	19	18	17	16	16	15	15	14	14	13	**12**
Dl2	**1**	1	2	1	**5**	5	1	5	5	**11**	2	5	1	1	5	13	8	5	4	5	8	13	20	5	15	1	14	1	17	**31**

For most periodicities (number of days per period: 1-30) two solutions are possible for the number of periods per year (NpY1, NpY2). Dl1 and Dl2 indicate the number of days in the last period (NB: For leap years -when accounted for- Dl1 and Dl1 should be incremented by 1). The greyed periodicities are exceptions because NpY1=NpY2 (number of days per period = 1/5) or because they are treated differently (number of days per period = 10/30).

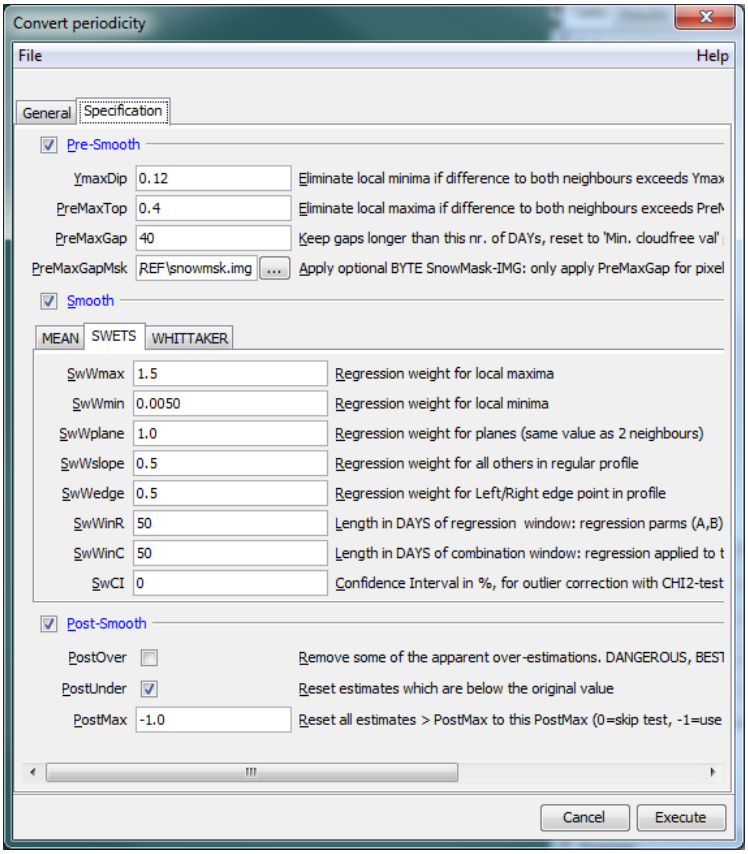
Periodicity 10 follows the common convention where each year comprises 36 periods ("dekads"), always three per month and starting on days 1/11/21. The first two dekads always count 10 days, while the third one has a variable number of days (8-11). In such case, the number of periods/dekads per year is fixed to 36, and the leap years are always accounted for.

Periodicity 30 follows the civil year convention where each year comprises 12 months, each with its specific number of days. In this case, the number of periods/months per year is fixed to 12, and the leap years are always accounted for.

Tool



Convert periodicity tool - general parameters example



Convert periodicity tool - specifications example

3.42. Flag VGT NDVI

Goal
Add flags from status mask IMG and land mask IMG to SPOT-VGT NDVI IMG.

Context
NDVI and Status Mask (SM) SPOT-VGT images can be imported from e-STation, DevCoCast and CTIV. After conversion one ends up with two IMGs (NDVI, SM) in the correct ENVI-GLIMPSE format but with the original contents as provided by CTIV. There are two problems:

- For the further analysis it is annoying that the full information is spread over two separate IMGs. It would be easier if the most significant information of the SM (snow, cloud, sea, error) would be transferred as flags to the NDVI-image.
- The Land/Sea bit in the SM is not really useful, for two reasons:
- Along the coasts, a wide rim of sea pixels is marked as land.
- During the local winter the high latitude pixels (no light) have value 0 in all the S10-images, incl. the SM. Thus in that case, the land/sea distinction is completely lost.

Import from e-STation (TIF files)
The e-Station files have the following features:

- The filenames (at least for the samples we received from JRC) are GLIMPSE compatible: YYYYMMDD_NDV.tif and YYYYMMDD_SM.tif. That is: prefix=none, date format=1, suffix=_NDV/_SM). If the original filenames would be different, they can be adapted in advance with the Rename utility.
- The contents of the files is identical to the ones distributed by CTIV, only the format has been modified from HDF4 to TIF. Fortunately, the information from the CTIV LOG file has been included, so the TIFs do contain the full map information.

The import (conversion from TIF to ENVI-IMG/HDR) can be realised with the Generic Import tool. One must only take care that the date fields in the HDR are correctly filled: DATE=YYYYMMDD, DAYS=10 (or maybe sometimes 1 for S1-composites). In the *single file* mode this must be done manually, in *time series* mode, the dates are automatically filled (though not DAYS=10/1!). Lacking date information will always raise problems in later steps. For the rest, the HDRs are OK (map info, etc.). Of course, the procedures must be repeated separately for both image types (NDVI, SM).

Import from DevCoCast and CTIV (HDF4)
The VGT-data distributed by DevCoCast have exactly the same format as the original ones from CTIV:

- One HDF for each image layer. CTIV provides 11 layers/HDF in each composite: 4 reflectances (Blue, Red, NIR, SWIR – falsely called MIR), 4 angles (SZA, SVA, VZA, VAA), a time grid (registration date/time), NDVI and the status mask (SM). DevCoCast only distributes the most relevant NDVI and SM.
- An ASCII-formatted LOG file provides all ancillary information: period (S1/S10), start date, map info, etc. – though not the spectral scaling of the data (this is only provided in the CTIV-website!). On the contrary, the HDFs contain the image data and two “tags” with the numbers of columns and records, but for the rest they are “empty”.
- Files are called nnnn_suf.ext, with nnnn a sequential number (mostly 0001) and suf a specific suffix. For instance, a DevCoCast VGT-S10 might comprise the files 0001_NDV.hdf, 0001_SM.hdf and 0001_LOG.txt.
- To note that CTIV and DevCoCast mostly distribute the data in ZIP-form.

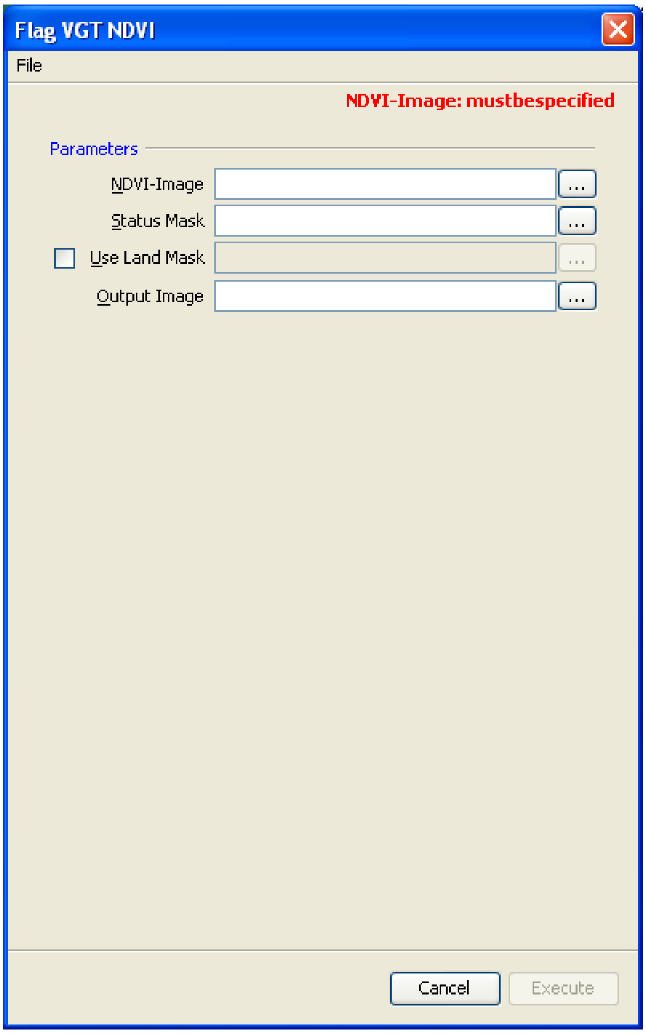
For the import (conversion to ENVI-IMG/HDR) several methods can be followed, e.g.:

- Generic Import tool: It works but as the HDFs only contain the minimum set of ancillary information, the generated HDRs are quite deceiving: no map info, sensor, date, etc. This can be solved using the and Adapt HDR option in the importer, but it involves some work.
- VGTextract: Users who receive their VGT-data from DevCoCast automatically dispose of the VGTextract software. This very program also contains a module to "export" the HDFs to the desired ENVI-format.

Parameters

- the NDVI IMG;
- the status mask (SM) IMG;
- the land mask IMG;
- the output IMG.

Tool



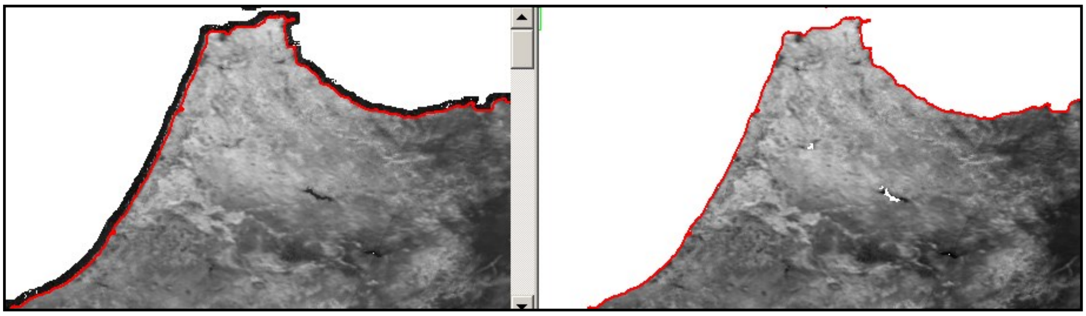
Flag VGT NDVI example

The program does the following:

- The input NDVI-image is copied but all digital values are shifted downwards with 5 units. So the range [0 - 255] becomes [-5 – 250], but to maintain the BYTE data type the negative values are reset to 0. That is no problem, because they represent NDVI-values of -0.08 to -0.10, which can only be water, snow or deep clouds. This of course changes the scaling:
  - CTIV: NDVI [-0.10 → +0.92] = -0.10 + 0.004 * V [0 → 255]
  - Now: NDVI [-0.08 → +0.92] = -0.08 + 0.004 * V [0 → 250]
- So in the output image, the upper digital range [V=251-255] is now available to store flags. As in MARSOP, these go as follows: 251=missing or error, 252=cloud or cloud shadow, 253=snow/ice, 254=water, 255=background (not covered or unknown).
- If no Land Mask (p3) is provided, the flags are solely assigned on the base of the SM:
  - Errors in RED/NIR → 251
  - Cloud/shadow 252
  - Snow/ice 253
  - SM=0 255 (water & boreal pixels in winter)

In this case, there are no pixels with flag 254. Moreover, the false land rims along the coasts remain.

- If a Land Mask (p3) is provided (recommended), it must be BYTE and all positive values are considered as land, the 0-values as water. We mostly use GLC2000 for this goal. In this case the flags are assigned as follows:
  - If Land Mask=0 254 (water, so the rest should be land)
  - If SM=0 255 (very exceptional)
  - Errors in RED/NIR 251
  - Cloud/shadow 252
  - Snow/ice 253



Flagged VGT-NDVI for Morocco: left without, right with Land Mask (GLC2000).

3.43. Reproject SHP

Goal

Project/re-project ESRI shape (SHP) files.

Besides the actual SHP (re-)projection, the input SHP can be restricted to an ROI, thereby limiting the output SHP to those features whose geometry intersects the extents defined by the ROI.

Parameters

The tool is based on the ogr2ogr utility from the OGR utilities (OGR Simple Features Library; OGR is a part of the GDAL library).

The ogr2ogr syntax used is as follows:

```
ogr2ogr [-spat xmin ymin xmax ymax [-clipsrc spat_extent]]
        [-t_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
        [-s_srs  EPSG:epsg|ogc_wkt_file|ESRI::esri_wkt_file]
        -f ESRI Shapefile
        -overwrite
         output_shp
         input_shp
```

These parameters are captured in the UI as follows:

- the input SHP coordinate system (“Spatial Reference Set” – SRS) can be specified via:
  - the metadata of the input SHP itself (if present);
  - an EPSG code (for instance EPSG:27700 is the British National Grid);
  - a file containing the SRS in OpenGis Well Known Text format (OGC-WKT);
  - a file containing the SRS in ESRI Well Know Text format (ESRI-WKT).
- the SRS for the output SHP can be specified via similar options.

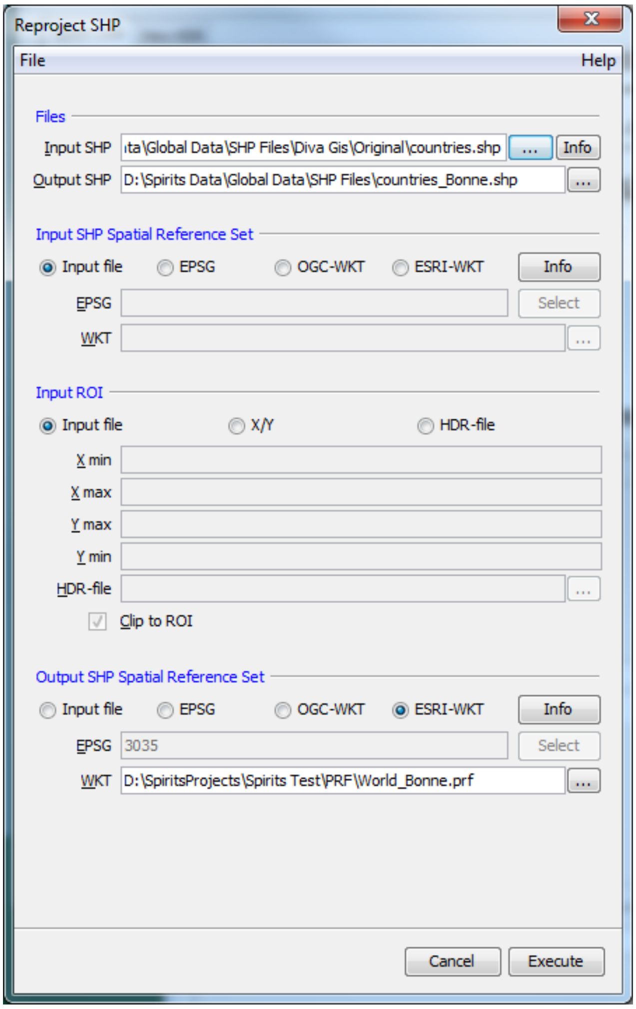
In case the “Input file” option is selected for the output SRS, it means there will be no actual reprojection as such, since the input and output SHP will have the same SRS. This configuration can be used for example to extract an ROI from a SHP file.

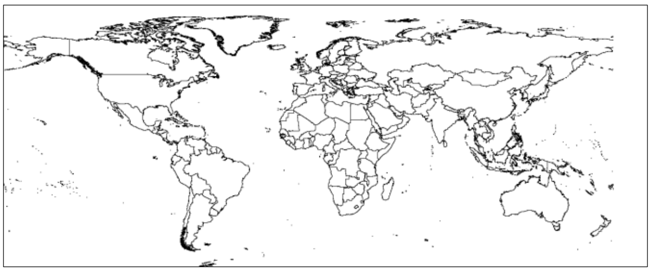
- the ROI can be specified via:
  - the input file, meaning ogr2ogr will use the SHP as-is (the ogr2ogr “-spat” parameter is not used).
  - X/Y coordinates. In this case, the extent is to be specified via the X min, X max, Y max and Y min coordinates. These coordinates need to be specified according to the selected input SRS;
  - an existing HDR. The extent (X min, X max, Y max, Y min) values are then calculated from the values found in the “map info” entry of this HDR, and passed to ogr2ogr as in previous case.

In case the ROI is actually specified (via X/Y or HDR) the output SHP will be limited to those features whose geometry intersects the extents defined by the ROI. These features can optionally be clipped to the extent via the “Clip to ROI” option.

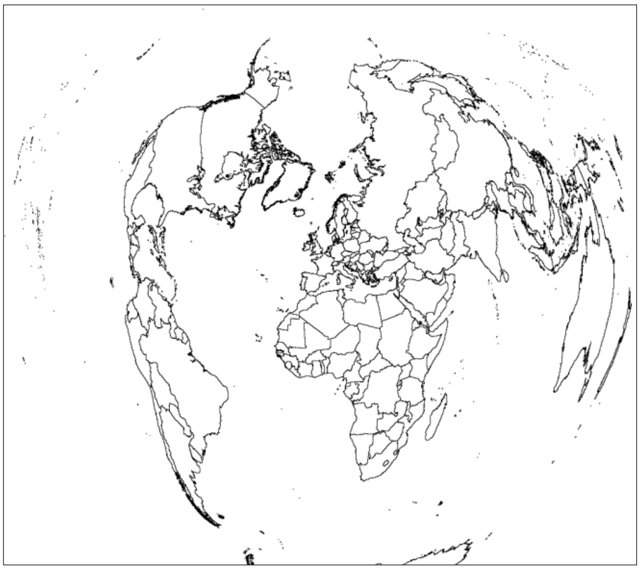
Remark: in some cases (especially with projection systems which are not globally valid) omitting an explicit ROI can yield unpredictable results.

Tool



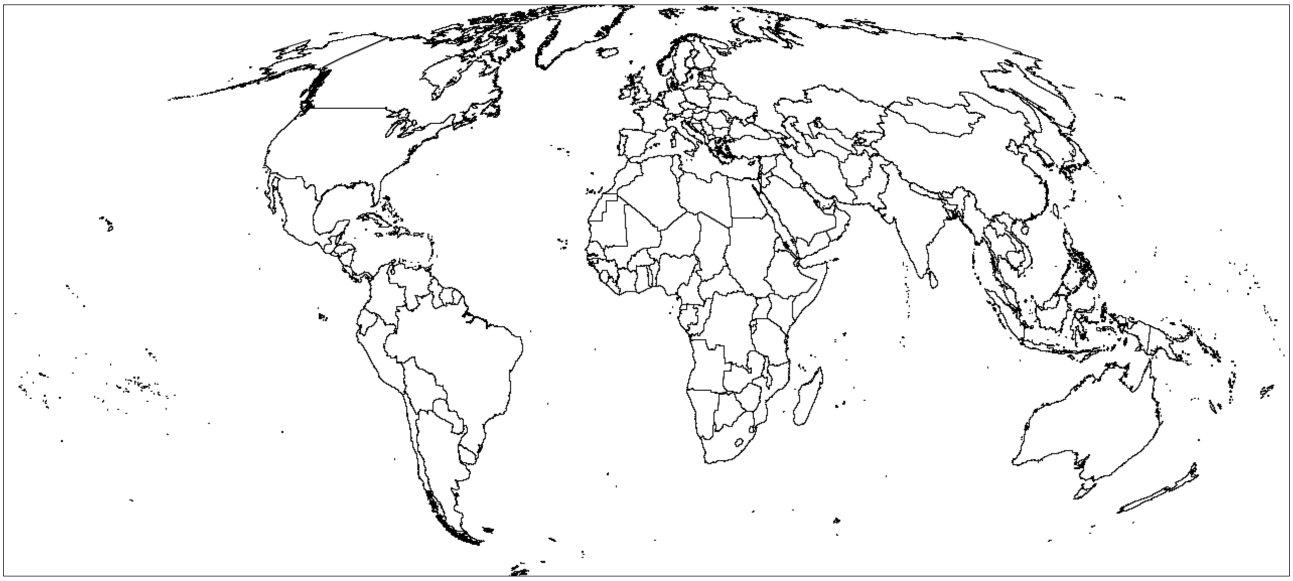


input SHP (EPSG 4326 - “WGS 84”)



Reproject SHP Tool example

output SHP (Bonne)



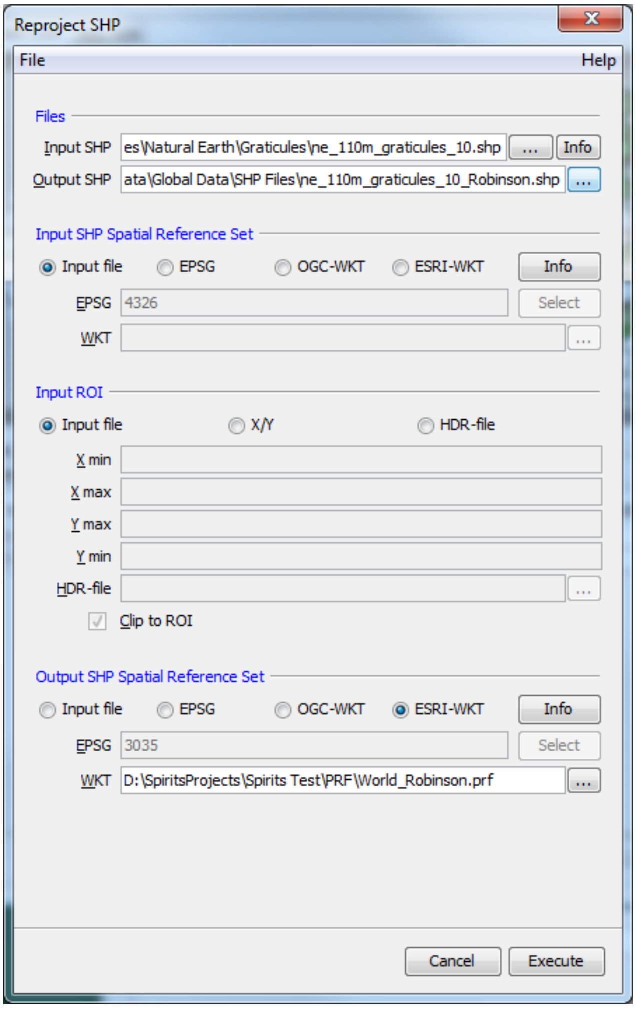
output SHP (Mollweide)

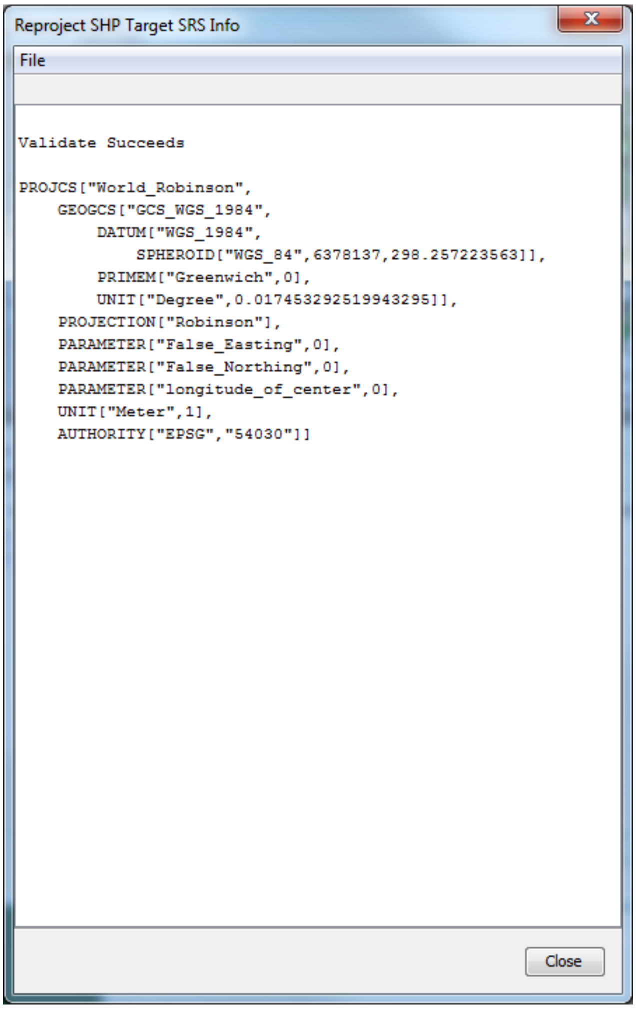
Info buttons:

Via SRS “Info” buttons, the selected input and output SRS can be validated and examined. Subpanels will show the validation result and detailed information in Well Known Text format (OGC type).

Once these panels are activated, they can remain on the screen while selecting other SRS sources.

The validation validation and the information retrieval for an SRS is based on the gdalsrsinfo utility from the GDAL utilities as described for the Reproject IMG Tool.





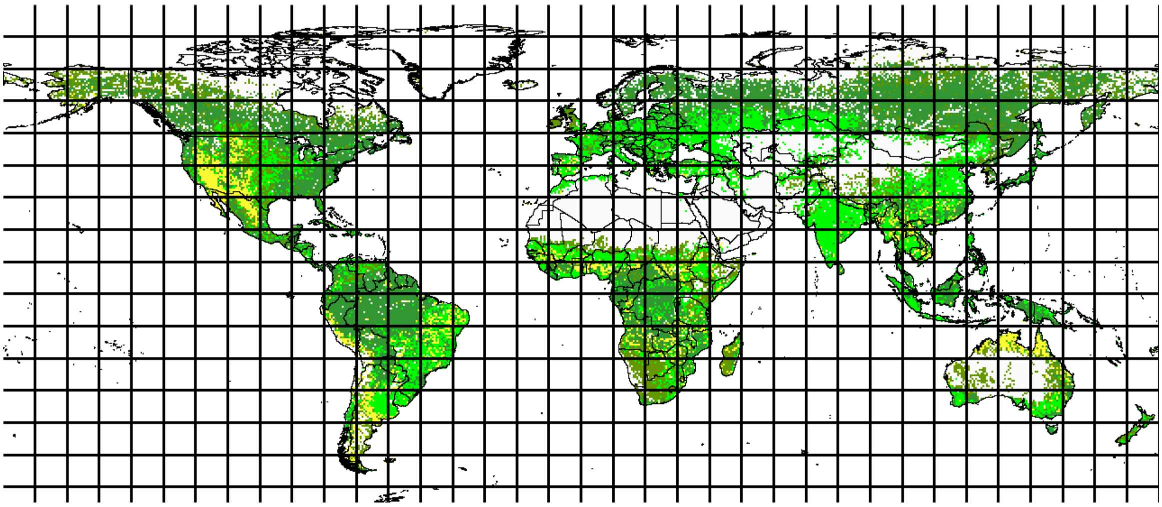
Reproject SHP Tool example

Output SHP SRS Info panel

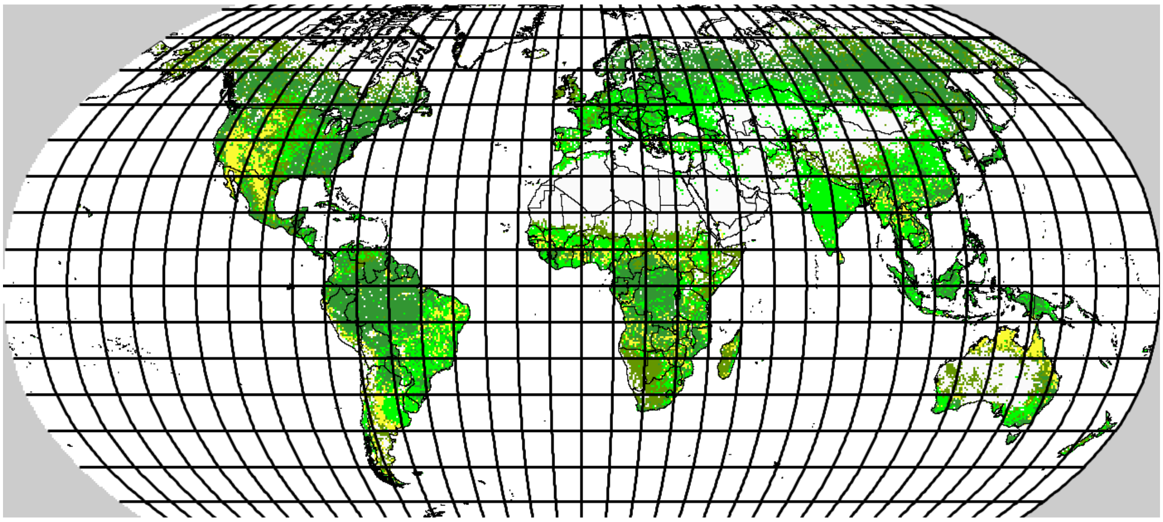
Via the Input SHP “Info” button, the information from the selected SHP file can be examined. This information originates from the gdalsrsinfo utility from the GDAL utilities. The gdalsrsinfo syntax used is as follows:

```
gdalinfo -so input_shp input_shp_filename
```

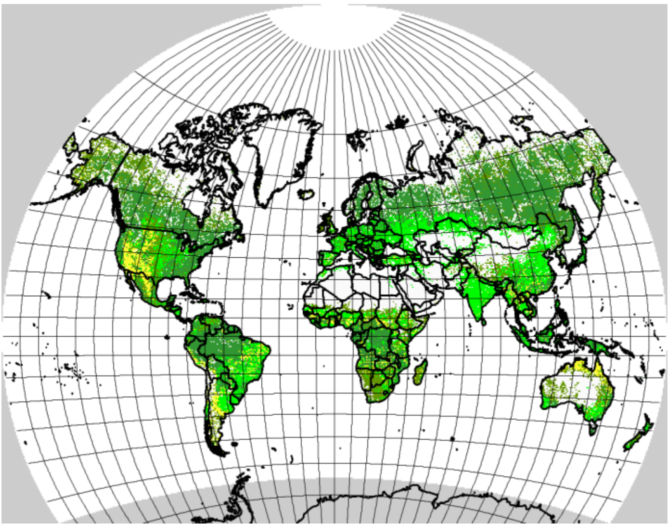
example: some IMG & SHP Projections

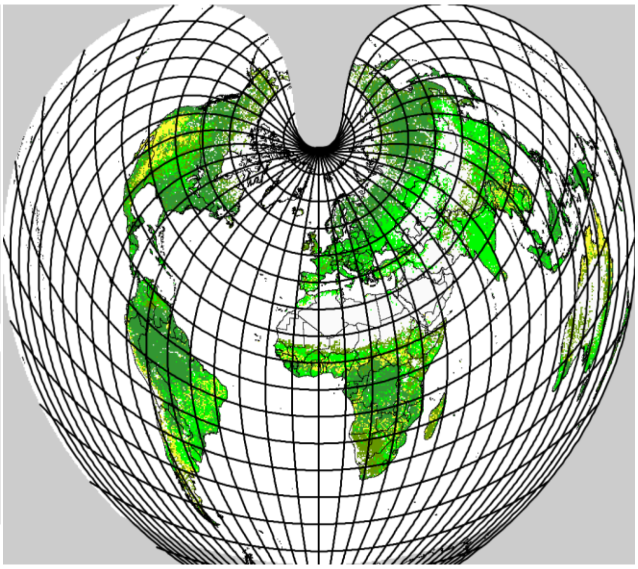


WGS 84



Robinson

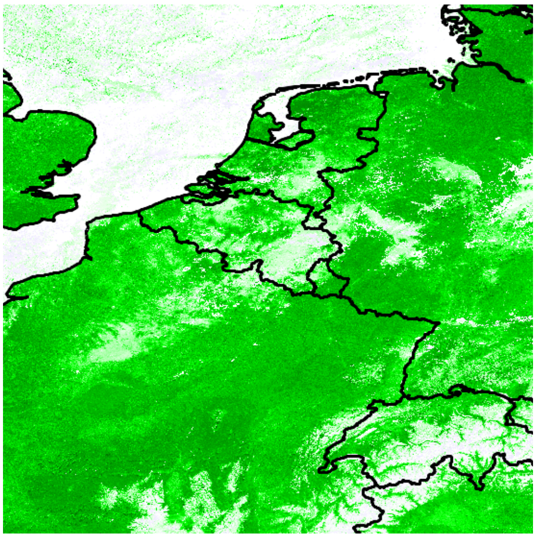




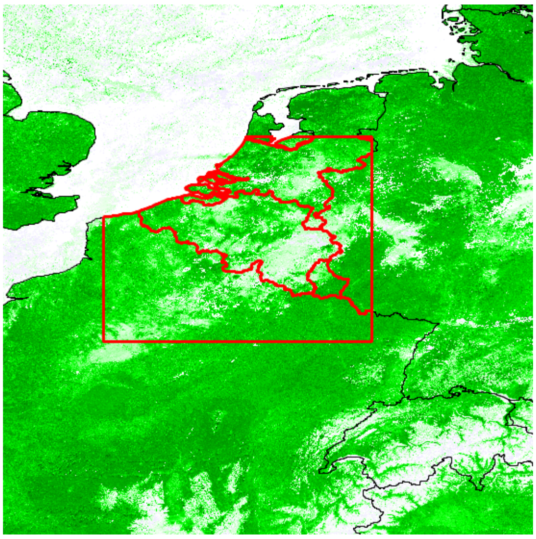
Van der Grinten

Bonne

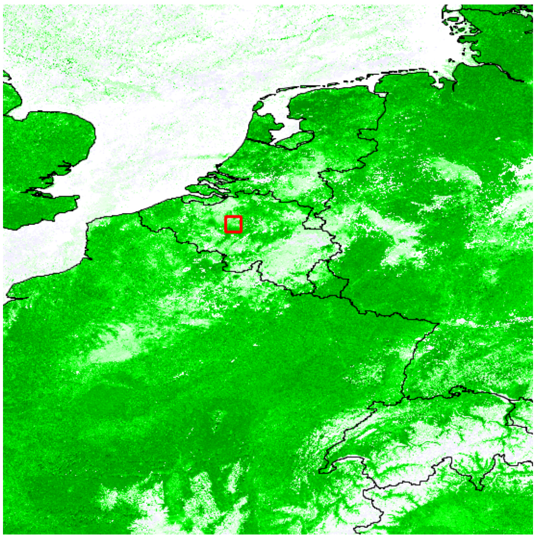
example: clipping



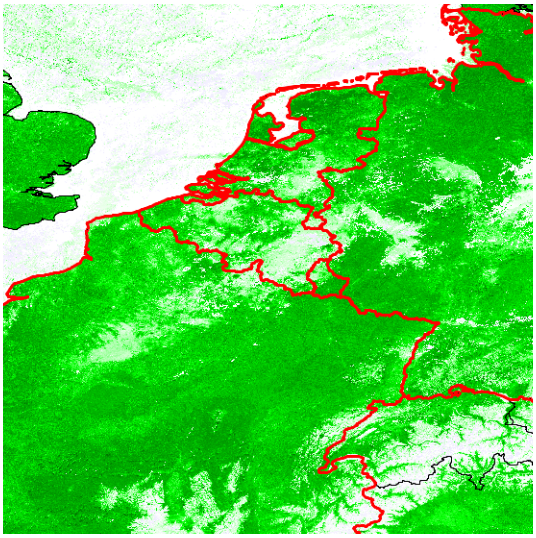
input SHP: global country boundaries



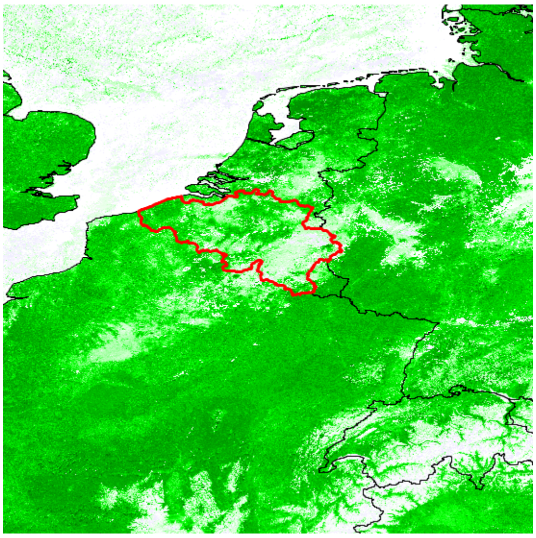
output SHP: ROI around Belgium–clipped



output SHP: ROI inside Belgium–clipped



output SHP: ROI around Belgium–not clipped



output SHP: ROI inside Belgium–not clipped

3.44. Rasterize SHP

Goal

Rasterize an ESRI shapefile (SHP) into an ENVI raster image file (IMG).

Parameters

The tool is based on the gdal_rasterize utility from the GDAL Utilities (Geospatial Data Abstraction Library from the Open Source Geospatial Foundation).

The gdal_rasterize syntaxes used are as follows:

```
gdal_rasterize -a attribute_name -l layername -of ENVI
               -a_nodata nodata_value -init nodata_value
               -co INTERLEAVE=BSQ
               -te xmin ymin xmax ymax
               -ts width height
               -ot {Byte, Int16, Int32, Float32}
               src_datasource dst_filename

gdal_rasterize -a attribute_name -l layername -of ENVI
               -a_nodata nodata_value -init nodata_value
               -co INTERLEAVE=BSQ
               -te xmin ymin xmax ymax
               -tr xres yres
               -ot {Byte, Int16, Int32, Float32}
               src_datasource dst_filename
```

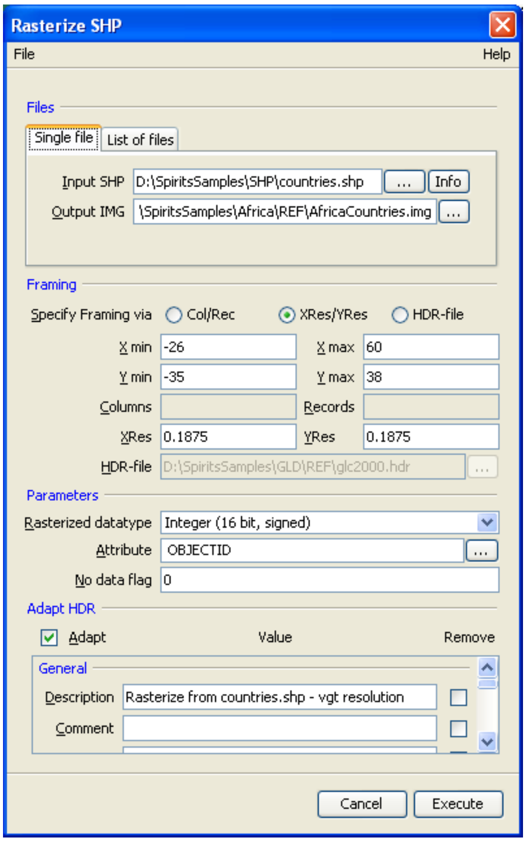
These parameters are captured in the the UI as follows:

- the output IMG framing. Three options are available:
  - in terms of the map info from an existing HDR file;
  - in terms of corner coordinates (X / Y) and resolution (Xres/Yres);
  - in terms of corner coordinates (X / Y) and IMG size (columns/records).
- the rasterized data type. Byte(8 bit, unsigned), Integer (16 bit, signed), Long (32 bit, signed) or Float (32 bit);
- the name of attribute of which the values will used as raster values in the IMG. The selected attribute should have numerical values, in the range of the selected data type. The attributes available in the SHP's DBF file can be viewed and selected via the chooser button;
- the flag value to be used for missing data (also used as initial value for all pixels in the resulting IMG);

Remark:

To give the user an indication if gdal_rasterize recognises the input file, there is the "Info" button on the panel (only in case of "Single File"). This button opens a window which shows the result of another utility: ogrinfo. In case gdal recognises the current input file, ogrinfo, will list various information in this window. The content of this info-window can be saved as a text file via its File menu.

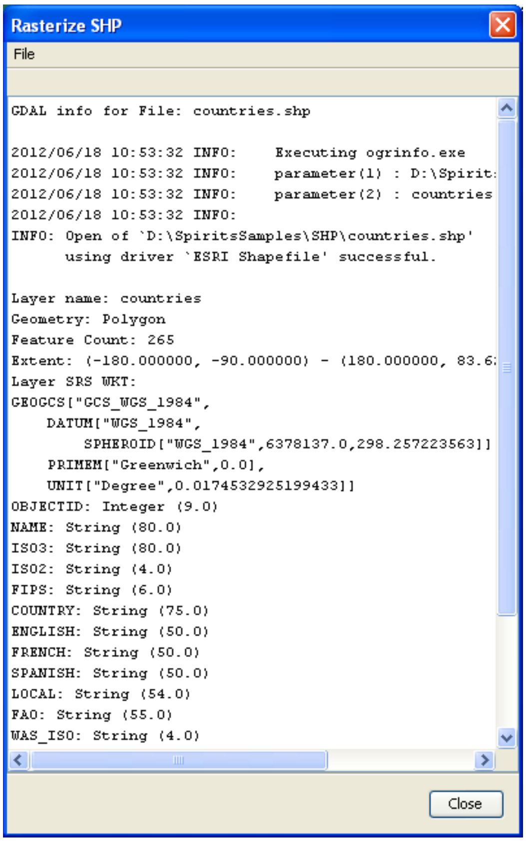
Tool



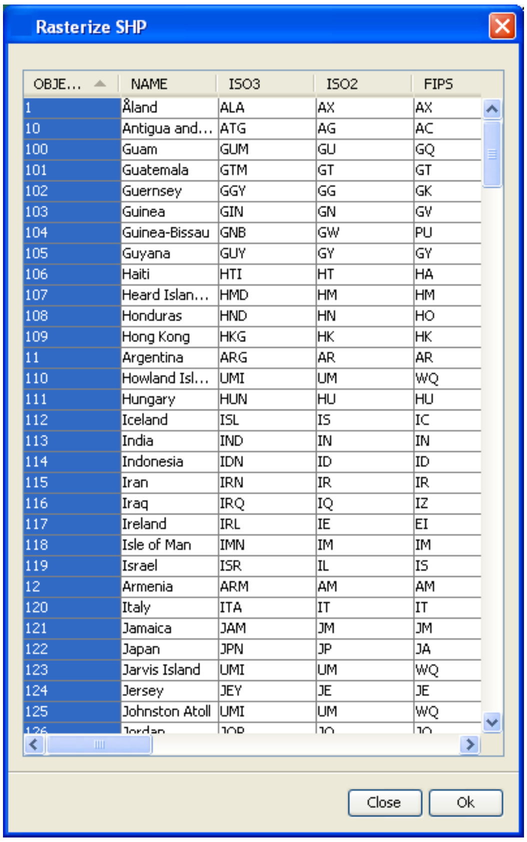
Remark:

- the SHP and DBF files are mandatory;
- in case no PRJ file is present the map info in the resulting IMGs HDR will be Unknown;

Rasterize SHP Tool example



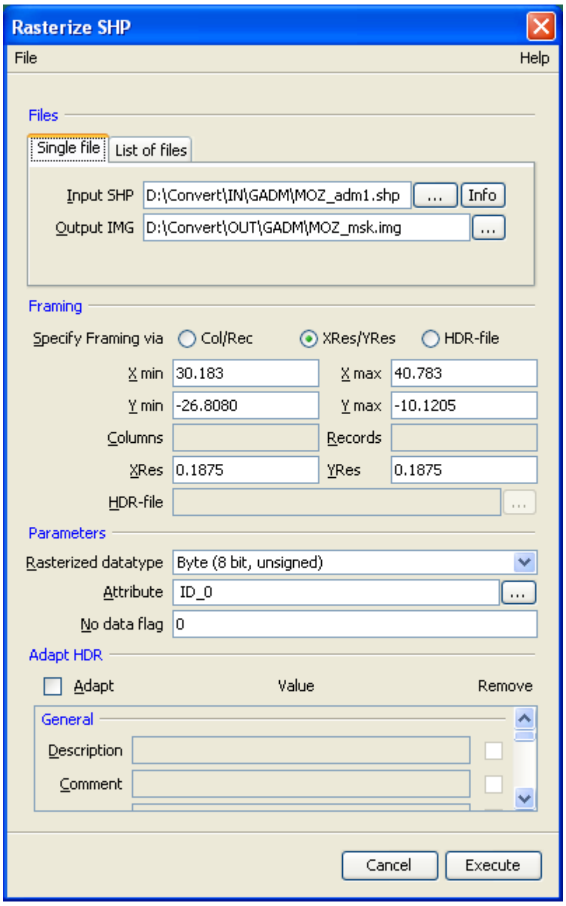
Rasterize SHP - info panel example

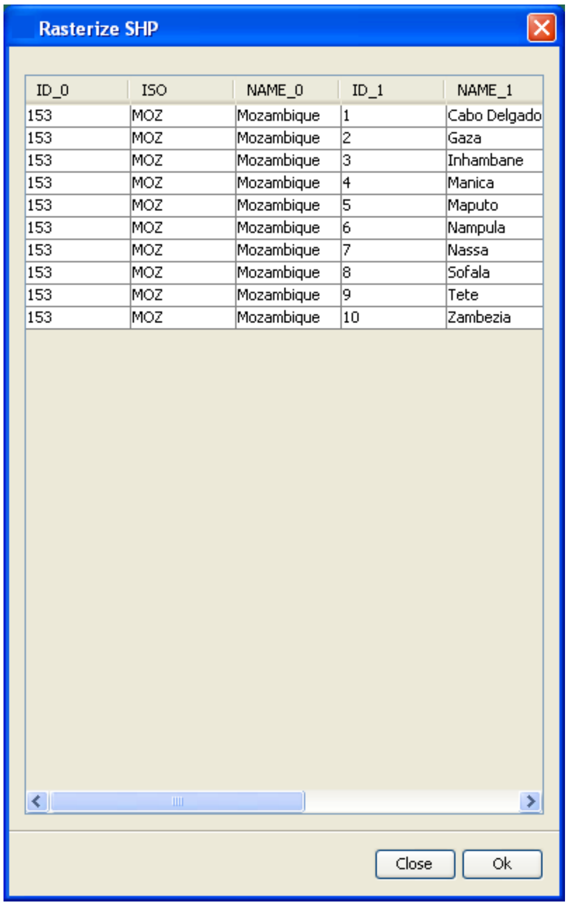


Rasterize SHP - attribute selection panel example

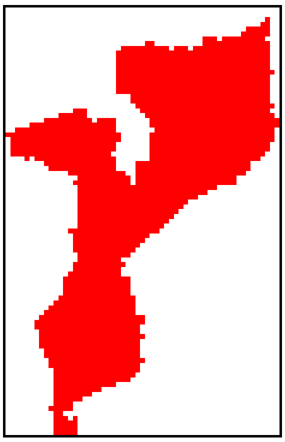
Rasterize mask and regions IMGs example

In this example an SHP for the Mozambique regions is rasterized. Examination of the DBF file shows the file has multiple attributes. Attribute ID_0 is a constant value for all regions, attribute ID-1 gives a region-id from 1 - 10. Rasterizing this file to attribute ID_0 gives an IMG which can be used as a mask. Rasterizing this file to attribute ID_1 gives an IMG which can be used as a regions-IMG for RUM extraction.

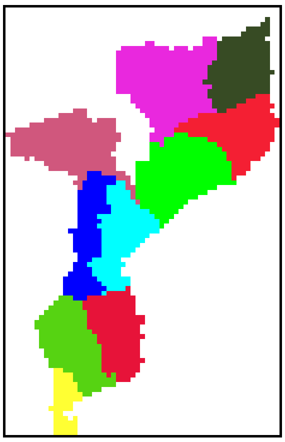




ID_0	ISO	NAME_0	ID_1	NAME_1
153	MOZ	Mozambique	1	Cabo Delgado
153	MOZ	Mozambique	2	Gaza
153	MOZ	Mozambique	3	Inhambane
153	MOZ	Mozambique	4	Manica
153	MOZ	Mozambique	5	Maputo
153	MOZ	Mozambique	6	Nampula
153	MOZ	Mozambique	7	Nassa
153	MOZ	Mozambique	8	Sofala
153	MOZ	Mozambique	9	Tete
153	MOZ	Mozambique	10	Zambezia



rasterized with attribute "ID_O" (constant 153 for each region)



rasterized with attribute "ID_1" (values 1-10 varies over the regions)

3.45. Area Fraction IMGs

Goal

Create low resolution Area Fraction IMGs (AFIs) from a high resolution classification IMG (hard classification). Typically AFIs will be used in the Extract RUM tool (starting with low resolution IMGs and a high resolution land cover classification IMG).

Parameters

The tool can operate in two modes. In the "Thinning" mode, the resolution of the original (high resolution) classification IMG is degraded by an integer factor, its framing is kept as is. In the "Resampling" mode, the original classification IMG is resampled to obtain the required resolution and framing.

Parameters for both cases:

- the high resolution hard classification IMG;

This IMG must be a classification (HDR entry file type = ENVI Classification) of a byte data type (HDR entry data type = 1), and its HDR file must contain the classes entry (e.g. classes = 230).

- the output directory where the resulting AFIs should be stored;
- the filename prefix for the AFIs. The created AFIs filename structure will be: ***prefixCCC.img***, with CCC the Class-ID's (example: Globcover_011.img). An AFI will be created for each class;
- the AFIs data type: byte or integer. The AFIs physical values are percentages (0.0-100.0). In case of byte data type they will be scaled (digital values) 0 - 200, in case of integer data type 0 - 1000;
- the operation modes:
    - Thinning: in case the framing of the required AFIs have the same framing as the input classification IMG, and their resolution is an n-fold of that of input classification IMG;
    - Resampling: in case a resampling the input classification IMG is needed obtain the required framing and resolution.

Parameters for Thinning:

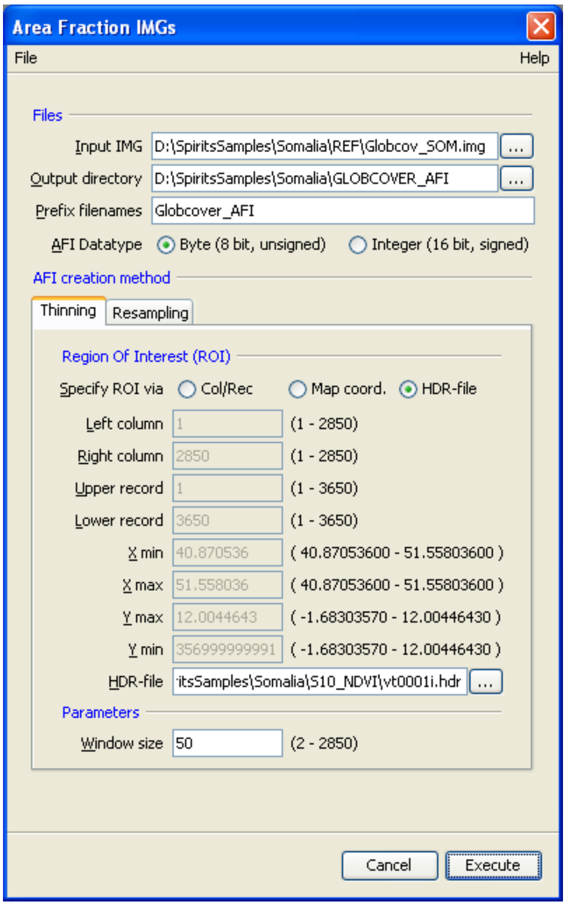
- the ROI to be extracted;
- the window size (side of the window square of input pixels to be combined into one output pixel);

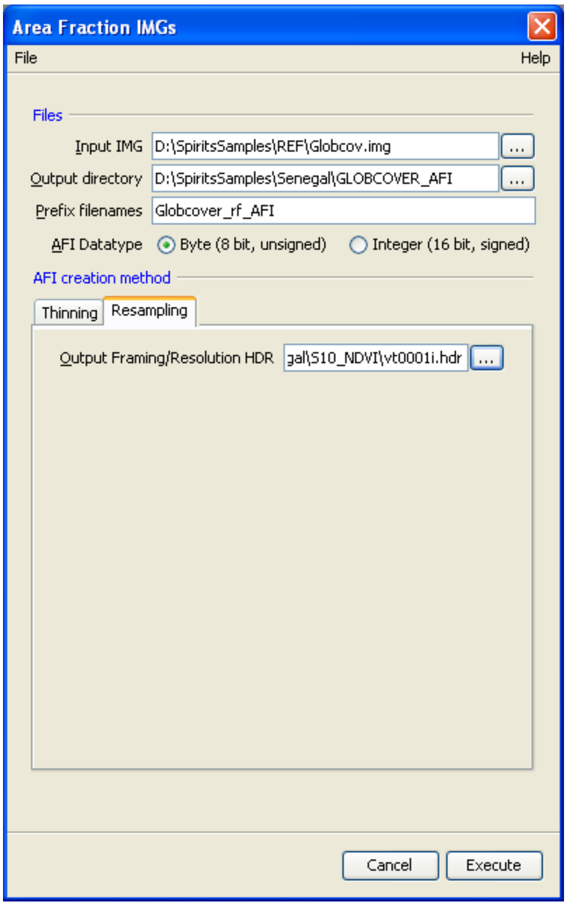
The ROI can be specified in terms of IMG coordinates (Columns / Records), Map coordinates (X / Y) or Map coordinates from an existing HDR file.

Parameters for Reampling:

- an existing HDR file specifying the ROI to be extracted in its required framing and resolution.

Tool





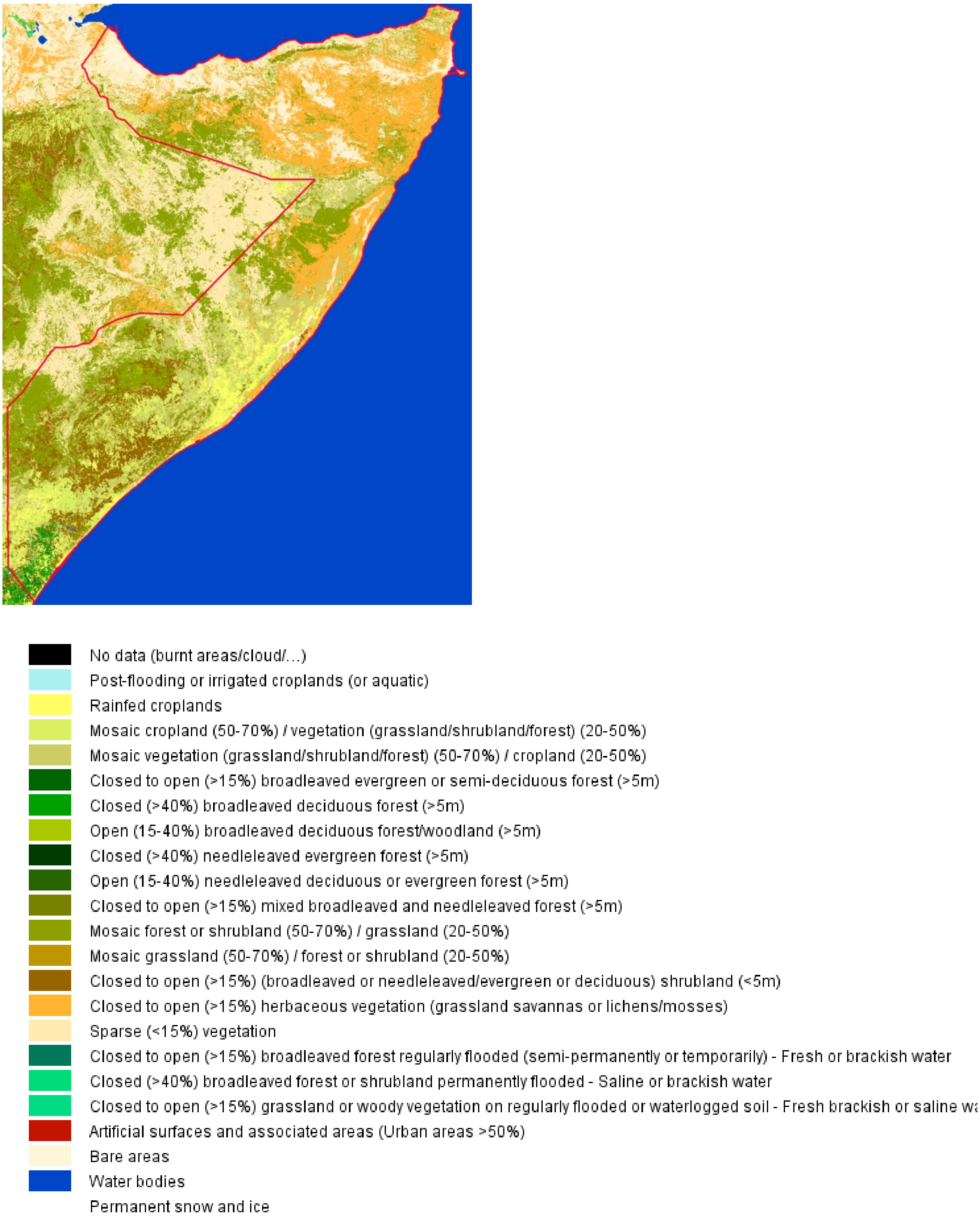
Area Fraction IMGs Tool -thinning example Area Fraction IMGs Tool - resampling example

Remarks:

- only 'non empty' AFIs are created (at least 1 pixel with AF>0);
- a 'garbage-AFI' containing the AF's for all non-classified pixels (filename: prefix000.img) and additional VAR and MTA files (file names: prefix.VAR and prefix.MTA) are always created.
- the sum of the pixel values of all created AFIs will be 100%.;

AFIs example

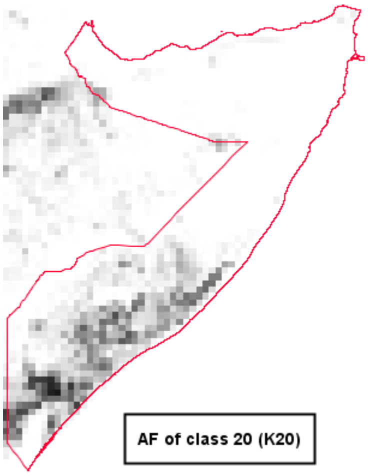
example: area fraction IMGs from Globcover classification for Somalia ROI.



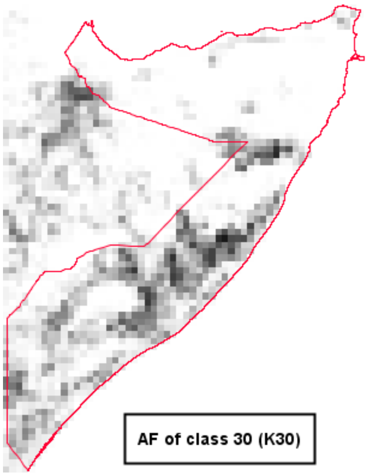
input high resolution hard classification IMG

```
...
   1   14 d:\spiritssamples\somalia\globcover_afi\globcover_afi014
   2   20 d:\spiritssamples\somalia\globcover_afi\globcover_afi020
   3   30 d:\spiritssamples\somalia\globcover_afi\globcover_afi030
   4   40 d:\spiritssamples\somalia\globcover_afi\globcover_afi040
   5   50 d:\spiritssamples\somalia\globcover_afi\globcover_afi050
   6   60 d:\spiritssamples\somalia\globcover_afi\globcover_afi060
...
  15  160 d:\spiritssamples\somalia\globcover_afi\globcover_afi160
  16  180 d:\spiritssamples\somalia\globcover_afi\globcover_afi180
  17  190 d:\spiritssamples\somalia\globcover_afi\globcover_afi190
  18  200 d:\spiritssamples\somalia\globcover_afi\globcover_afi200
  19  210 d:\spiritssamples\somalia\globcover_afi\globcover_afi210
  20    0 d:\spiritssamples\somalia\globcover_afi\globcover_afi000
```

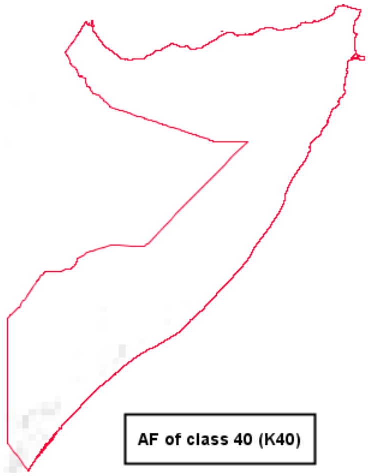
extract from Globcover_AFI.var



class ID 20
(Mosaic cropland / vegetation)

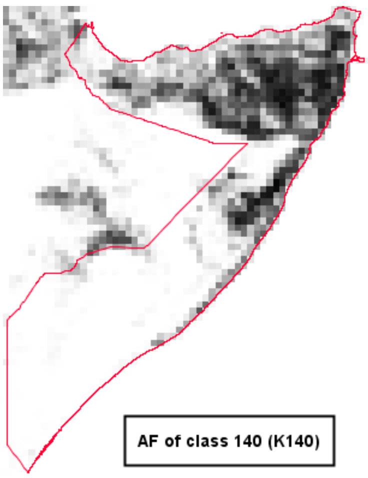


class ID 30
(Mosaic vegetation/ cropland)



class ID 40...

...



class ID 140....

...

class 210: Water bodies

created AFI IMGs

3.46. Extract RUM

3.46.1. General

Goal

Extract regional and unmixed mean values from IMGs by overlay with regions (administrative or other) and Land Use data maps.

Parameters

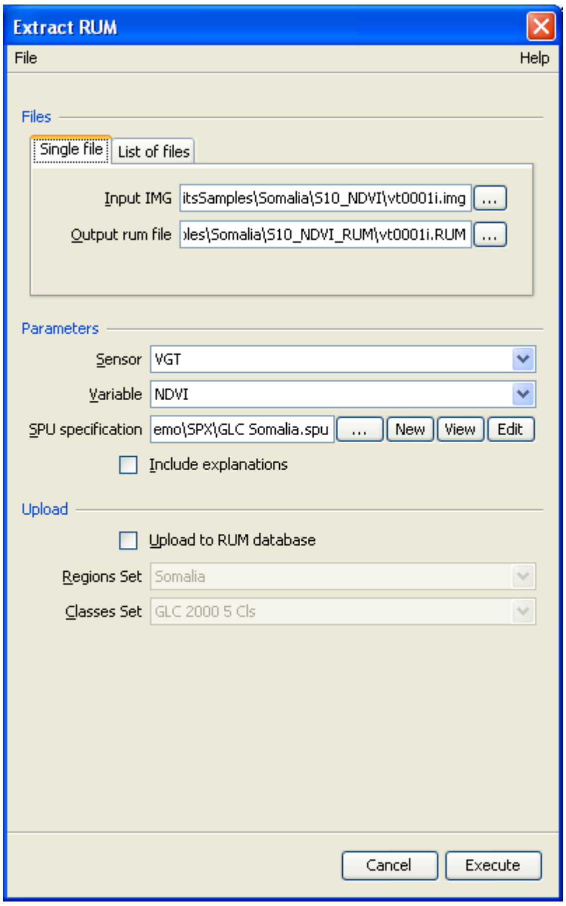
- the Sensor from which the values will be extracted;
- the Variable from which the values will be extracted;
- an SPU file, Unmixing specification file, containing the detail specifications for the extraction;
- a selection whether or not the extracted data table is to be preceded with its legend;

Optionally the extracted data can be uploaded directly into the projects RUM database, used to visualize them via the RUM Chart utility. If so, additional parameters are:

- the Regions Set containing the Regions used in the extraction;
- the Classes Set containing the (land use/land cover) Classes used in the extraction;

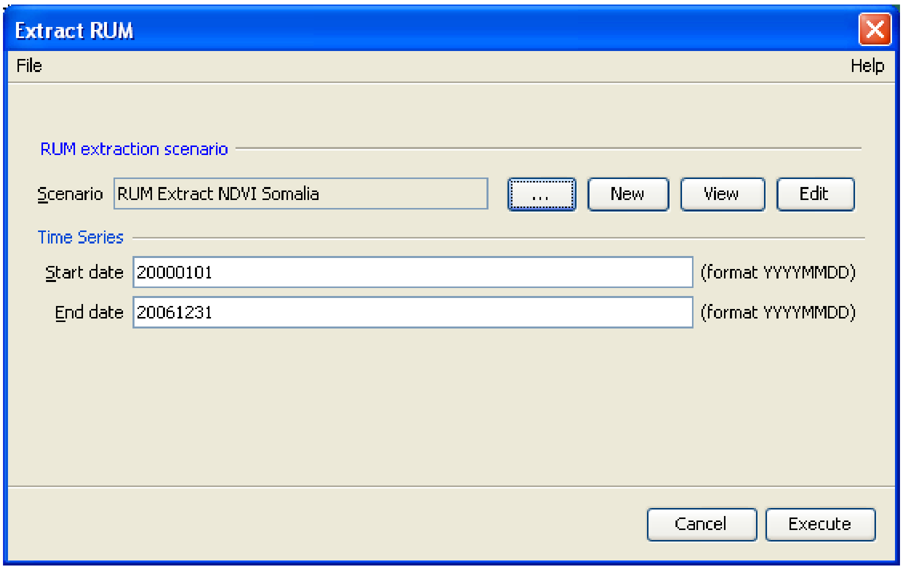
Remark: the Sensor, Variable, Regions Set and Classes Set need to be defined in the projects RUM database in advance.

Tool



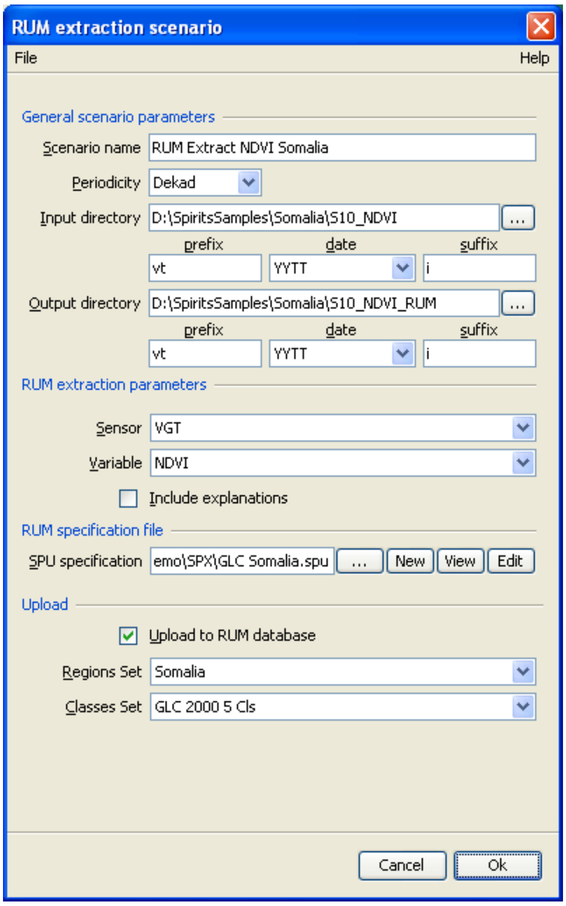
Extract RUM example

Time Series



Extract RUM Time Series example

Scenario



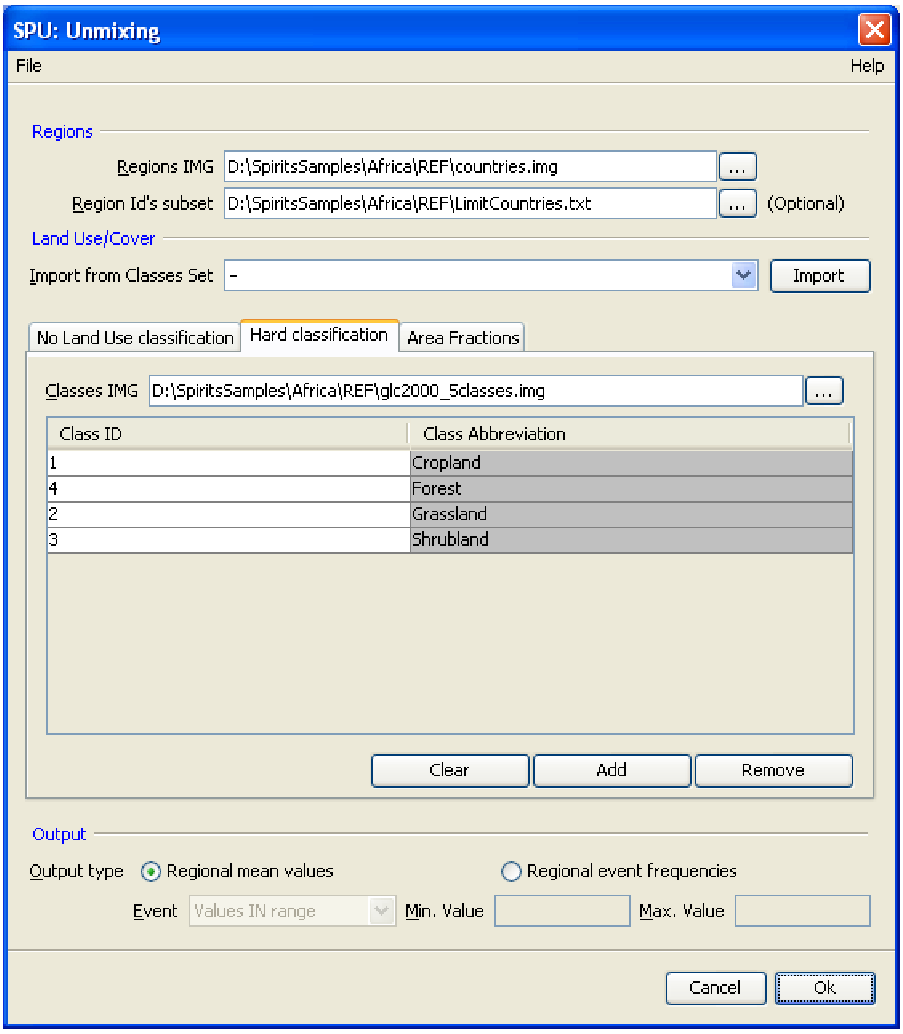
Extract RUM Scenario example

3.46.2. SPU File: Unmixing specification

Description

An SPU file, Unmixing specification file, contains detail specifications for the Extract RUM tool and time series. These files can be created and edited with the SPC-editor.

Editor



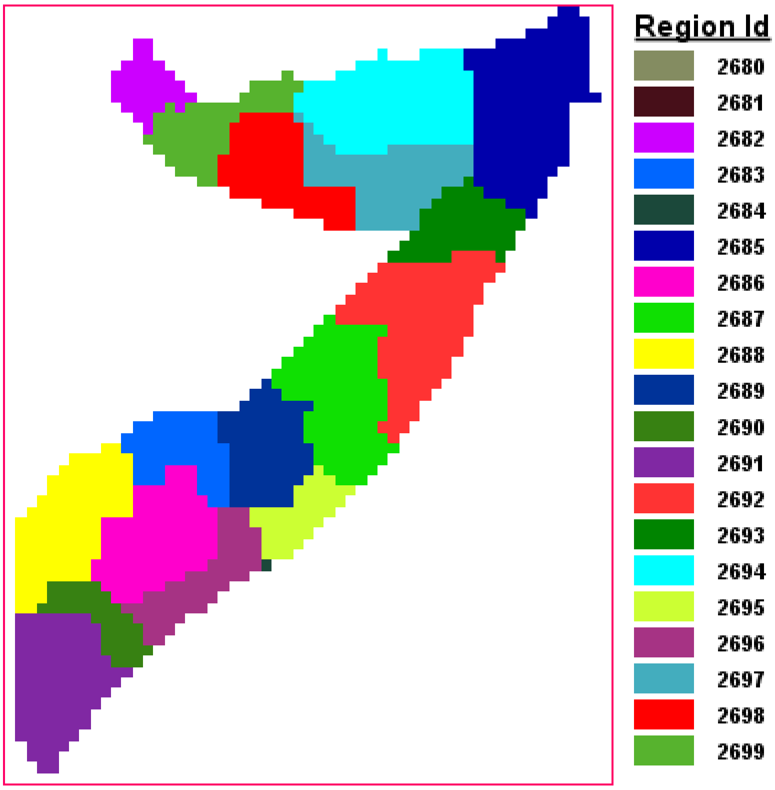
General parameters

- Output type:
  - regional mean values;
  - regional frequencies.

In case regional frequencies are to be extracted, the event of which the frequencies will be calculated needs to be specified by via:

- the event range (minimum and maximum physical value)
- the event type (values occurring within the specified range or beyond the specified range);

- Regions IMG: (mandatory) IMG with Region-ID per pixel.



Regions IMG example

The Regions IMG must be of an integer data type.

The Regions IMG and the IMGs the values will be extracted from, must have identical geo-referencing.

The Regions IMG HDR file must contain the values entry. The region-ID's must belong to the [Vlo,Vhi] interval. All region-ID's must be greater than 0 (thus also Vlo).

```
values = {ID, -, 2682, 2699, 2682, 2699, 0, 1}
```

Example: values entry in a Regions IMG HDR file: values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo }

In case the extracted data will be uploaded into the projects RUM database, the ID's in this Regions IMG must agree with the Id's of the Regions contained in the selected Regions Set from the database: this Id is one of the links between the data extracted by the Extract RUM tool and the data imported into the RUM database.

- Region Id's subset: (optional): an ASCII text file specifying region-ID's to include in the extraction. The purpose of specifying such file, is to **limit** the extraction to a subset of the actual region-ID's in the Regions IMG. In case this file is not specified, the extraction will include all region-ID's (in the [Vlo,Vhi] range) found in the Regions IMG.

Typically this would be a 'comma-separated values' (CSV) file. The first value on each line should be a region-ID to include. Lines not starting with a numerical value will be ignored.

```
'2682, Awdal, Awdal
'2683, Bakool, Bakool
'2684, Banaadir, Banaadir
'2685, Bari, Bari
2686, Bay, Bay
2687, Galguduud, Galguduud
'2688, Gedo, Gedo
'2689, Hiiraan, Hiiraan
'2690, Jubbada Dhexe, Jubbada Dhexe
2691, Jubbada Hoose, Jubbada Hoose
'2692, Mudug, Mudug
'2693, Nugaal, Nugaal
'2694, Sanaag, Sanaag
'2695, Shab.Dhexe, Shabeellaha Dhexe
2696, Shab.Hoose, Shabeellaha Hoose
'2697, Sool, Sool
'2698, Togdheer, Togdheer
2699, Woqooyi Galbeed, Woqooyi Galbeed
```

Example: Region Id's subset file, limiting the region-ID's included in the extraction by inserting a '.

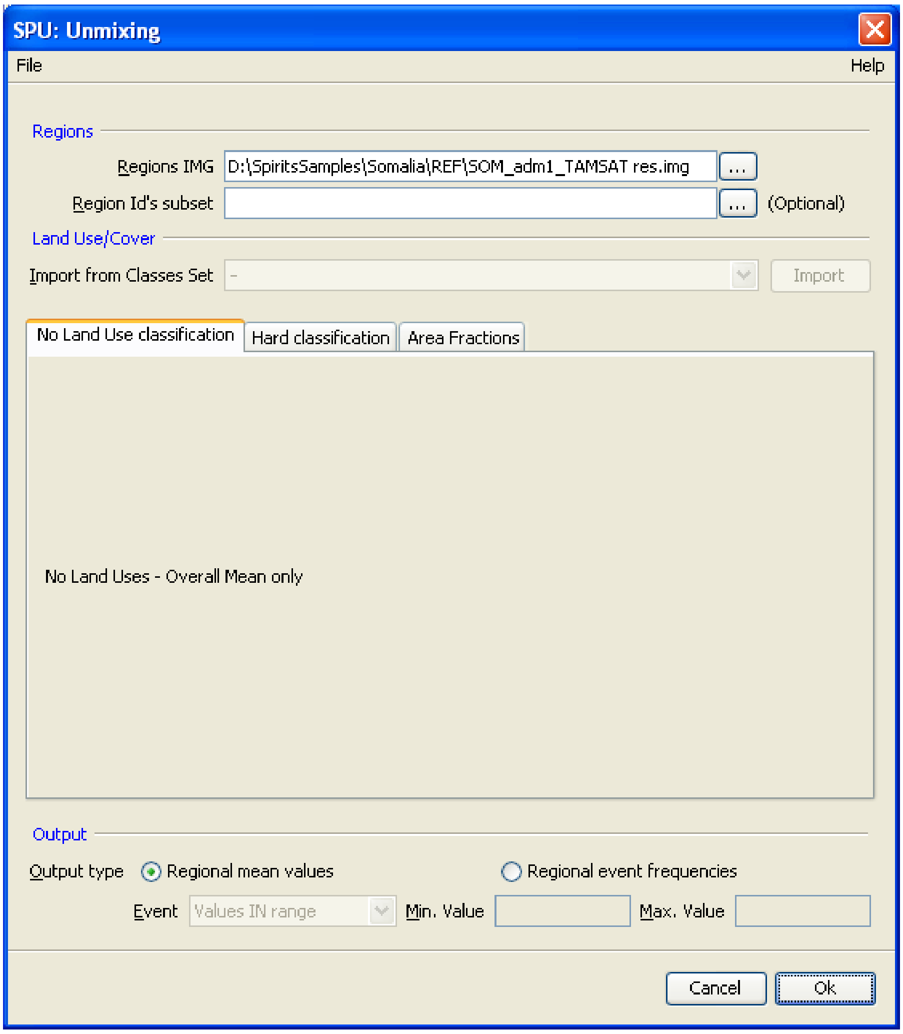
Remark: In case the extracted data will be uploaded into the projects RUM database, the Regions Set and its Regions need to be defined in the projects RUM database in advance. In that case an ASCII text file containing all Regions in the Regions Set can be obtained via the Copy/Save As function of the Regions Panel. This file can then be modified by any ASCII editor, and used as a Region Id's subset file. Removing a region-ID could be done by deleting its line, or by inserting a non numerical (non-whitespace) character on the first position of its line.

- the method to be used, depending on the land use/land cover information three methods are available:
  - regional means: method without land use/land cover information;
  - hard classification, using a Classes IMG. Each pixel belongs for 100% to a land use/land cover class;
  - classification using Area Fraction IMGs (AFIs, resulting from soft classification)..

Parameters for Regional means method

In case no land use/land cover information is used, only the regional means ("Overall Mean") are calculated: one mean value per region, without unmixing.

In this case no further parameters are required.



SPU for regional means example

Remark: The other methods (hard classification or area fractions) will also calculate the overall means automatically besides the unmixed means.

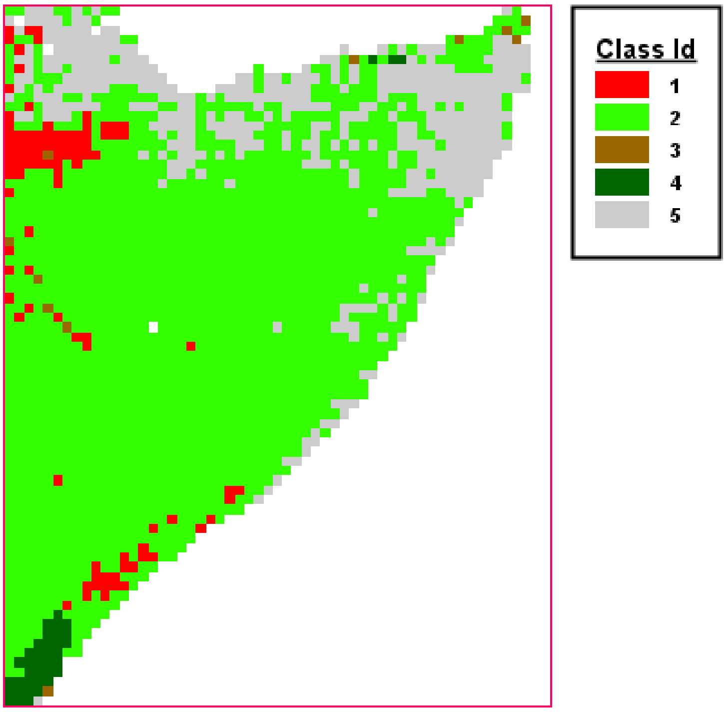
Parameters for Hard classification method

Besides the overall means (one mean value per region, without unmixing), the unweighted means are calculated (one mean value per class per region).

Additional parameters are required: the (mandatory) Classes IMG and the (optional) Class ID's list.

• Classes IMG

A Classes IMG is required, containing the Class-ID per pixel. Each pixel belongs for 100% to a single land use/land cover class.



Classes IMG example

The Classes IMG must be of an byte data type.

The Classes IMG, the Regions IMG and the IMGs the values will be extracted from, must have identical geo-referencing.

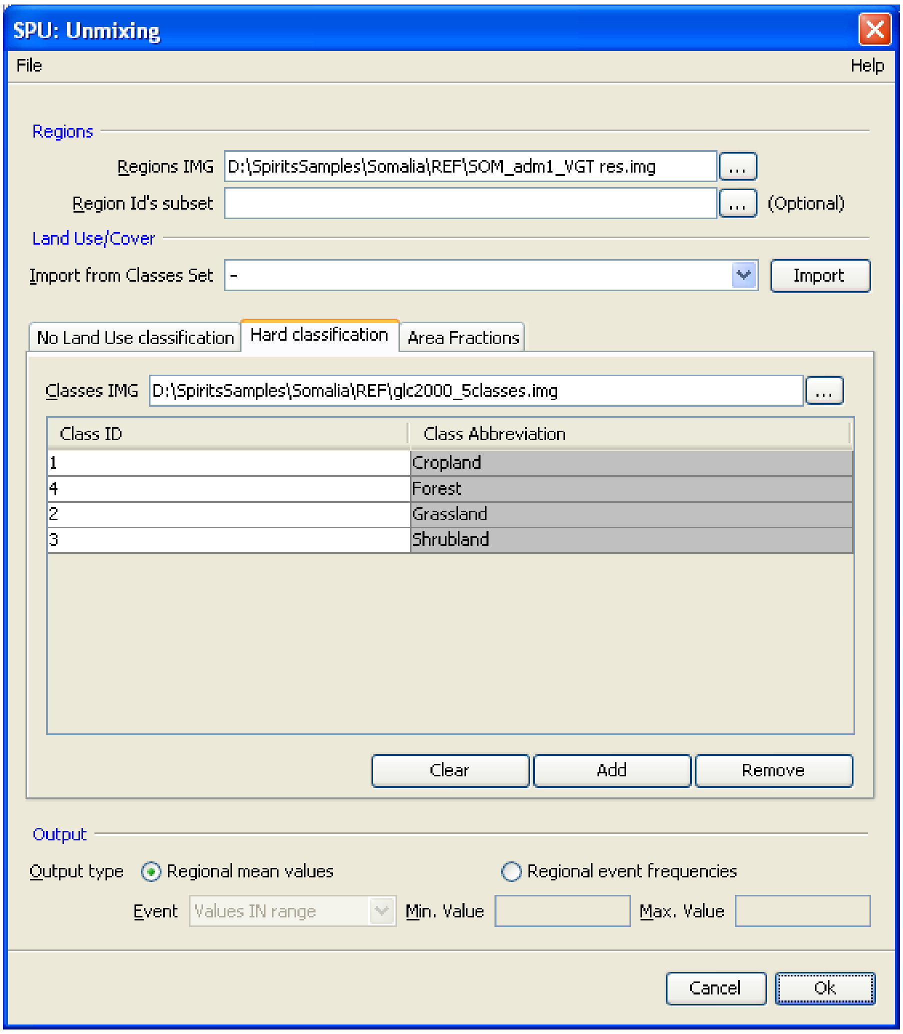
The Classes IMG HDR file must contain the values entry. The class-ID's must belong to the [Vlo,Vhi] interval in range 1-255.

```
values = {GLC2000-classes, -, 1, 255, 1, 5, 0, 1}
```

Example: values entry in a Classes IMG HDR file: values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo }

In case the extracted data will be uploaded into the projects RUM database, the ID's in this Classes IMG must agree with the Id's of the Classes contained in the selected Classes Set from the database: this Id is one of the links between the data extracted by the Extract RUM tool and the data imported into the RUM database.

- Class ID's list : (optional): a list of class-ID's to include in the extraction can be specified. The purpose of specifying such list, is to **limit** the extraction to a subset of the actual class-ID's in the Classes IMG. In case this list is left empty, the extraction will include all class-ID's (in the [Vlo,Vhi] range) found in the Classes IMG.



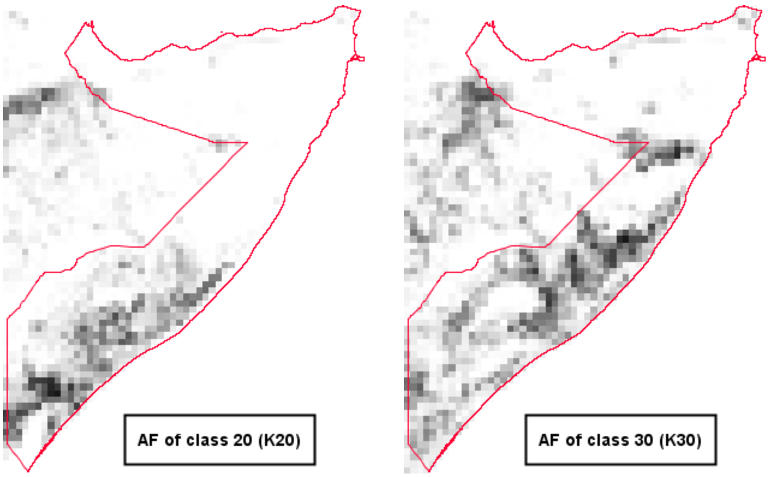
SPU for hard classification example

Remark: In case the extracted data will be uploaded into the projects RUM database, the Classes Set and its Classes need to be defined in the projects RUM database in advance. In that case the Class ID's list can be imported from the selected Classes Set via the Import button. Subsequently the class-ID's not to be included in the extraction can be removed from the list.

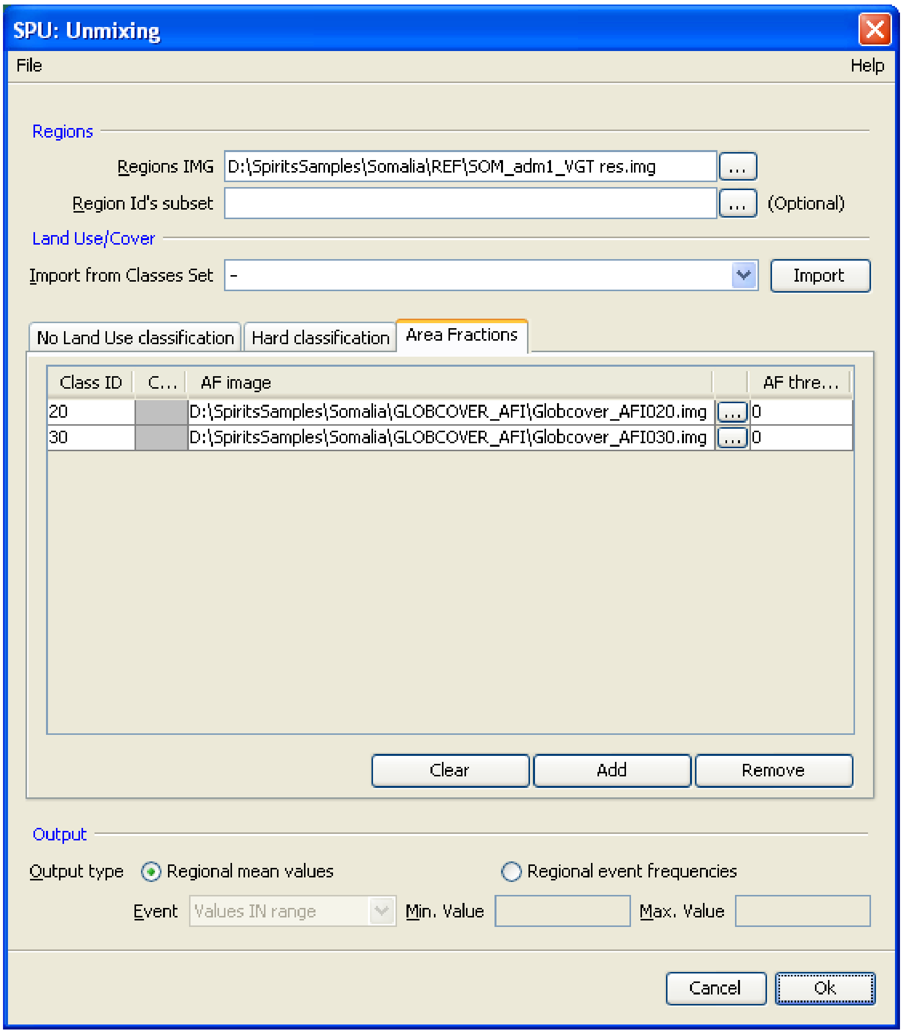
Parameters for Area fractions method

Besides the overall means (one mean value per region, without unmixing), the weighted means are calculated (one mean value per class per region, weighted by the AFI of that class.).

Additional parameters are a Class ID's list, with for each ID (land use/land cover) an AFI, which contains for each pixel the fraction covered by that class, and a threshold value.



AFI IMGs examples



SPU for area fractions example

3.46.3. RUM File: Regional Unmixed Means

Description

The outputs of the Extract RUM tool and time series are so called RUM files (*.RUM). These are ASCII 'comma-separated values' (CSV) files, containing the extraction results.

```
2690,  0,0,   0,  2,  1, 10, 20000101,100.000,100.000,     0.548,     0.097
2690,  1,1,100,  2,  1, 10, 20000101, 25.641, 25.641,     0.650,     0.052
2690,  2,1,100,  2,  1, 10, 20000101, 74.359, 74.359,     0.513,     0.084
2691,  0,0,   0,  2,  1, 10, 20000101, 99.065, 99.065,     0.583,     0.099
2691,  1,1,100,  2,  1, 10, 20000101,  2.804,  2.804,     0.653,     0.043
2691,  4,1,100,  2,  1, 10, 20000101, 26.168, 26.168,     0.675,     0.057
2691,  2,1,100,  2,  1, 10, 20000101, 69.159, 69.159,     0.547,     0.089
2691,  3,1,100,  2,  1, 10, 20000101,  0.935,  0.935,     0.536,     0.000
2692,  0,0,   0,  2,  1, 10, 20000101,100.000,100.000,     0.170,     0.036
2692,  2,1,100,  2,  1, 10, 20000101, 80.882, 80.882,     0.176,     0.036
```

Example: (part of a) RUM file.

Each entry contains twelve fields. These fields (from left to right) are:

- Region-ID: the region-ID's as determined by the Regions IMG specified in the SPU file;
- Class-ID:
  - for overall regional means: 0;
  - for hard classification: the class-ID's as determined by the Classes IMG specified in the SPU file;
  - for area fractions: the class-ID's as assigned to the AFIs specified in the SPU file.
- unmixing method:
  - for overall regional means: 0;
  - for hard classification: 1;
  - for area fractions: 2.
- threshold:
  - for overall regional means: 0 (not applicable);
  - for hard classification: 100 (not applicable);
  - for area fractions: threshold values (%) assigned to the AFIs specified in the SPU file.
- Sensor-ID: the id of the Sensor specified in the Extract RUM tool parameters;
- Variable-ID: the id of the Variable specified in the Extract RUM tool parameters;
- periodicity: the periodicity of the input IMG in days. This value originates from the "days" entry in the input IMG HDR file. In case this entry is not available, the periodicity field value will be 0. In case the extracted data will be uploaded into the projects RUM database, this value needs to be present, and should be 1, 10, 30 or 360;
- date: the date of the input IMG in YYYYMMDD format. This value originates from the "date" entry in the input IMG HDR file. In case this entry is not available, the date field value will be 0. In case the extracted data will be uploaded into the projects RUM database, this value needs to be present;
- RA1: relative area 1 - lines with RA1 = 0 are considered to indicate a missing value;
- RA2: relative area 2;
- Mean: the calculated regional means or regional frequencies;
- Standard deviation;

3.47. RUM to Database

Goal

Upload extracted RUM files into the projects RUM database, used to visualize them via the RUM Chart utility.

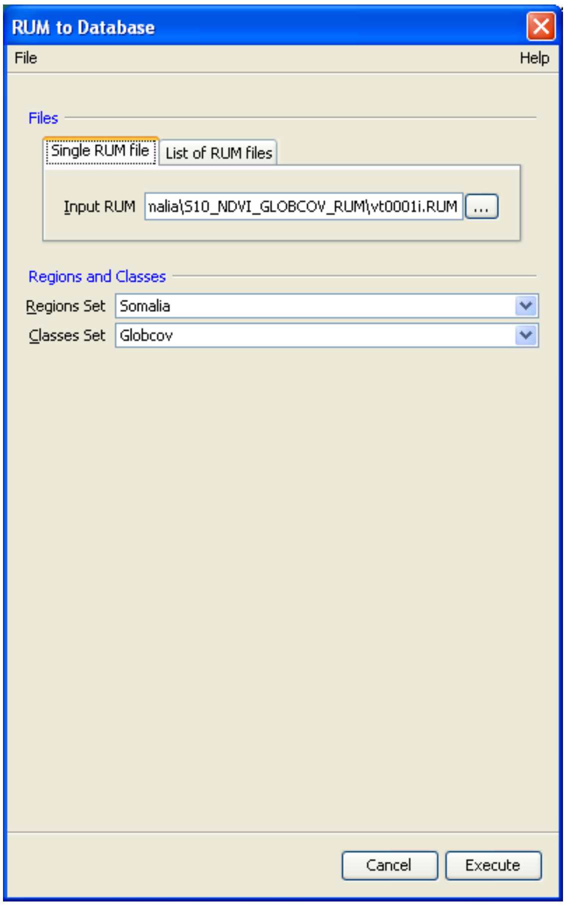
Parameters

- the Regions Set containing the Regions used in the RUM extraction;
- the Classes Set containing the (land use/land cover) Classes used in the RUM extraction;

Remarks:

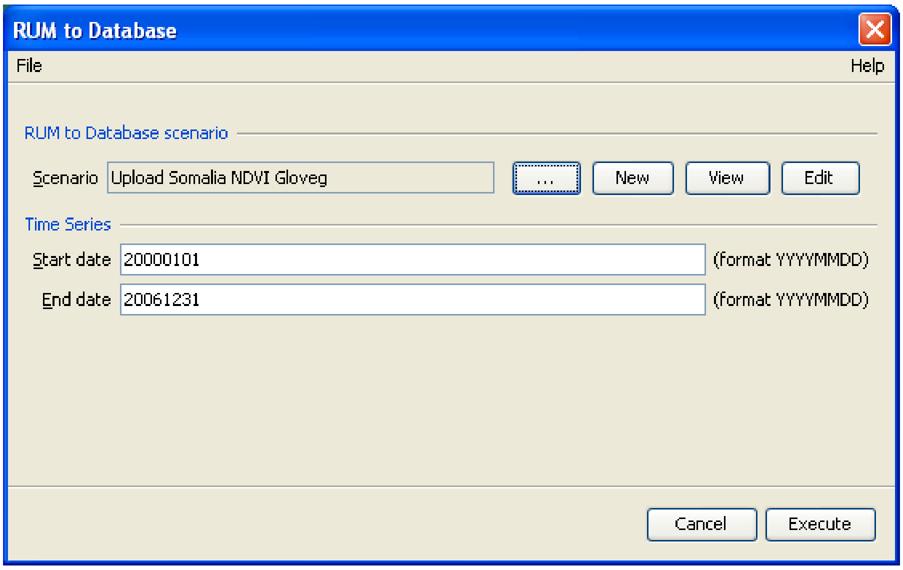
- existing database entries will silently be overwritten. In case the 'new' entry indicates a missing value (RA1 = 0), the existing entry will be removed;
- in case none of the RUM file entries can be uploaded in the database, an error will be issued;
- in case not all RUM file entries can be uploaded in the database, a warning will be issued;
- evidently RUM file entries can only be uploaded if they honour the RUM file format and are complete (e.g. date and periodicity fields must be present and valid);
- RUM file entries can only be uploaded if the Regions-ID and Class-ID in the RUM file entries are known in the Regions Set and Classes Set specified in the tools parameters;

Tool



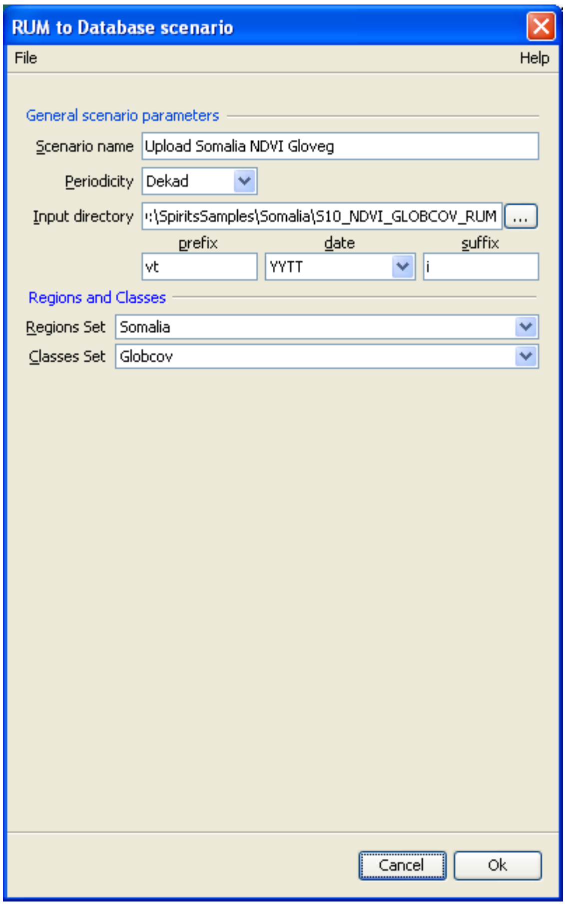
RUM to Database Tool example

Time Series



RUM to Database Time Series example

Scenario



RUM to Database Scenario example

3.48. Reconvert RUM to IMG

Goal

Create IMGs, containing the RUM values per region, from a regions IMG and a RUM file extracted for these regions.

Parameters

- the unmixing method. This method must be Overall mean, or it must be identical to the one used in the RUM extraction;
- the Class Id (unless the Overall mean unmixing method is selected) of the land use/land cover class for which the RUM values will be reconverted. These Class Id's must be identical to the ones used in the RUM extraction;
- the Threshold (only in case the Weighted means unmixing method, aka Area fractions method, is selected). These thresholds must be identical to the ones used in the RUM extraction;
- the regions IMG, using identical Region-ID's as the regions IMG used in the RUM extraction;
- an optional mask IMG and mask range. Pixels with values in the regions IMG, beyond the mask range will be flagged in the output IMG;
- the spectral parameters of the output IMG. A minimal set can be specified manually, but preferably they are specified via a reference HDR file. These parameters specify the data type and scaling of the values in the output IMG. The minimal set of parameters consists of:
    - the data type of the output IMG: Byte(8 bit, unsigned), Integer (16 bit, signed), Long (32 bit, signed) or Float (32 bit);
    - the values name and unit to be used (i.e. ***Vname***, ***Vunit*** - optional);
    - the minimum and maximum digital values of the significant range to be used (i.e. ***Vlo***, ***Vhi***);
    - the intercept and slope values of the linear scaling to be used (i.e. ***Vint***, ***Vslo***);
    - the flag values to be used for pixels in flagged regions, for masked pixels and for pixels with missing RUM values;

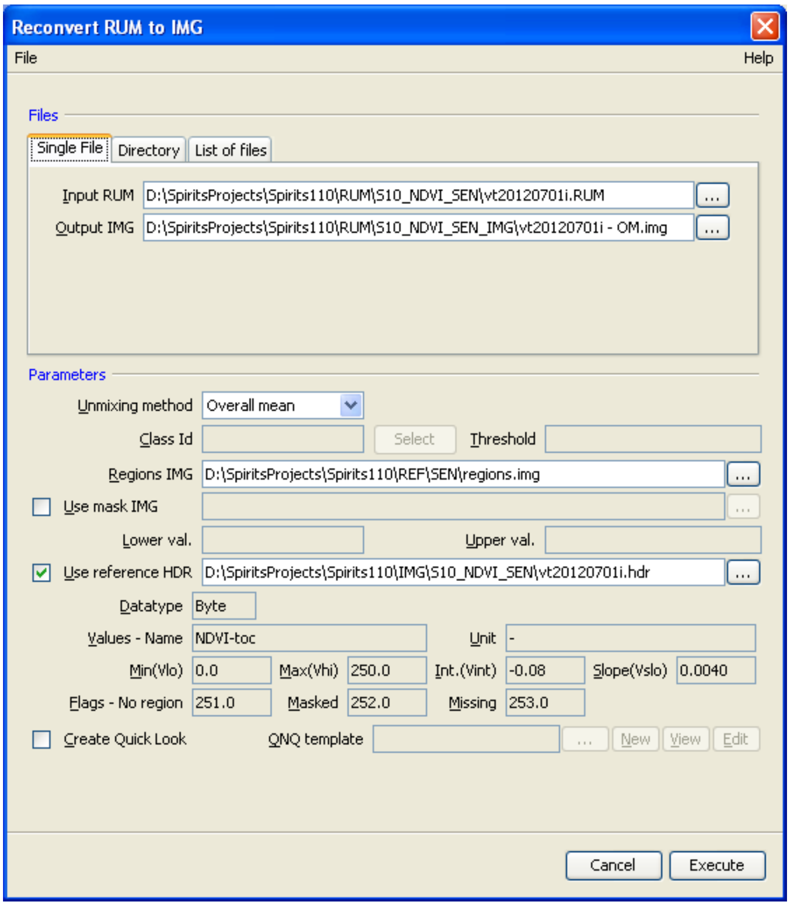
  Reminder: The physical values in the output IMGs (in this case RUM values) are specified by:
  Physical value = Vint + Vslo * Digital IMG value
  with Vint and Vslo the intercept and slope specified in the values entry in the IMG HDR:
  values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo }.
  Digital IMG values in the [Vlo, Vhi] range are considered significant, values outside this range are regarded as flags.

- an optional QNQ file to be used as template. If specified, additional to the output IMG, an output PNG file will be created, based on the created output IMG and the specified QNQ template. This PNG file will have the same name and location as the output IMG, but with the PNG extension.

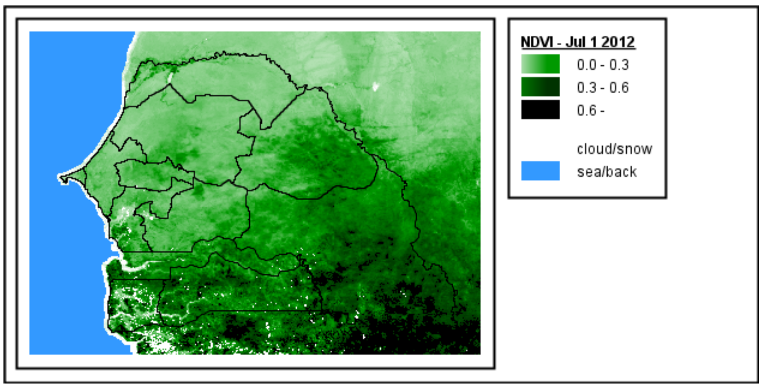
Remarks:

- the spatial features (projection, framing, resolution,...) of the output IMG, will be those of the regions IMG. Typically the regions IMG used would be a thinned version of the regions IMG used for the RUM extraction;
- the reference HDR file could typically be derived from the HDR of the original input IMG from which the RUM values were extracted. It needs the entry "flags = {x=NoReg, y=Masked, z=Missing}" to be literally present.

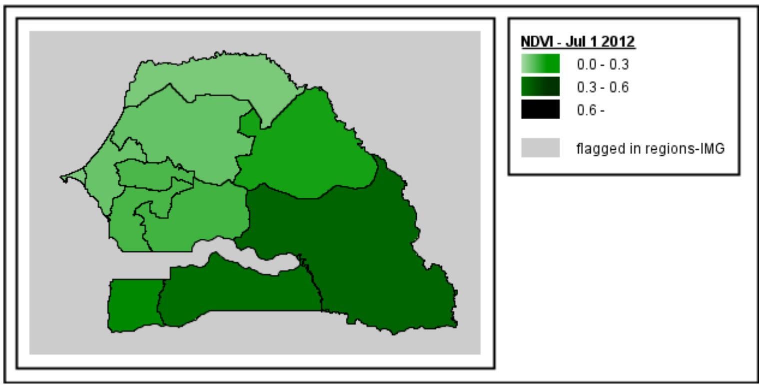
Tool



Reconvert RUM to IMG Tool example

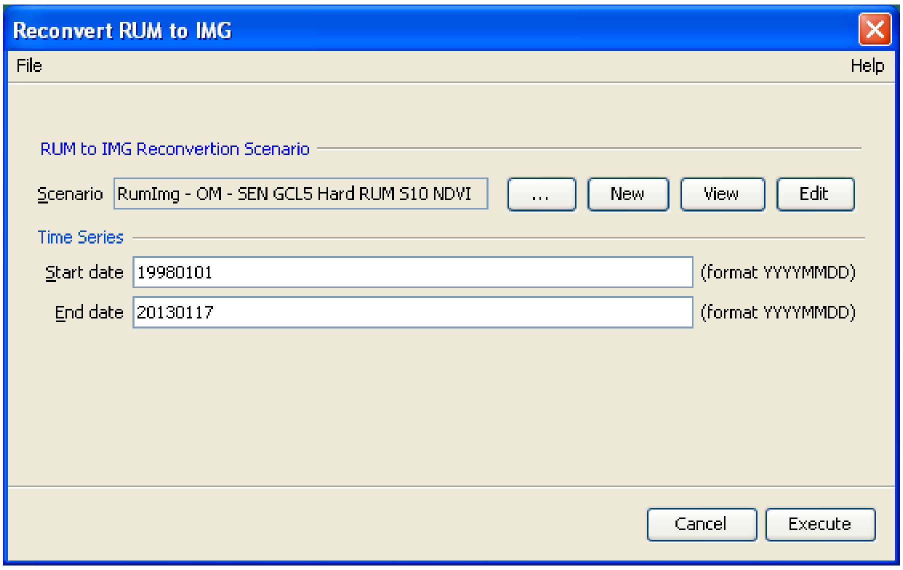


original NDVI IMG from which RUM values were extracted.



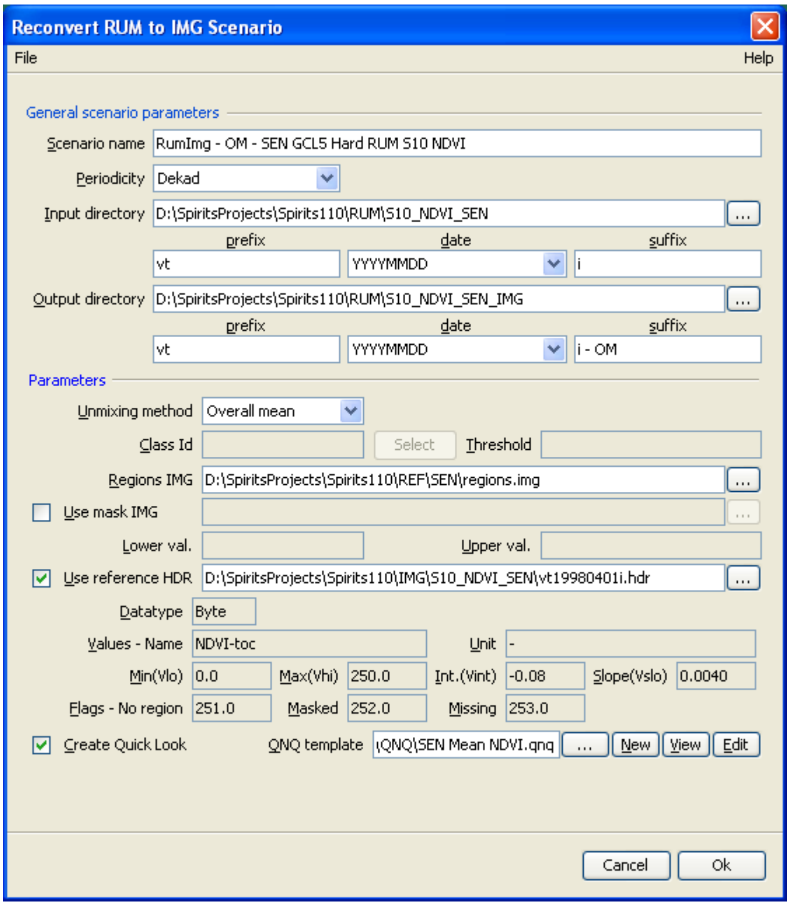
reconverted IMG with NDVI RUM values (overall mean) per region.

Time Series



Reconvert RUM to IMG Time Series example

Scenario



Reconvert RUM to IMG Scenario example

4. RUM Statistics

4.1. RUM Database

4.1.1. Introduction

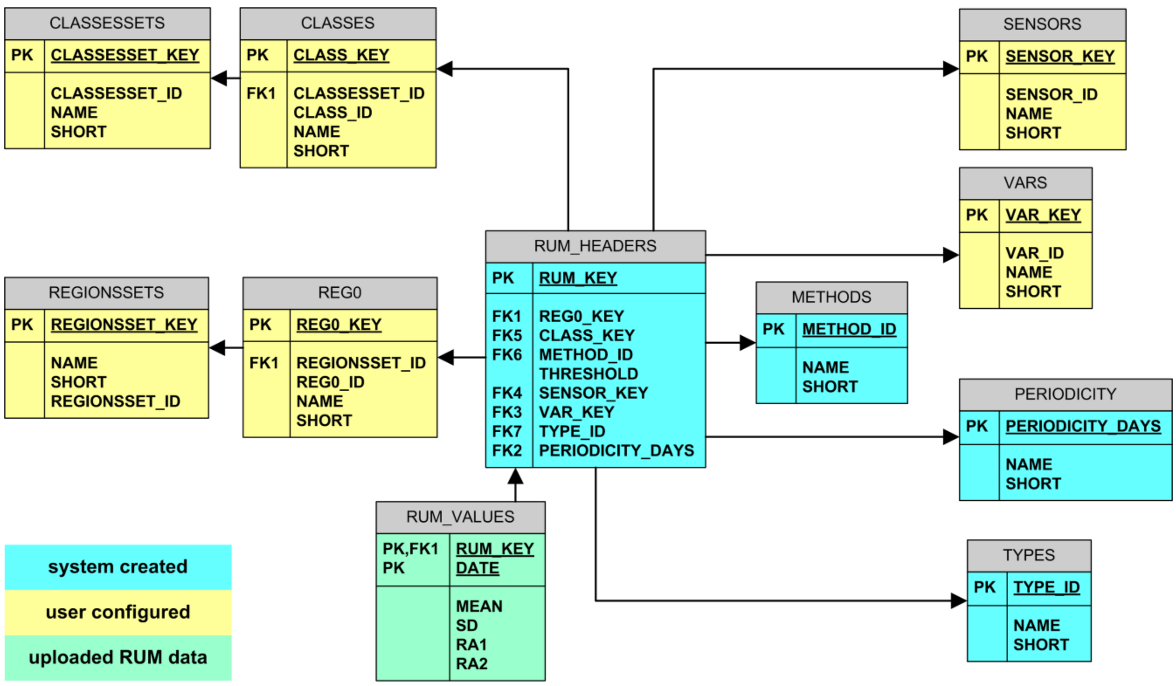
Each Spirits project contains a small in-process database to store RUM (Regional and Unmixed Means) data, obtained via the Extract RUM tool.

The purpose of this database is to enable a fluent visualisation of this data via the RUM Chart utility.

Before actual RUM values can stored in the database, it needs to be accommodated with ancillary data needed to identify the RUM values: the sensors, variables, regions and land use classes the RUM values belong to.

The RUM values can be uploaded during the RUM extraction process, by the Extract RUM tool, or afterwards by the RUM to Database tool.

Once the database is configured and data is uploaded, the available temporal series of RUM values (RUM Datasets) can be inspected via the RUM Browser, and sent to the RUM Chart utility.



Rum database structure

4.1.2. Sensors

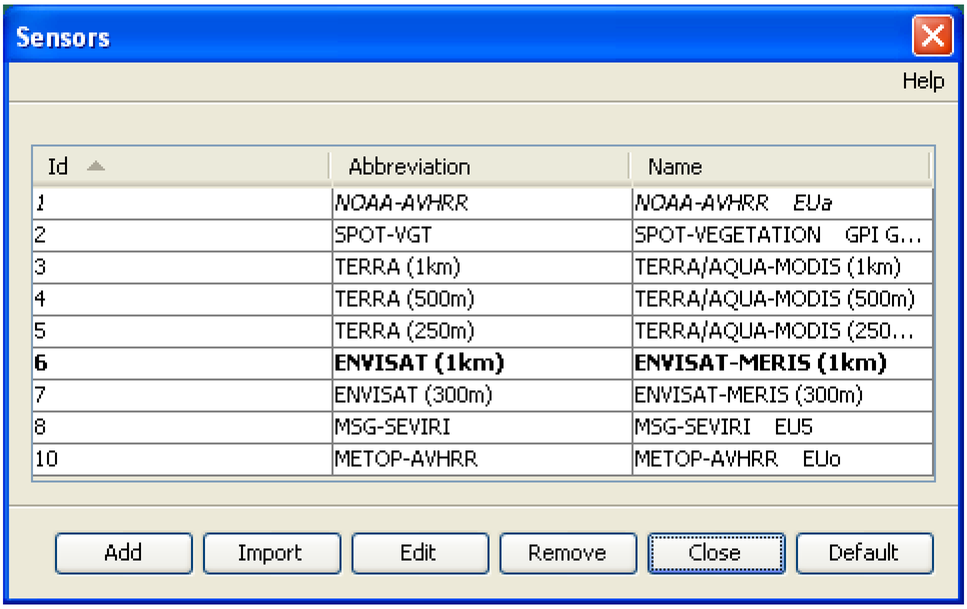
Sensor attributes

Any sensor in a Spirits project sensors collection has following attributes:

- Id: a unique integer between 1 and 32767. This Id is one of the links between the data extracted by the Extract RUM tool and the data imported into the RUM database;
- Name: a unique non-empty sting, maximum 256 characters, describing the sensor;
- Abbreviation: a unique non-empty string, maximum 16 characters, used as a sensor mnemonic.

Sensors panel

The sensors collection can be inspected and manipulated in the Sensors panel.



Sensors panel example

Following functions are available in the Sensors panel:

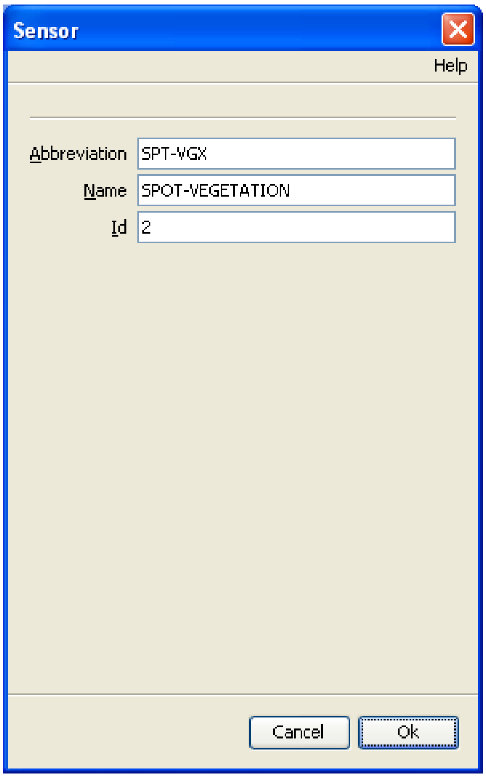
Add	Add a new sensor to the sensors collection of the project.
Import	Import sensors from an text file.
Edit	Edit selected sensor (only name and abbreviation can be modified).
Remove	Remove selected sensor. All data linked to this sensor will be removed from the project database.
Close	Close the Sensors panel.
Default	Set the selected sensor as default sensor. (Used for example in the Extract RUM tool and the RUM database browser).
Copy/Save As (via right-click on the table in the panel)	Copy the sensors table to the clipboard / Save the sensors table as a CSV file.

Add sensor:

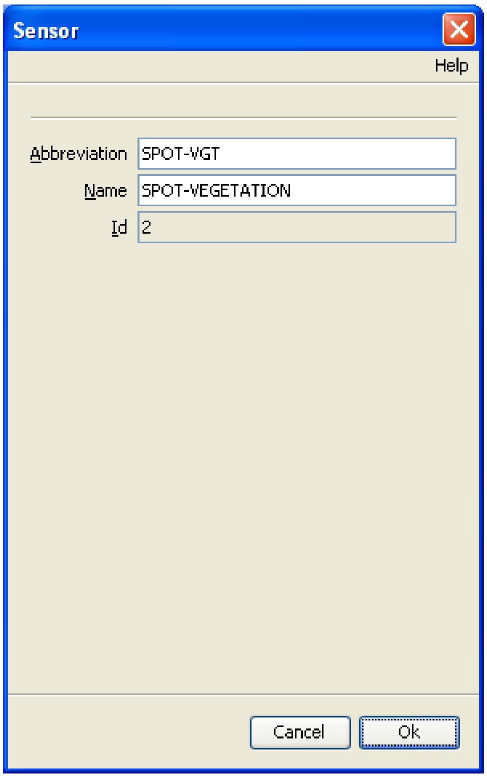
Individual sensors can be added to the sensors collection via the Add function in the Sensors panel. A new panel will appear where the new sensors attributes can be specified. Each of these attributes need to be unique with respect to the sensors already present in the collection.

Edit sensor:

The sensor selected in the Sensors panel can be edited via the Edit function in the Sensors panel. A new panel will appear where the sensors name and/or abbreviation attributes can be modified. The Id of an existing sensor cannot be changed.



Add sensor example



Edit sensor example

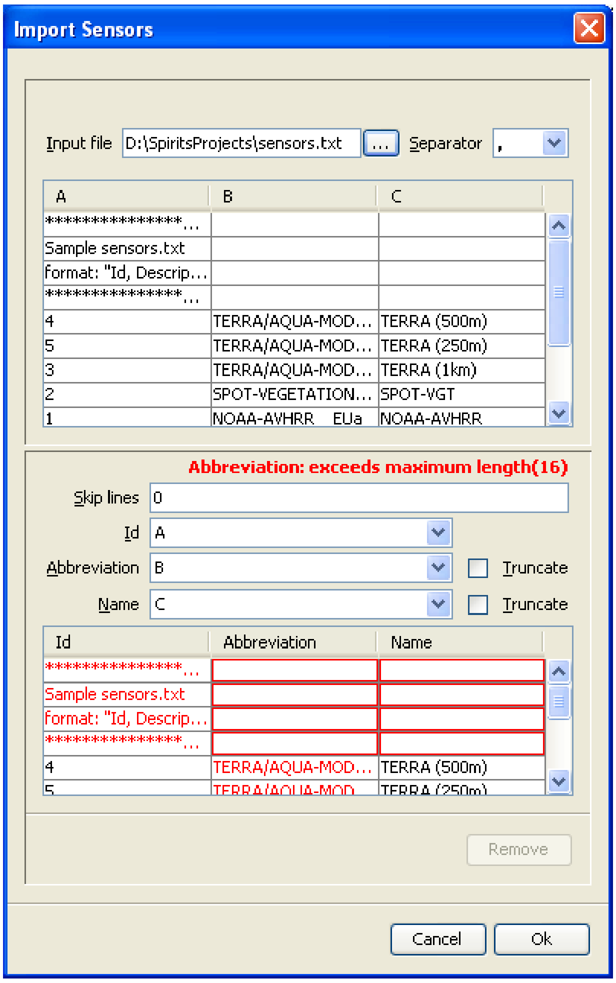
Import sensors:

Via the import function in the Sensors panel, a set of sensors can be imported from a text file. Typically this would be a 'comma-separated values' (CSV) file containing a sensor id, name and description on each line.

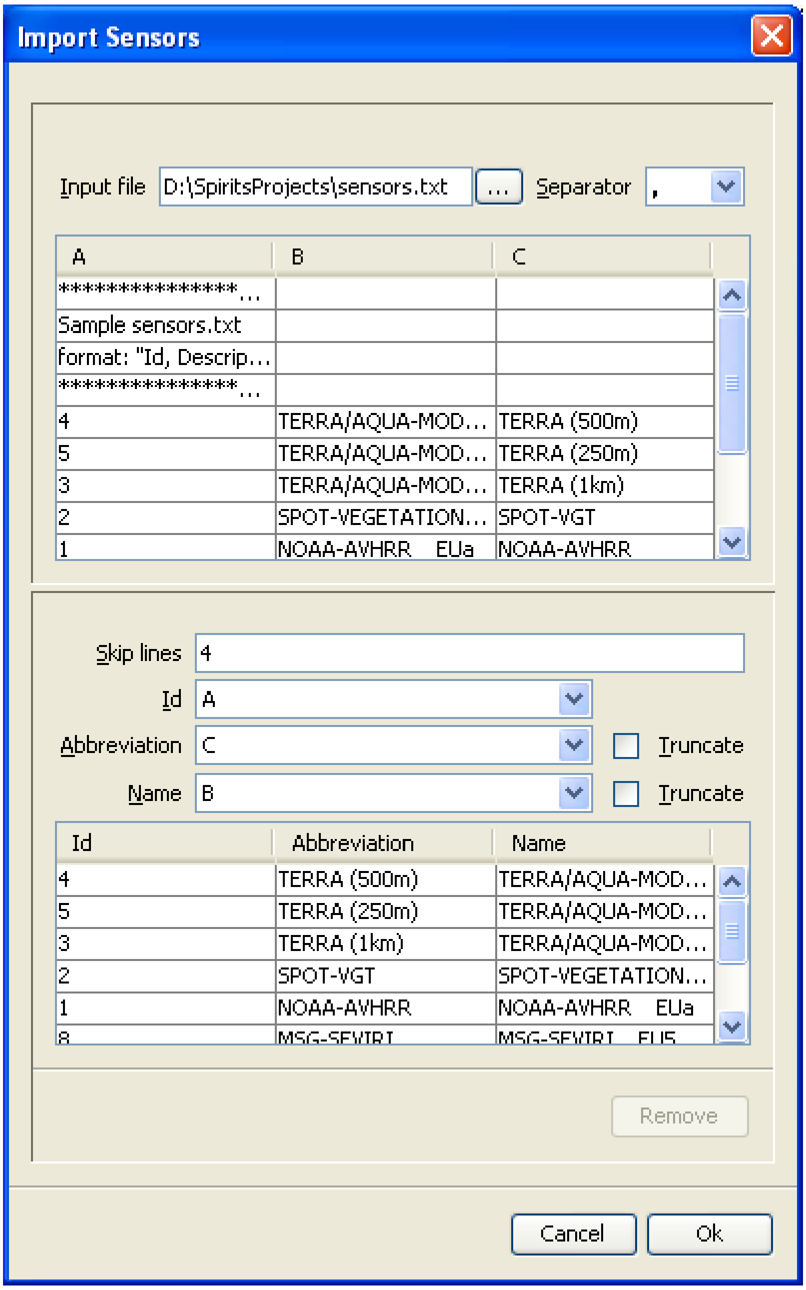
```
****************************************************
Sample sensors.txt
format: "Id, Description, Short"
****************************************************

4, TERRA/AQUA-MODIS (500m), TERRA (500m)
5, TERRA/AQUA-MODIS (250m)    EUq, TERRA (250m)
3, TERRA/AQUA-MODIS (1km), TERRA (1km)
2, SPOT-VEGETATION    GPI GLD...CCA, SPOT-VGT
1, NOAA-AVHRR    EUa, NOAA-AVHRR
8, MSG-SEVIRI    EU5, MSG-SEVIRI
10, METOP-AVHRR    EUo, METOP-AVHRR
7, ENVISAT-MERIS (300m), ENVISAT (300m)
6, ENVISAT-MERIS (1km), ENVISAT (1km)
```

example sensors text file

Importing sensors is an interactive process executed by the Import Sensors panel.



- In the Import Sensors panel, the file to be imported and the separator character to be used have to be chosen.
- An attempt is made to parse the file. The result is shown in the top half of the panel in tabular form.
- In the middle part of the panel, the sensor attributes (Id, Abbreviation and Name) must be assigned one of the columns found by the parser.
- Since typical files contain some leading lines describing the file contents, the Skip lines field allows to specify a number of lines to be ignored.
- In the bottom half of the panel the resulting entries to be imported as sensor are shown in a table. These entries are tested against the sensors already present in the sensors collection and against other entries in this table. These test will for example prevent duplicate values, ensure integer values for the Id, etc.
- Invalid values are indicated in the table and an error message is displayed on top of the table.
- The values in the table can be edited or entries can be removed.
- Upon selection of the Ok button the valid entries are imported as new sensors into the sensors collection, whereas invalid entries are ignored.



specify text file and separator

parsing result: the parser finds three columns in the file

skip the first lines containing the description of the file

assign columns to Id, Abbreviation and Name

resulting entries which can be edited and imported.

Import Sensors example

Remarks:

- The separator characters supported by the parser are the comma, semicolon, colon and tab characters.
- Unless the tab character is used as separator, tabs are converted into four blanks by the parser.
- Parsed columns may be assigned multiple times. E.g. in case the text file only contains Id's, which would be parsed into the single first column 'A', this column 'A' could be assigned to the sensors Id, its Abbreviation and its Name as well. The 'real' Name and Abbreviation values could then be specified by editing them in the table itself, prior to importing the entries, or after importing them via the Edit function in the Sensors panel.

4.1.3. Variables

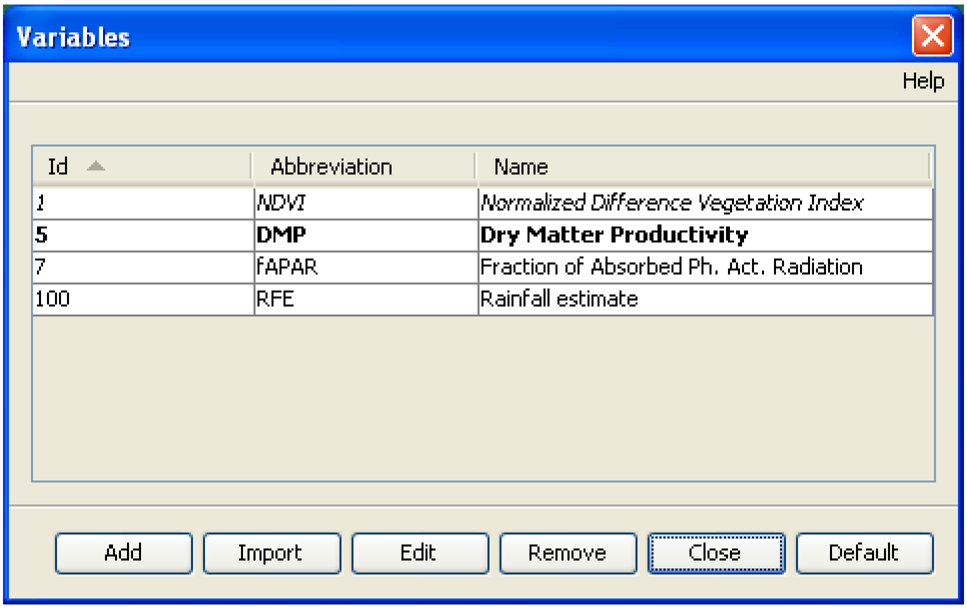
Variable attributes

Any variable in a Spirits project variables collection has following attributes:

- Id: a unique integer between 1 and 32767. This Id is one of the links between the data extracted by the Extract RUM tool and the data imported into the RUM database;
- Name: a unique non-empty sting, maximum 256 characters, describing the variable;
- Abbreviation: a unique non-empty string, maximum 16 characters, used as a variable mnemonic.

Variables panel

The variables collection can be inspected and manipulated in the Variables panel.



Variables panel example

Following functions are available in the Variables panel:

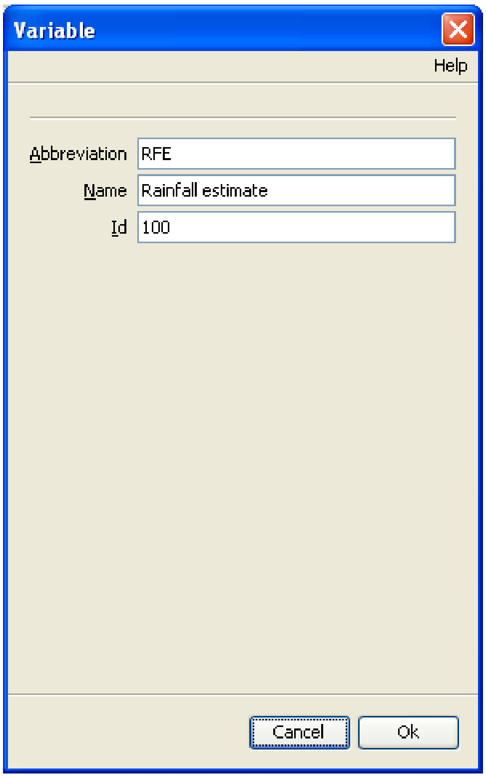
Add	Add a new variable to the variables collection of the project.
Import	Import variables from an text file.
Edit	Edit selected variable (only name and abbreviation can be modified).
Remove	Remove selected variable. All data linked to this variable will be removed from the project database.
Close	Close the Variables panel.
Select	Set the selected variable as default variable. (Used for example in the Extract RUM tool and the RUM database browser).
Copy/Save As (via right-click on the table in the panel)	Copy the variables table to the clipboard / Save the variables table as a CSV file.

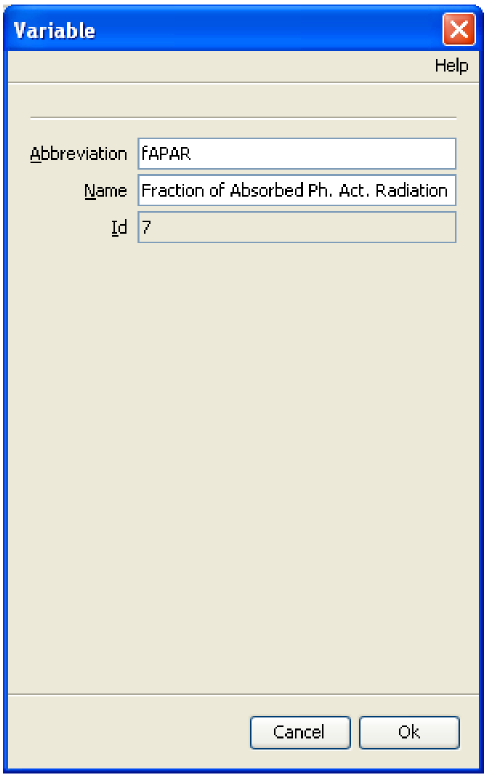
Add variable:

Just as for sensors individual variables can be added to the variables collection via the Add function in the Variables panel. A new panel will appear where the new variables attributes can be specified. Each of these attributes need to be unique with respect to the variables already present in the collection.

Edit variable:

The variable selected in the Variables panel can be edited via the Edit function in the Variables panel. A new panel will appear where the variables name and/or abbreviation attributes can be modified. The Id of an existing variable cannot be changed.





Add variable example

Edit variable example

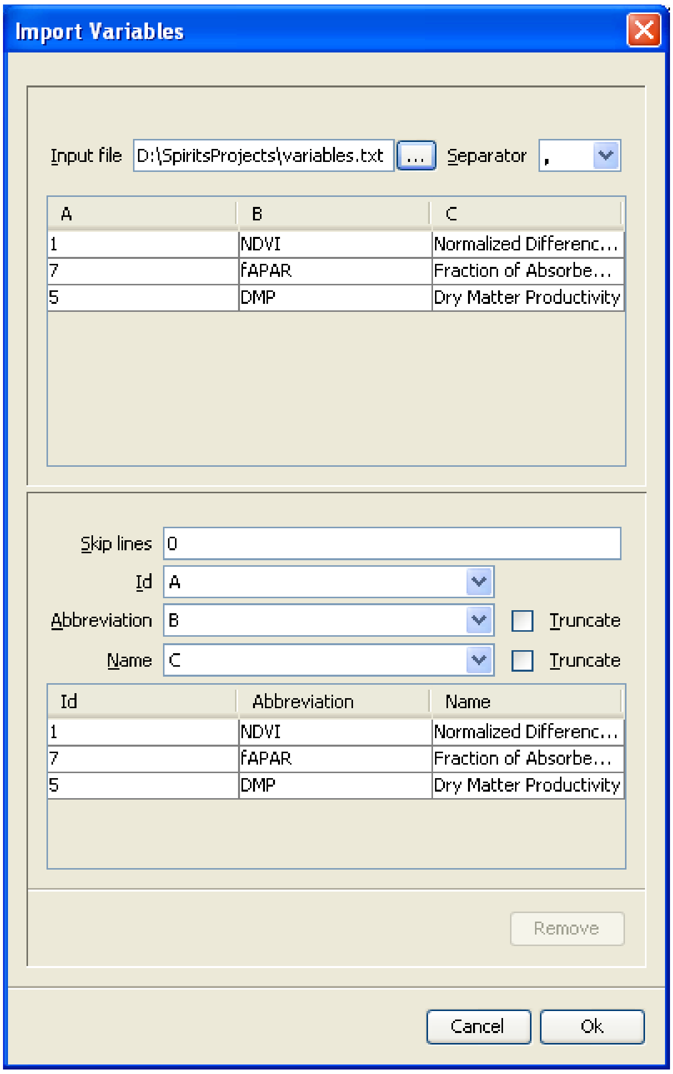
Import variables:

Via the import function in the Variables panel, a set of variables can be imported from a text file. Typically this would be a 'comma-separated values' (CSV) file containing a variable id, name and description on each line.

```
1, NDVI, Normalized Difference Vegetation Index
7, fAPAR, Fraction of Absorbed PAR-Radiation
5, DMP, Dry Matter Productivity
```

example variables text file

The import of variables is driven by the Import Variables panel, which functions exactly the same as the Import Sensors panel.



Import Variables panel example

4.1.4. Regions

A Regions Set assembles a set of Regions. Regions can only be specified in the context of a Regions Set.

Regions Set attributes

Any Regions Set in a Spirits project Regions Set collection has following attributes:

- Id: a unique integer between 1 and 32767;
- Name: a unique non-empty sting, maximum 256 characters, describing the Regions Set;
- Abbreviation: a unique non-empty string, maximum 16 characters, used as a Regions Set mnemonic.

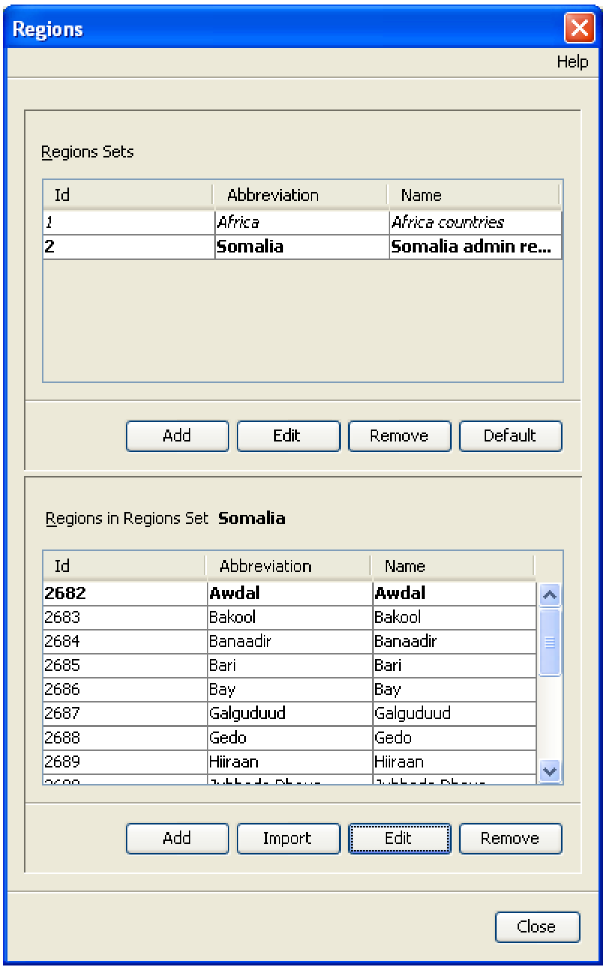
Region attributes

Any Region in a Spirits project Regions collection has following attributes:

- Regions Set: the Regions Set it belongs to: one of the regions sets in the Regions Set collection;
- Id: an integer unique over the regions in the Regions Set it belongs to. This Id is one of the links between the data extracted by the Extract RUM tool and the data imported into the RUM database;
- Name: a non-empty sting, maximum 256 characters, describing the variable;
- Abbreviation: a non-empty string, maximum 16 characters, used as a variable mnemonic.

Regions panel

The Regions Set collection and the Regions can be inspected and manipulated in the Regions panel.



Regions panel example

Following functions are available in the Regions panel:

Regions Sets functions	
Add	Add a new Regions Set to the Regions Set collection of the project.
Edit	Edit selected Regions Set (only name and abbreviation can be modified).
Remove	Remove selected Regions Set. All data linked to this Regions Set will be removed from the project database.
Select	Set the selected Regions Set as default.
Copy/Save As (via right-click on the table in the table)	Copy the Regions Sets table to the clipboard / Save the Regions Sets table as a CSV file.

Remark: since typically the number of Regions Sets in a Spirits project will be limited, there is no Import function.

Regions functions	
Add	Add a new Region to the Regions collection of the project.
Import	Import Regions from an text file, a HDR file or SHP file.
Edit	Edit selected Region (only name and abbreviation can be modified).
Remove	Remove selected Region. All data linked to this Region will be removed from the project database.
Copy/Save As (via right-click on the table in the table)	Copy the variables table to the clipboard / Save the variables table as a CSV file.
Close	Close the Regions panel.

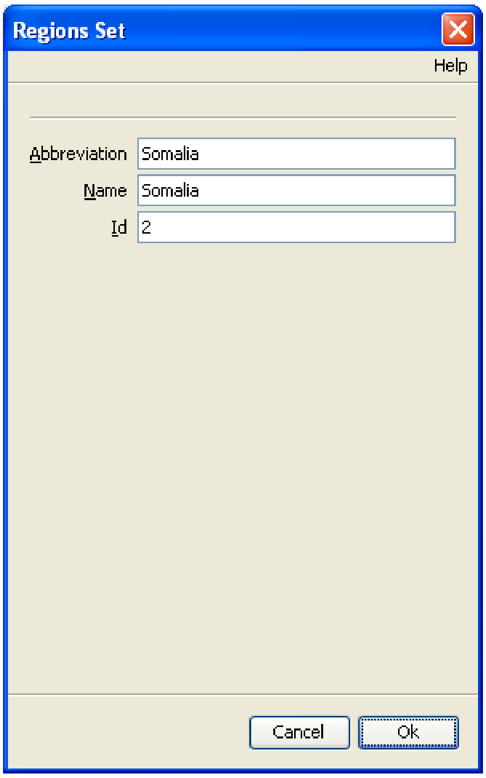
Remark: since Regions can only be specified in the context of a Regions Set, the Regions Set must be available (added) before Regions can be specified (added or imported).

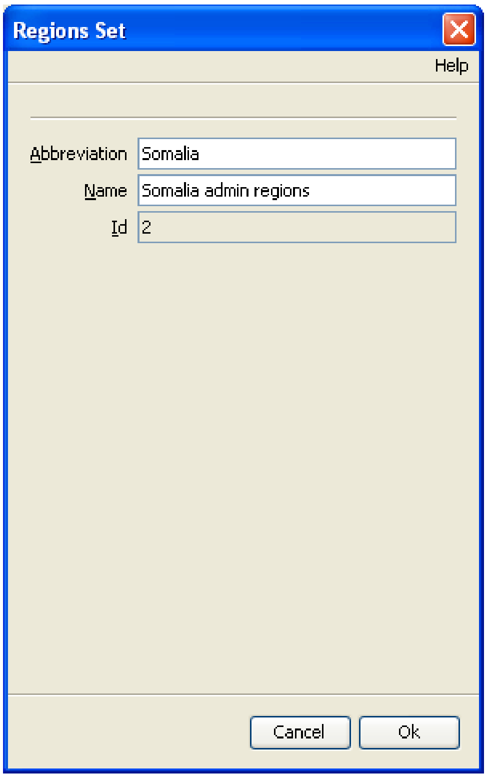
Add Regions Set:

Regions Sets can be added to the Regions Set collection via the Add function in the regions sets part (top half) of the Regions panel. A new panel will appear where the new Regions Set attributes can be specified. Each of these attributes need to be unique with respect to the Regions Sets already present in the collection.

Edit Regions Set:

The Regions Set selected in the Regions panel can be edited via the Edit function in the regions sets part of the Regions panel. A new panel will appear where the Regions Set Name and/or Abbreviation attributes can be modified. The Id of an existing Regions Set cannot be changed.





Add Regions Set example

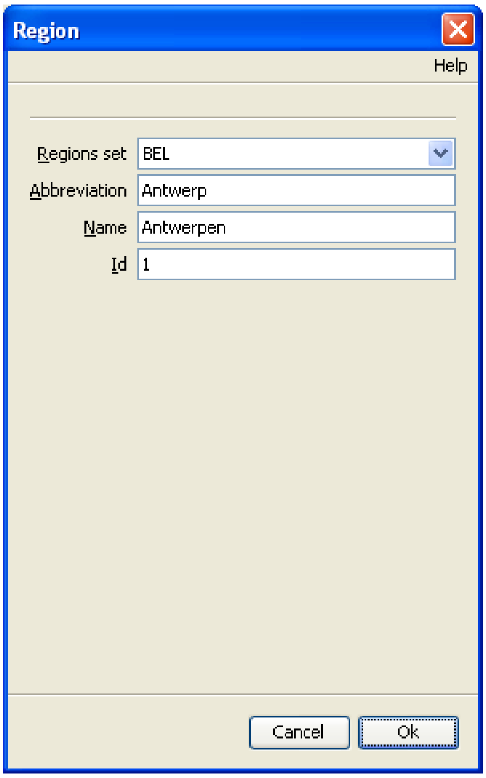
Edit Regions Set example

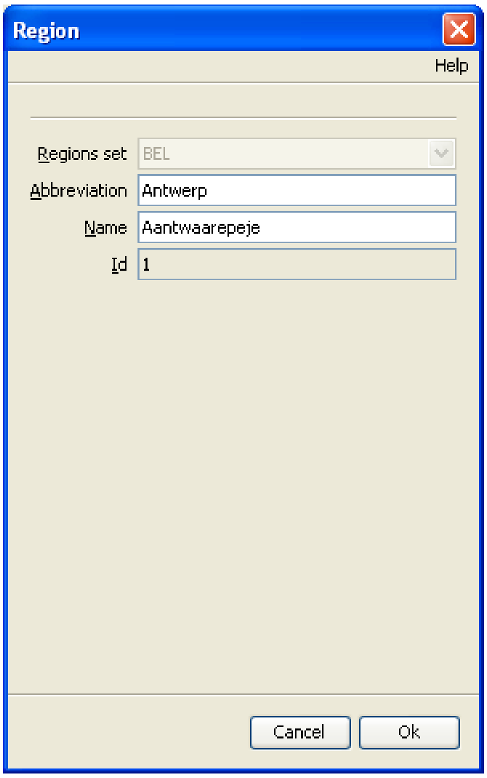
Add Region

Regions can be added to the Regions collection via the Add function in the regions part (bottom half) of the Regions panel. A new panel will appear where the new Region attributes can be specified. The Regions Set attribute of the Region must be chosen out of the existing Regions Sets. The Region Id attribute need to be unique with respect to the Regions in the same Regions Set. The Region Name and Abbreviation are not required to be unique, however it is advised to make them so.

Edit Region

The Region selected in the Regions panel can be edited via the Edit function in the regions part of the Regions panel. A new panel will appear where the Region Name and/or Abbreviation attributes can be modified. The Id and the Regions Set of an existing Region cannot be changed.





Add Region example

Edit Region example

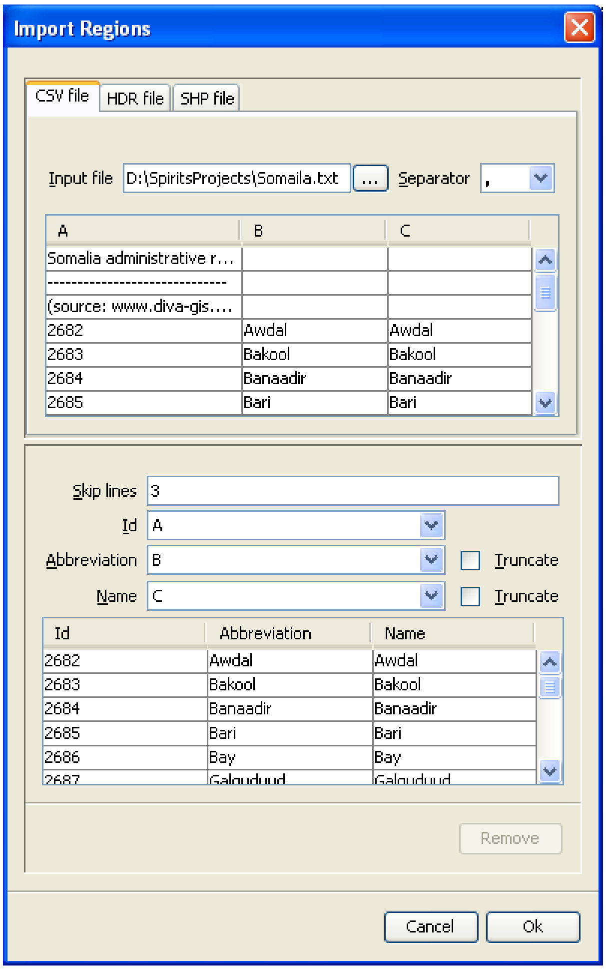
Import Regions

Via the import function in the regions part (bottom half) of the Regions panel, a set of regions can be imported from a text file, a HDR file or SHP file.

- In case of a text file, this would typically be a 'comma-separated values' (CSV) file containing a region id, name and description on each line.
- In case of an HDR file, its type needs to be an "ENVI Classification", it must contain the "classes" key which specifies as value the number of classes, and the "class names" key which specifies as values set a collection of names.
- In case of an SHP file, it needs to be accompanied by its DBF file, since it is actually the DBF file which will be parsed, and of which the columns found will be selectable for import. This file should typically contain a region id, name and description in its records.

The import of regions is driven by the Import Regions panel, which functions exactly the same as the Import Sensors panel.

Importing regions from text file example

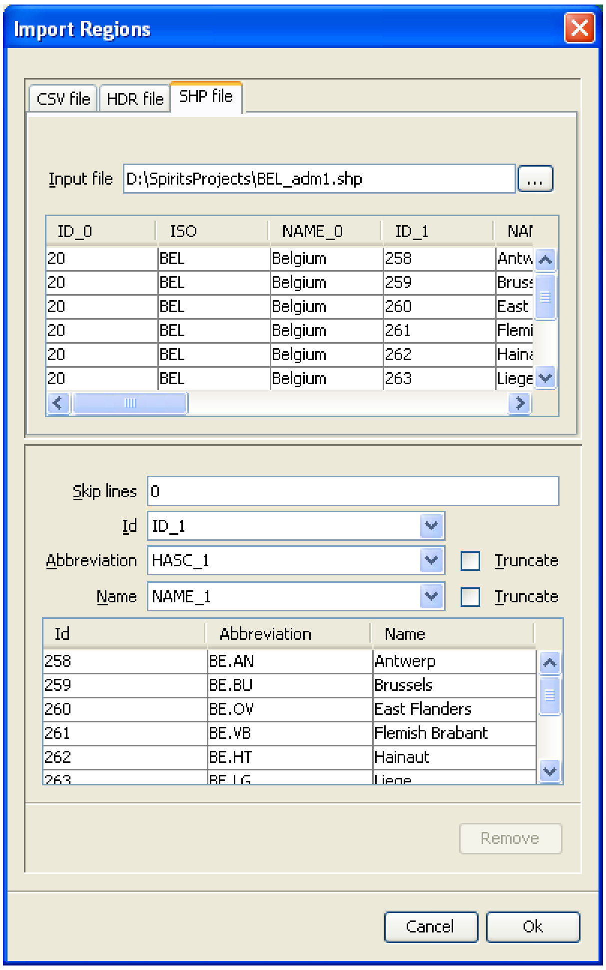


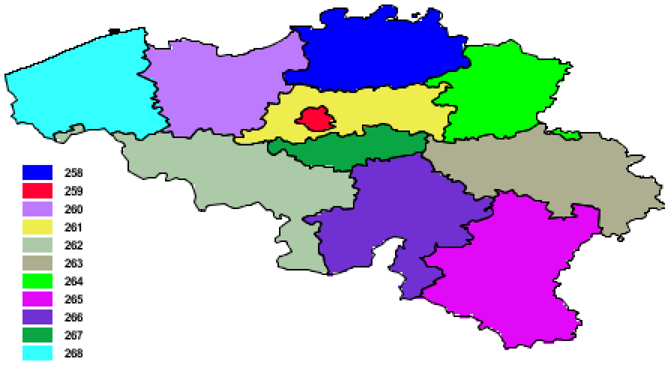
Import Regions - from text file example

```
Somalia administrative regions
------------------------------
(source: www.diva-gis.org)

2682, Awdal, Awdal
2683, Bakool, Bakool
2684, Banaadir, Banaadir
2685, Bari, Bari
2686, Bay, Bay
2687, Galguduud, Galguduud
2688, Gedo, Gedo
2689, Hiiraan, Hiiraan
2690, Jubbada Dhexe, Jubbada Dhexe
2691, Jubbada Hoose, Jubbada Hoose
2692, Mudug, Mudug
2693, Nugaal, Nugaal
2694, Sanaag, Sanaag
2695, Shab. Dhexe, Shabeellaha Dhexe
2696, Shab. Hoose, Shabeellaha Hoose
2697, Sool, Sool
2698, Togdheer, Togdheer
2699, Woqooyi Galbeed, Woqooyi Galbeed
```

regions text file

Importing regions from SHP example





SHP file

Import Regions - from SHP example

4.1.5. Classes

A Classes Set assembles a set of classes. Classes can only be specified in the context of a Classes Set.

Classes Set attributes

Any Classes Set in a Spirits project Classes Set collection has following attributes:

- Id: a unique integer between 1 and 32767;
- Name: a unique non-empty sting, maximum 256 characters, describing the Classes Set;
- Abbreviation: a unique non-empty string, maximum 16 characters, used as a Classes Set mnemonic.

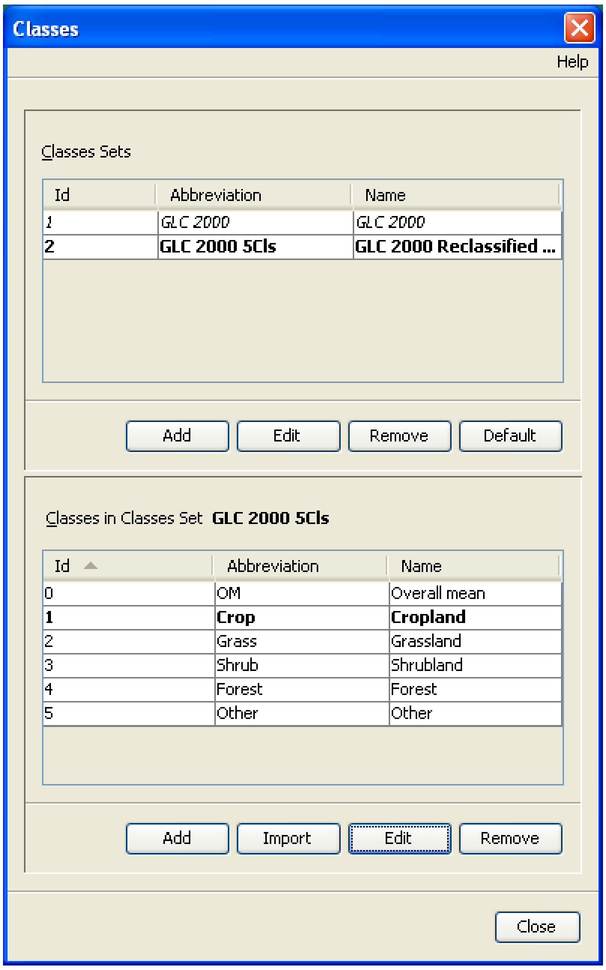
Class attributes

Any Class (short for land-use-class) in a Spirits project Class collection has following attributes:

- Classes Set: the Classes Set it belongs to: one of the classes sets in the Classes Set collection;
- Id: an integer unique over the classes in the Classes Set it belongs to. This Id is one of the links between the data extracted by the Extract RUM tool and the data imported into the RUM database;
- Name: a non-empty sting, maximum 256 characters, describing the variable;
- Abbreviation: a non-empty string, maximum 16 characters, used as a variable mnemonic.

Classes panel

The Classes Set collection and the Class collections can be inspected and manipulated in the Classes panel.



Classes panel example

Following functions are available in the Classes panel:

Classes Sets functions	
Add	Add a new Classes Set to the Classes Set collection of the project.
Edit	Edit selected Classes Set (only name and abbreviation can be modified).
Remove	Remove selected Classes Set. All data linked to this Regions Set will be removed from the project database.
Select	Set the selected Classes Set as default.
Copy/Save As (via right-click on the table in the table)	Copy the Classes Sets table to the clipboard / Save the Classes Sets table as a CSV file.

Remark: since typically the number of Classes Sets in a Spirits project will be limited, there is no Import function.

Remark: Each Classes Set will contain the special "Overall Mean" Class. This Class is added automatically to the Classes of a Classes Set as soon as a Classes set is specified (added). This Class cannot be deleted or modified.

Classes functions	
Add	Add a new Class to the classes collection of the project.
Import	Import Classes from an text file, a HDR file or SHP file.
Edit	Edit selected Class (only name and abbreviation can be modified).
Remove	Remove selected Class. All data linked to this Class will be removed from the project database.
Copy/Save As (via right-click on the table in the table)	Copy the classes table to the clipboard / Save the classes table as a CSV file.
Close	Close the Classes panel.

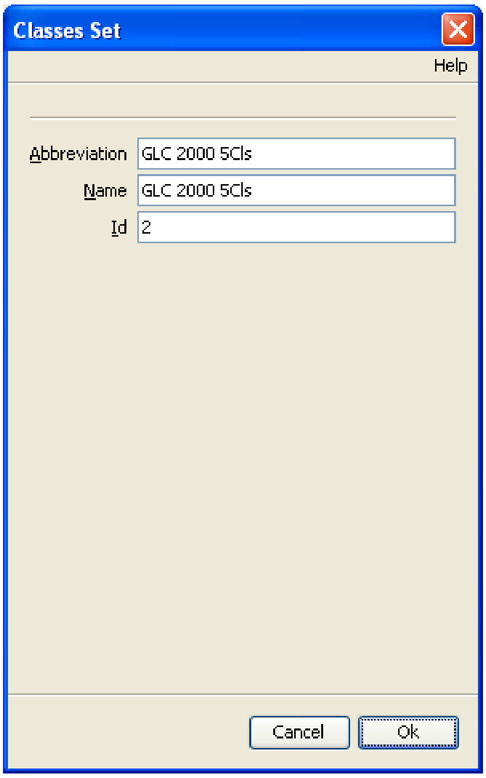
Remark: since Classes can only be specified in the context of a Classes Set, the Classes Set must be available (added) before Classes can be specified (added or imported).

Add Classes Set:

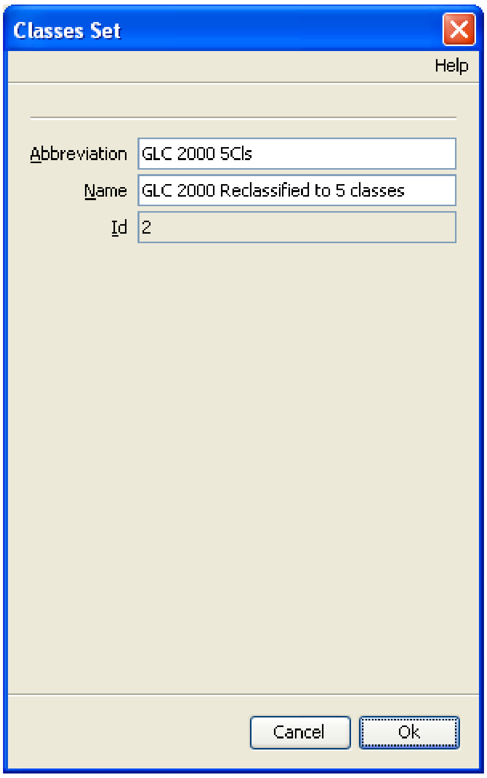
Classes Sets can be added to the Classes Set collection via the Add function in the classes set part (top half) of the Classes panel. A new panel will appear where the new Classes Set attributes can be specified. Each of these attributes need to be unique with respect to the Classes Sets already present in the collection.

Edit Classes Set:

The Classes Set selected in the Classes panel can be edited via the Edit function in the classes set part of the Classes panel. A new panel will appear where the Classes Set name and/or abbreviation attributes can be modified. The Id of an existing Classes Set cannot be changed.



Add Classes Set example



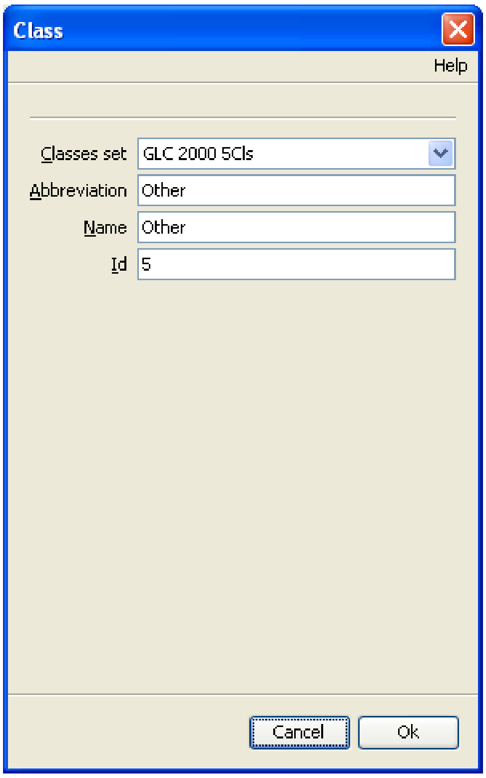
Edit Classes Set example

Add Class

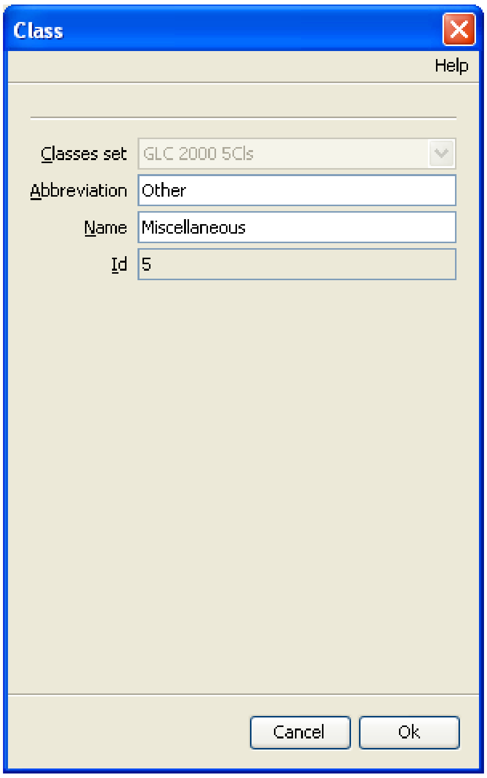
Classes can be added to the Classes collection via the Add function in the classes part (bottom half) of the Classes panel. A new panel will appear where the new Class attributes can be specified. The Classes Set attribute of the Class must be chosen out of the existing Classes Sets. The Class Id attribute needs to be unique with respect to the Classes in the same Classes Set. The Class Name and Class Abbreviation are not required to be unique, however it is advised to make them so.

Edit Class

The Class selected in the Classes panel can be edited via the Edit function in the classes part of the Classes panel. A new panel will appear where the Class Name and/or Abbreviation attributes can be modified. The Id and the Classes Set of an existing Class cannot be changed.



Add Class example



Edit Class Example

Import Classes

Via the import function in the classes part (bottom half) of the Classes panel, a set of classes can be imported from a text file, a HDR file or SHP file.

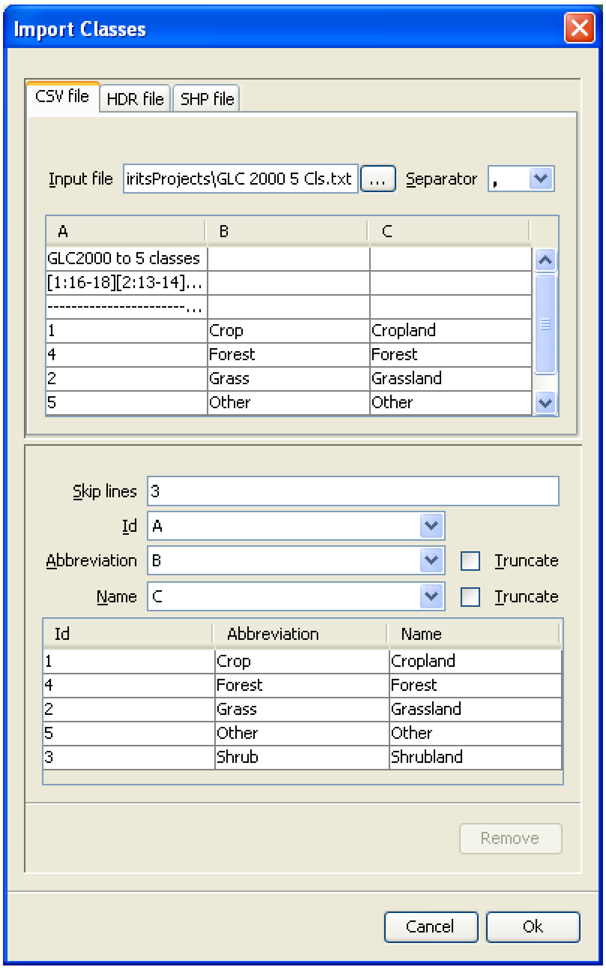
In case of a text file, this would typically be a 'comma-separated values' (CSV) file containing a class id, name and description on each line.

In case of an HDR file, its type needs to be an "ENVI Classification", it must contain the "classes" key which specifies as value the number of classes, and the "class names" key which specifies as values set a collection of names.

In case of an SHP file, it needs to be accompanied by its DBF file, since it is actually the DBF file which will be parsed, and of which the columns found will be selectable for import. This file should typically contain a class id, name and description in its records.

The import of classes is driven by the Import Classes panel, which functions exactly the same as the Import Sensors panel.

Importing classes from text file example



Import Classes from text file example

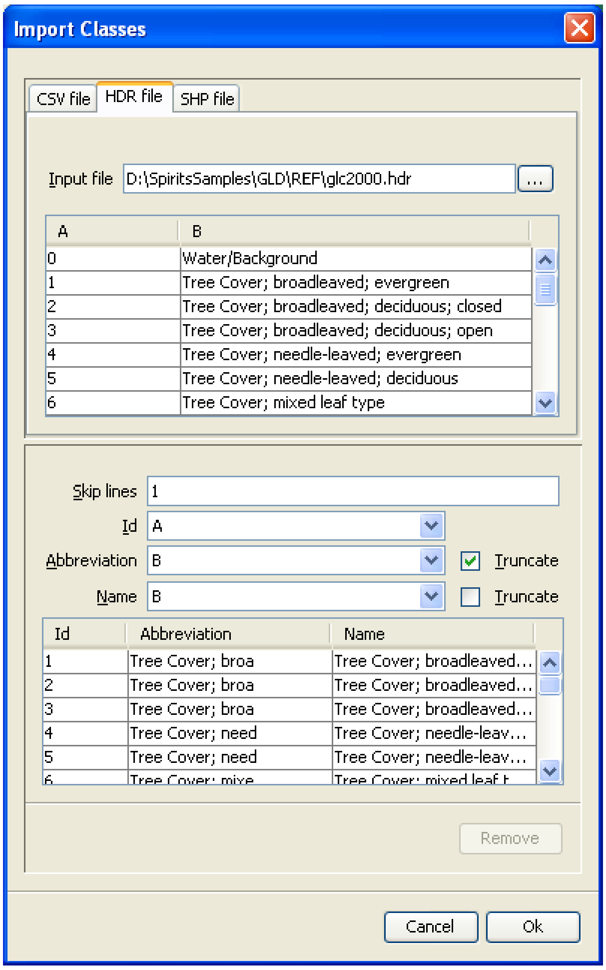
```
GLC2000 to 5 classes
[1:16-18][2:13-14][3:11-12]
-----------------------------
1, Crop, Cropland
4, Forest, Forest
2, Grass, Grassland
5, Other, Other
3, Shrub, Shrubland
```

classes text file

Importing classes from HDR example

```
ENVI
description = {GCL2000 Land Cover Classification.}
samples = 1920
lines = 698
bands = 1
header offset = 0
file type = ENVI Classification
...
classes = 24
class names = {
    Water/Background,
    Tree Cover; broadleaved; evergreen,
    Tree Cover; broadleaved; deciduous; closed,
...
    Bare Areas,
    Water Bodies (reset to 0),
    Snow and Ice,
    Artificial surfaces and associated areas,
    No data (small islands)
}
...
```

HDR file



Import Classes - from HDR example

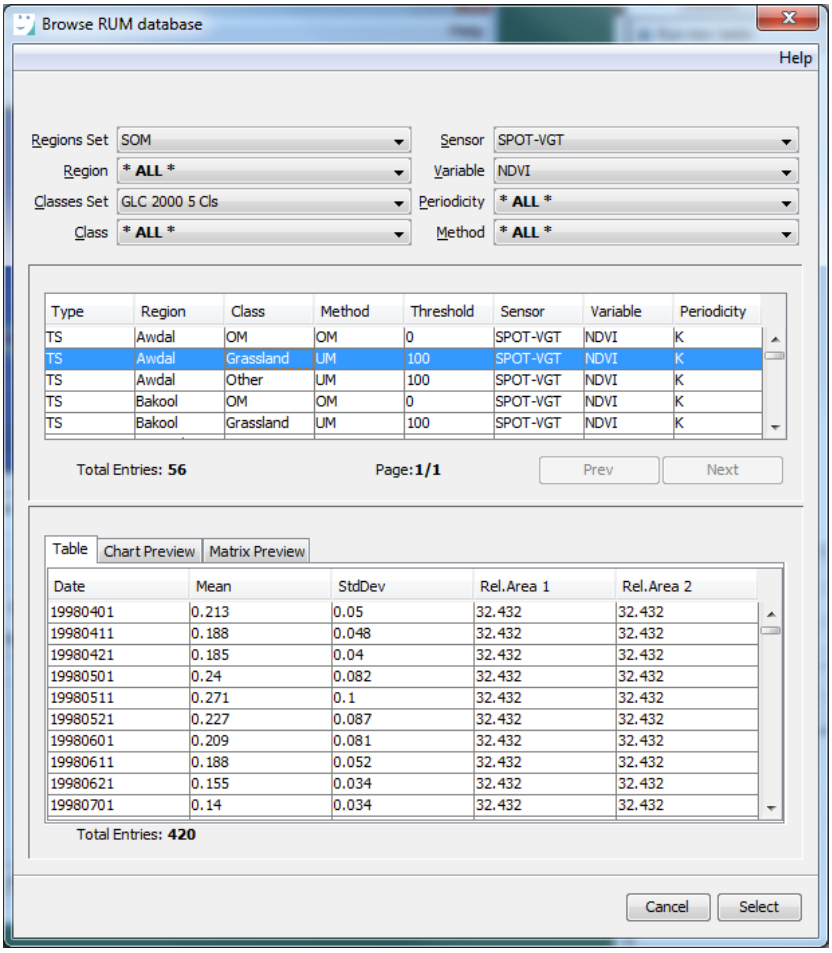
4.1.6. RUM Datasets

A RUM Dataset is a temporal series of RUM values identified by following attributes:

- a Sensor (from the user-configured Sensors collection in the database);
- a Variable (from the user-configured Variables collection in the database);
- a Region (from the user-configured Regions collection in the database);
- a Land Use Class (from the user-configured Classes collection in the database);
- an extraction method:
    - Overall mean - in case the RUM values were extracted via the regional means method
    - Unweighted mean - in case the RUM values were extracted via the hard classification method
    - Weighted mean- in case the RUM values were extracted via the method using area fraction IMGs
- a periodicity (Day, Dekad, Month or Year)
- a series Type:
    - a normal time series
    - one of the Long Term Averages obtained via the Long Term Average tool. When uploading RUM values into the database, Long Term Averages will be recognized by their specific year codes (1950 - 1964);
- a threshold. Only significant for the RUM values extracted via the method using area fraction IMGs.

4.1.7. RUM Browser

The RUM Browser enables the inspection and selection of the RUM Datasets in the database.



Filter area

Available datasets

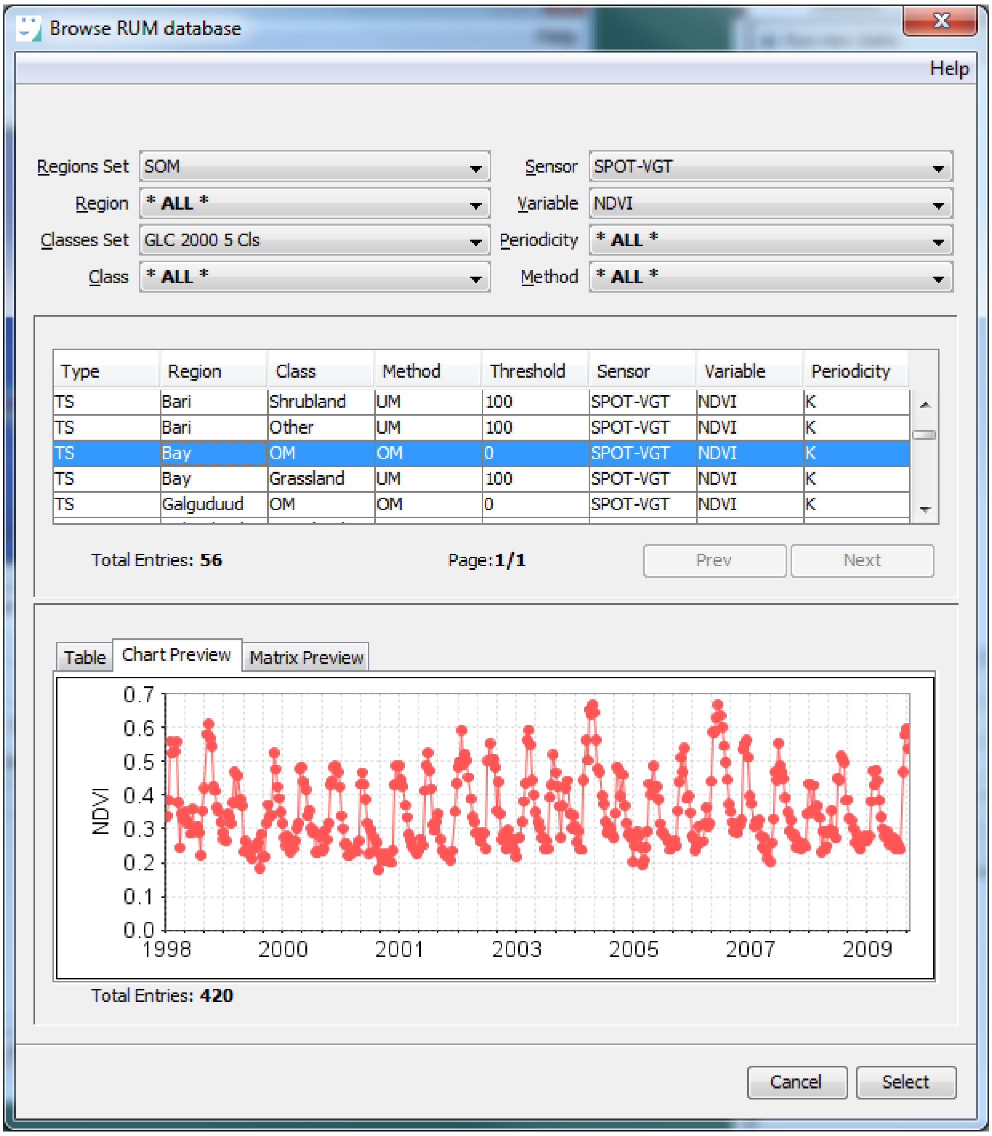
Values of the selected dataset

RUM Browser

In the top part of the browser UI a filter can be defined according to most relevant dataset attributes.

In the middle part, the attributes of the available datasets, satisfying the filter condition are displayed.

In case one of these datasets is selected, its values are displayed at the bottom part of the UI. Besides the values table, they can also be previewed in a chart.



RUM Browser - dataset preview

Once a dataset is selected,

- via the New Chart action button, it can be used to start a new instance of the RUM Chart utility, showing this dataset, or
- via the Add to Chart action button, it can be send to the last selected existing RUM Chart utility, if any, to be added as an additional dataset in this Chart.

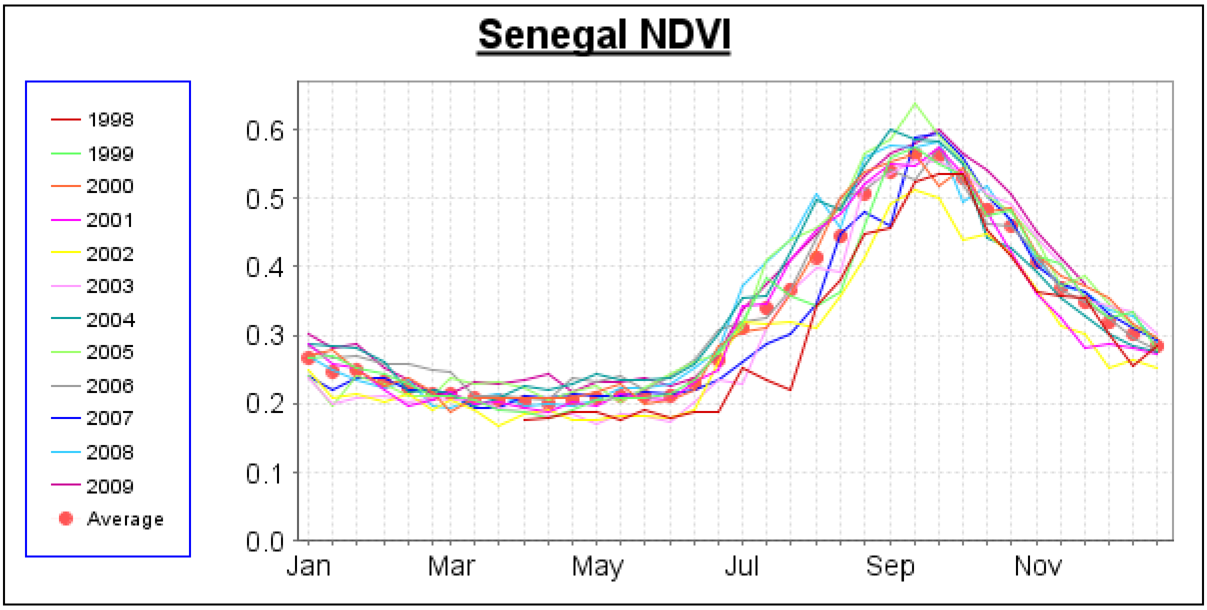
4.2. RUM Charts

4.2.1. Introduction

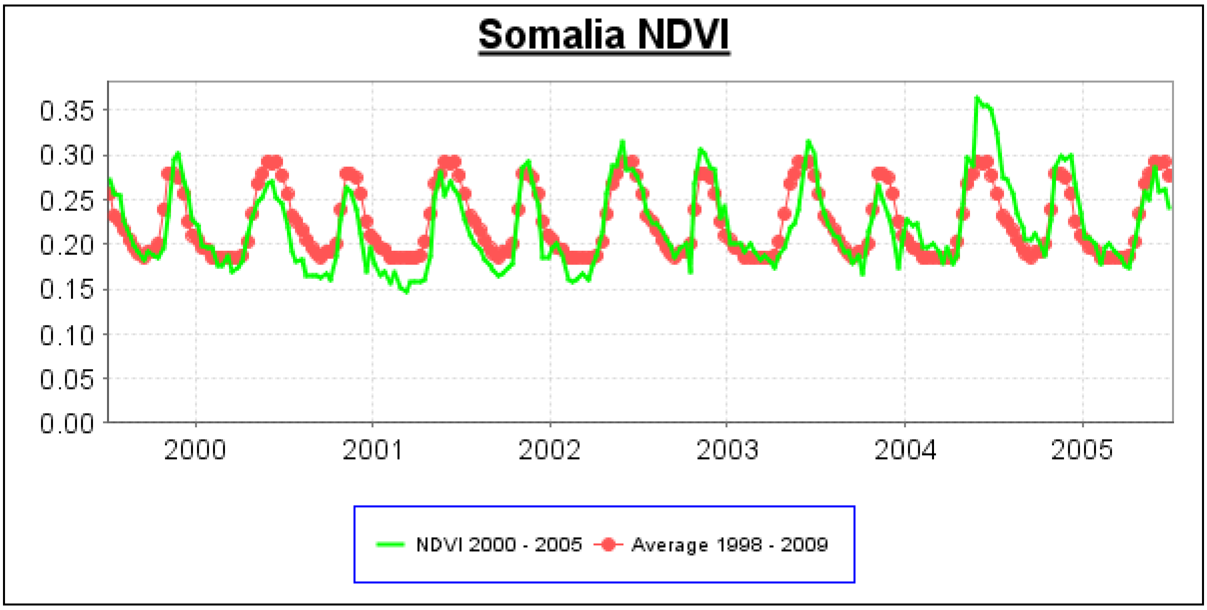
The RUM Chart utility enables the graphical visualisation of the RUM data collected in the RUM database.

A basic RUM Chart is an X-Y graph, plotting the datasets values (means) against the Y-axis, the datasets times (periods) against the X-axis. Two distinct X-axis modes are available:

- continuous X-axis: the (value, time) points are plotted as a continues series;
- annual X-axis: the (maximum) X-axis interval represents one single year. The dataset is divided into multiple dataset series, each containing the points for a single one year interval. The (value, period) points are plotted relative to the selected start period of the X-axis.

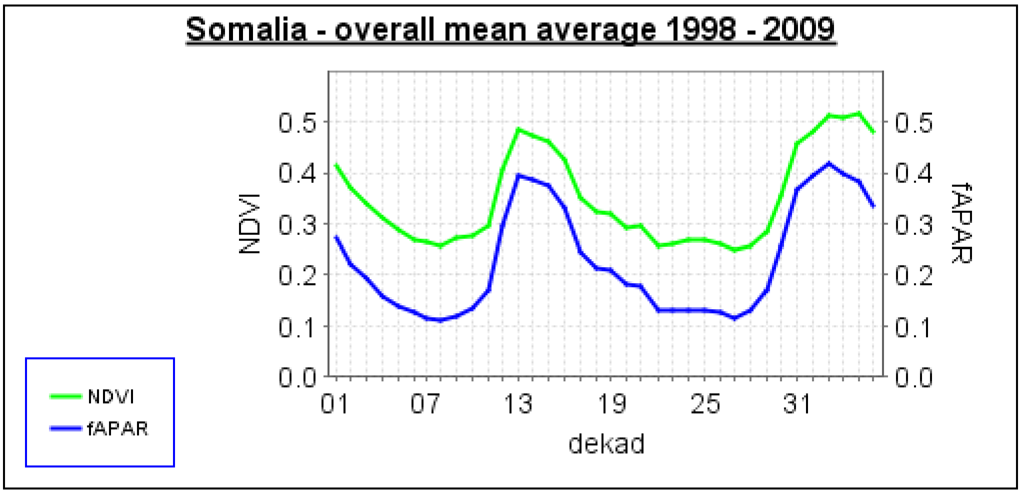


RUM Chart annual X axis example



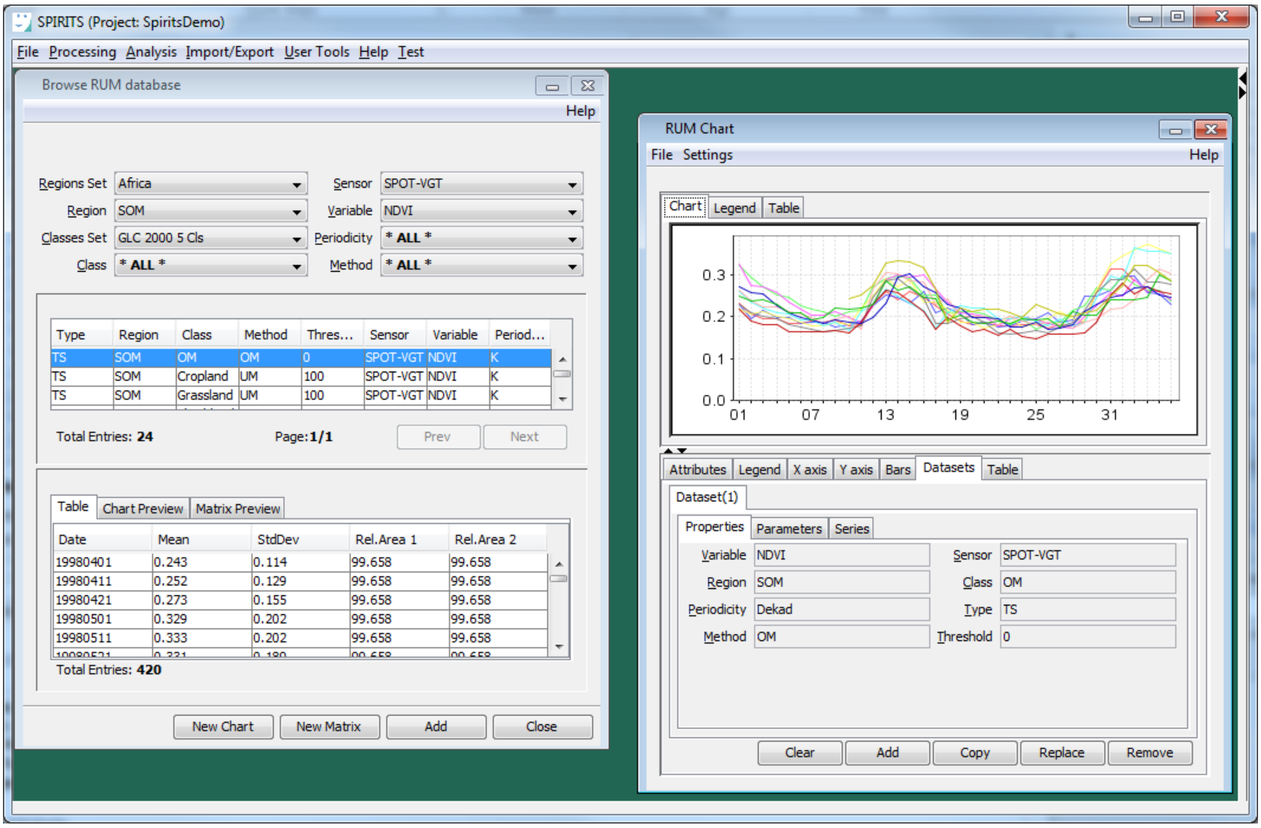
RUM Chart continuous X axis example

RUM Charts can contain multiple datasets. In case these datasets belong to different variables additional Y-axis appear.



RUM Chart multiple variables example

Datasets can be sent from the database browser to a Charts Form, or can be selected from a RUM Chart Form directly.



Browser :	**New Chart**	opens a new RUM Chart form containing the selected dataset;
Browser:	**Add**	adds the selected dataset to the last active RUM Chart form;
Chart:	**Add**	opens a Browser to select a dataset to be added to the Chart.

4.2.2. Datasets

The datasets can be plotted as they are, or they can be processed by means of smoothing, applying operations, restriction of the date interval or cumulating.

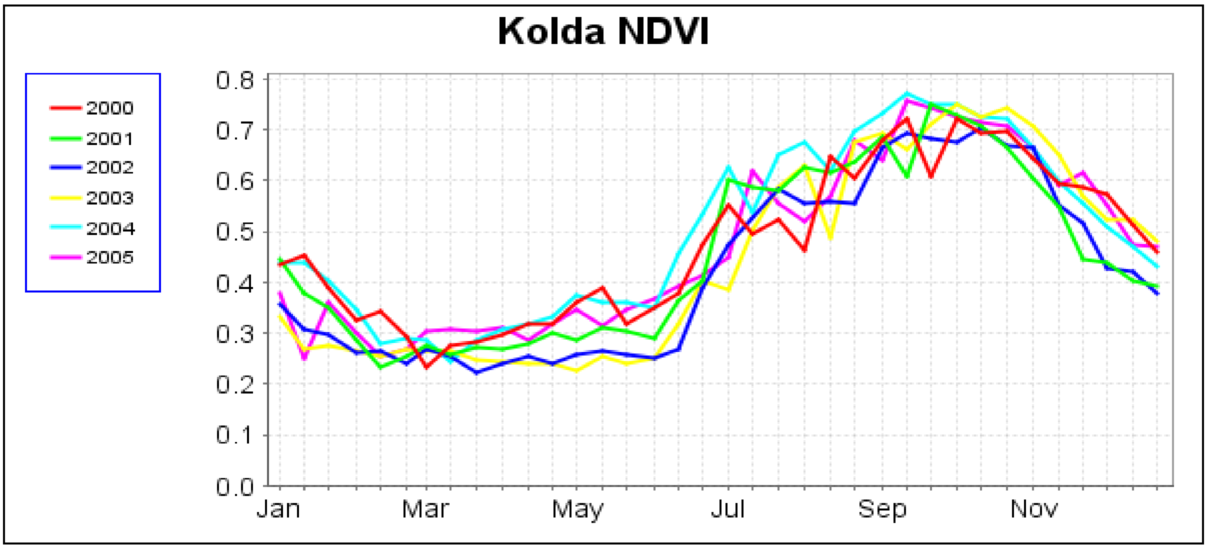
4.2.2.1. Smoothing

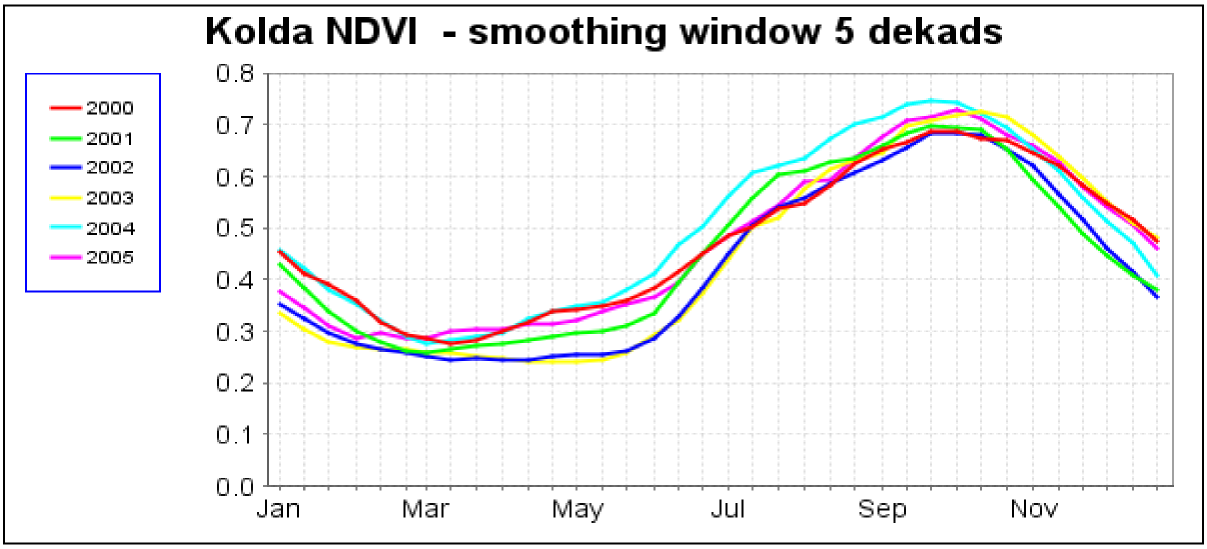
Optionally, datasets can be smoothed. A smoothing window type and its size can be specified. The smoothed value for a period P will then be the average of the values in the window. Depending on the window type these values will be:

type Left	{ value(P-size+1) ... value(P) }	
type Centre	{ value(P-size/2) ... value (P+size/2) }	(size must be odd)
type Right	{ value(P) ... value(P+size-1) }	

Smoothing example

- original dataset values: X(period)
- smoothed dataset values: SX(period)
- smoothing window type: centred
- smoothing window size: 5
- => SX(period) = { X(period-2) + X(period-1) + X(period) + X(period+1) + X(period+2) } / 5

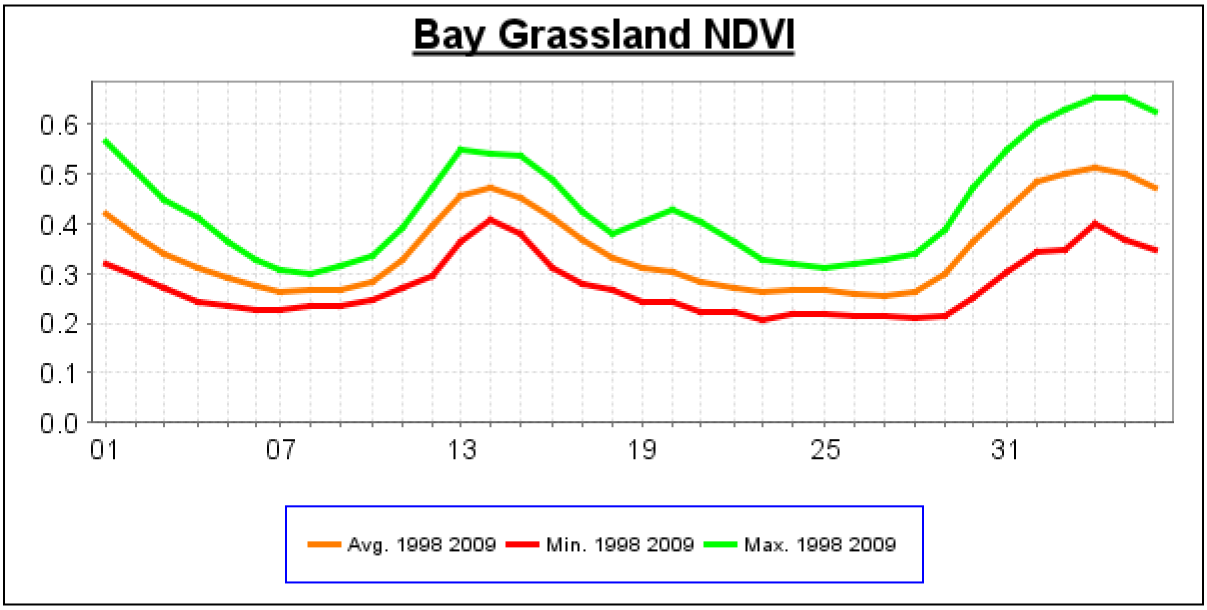




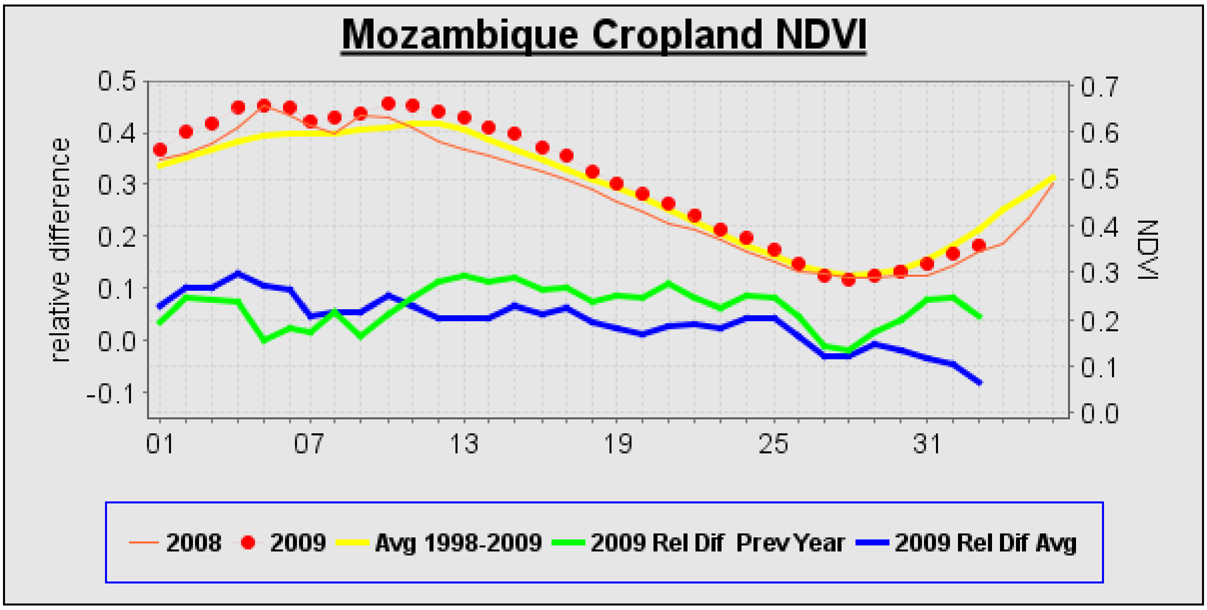
4.2.2.2. Operations

Operations can be performed on datasets. Following operations are available:

Average (Avg)	(P)	= Sum(X(y,P)) / Count(X(y,P))
Minimum (Min)	(P)	= Min(X(y,P))
Maximum (Max)	(P)	= Max(X(y,P))
Standard Deviation (Std)	(P)	= Square Root{ Sum(X(y,P) - Average(P))² / Count(X(y,P) }
		over all years y
Historical Average (Hist.Avg)	(Y,P)	= Sum(X(y,P)) / Count(X(y,P))
Historical Minimum (Hist.Min)	(Y,P)	= Min(X(y,P))
Historical Maximum (Hist.Max)	(Y,P)	= Max(X(y,P))
Historical Standard Deviation (Hist.Std)	(Y,P)	= Square Root{ Sum(X(y,P) - Hist.Avg(P))² / Count(X(y,P) }
		over all years y, except y=Y
Absolute Difference Previous Period (ADpp)	(Y,P)	= X(Y,P) - X(Y,P-1)
Absolute Difference Previous Year (ADpy)	(Y,P)	= X(Y,P) - X(Y-1,P)
Absolute Difference Average (ADav)	(Y,P)	= X(Y,P) - Avg(P)
Absolute Difference Historical Average (ADha)	(Y,P)	= X(Y,P) - Hist.Avg(Y,P)
Relative Difference previous period (RDpp)	(Y,P)	= { X(Y,P) - X(Y,P-1) } / X(Y,P-1)
Relative Difference previous year (RDpy)	(Y,P)	= { X(Y,P) - X(Y-1,P) } / X(Y-1,P)
Relative Difference average (RDav)	(Y,P)	= { X(Y,P) - Avg(P) } / Avg(P)
Relative Difference historical average (RDha)	(Y,P)	= { X(Y,P) - Hist.Avg(Y,P) } / Hist. Avg(Y,P)
Z-score (Z)	(Y,P)	= { X(Y,P) - Avg(P) } / Std(P)
Historical Z-score (Zh)	(Y,P)	= { X(Y,P) - Hist.Avg(Y,P) } Hist.Std(Y,P)



RUM Chart Minimum, Maximum, Average operations example

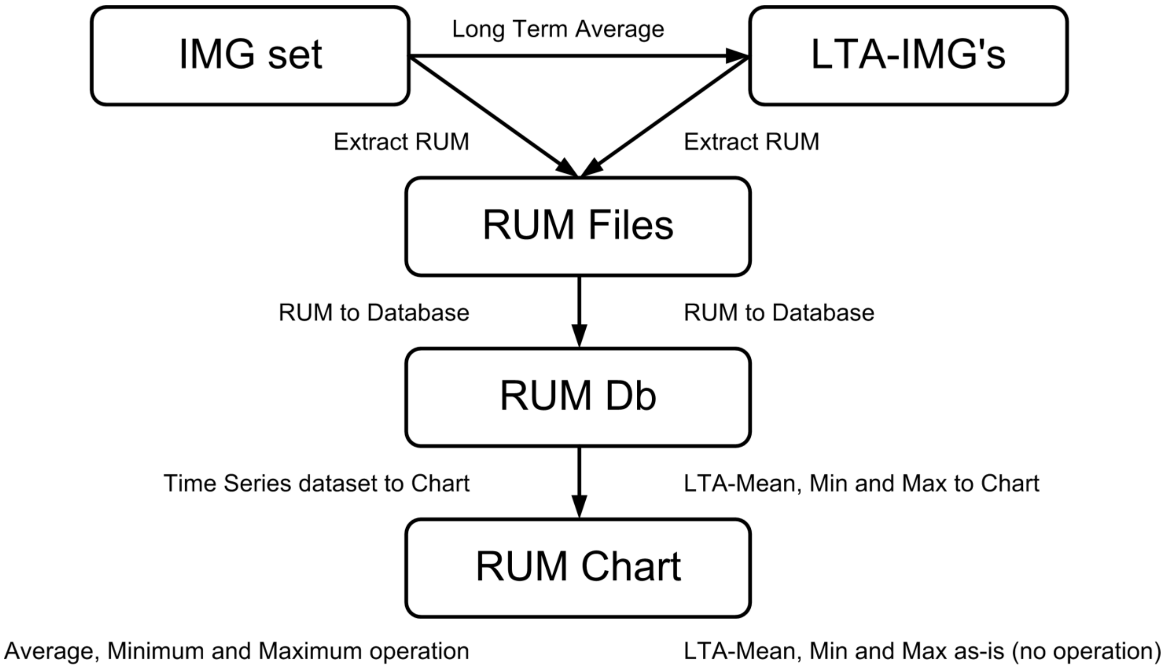


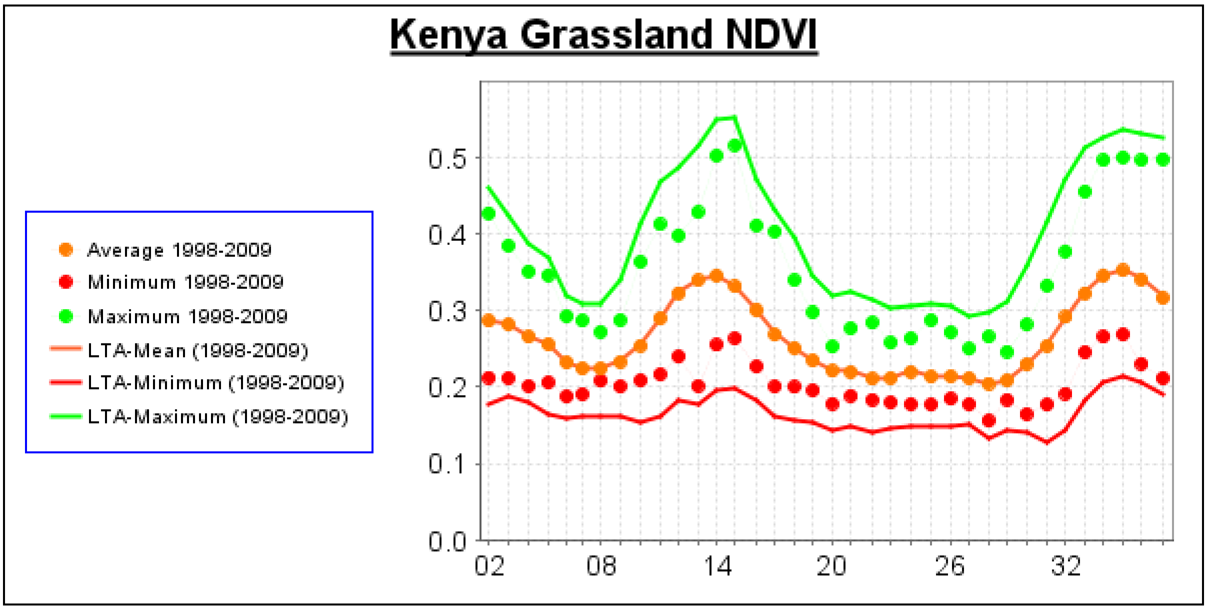
RUM Chart Relative Differences previous year and average operations example

Operations example: Long Term Averages vs Average, Minimum and Maximum operations

comparison between:

- the RUM values extracted from the LTA (Long Term Average) Mean, Minimum and Maximum IMGs (created by the Long Term Average tool);
- the values obtained by applying the Average, Minimum and Maximum chart operations on RUM values extracted from the original IMGs (as used by the LTA tool).





In the LTA case, LTA Minimum and Maximum IMGs will initially keep the individual pixel values, prior to averaging over a region/class in the RUM extraction process. Thus, for the RUM values of a series of IMGs and those of their LTA's:

LTA-Min <= Minimum <= LTA-Mean == Average <= Maximum <= LTA-Max

or:

RUM[minimum(X*p*)] <= minimum[RUM(X*p*)]

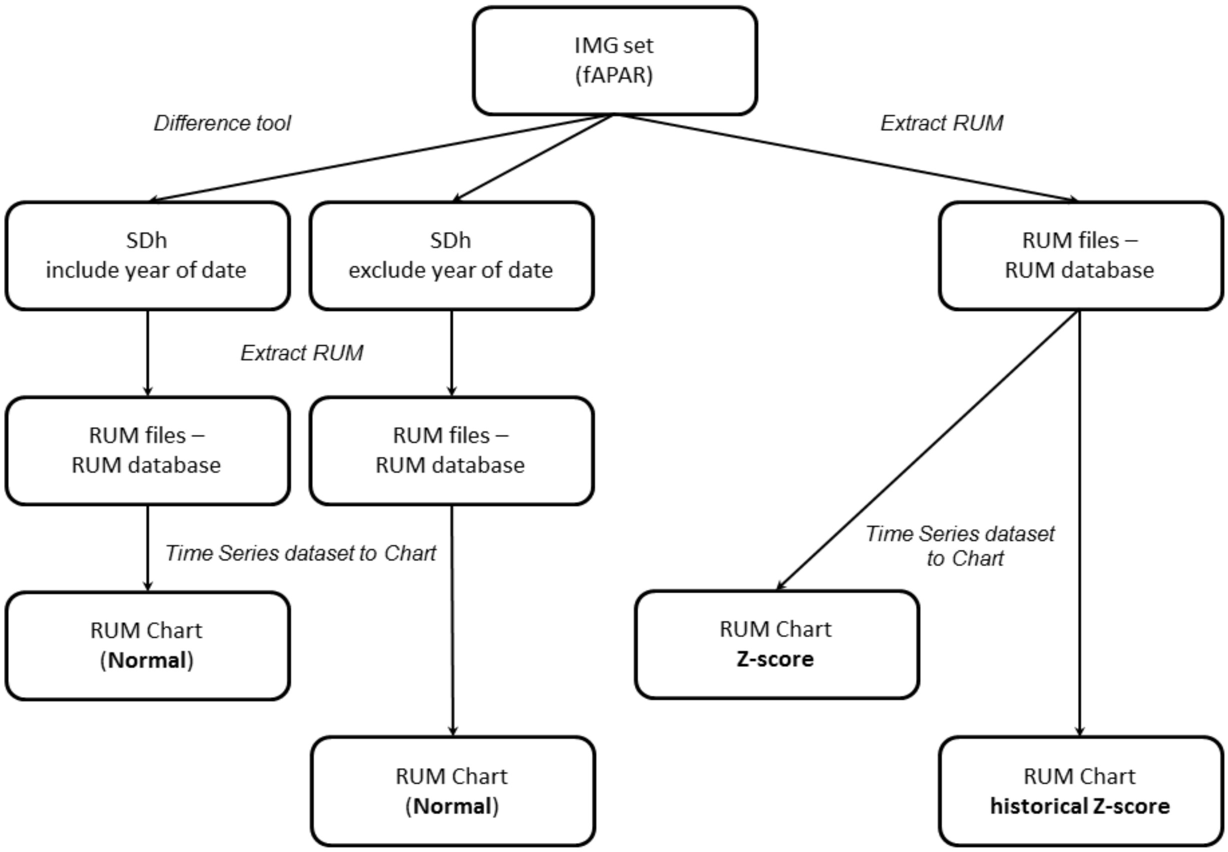
RUM[mean(X*p*)] == mean[RUM(X*p*)]

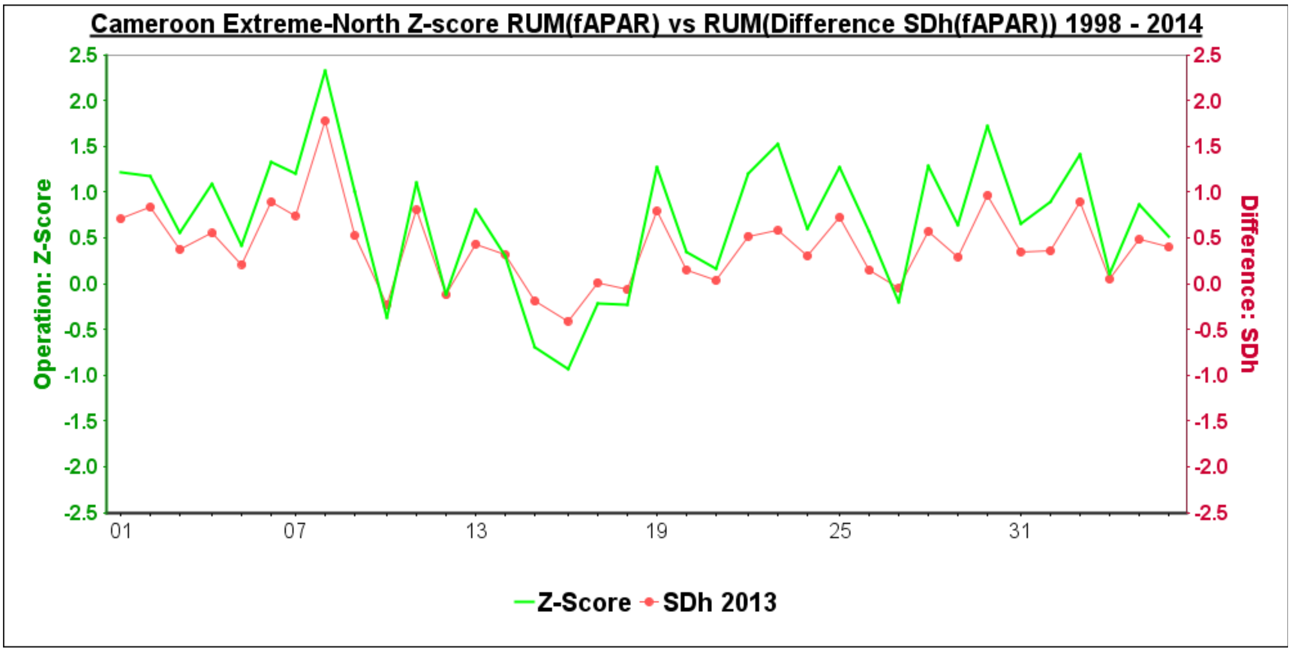
RUM[maximum(X*p*)] => maximum[RUM(X*p*)]

for period *p*

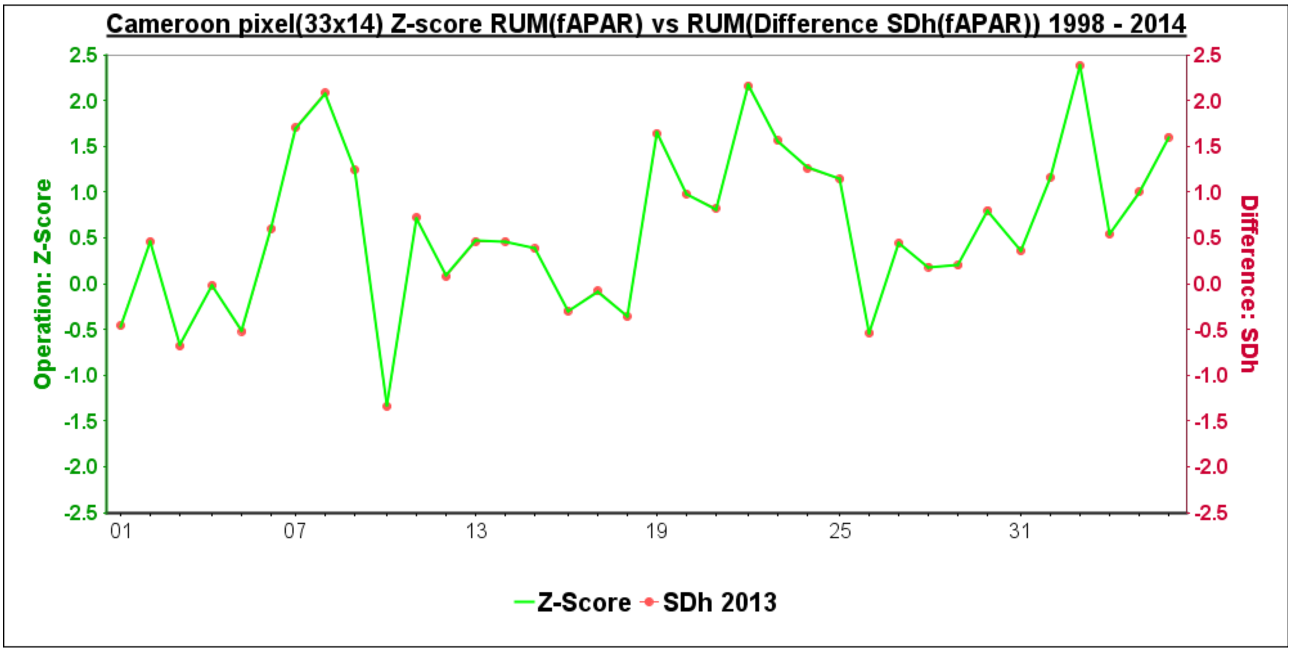
Operations example: Z-score operation vs Standardized Difference (SDh)

- comparison between RUM values extracted from Standardized Difference images (Difference tool), and
- the values obtained by applying the Z-Score chart operations on RUM values, extracted from the original input IMGs as used by the Difference tool. In this example VGT fAPAR IMGs for Cameroon between 1998 and 2014 are used.

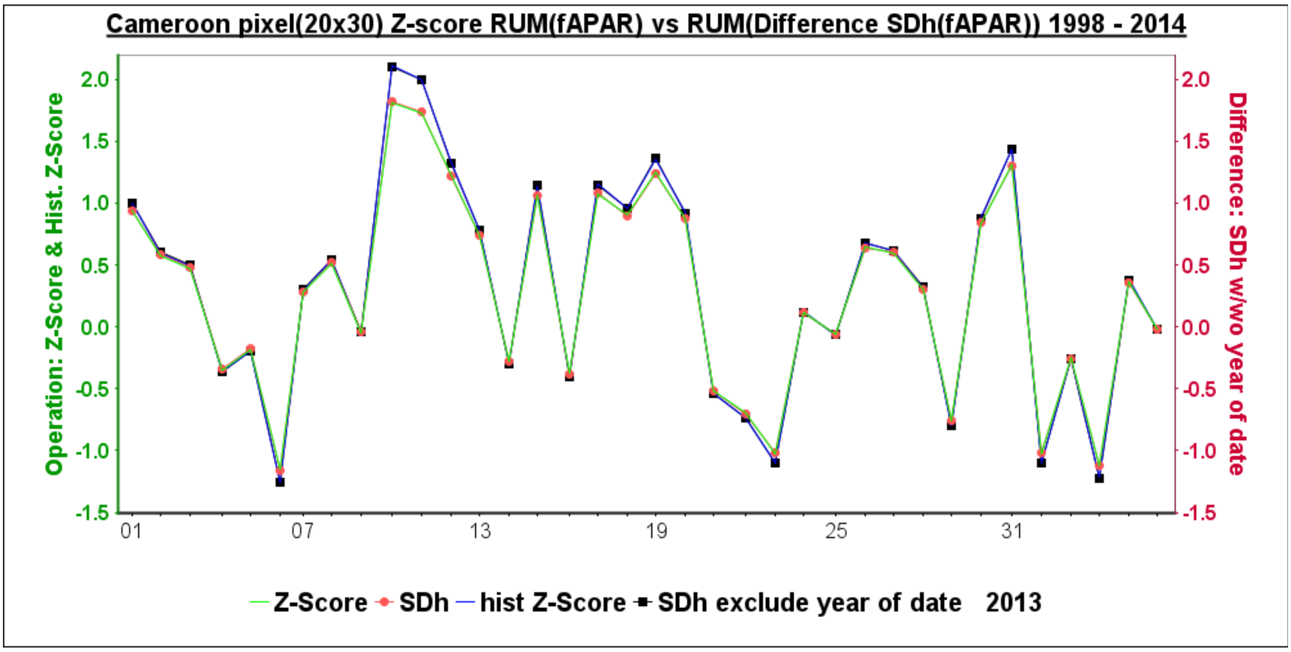




SDh and the RUM Chart Z-scores do show similar trends but are different due to the regional averaging calculation prior to calculating the Z-score.



when restricting the regions to discrete IMG pixels, the results become quasi (rounding) identical.

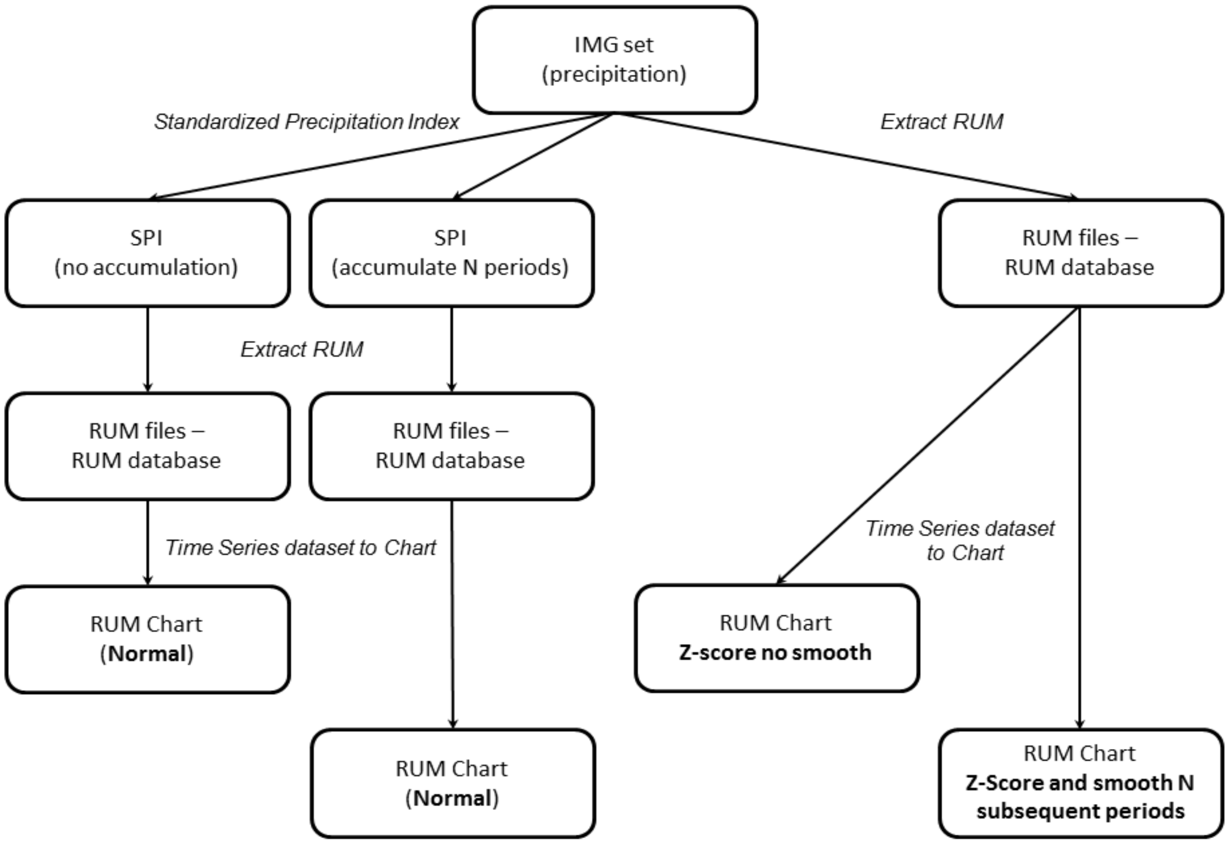


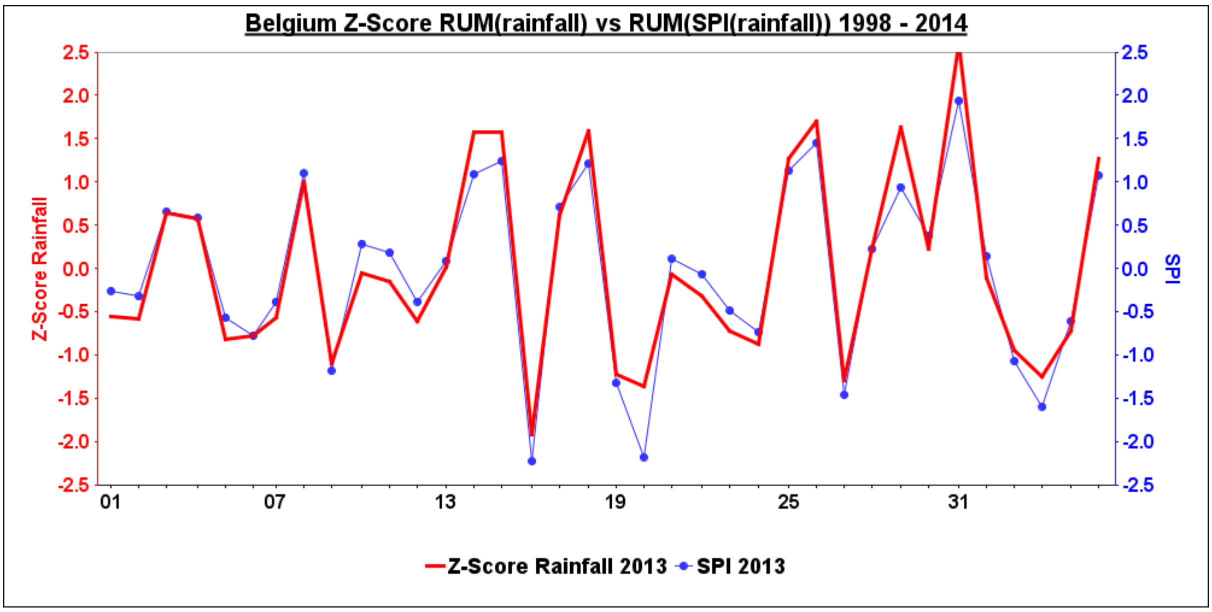
the 'normal' Z score correlates with SDh calculated including the 'year of date'

the 'historical' Z score correlates with SDh calculated excluding the 'year of date'

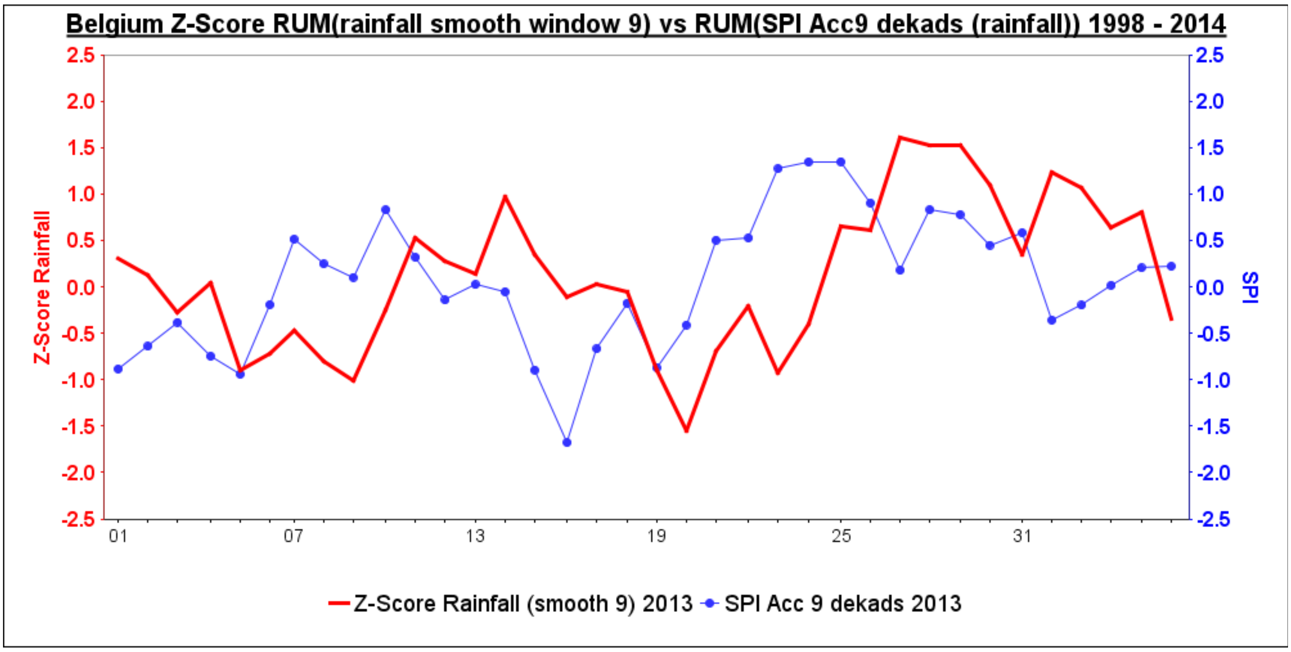
Operations example: Z-score operation vs Standardized Precipitation Index

- comparison between RUM values extracted from Standardized Precipitation Indices with and without accumulations (see Standardized Precipitation Index tool), and
- the values obtained by applying the Z-Score chart operations on non-smoothed and smoothed-over-subsequent periods RUM values, extracted from the original (precipitation) IMGs (as used by the Standardized Precipitation Index tool).

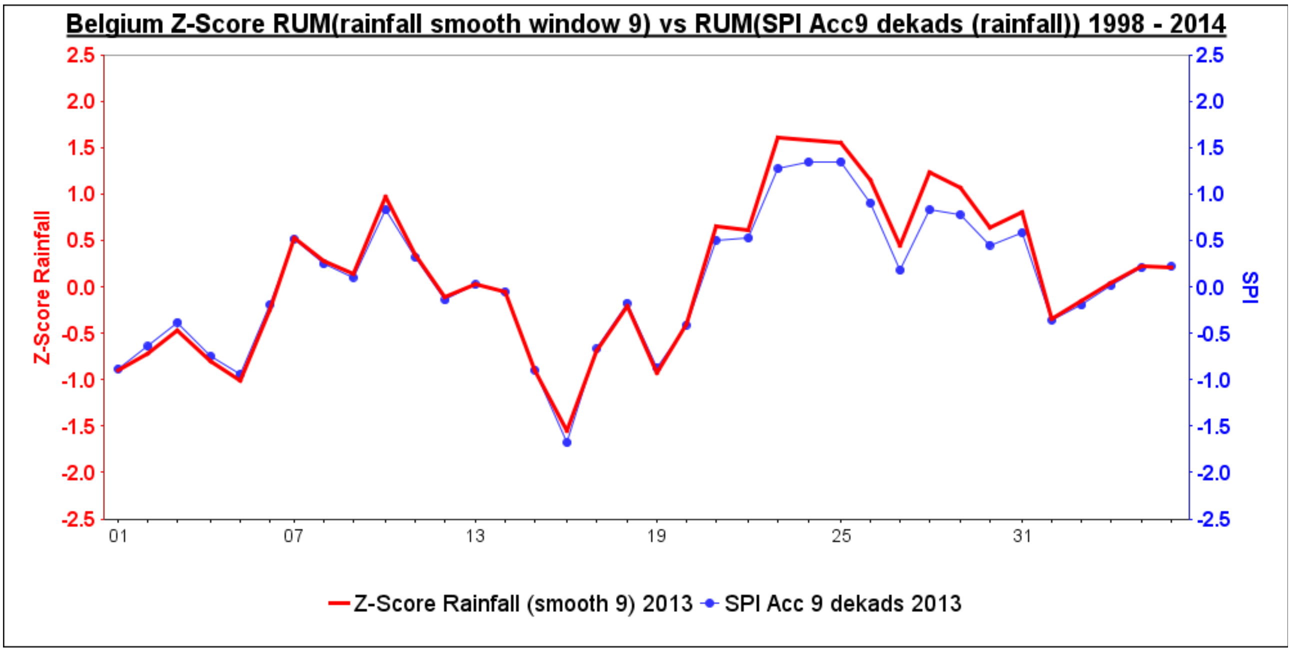




SPI without accumulation correlates directly with the RUM Chart Z-scores.



RUM-values smoothed from the centre (default) over 9 periods
=> show the same trend as SPI with accumulation length 9, but shifted

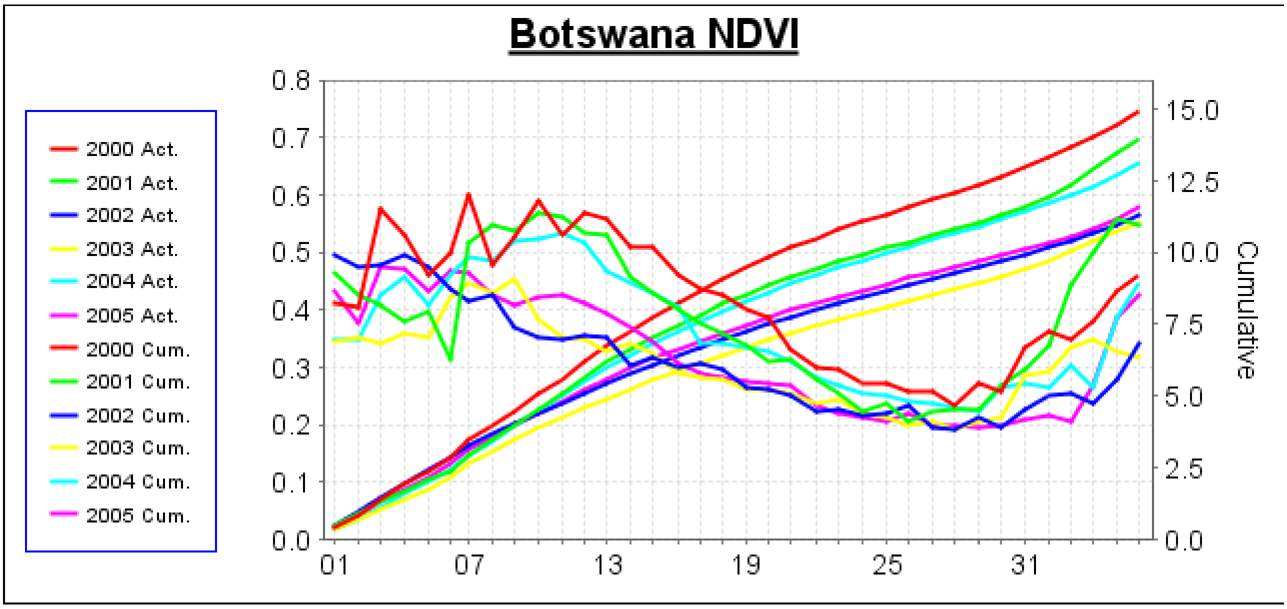


RUM-values smoothed using subsequent 9 periods => give the expected correlation

Since the Standardized Precipitation, Index calculated with an accumulation length of N periods, considers the mean (or cumulative) value from N periods, starting from period P up to (P + N - 1), a RUM Chart smoothing window of size N, using subsequent periods, has to be applied on the original (precipitation) dataset, to obtain comparable results via the RUM Chart Z-score operation.

4.2.2.3. Mode

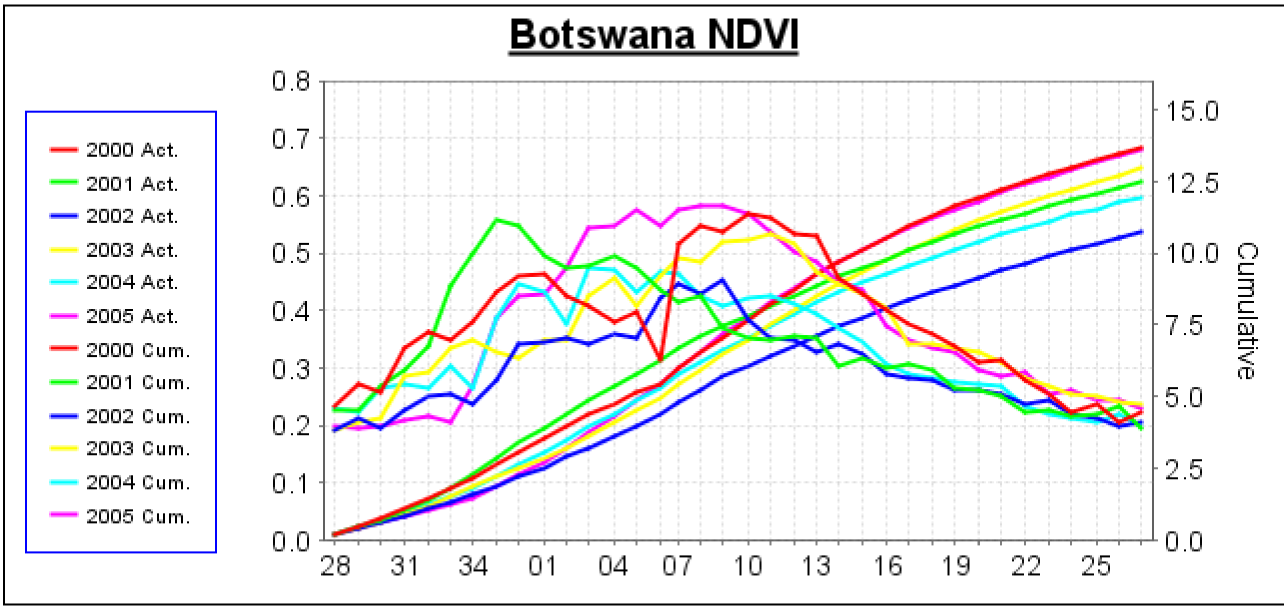
Dataset values can be show as they are ("Actual" mode) or they can be shown cumulative ("Cumulative" mode). In case of an annual X-axis, the sum is reset at the first period of the X-axis.



RUM Chart actual and cumulative mode example

In case of an annual X-axis, the sum is reset at the first period of the X-axis.

As a consequence, the cumulative values calculated are dependent on the X-axis type and range (the period selected to be the start of the X-axis).



RUM Chart actual and cumulative mode - shifed X axis - example

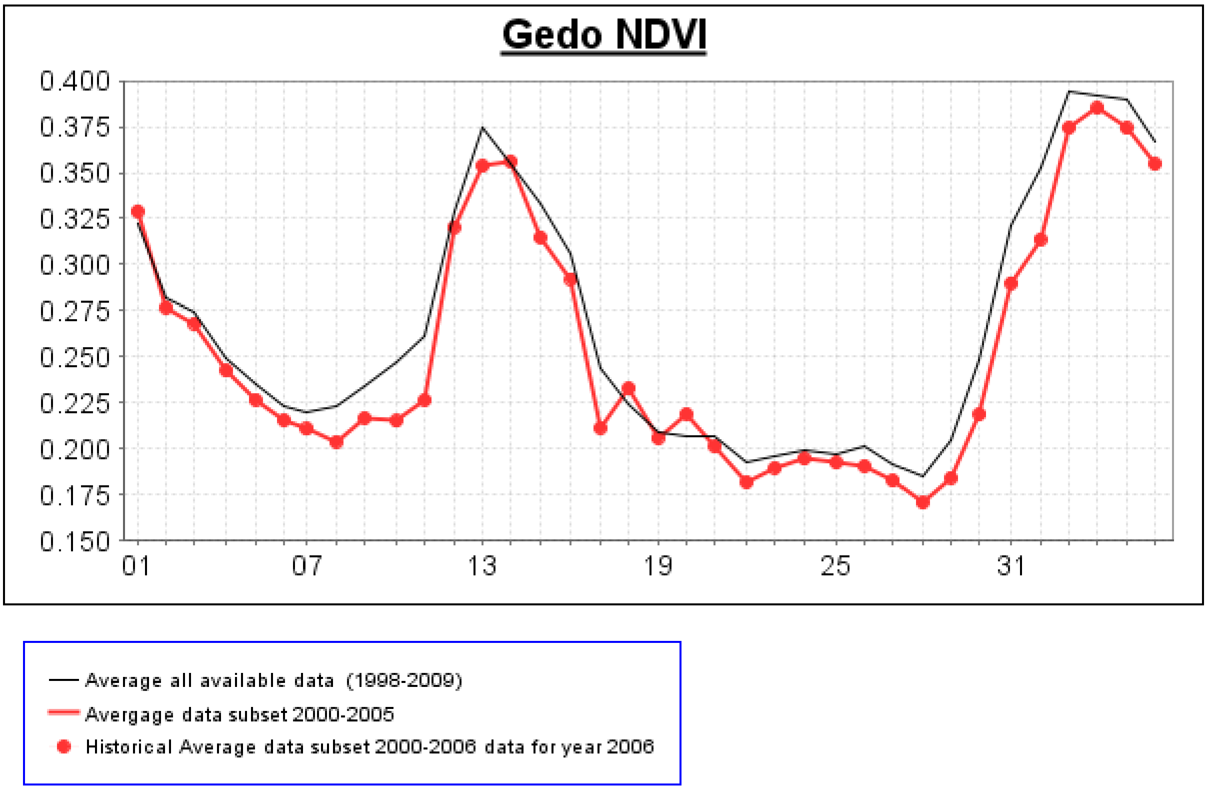
4.2.2.4. Date range

Optionally, the data used in a chart dataset, can be restricted to a subset of the dataset available in the database. This by specifying upper and/or lower date limit(s) for the data to be used in the chart. Such restrictions mainly affect operations performed on the dataset, such as averaging.

Date range example

comparison between

- the (overall) average from the complete (unrestricted) dataset;
- the (overall) average from a subset of the dataset from 2000 - 2005 (20000101-20051231)
- the historical average for 2006 from a subset of the dataset from 2000 - 2006 (20000101-20061231)



Since

the historical average for 2006 from the 2000-2006 subset is defined as:

Sum(X(y,P)) /Count(X(y,P)) over 2000 - 2006 except 2006

the (overall) average from the 2000-2005 subset is defined as:

Sum(X(y,P)) /Count(X(y,P)) over 2000 - 2005

these cases give identical results.

4.2.3. RUM Charts Form and Panels

4.2.3.1. Views

Chart view

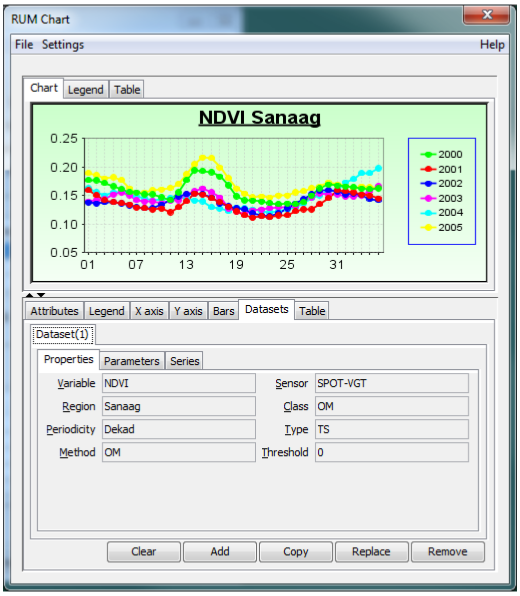
The main view for RUM Charts is the Chart view panel, which contains the graphical representation of the chart data. By right clicking the panel, its contents can be copied or saved as a PNG image. A PNG file of a RUM Chart can be also created via the Export PNG entry in the File menu.

Legend view

The Legend view panel contains only the charts legend. This can be used to obtain the legend information separately in case it would occupy too much space on the chart itself. By right clicking the panel, its contents can be copied or saved as a PNG image.

Table view

The Table view panel contains the chart data in table format. By right clicking the panel, its contents can be copied or saved as an ASCII file.



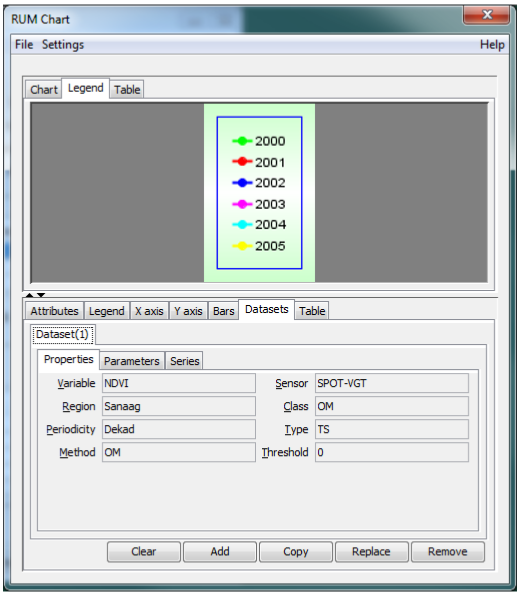


Chart view Legend view

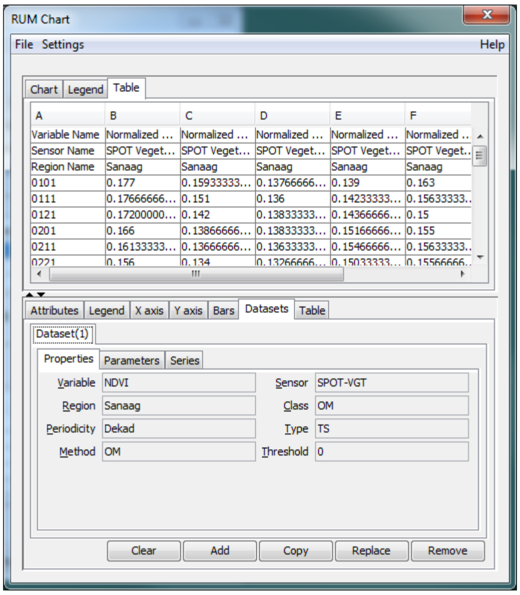


Table view

4.2.3.2. Attributes

In the attributes panel, the chart title, background, gridlines and size can be specified.

Chart title

The font, the font size, the font colour and the font type of the chart title can be specified. It can be positioned on fixed locations (Top-Left, ...Right-Bottom, default is Top-Center).

The title content can be a parameterized string: a mixture of constant text and parameters ("%0", "%1,...).

The available parameters can be inspected via the Title Parameters button. Their actual values originate from the properties of the first dataset series of the first dataset in the chart.

Chart background

The background of the chart can be specified. This background can be a single solid colour, or a two-colour gradient chosen from a fixed set of gradient types (horizontal, vertical, diagonal, ...).

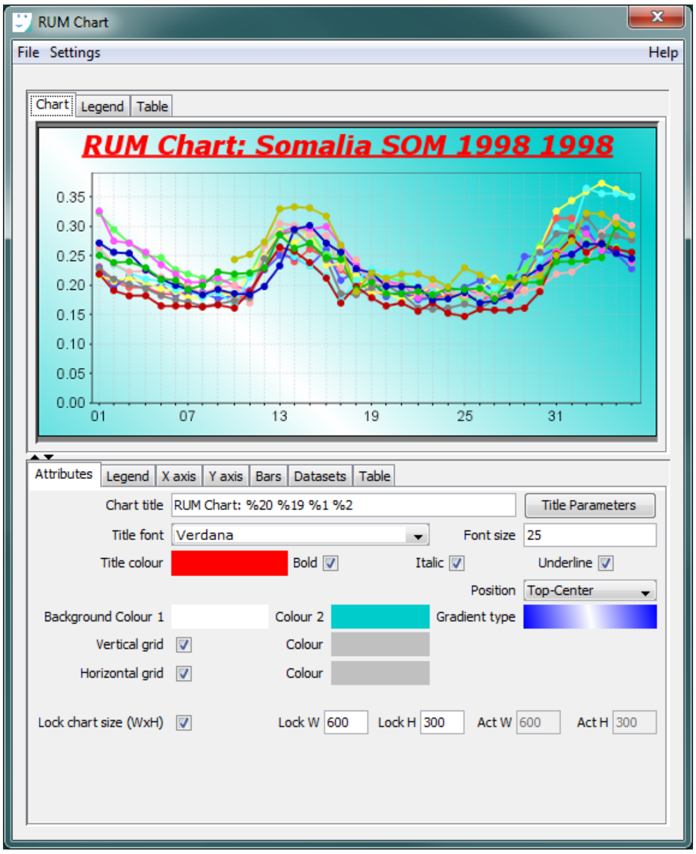
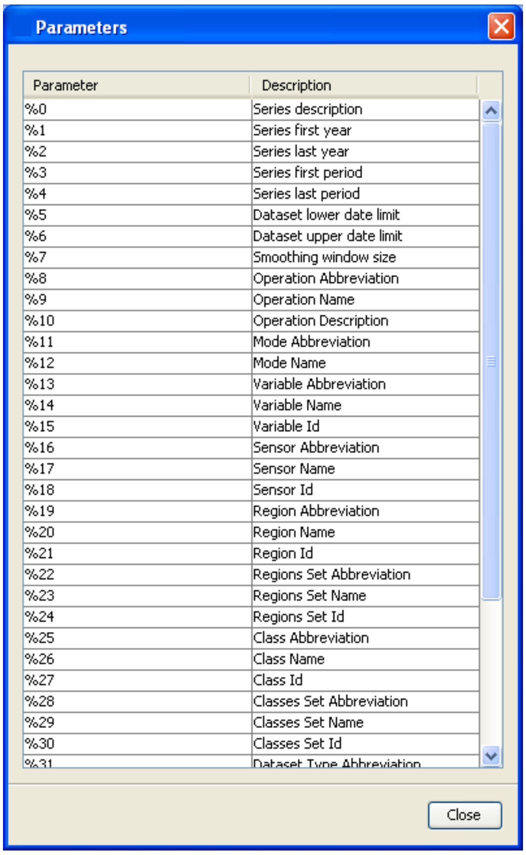
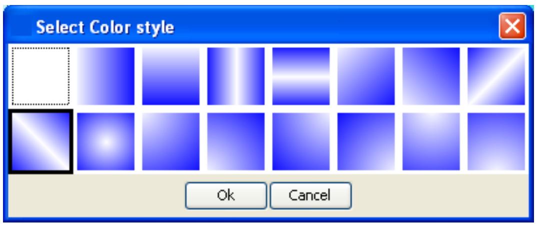


Chart title example



Parameters panel



Background Gradient panel

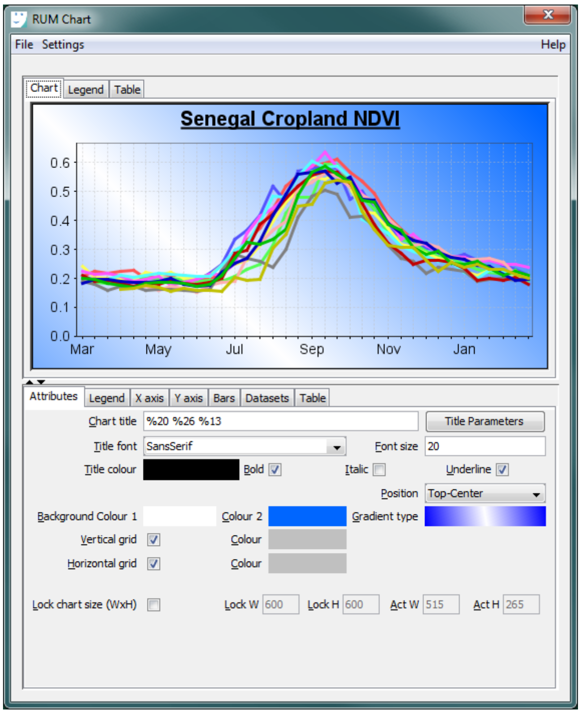
Gridlines

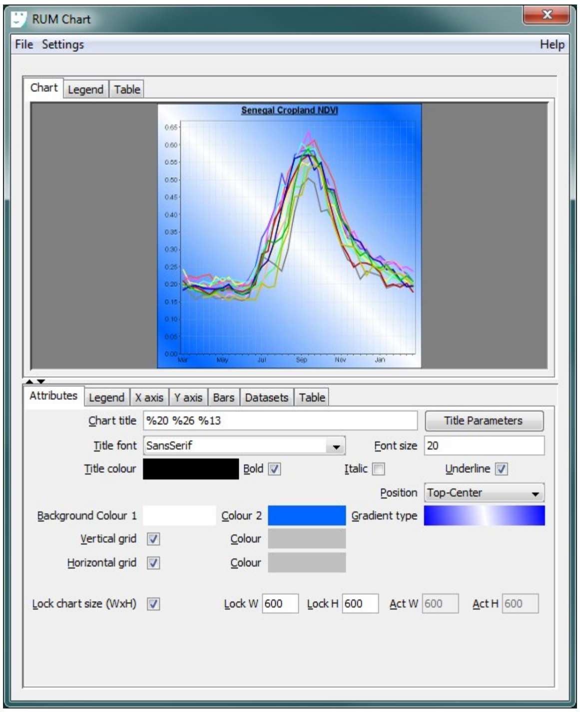
Horizontal and vertical gridlines can shown or hidden. Their colours can be specified.

Chart size

Two distinct working modes concerning the size of the actual chart, and of the PNG image that can be exported, are available:

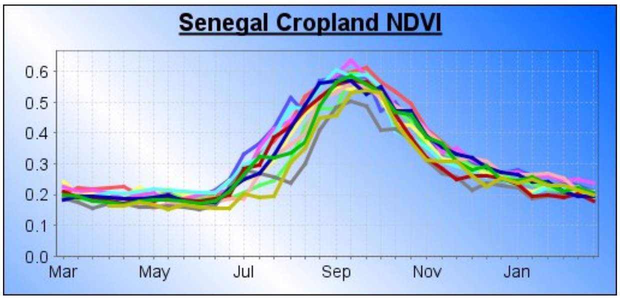
- free running mode: In free running mode, the screen is put to its optimal use. The chart is stretched to the available size of its panel, and this size will also be the size of the PNG image when exported.
- locked size mode: In locked size mode, the 'real' chart size is specified by the user. This means the user controls the exact size (and thus the quality) of the PNG image when exported. Since the RUM Chart utility strives for WYSIWYG, the image shown on the screen will have to be re-sampled. In most cases this means that the on-screen quality will be worse than that of the actual PNG image when exported.

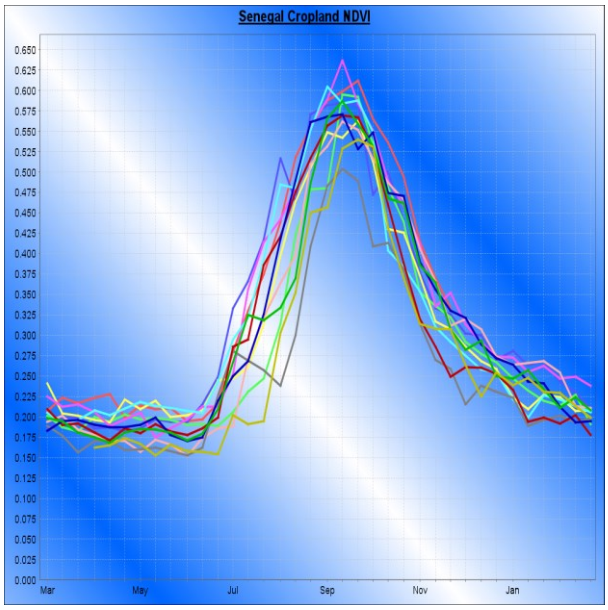




example free running mode

locked size mode (600 x 600)





exported png: size is 515 x 265

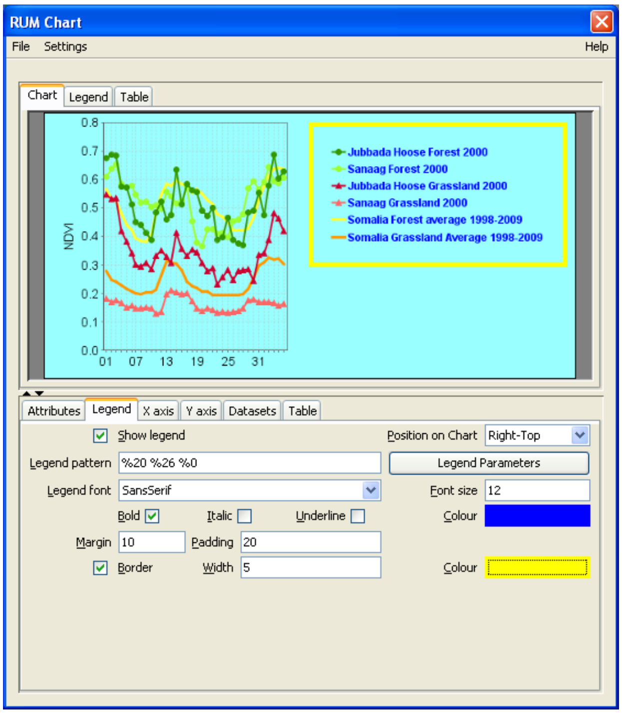
exported png: size is 600 x 600

4.2.3.3. Legend

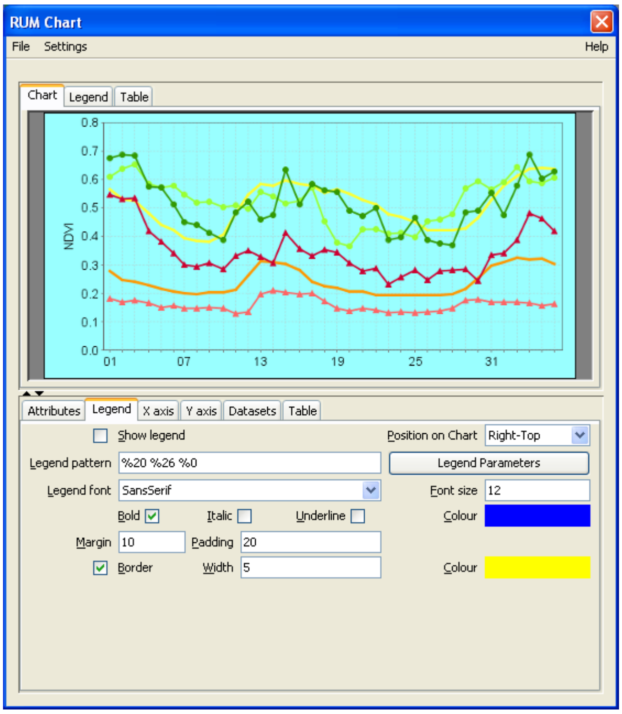
A legend can be shown on the chart, but is also available separately in the Legend View panel.

Legend look and feel

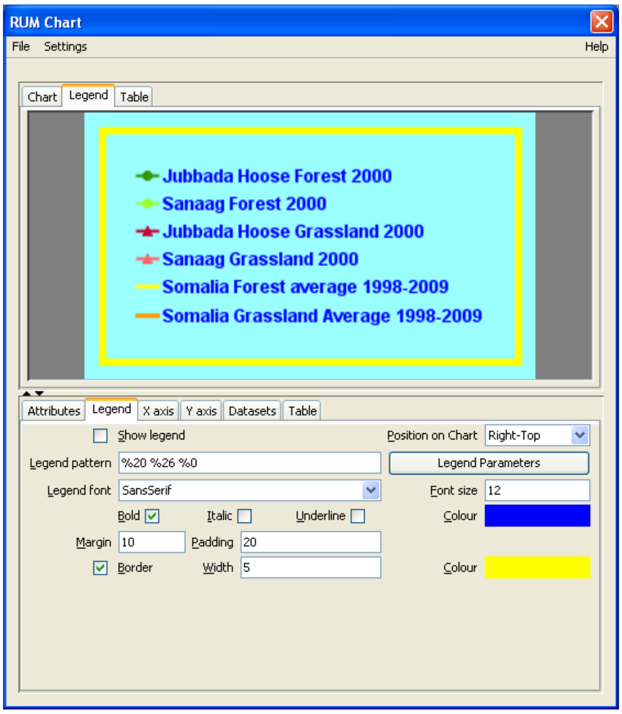
The font, the font size, the font colour and the font type for the legend contents can be specified. The legend contents can optionally be surrounded by a border. The border width and colour can be specified. Furthermore the free space outside the legend contents (Padding), and the free space outside the legend border (Margin) can be specified.



example: legend on chart



no legend on chart



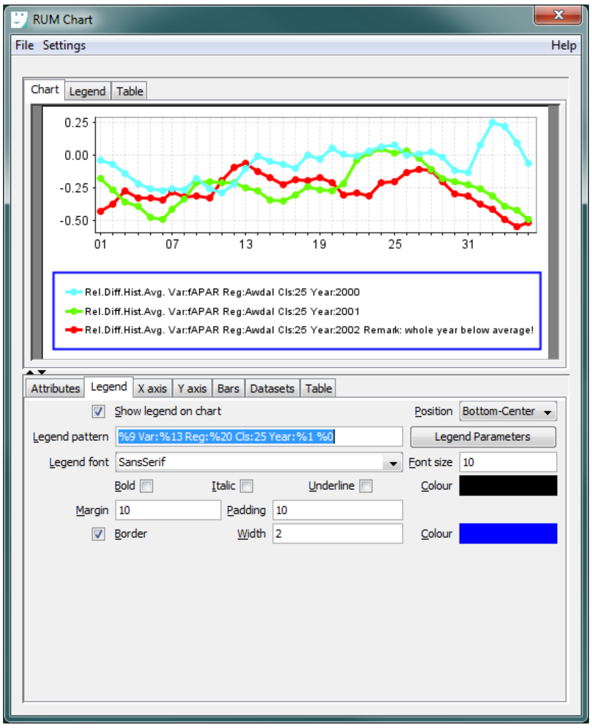
separate legend in Legend view panel

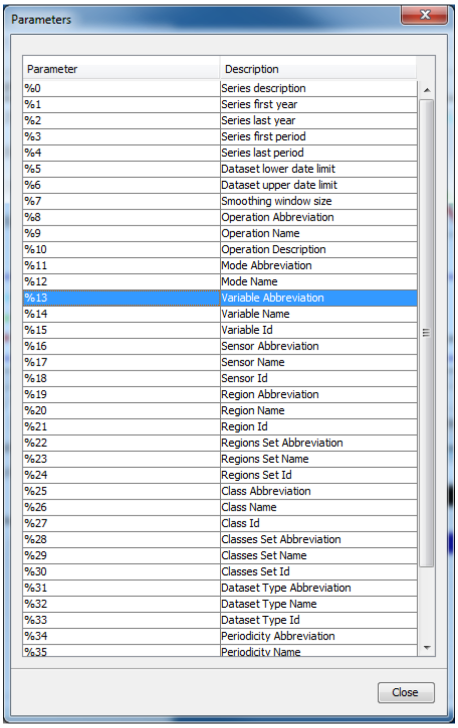
Legend contents

The legend contents can be specified by means of a parameterized string, a combination of:

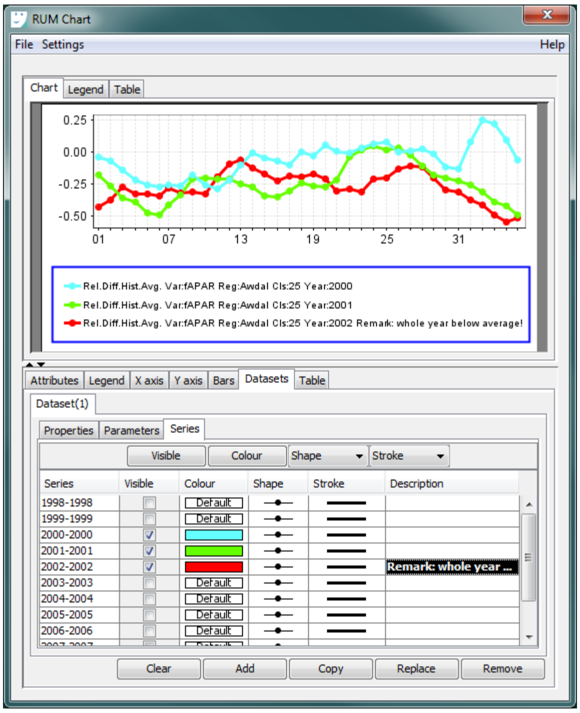
- constant text
- parameters (%0 - %...) representing dataset info (e.g. variable), series info (e.g. first year), operation info and a user defined series description

The available parameters can be viewed via the Legend Parameters button. The user defined series description can be filled out for each dataset series of each dataset in the chart, at the Dataset panels.





legend parameterized string example available parameters panel



user defined series description

4.2.3.4. X-axis

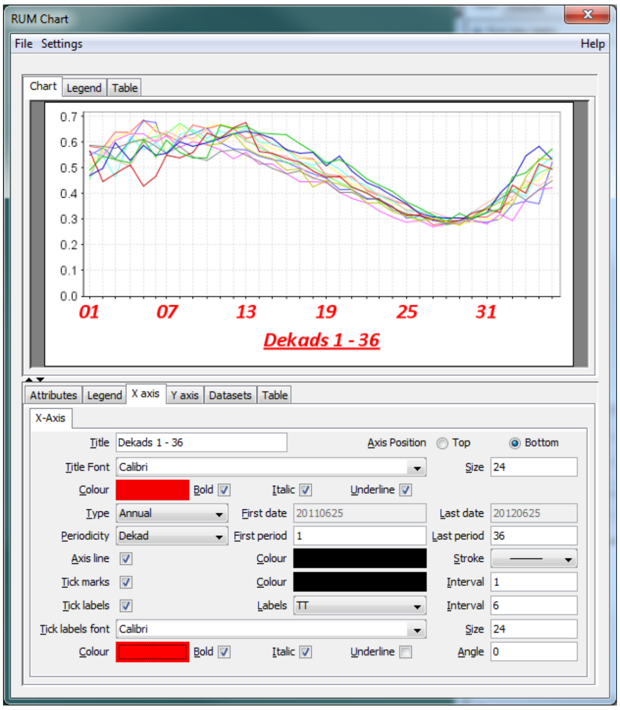
Charts contain a single X-axis. It can be positioned at the top or at the bottom of the chart.

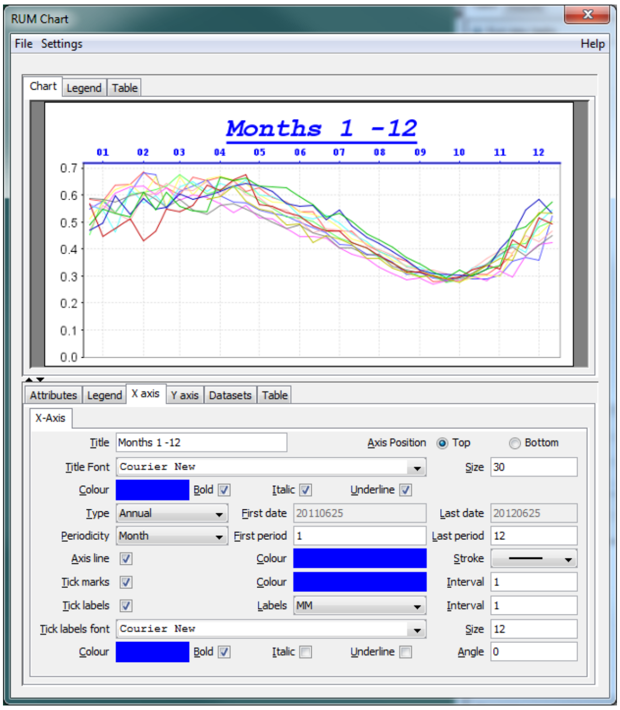
X-axis title

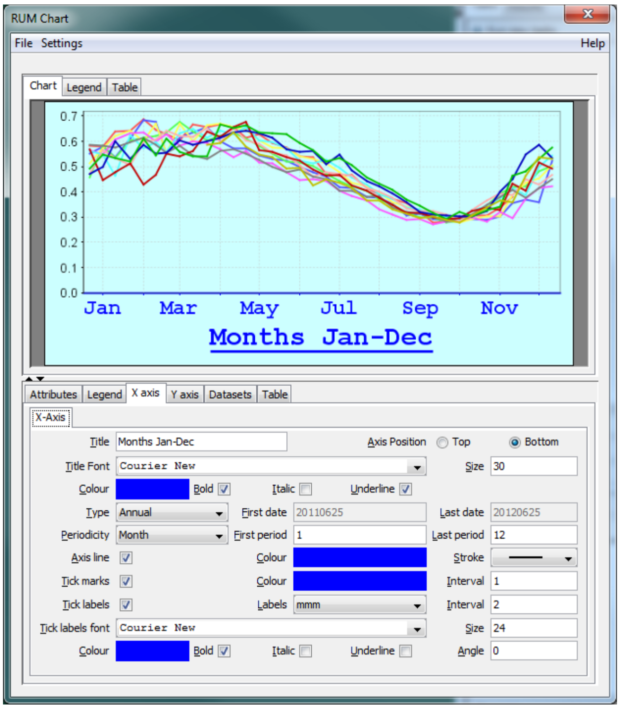
The font, the font size, the font colour and the font type of the X-axis title can be specified.

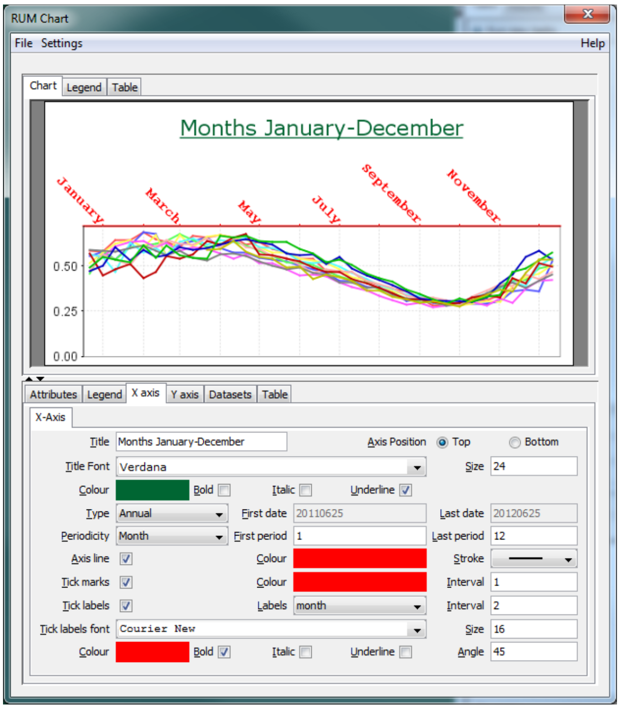
X-axis periodicity, ticks and labels

The X-axis tick marks and tick labels intervals can be specified. These will be expressed in numbers of periods of the selected periodicity (Day, Dekad, Month or Year). The font, the font size, the font colour and the font type of the X-axis labels can be specified.









The format of the labels can be chosen from a fixed set of format types. Some format type examples in table below.

code	content	format and range
none	blank	
DDD	day in year	000 - 365
TT	dekad in year	00 - 36
mmm	month	Jan - Dec
month	month	January - December
MM	month	00 - 12
mmm DD	month, day in month	Jan 01 - Dec 31
month DD	month, day in month	January 01 - December 31
MM DD	month, day in month	00 01 - 12 31
MM:DD	month, day in month	00:01 - 12:31
MM-DD	month, day in month	00-01 - 12-31
YY	year	50 - 49
YYYY	year	1950 - 2049
YY TT	year, dekad in year	50 01 - 49 36
YY:TT	year, dekad in year	50 01 - 49 36
YY-TT	year, dekad in year	50 01 - 49 36
YY mmm	year, month	50 Jan - 49 Dec
YY month	year, month	50 January - 49 December
...		

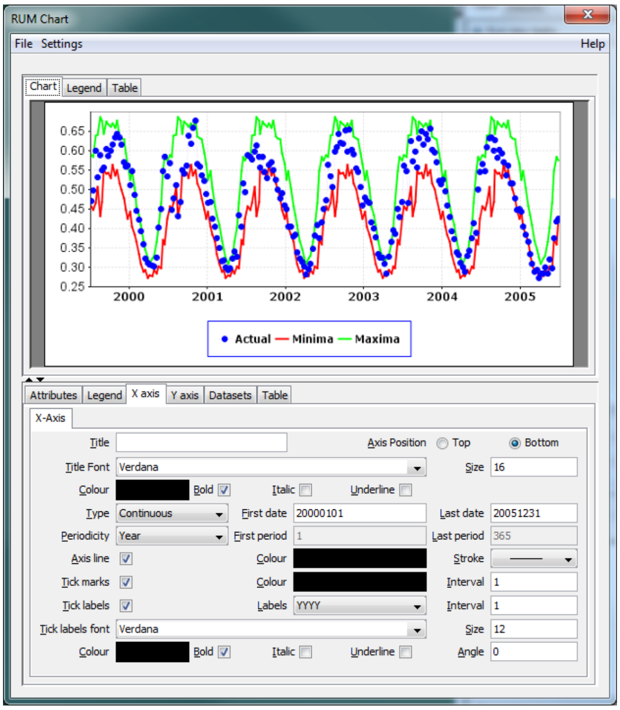
X-axis type and range

The X-axis type can be continuous or annual.

continuous X-axis

In case of a continuous X-axis, the date interval to be plotted must be specified via the first and last date fields.

In case the dataset type to be plotted is a long term average type, or an Average, Minimum or Maximum operation is selected, its periodic values are repeated over the complete interval.

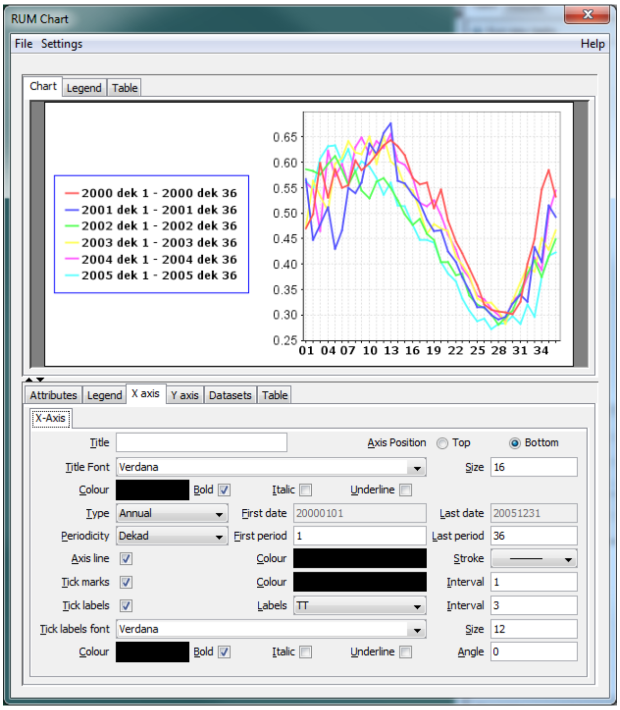


continuous X-axis example

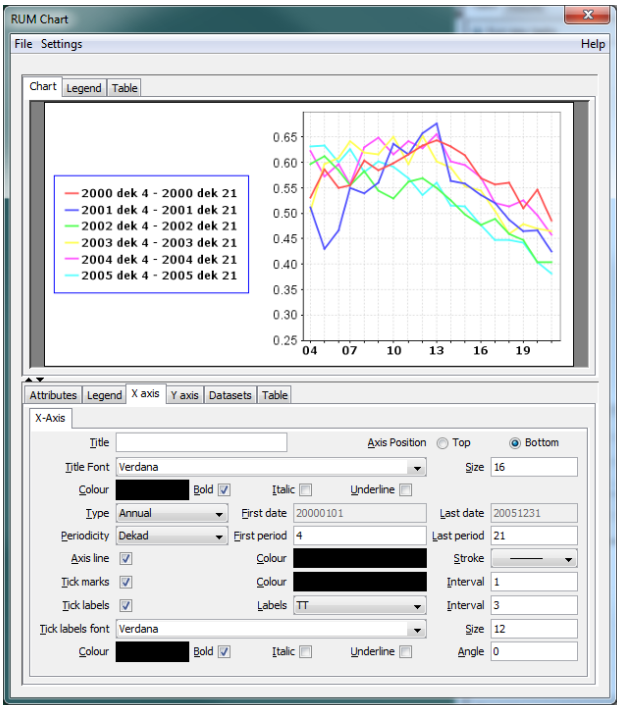
annual X-axis

In case of an annual X-axis, the date interval to be plotted must be specified via the first and last period fields. This interval covers at most one year and starts from the first period. Datasets are divided into multiple dataset series, each containing the points for a single one year interval. These intervals start at the specified first period and stretch till the specified last period.

In case the last period is greater than the first period, each dataset series will contain points from one calendar year (complete or a subset).

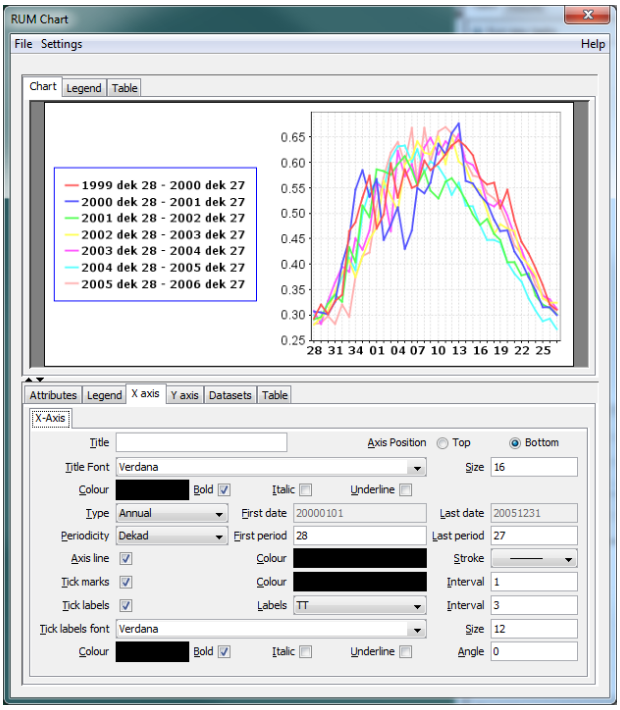


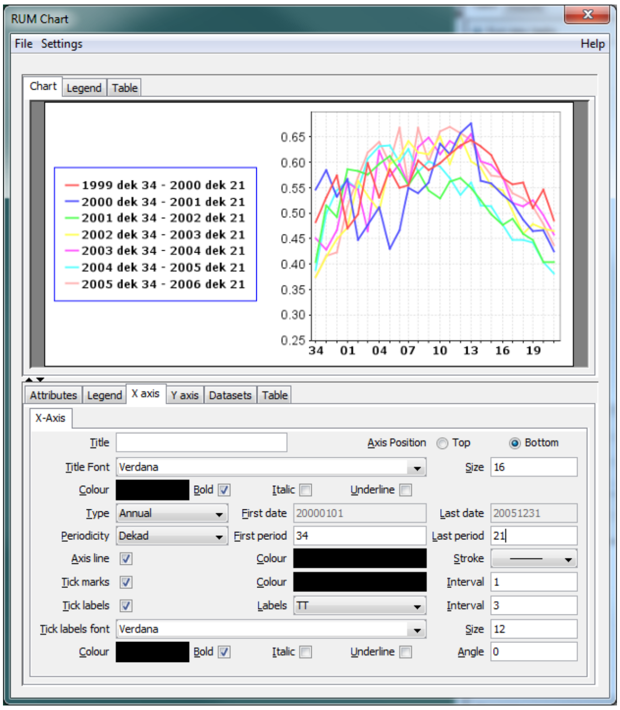
annual X-axis, complete calendar year example



annual X-axis, subset of a calendar year example

In case the last period is smaller than the first period, it is considered to belong to the next calendar year, each dataset series will therefore contain points from two successive calendar years.





annual X-axis, complete year example

annual X-axis, subset of a year example

4.2.3.5. Y-axis

Charts contain at least one Y-axis. Y-axis can be positioned at the top or at the left or right side of the chart.

In case the Chart contains multiple datasets which belong to different variables, or which are processed with different operation types or modes, additional Y-axis will appear.

Y-axis title

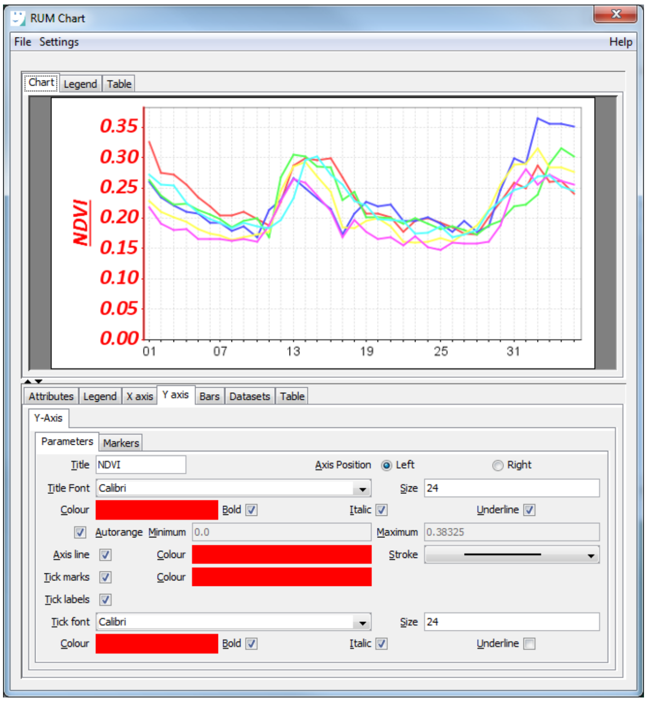
The font, the font size, the font colour and the font type of the Y-axis title can be specified.

Y-axis ticks and labels

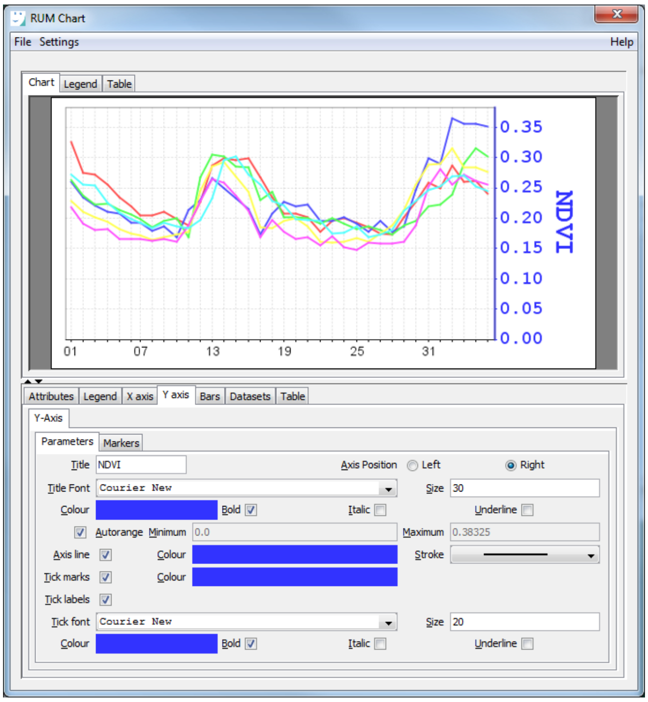
The Y-axis tick marks and tick font, the font size, the font colour and the font type of the axis labels can be specified.

Y-axis range

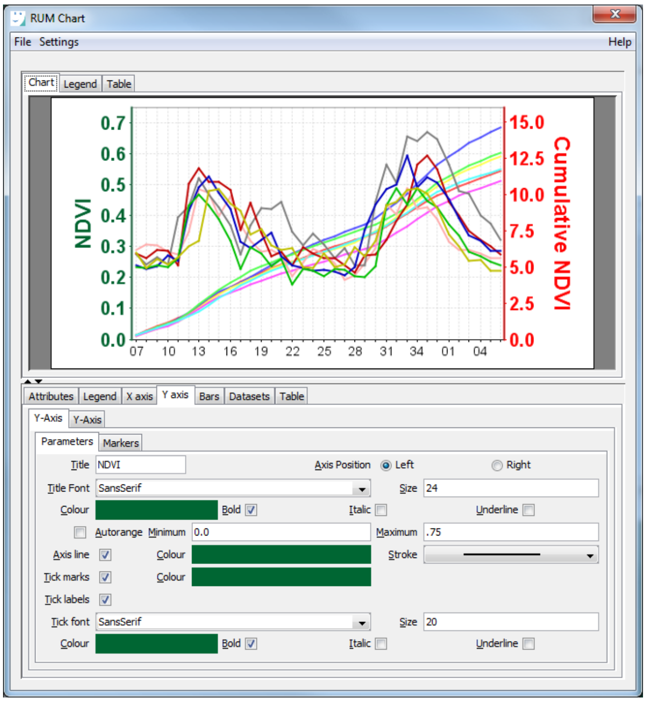
The Y-axis range can be determined automatically, or it can be specified explicitly via the minimum and maximum fields.



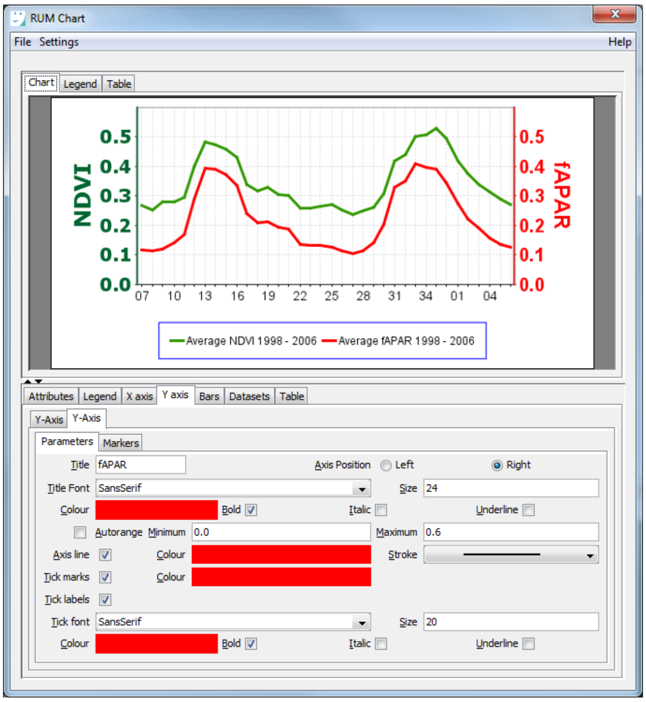
Y-axis position, colour, font, ... example



Y-axis position, colour, font, ... example



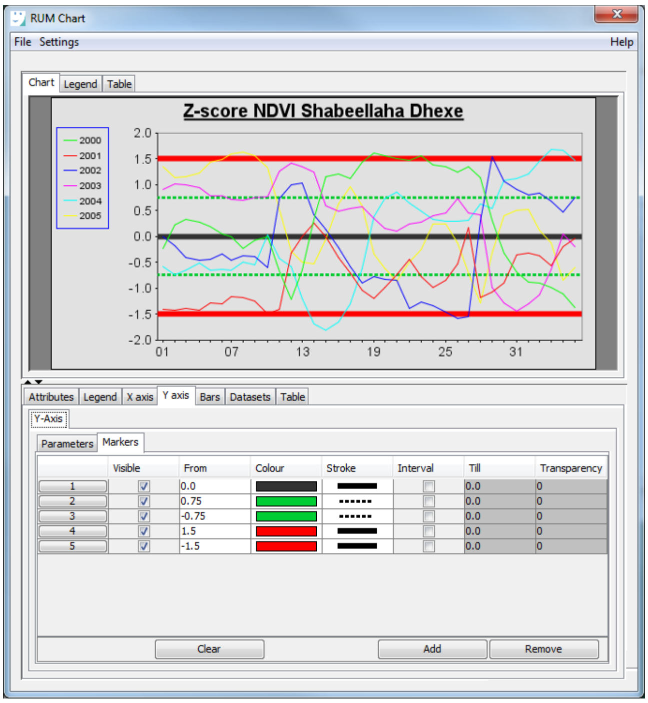
second Y-axis due to different modes



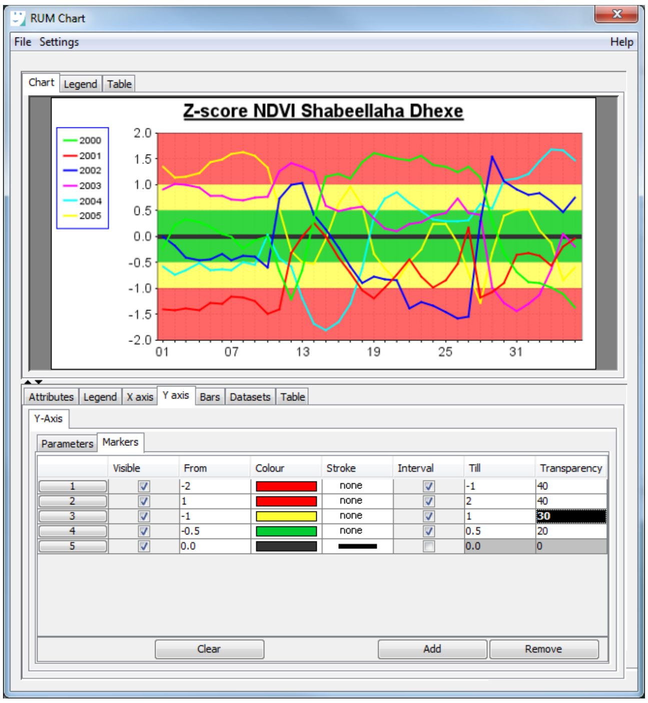
second Y-axis due to different variables

Y-axis markers

Horizontal ‘marker’ lines or intervals can be assigned to an Y-axis. In case of line-markers, their position, colour and stroke must be specified. For interval markers an additional till-value and a transparency factor (0-100) for the interval must be specified.



Y-axis marker lines example



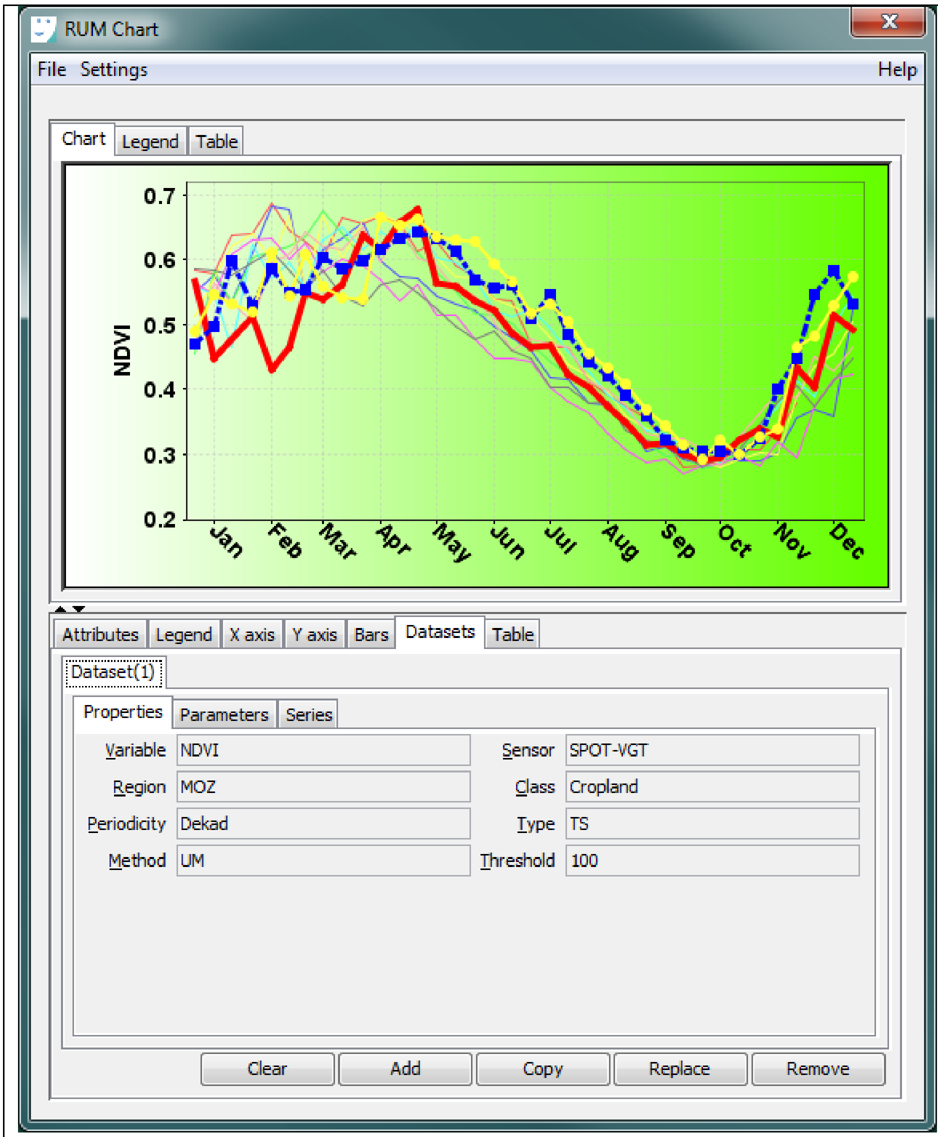
Y-axis marker intervals example

4.2.3.6. Datasets

Each dataset in a Chart has its own Dataset panel.

From such panel, datasets can be manipulated: datasets can be added, removed, copied and replaced. Furthermore, it contains three subpanels: a Properties part, a Parameters part and a Series part.

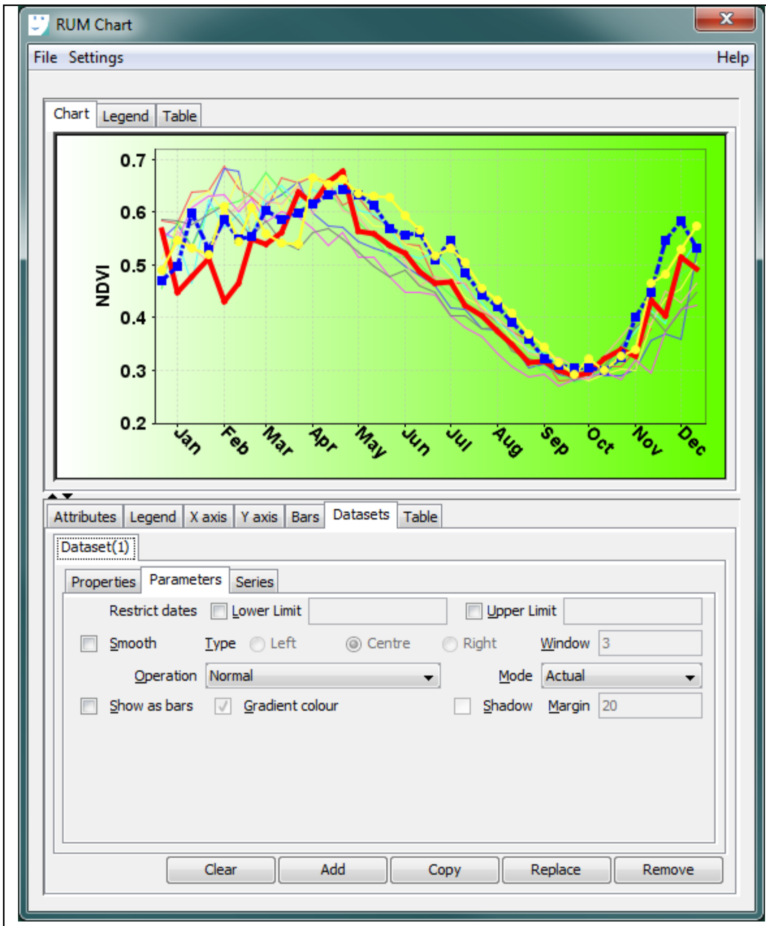
- In the Properties part the information about the actual dataset is shown;
- In the Parameters part the overall parameters can be specified (date range, smoothing, operation,…).
- In the Series part, the look and feel parameters for the individual dataset series of the dataset can be specified (colour, shape, stroke,…).



Properties:
the Region, Class, Sensor, Variable, Dataset Type, Periodicity and Unmixing Method abbreviations, and in case of Weighted Means unmixing the Threshold value

Dataset manipulations

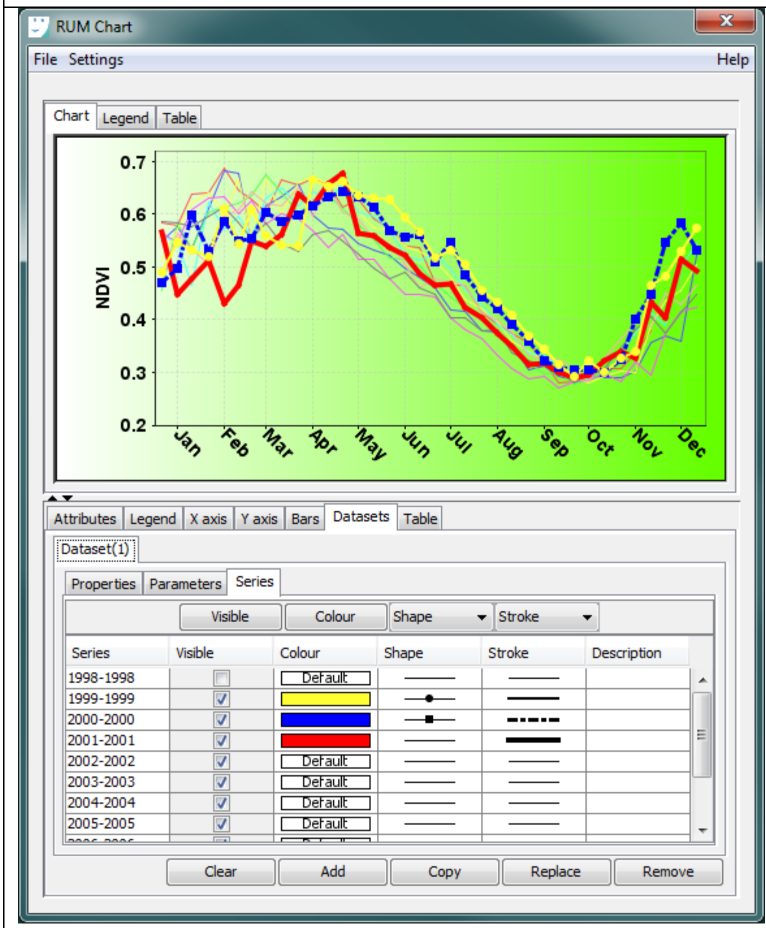
RUM Chart Dataset panel – Properties subpanel



Parameters:
Dataset range
Dataset operations
Dataset common Look and Feel

Dataset manipulations

RUM Chart Dataset panel – Parameters subpanel



Series:
- show/hide
- colour
- shape
- stroke
- description

Dataset manipulations

RUM Chart Dataset panel – Series subpanel

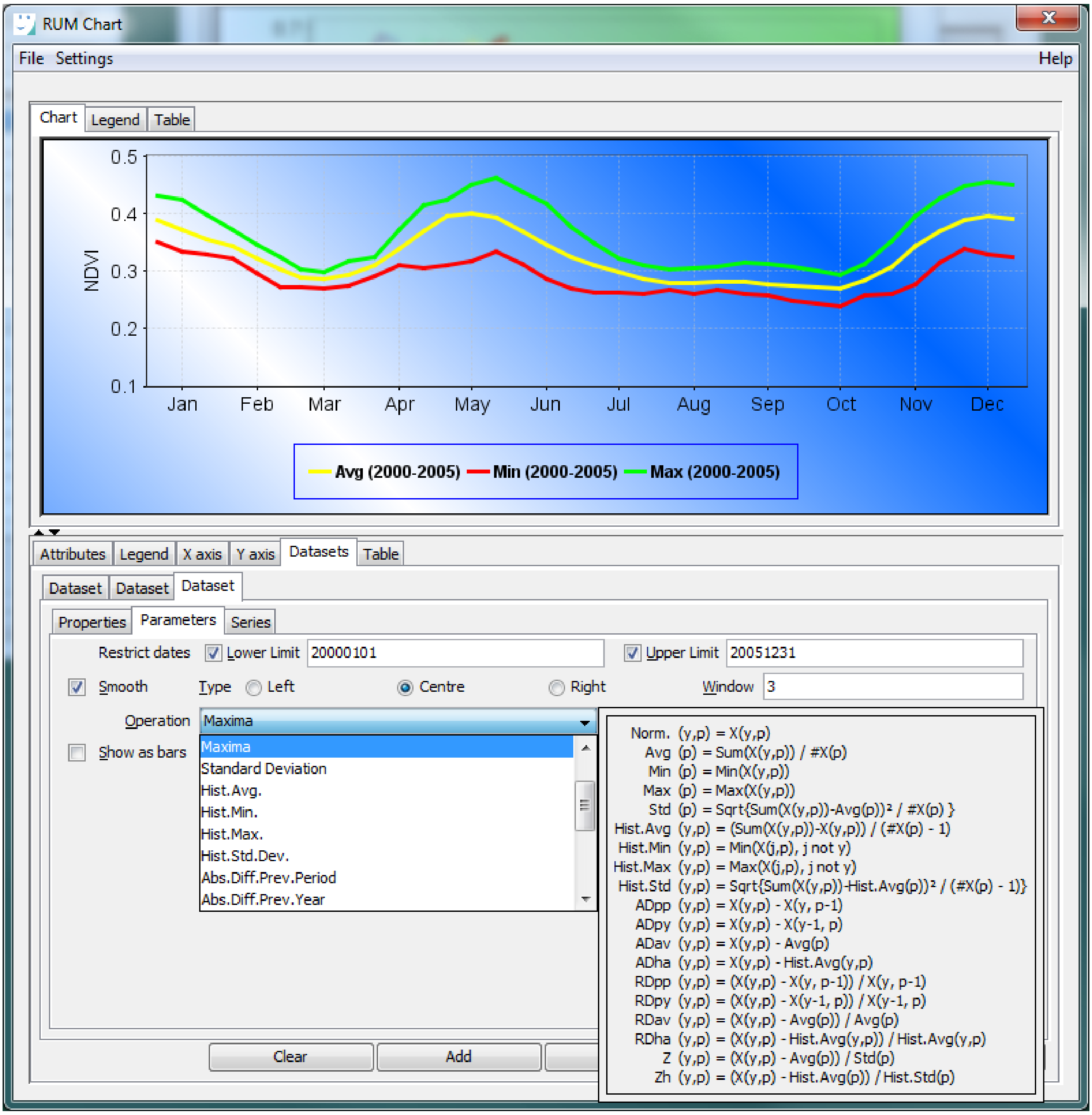
Date range restriction

For each dataset in a Chart, an upper and lower date restriction can be selected and specified in its dataset parameters panel, thereby restricting the actual data from the dataset to be used in the Chart.

Processing

For each dataset in a Chart, the processing parameters (for smoothing, operations, date interval restriction and cumulating) can be specified or selected in its dataset panel. Certain limitations apply:

- date interval restriction and operations are only available for time series datasets;
- cumulative mode is not available for relative difference operations, standard deviation or Z-scores.



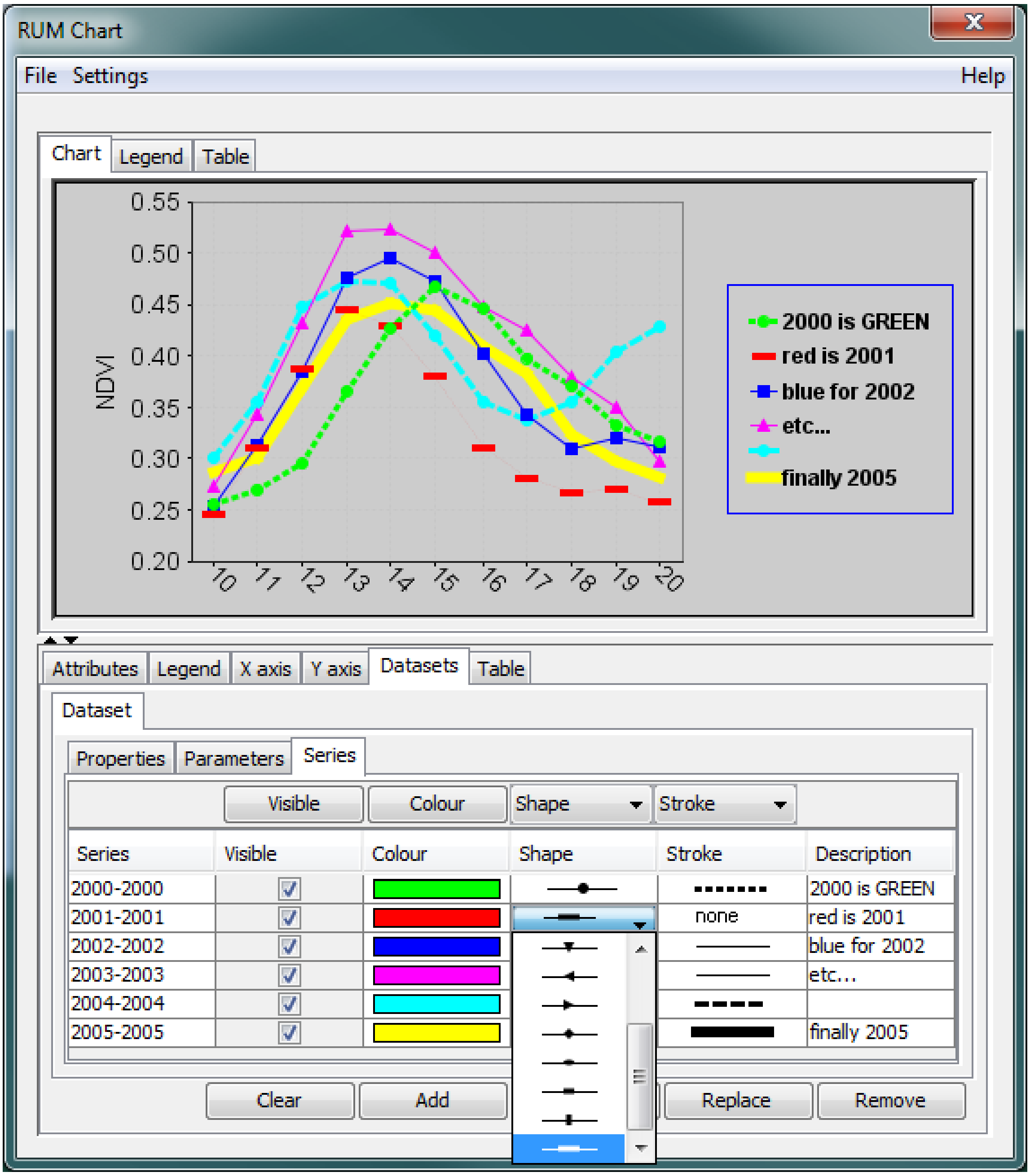
RUM Chart Dataset panel – Parameters subpanel: operations and date range restriction

Dataset series

In the Series subpanel, for each dataset series in a dataset, the colour, shape and stroke can be selected. (In case of "Bars", the shape option is not applicable).

Series can be made invisible in the graph. This has only to do with the visual aspect of the chart, values obtained from operations or smoothing are not affected.

A specific description can be specified for each series. These descriptions can be used in the legend (by using the %0 parameter in the legend string).



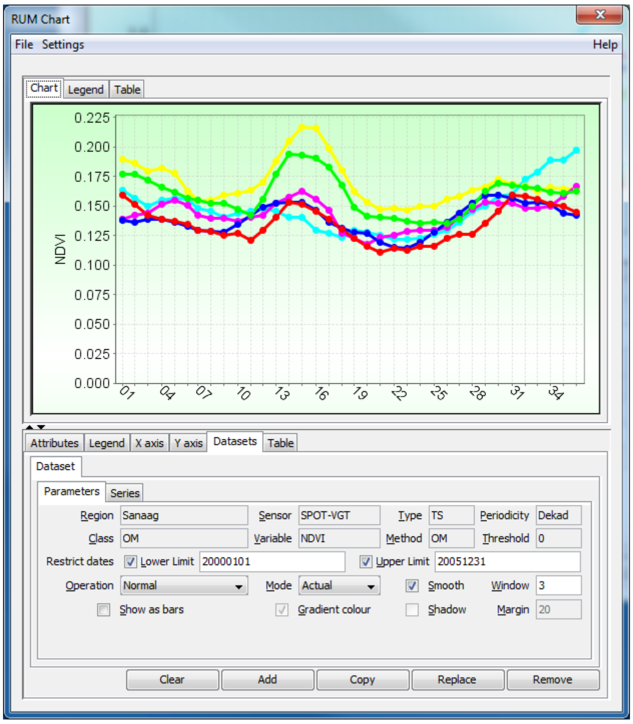
RUM Chart Dataset panel – Series subpanel: series Look and Feel

Manipulations

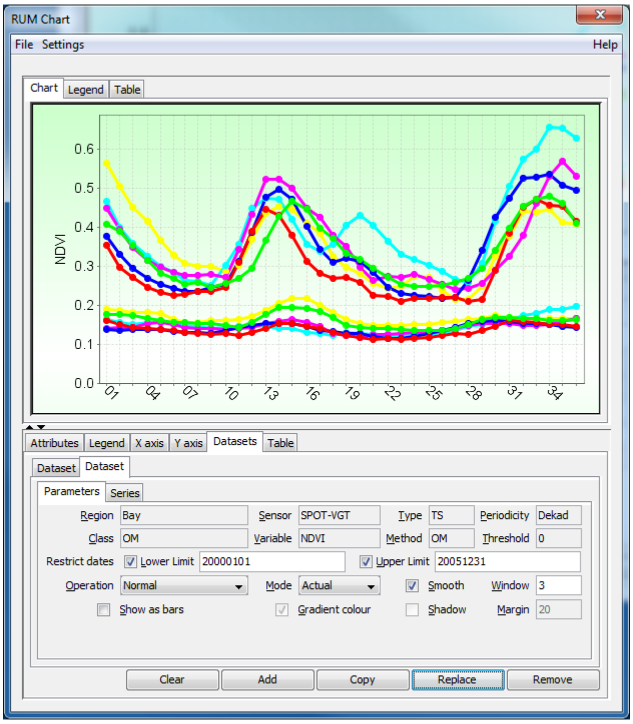
Following dataset manipulation functions are available:

Clear	Removes all datasets from the chart.
Add	Add a dataset to the chart. A database browser window will open, allowing the selection of the Dataset to be added.
Copy	Copies the selected dataset in the chart. This copy includes the settings (e.g. colours, shapes and strokes) of the selected dataset.
Replace	Replaces the selected dataset in the chart. A database browser window will open, allowing the selection of a new dataset. The settings (e.g. colours, shapes and strokes) of the original selected dataset will be kept.
Remove	Removes the selected dataset from the chart.

example: comparing regions



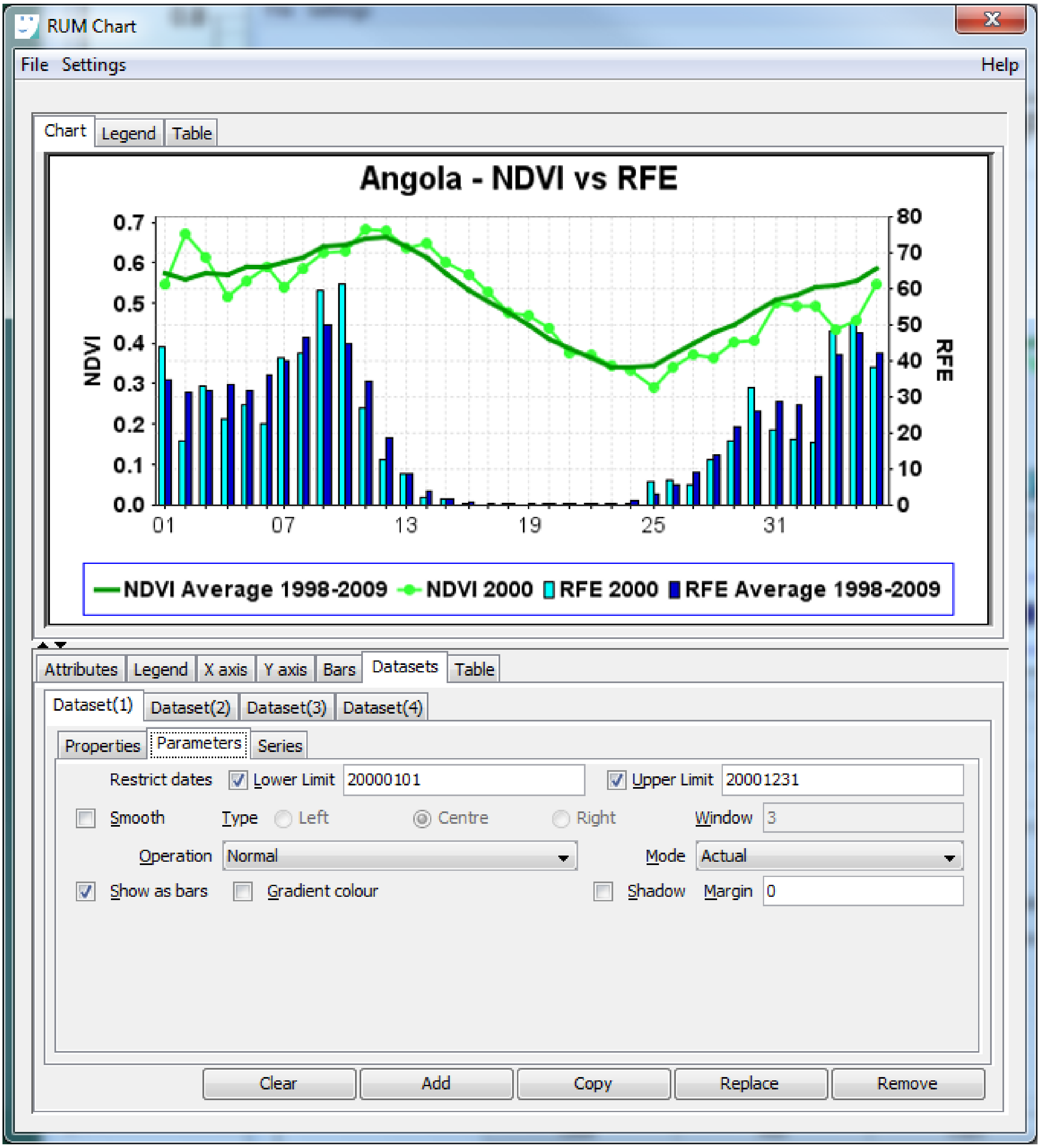
set up chart for the first dataset (first region)



use Copy to clone the dataset - including its settings
use Replace to select another region (second region)

4.2.3.7. Bars

A dataset in a chart can be displayed as a set of lines or bars, however, in a chart containing multiple datasets, only those with similar periodicity can selected to be displayed as bars.



RUM Chart Dataset panel – Parameters subpanel

When selecting the “Show as bars” option of a dataset, this option will automatically be de-selected for all other datasets with different periodicity in the chart.

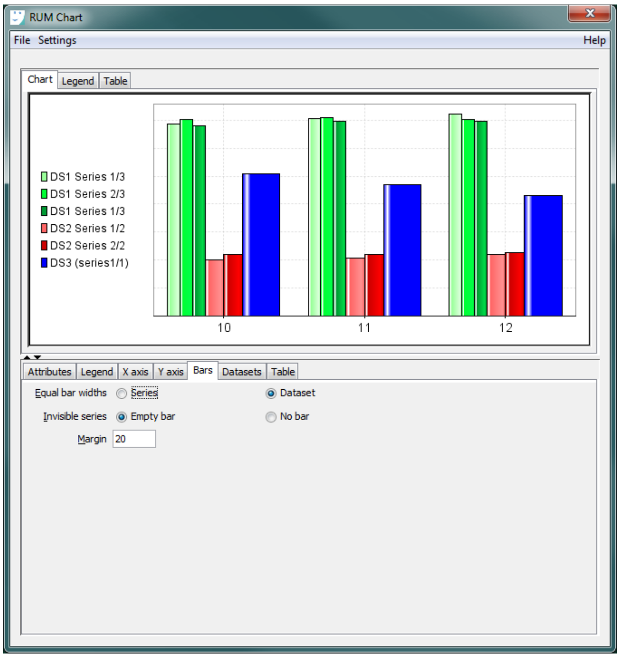
The controls, related to bars, are split over two regions:

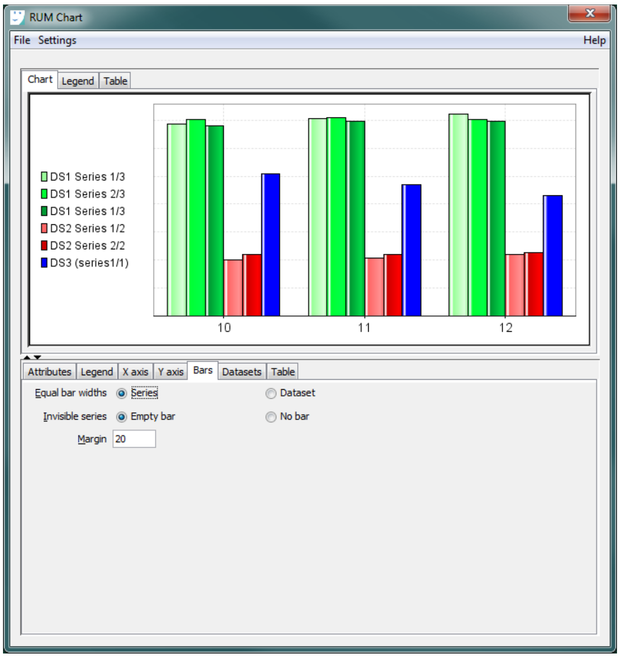
- the “Bars” panel itself, which contains overall controls, which are mostly only applicable in case multiple datasets are to be displayed as bars simultaneously;
- the “Dataset-Parameters” panels of each dataset, which contain the selection whether or not to display the dataset as (a set of) bars, and some settings specific for the dataset;

Controls on the Bars panel:

- “Equal bar widths”: specifies how to distribute the space used to draw the bars.
  Options are “Dataset” or “Series”:
  - In case “Dataset” is selected, each dataset (displayed as bars) will have the same amount of space. This means that the width of the dataset series can be different per dataset.
  - In case “Series” is selected, each series of the datasets (displayed as bars) will have the same amount of space. This means that the total width of the group of bars representing a dataset can be different per dataset.

in this example we have three datasets (DS1, DS2, DS3)
DS1 contains 3 series;
DS2 contains 2 sries;
DS3 contains 1 series.





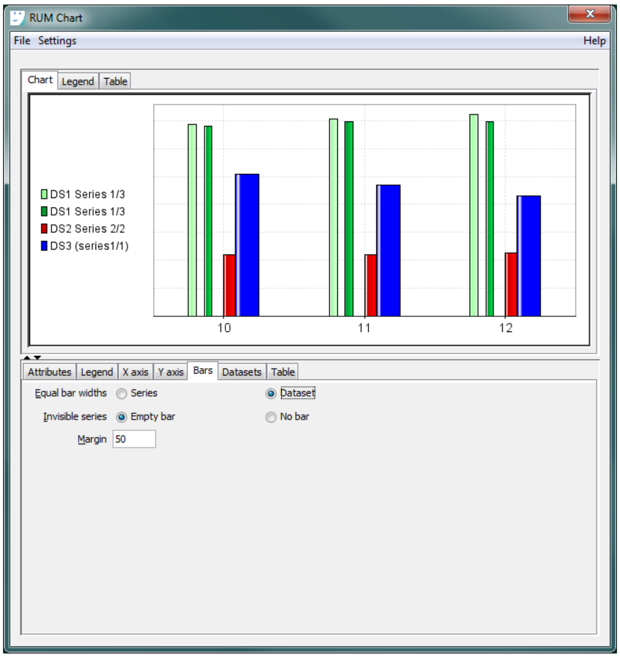
Dataset Equal bar widths example Series Equal bar widths example

- “Invisible series”: specifies the handling of dataset series which are flagged as “invisible” on the Dataset-Series subpanel.
  - In case “Empty bar” is selected, invisible dataset series will be displayed as empty bars;
  - In case “No bar” is selected, they will disappear completely.

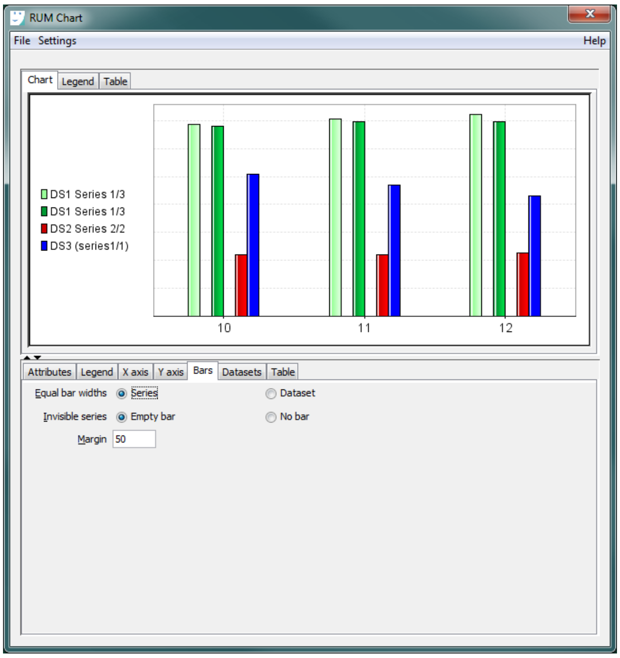
Remark: this option acts on (complete) invisible series. This has nothing to do with (specific) missing values in a series. Missing values in a visible series will show as empty bars.

in this example we have three datasets (DS1, DS2, DS3)

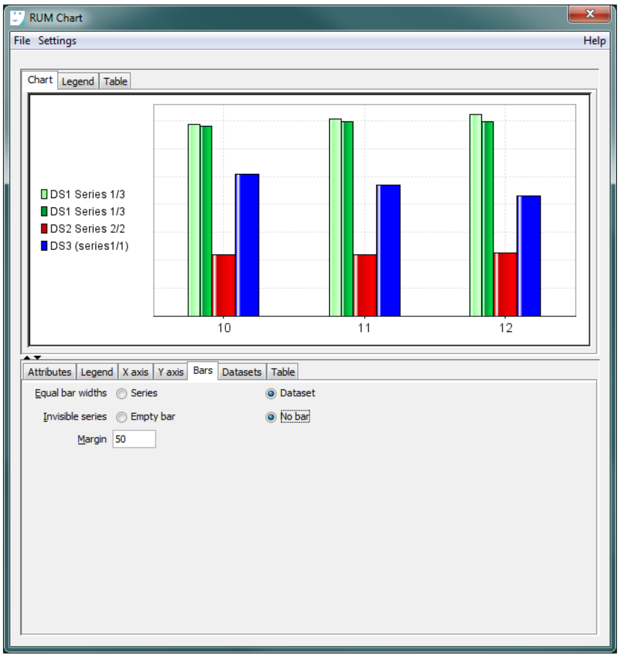
DS1 contains 3 series, of which the middle one is invisible;
DS2 contains 2 series, of which the first one is invisible;
DS3 contains 1 (visible) series.



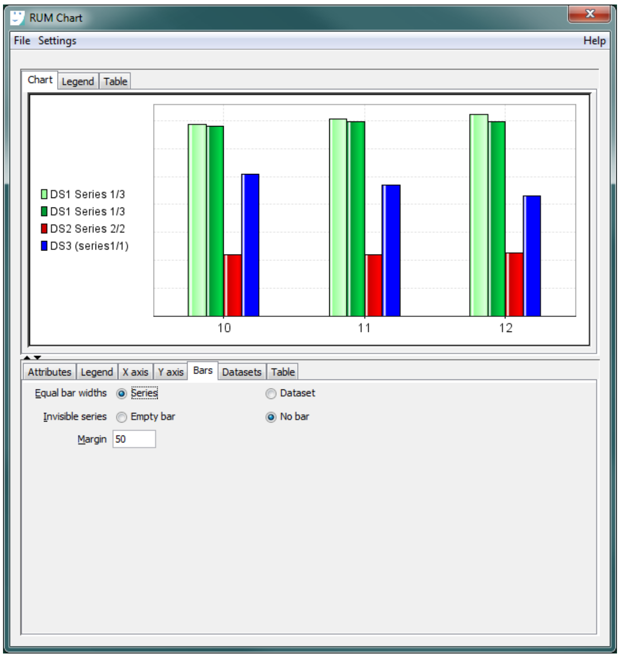
Equal bar width: Dataset
Invisible series: Empty bar



Equal bar width: Series
Invisible series: Empty bar



Equal bar width: Dataset
Invisible series: No bar



Equal bar width: Series
Invisible series: No bar

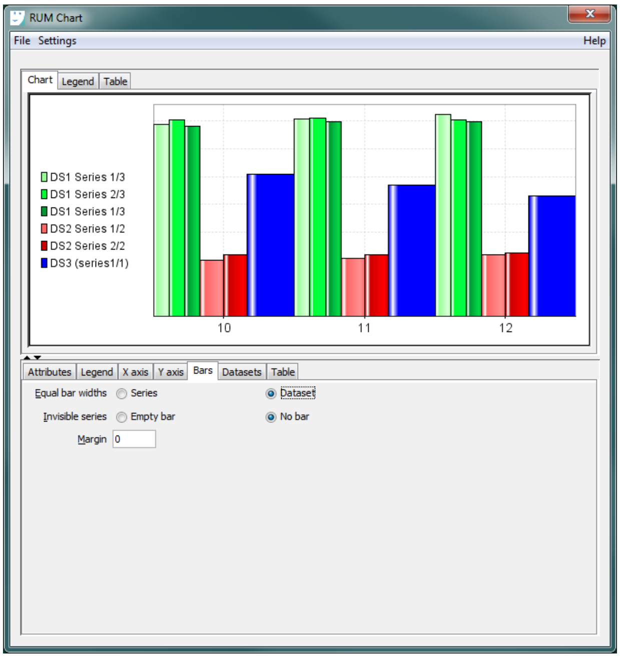
- “Margin”: specifies the amount of space (percentage) around the (groups of) bars in the (time) period in which they are drawn. This margin combines with the margins specified in the Dataset-Parameters panels.

Bar – related controls on the Dataset-Parameters subpanel:

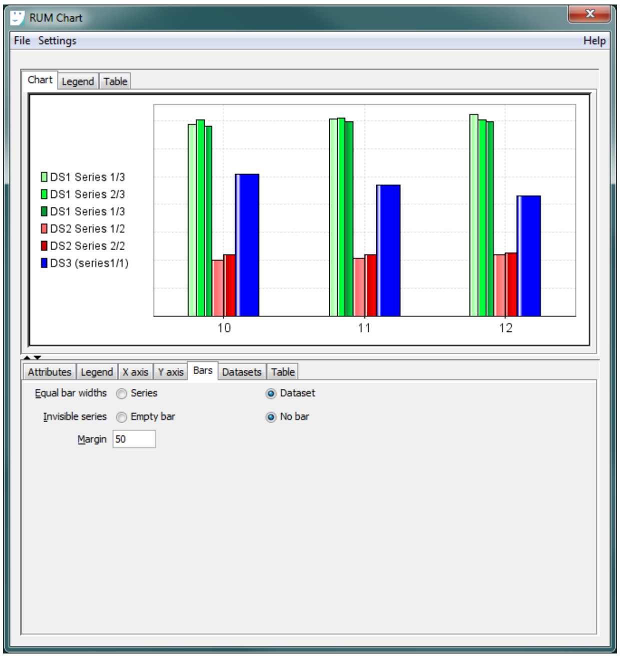
- “Margin”: specifies the amount of space (percentage) around the (groups of) bars for the dataset;

in this example we have three datasets (DS1, DS2, DS3)

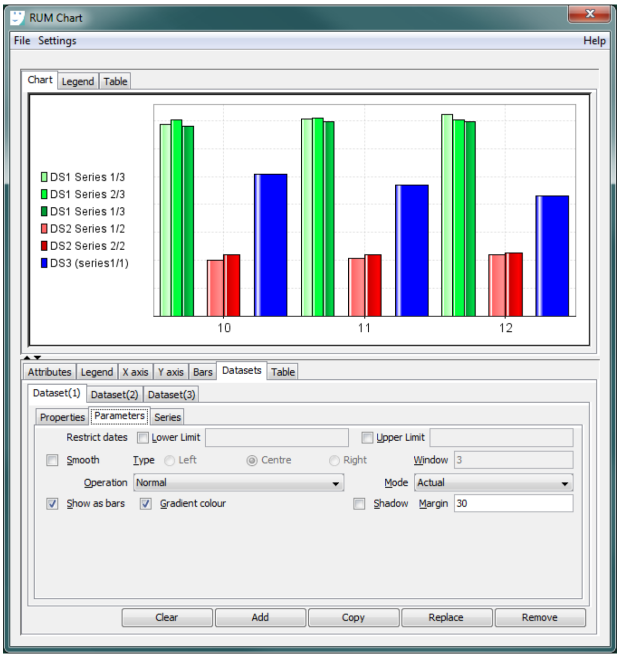
DS1 contains 3 series;
DS2 contains 2 series;
DS3 contains 1 series.



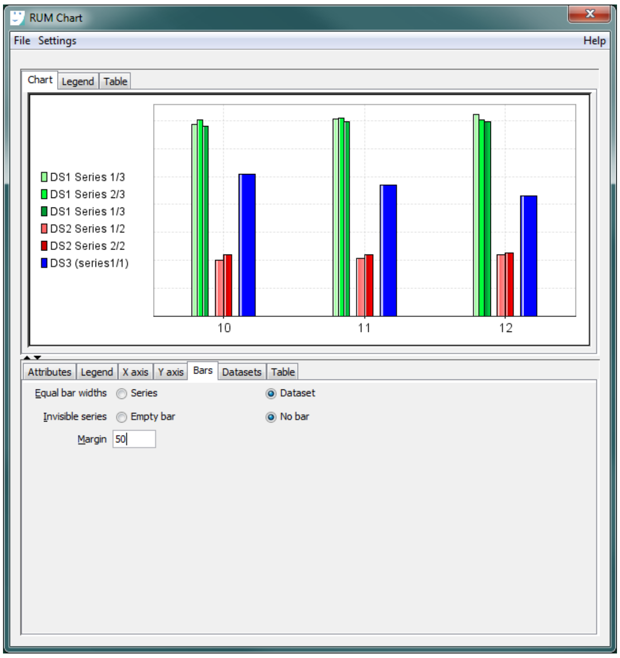
Bar-Panel Margin: 0
(Each) Dataset Parameters Margin: 0



Bar-Panel Margin: 50 (%)
(Each) Dataset Parameters Margin: 0

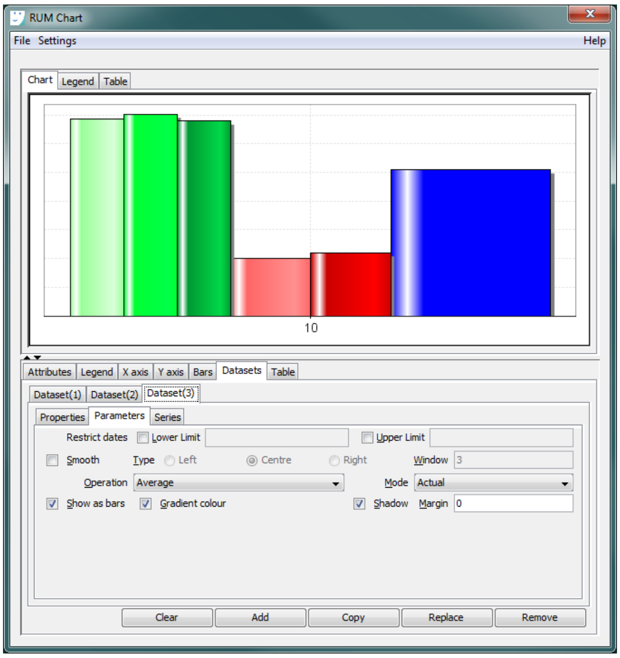


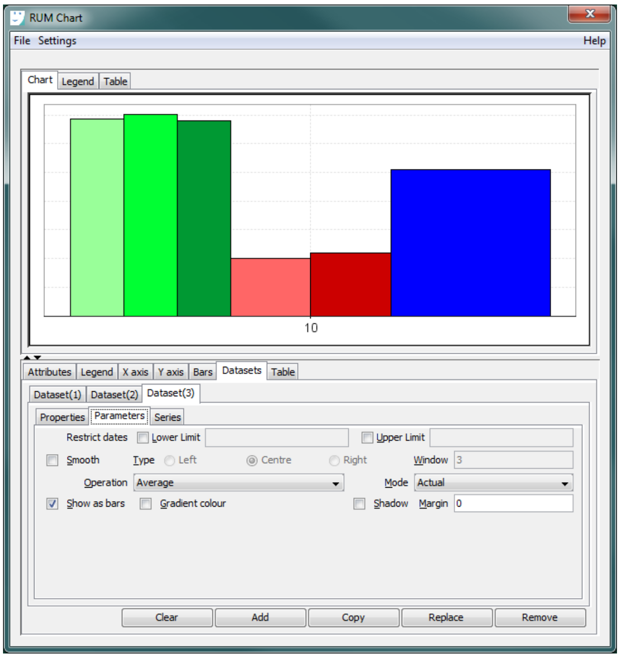
Bar-Panel Margin: 0
(Each) Dataset Parameters Margin: 30 (%)



Bar-Panel Margin: 50 (%)
(Each) Dataset Parameters Margin: 30 (%)

- “Gradient colour”: toggles between a solid or gradient colour used to fill the bars;
- “Shadow”: allows the bars to have a drop shadow.





Gradient colour and Shadow selected — Gradient colour nor Shadow selected

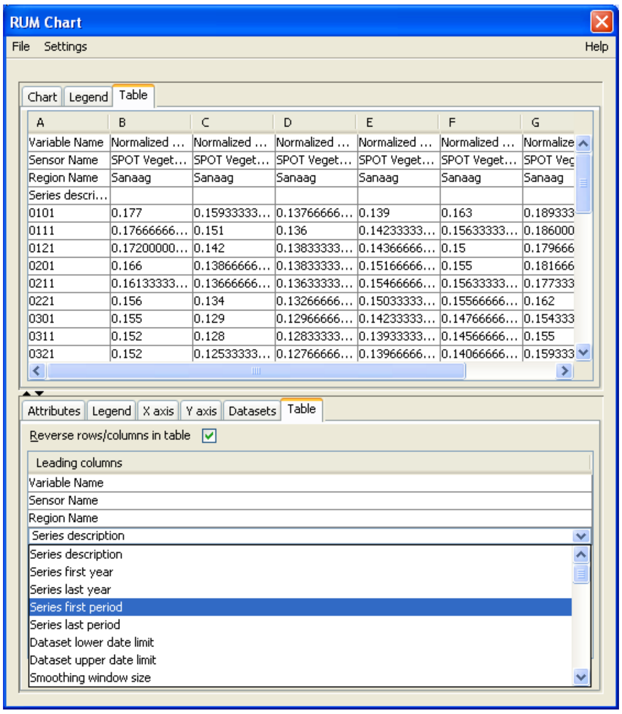
4.2.3.8. Table

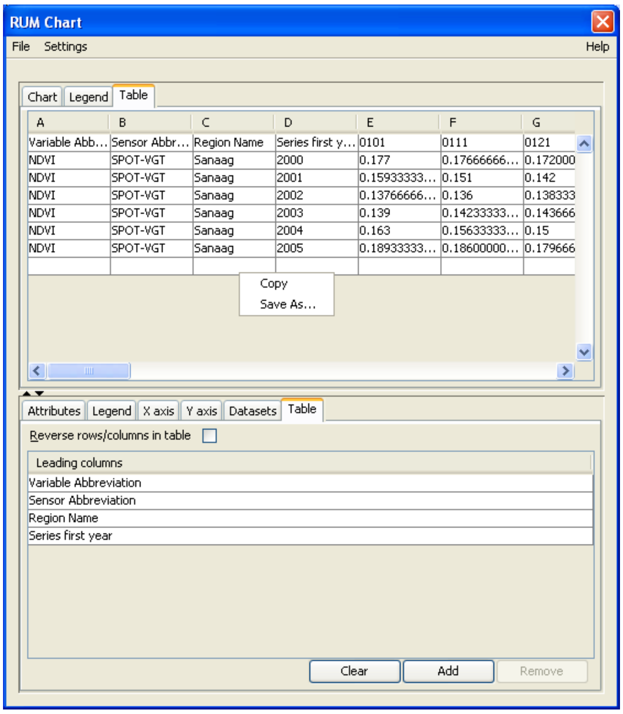
The data collected in a RUM Chart can be exported as an ASCII table.

The dataset and dataset series parameters which will appear as leading columns in the table view can be selected. The parameters available are the same as those for the legend.

The table layout (column or row orientated layout) can selected.

The table can be exported by right clicking in the table view panel, and selecting its contents to be copied or saved as an ASCII file





4.2.4. RUM Charts Settings

When starting a RUM Chart, it uses defaults for most general settings such as the chart size and its background. These settings can be changed by the user, and saved as new default settings via Settings>Set defaults . From then on these will be used upon starting new RUM Charts. The actual settings for which defaults are saved are:

- Chart Title font name, Font size, Title colour, Bold, Italic, Underline, Position. Background Colour1, Background Colour2, Background Gradient type. Grids and Grid colours, Lock chart size, Locked width, Locked Height.
- Show legend, Legend Position, Legend pattern, Legend font name, Font size, Legend colour, Bold, Italic, Underline. Margin, Padding, Border, Border Width, Border Colour.
- X-axis Title font name, Font size, Title colour, Bold, Italic, Underline, Position. Line and Line colour, Marks and Marks colour, Tick labels, Tick labels font name, Font size, Tick labels colour, Bold, Italic, Underline.
- Y-axis Title font name, Font size, Title colour, Bold, Italic, Underline, Position. Line and Line colour, Marks and Marks colour, Tick labels, Tick labels font name, Font size, Tick labels colour, Bold, Italic, Underline.

The original default settings can also be restored via Settings>Reset defaults.

Remark: these settings will be used by the RUM Matrix charts as well.

4.2.5. RUM Charts File menu

A PNG file of a RUM Chart can be created via the Export PNG entry in the File menu.

A RUM Chart itself can be saved / re-opened as a CNC file via the File menu.

Such CNC file contains all chart settings and its datasets properties. It does not contain the actual values from the datasets. Instead each time a CNC file is re-opened, the dataset values are retrieved from the database via the datasets properties. This means that a chart can be updated, after new values have been added to the database, just by re-opening its CNC file.

CNC files can also be used as templates:

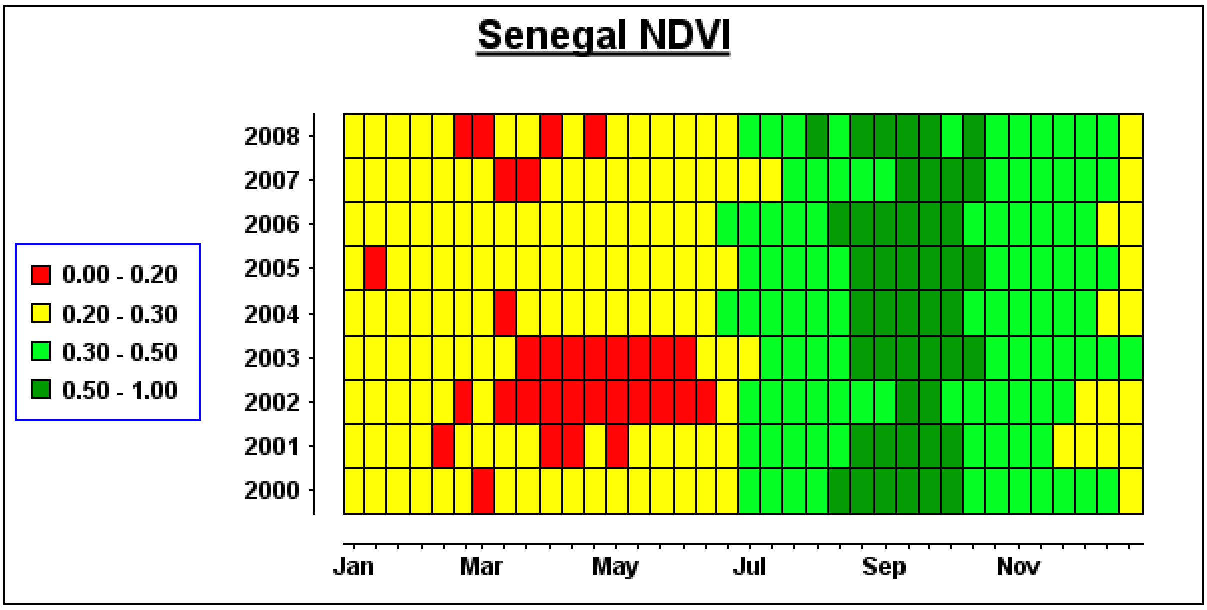
- by the RUM Chart utility itself, when creating new Charts by replacing the datasets in existing Charts.
- by the RUM Chart series tool which facilitate the creation of series of similar PNG files, based on an existing RUM Chart.

4.3. RUM Matrix charts

4.3.1. Introduction

The RUM Matrix chart utility offers an alternative visualisation of the RUM data.

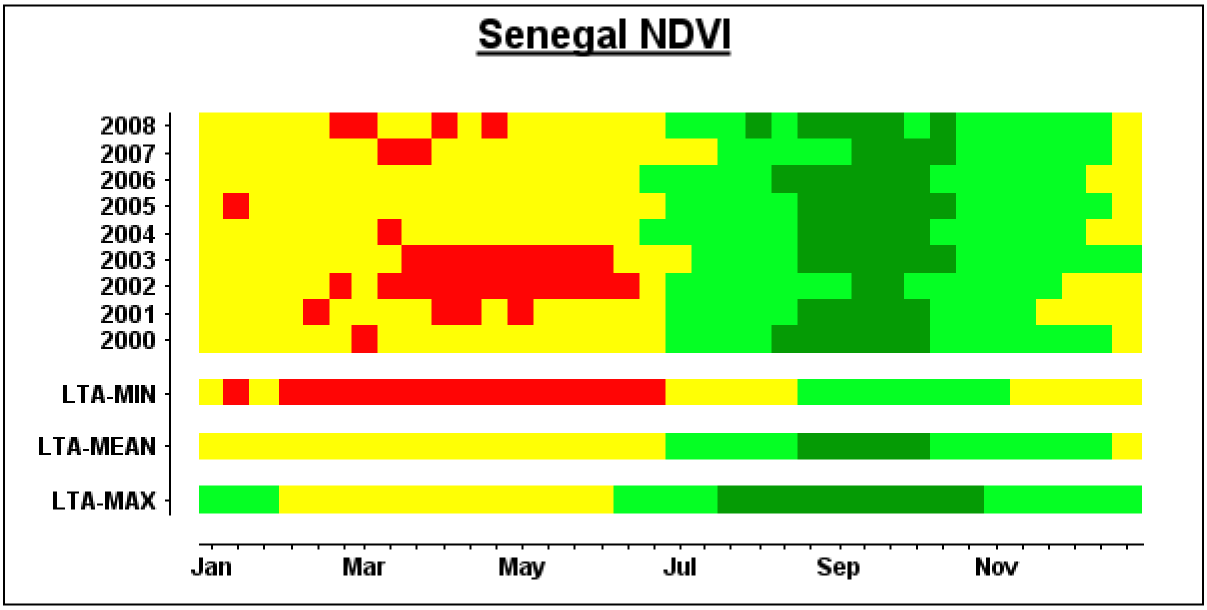
A basic RUM Matrix chart is a table view of a dataset. The table cells represent the datasets values by means of a colour map. The table rows (Y-axis) contain the dataset series, the columns (X-axis) the period (time) of the dataset values.



RUM Matrix chart example

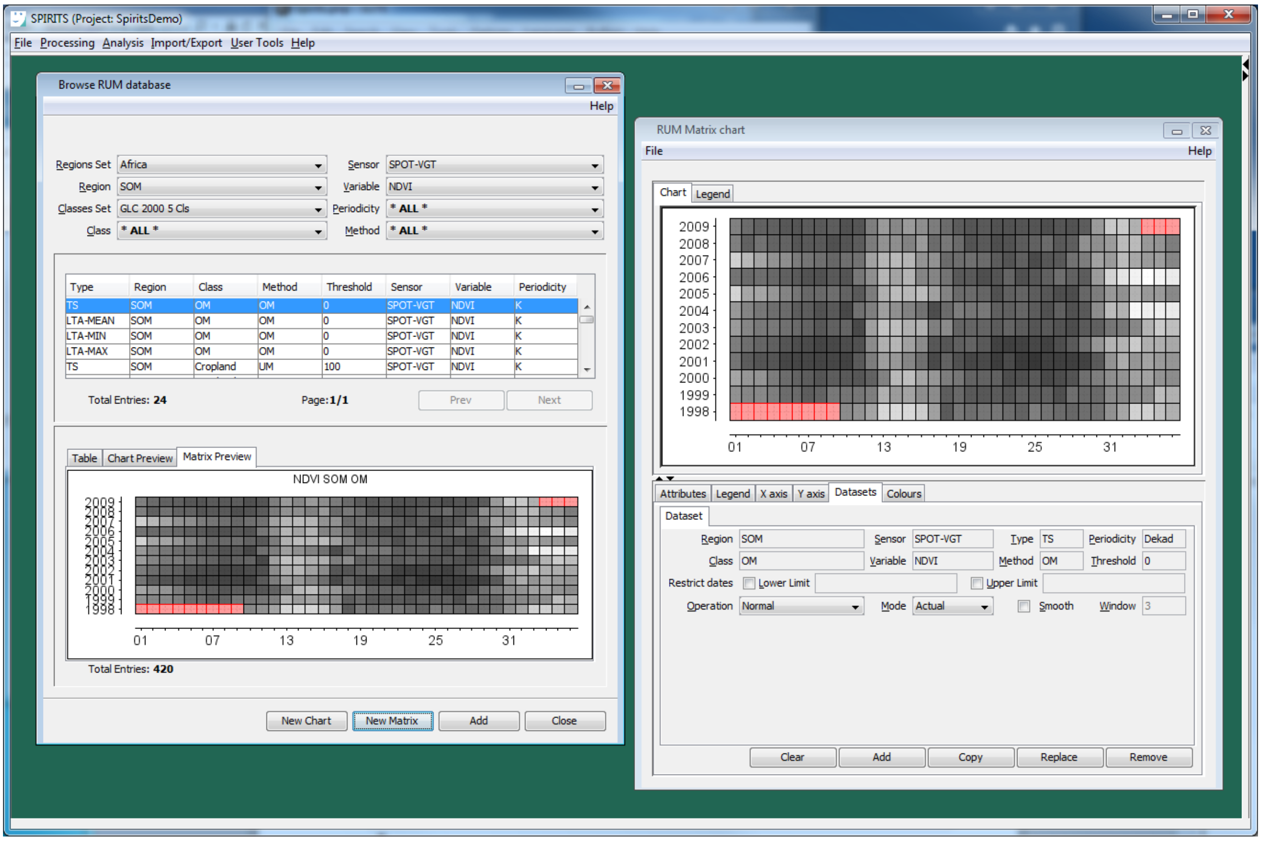
A typical RUM Matrix chart will show a single dataset, using an annual X-axis.

Multiple datasets (and continuous X-axis) are possible, but only one single colour scale is supported, thus in general matrix charts combining datasets with different variables and/or different operation types will be hard to interpret.



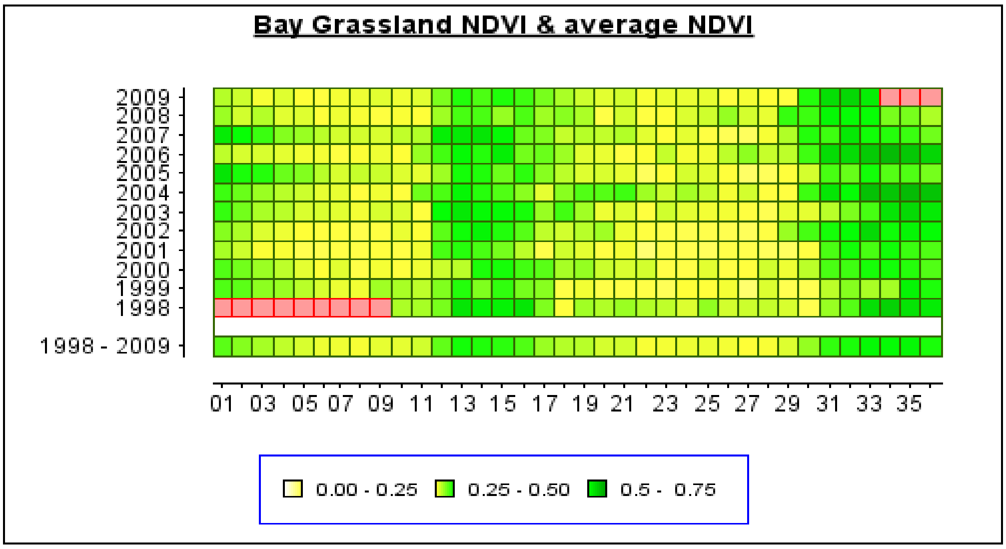
RUM Matrix chart multiple datasets example

Datasets can be sent from the database browser to a Matrix charts Form, or can be selected from a Matrix chart Form directly.



Browser :	**New Matrix**	opens a new Matrix chart containing the selected dataset;
Browser:	**Add**	adds the selected dataset to the last active chart form (Chart or Matrix);
Matrix chart:	**Add**	opens a Browser to select a dataset to be added to the Matrix chart.

Just as in the case of regular Charts, datasets can be smoothed, operations can be performed on datasets, datasets can be shown in “actual” or “cumulative” mode and the data range of the datasets can be restricted;



RUM Matrix chart Average operation example

4.3.2. RUM Matrix charts Form and Panels

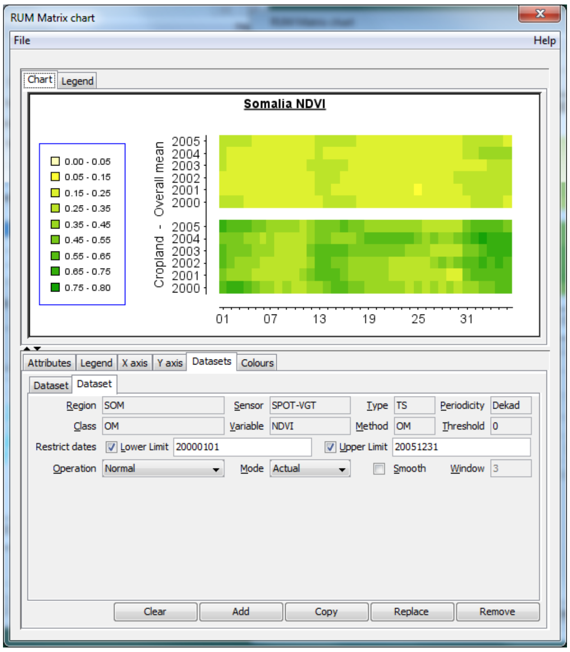
4.3.2.1. Views

Chart view

The main view for RUM Matrix charts is the Chart view panel, which contains the graphical representation of the chart data. By right clicking the panel, its contents can be copied or saved as a PNG image. A PNG file of the chart can be also created via the Export PNG entry in the File menu.

Legend view

The Legend view panel contains only the charts legend. This can be used to obtain the legend information separately in case it would occupy too much space on the chart itself. By right clicking the panel, its contents can be copied or saved as a PNG image.



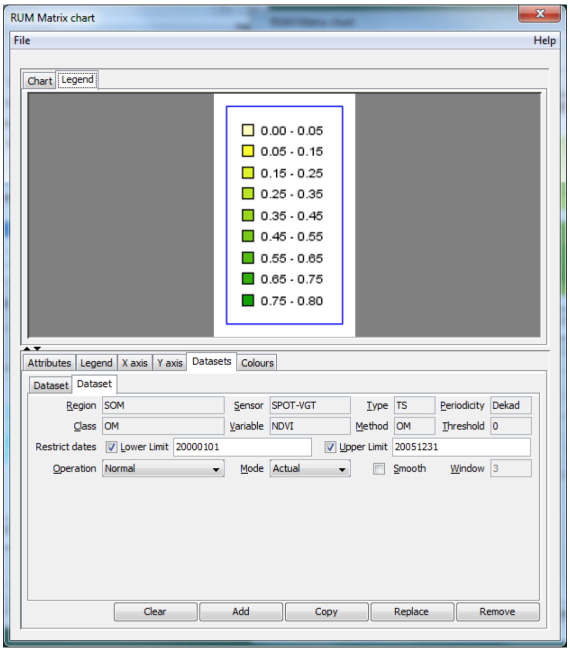
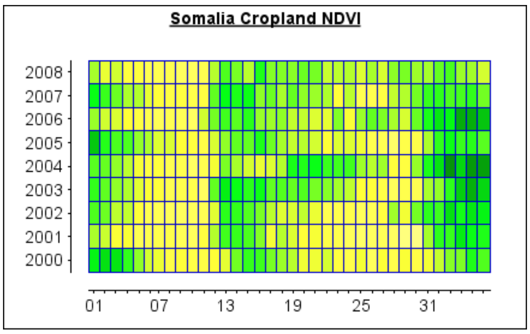
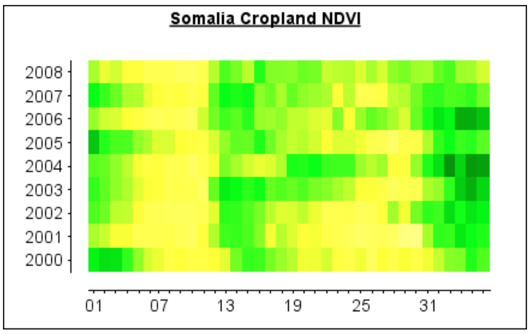


Chart view Legend view

4.3.2.2. Attributes

As with regular Charts, the chart title, background, gridlines and size can be specified in the attributes panel. In addition, it can be specified whether the matrix cells should have a border, and if so, the colour of the cell border and whether missing values should be shown (or left transparent) and if so, their colour.





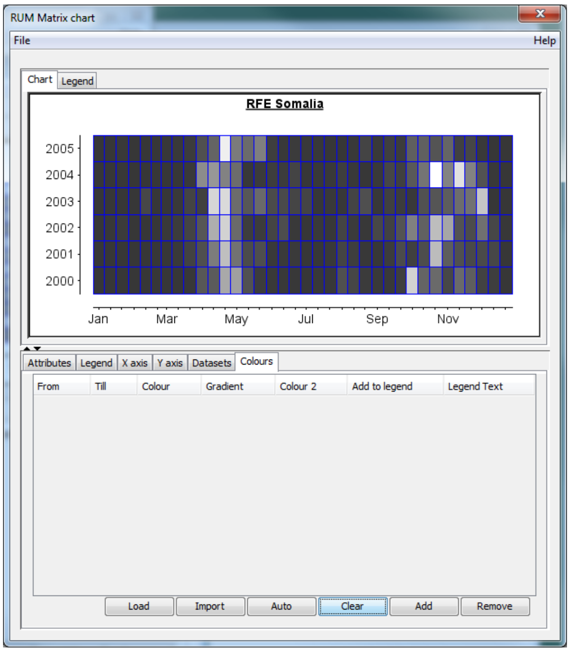
Matrix Chart with cell borders Matrix Chart without cell borders

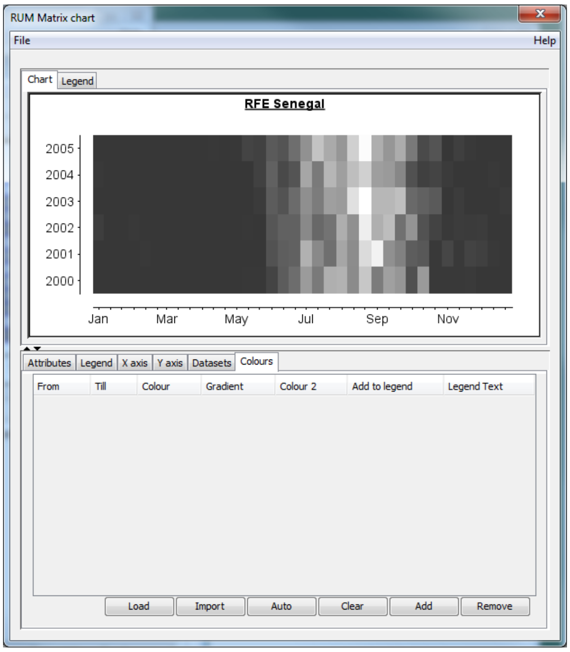
4.3.2.3. Legend

A legend can be shown on the chart, but is also available separately in the Legend View panel. The settings regarding the legends look and feel are the same as for regular charts. Its contents however originate from the colour map.

4.3.2.4. Colours

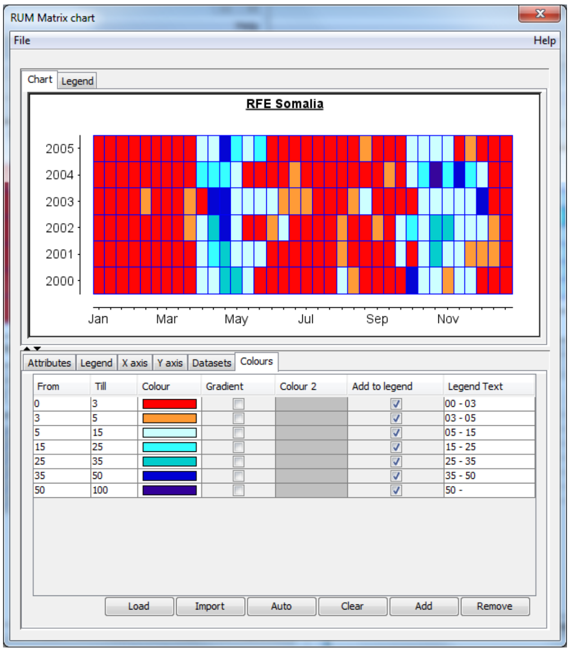
Initially (until colours are explicitly assigned) a RUM Matrix chart uses a default colour map, being a grayscale stretching between the minimum and maximum RUM values in the chart.

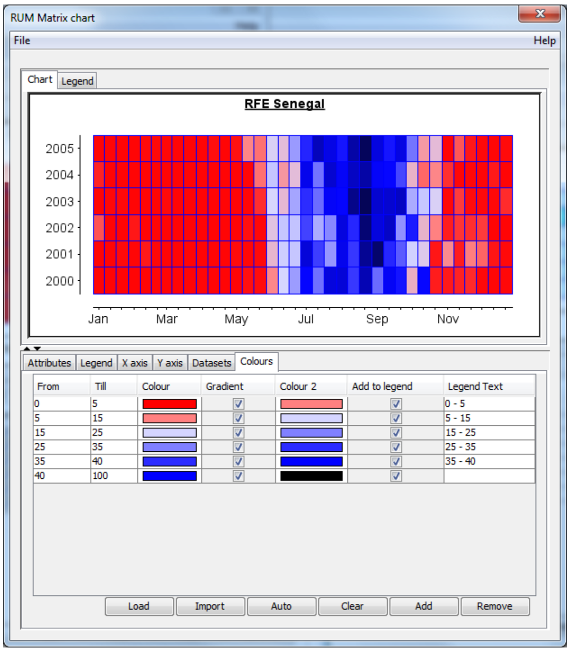




Matrix charts using default grayscales

In the Colours panel, colours can be assigned to ranges of RUM values (From value/Till value). Ranges can be assigned a solid (single) colour, or a gradient.

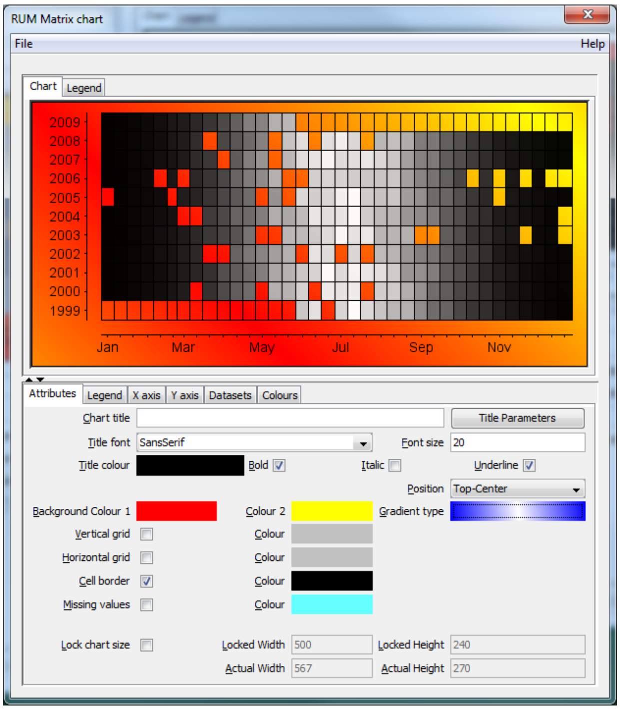


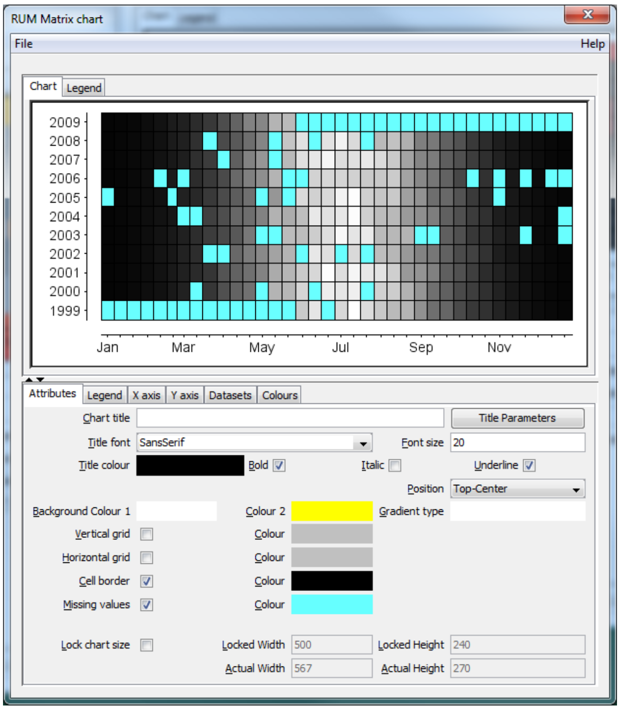


Colours panel - using single colours

Colours panel – using gradients

Missing values in a dataset series can be indicated by a selected colour, or left transparent.

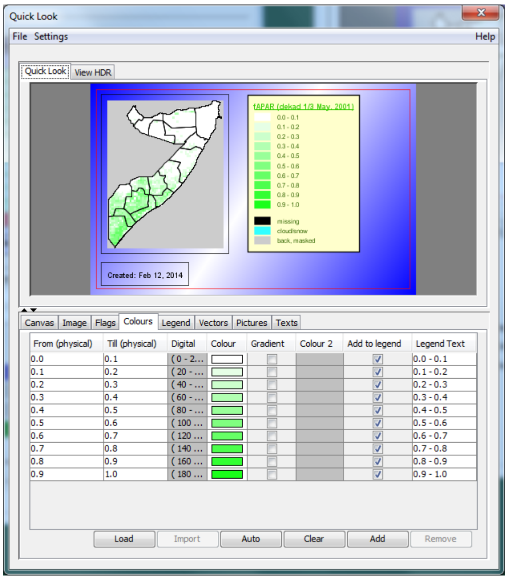


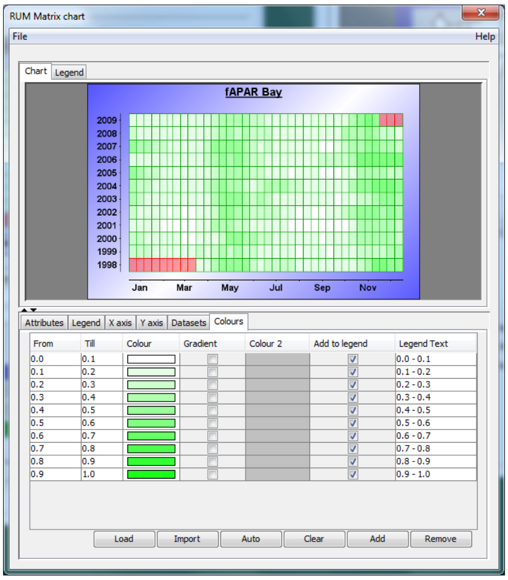


Matrix charts with no (transparant) missing values Matrix charts with missing values

For the greater part, the operation of the Colours panel is the same as for Quick Looks. Their colour panels only differ in their Load and Import actions. In this case:

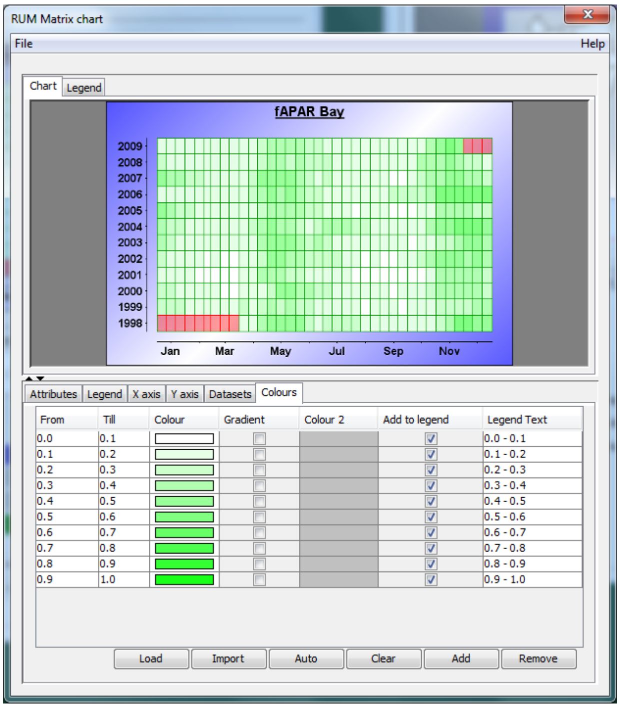
- Load : allows to load the colour table from an existing Matrix chart (*.CNM file);
- Import : allows to import the colour table from an existing Quick Look (*QNQ file);



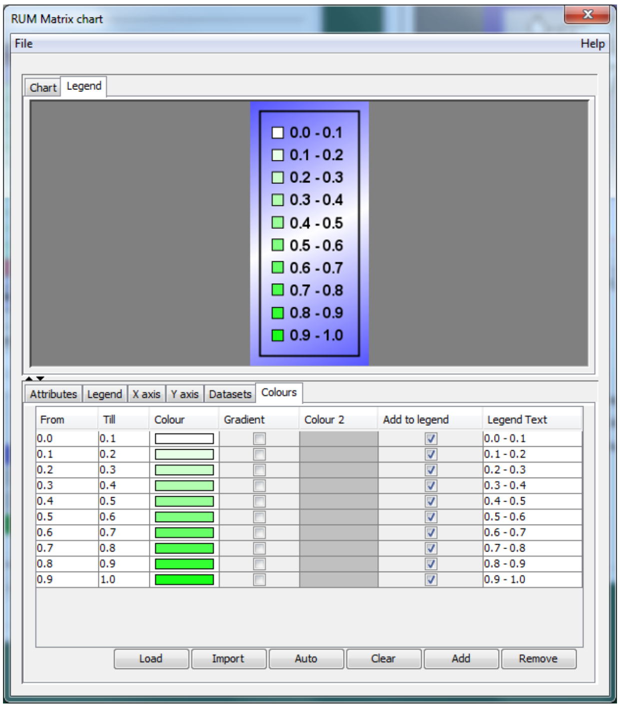


Quick Lool Colours panel Matrix Colours panel – imported from Quick Look

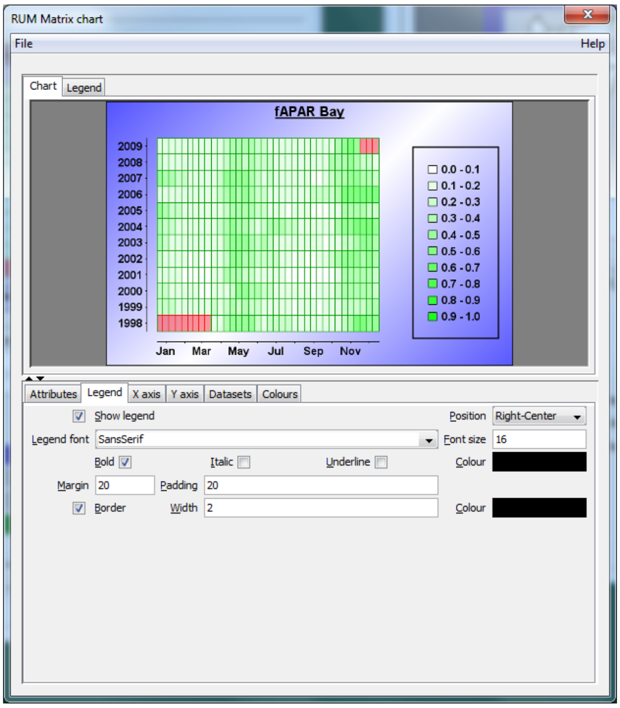
The “Add to legend” options and the “Legend Text” fields of the colour entries determine the contents of the Matrix chart legend.



Matrix Colours panel specifying legend contents



Matrix Legend view



Matrix chart showing legend

4.3.2.5. X-axis

RUM Matrix charts contain a single X-axis. It can be positioned at the top or at the bottom of the chart. Its settings are the same as those of regular Charts.

4.3.2.6. Y-axis

RUM Matrix charts contain a single Y-axis. It can be positioned at the left or at the right side of the chart. Its settings are the same as those of regular Charts. Its values will be the year corresponding to the matix' row.

4.3.3. RUM Matrix charts File menu

A PNG file of a RUM Matrix chart can be created via the Export PNG entry in the File menu.

A RUM Matrix chart itself can be saved / re-opened as a CNM file via the File menu.

Such CNM file contains all chart settings and its datasets properties. It does not contain the actual values from the datasets. Instead each time a CNM file is re-opened, the dataset values are retrieved from the database via the datasets properties. This means that a chart can be updated, after new values have been added to the database, just by re-opening its CNM file.

CNM files can also be used as templates:

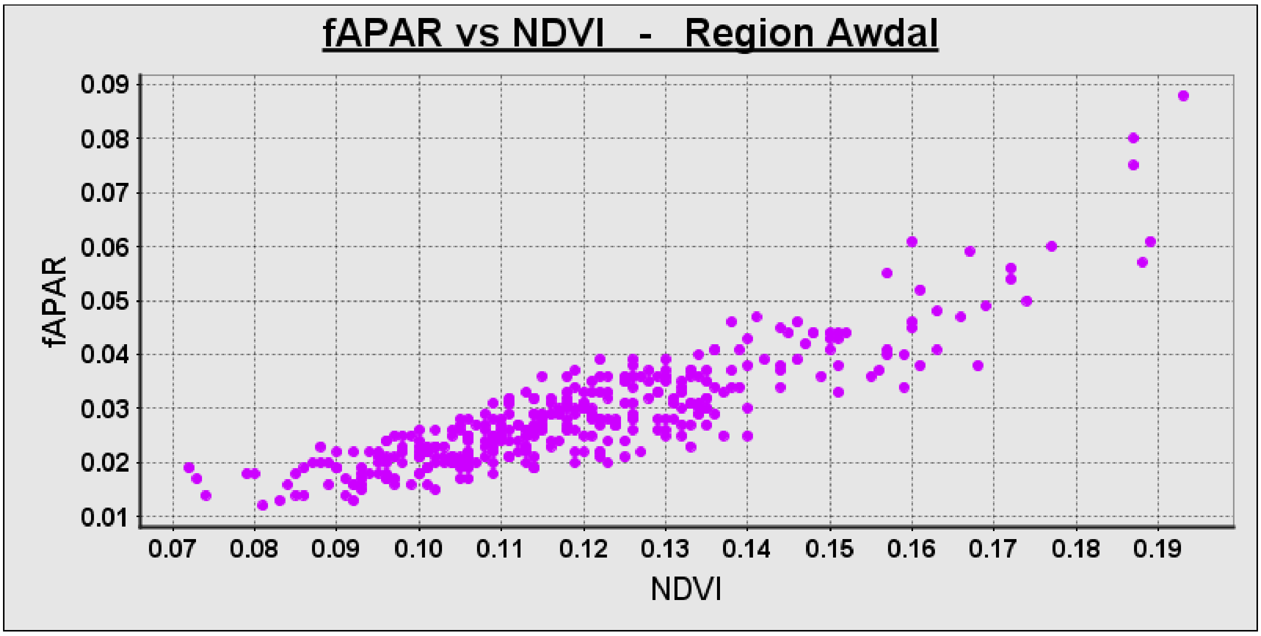
- by the RUM Matrix chart utility itself, when creating new Charts by replacing the datasets in existing charts.
- by the RUM Chart series tool which facilitate the creation of series of similar PNG files, based on an existing charts.

4.4. RUM Scatter charts

4.4.1. Introduction

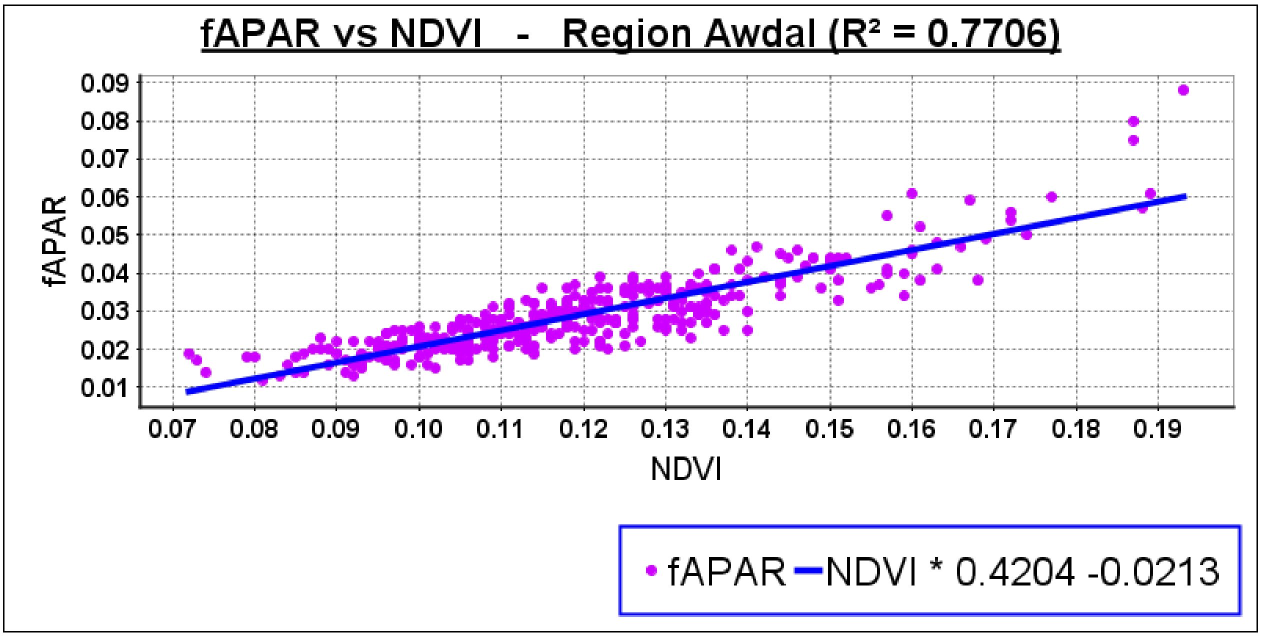
The RUM Scatter chart utility allows to plot RUM datasets against each other in a scatter chart or plot.

A RUM Scatter chart contains at least two datasets. One of these datasets acts as the X-variable (independent variable), while the other datasets act as Y-variables (dependent variables). The chart will contain points (X,Y), where X and Y are the RUM values, with equal time (period), of the X and Y datasets.



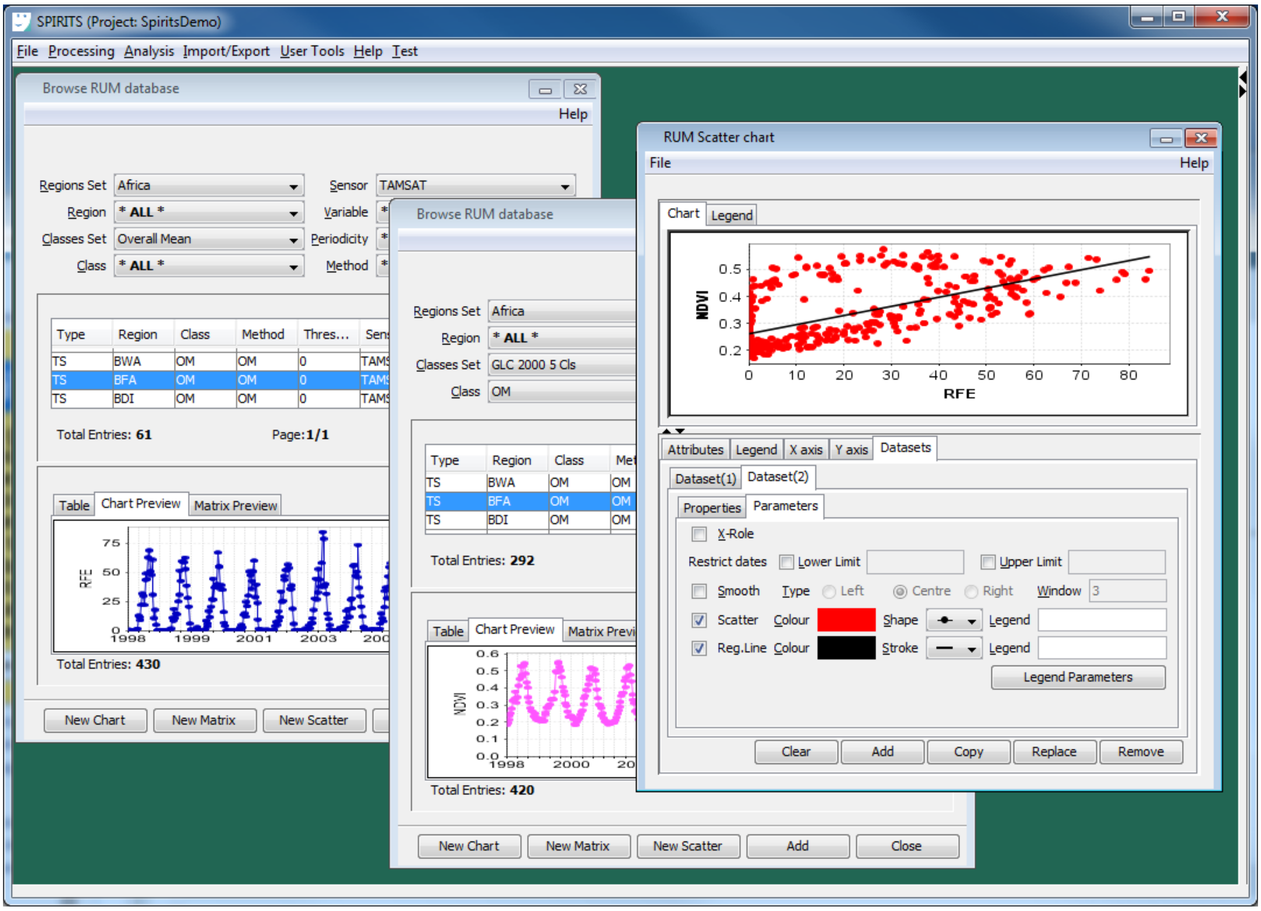
RUM Scatter chart example

Besides the (X,Y) points, a regression line can be plotted (simple linear regression - ordinary least squares). The parameters of the regression line (Intercept and Slope) and the Pearson's correlation coefficient (R-squared) are calculated and can be used in the charts title and legend.

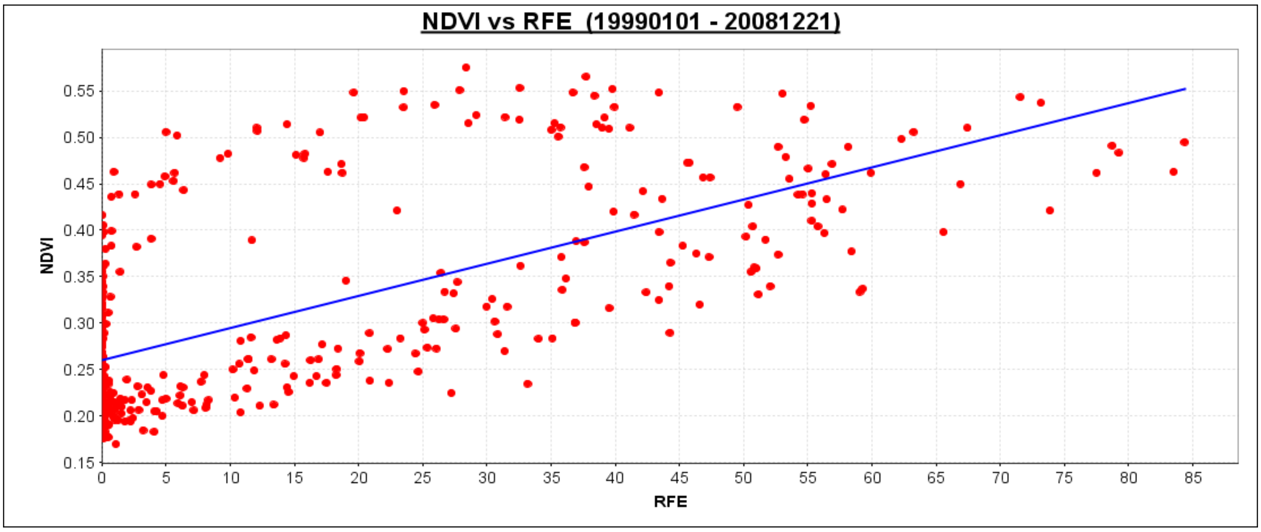


RUM Scatter chart example

Datasets can be sent from the database browser to a Scatter charts Form, or can be selected from a Scatter chart Form directly. Only Time Series datasets can be used in Scatter charts.



Browser :	**New Scatter**	opens a new Scatter chart containing the selected dataset as X dataset;
Browser:	**Add**	adds the selected dataset to the last active chart form as Y dataset;
Matrix chart:	**Add**	opens a Browser to select a dataset to be added to the Scatter chart.



RUM Scatter chart example

4.4.2. RUM Scatter charts Form and Panels

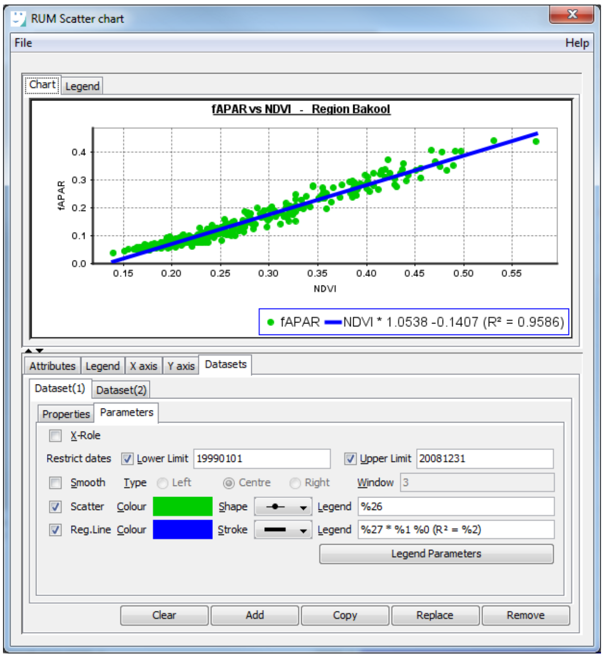
4.4.2.1. Views

Chart view

The main view for RUM Matrix charts is the Chart view panel, which contains the graphical representation of the chart data. By right clicking the panel, its contents can be copied or saved as a PNG image. A PNG file of the chart can be also created via the Export PNG entry in the File menu.

Legend view

The Legend view panel contains only the charts legend. This can be used to obtain the legend information separately in case it would occupy too much space on the chart itself. By right clicking the panel, its contents can be copied or saved as a PNG image.



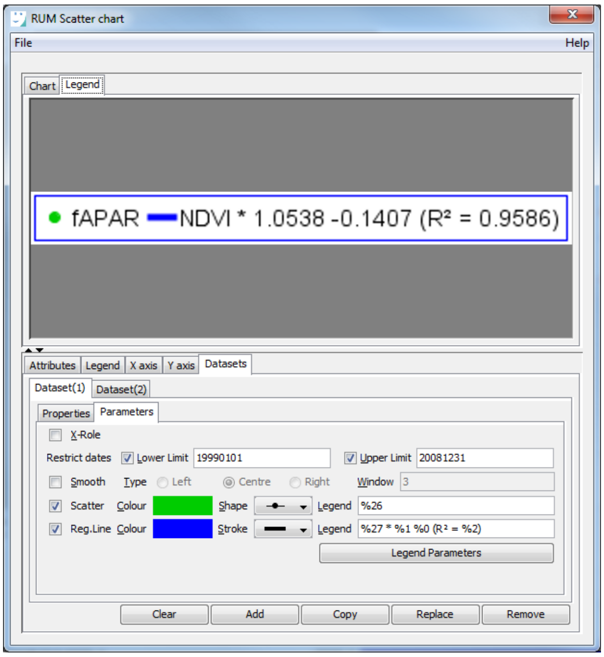


Chart view Legend view

4.4.2.2. Attributes

As with regular Charts, the chart title, background, gridlines and size can be specified in the attributes panel. The parameters available for the chart title will be those from the dataset which has been assigned the X-role, and the first dataset with an Y-role (thus from the first 'scatter plot').

4.4.2.3. Legend

A legend can be shown on the chart, but is also available separately in the Legend View panel. The settings regarding the legends look and feel are the same as for regular charts. Its contents however originates from the datasets with an Y-role.

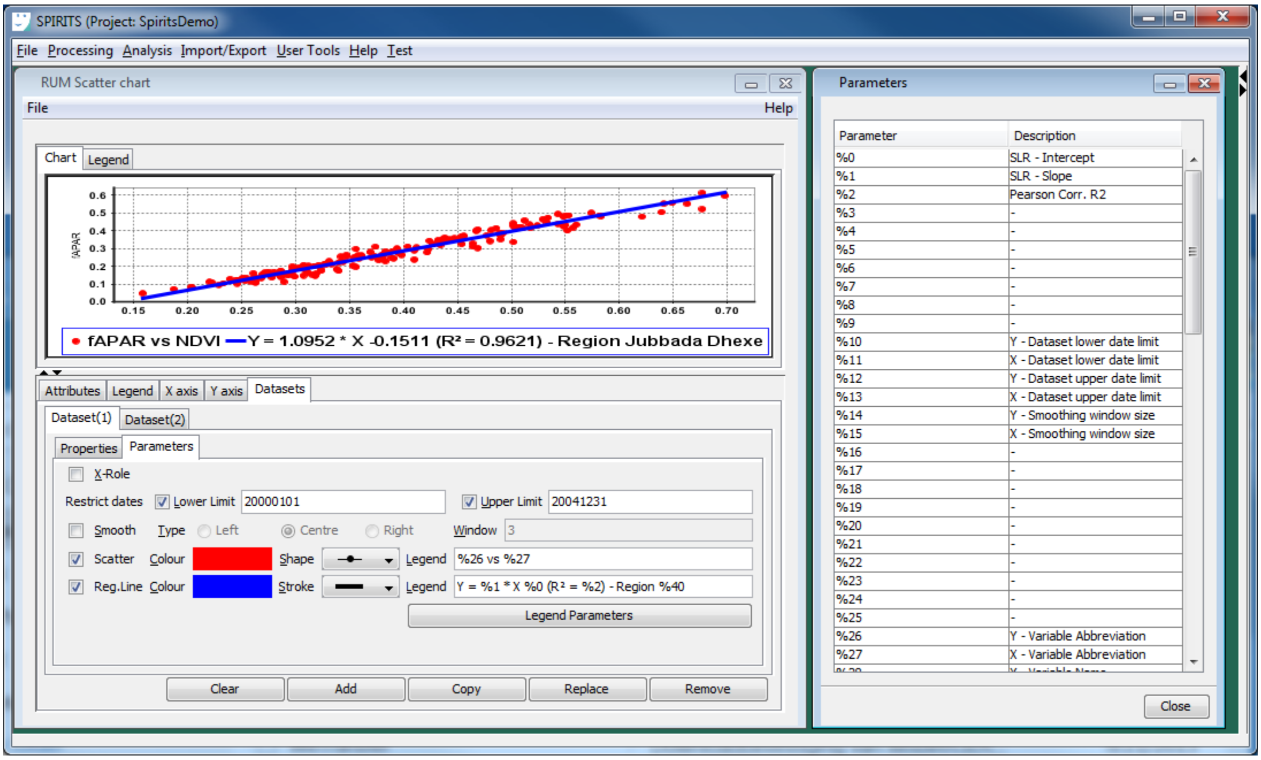
4.4.2.4. X-axis and Y-axis

RUM Scatter charts contain a single X-axis and one or more Y-axis. They can be positioned at the top/left or at the bottom/right of the chart. Their settings are the same as those of the Y-axis in regular Charts. In case the Chart contain multiple Y-datasets which belong to different variables, additional Y-axis will appear.

4.4.2.5. Datasets

Only Time Series datasets can be used in Scatter charts. At least two datasets are required, one of them acting as independent variable (X-role). Each dataset has a “Properties” and a “Parameters” pane. The properties pane shows the dataset information (variable, sensor, region, land use class, periodicity,...). In the parameters pane:

- the role of the dataset can be specified (X or Y). Exactly one dataset is to be assigned the X-role;
- optionally an upper and lower date restriction can be selected and specified;
- optionally a smoothing window type and its size can be specified;
- only for Y-datasets:
  - the scatter plot’s visibility, points shape, shape colour and legend can be specified;
  - the regression line’s visibility, stroke, line colour and legend can be specified.
  - both legends can be parameterized strings: a mixture of constant text and parameters ("%0", "%1,...").
  - The available parameters can be inspected via the Legend Parameters button. Their actual values originate from the properties of the actual (Y) dataset in the panel, and the common X-dataset.



4.4.3. RUM Scatter charts File menu

A PNG file of a RUM Scatter chart can be created via the Export PNG entry in the File menu.

A RUM Matrix chart itself can be saved / re-opened as a CNS file via the File menu.

Such CNS file contains all chart settings and its datasets properties. It does not contain the actual values from the datasets. Instead each time a CNS file is re-opened, the dataset values are retrieved from the database via the datasets properties. This means that a chart can be updated, after new values have been added to the database, just by re-opening its CNS file.

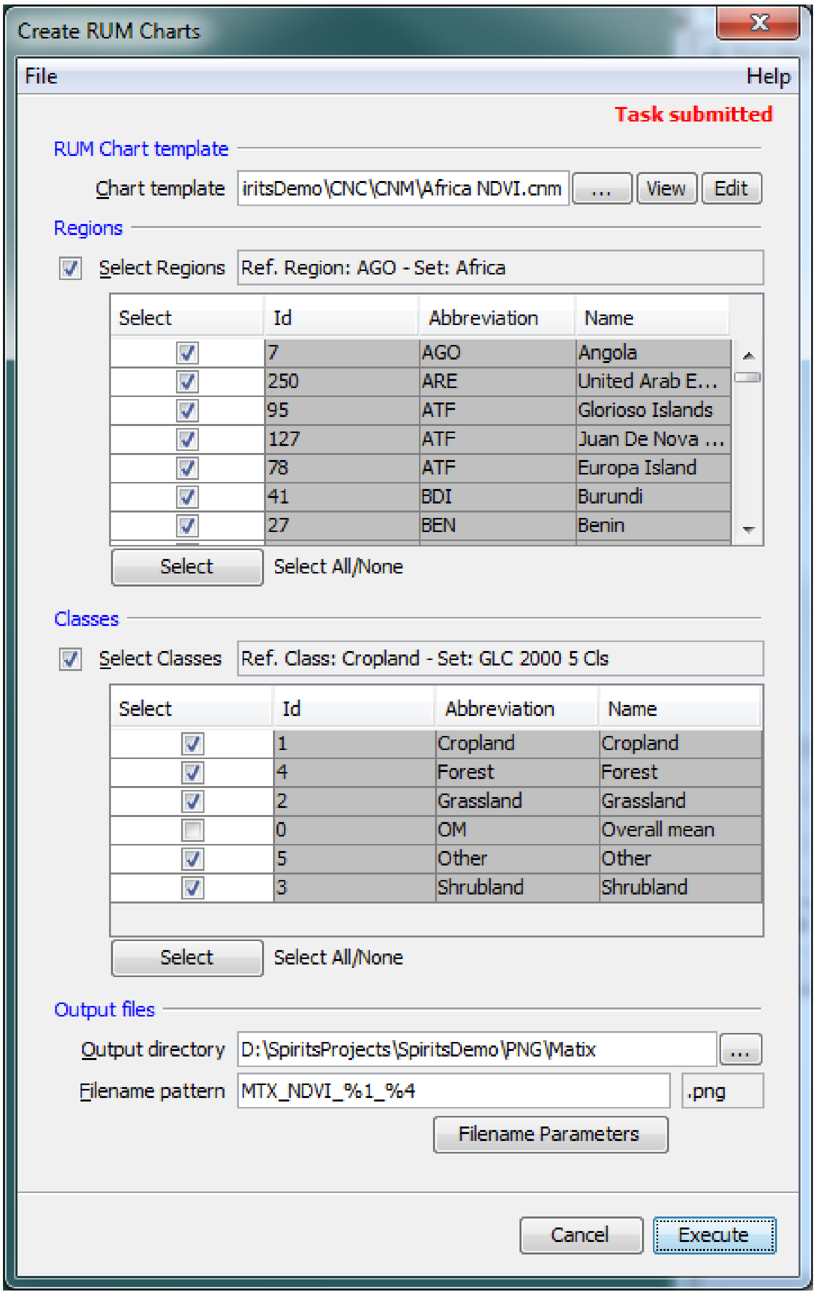
CNS files can also be used as templates:

- by the RUM Scatter chart utility itself, when creating new Charts by replacing datasets in existing charts.
- by the RUM Chart series tool which creates series of similar PNG files, based on an existing charts.

4.5. RUM Chart series tool

The RUM Chart series tool facilitates the creation of series of similar PNG files of charts, based on an existing RUM Chart.

The tool starts from an existing saved RUM chart (CNC, CNM or CNS file) which is used as a template for the PNG's to be created. Next, the regions and/or classes for which a PNG has to be created, can be selected.



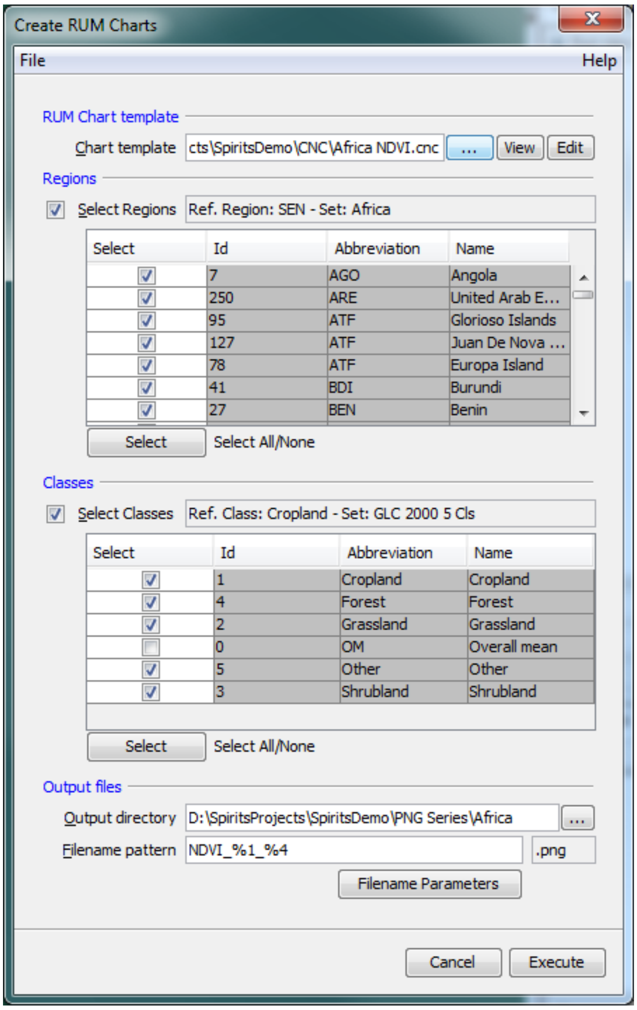
Since a RUM chart (and thus its CNC/CNM/CNS file) can address multiple Datasets, the first Dataset in the chart will function as 'reference', when using the chart as template. This dataset determines the 'reference-region' and the 'reference-class', which will be substituted by the selected regions and classes during the creation of the PNG's.

Selection of regions and classes

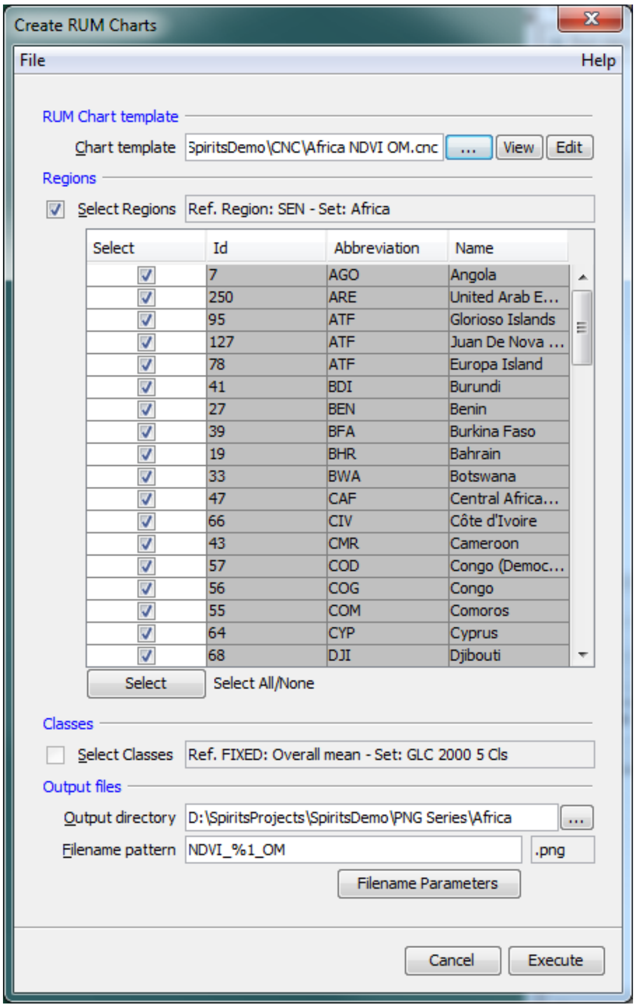
The available regions to choose from, are those regions found in the database, belonging to the same regions-set as the regions-set of the first dataset in the template file.

The available classes to choose from, are those classes found in the database, belonging to the same classes-set as the classes-set of the first dataset in the template file.

In case the first dataset in the chart has "Overall Mean" as its unmixing method, no classes can be chosen, only the "Overall Mean" proxy class (id = 0) is available.



reference dataset in the template chart contains a "normal" class



reference dataset in the template chart is an "overall mean" dataset

Output files

The names for the output files will be specified by means of an output directory and a parameterized string: a mixture of constant text and parameters. Available filename parameters are the properties of the regions and classes (Id, Abbreviation and Name). These can be viewed via the 'Filename Parameters' button.

Since the region and class Id's are unique, it is recommended to use these in the file names pattern. Example: "MyChart_Reg(%0)_Cls(%3)", %0 and %3 representing the regions Id and classes Id.

Created PNG's

For charts containing a single dataset, PNG's will be created by replacing the (region X class) in this dataset with each (selected-region X selected-class) combination.

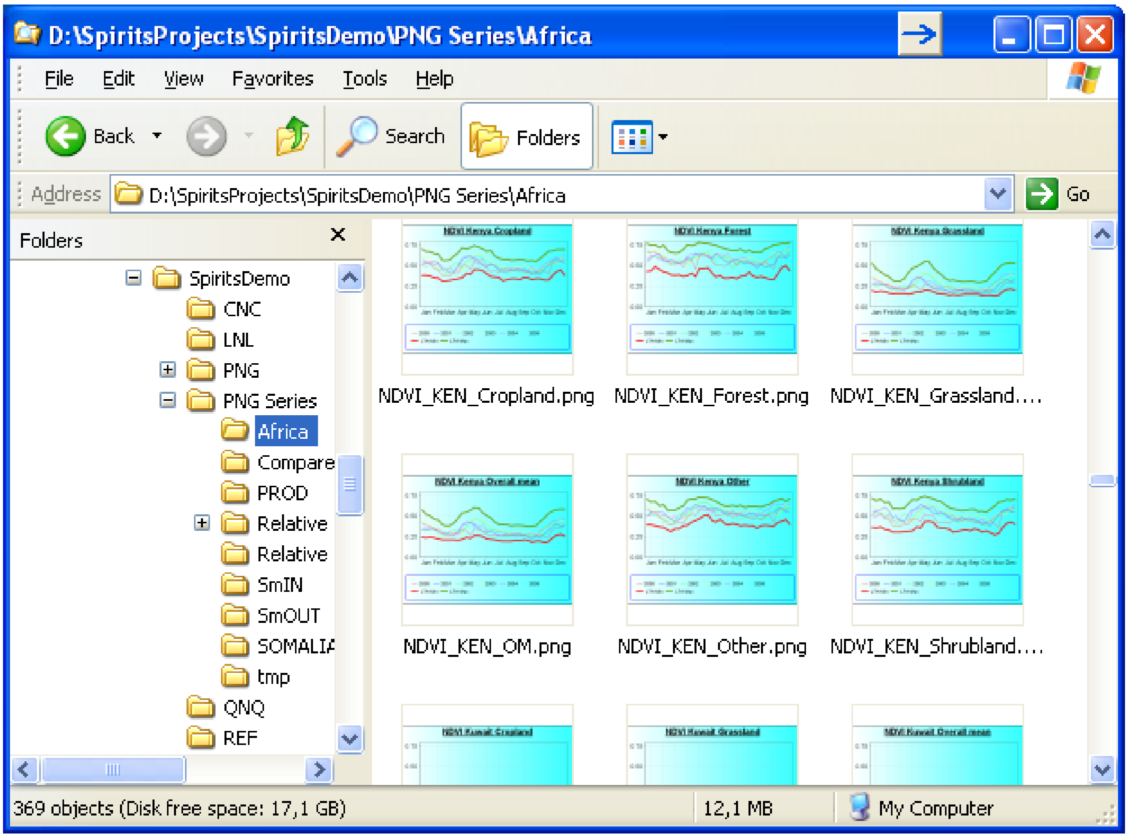
In case of 'multi-dataset' charts, the situation is far less intuitive.

As stated before, the first dataset in the chart will function as 'reference'. This dataset determines the 'reference-region' and the 'reference-class'.

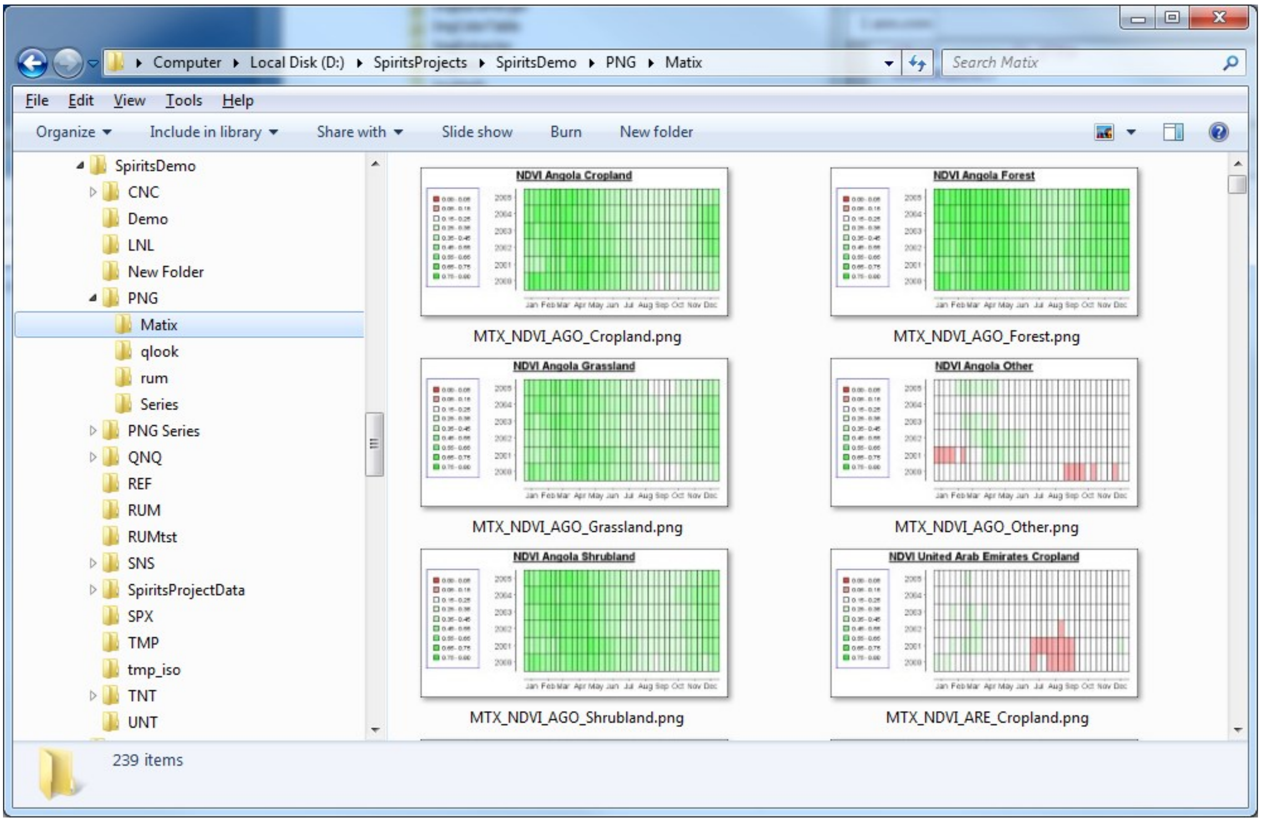
A loop will be performed over all combinations (selected-region X selected-class).

With each (selected-region X selected-class) combination, a chart will be instantiated where:

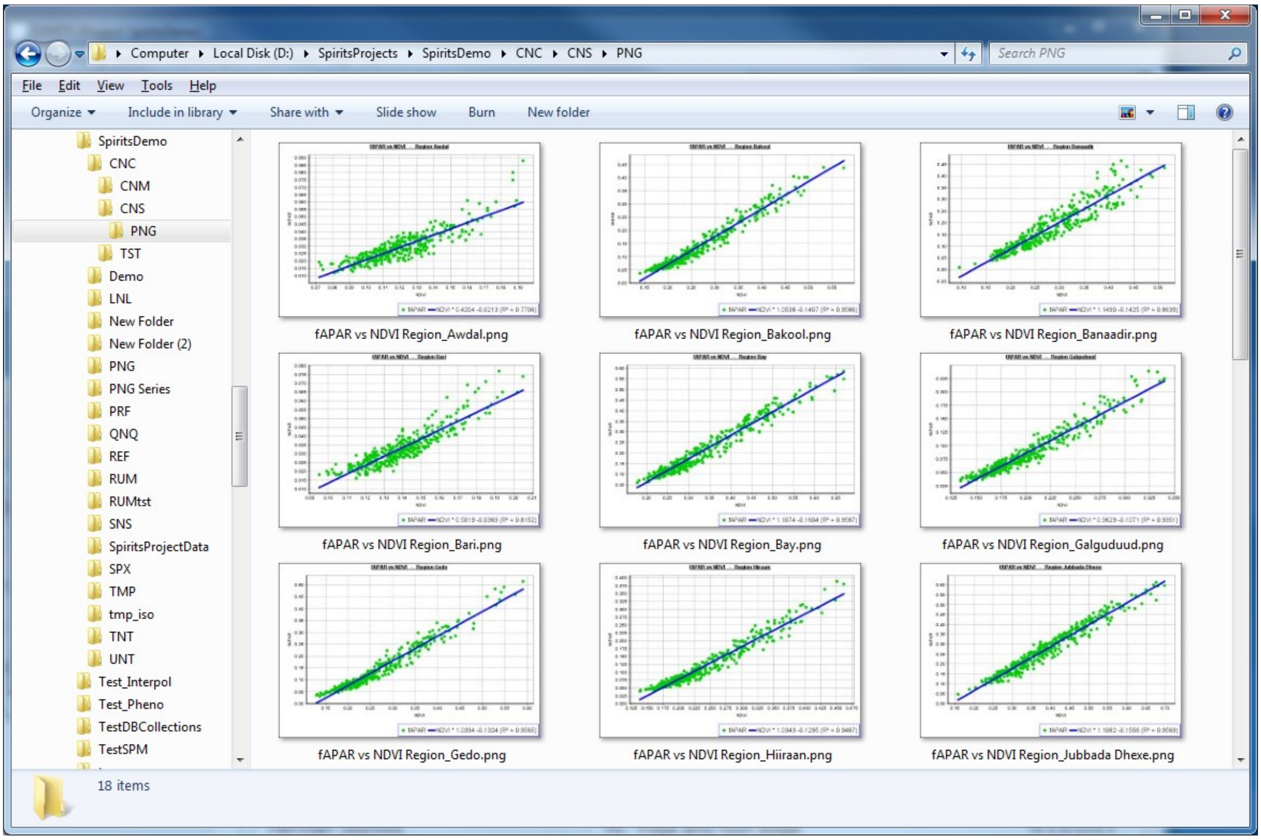
> for each Dataset in the chart, which has the same region as the reference-region, this region will be replaced by the selected-region;
>
> for each Dataset in the chart, which has the same class as the reference-class, this class will be replaced by the selected-class;
>
> Datasets in the chart which neither have their region nor their class corresponding to the reference dataset, are kept as they are;
>
> With this chart instance, the database will be queried to obtain the RUM values for each of the datasets.
>
> > In case none of the datasets in the chart can retrieve any values, no PNG will be created for this (selected-region X selected-class) combination. An error will be reported instead.
> >
> > Otherwise, the chart will export its PNG.



example RUM Chart series - created PNG's



example RUM Matrix chart series - created PNG's

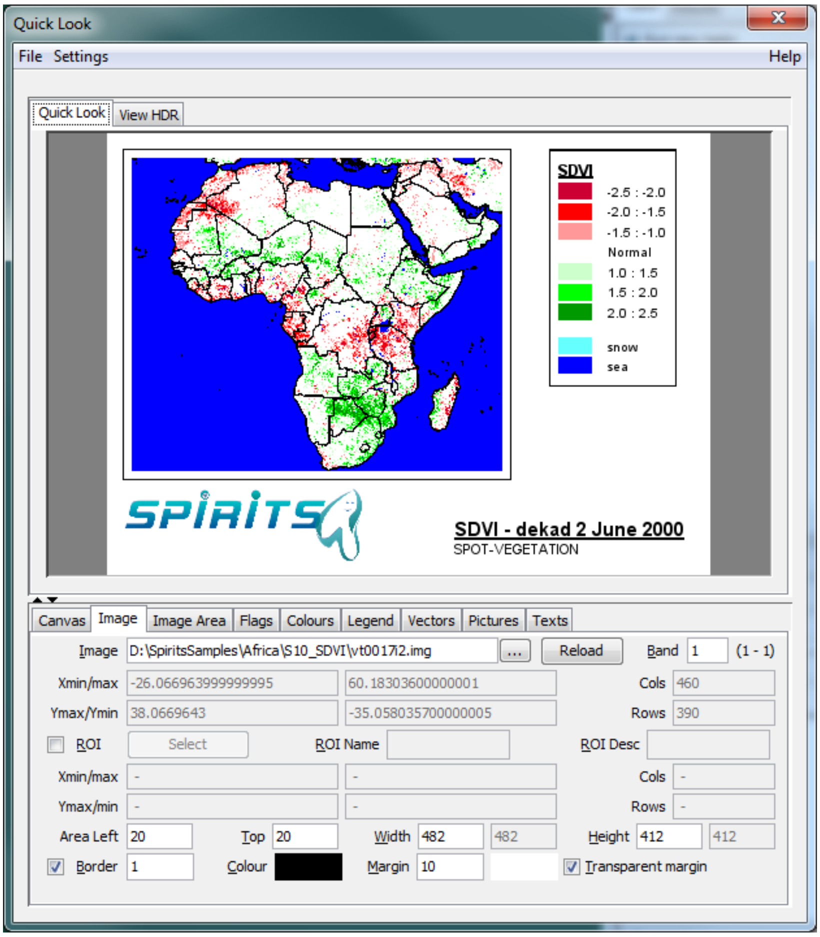


example RUM Scatter chart series - created PNG's

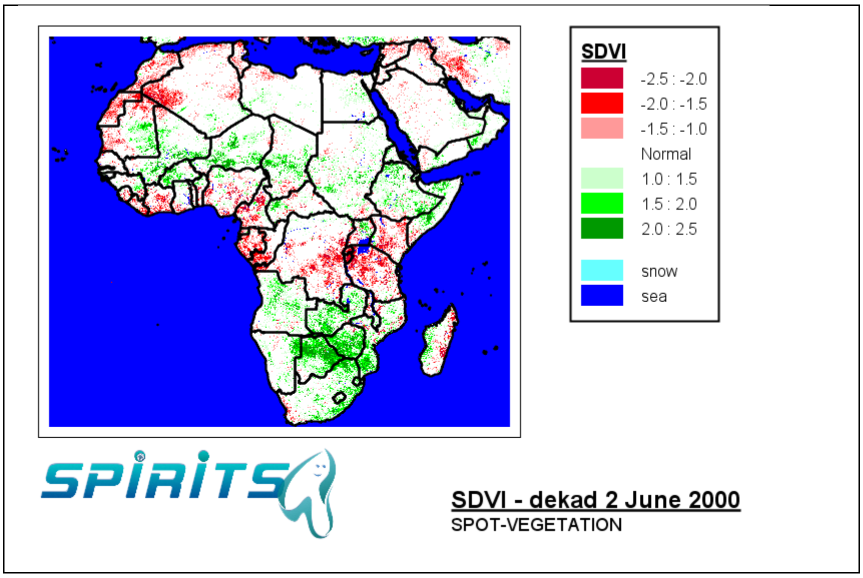
5. Quick Looks

5.1. Quick Look

The Quick Looks utility enables the visualisation of an IMG, and overlay it with vector layers, a legend, pictures (e.g. graphical picture files in JPG, PNG or GIF format, typically logo's), and texts. The resulting Quick Look can be exported as a PNG file.



Quick Look example: Standardized Difference Vegetation Index Africa

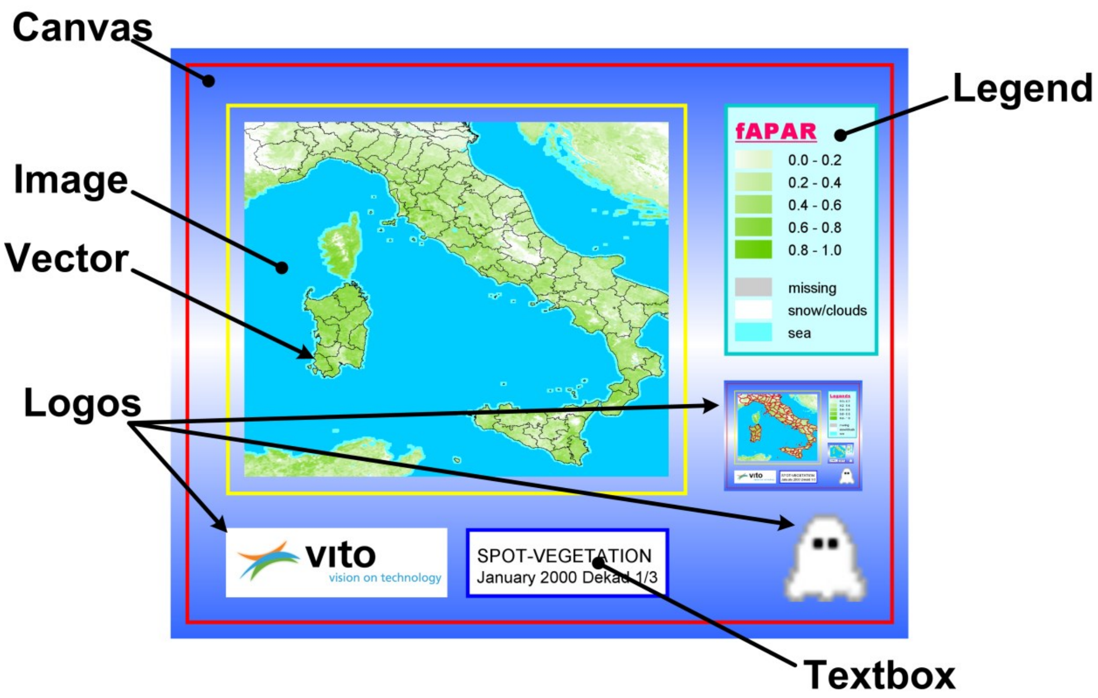


exported PNG

5.1.1. Structure of a Quick Look

The main components of a Quick Look are:

- The canvas, on which the other Quick Look components are drawn;
- The image, the IMG file (or an ROI of this IMG) which will be rendered;
  - A colour table specifying how to render the flags found in the IMG file;
  - A colour table specifying how to render the data values found in the IMG file;
- An optional legend.
- An optional collection of vector files which will be overlay the rendered image;
- An optional collection of pictures (e.g. logo's);
- An optional collection of textboxes containing lines of text;



Quick Look structure

5.1.2. Quick Look Form and Panels

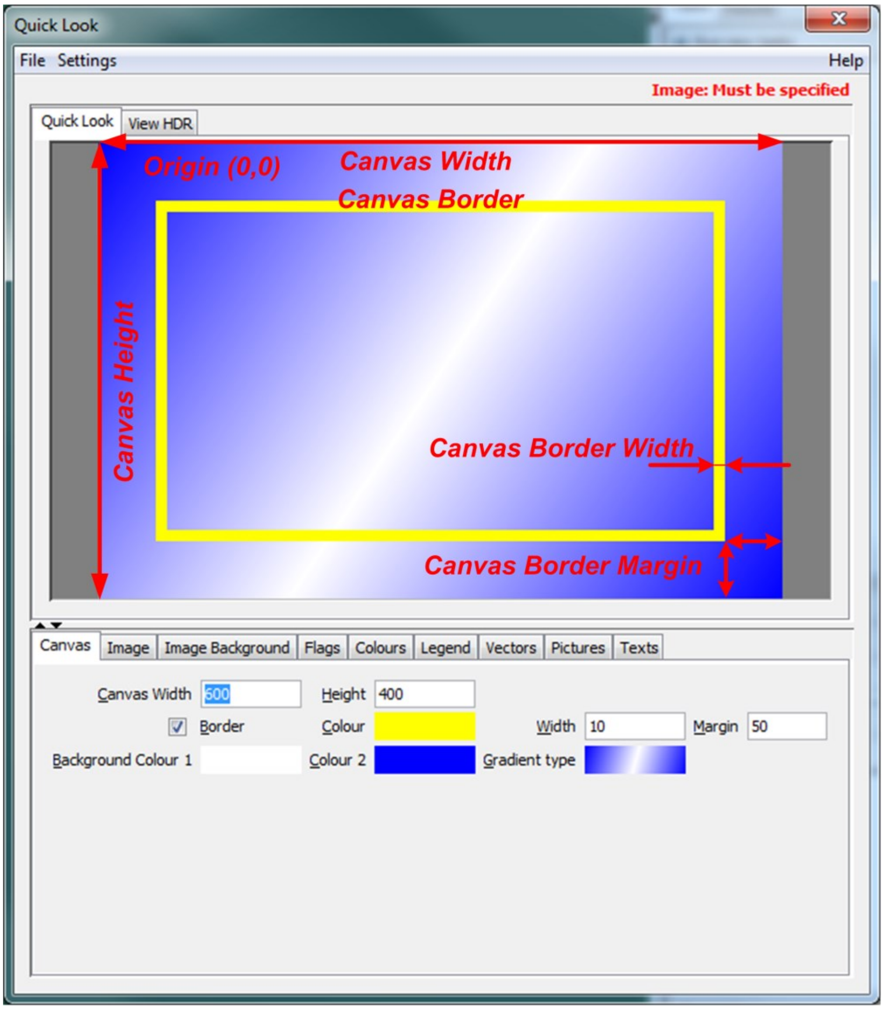
5.1.2.1. Canvas

The canvas specifies the absolute dimension (Width x Height), in pixels, of the resulting PNG file of the Quick Look.

The canvas also specifies the overall background of the Quick Look. This background can be a solid (single) colour or a gradient.

Optionally a canvas border can be specified, parameterized by its colour, its width, and the distance to the edge of the canvas (margin).

All Quick Look components which can be positioned (image, legend, logo(s) and textbox(es) use the upper left corner of the canvas as the origin of the coordinates system. Their positions and sizes will always be expressed in pixels.



5.1.2.2. Image

In the Image tab, the IMG file to be visualized is specified. In case of a multi-band image, one can also select the requested band. Optionally one can choose to restrict the visualization to a selected ROI of the IMG.

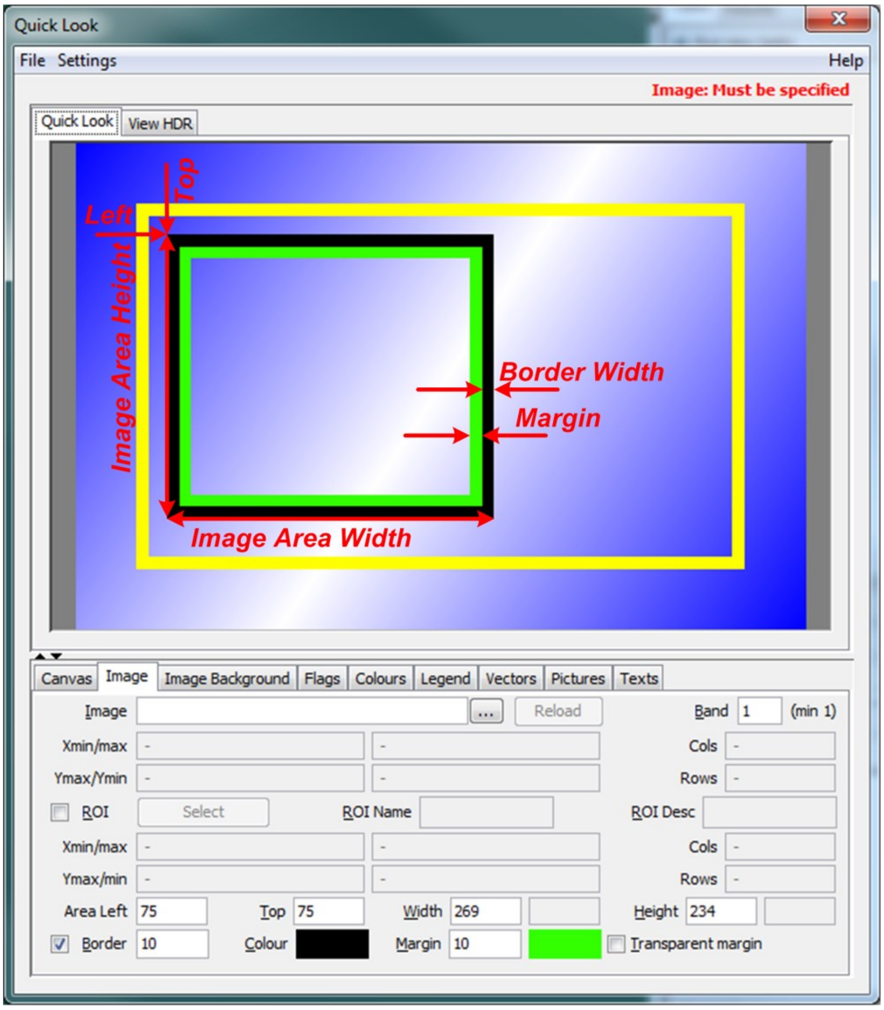
The image itself will be rendered in the "image area". This area consists of

- an optional border around the actual rendered image;
- an optional margin between this border and the actual rendered image;
- the actual rendered image itself.

The "Left" and "Top" parameters specify the position of the upper left corner of the image area with respect to the canvas origin.

The "Width" and "Height" parameters specify the dimension of the image area. "Width" and "Height" will adapt automatically so that the aspect ratio (Samples/Lines from the HDR file) of the image is kept intact.

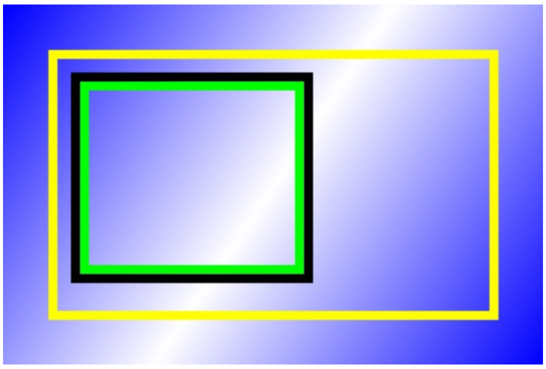
Optionally an image area border can be specified, parameterized by its colour, its width, and the distance between the border and the edge of the actual rendered image (margin). This margin itself can be transparent or filled with a solid colour.



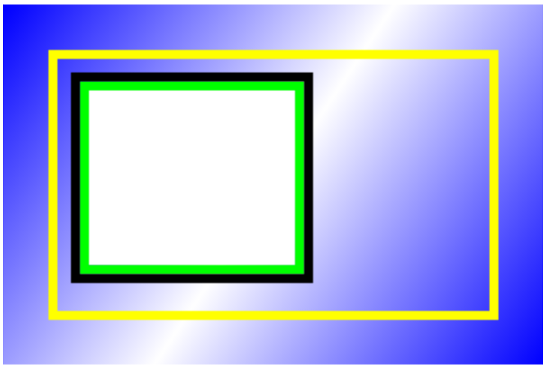
5.1.2.3. Image Background

The background of the image area can be chosen:

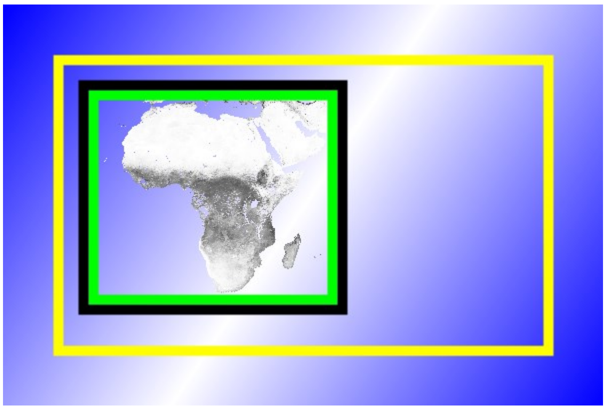
- transparent (meaning the background of the image area will be that of the canvas);
- solid (a single colour);
- a "colour scale" representation of the image. The range and colours can be chosen. In case the IMG values fall outside the selected range (From/Till values), the canvas background shines through.



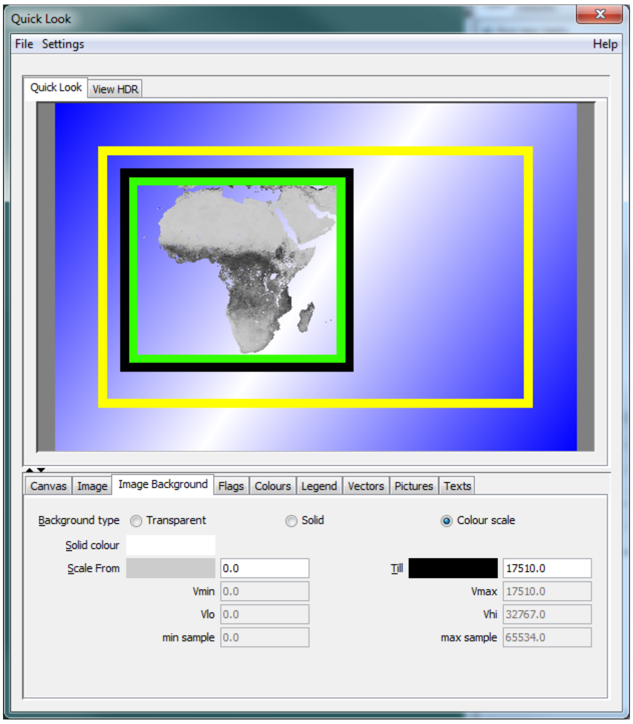
Transparent background



Solid background



Colour scale background:
the canvas background shines through where the IMG values fall outside the specified From/Till range



As an aid in choosing a colour scale range, the panel indicates:

- Vmin/max: lowest/highest digital values of significant range (if indicated in the HDR).
- Vlo/Vhi: lowest/highest digital values occurring in this IMG (if indicated in the HDR).
- min/max sample: lowest/highest digital values actually sampled from the IMG.

5.1.2.4. Flags

In the Flags tab, colours can be assigned to flag values.

A flag value is to be specified as "digital value", meaning the value as-is in the IMG file.

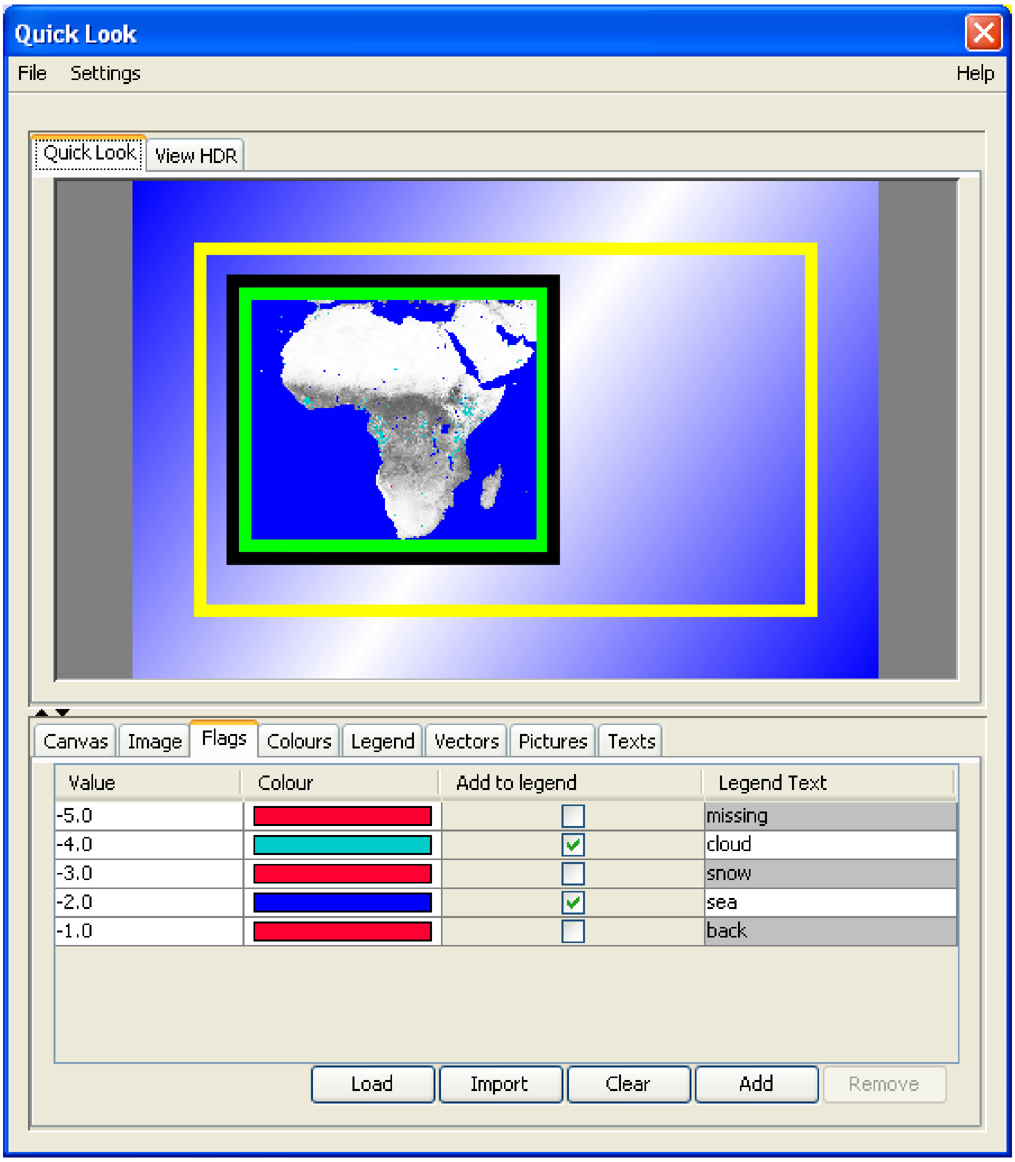
A flag value is a single value, meaning each individual flag value has to be specified separately, there are no "ranges" for flags. (Ranges can be specified in the Colours tab.)

In case the IMG file has been specified in the Image tab, and its HDR file contains the "flags" keyword, the flags can also be imported directly from the HDR file (value + description). During this import, the flag values and descriptions are imported and random colours are assigned.

```
flags = {251=missing, 252=cloud, 253=snow, 254=sea, 255=back }
```

```
flags = {-5=missing, -4=cloud, -3=snow, -2=sea, -1=back }
```

Example: typical flags entries in HDR files



Flags example: imported from HDR, colours manually changed

5.1.2.5. Colours

In the Colours tab, colours can be assigned to the IMG values.

Colours can be assigned to a range of values(From value/Till value) or to a single value(by choosing the From value equal to the Till value). Ranges would typically be used for ordinal images, while single values would be used for categorical images (classifications).

Values are to be specified as "physical values". The physical value is calculated, based on the "digital values" (value as-is in the IMG file) and on the (GLIMPS specific) "values" entry in the HDR file.

```
Physical value = Vintercept + Vslope * Digital value
```

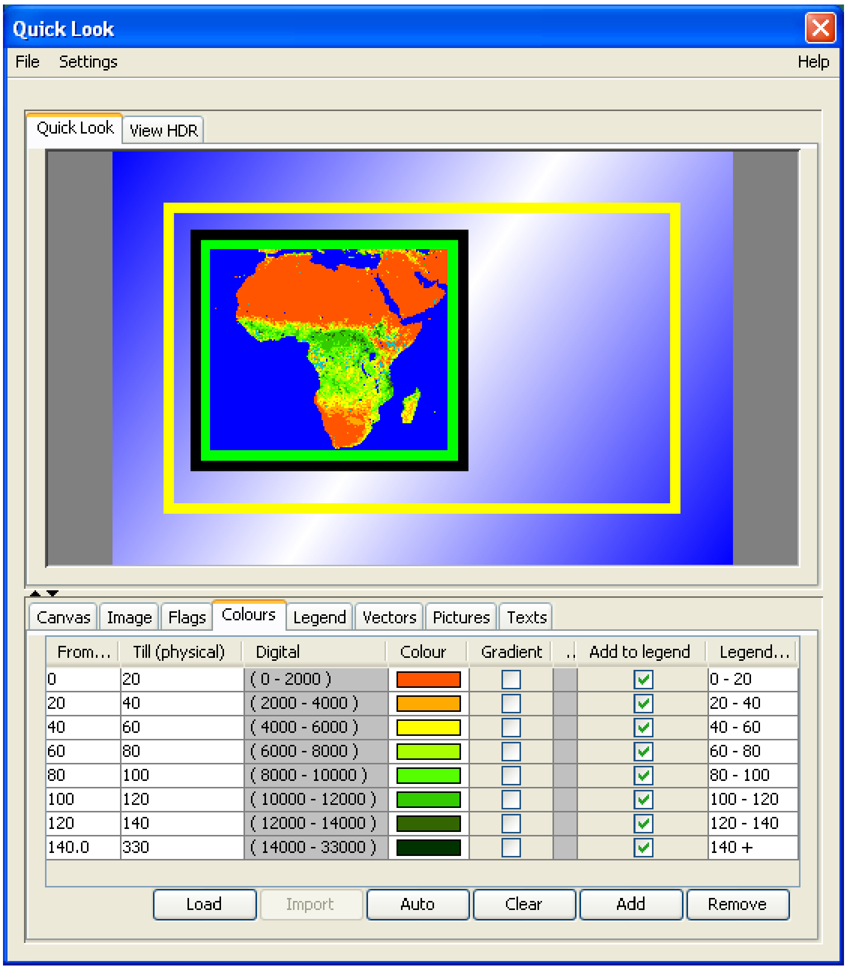
Physical value = F(Digital value, Vintercept, Vslope)

In case the "values" entry is not present in the HDR file, Vint and Vslo default to 0 and 1 respectively, thereby making the physical values equal to the digital values.

```
values = {SDVI[NDVI-toc], -, 0, 250, 0, 250, -2.5, 0.02}
```

```
values = {DMP, kgDM/ha/day, 0, 32767, 0, 17510, 0, 0.01}
```

Example: typical values entries in HDR files: values = { Vname, Vunit, Vlo, Vhi, Vmin, Vmax, Vint, Vslo }

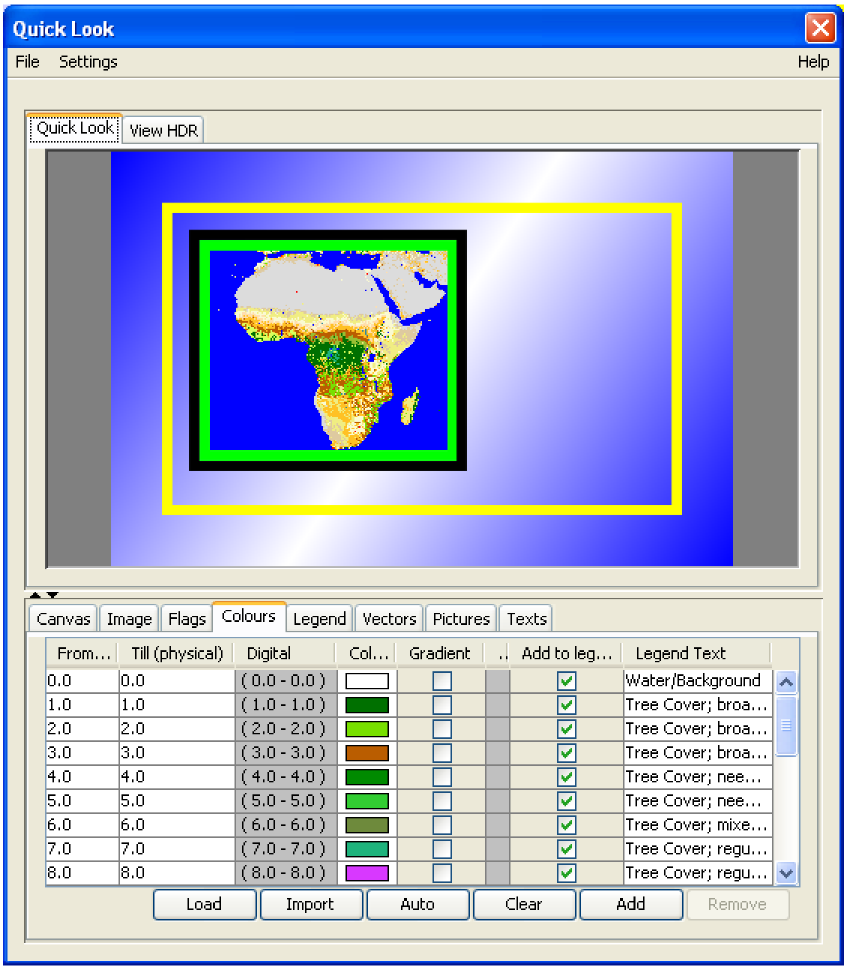


example: colours assigned as ranges of values

In case the IMG file has been specified in the Image tab, and its HDR file contains the "class names" entry, these classes and their colours (RGB sets from the HDR file "class lookup" entry) can also be imported directly from the HDR file. During this import, the classes are assigned single incrementing integer values starting from 0. In case the "class lookup" entry is not present, random colours are assigned.

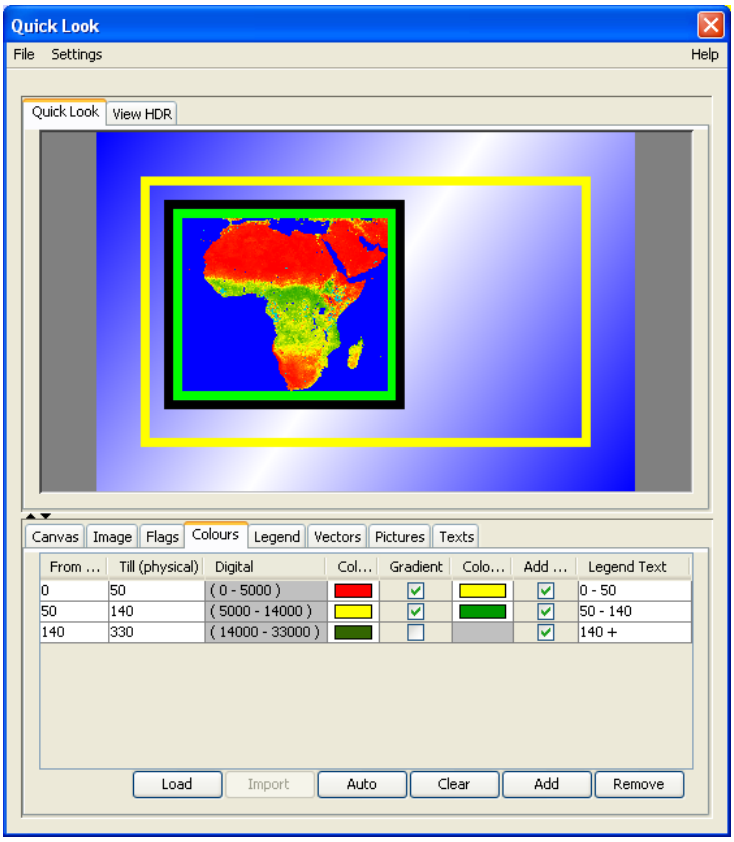
```
class names = {
    Water/Background,
    Tree Cover; broadleaved; evergreen,
    Tree Cover; broadleaved; deciduous; closed,
    Tree Cover; broadleaved; deciduous; open,
    Tree Cover; needle-leaved; evergreen,...
}
class lookup = {
   255,255,255,
     0,112,  0,
   120,224,  0,
   186, 93,  0,
     0,138,  0,...
}
```

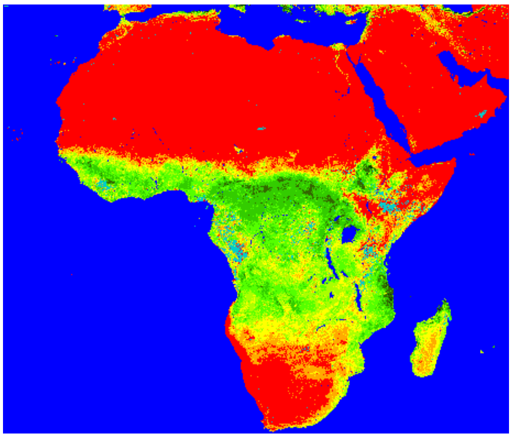
Example: class names and class lookup entries in a HDR file.



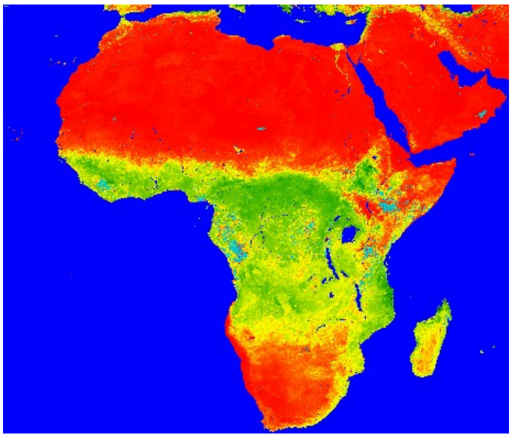
example: colours imported from classification IMG

Ranges can be assigned a solid (single) colour, or a gradient.





8 ranges of values with solid colours



2 ranges of values with gradients

example: gradient colours

In uninterrupted sequences (meaning that the "Till" value of a range is equal to the "From" value of the next range) of values, rages behave like half open intervals. In interrupted sequences of values, they behave like closed intervals.

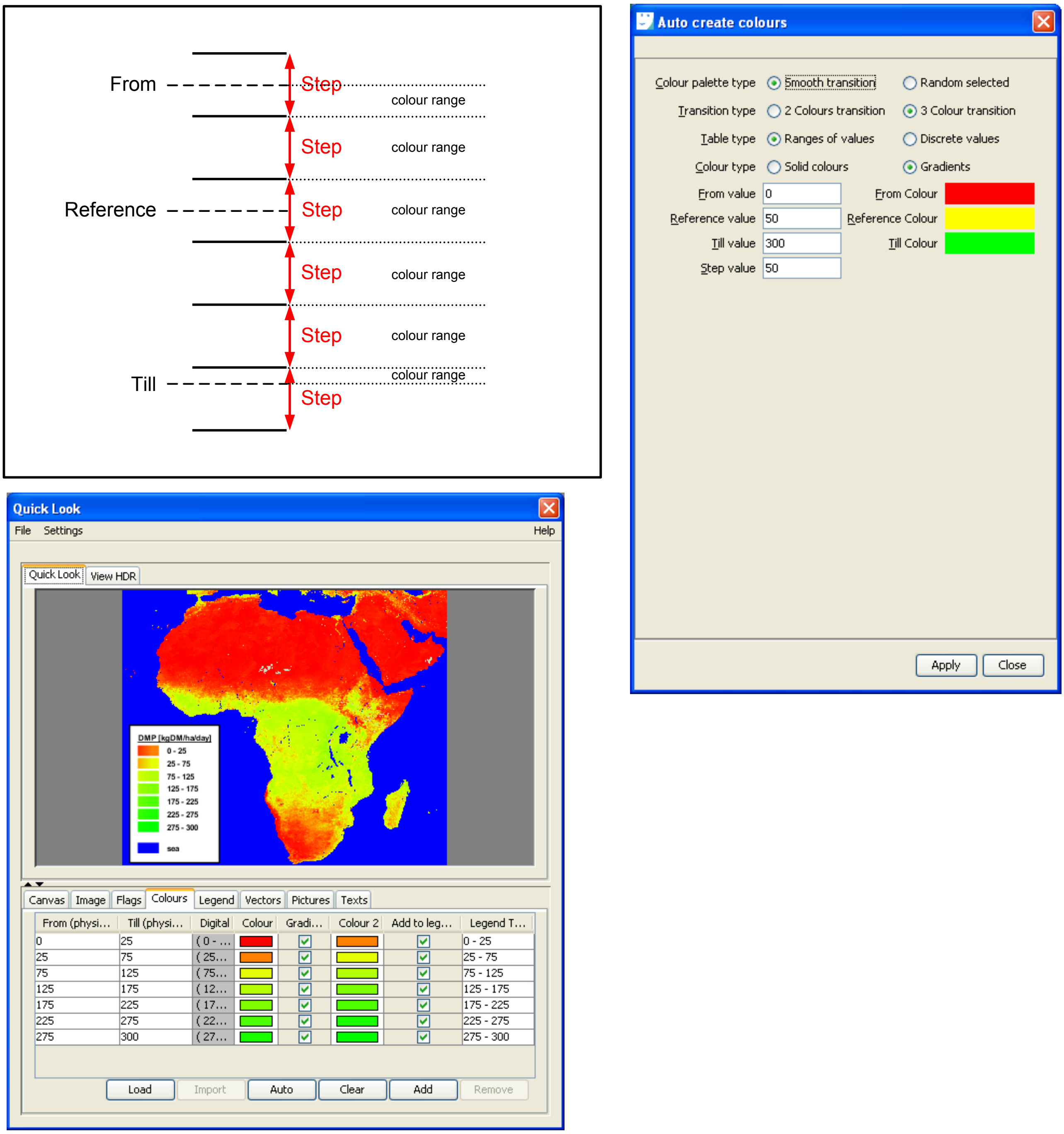
From value	Till value	Behaviour	
0	50	[0 - 50[	50 not included
50	100	[50 - 100[	100 not included
100	150	[100 - 150[	150 not included
150	199	[150 - 199]	199 included
200	249	[200 - 249]	249 included
250	300	[250 - 300]	300 included (last interval is always closed)

ranges: numerical example

As an aid, default colour tables can be build via the "Auto" button. Colour tables can be created for discrete values or ranges of values. The colour palette can be a set of random selected colours, or a transition between colours. In this last case, two or three colour transitions can be chosen, and the colour type can be selected as a fixed colour per range, or a gradient per range, between the ranges.

Two colour transitions are defined by two(value-colour) pairs: From/Till. The generated ranges start "upward" from "From" to "Till".

Three colour transitions are defined by three (value-colour) pairs: From/Reference/Till. The generated ranges start "upward" and "downward" from "Reference" to "From" and "Till".

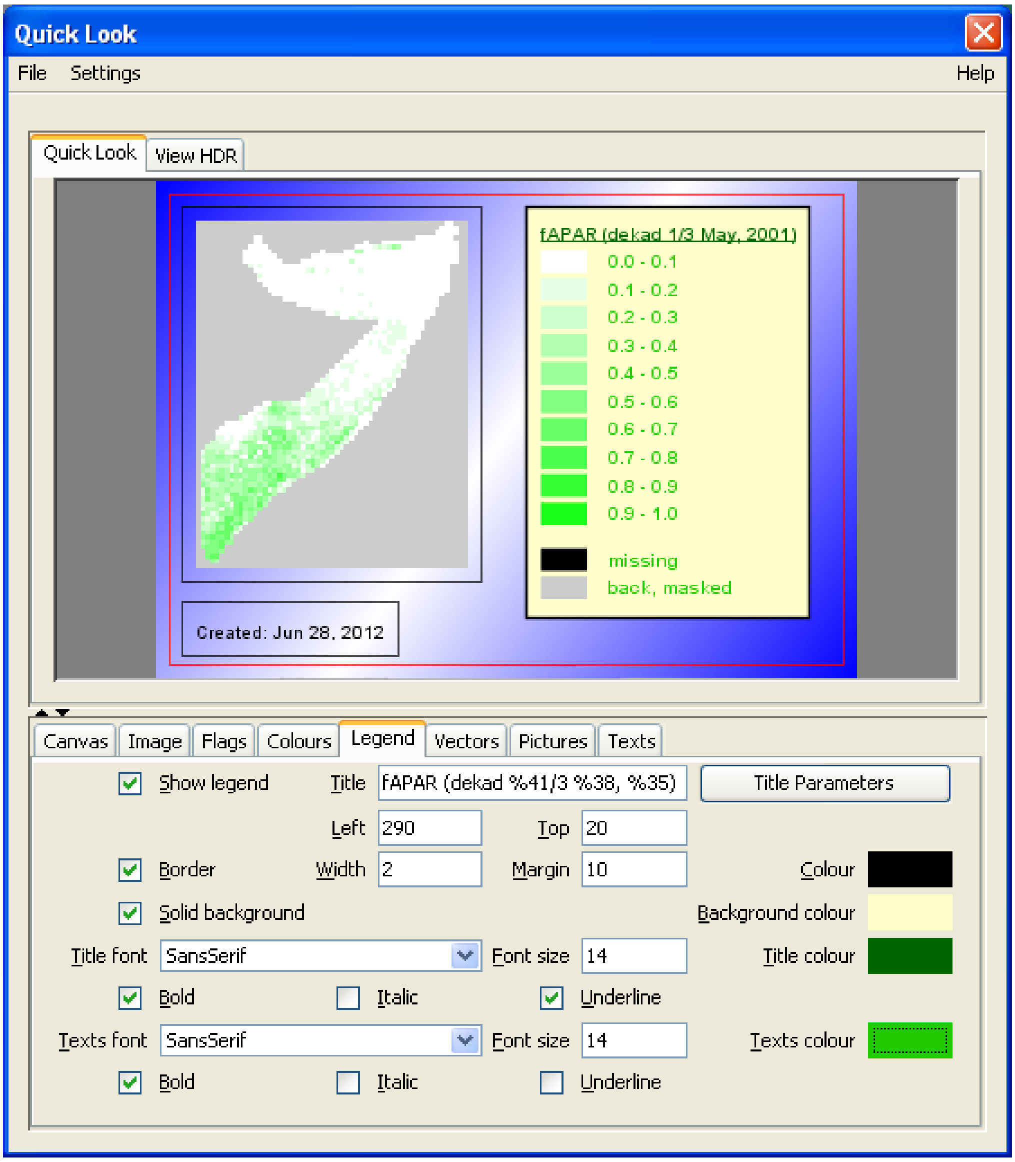


3-colours transition ranges with colour gradients.

As a special case one can specify "0" for the step value, resulting in one single gradient between "From" and "Till" for a two colour transition, and in two gradients respectively between "From" and "Reference" and "Reference" and "Till" for a three colour transition.

5.1.2.6. Legend

Under the Legend tab, one can select whether a legend should be drawn, and specify the overall legend look-and-feel.



Legend example

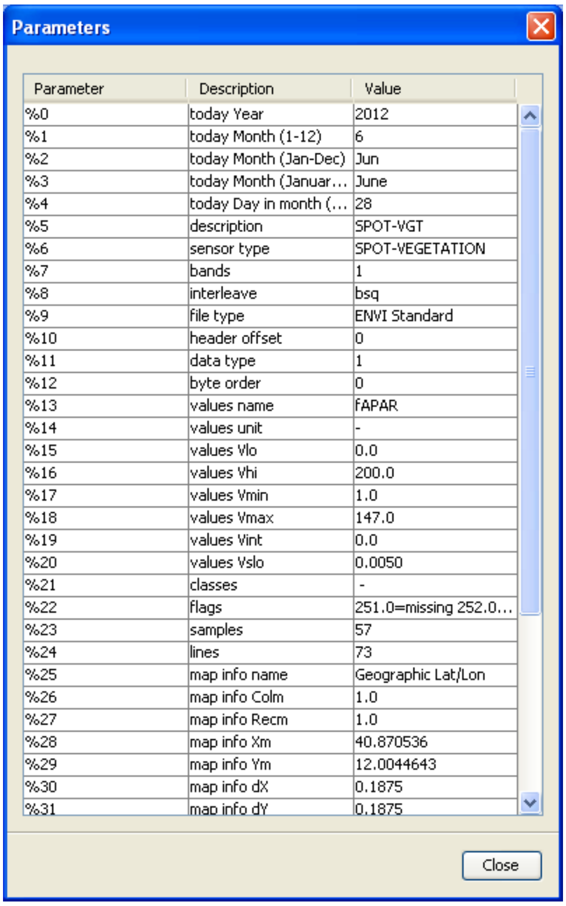
The "Left" and "Top" parameters specify the position of the upper left corner of the legend with respect to the canvas origin.

The width and height of the legend cannot be specified, they are determined by the border, the margin and the content (legend title and entries from Flags and Colours).

Optionally a legend border can be specified, parameterized by its colour, its width, and the distance between the border and its content (margin). The background of the legend can be filled with a solid colour or stay transparent.

The legend title and its entries can have their font, font size, colour and style (bold, italic and underlined) specified.

The legend title supports the use of parameters (%0..%n). The values of these parameters can be previewed via the Title Parameters button. These values originate from the current date, the ROI name and Description and the content of the HDR file in case a (valid) IMG file has been specified in the Image tab.



Parameters

Parameter	Description	Value
%0	today Year	2012
%1	today Month (1-12)	6
%2	today Month (Jan-Dec)	Jun
%3	today Month (Januar…	June
%4	today Day in month (…	28
%5	description	SPOT-VGT
%6	sensor type	SPOT-VEGETATION
%7	bands	1
%8	interleave	bsq
%9	file type	ENVI Standard
%10	header offset	0
%11	data type	1
%12	byte order	0
%13	values name	fAPAR
%14	values unit	-
%15	values Vlo	0.0
%16	values Vhi	200.0
%17	values Vmin	1.0
%18	values Vmax	147.0
%19	values Vint	0.0
%20	values Vslo	0.0050
%21	classes	-
%22	flags	251.0=missing 252.0…
%23	samples	57
%24	lines	73
%25	map info name	Geographic Lat/Lon
%26	map info Colm	1.0
%27	map info Recm	1.0
%28	map info Xm	40.870536
%29	map info Ym	12.0044643
%30	map info dX	0.1875
%31	map info dY	0.1875

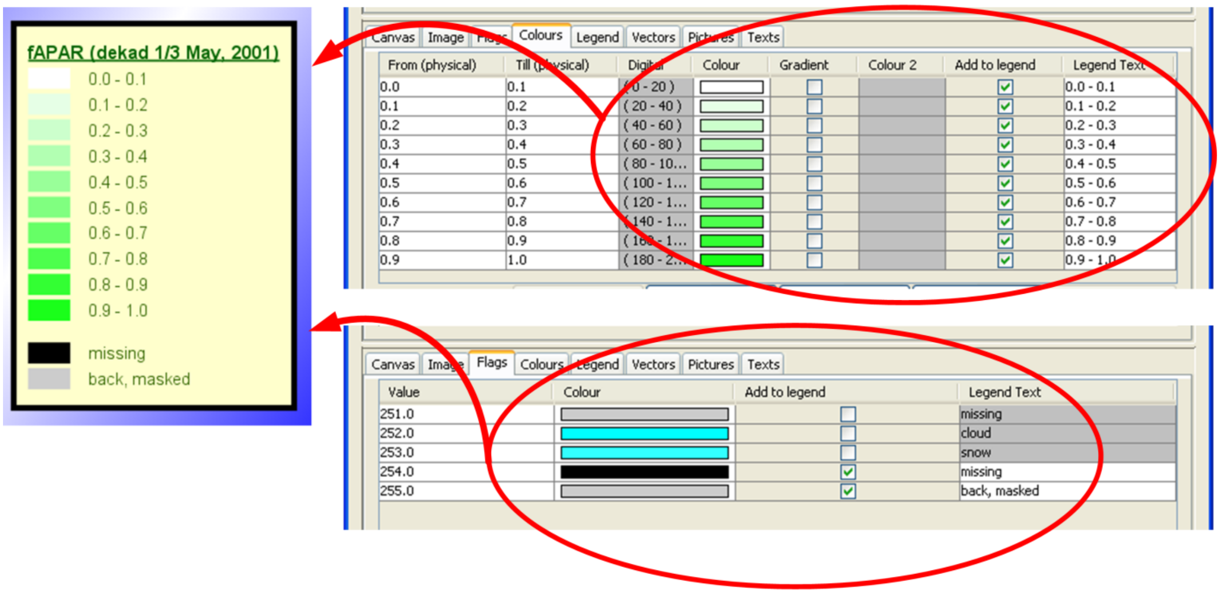
Close

current date info

IMG HDR info

Legend Title Parameters

The actual content of the legend (besides its title) originates from the Flags and Colours tabs. For each entry in these tabs, one can specify whether the entry should occur in the legend, and what the accompanying text should be.

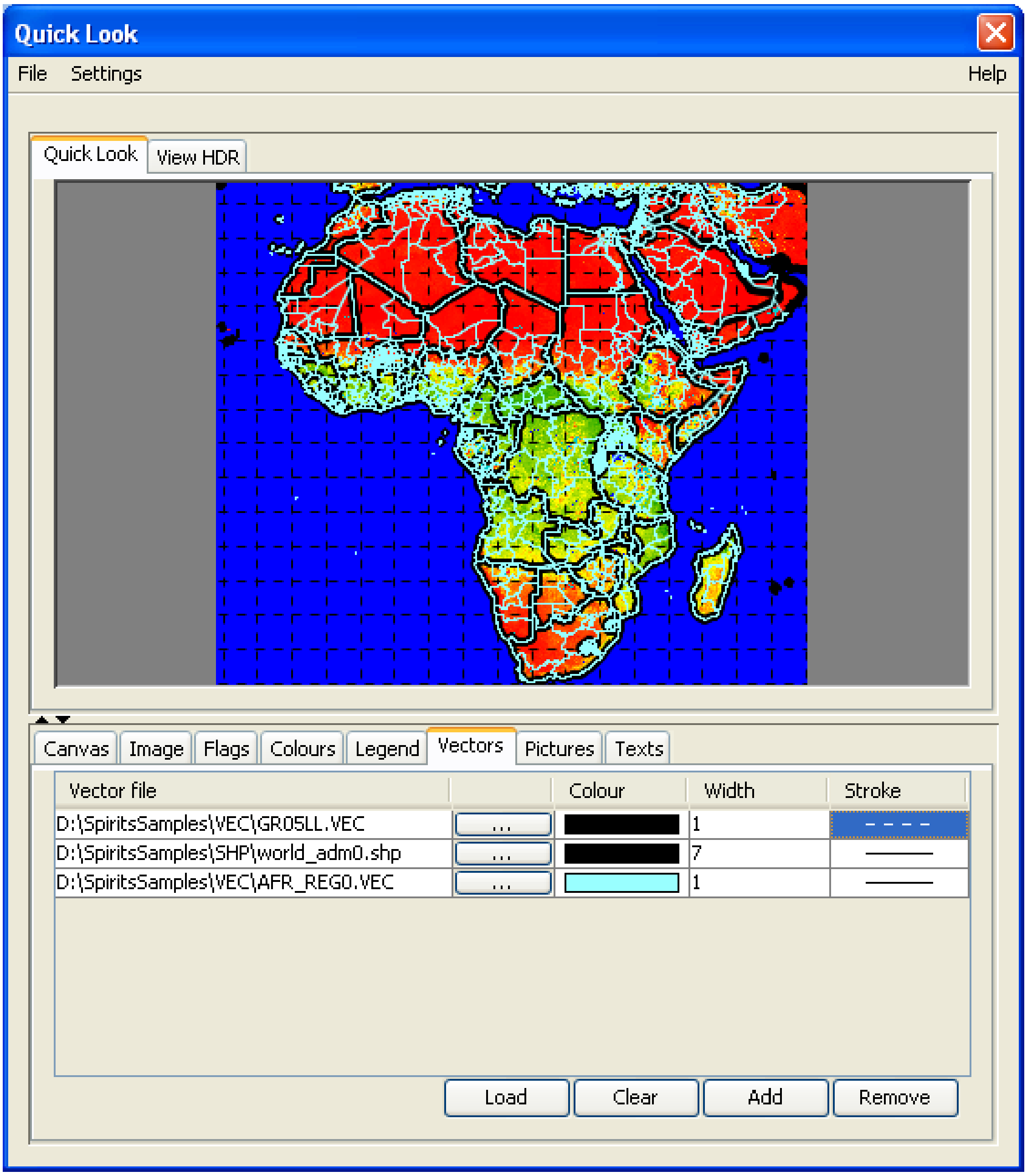


Legend content

5.1.2.7. Vectors

Under the Vectors tab, vector files can be selected to overlay the rendered image.

Limited support is provided for Idrisi VCT, Idrisi VEC and Esri SHP files.



vectors overlay

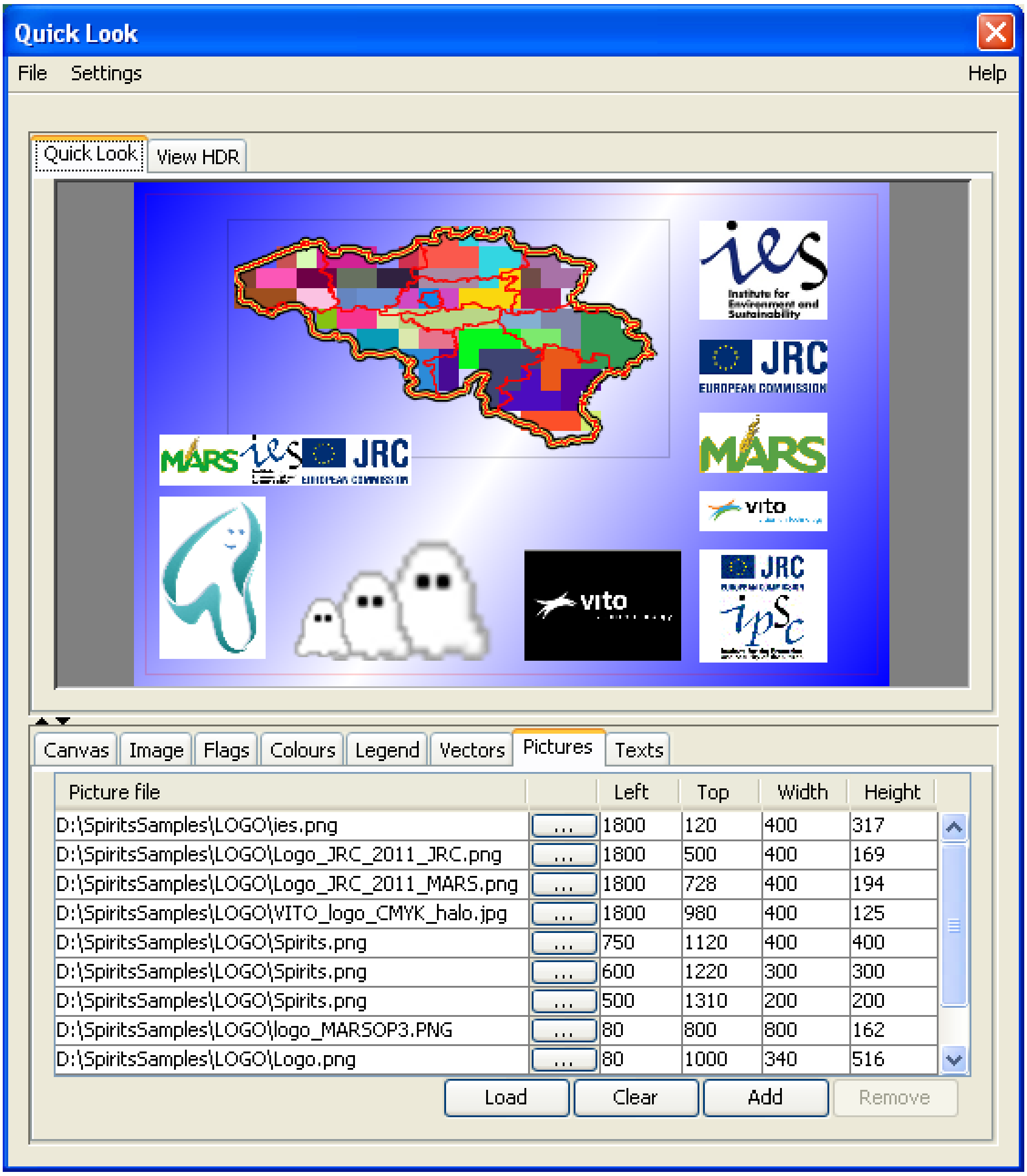
5.1.2.8. Pictures

Under the Pictures tab, graphical picture files (typically logos), can be selected to be added to the Quick Look.

JPG, PNG and GIF file formats are supported.

The "Left" and "Top" parameters specify the position of the upper left corner of the picture with respect to the canvas origin.

The "Width" and "Height" parameters specify the dimension of the picture. "Width" and "Height" will adapt automatically so that the aspect ratio of the picture is kept intact.



5.1.2.9. Texts

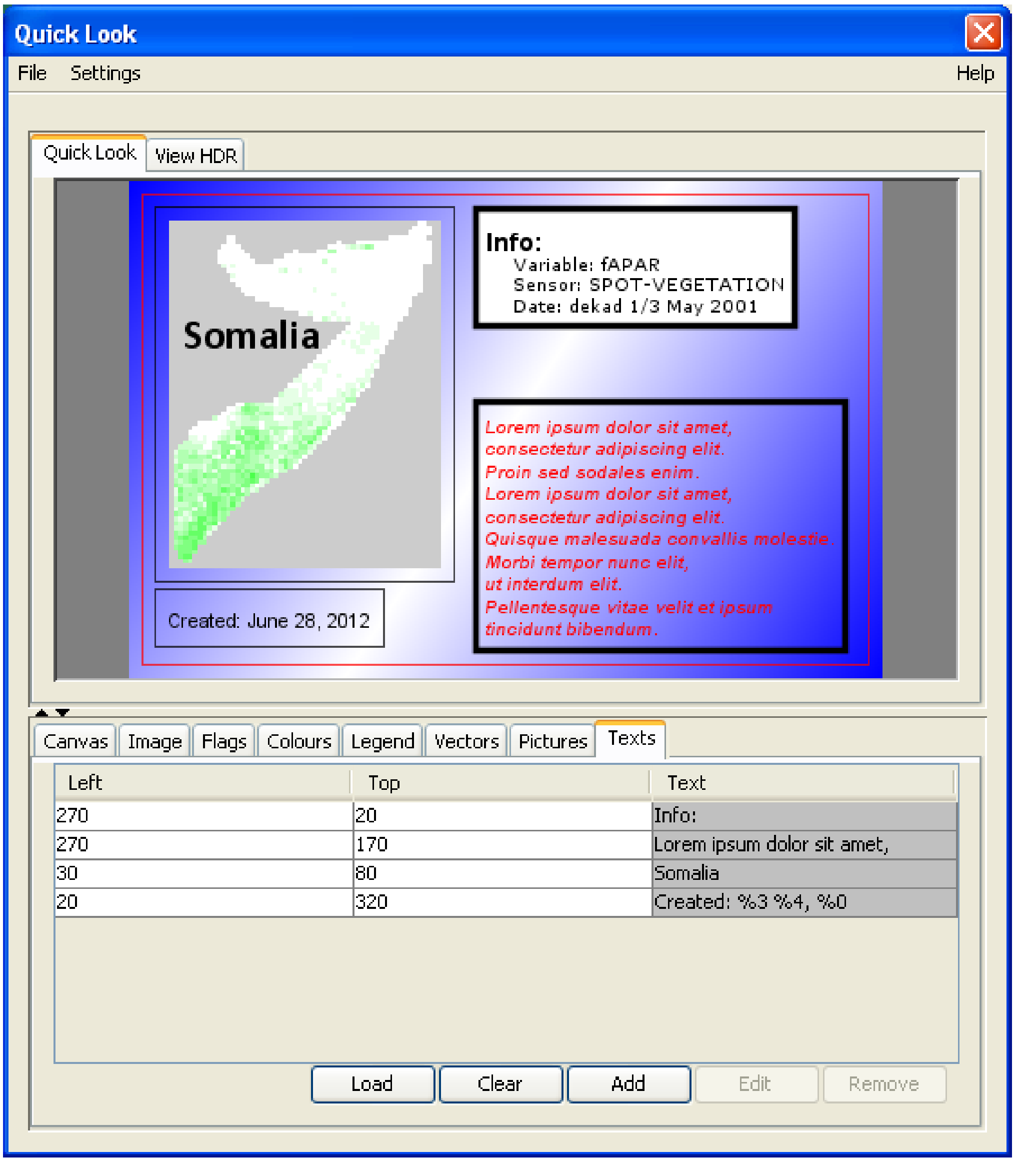
Under the Texts tab, textboxes containing lines of text can be created and added to the Quick Look.

A textbox is basically a rectangular area in which one or more lines of text are rendered.

The "Left" and "Top" parameters specify the position of the upper left corner of the textbox with respect to the canvas origin.

The width and height of a textbox cannot be specified, they are determined by the border, the margin and the content (lines of text) of the textbox.

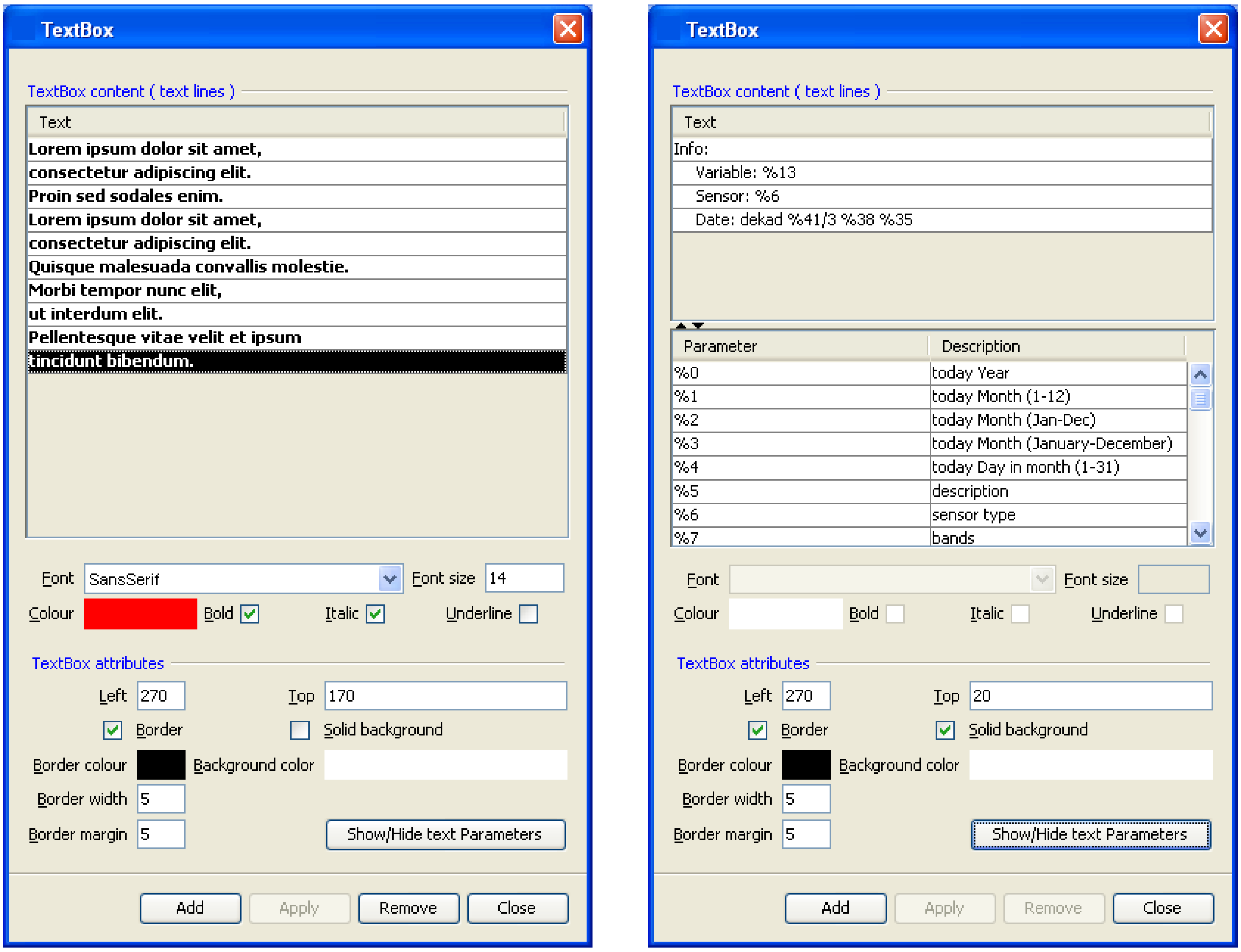
Optionally a textbox border can be specified, parameterized by its colour, its width, and the distance between the border and its content (margin). The background of a textbox can be filled with a solid colour or stay transparent.



For each separate line of text in a textbox, the font type, size, colour and style (bold, italic and underlined) can be specified.

Text lines support the use of leading blanks and the use of parameters (%0..%n).

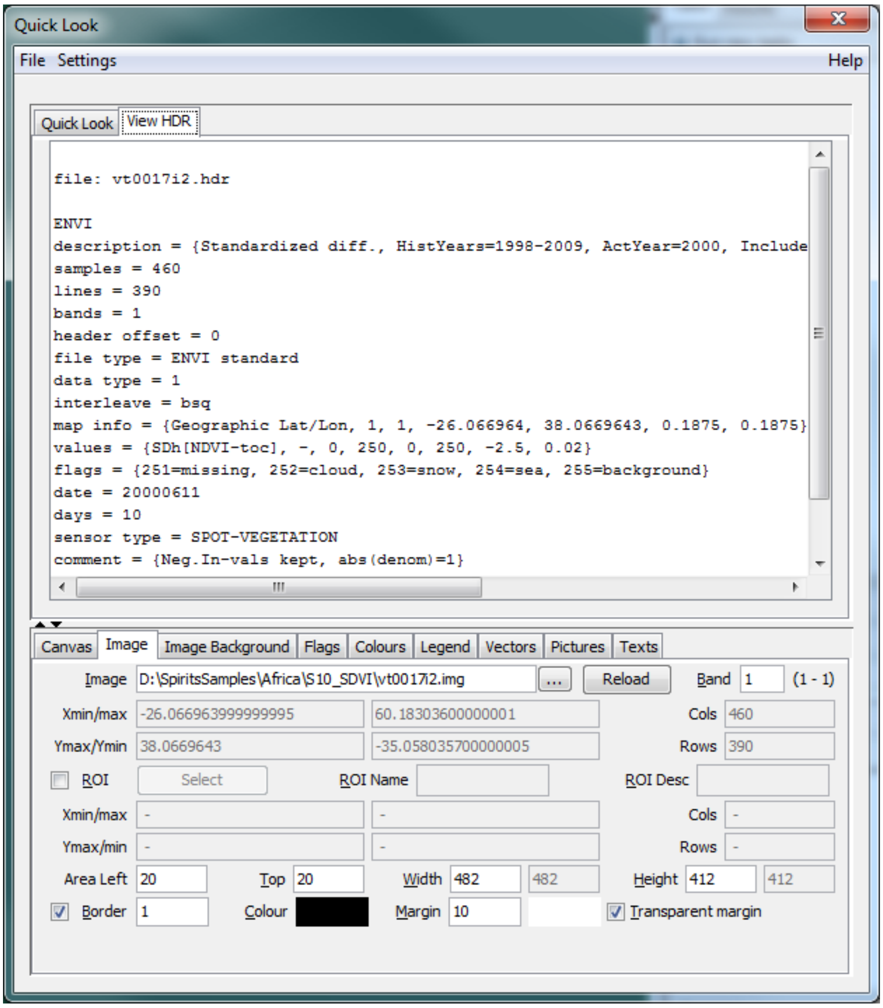
The values of these parameters can be previewed via the 'Text Parameters' button in the Textbox panel. These values originate from the current date, and from the content of the HDR file in case a (valid) IMG file has been specified in the Image tab.



5.1.2.10. View HDR

As an aid the View HDR panel is available.

This panel shows the content of the HDR file in case a (valid) IMG file has been specified in the Image tab.

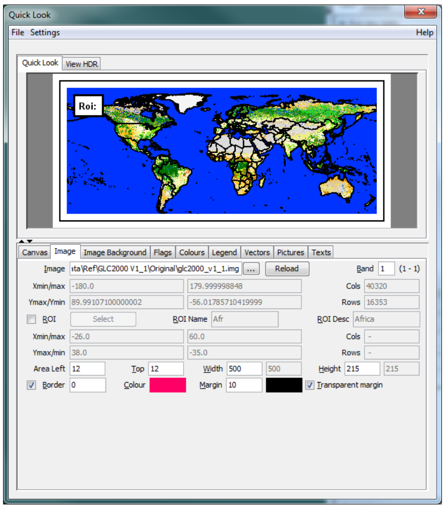


The ROI information of the IMG, and its dimensions in pixels is retrieved from the HDR information and visible on the Image panel: Xmin/Xmax, YMax/YMin and Cols/Rows.

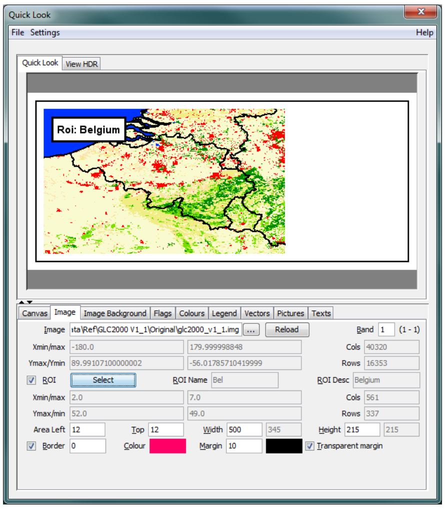
5.1.2.11. Image ROI selection.

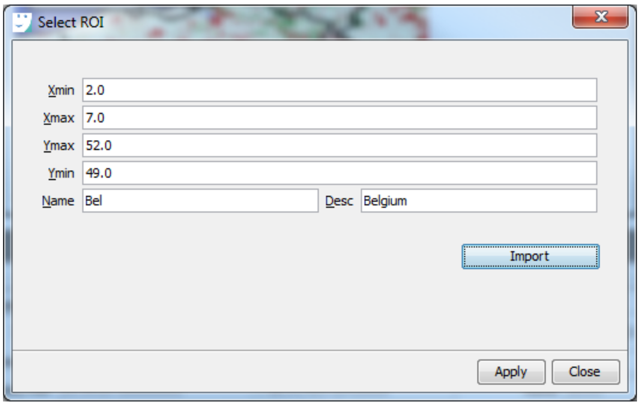
On the Image panel, one can choose to restrict the visualization to an ROI of the actual IMG.

Once the ROI option is enabled, the ROI itself can be specified via a panel activated via the "Select" button. On this panel the ROI can be defined via map-coordinates (Xmin/Xmax, Ymax/Ymin). It is also possible to specify a name and a description of the ROI, which can be used as parameters (%50 and %51) in the legend and texts. The selection panel offers the possibility to import the ROI boundaries from an existing HDR file.

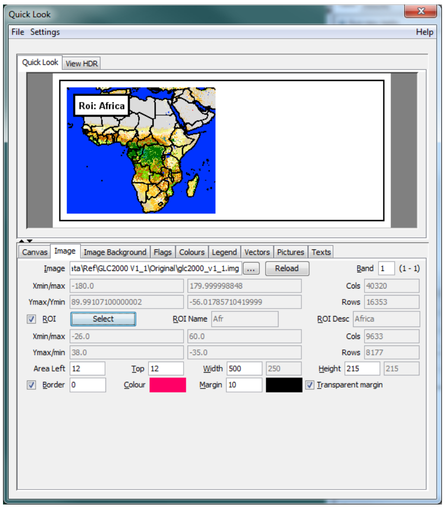


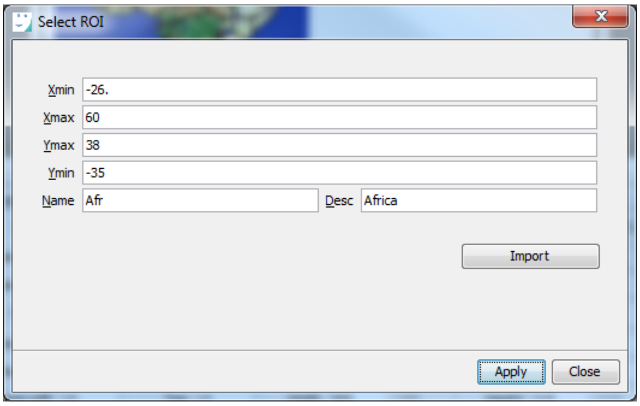
No Roi selected – full IMG file.





Belgium ROI selected.





Africa ROI selected.

5.1.3. Quick Look Settings

When starting a Quick Look, it uses defaults for most general settings such as the Canvas Width and Height. These settings can be changed by the user, and saved as new default settings via Settings>Set defaults . From then on these will be used upon starting new Quick Looks. The actual settings for which defaults are saved are:

- Canvas Width, Height, Border, Border Colour, Border Width, Border Margin, Background Colour1, Background Colour2, Background Gradient type.
- Image position Left, Top, Width and Height, Image Border, Border Width, Border Margin, Border Colour, Margin Colour, Margin Transparent, Image Background type, Image solid colour background Colour, Image colour scale background From and Till colours.
- Legend, Legend position Left and Top, Legend Border, Border Width, Margin, Colour, Legend Solid Background, Legend Background Colour, Legend Title Font Name, Size, Bold, Italic Underline and Colour, Legend Text Font Name, Size, Bold, Italic Underline and Colour.

The original default settings can also be restored via Settings>Reset defaults.

5.1.4. Quick Look File menu

A Quick Look can be saved / re-opened as a QNQ file via its File menu. These QNQ files can also be used as templates:

- by the Create Quick Look tool and the Create Quick Looks time series, to facilitate the creation of series of similar PNG files, based on a single Quick Look.
- In the Quick Look Form in the Flags, Colours, Vectors, Pictures and Texts panels, where Flags/Colours/Vectors/Pictures and Texts can be loaded from another Quick Look (QNQ) file

A PNG file of a Quick Look can be created via the Export PNG item in this File menu.

5.2. Create Quick Look tool

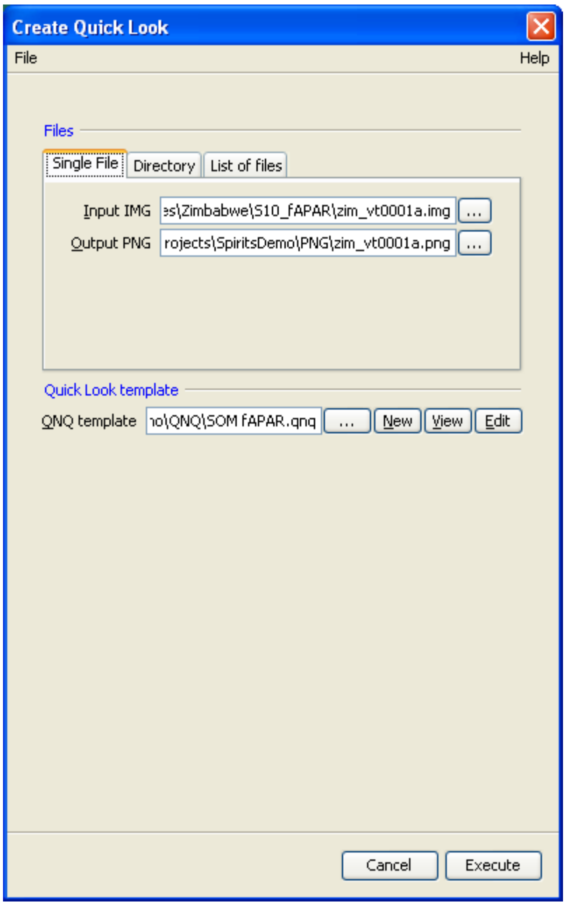
Goal

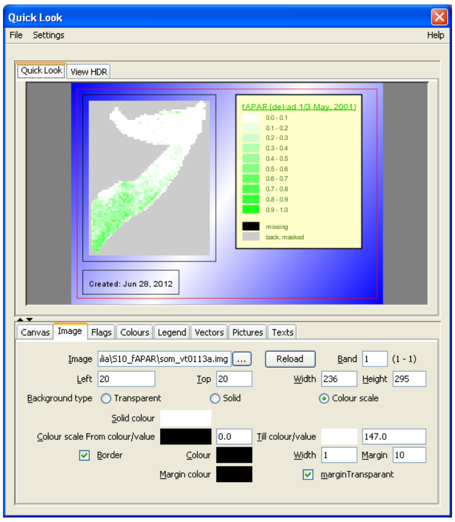
The Quick Look tool can create PNG files for a single input IMG file, a list (LNL file) of IMG files or a subset of the IMG files located in a directory.

Parameters

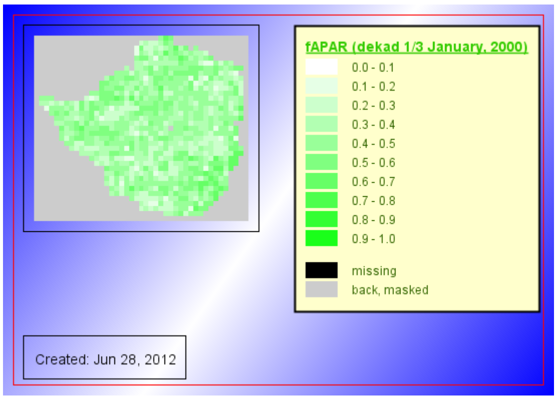
- existing input IMG file(s);
- output PNG file(s);
- an existing Quick Look (QNQ file) to be used as template.

Tool





QNQ for Somalia IMG used as template



Create Quick Look Tool example

PNG created for Zimbabwe IMG

For the Directory mode, the input files will be specified by their input directory and a wildcard pattern, using '*' and '?' as wildcards:

- '*' will be interpreted as 'any character, zero or more times';
- '?' will be interpreted as 'any character, exactly one time';
- all other characters in the wildcard pattern will be considered as constant.

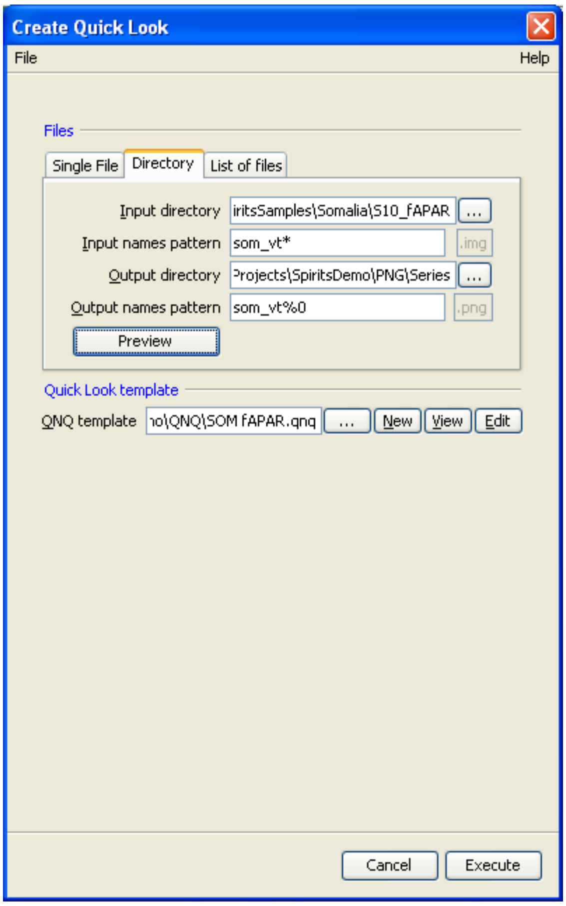
All files in the specified input directory, with their filenames matching the pattern, and having a (fixed) IMG extension, will be selected as input IMG file for the tool.

The wildcard pattern may contain up to ten wildcards. Each wildcard in the pattern, results in a corresponding parameter. For each selected file, the value of such parameter is the part of the filename covered by the wildcard. These parameters can be referred to as "%0", "%1,..."%9", and will be used to define the filenames of the PNG files to be created.

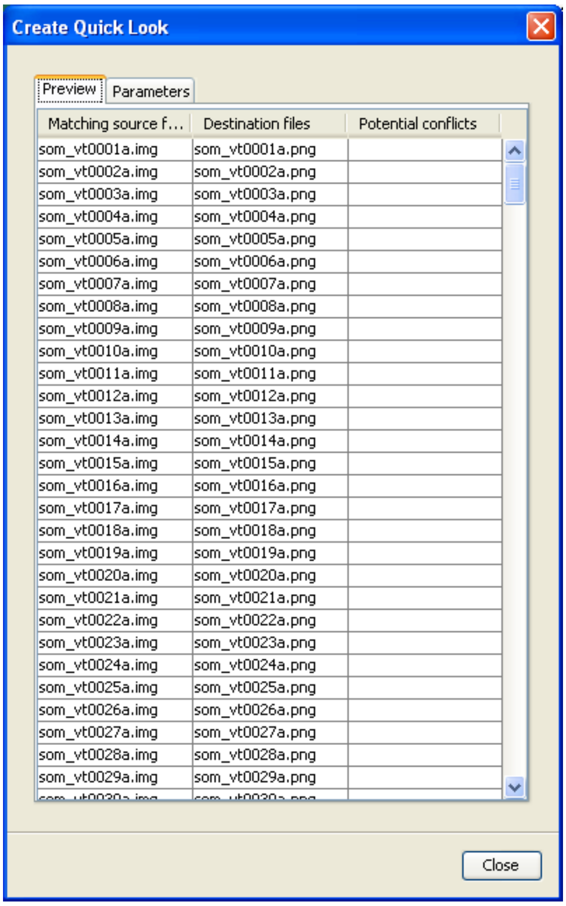
The names for the output PNG files will be specified by means of an output directory and a parameters pattern. This pattern can be a mixture of constant text and parameters ("%0", "%1,..."%9") obtained from the selected input files. The PNG extension is added automatically to the output file names pattern.

The output filenames, and potential conflicts (duplicate filenames etc.) can also be inspected in the Preview dialog.

By means of the Preview button, a dialog can be opened, showing the interpretation of the wildcards pattern, the input IMG files matching this pattern, the extracted parameters with their values, the names of the output PNG files and potential conflicts.



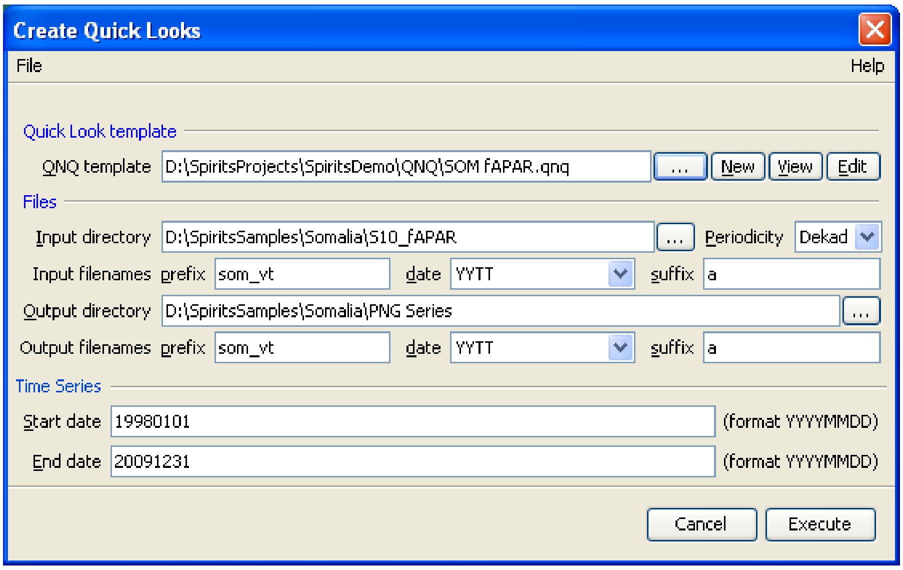
Create Quick Look tool - Directory mode

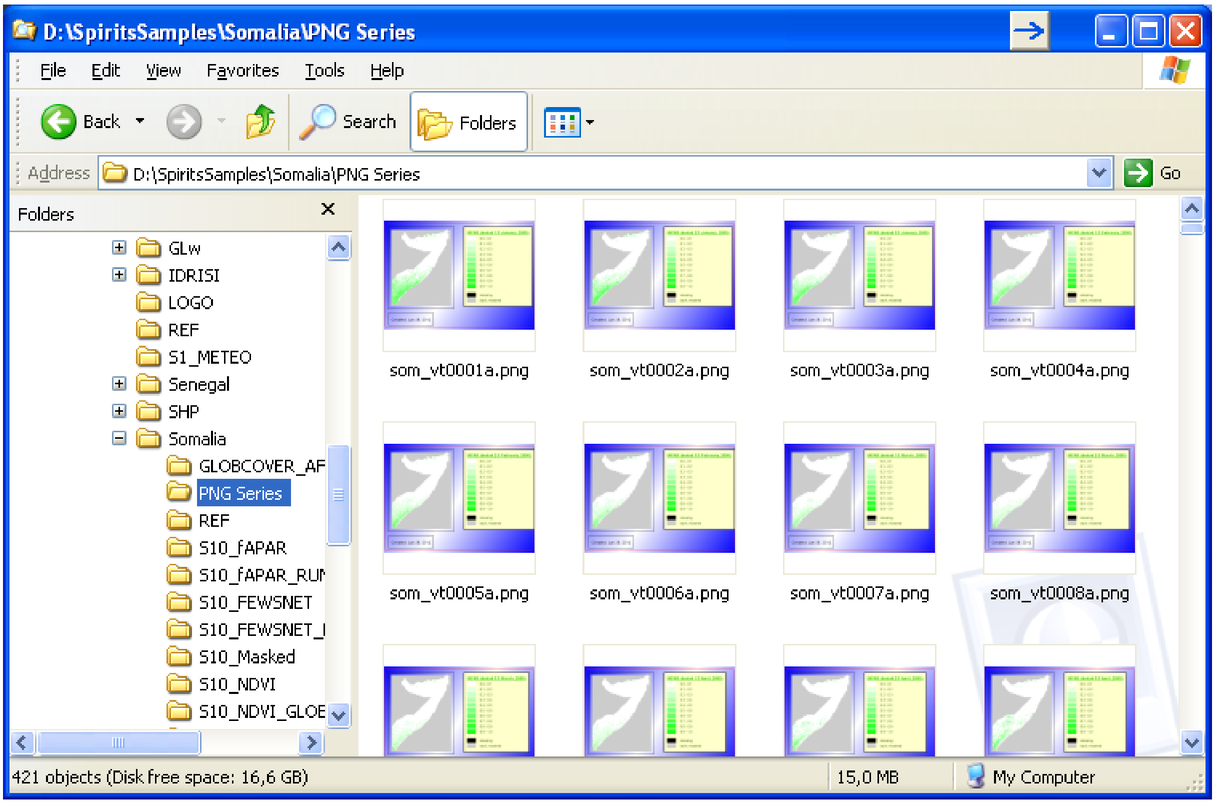


preview file names and parameters

Time Series

The Create Quick Looks time series tool can create PNG files for a time series of IMG files. Due to the limited number of parameters needed by the module, no separate scenario is used. Apart from the typical parameters needed in any time series tool (input files, output files, periodicity, start and end dates), an existing Quick Look (QNQ file) to be used as template is required.





example Create Quick Look Time Series - created PNG's

5.3. Create ROI Quick Looks tool

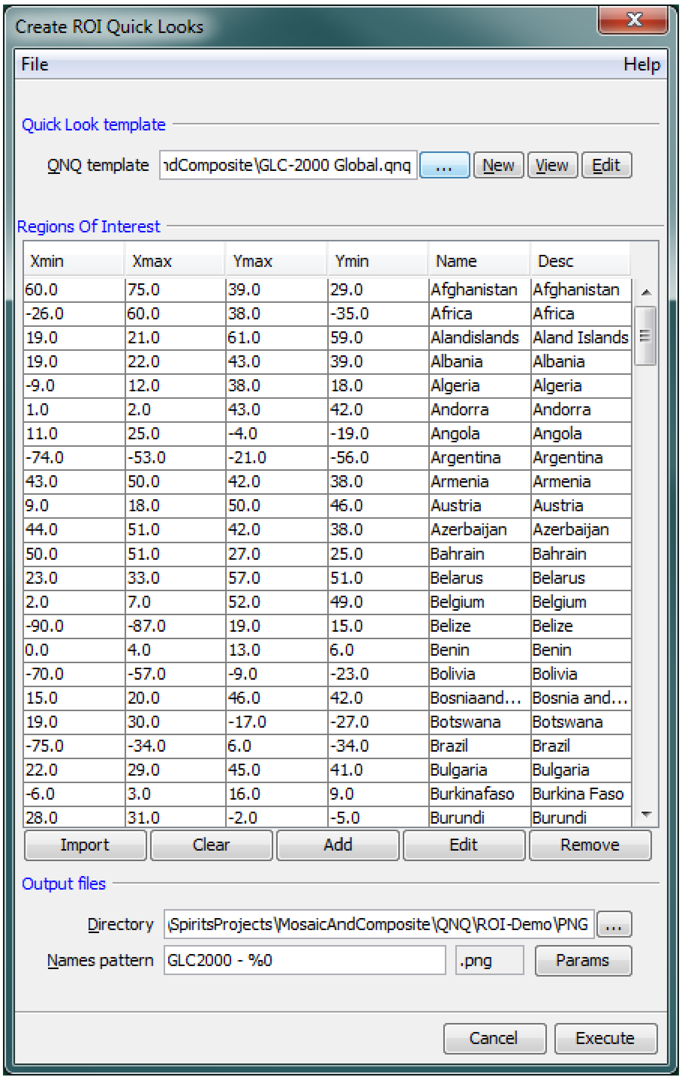
Goal

The ROI Quick Looks tool can create PNG files for a collection of ROI's from a single input IMG file (specified in a Quick Look template).

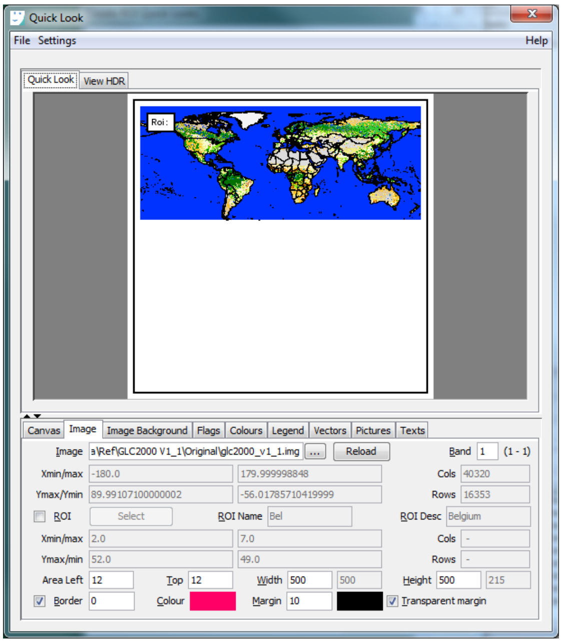
Parameters

- existing Quick Look (QNQ file) specifying the IMG and used as template.
- a collection of ROI's of the IMG in the QNQ;
- the location and filename of the resulting PNGs: the names for the output PNG files will be specified by means of an output directory and a parameters pattern. This pattern can be a mixture of constant text and parameters ("%0" or, "%1) obtained from the name or description of the ROI's. At least one of the available parameter must be used in the pattern. The ROI's names and/or descriptions will also determine the value of the corresponding parameters in the QNQ template, thus they can be used in legend and textboxes).

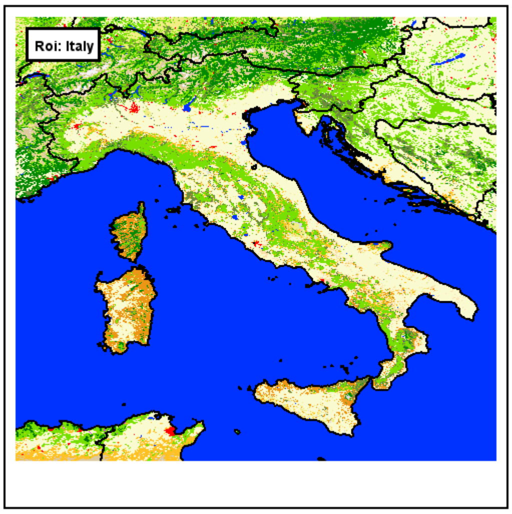
Tool



Create ROI Quick Look Tool example



QNQ used as template – contains a global IMG



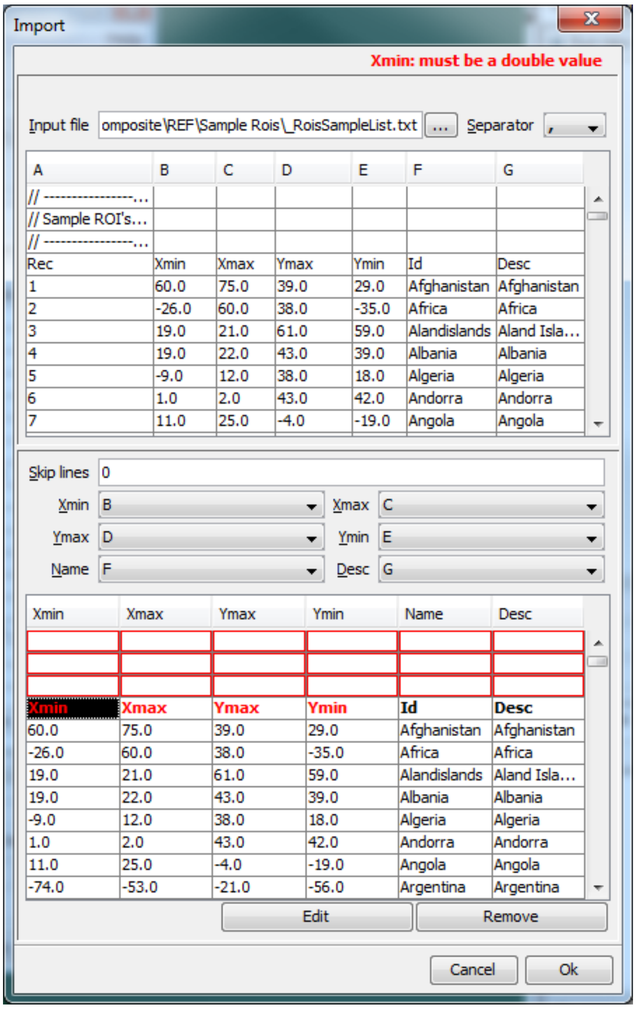
sample PNG created for Italy ROI

Following functions are available to specify/modify the ROI':

Add	Add a new ROI.
Edit	Edit selected ROI.
Remove	Remove selected ROI.
Clear	Remove all ROI's.
Copy/Save As	Copy the table to the clipboard / Save as a CSV file. (Via right-click on the table)
Import	Import a series of ROI's a (CSV) text file.

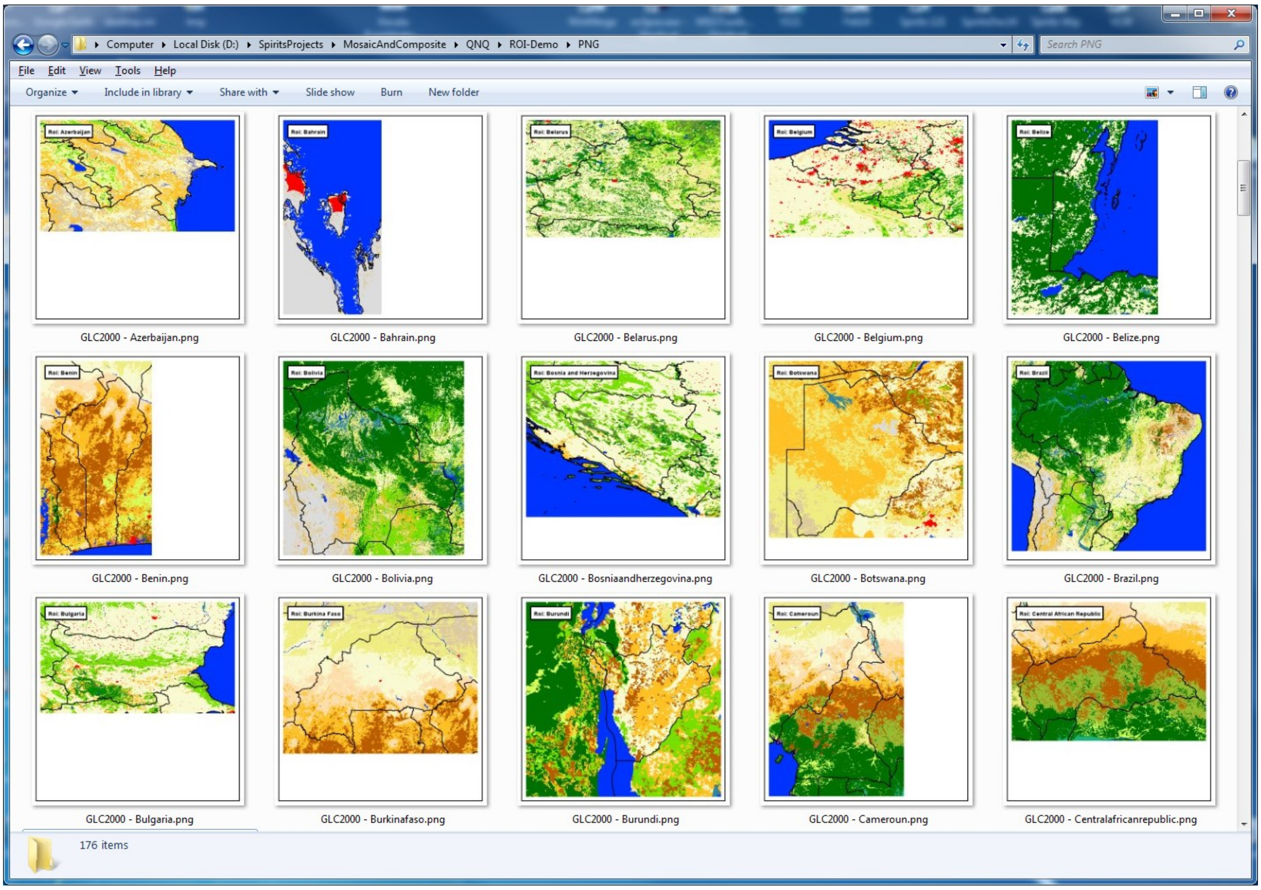
Series of ROI's can be imported from an ASCII CSV file. Such file (example below) should contain the ROI attributes (Xmin/Xmax/Ymax/Ymin/Name/Description). Coordinate values must use "." as decimal separator. The fields should be separated by comma, colon, semicolon or tab characters.

```
// ------------------------
// Sample ROI's ASCII list
// ------------------------

Rec,     Xmin,   Xmax,   Ymax,   Ymin,   Id,                           Desc
1,       60.0,   75.0,   39.0,   29.0,   Afghanistan,                  Afghanistan
2,      -26.0,   60.0,   38.0,  -35.0,   Africa,                       Africa
...
176,     42.0,   55.0,   20.0,   12.0,   Yemen,                        Yemen
177,     22.0,   34.0,   -8.0,  -19.0,   Zambia,                       Zambia
178,     25.0,   34.0,  -15.0,  -23.0,   Zimbabwe,                     Zimbabwe
```



Importing ROI's is an interactive process executed by the Import panel:

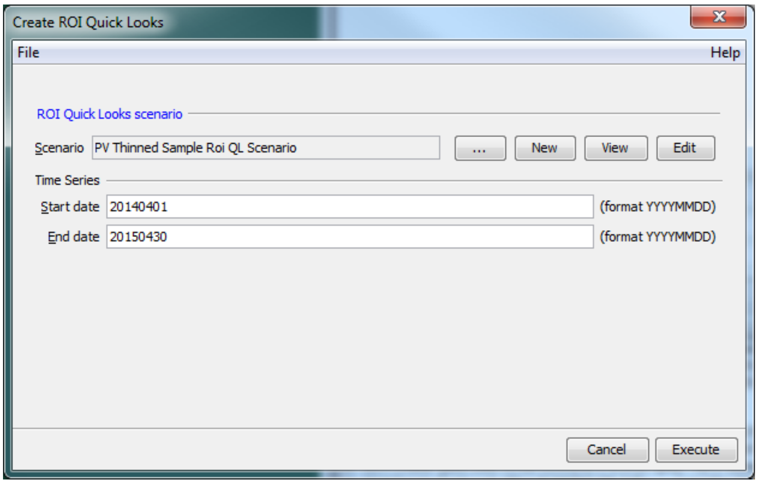
- In the Import panel, the file to be imported and the separator character to be used have to be chosen.
- An attempt is made to parse the file. The result is shown in the top half of the panel in tabular form.
- In the middle part of the panel, the ROI attributes (Xmin/Xmax/Ymax/Ymin/Name/Description) must be assigned one of the columns found by the parser.
- Since typical files contain leading lines describing the file contents, the Skip lines field allows to specify a number of lines to be ignored.
- In the bottom half of the panel the resulting entries to be imported are shown in a table.
- Invalid values are indicated in the table and an error message is displayed on top of the table.
- The ROI's in the table can be edited or removed.
- Upon selection of the Ok button the entries are added to the ROI's list in the Tool panel.



example Create ROI Quick Looks - created PNG's

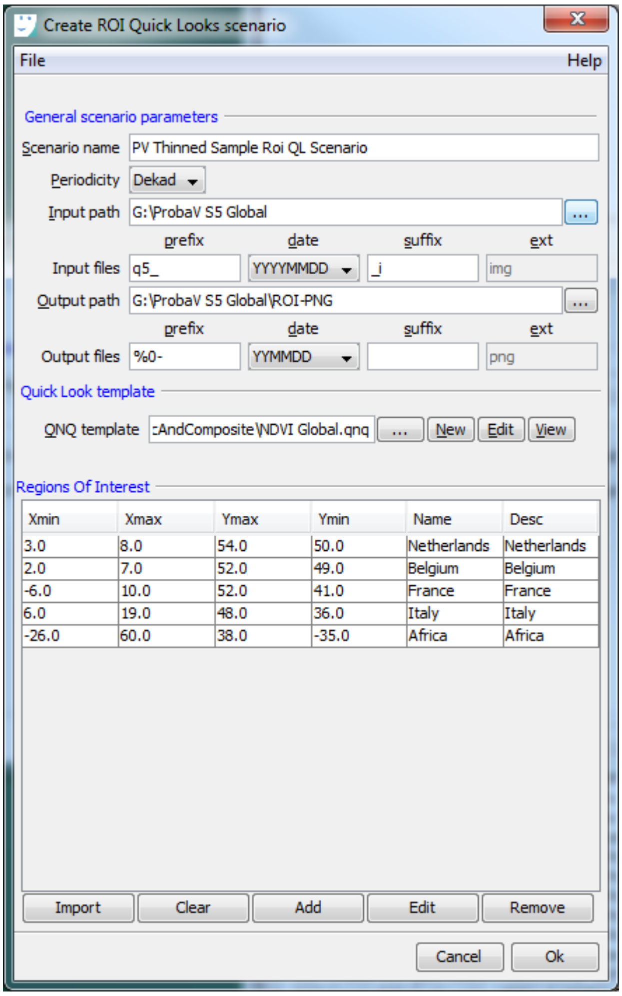
Time Series

The Create ROI Quick Looks time series tool can create PNG files, for a collection of ROI's, for a time series of IMG files.



Create ROI Quick Looks Time Series example

Scenario



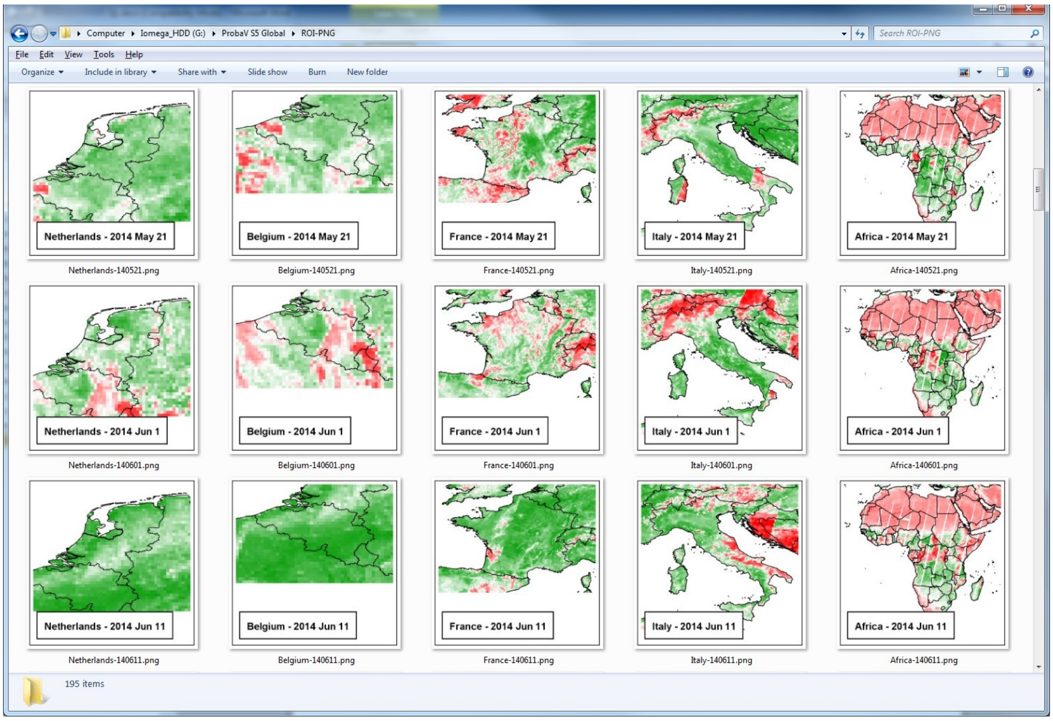
The ROI Quick Look scenario supports the use of parameters ('%0' and '%1') in the prefix and suffix fields specifying the output files.

At least one of the available parameter must be used in the prefix and/or suffix pattern.

The time series creates a loop between the dates specified in the time series tool, with a step according to the periodicity specified in the scenario.

For each of these dates, the IMG in the QNQ template will be replaced by the input IMG. For each ROI specified, an output PNG will be generated covering this ROI of the input IMG.

ROI Quick Looks Scenario example



example Create ROI Quick Looks Time Series - created PNG's

6. External programs and commands

6.1. Command Line Tool

6.1.1. Goal

The command line tool allows users to launch external executables and scripts (EXE, BAT, CMD...) from within the Spirits environment.

6.1.2. Tool

The command line tool parameters are:

- the input and output files;
- an optional set of environment settings for the process;
- the working directory for the process;
- the command string which signifies the external program file to be invoked and its arguments;
  - the command string could start with:
    - the full pathname of an executable file,
    - or the filename of an executable file in case it lives in a directory of the system path,
    - however it is recommended to launch the executable via the windows command line interpreter (cmd.exe - also known as command shell, command prompt or (archaic) DOS prompt).

      Since "cmd.exe" normally lives in the system path, it can referred with its filename. Since it is an ".exe" it can be directly referred as "cmd".

      In this case the command line should always start then with "**cmd /c**" (using the "/c" switch) otherwise the shell will not close after executing its commands, hence the task will never end and block the task queue.
  - the command string must contain at least one of the available symbolic in/out file constants. During execution these will be substituted by the values originating from the selected input and output files.

from input file	from output file	value	example
`%SRC%`	`%DST%`	`full filename`	`C:\MyData\MyFile.xyz`
`%SRCNOEXT%`	`%DSTNOEXT%`	`full filename without extension`	`C:\MyData\MyFile`
`%SRCDIR%`	`%DSTDIR%`	`directory only`	`C:\MyData`
`%SRCFILE%`	`%DSTFILE%`	`filename only`	`MyFile.xyz`
`%SRCFILENOEXT%`	`%DSTFILENOEXT%`	`filename only without extension`	`MyFile`
`%SRCFILEEXT%`	`%DSTFILEEXT%`	`extension only`	`xyz`

symbolic in/out file constants

Remarks:

- in some cases the actual command does not need an output file, but since the (generic) UI insists on one, some "dummy" file could be selected;
- depending on the actual command, filenames containing whitespace, may not be or must be enclosed by quotes. The substitution does NOT add quotes. if needed they should be added explicitly.
- the symbolic in/out file constants can be combined if needed, e.g. ***%SRC%*** gives the same value as ***%SRCDIR%\%SRCFILENOEXT%.%SRCFILEEXT%***.

6.1.3. Examples

Extracting ZIP archives using 7-Zip

7-Zip is a file archiver with a high compression ratio. It can be downloaded via http://www.7-zip.org/. A Command Line Version User's Guide is available at https://sevenzip.osdn.jp/chm/cmdline/index.htm. following examples assume 7-Zip has been installed properly.

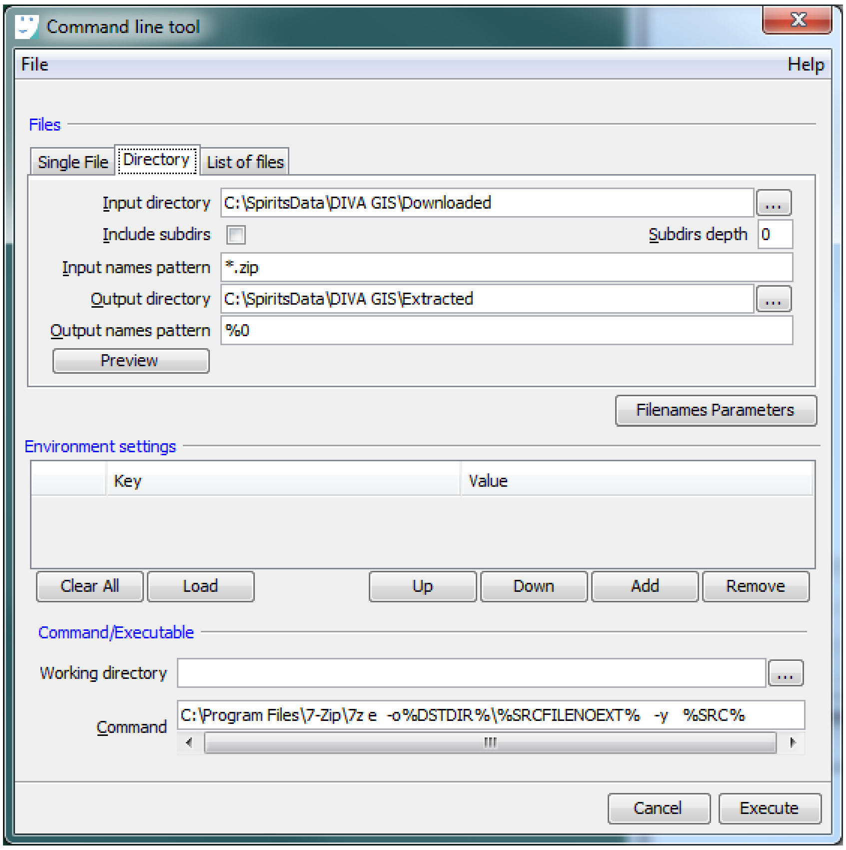
7-Zip installation:

```
7-Zip installation directory   C:\Program Files\7-Zip
7-Zip executable               7z.exe
```

extract of the 7-Zip command line syntax used in the examples

```
      7z <command> [<switch>...] <base_archive_name> [<arguments>...]
         <command>
                     e      Extract
                     x      Extract with full paths
         <switch>
                     -o     Set Output directory: -o{dir_path}
                     -r     Enable recurse subdirectories
                     -ao    Overwrite mode:
                            -aoa: Overwrite existing files without prompt.
                            -aos: Skip extracting of existing files
                     -y     Assume Yes on all queries
         7-Zip uses wild name matching similar to Windows 95
                     '*' means a sequence of arbitrary characters
                     '?' means any character.
```

example: simple extraction of all zip files in a directory

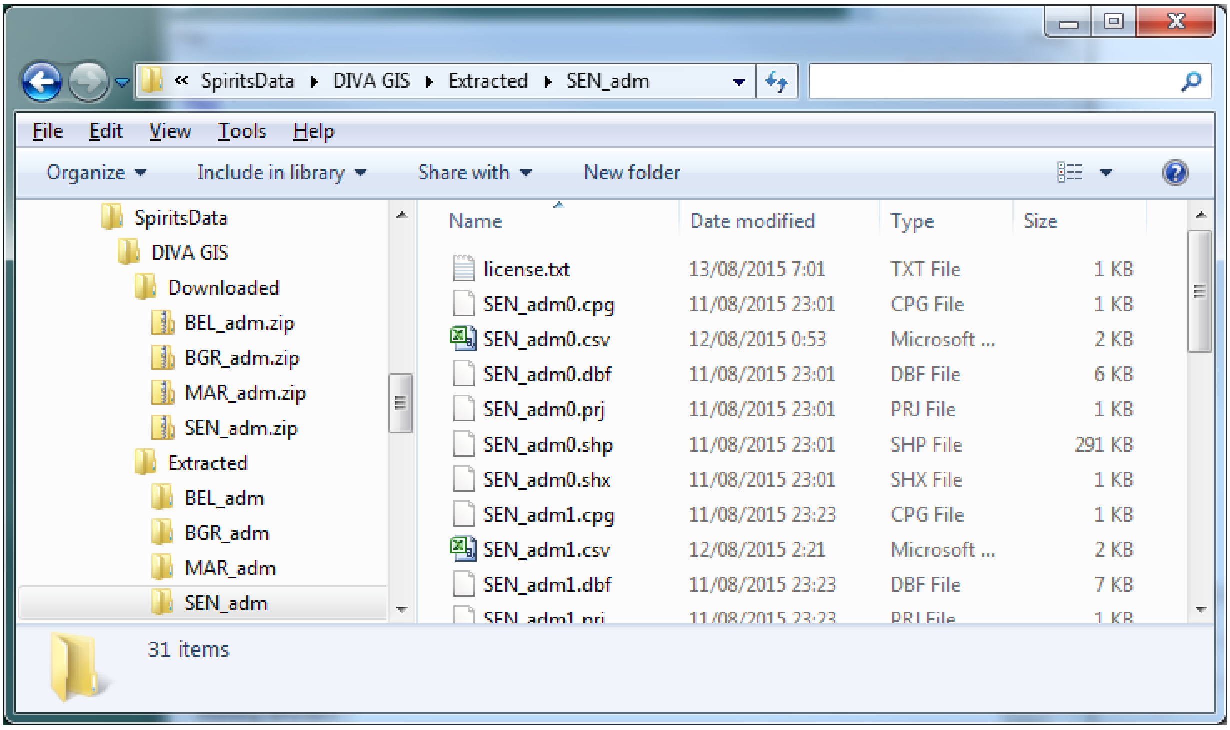


example: extraction of all zip files in a directory

The command in this example is:

C:\Program Files\7-Zip\7z e -o%DSTDIR%\%SRCFILENOEXT% -y %SRC%

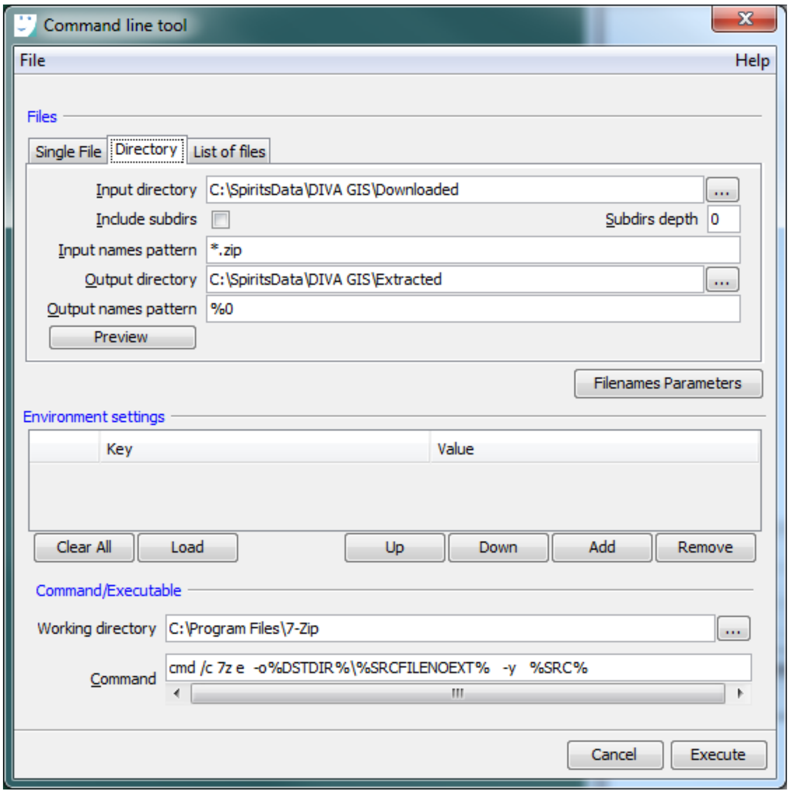
- C:\Program Files\7-Zip\7z : selects the 7-Zip executable (7z.exe) by specifying its full pathname;
- e : the 7-Zip "e" command selects the extraction mode; extract the contents of an archive;
- -y: specifies to assume Yes on all queries. This switch is important to avoid "hanging". E.g. when running this task twice, the output files will already exist. Without the "-y", 7-Zip would stop and wait for user input to decide whether to overwrite or skip the existing files, so the task would "hang" and block the task queue.
- %SRC% : the archive (zip file) to be un-zipped. During execution this will be substituted by the values originating from the selected input files.
- -o: the -o switch specifies the output directory:
    - in case this switch would be omitted, the extracted files would end up in the current working directory. Since we did not specify this, it could be anywhere –probably the location from where Spirits was launched- what would definitely not be our intention;
    - this output directory is specified as %DSTDIR%\%SRCFILENOEXT%. During execution this will be these will be substituted by the values originating from the selected input and output files. In our example this gives following results:



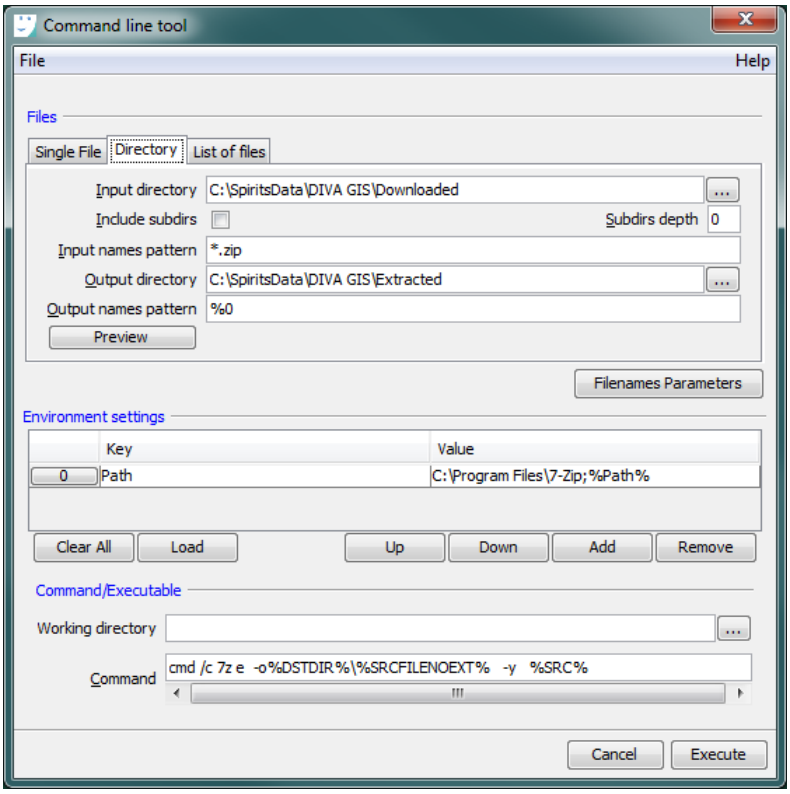
each .zip file in the "Downloaded" directory (the %SRC% file) is unzipped into output directory %DSTDIR%\%SRCFILENOEXT%, being a (new) subdirectory (named by %SRCFILENOEXT%) in the "Extracted" output directory (%DSTDIR%).

alternative commands

by using the windows command line interpreter (starting the command with **cmd /c**) the syntax can be simplified. E.g. by setting 7-Zip installation directory to be the working directory or by adding it to the system path, the 7z executable can be referred without specifying its full pathname.



using the working directory

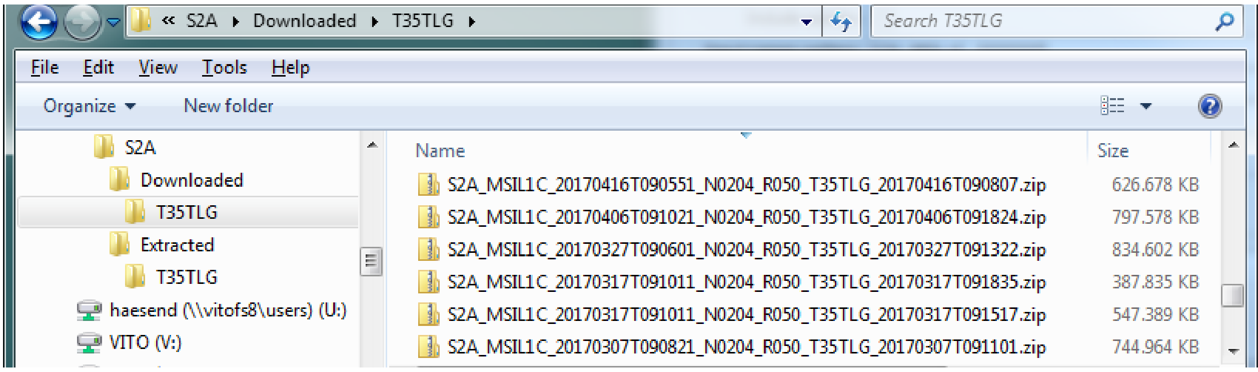


extending the system path

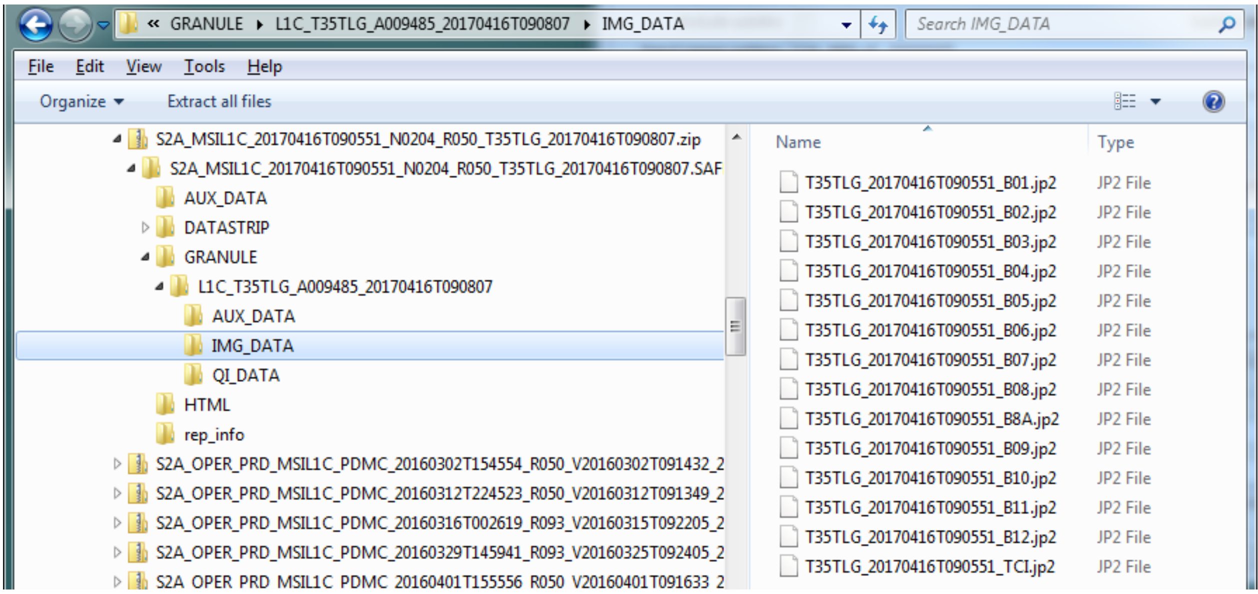
Extracting specific files from ZIP archives using 7-Zip: Sentinel 2 archives

Sentinel 2 Level 1C data can be downloaded e.g. from the Copernicus Open Access Hub (https://scihub.copernicus.eu/). This data comes in a .zip archive containing a complex folder structure of directories and files with meta data, vectors, JPEG2000 images,...of which the format and specifications can be found at the Copernicus site.

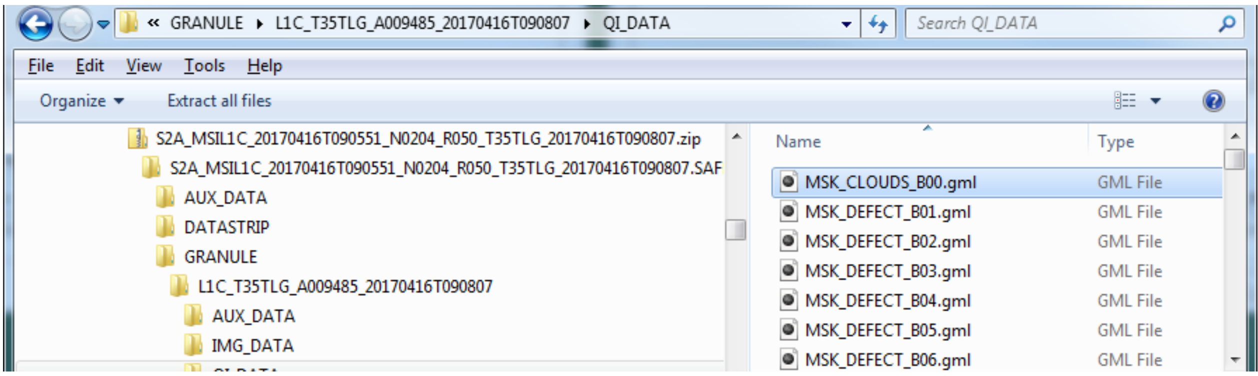
In this example specific files (bands B04 and B08 (JPEG2000 format) and the cloud mask (GML format)) will be extracted from a series of downloaded Sentinel 2 archives for a specific tile.



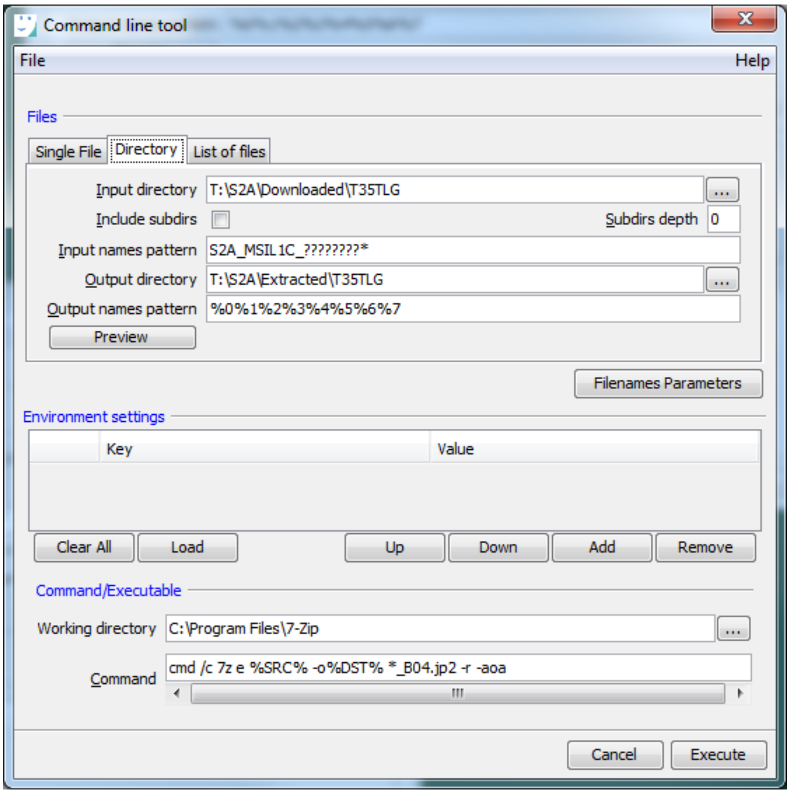
Sentinel 2 tile T35TLG archives, as downloaded from Copernicus Open Access Hub



Sentinel 2 actual bands (JPEG2000) in an archive



Sentinel 2 Clouds mask (GML) in an archive



The actual command is similar to that of previous examples.

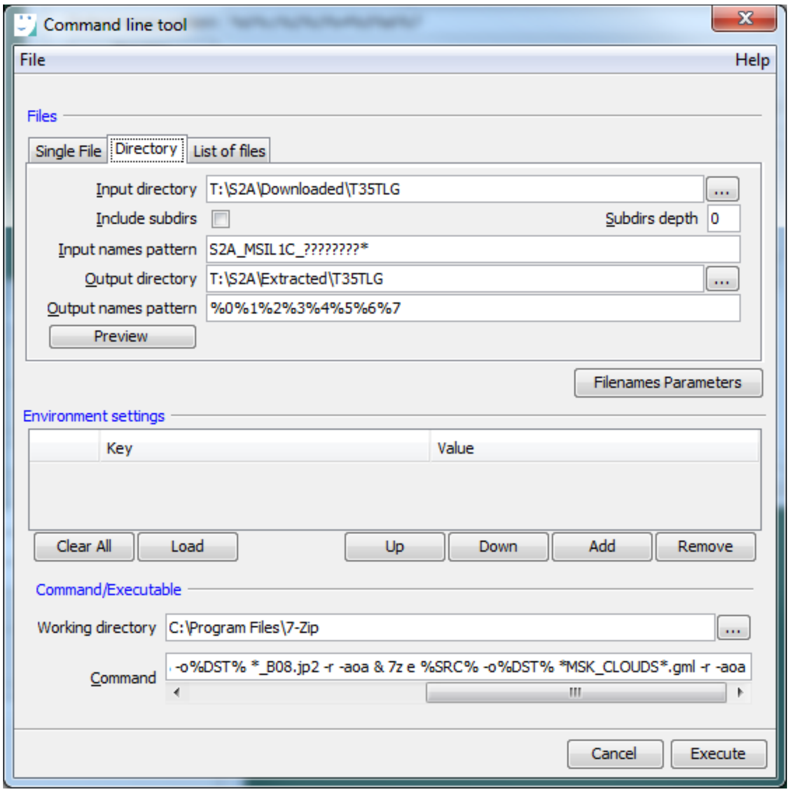
Instead of extracting all files in the archive, we only select the Band 4 images via the *_B04.jp2 pattern.

The –r switch is needed to allow recursion into subdirectories since the archives have the images in a deep directory tree.

Via the wildcard pattern used in the input files selection, we can retrieve the date coded in the archive filename, and use this date –via the output names pattern- to specify the subdirectory in which the images will be extracted.

example: extraction of the B04 bands from the Sentinel 2 archives

By changing the *_B04.jp2 pattern into *_B08.jp2 and *MSK_CLOUDS*.gml we could create tasks for the B08 band and the clouds mask. However, the windows command line interpreter allows to specify multiple commands by separating them with a '&'. This way we can extract all files we need in a single task.



Command : the windows command line interpreter calls 7-Zip three times:

```
cmd /c 7z e %SRC%
 -o%DST% *_B04.jp2 -r -aoa
& 7z e %SRC%
-o%DST% *_B08.jp2 -r -aoa
& 7z e %SRC%
-o%DST% *MSK_CLOUDS*.gml -r
-aoa
```

example: extraction of the B04 and B08 bands plus the clouds mask from the Sentinel 2 archives

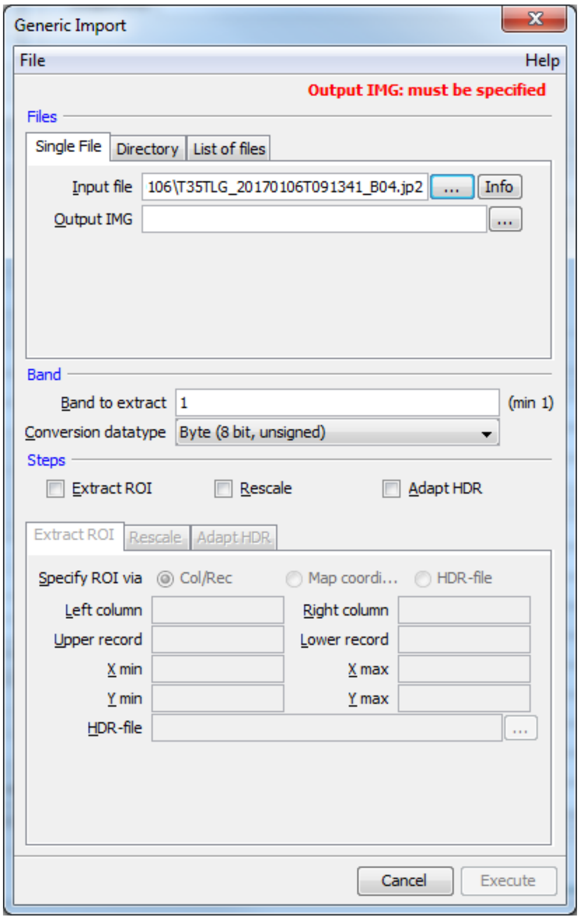
Rasterizing GML files using gdal_rasterize: Sentinel 2 clouds mask

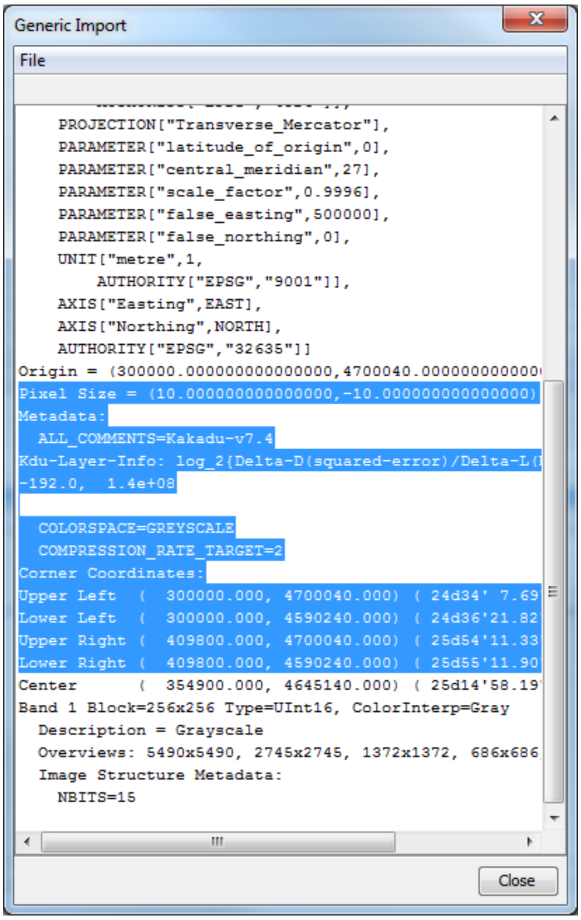
In this example the Sentinel 2 Level 1C cloud masks (GML format) as extracted in previous example, will be rasterized using gdal_rasterize utility from the GDAL Utilities (Geospatial Data Abstraction Library from the Open Source Geospatial Foundation).

The gdal_rasterize syntax used is as follows:

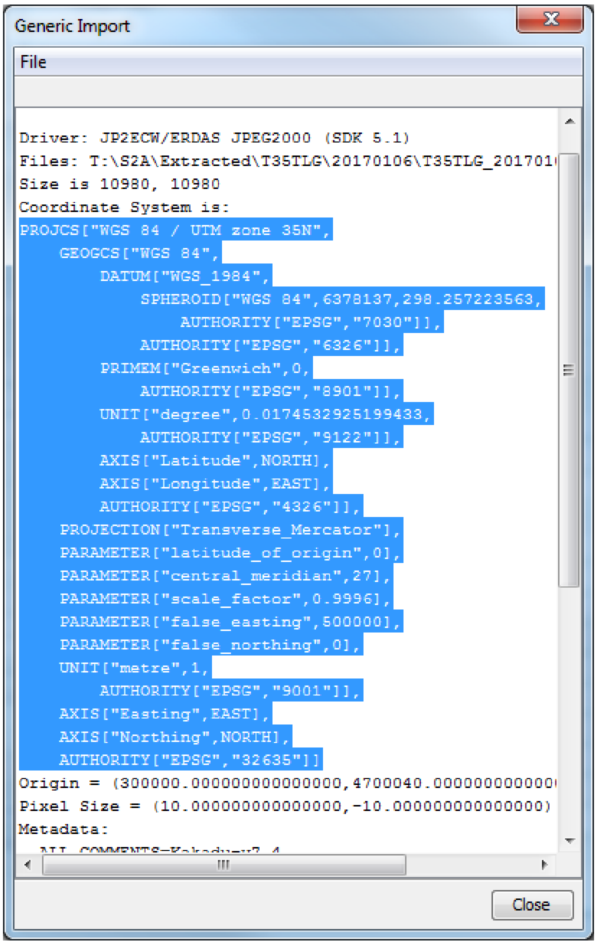
```
gdal_rasterize
  -burn value                    fixed value to burn into a band for all objects
  -te xmin ymin xmax ymax        georeferenced extents
  -tr xres yres                  target resolution
  -a_srs srs_def                 specify projection for the output file
  -of format                     output format
  -ot type                       output data type
  <src_datasource>               datasource
  <dst_filename>                 output file
```

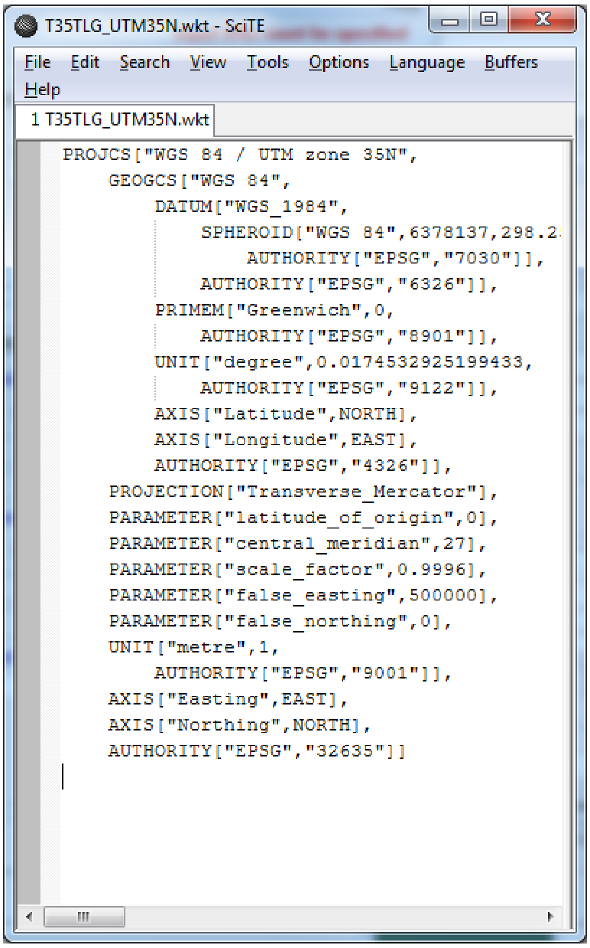
The rasterized cloud mask needs to have the same extent and resolution as the B04 and B08 bands. This information can be retrieved by selecting one of these files in the Generic Import Tool and use the "Info" function (which is based on the gdalinfo utility).





This "info" also contains the projection information. This can be copied into some text editor (Notepad, Scite, ...), and saved as a .wkt file which then can be used to specify the projection info of the rasterized output file.





All clouds mask files have the same name (MSK_CLOUDS_B00.gml). However when extracting them in previous example these files were extracted in a subdirectory which also contains both bands (T35TLG_..._B04.jp2 and T35TLG_..._B08.jp2). It is now possible to select one of these band files as input, and tinker with the symbolic in/out file constants in the command itself.

File selection:

input names pattern	T35TLG_*T*B04.jp2
output names pattern	T35TLG_%0_MSK_CLOUDS.img

will select the B04 files, and extract their dates as first wildcard parameter, which can be used in the output filenames.

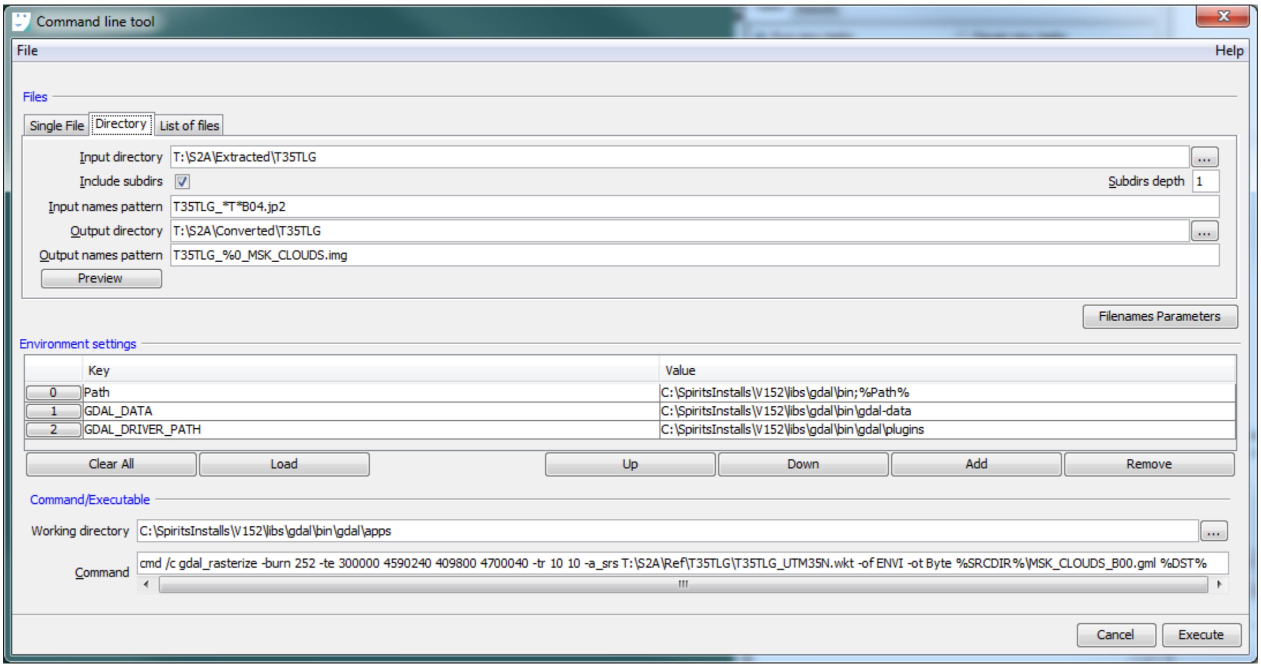
Command:

gdal_rasterize src_datasource	%SRCDIR%\MSK_CLOUDS_B00.gml
gdal_rasterize dst_filename	%DST%

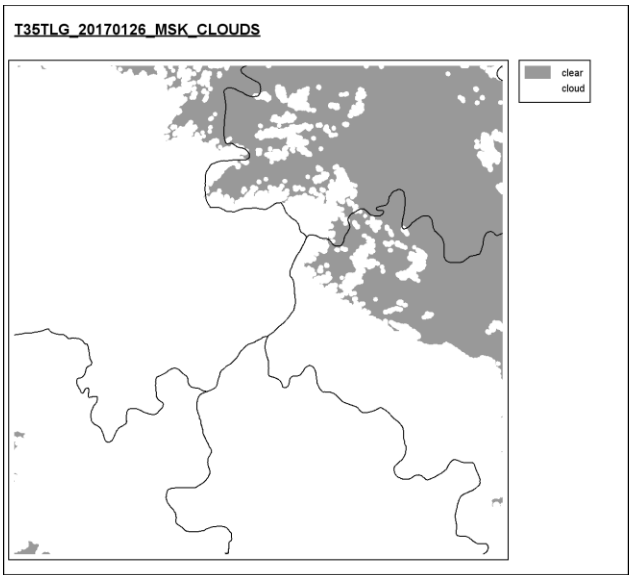
From the selected input file (the B04 band) only the directory part is used. This directory, combined with the fixed MSK_CLOUDS_B00.gml filename can then serve as gdal_rasterize src_datasource parameter.

With this information we can assemble the command to be used in the Command Line Tool.

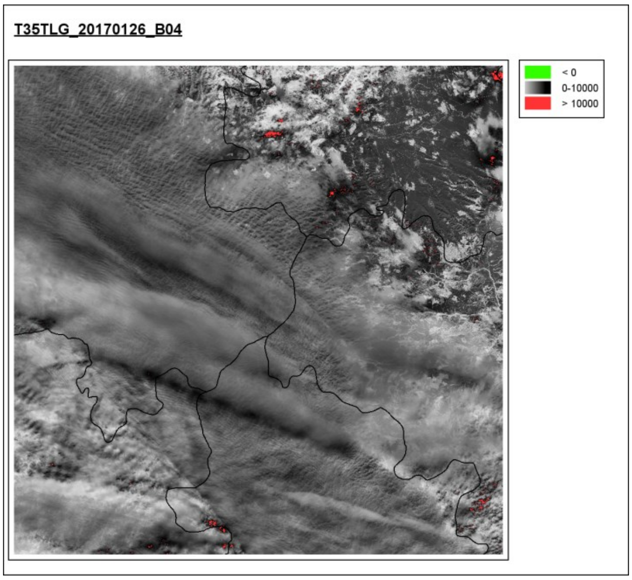
Command	
	cmd /c
gdal_rasterize	**gdal_rasterize**
-burn value	**-burn 252**
-te xmin ymin xmax ymax	**-te 300000 4590240 409800 4700040**
-tr xres yres	**-tr 10 10**
-a_srs srs_def	**-a_srs T:\S2A\Ref\T35TLG\T35TLG_UTM35N.wkt**
-of format	**-of ENVI**
-ot type	**-ot Byte**
<src_datasource>	**%SRCDIR%\MSK_CLOUDS_B00.gml**
<dst_filename>	**%DST%**



example: rasterizing the clouds mask from the Sentinel 2 archives



rasterized cloud masks for 2017 January 26



Band B04 2017 January 26

6.2. User Tools

6.2.1. Goal

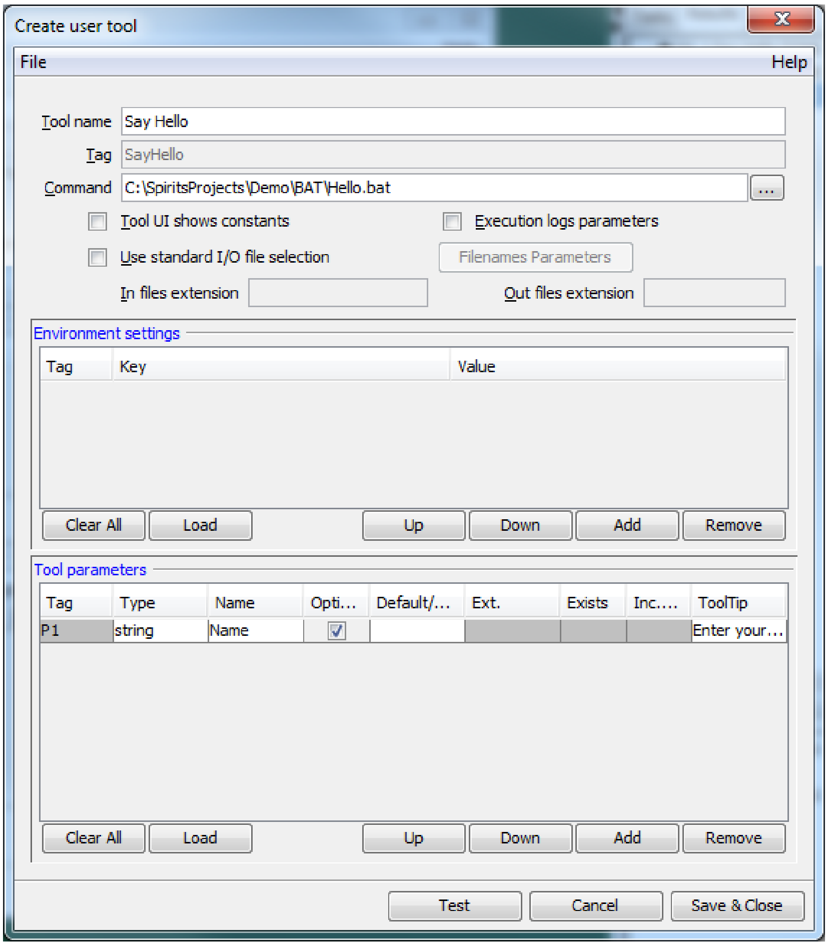
The goal of the "user tools" concept, is to allow users to integrate external executables or scripts (EXE, BAT,...) as tools in the Spirits environment. Typical targets would be programs such as the GDAL Utilities (Geospatial Data Abstraction Library) from the Open Source Geospatial Foundation or simple .BAT or .CMD scripts chaining such programs.

User tools enable the user:

- to create simple UIs to collect the parameter values needed by the tool;
- to select one of the tools created this way;
- to display its UI, which will then interact with the user just as the 'normal' Spirits tools: after specifying (valid) parameter values, a task can be submitted to be executed via the task queue.

```
::
::     Say Hello
::
@ECHO off
ECHO Hello %~1
EXIT 0
```

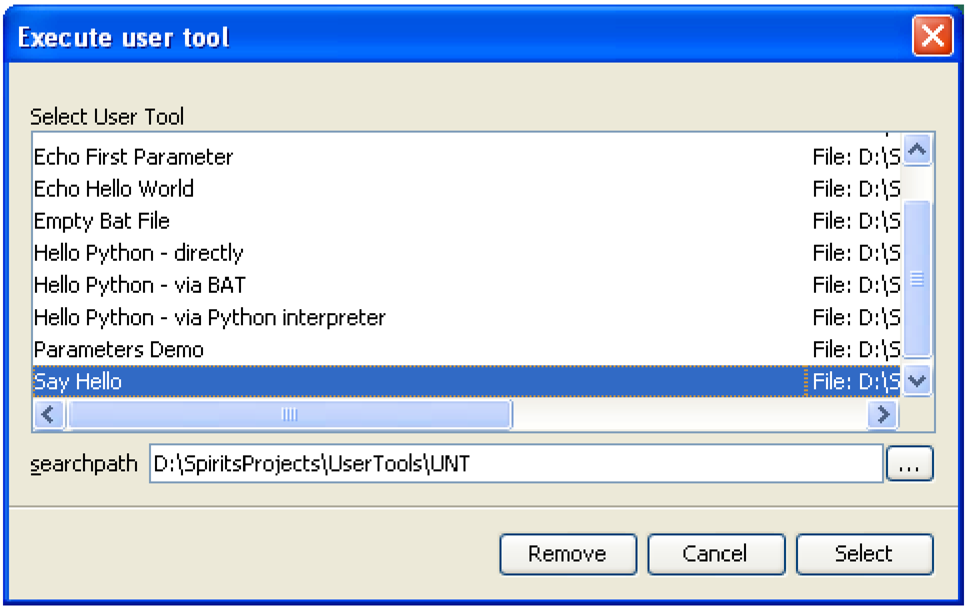
example: external executable: Hello.bat which uses one parameter



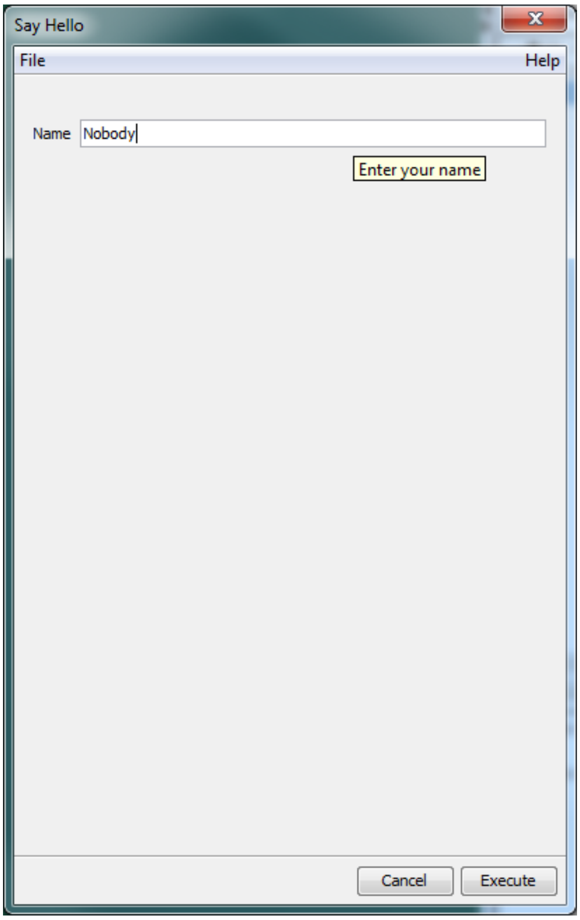
select program “Hello.bat”

specify single optional parameter of string type under UI label “Name”

example: creation of a user tool to call the “Hello.bat” program with one parameter



example: selection of the created user tool for “hello.bat”




```
2012/05/30 10:38:59 STATE:   RUNNING
Hello Nobody
2012/05/30 10:38:59 STATE:   DONE
```

example: UI of the user tool for Hello.bat

example: execution output in Task results queue

6.2.2. Creating User Tools

The Create user tool utility enables the user to create a simple UI for an external program.

The result of this process will be saved as a user tool template file (*.UNT). Such UNT file can later be re-opened in the Create user tool utility to be inspected, edited or re-used to create another user tool.

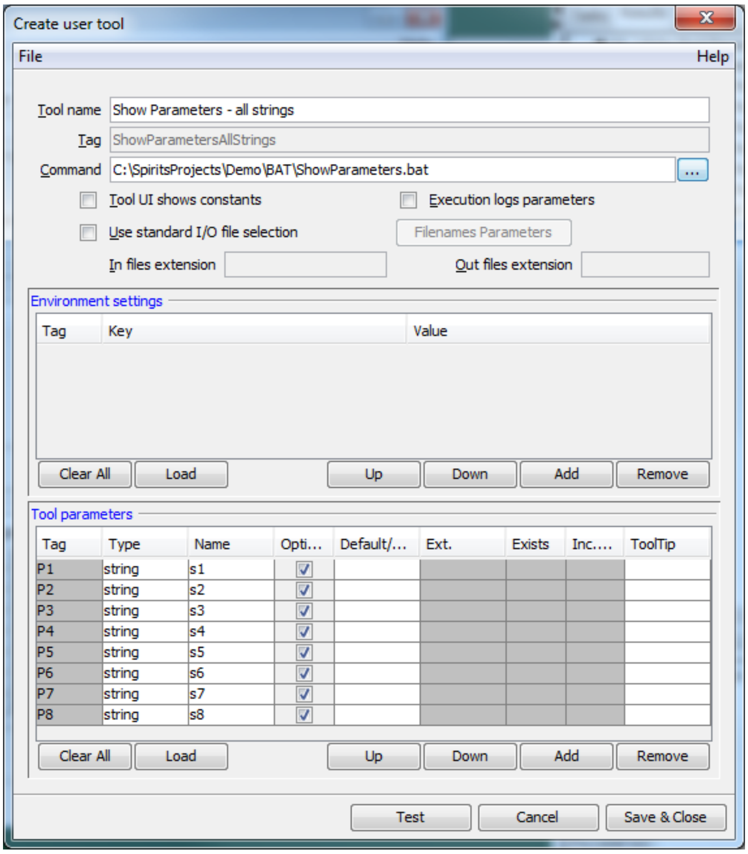
The Create user tool parameters are:

- the Tool name: a description of the tool which will be used to select it from the User tools list and which will be the window title in the UI;
- the Command: the actual executable file to be launched;
- an option whether or not the UI should show the constant parameters (with their fixed values);
- an option whether or not the parameter values should be logged in the execution output;
- an option whether or not the UI should contain the “standard I/O file selection” panel (which is used in most ‘normal’ Spirits tools). In that case one can also specify the mandatory in and out file extensions;
- an optional set of environment settings as needed by the executable;
- an optional set of argument-parameters as used by the executable.

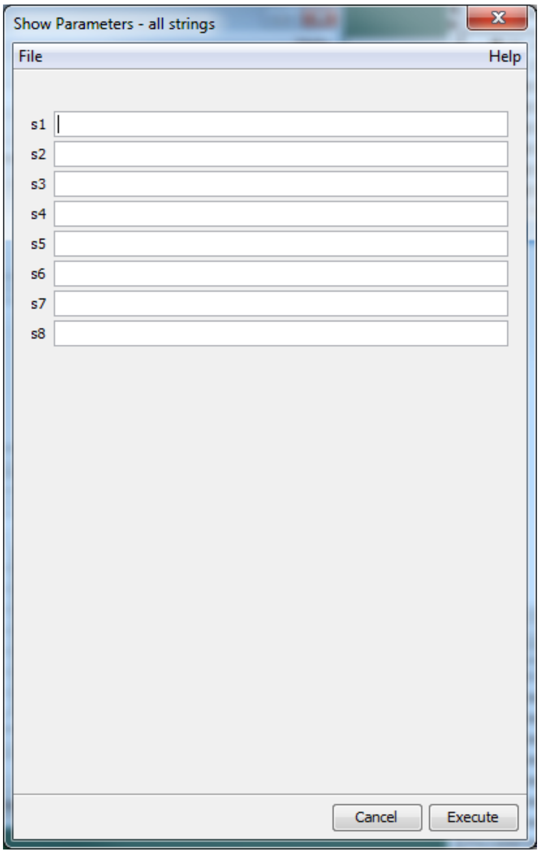
Upon selection of the created tool from the User tools list, these argument-parameters will be presented to the user in a simple UI, where they can be filled out.

```
@echo off
:NEXT
if "%~1"==""  exit 0
echo Parameter: %~1
shift
goto NEXT
```

example: external executable: ShowParameters.bat which outputs each parameter it receives



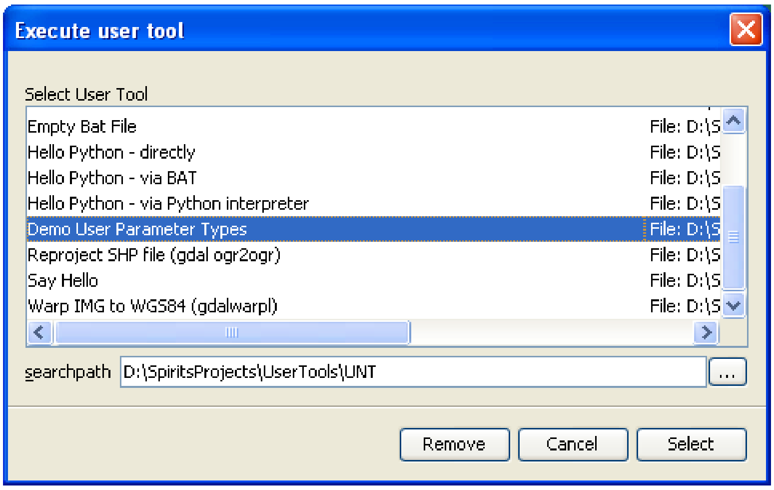
example: create user tool calling “ShowParameters.bat” with 8 string parameters



resulting user tool UI

6.2.3. Executing User Tools

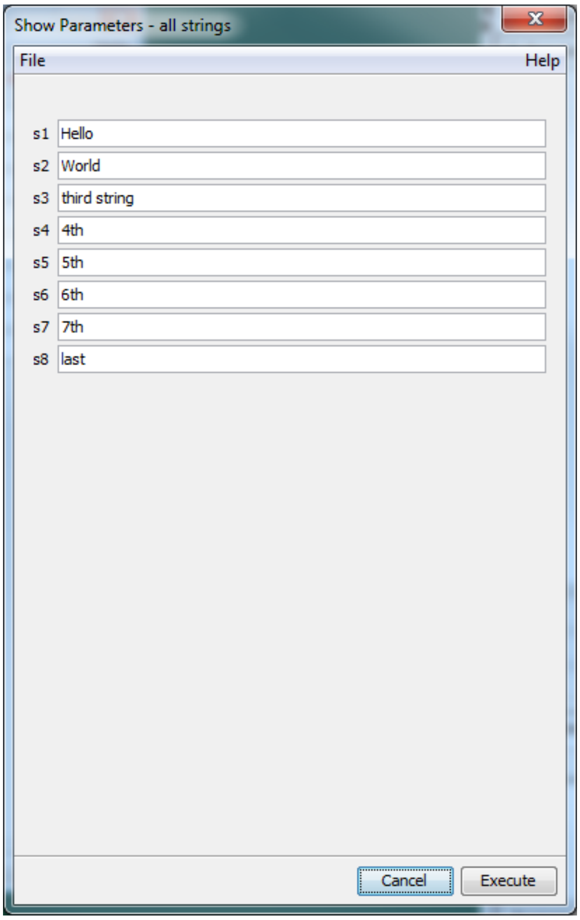
The Execute user tool utility enables the user to select, interact with and submit user tools. The Execute user tool panel shows all user tools available in the specified directory. This directory defaults to the User tool files directory (specified in Project settings) but can be changed to select tools from another directory.



Execute user tool: selection panel

Upon selection the user tools UI opens. Such user tool UI behaves just the 'normal' Spirits tools:

- some basic checks will be performed on the parameter values entered;
- the tool can be submitted to be handled by the task queue;
- upon completion, its results can be inspected in the results queue;
- the parameters (task) can be saved to be reused afterwards in a *.TNT file.




```
2017/04/26 05:30:57 STATE:   RUNNING
Parameter: Hello
Parameter: World
Parameter: third string
Parameter: 4th
Parameter: 5th
Parameter: 6th
Parameter: 7th
Parameter: last
2017/04/26 05:30:57 STATE:   DONE
```

example: UI of the user tool for “ShowParameters.bat”   example: execution output in Task results queue

6.2.4. User Tools parameters and UIs

In the Create user tool, each argument-parameter can be assigned:

- the Type of the parameter (mandatory). Nine parameter types are supported: string, integer, float, file, directory, constant, prefix and pattern. The parameter type allows the created tool to perform some basic checks on the values entered by the user;
- the Name of the parameter (mandatory). This name will appear in the UI of the created tool;
- the ToolTip for the parameter(optional). This tooltip will show in the UI of the created tool;

Depending on the selected type, additional settings are applicable.

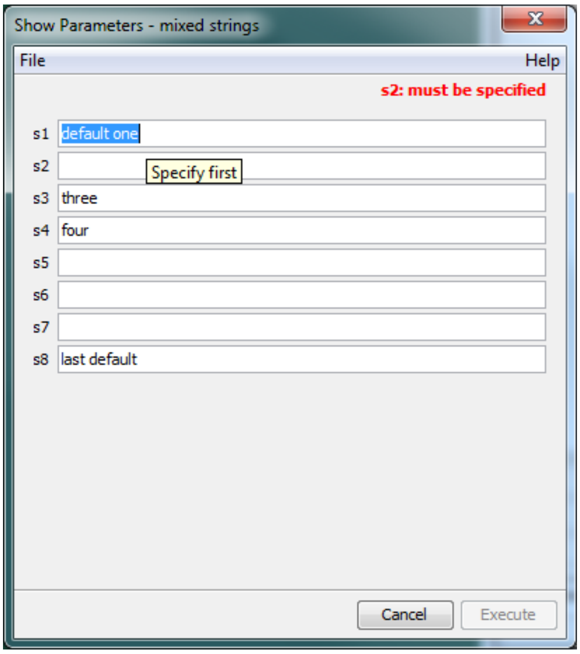
string type parameter:

- can be specified as being an optional parameter. If so, the basic checks performed by the tool will allow its value to be unspecified (blank). In that case, upon execution a single blank character will be passed to the executable;
- optionally a default value can be assigned. If so, this value will be filled out in the UI upon its initial display where it can be overwritten or left as is;
- for non-optional parameters, the value will be checked to be an non empty string.

Tool parameters

Tag	Type	Name	Opti…	Default/Value	Ext.	Exists	Inc….	ToolTip
P1	string	s1	☑	default one				Specify first
P2	string	s2	☐					enter second
P3	string	s3	☑	three				third as well
P4	string	s4	☐	four				nr 4
P5	string	s5	☑					string nr 5
P6	string	s6	☑					6th
P7	string	s7	☑					7th
P8	string	s8	☑	last default				Last Tip

example: create user tool string parameters: optional/mandatory, default values and tooltips



resulting user tool UI

Checks: parameter “s2” is mandatory.

Tooltips can be specified.

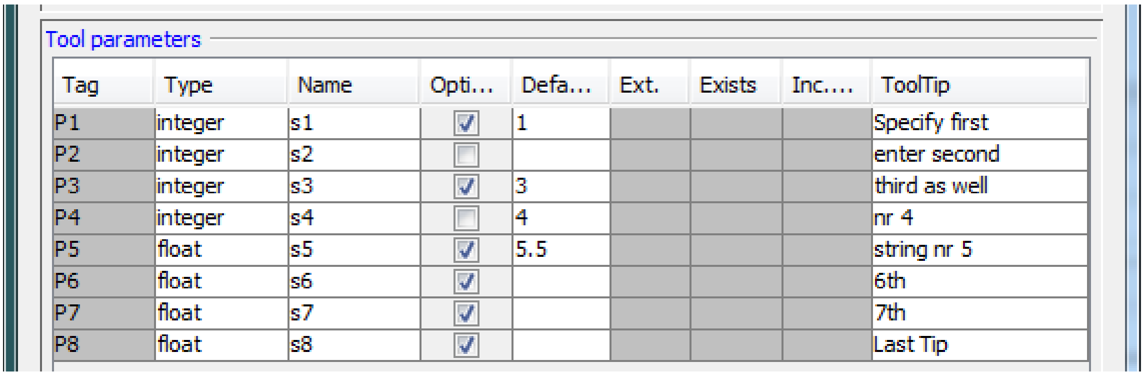
Default values can be specified.

integer type parameter:

- can be specified as being an optional parameter;
- optionally a default value can be assigned;
- non-optional parameters will be checked to contain an integer value.

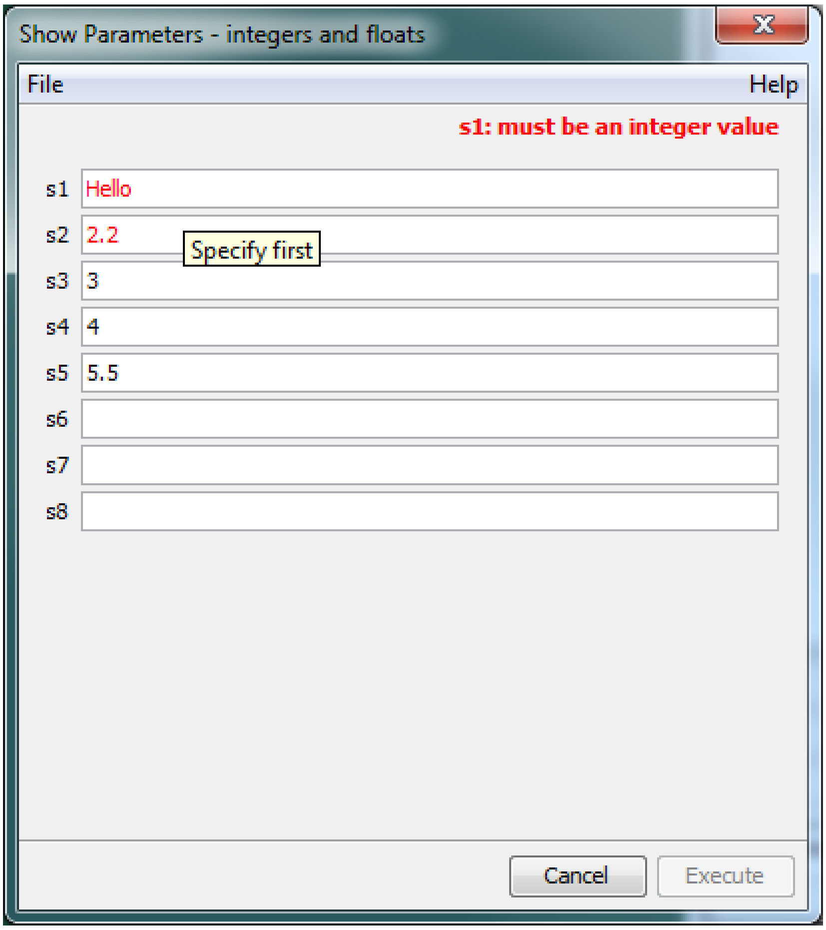
float type parameter:

- can be specified as being an optional parameter;
- optionally a default value can be assigned;
- non-optional parameters will be checked to contain a numerical value.


Tool parameters

Tag	Type	Name	Opti…	Defa…	Ext.	Exists	Inc.…	ToolTip
P1	integer	s1	☑	1				Specify first
P2	integer	s2	☐					enter second
P3	integer	s3	☑	3				third as well
P4	integer	s4	☐	4				nr 4
P5	float	s5	☑	5.5				string nr 5
P6	float	s6	☑					6th
P7	float	s7	☑					7th
P8	float	s8	☑					Last Tip

example: create user tool numeric parameters: optional/mandatory, default values and tooltips



resulting user tool UI

Checks:
parameter “s1” cannot be a string
parameter “s2” cannot be a float

Tooltips can be specified.

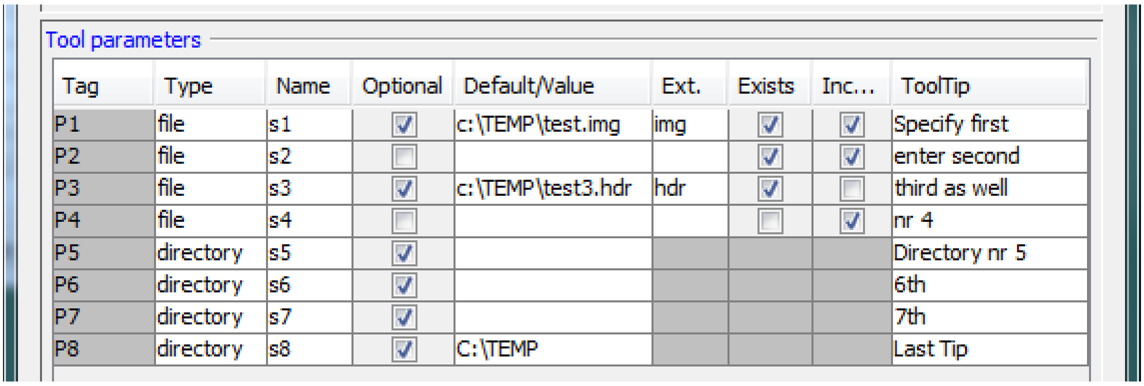
Default values can be specified.

file type parameter:

- can be specified as being optional;
- optionally a default value can be assigned;
- optionally a (mandatory) file extension can be specified;
- it can be specified if it must be an existing file;
- it can be specified whether the filename value, passed to the executable must or may not include its extension. (Certain executables (e.g. GLIMPSE) expect filename parameters without extensions);
- checked to be a valid filename, and if applicable, existing and with the extension specified.

directory type parameter:

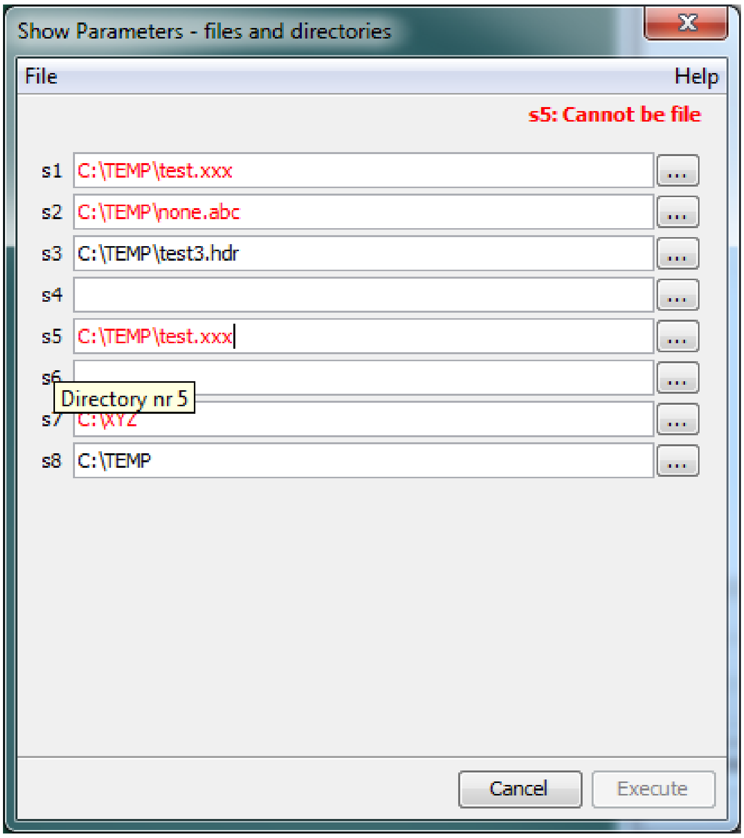
- can be specified as being optional;
- optionally a default value can be assigned;
- checked to be a valid existing directory.



Tool parameters

Tag	Type	Name	Optional	Default/Value	Ext.	Exists	Inc...	ToolTip
P1	file	s1	☑	c:\TEMP\test.img	img	☑	☑	Specify first
P2	file	s2	☐			☑	☑	enter second
P3	file	s3	☑	c:\TEMP\test3.hdr	hdr	☑	☐	third as well
P4	file	s4	☐			☐	☑	nr 4
P5	directory	s5	☑					Directory nr 5
P6	directory	s6	☑					6th
P7	directory	s7	☑					7th
P8	directory	s8	☑	C:\TEMP				Last Tip

example: create user tool file and directory parameters



resulting user tool UI

Checks:
parameter “s1” must be “.img” file
parameter “s2” must be existing file
parameter “s4” must be specified
parameter “s5” must be directory, not file
parameter “s7” must be existing directory

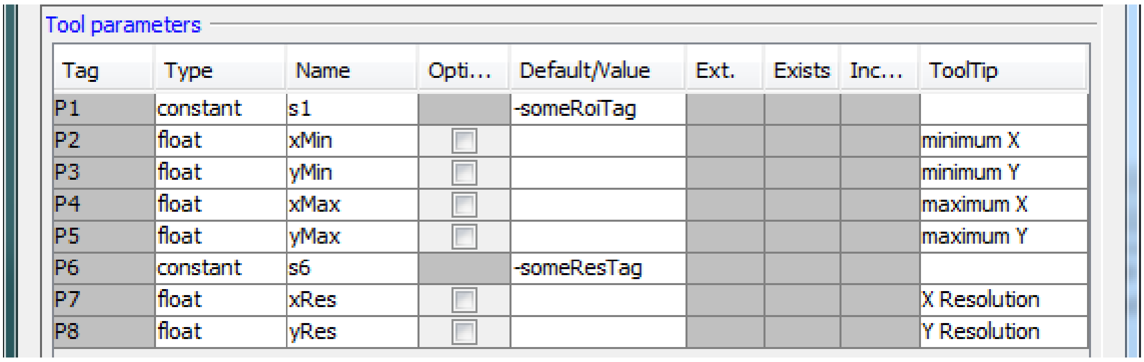
Tooltips can be specified.

Default values can be specified.

The UI will offer a file chooser button.

constant and prefix type parameter:

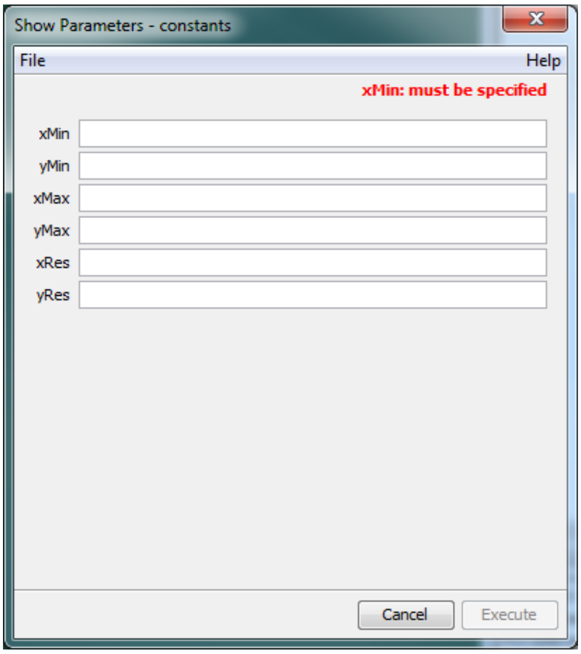
- a (default) value is mandatory, it cannot be modified in the UI.
- constants are typically used for constant values, constant files or (constant) options, switches and/or tags typical for the executable. Example:. for an executable using a syntax of the format:
  someExecutable -someRoiTag Xmin Ymin Xmax Ymax –someResTag Dx Dy
  the tags "-someRoiTag" and "–someResTag" could be specified as constant parameters.
- the difference between prefixes and constants is that a prefix will be concatenated with the next parameter (without whitespace in between). Prefixes are typically used for options, switches and/or tags typical for the executable, which require additional (variable) information, but do not accept whitespace in between. Example:. for an executable using a syntax of the format:
  someExecutable -src=sourcefile -dst=destfile, the keys "**-src=**" and "**-dst=**" can be specified as prefix parameters.



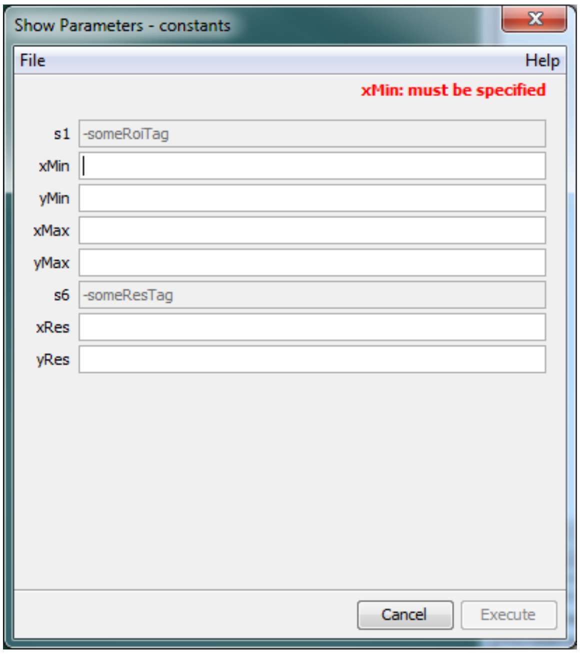
Tool parameters

Tag	Type	Name	Opti…	Default/Value	Ext.	Exists	Inc…	ToolTip
P1	constant	s1		-someRoiTag				
P2	float	xMin	☐					minimum X
P3	float	yMin	☐					minimum Y
P4	float	xMax	☐					maximum X
P5	float	yMax	☐					maximum Y
P6	constant	s6		-someResTag				
P7	float	xRes	☐					X Resolution
P8	float	yRes	☐					Y Resolution

example: create user tool with constant and numeric parameters



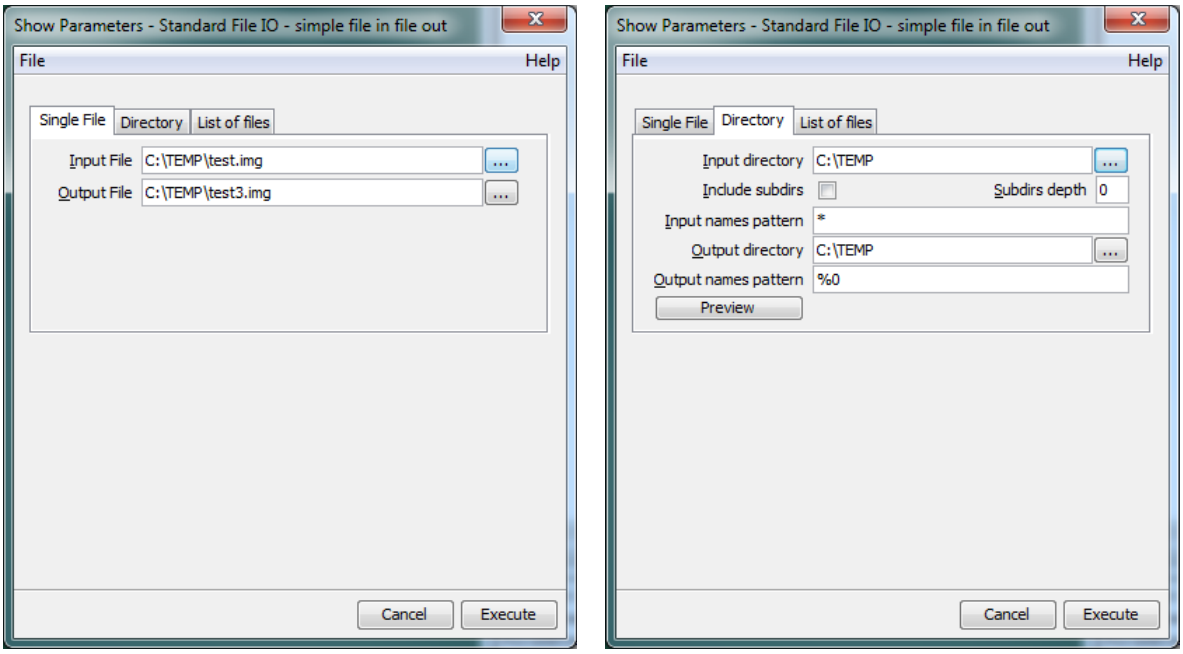
resulting user tool UI – created with option "Tool UI shows constants" not selected



resulting user tool UI – created with option "Tool UI shows constants" selected

pattern type parameter:

The pattern type parameter is allowed only for user tools, created with the “Use standard I/O file selection” option. The UI will then contain the typical Spirits tools panel used to specify input and output files.



typical user tool UI created with option “Use standard IO file selection” selected

These tools focus on typical “file-in – file-out” applications. The UI will allow selection of single input and output files, selection via input and output directories using an (input) wildcard pattern, using '*' and '?' as wildcards and (output) parameterized strings containing parameters "%0", "%1,..."%9" which will be replaced by the values of the corresponding wildcard for the matching input file names, and selection via a list of input files (.LNL).

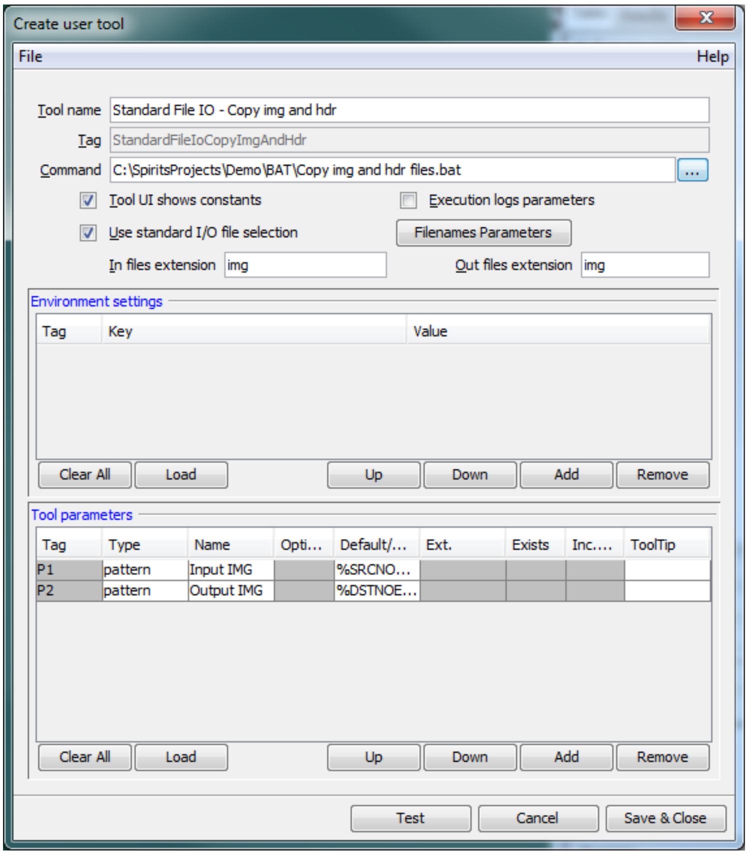
The selected input and output files can then be used as parameters for the selected executable by means of the pattern type parameters:

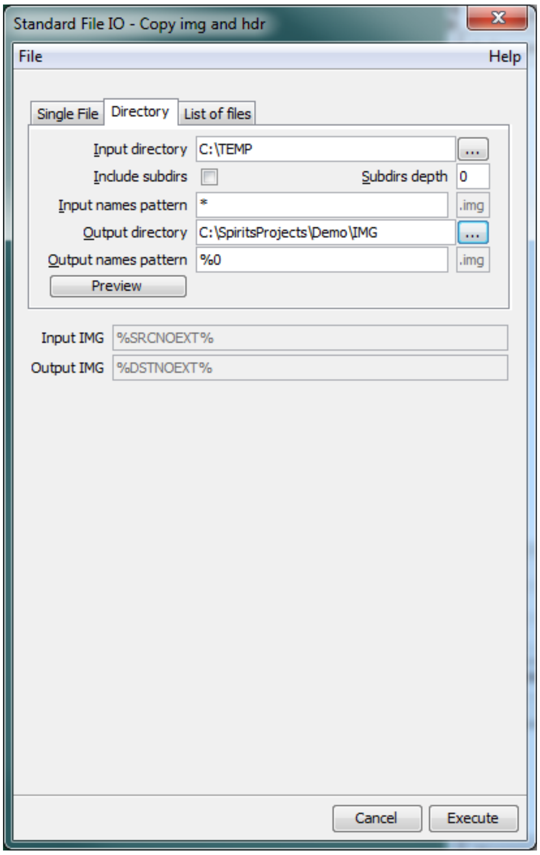
- can only be used in case the tool (and its UI) uses the “standard I/O file selection”;
- a (default) value is mandatory, it cannot be modified in the UI.
- this value must contain one of the available symbolic in/out file constants:

from input file	from output file	value	example
%SRC%	%DST%	full filename	C:\MyData\MyFile.xyz
%SRCNOEXT%	%DSTNOEXT%	full filename without extension	C:\MyData\MyFile
%SRCDIR%	%DSTDIR%	directory only	C:\MyData
%SRCFILE%	%DSTFILE%	filename only	MyFile.xyz
%SRCFILENOEXT%	%DSTFILENOEXT%	filename only without extension	MyFile
%SRCFILEEXT%	%DSTFILEEXT%	extension only	xyz

```
::
::    Copy img and hdr
::    %1 = input file  - no extension - assume .img and .hdr exist
::    %2 = output file - no extension
::
@echo off
copy %1.img %2.img
copy %1.hdr %2.hdr
```

example: external executable: “Copy img and hdr files.bat”





example: create user tool calling “Copy img and hdr files.bat” resulting user tool UI

Remarks:

- in some cases the actual command does not need an output file, but since the (generic) UI insists on one, some “dummy” file could be selected;
- depending on the actual command, filenames containing whitespace, may not be or must be enclosed by quotes. The substitution does NOT add quotes. if needed they should be added explicitly.
- the symbolic in/out file constants can be combined if needed, e.g. ***%SRC%*** gives the same value as ***%SRCDIR%\%SRCFILENOEXT%.%SRCFILEEXT%***.

6.2.5. User Tools examples

Extracting ZIP archives using 7-Zip

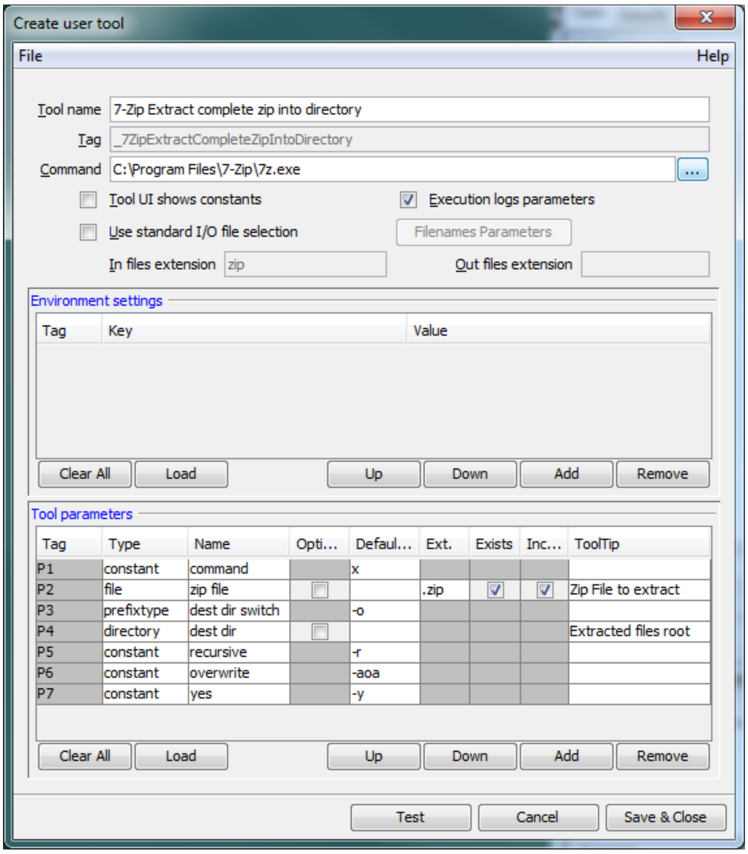
For the 7-Zip info an syntax - see the Command Line Tool example.

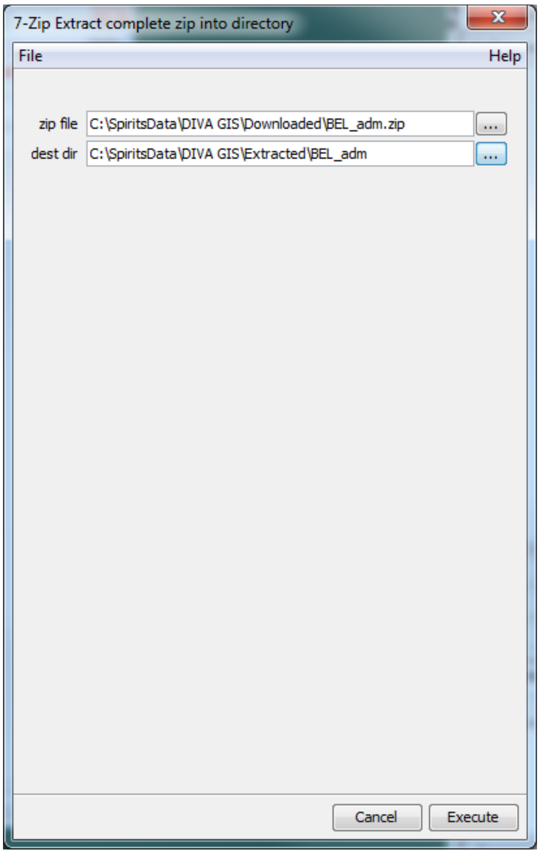
In this example a simple tool is created which allows selection of a .zip file and extracts it completely in a specified directory.

```
7z x <zipfile> -o<outputdir> -r -aoa -y
```

Remark: according to the 7-Zip syntax, "-y" assumes Yes on all queries. "-aoa" allows to overwrite existing files. This "kind" of options are important when creating user task; they prevent “hanging”. E.g. when running this task twice (without these switches), the output files would already exist, hence 7-Zip would stop and wait for user input, so the task would “hang” and block the task queue.

Since 7-Zip does not allow whitespace between the -o switch and the output directory name, "-o" needs to be specified as a prefix type.





example: creating 7-zip user tool resulting 7-Zip user tool

Sending INFO WARNING and ERRORS

Just like 'normal' Spirits tasks, User Tool Tasks are executed via the worker thread of the task queue. If needed, they can pass information to the task queue to report errors, warnings or progress. This communication follows a simple protocol:

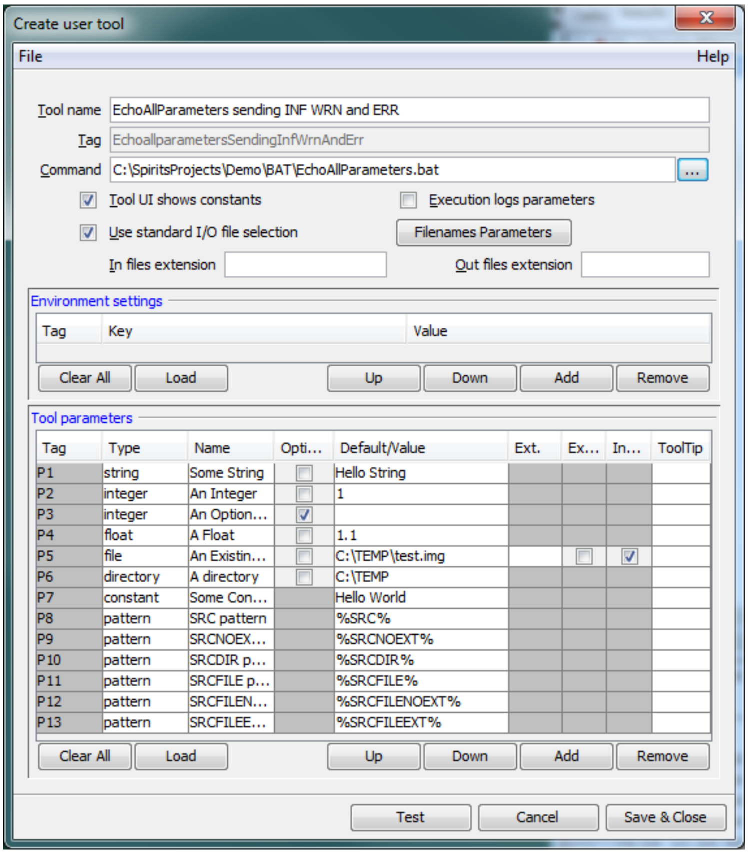
Errors:	**<error>**The actual error text**</error>**
Warnings:	**<warning>**The actual warning text**</warning>**
Information:	**<info>**The actual information text**</info >**
Progress:	**<progress percentage=**"xxx"**/>**

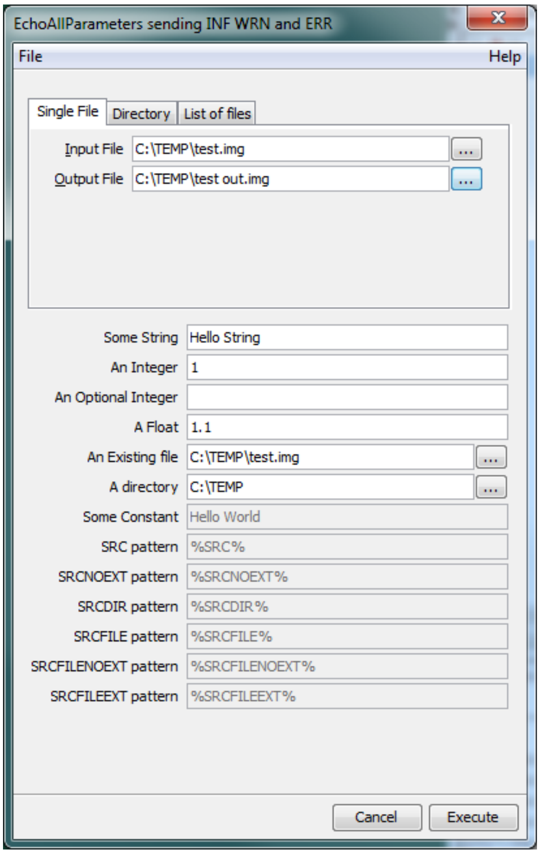
To send these **<error>**, **<warning>**... tags from a .BAT or .CMD file, care must be taken to "escape" the **<, /** and **>** characters, since -unless escaped- they have a dedicated meaning in script files themselves.

In a .BAT file, they can be escaped by prefixing them with the ^ character as in example below.

```
@ECHO off
ECHO ----------------------------------------------------------------

SET receivedatleastonenonblank=0

:LabelHandleNextParameter
IF "%~1"==""  GOTO LabelDoneThemAll

IF "%~1"==" " GOTO LabelWarnBlankParameter
ECHO ^<info^>Received ( non blank ) parameter: %~1^<^/info^>
SET receivedatleastonenonblank=1
GOTO LabelShiftToNextParameter

:LabelWarnBlankParameter
ECHO ^<warning^>Received Whitespace parameter^<^/warning^>

:LabelShiftToNextParameter
SHIFT
GOTO LabelHandleNextParameter

:LabelDoneThemAll
IF %receivedatleastonenonblank%==1 GOTO LabelExit
ECHO ^<error^>Received ONLY Whitespace parameters^<^/error^>

:LabelExit
ECHO ^<info^>Ready^<^/info^>

ECHO ----------------------------------------------------------------
```

example: external executable: “EchoAllParameters.bat”





example: create user tool calling "EchoAllParameters.bat" resulting user tool UI

```
2017/04/26 10:35:54 STATE:   RUNNING

----------------------------------------------------------------

2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: Hello String
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: 1
2017/04/26 10:35:54 WARNING: Received Whitespace parameter
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: 1.1
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: C:\TEMP\test.img
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: C:\TEMP
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: Hello World
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: C:\TEMP\test.img
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: C:\TEMP\test
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: C:\TEMP
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: test.img
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: test
2017/04/26 10:35:54 INFO:    Received ( non blank ) parameter: img
2017/04/26 10:35:54 INFO:    Ready

----------------------------------------------------------------
2017/04/26 10:35:54 STATE:   DONE
```

example: execution output in Task results queue

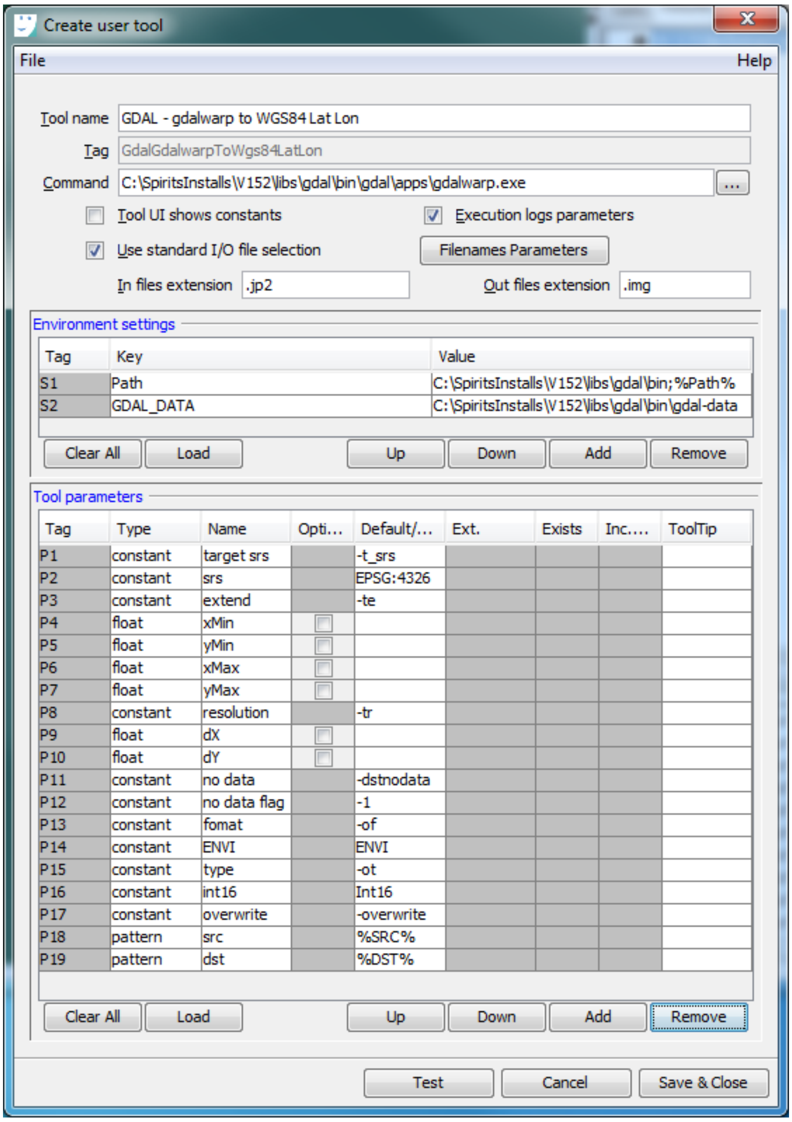
Reprojecting and converting using gdalwarp

In this example the gdalwarp tool from the GDAL Utilities is used, to reproject the JPEG2000 images, from a Sentinel 2 Level 1C product, to WGS84 and convert them to ENVI format.

The gdal_ gdalwarp syntax used is as follows:

```
gdalwarp
  -t_srs srs_def                target spatial reference set
  -te xmin ymin xmax ymax       georeferenced extents (in target SRS)
  -tr xres yres                 target resolution (in target SRS)
  -dstnodata                    nodata values for output bands
  -of format                    output format
  -ot type                      output data type
  -overwrite                    overwrite the target if it exists
  <src_datasource>              datasource
  <dst_filename>                output file
```

This line can be implemented into a user tool as shown below:



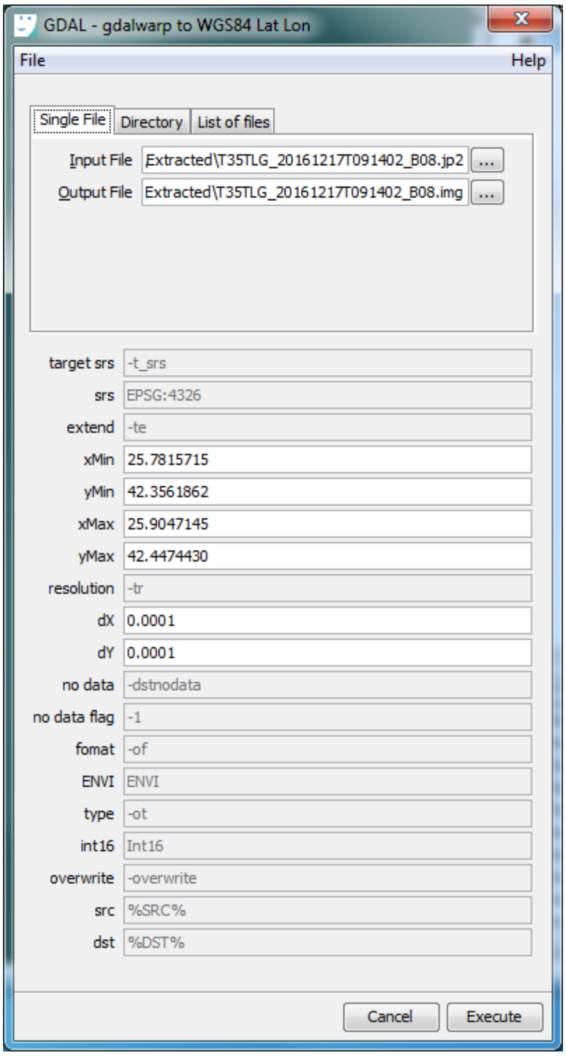
- fixed target SRS: EPSG:4326

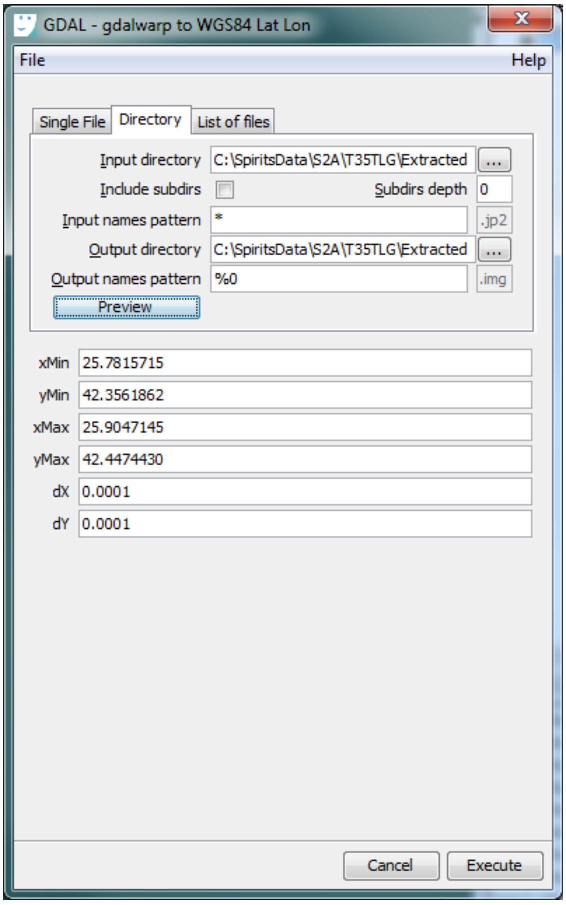
- ROI and resolution from user input

- fixed no-data value "-1" (Sentinel 2 JPEG2000 images contain only positive values, hence -1 can be used as flag)

- fixed output format and data type: ENVI - Int16

example: creating a user tool using gdalwarp





resulting user tool UI – created with option "Tool UI shows constants" selected

- created normally ("Tool UI shows constants" not selected)

Result:

original: UTM 35 N

reprojected: WGS84 Lat/Lon

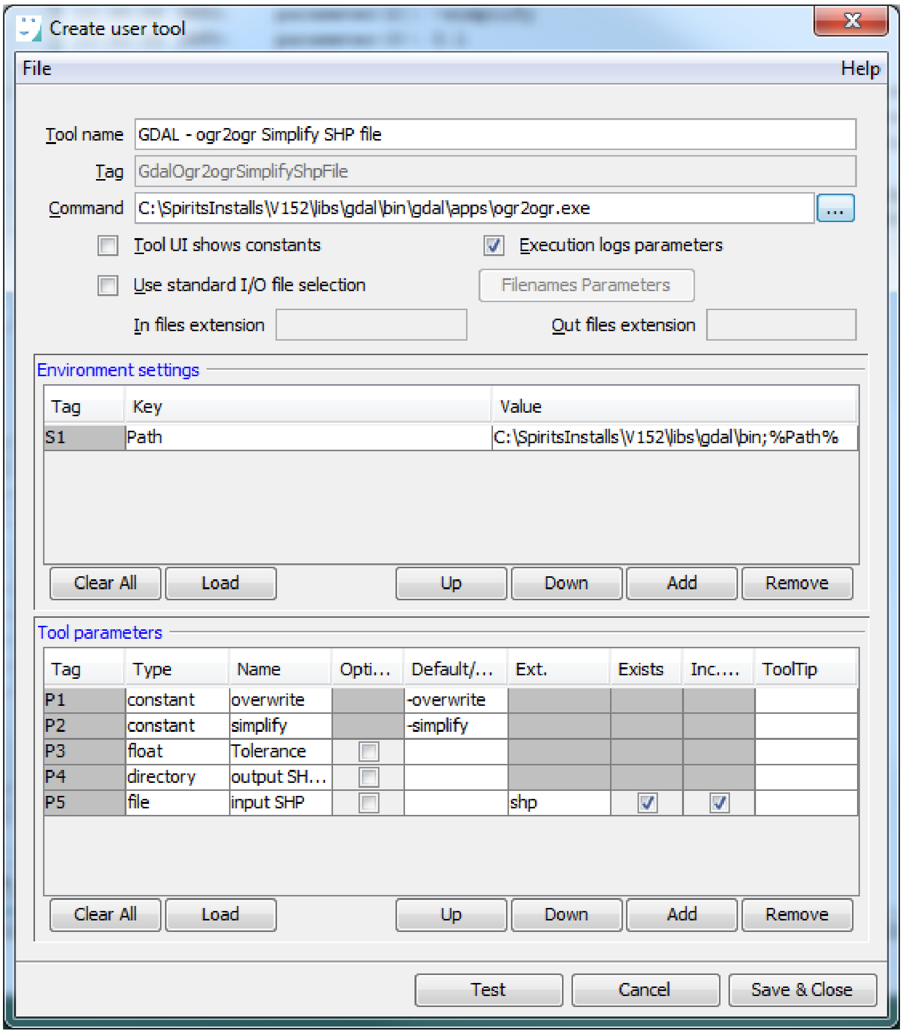
Simplify a SHP file using ogr2ogr example

In this example the ogr2ogr tool from the OGR Utility Programs (OGR is a part of the GDAL library) is used, to simplify SHP files. This can be used to reduce the size of SHP files in case no detailed information is needed (e.g. Quick Looks).

The settings and parameters necessary must be determined by studying the gdal documentation (ref. www.gdal.org/ogr2ogr.html). This indicates the basic syntax needed is as follows:

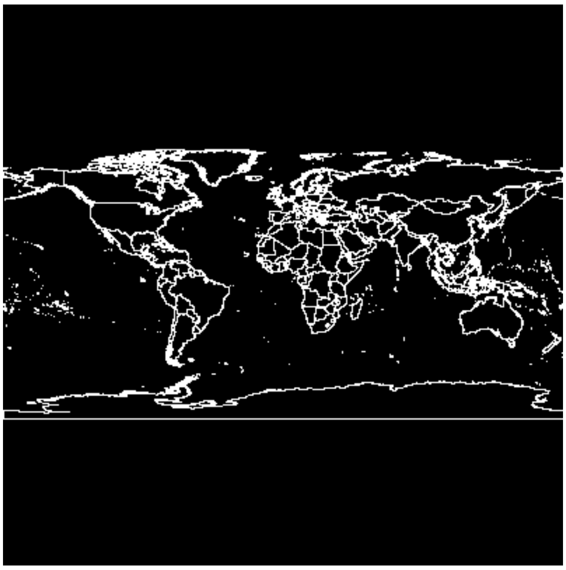
```
ogr2ogr -overwrite -simplify allowed_distance_tolerance
        destination_directory_name source_file_name
```

This command line can be implemented in a user tool as shown below:

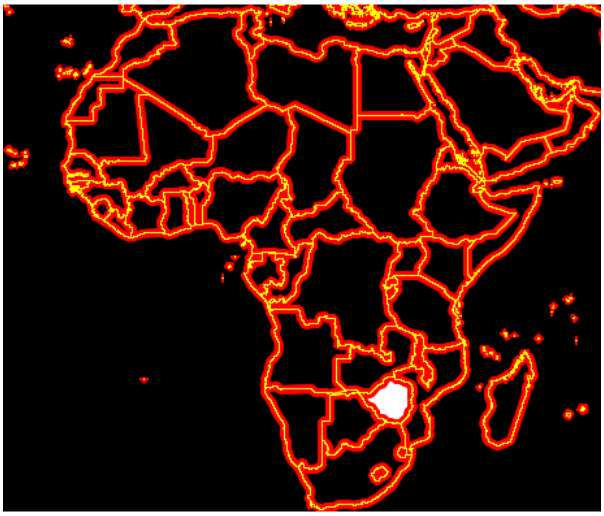


example: creating a user tool using ogr2ogr

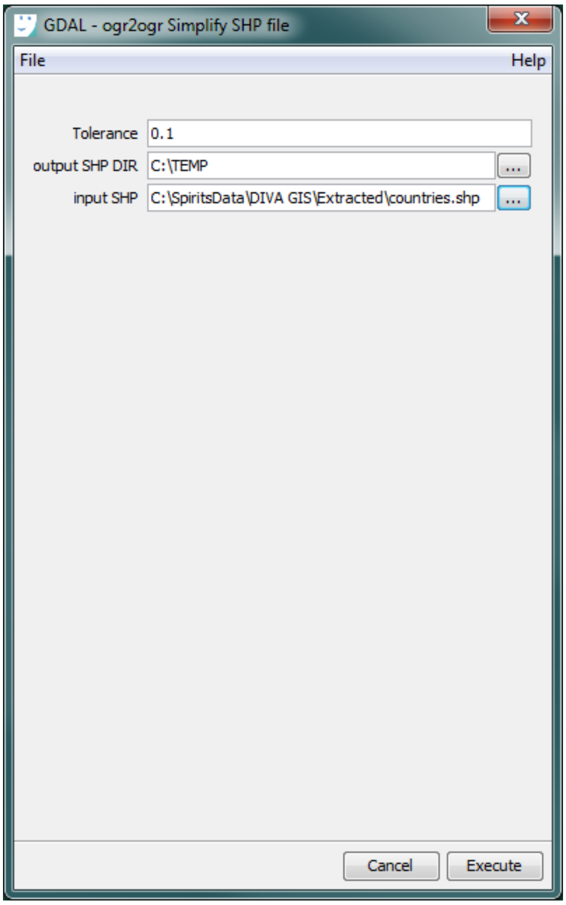
Results:



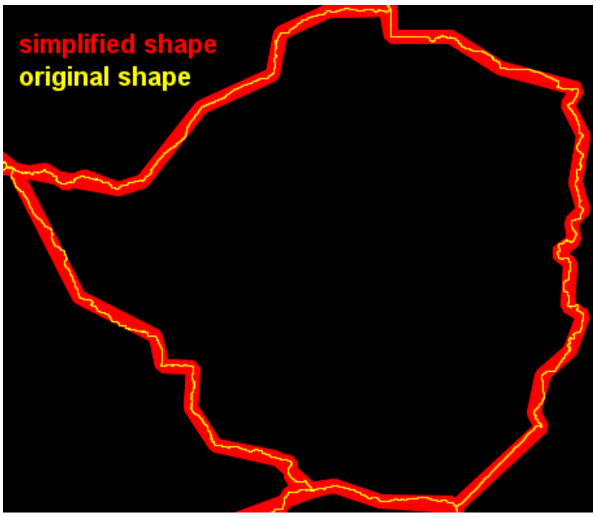
input SHP



simplified SHP - detail Africa



example: simplify SHP user tool



simplified SHP - detail Zimbabwe

7. Miscellaneous

7.1. Rename files

Goal

The Rename utility facilitates the renaming of files, especially files representing time series. To avoid risks, files will actually be copied under their new names, rather than being renamed "in place".

The utility has two distinct modes of operation:

- Rename mode
- Reformat dates mode

The Rename mode is meant for general copying of files under another name, whereas in the Reformat dates mode, the utility will attempt to extract date information from the input filenames, and copy the files under names respecting the "prefix-date format-suffix" (PiDiSi.ext) standard, used in time series.

7.1.2. Selecting the input files

The input files will be specified by their input directory and a wildcard pattern, using '*' and '?' as wildcards:

- '*' will be interpreted as 'any character, zero or more times';
- '?' will be interpreted as 'any character, exactly one time';
- all other characters in the wildcard pattern will be considered as constant.

All files in the specified input directory, with their filenames matching the pattern, will be selected to be copied. They will be shown at the left hand side of the Preview pane.

The search path can be extended via the option to include subdirectories (of the root directory) in the search, and if so the depth of this directory tree to be searched. To avoid memory problems and application freezes, this option should be used with care, and it is strongly advised to keep this depth small (e.g. 1 or 2. max is 3).

Examples:

- pattern "*" will select all files in the specified input directory;
- pattern "*.tif" will select all files with extension "tif" in the specified input directory;
- pattern "rfe*dk1*" will select all files with filenames starting with "rfe", and containing "dk1", in the specified input directory.

The wildcard pattern may contain up to ten wildcards. Each wildcard in the pattern, results in a corresponding parameter. For each selected file, the value of such parameter is the part of the filename covered by the wildcard. These parameters can be referred to as "%0", "%1,..."%9", and will be used to define the filenames of the copied files.

The interpretation of the wildcards pattern, and the resulting parameter values for each of the selected files, can be inspected in the Parameters pane.

7.1.3. Specifying the output files - Rename mode

In the Rename mode, the names for the output files will be specified by means of an output directory and a parameters pattern. This pattern can be a mixture of constant text and parameters ("%0", "%1,..."%9") obtained from the selected input files.

The output filenames will be shown at middle of the "Preview pane". In case of conflicts (duplicate filenames etc.) warnings will be shown at the right hand side of the "Preview pane".

Optionally one can replace constant substrings in the parameters with other constant strings (including empty strings). In the Substitute substrings panel, rules can be added, specifying the substring to replace, the parameter in which it should be replaced, and the new string. Only the first occurrence of the specified substring will be replaced.

7.1.4. Specifying the output files - Reformat dates mode

In the Reformat dates mode, the names for the output files will be specified as:

- an output directory;
- a prefix, specified by a parameters pattern - a mixture of constant text and parameters "%0", "%1,..."%9";
- the date, coded according one of the time series dates formats (YYYYMMDD, YYMMDD, ...);
- a suffix, specified by a parameters pattern - a mixture of constant text and parameters "%0", "%1,..."%9";
- the extension of the input file;

The date itself needs to be extracted from the input filename. This should be done by choosing an appropriate wildcards pattern for the input files, so the information necessary can be retrieved from the parameters.

The extraction is to be specified as follows:

First the periodicity of the files (Day, Dekad, Month or Year) is specified.

Depending on this periodicity, the utility decides which information it needs, and offers following possibilities to retrieve it:

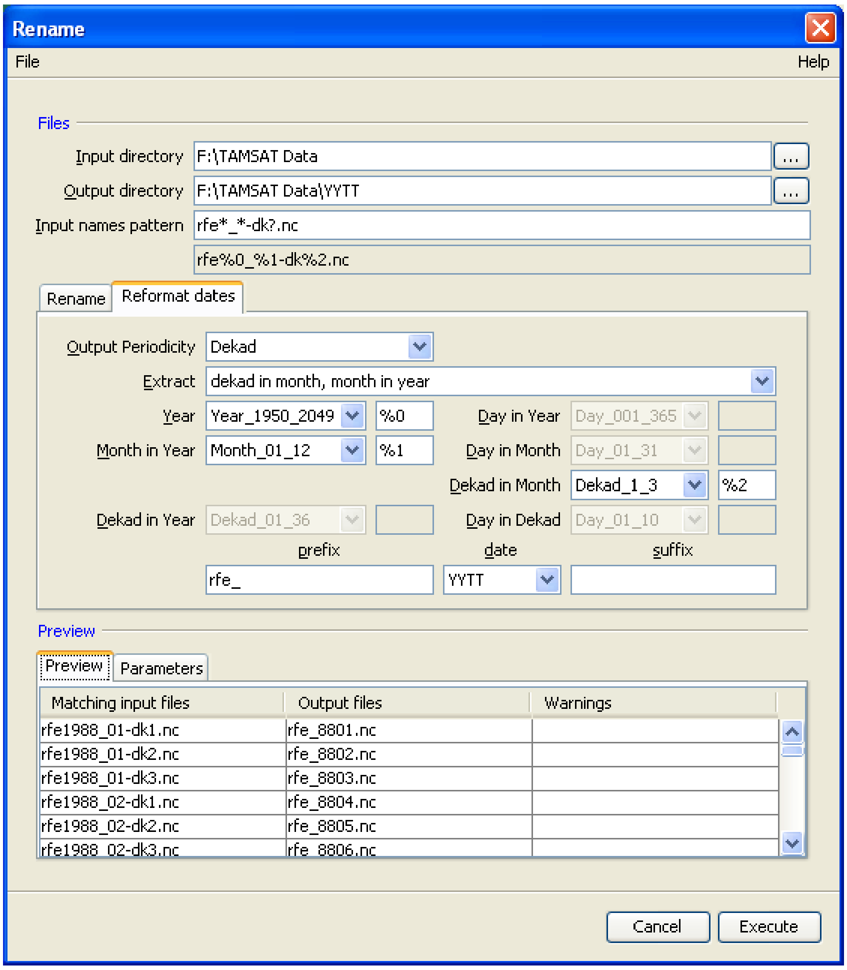
Periodicity	Information needed to determine a date	possible alternatives for retrieving the information necessary from the input filenames
Year	only the year is needed	year
Month	the month in the year and the year are needed	year + month in the year, OR year + dekad in the year, OR year + day in the year
Dekad	the dekad in the year and the year itself are needed	year + dekad in the year, OR year + dekad in the month + month in the year, OR year + day in the month + month in the year, OR year + day in the year
Day	the day in the year and the year itself are needed	year + day in the year, OR year + day in the month + month in the year year + day in the dekad + dekad in the year year + day in the dekad + dekad in the month + month in the year

Depending on the selected extraction method, the appropriate retrieval items (Year, Month in Year, Day in Year,...) are enabled, offering a selection of the different supported formats for the item:

Item	currently supported formats
Year	1950...2049 (four digits) or 50..49 (two digits)
Month in Year	01..12 (two digits) or 1..12 (one or two digits)
Dekad in Year	01..36 (two digits) or 1..36 (one or two digits)
Day in Year	001..365(three digits) or 1..365 (one, two or three digits)
Dekad in Month	1..3 (one digit)
Day in Month	1..31 (two digits) or 01..31 (one or two digits)
Day in Dekad	01..10 (two digits) or 1..10 (one or two digits)

Finally, for each item an appropriate selection must be made, and the string from which to retrieve it must be filled out. These strings are specified by parameters patterns - mixtures of constant text and parameters "%0", "%1,..."%9".

Rename TAMSAT RFE time series example



Tamsat file name convention:

rfeYYYY_MM[-dkD]

YYYY is the four-character year, [1983 to present]

MM is the two-character month, [01-12]

D is the one-character dekad, [1-3]

7.2. Create VAR/MTA files

Goal

Facilitate the creation of MTA files (ENVI metafiles) and VAR files (GLIMPSE metafiles).

Context:

Metafiles are simple ASCII files, basically with the names of the images which belong together for a certain application or analysis.

MTA files follow the ENVI metafiles format. The first line contains the string "ENVI META FILE". Then follow 3 lines per image layer, indicating the image file name, the band to select (in case the IMG is 3D) and the image window to consider.

VAR files are GLIMPSE specific metafiles. Each data line contains 3 information items for a specific image variable: a user-specified ID-number (Vu, greater than 0) in the 5 leftmost columns, the code of the variable (1-3 characters) in columns 6-10, and the image name (without extension!) from column 11 onwards. Lines whose 5 leftmost characters do not contain a value greater than zero, are considered as comments and skipped. The Vu-IDs and codes are mainly intended for the classification modules. In time series analyses, they become less relevant. Yet they may never be left blank.

```
GLIMPSE VAR-file with Periodic images,
---- ---- -----------------
  Vu CODE COMPLETE FILENAME
---- ---- -----------------
   1    1 c:\s1_meteo\wd20080501tmin
   2    2 c:\s1_meteo\wd20080502tmin
   3    3 c:\s1_meteo\wd20080503tmin
   4    4 c:\s1_meteo\wd20080504tmin
   5    5 c:\s1_meteo\wd20080505tmin

...

  25   25 c:\s1_meteo\wd20080525tmin
  26   26 c:\s1_meteo\wd20080526tmin
  27   27 c:\s1_meteo\wd20080527tmin
  28   28 c:\s1_meteo\wd20080528tmin
  29   29 c:\s1_meteo\wd20080529tmin
  30   30 c:\s1_meteo\wd20080530tmin
  31   31 c:\s1_meteo\wd20080531tmin
```

(part of) VAR file example

```
ENVI META FILE
File : c:\s1_meteo\wd20080501tmin.img
Bands: 1
Dims : 1-345,1-293

File : c:\s1_meteo\wd20080502tmin.img
Bands: 1
Dims : 1-345,1-293

File : c:\s1_meteo\wd20080503tmin.img
Bands: 1
Dims : 1-345,1-293

File : c:\s1_meteo\wd20080504tmin.img
Bands: 1
Dims : 1-345,1-293

...

File : c:\s1_meteo\wd20080530tmin.img
Bands: 1
Dims : 1-345,1-293

File : c:\s1_meteo\wd20080531tmin.img
Bands: 1
Dims : 1-345,1-293
```

(part of) MTA file example

Parameters

The tool has three distinct input modes:

starting from a time series of IMGs: parameters are:

- the directory containing the IMGs;
- the filenames prefix, suffix and dateformat;
- the periodicity of the IMG series;
- the first and last date of the IMGs to be included in the VAR and MTA files.

starting from a non periodic series of IMGs: parameters are:

- the (root) directory containing the IMGs;
- the wildcard pattern describing the filenames of the IMGs: a combination of constant characters with '*' and '?' wildcards:
  - '*' will be interpreted as 'any character, zero or more times';
  - '?' will be interpreted as 'any character, exactly one time';
  - all other characters in the wildcard pattern will be considered as constant.
- the option to include subdirectories (of the root directory containing the IMGs) in the search, and if so the depth of this directory tree to be searched. To avoid memory problems and application freezes, this option **should be used with care**, and it is strongly advised to **keep this depth small** (e.g. 1 or 2. max is 3).

All files in the specified input directory (optionally including its subdirectories), with their filenames matching the pattern, and having a (fixed) IMG extension, will be selected.

starting from an LNL file (List of files): parameters are:

- the LNL file containing the IMGs.

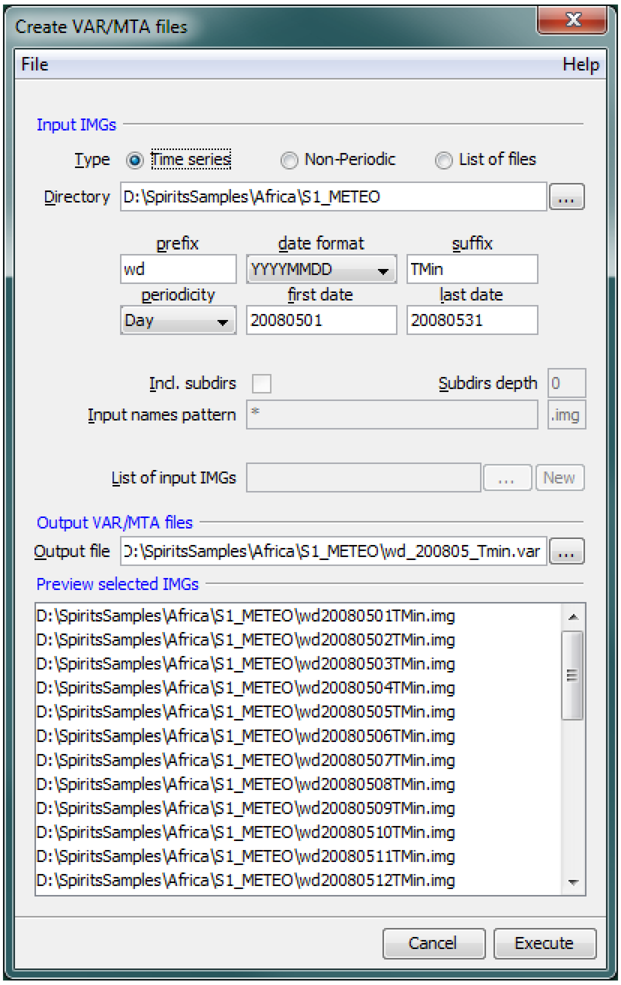
In each case the output file name must be specified.

By default the VAR file is specified, but both VAR and MTA files will be generated.

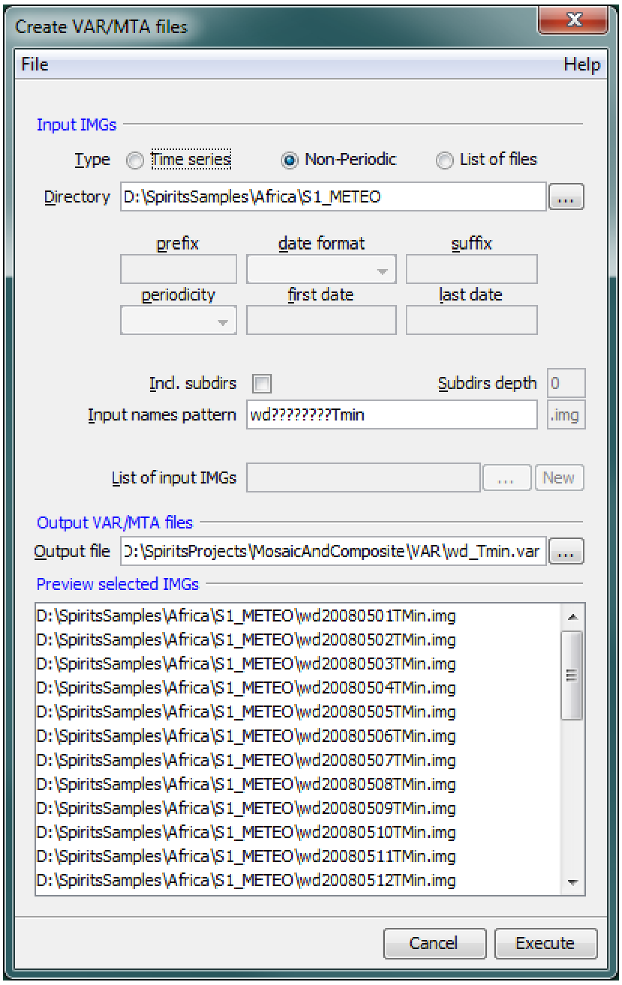
In the bottom half of the panel, a preview of the selected files is available.

Remark: since MTA and VAR files are ASCII files, they can be modified by any ASCII editor if needed.

Tool



Create VAR/MTA files utility example time series selection



Create VAR/MTA files utility example wildcards pattern selection

```
GLIMPSE VAR-file with Periodic images, created with VARmakeP.EXE (V1101/1009)
---- ---- -----------------
  Vu CODE COMPLETE FILENAME (without extension)
---- ---- -----------------
   1    1 d:\spiritssamples\africa\s1_meteo\wd20080501tmin
   2    2 d:\spiritssamples\africa\s1_meteo\wd20080502tmin
   3    3 d:\spiritssamples\africa\s1_meteo\wd20080503tmin
   4    4 d:\spiritssamples\africa\s1_meteo\wd20080504tmin
   5    5 d:\spiritssamples\africa\s1_meteo\wd20080505tmin
   6    6 d:\spiritssamples\africa\s1_meteo\wd20080506tmin
   7    7 d:\spiritssamples\africa\s1_meteo\wd20080507tmin
...
  23   23 d:\spiritssamples\africa\s1_meteo\wd20080523tmin
  24   24 d:\spiritssamples\africa\s1_meteo\wd20080524tmin
  25   25 d:\spiritssamples\africa\s1_meteo\wd20080525tmin
  26   26 d:\spiritssamples\africa\s1_meteo\wd20080526tmin
  27   27 d:\spiritssamples\africa\s1_meteo\wd20080527tmin
  28   28 d:\spiritssamples\africa\s1_meteo\wd20080528tmin
  29   29 d:\spiritssamples\africa\s1_meteo\wd20080529tmin
  30   30 d:\spiritssamples\africa\s1_meteo\wd20080530tmin
  31   31 d:\spiritssamples\africa\s1_meteo\wd20080531tmin
```

(part of) resulting VAR file

7.3. Help

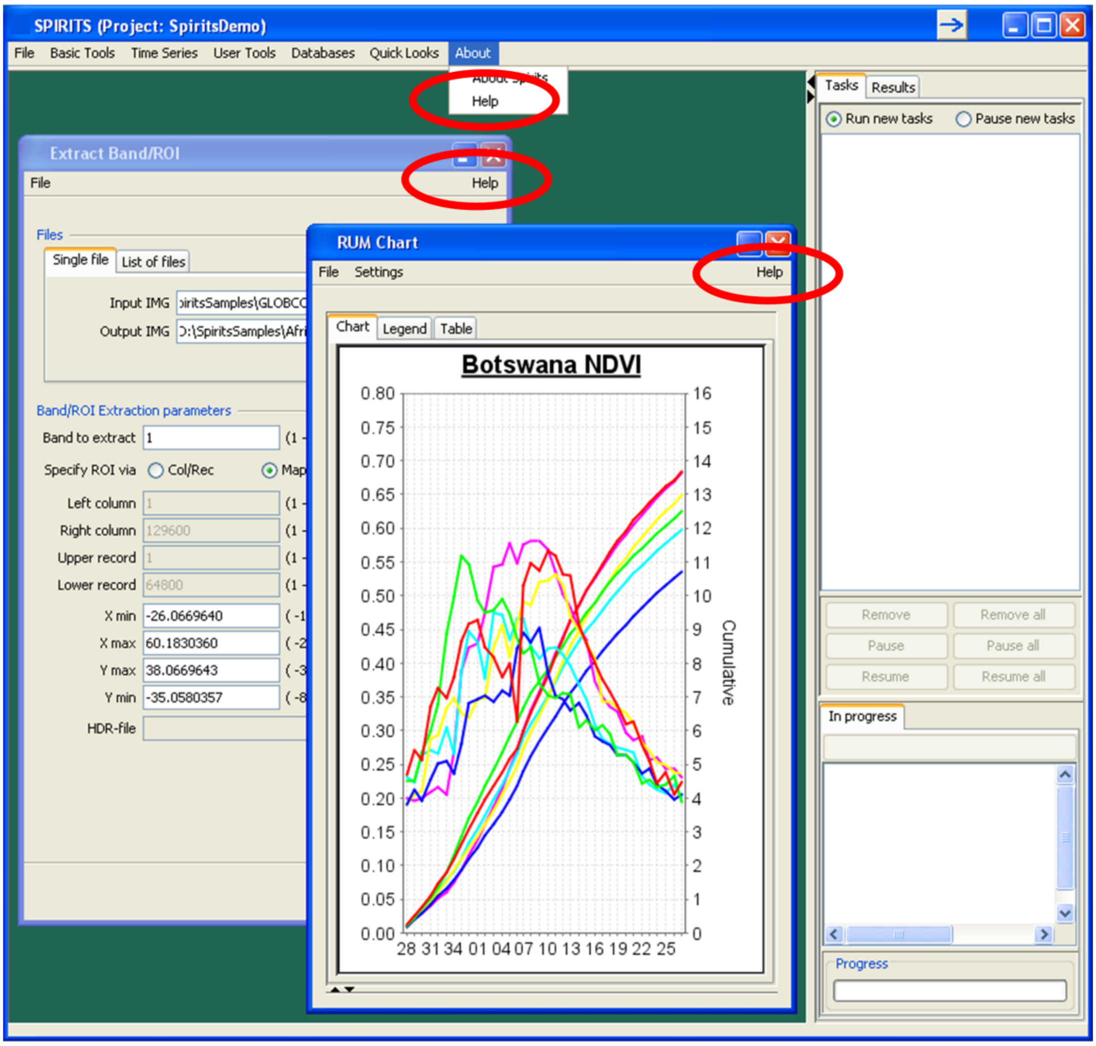
Goal

The Help function offers a basic on line help by linking the different tools and utilities to relevant entries in this manual and displaying them in the Help browser.

Usage

The Help browser can be opened from

- the Help item in the menu bar of most tools and utilities.
- the Help item in the About menu on the application main menu bar.



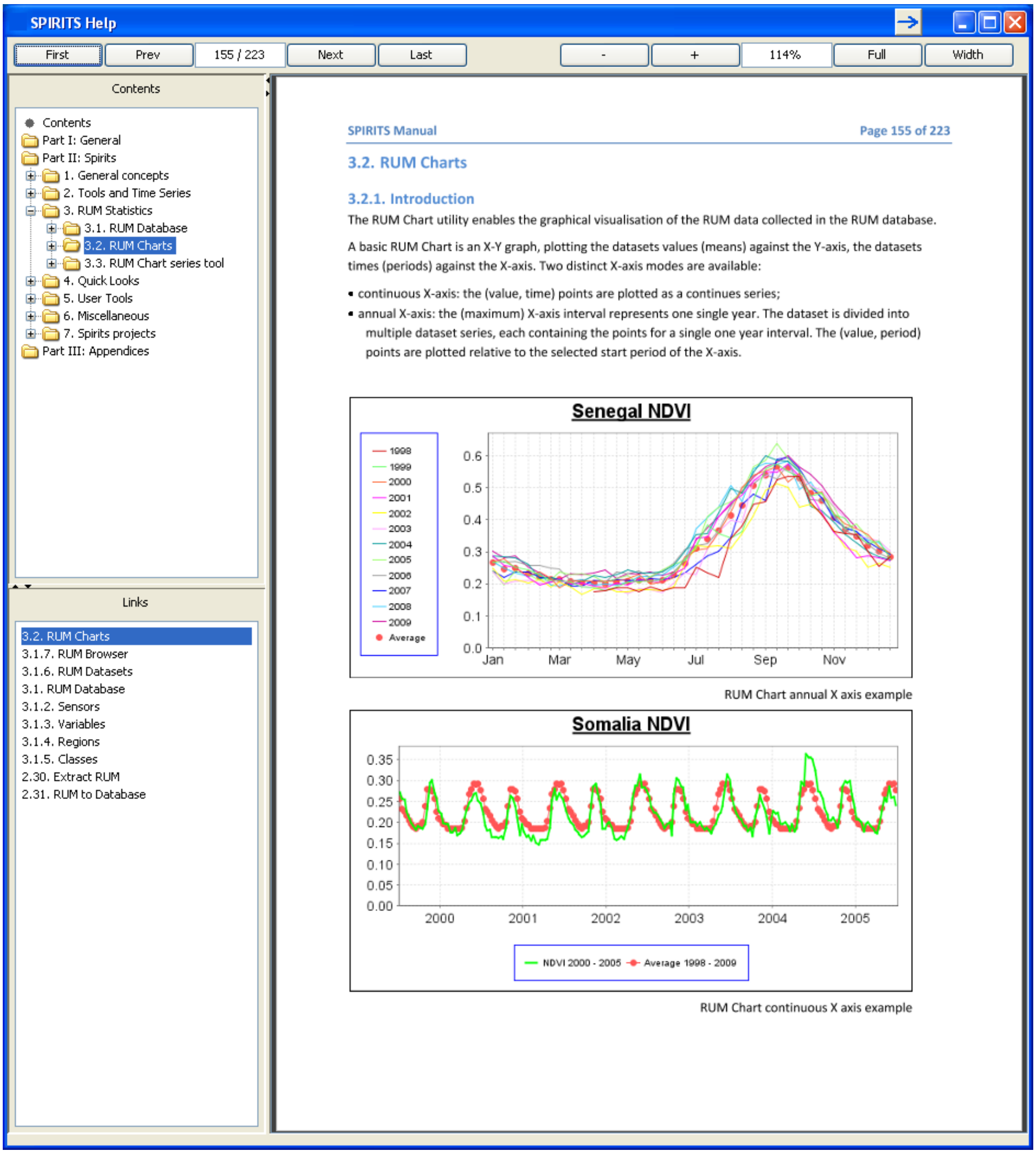
starting the Help utility

Starting the Help function from the main menu bar will open the browser positioned on the first page of the manual, while when starting it from a tool or utilities menu bar, the browser will open on a page related to that tool or utility

Browser

The Help browser consists of four subpanels:

- the commands panel containing command buttons and fields;
- the Contents panel showing the table of content of the manual;
- the Links panel showing the main link, followed by additional relevant links if any, for the tool or utility which triggered the Help function. This panel is empty in case the Help function was called from the main menu bar;
- the actual document panel, showing the selected page of the manual.

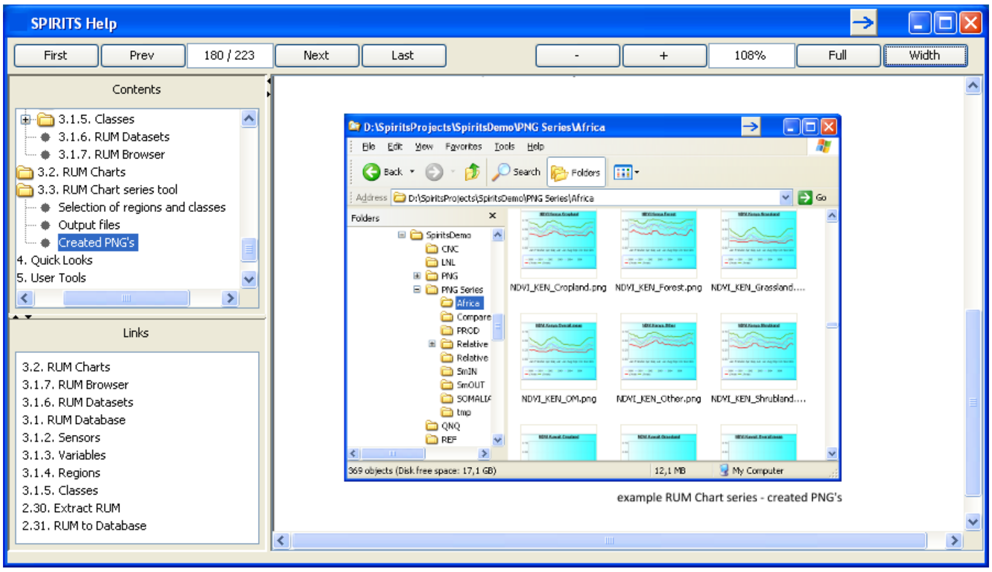


Help browser

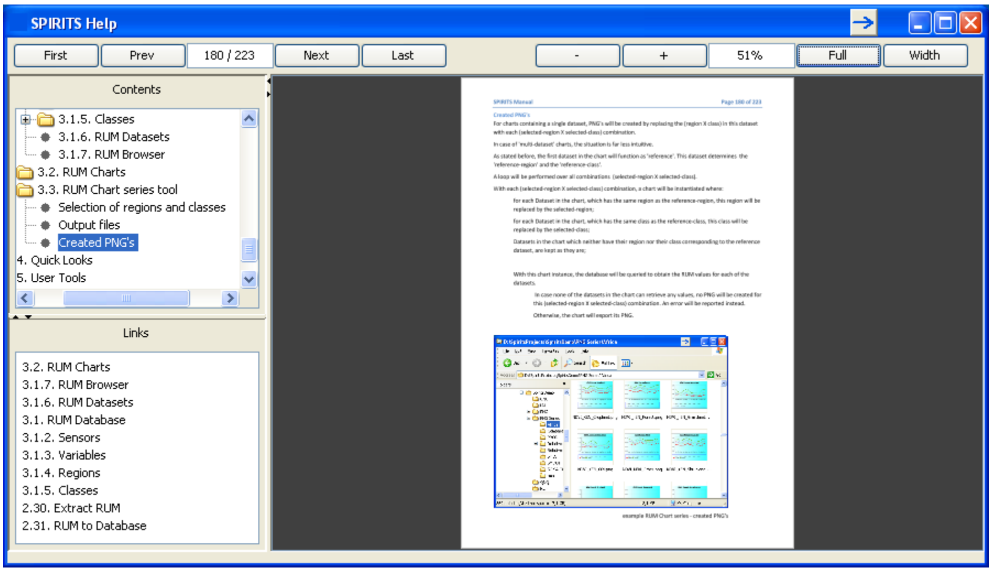
Via the Contents panel, the Links panel and the page number field in the commands panel, the required page can be selected.

When activating the document panel (by clicking in it), the next and previous pages can also be selected by using the PgDn and PgUp keys.

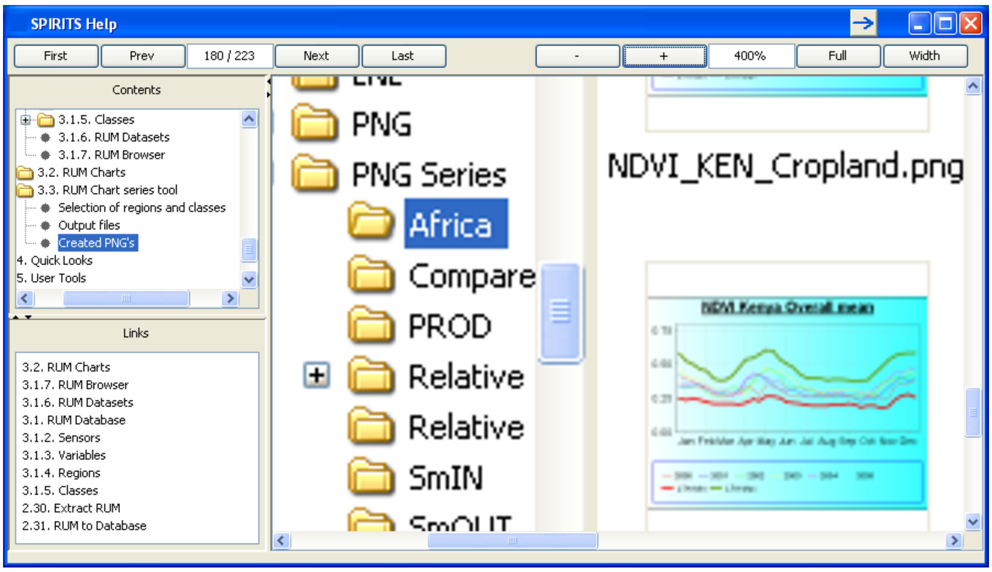
Within limits (25% - 400%) the zoom factor of the view can be increased, decreased and explicitly set via the "-", "+", "Full" and "Width" buttons, or the zoom field in the commands panel.



"Width" view



"Full" view



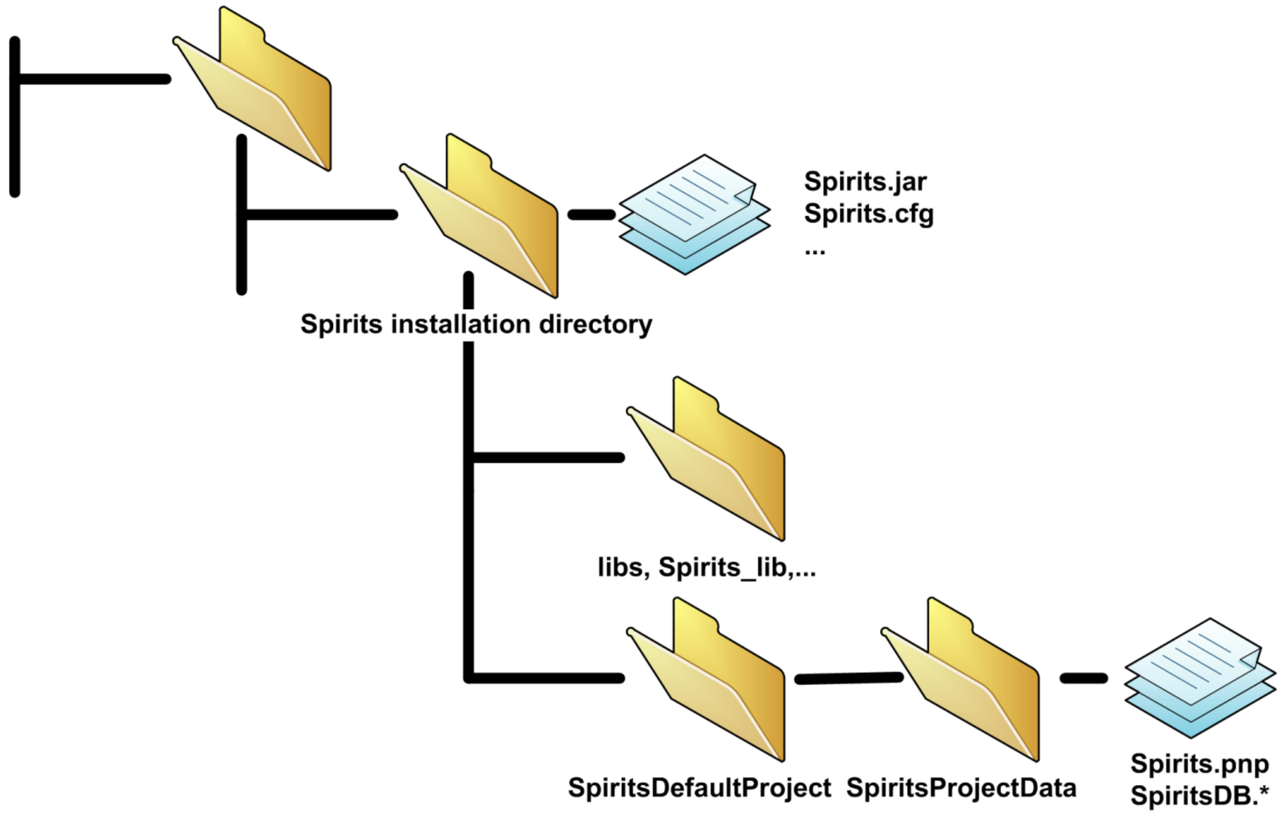
400% (maximum)

8. Spirits projects

8.1. Introduction

The projects concept offers the possibility to group related user data (IMGs, tools, scenario's, quick looks,...) without demanding a fixed or predefined file system structure.

Upon the first start of the application, a default project is created in the installation directory ('SpiritsDefaultProject'). Once additional projects have been defined and configured, users can swap between projects.



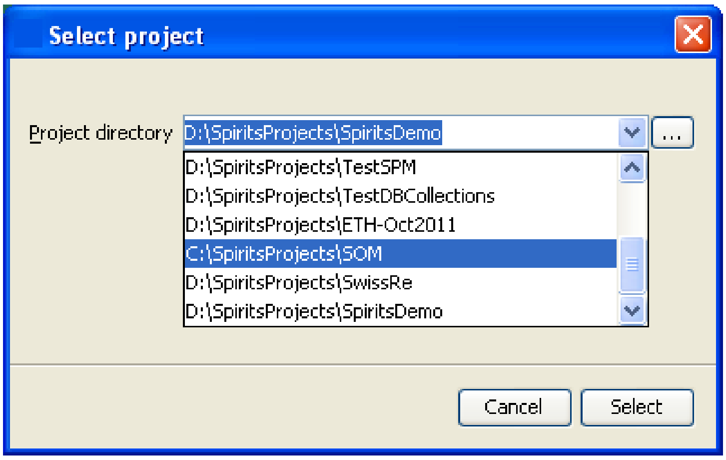
Spirits installation directories - SpiritsDefaultProject

Physically a project can be considered a directory on the file system. Upon creation of a new project, a directory has to be specified. In this directory the application will create a subdirectory ('SpiritsProjectData'), and in this subdirectory it will create an empty RUM database and a project settings file ('Spirits.pnp').

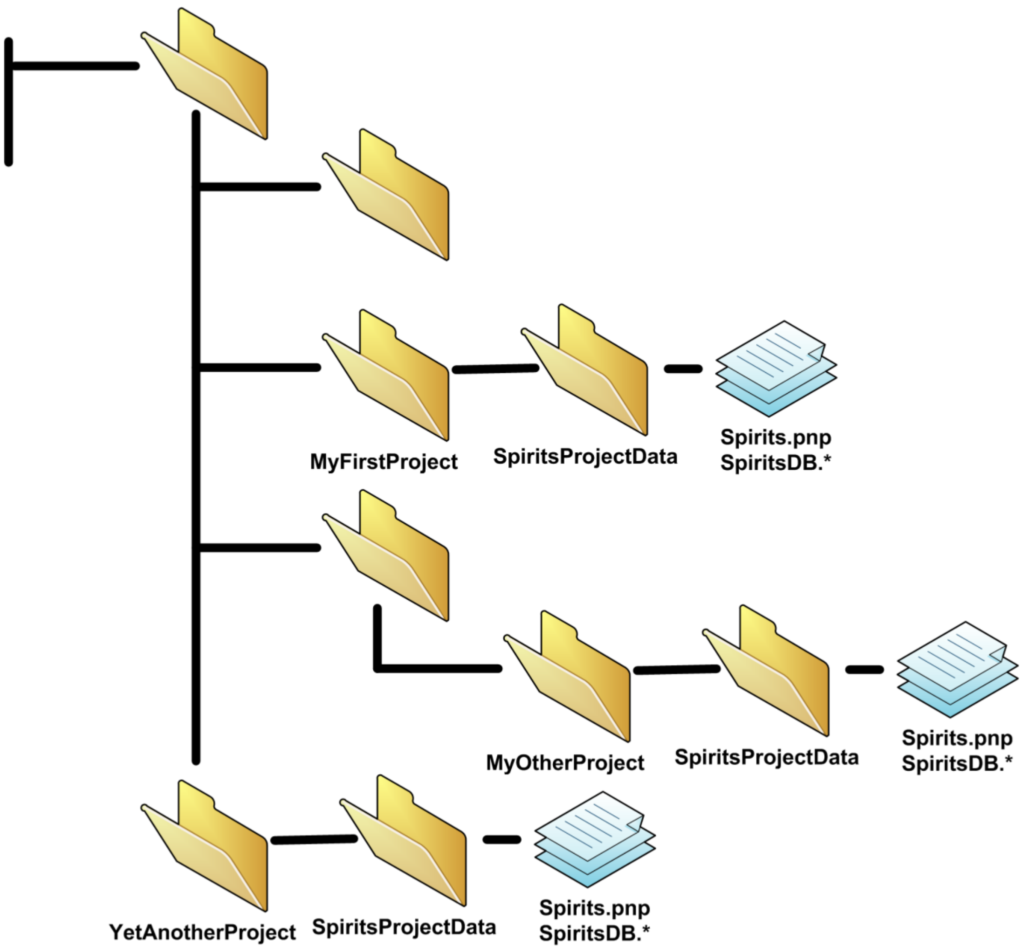
8.2. Selecting and creating projects

A list of directories, representing recently selected projects, is available in a dropdown list. Alternatively, via a directory chooser an existing directory can be chosen, or a new one can be specified.

In case the specified directory contains a project subdirectory ('SpiritsProjectData') and its settings file('Spirits.pnp'), the application swaps to this existing project. Otherwise, these files are created at the specified directory, thereby creating new project.



Project selection and creation



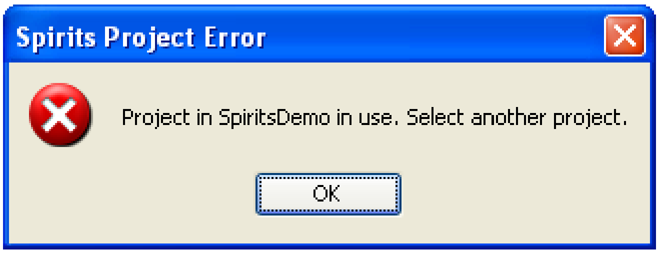
User created projects

Remark:

Although in principle it is possible to run multiple application instances simultaneously, a project can only be accessed by one application instance at a time.

By default the application will start up with the settings of the last active project.

When starting a second application instance which tries to access an active project a warning will be issued, and the user will be given the choice to select (or create) an other project.



8.3. Project settings

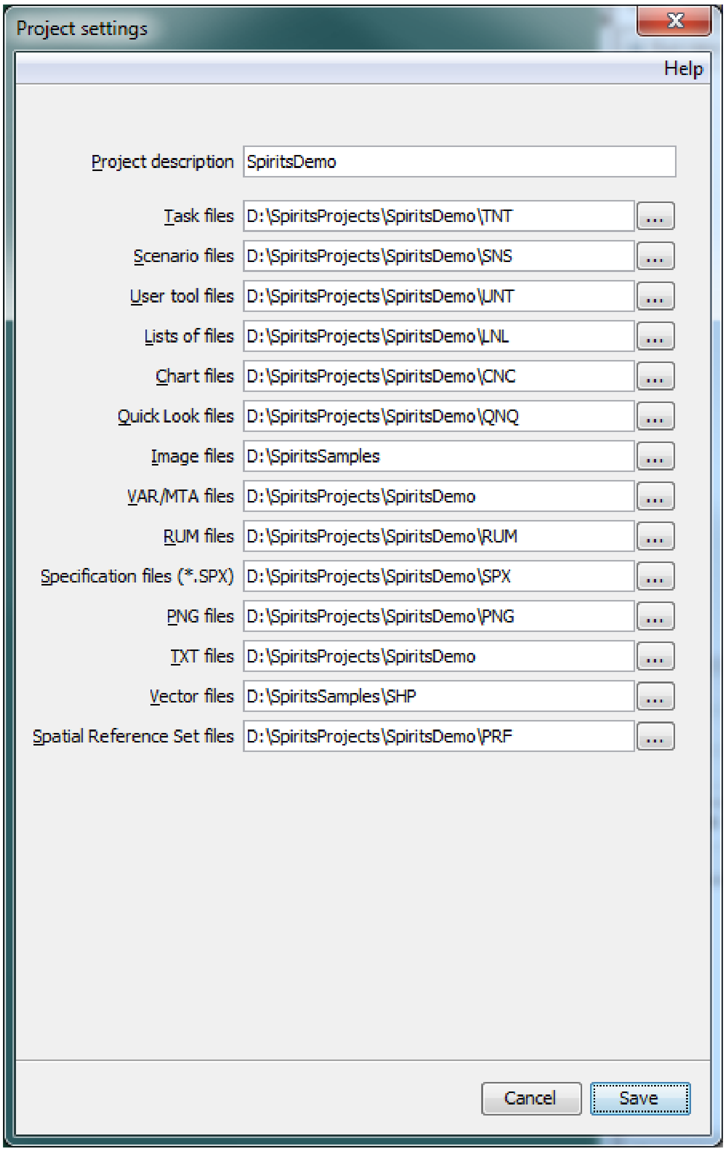
The project settings file contains:

- specific settings such as the RUM Chart Settings and Quick Look Settings, which can be specified by some tools and utilities;

- project settings: the project name and a collection of default paths per file type, for most of the file types used by the application: Task files (*.TNT), Scenario files (*.SNS), User tool files (*.UNT), Lists of files (*.LNL), Chart files (*.CNC and *.CNM), Quick Look files (*.QNQ), image files (*.IMG, *.HDR), VAR and MTA metafiles (*.VAR, *.MTA), Rum files (*.RUM), Specification files (*.SPC, *.SPP, *.SPS, *.SPU, *.SPM), PNG files (*.PNG), Text files (*.TXT) , vector files (*.SHP) and Spatial Reference Set files (*.PRF, *.WKT, *.PRJ).

All file choosers, spread across the different tools and utilities of the application, will use these default paths, thereby offering a the user experience of a dedicated environment per project.

Upon creation of a new project, these paths will be set to the project directory itself. It is then up to the user to configure the default paths for the project via the Project settings dialog.



Project settings dialog

Remarks:

A project can only be accessed by one application instance at a time, however this does not mean that tasks, scenarios, ...cannot be shared between application instances. Different projects could be set up for example to share the same default image or PNG locations.

The user is completely free to select the different default directories, however it is advisable choose some convention and stick to it over the different projects. A typical structure could be:

Task files	...\project_directory\TNT
Scenario files	...\project_directory\SNS
User tool files	...\project_directory\UNT
Lists of files	...\project_directory\LNL
Chart files	...\project_directory\CNC
Quick look files	...\project_directory\QNQ
...	...

Software for the Processing and Interpretation of Remotely sensed Image Time Series

USER'S MANUAL

Part III: Annexes

1. Spirits installation

System requirements

Spirits requires a Microsoft Windows system with a Java 7 Runtime Environment (JRE 1.7) or higher properly installed. (See http://www.oracle.com/technetwork/java/javase/downloads/index.html)

Installation

Spirits can be installed via a self extracting archive (SpiritsExtract.exe) or by (manually) extracting the ZIP archive (SpiritsExtract.zip). Following directories/files will be extracted:

Directory	Contents
.\SpiritsInstall	The Spirits executable jar (Spirits.jar) and additional jars (e.g. SpiritsCore.jar,...). Icons, the manual file, the epsg cross reference file and the Spirits configuration file (created at first run)
\libs	
\GLIMPSE	GLIMPSE executables
\gdal	GDAL libraries - unzipped GDAL release version as downloaded(zip) GDAL build SDK packages as downloaded (zip) ref: http://www.gdal.org/ ref: http://www.gisinternals.com/release.php
\hsqldb	HyperSQL package as downloaded(zip) and jar files. ref: http://hsqldb.org/
\jfreechart	JFreeChart library as downloaded(zip) and jar files. ref: http://www.jfree.org/jfreechart/
\jgoodies	JGoodies libraries as downloaded(zip) and jar files. ref: http://www.jgoodies.com/
\PDFRenderer	Pdf-renderer as downloaded(zip) and jar files. ref: http://java.net/projects/pdf-renderer/
\util	some sample bat files as debugging aids
\SpiritsDefaultProject	Spirits default project directory (created at first run)

2. Open Source Used in the SPIRITS Software

The SPIRITS software makes use of certain publicly available software. Below is the list of the publicly available software included in the SPIRITS software along with the licensing terms that pertain to these software programs. This list is for informational purposes only and is not intended to represent an exhaustive list of third party software contributions to the SPIRITS software.

(1) GDAL – Geospatial Data Abstration Library

GDAL is distributed under an X/MIT license reproduced below.

It is intended to give you permission to do whatever you want with the GDAL source code: download, modify, redistribute as you please, including building proprietary commercial software, no permission from Frank Warmerdam, OSGeo Foundation or or anyone else is required.

Copyright (c) 2000, Frank Warmerdam

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be include in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A more complete collection of license terms for GDAL/OGR and subcomponents is included within the software distribution in the LICENSE.TXT file.

(2) HSQLDB

HSQLDB (HyperSQL DataBase) is the leading SQL relational database engine written in Java.

Copyright (c) 2001-2010, The HSQL Development Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

(3) PDF-Renderer

The PDF Renderer is all Java library which renders PDF documents to the screen using Java2D. PDF-Renderer is licensed under the GNU Lesser General Public License, version 2.1 (LGPL-2.1). More information on LGPL-2.1 can be found at: http://opensource.org/licenses/lgpl-2.1.php.

(4) JFreeChart

JFreeChart is licensed under the GNU Lesser General Public License (LGPL). More information on the LGPL can be found at: http://www.gnu.org/licenses/lgpl.html.

(5) JGoodies : Jgoodies-common-1.2.1.jar and jgoodies-forms-1.4.2.jar.

The JGoodies are licensed under the terms of the BSD Open Source Licenseas specified below.

(6) HDF5 Library

---

(7) HDF Java Products

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois, Lawrence Livermore National Laboratory (LLNL), Sandia National Laboratories (SNL), Los Alamos National Laboratory (LANL), Fortner Software, Unidata Program Center (netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip), and Digital Equipment Corporation (DEC).

Portions of the HDF Java Products were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

---

3. List of Abbreviations

ABBREV.	MEANING
AFI	Area Fraction Image
AVHRR	Advanced Very High Resolution Radiometer, LR-sensor on-board of the NOAA and METOP satellites
BIL	Band Interleaved by Line, 3D (multiband) image format: first line 1 of band 1, then line 1 of band 2, etc. Then the same for the following lines.
BIP	Band Interleaved by Pixel, 3D-image format: first all spectral data of pixel 1, then those of pixel 2, etc.
BISE	Best Index Slope Extraction algorithm for cleaning time profiles
BSQ	Band Sequential, 3D-image format: first all band 1-data, then all band 2-data, etc.
BT	Brightness temperature
CORINE	CORINE land cover classification of Europe
CTIV	Centre de Traitement d'Images VEGETATION (VGT processing & archiving centre, hosted at VITO-TAP)
DB	Database
DCW	Digital Chart of the World
DEM	Digital Elevation Model
DMP	Dry Matter Productivity (kgDM/ha/day), RS vegetation-indicator derived with the Monteith-approach
EC	European Commission
ECMWF	European Centre for Medium Range Weather Forecasts
ENVI	ENvironment for Visualizing Images, commercial image processing software
EO	Earth Observation
EPSG	European Petroleum Survey Group.
ESA	European Space Agency, Rome (EU + Canada)
ET	Evapotranspiration
EUMETSAT	European Organisation for the Exploitation of Meteorological Satellites
EUMETCAST	EUMETSAT's Broadcast System for Environmental Data
FAO	Food and Agricultural Organisation of the United Nations
*f*APAR	Fraction of Absorbed Photosyntheticly Active Radiation (400-700 nm)
fCover	Fraction of vegetation cover
GDAL	Geospatial Data Abstraction Library. Includes the OGR Simple Features Library.
GIS	Geographical Information System, software for storage of geographical data, mostly in vector format
GLC2000	Global Land Cover classification based on SPOT-VGT images of the year 2000
GLIMPSE	GLobal IMage Processing SoftwarE
GPS	Global Positioning System
GUI	Graphical User Interface
HDF	Hierarchical Data Format
HDR	Header file of the ENVI-software
HR	High-Resolution imagery (pixels of 10-50 m), parcel structure visible
IMG	Image
JRC	Joint Research Centre of the European Union at Ispra, Italy
LAEA	Lambert Azimuthal Equal Area projection
LAI	Leaf Area Index
LandSAF	Spatial Applications Facility on Land Surface Analysis (MSG processing centre in Lissabon)
Lat	Latitude
Lon	Longitude
LR	Low-Resolution imagery (pixels of >100 m), parcel structure invisible, mixed pixels

ABBREV.	MEANING
LST	Land Surface Temperature
LTA	Long Term Average
MARS	Monitoring Agriculture by Remote Sensing, EU-JRC programme started in 1988
MERIS	Medium Resolution Image Spectrometer Instrument (sensor on-board of ENVISAT)
METOP	Meteorological Operational satellite programme (EUMETSAT, WMO)
MIR	Middle infrared range of the spectrum, roughly from 3 to 7 µm
MODIS	Moderate Resolution Imaging Spectroradiometer
MSG	Meteosat Second Generation (European geostationary satellite)
MVC	Maximum Value Compositing, used with NDVI/fAPAR to create S1/S10/S30-syntheses (minimising clouds)
NASA	National Administration for Space Applications (US)
NDVI	Normalised Difference Vegetation Index, RS-indicator for the amount of standing vegetation
NIR	Near infrared range of the spectrum, roughly from 780 nm to 1100nm (upper range of Silicium detectors)
NOAA	Series of near-polar satellites monitored by the US National Oceanographic and Atmospheric Administration
NPP	Net Primary Productivity ($gC/m^2/day$), RS vegetation-indicator derived with the Monteith-approach
NRT	Near-real time
OGC	Open Geospatial Consortium. An international voluntary consensus standards organization encouraging development and implementation of open standards for geospatial content and services, GIS data processing and data sharing.
PAR	Photosynthetically Active Radiation (400-700nm)
PI	Pseudo Images (compressed format)
QL(K)	Quicklook
RMSE	Root Mean Square Error
ROI	Region of Interest
RS	Remote sensing: earth observation with imaging sensors on-board of space/airborne platforms
RUM	Regional Unmixed Means (database)
S1, S10, S30	Mosaic image, synthesised over a given region, from the raw registrations of a certain day/dekad/month
SAVI	Soil Adjusted Vegetation Index
Sentinel	The Sentinels are a fleet of satellites designed specifically to deliver the wealth of data and imagery that are central to the European Commission's Copernicus programme.
SEVIRI	Spinning Enhanced Visible and Infrared Imager (MSG-sensor)
SM	Status mask (or map)
SMAC	Simplified Method for Atmospheric Correction
SPI	Standardized Precipitation Index
SPIRITS	Software for the Processing and Interpretation of Remotely sensed Image Time Series
SPOT	Systeme Probatoire d'Observation de la Terre, series of satellites monitored by CNES-France
SPx-File	File with specifications for some GLIMPSE routines (SPs for scaling, SPc for compositing, ...)
SRS	Spatial Reference Set: a coordinate-based local, regional or global system used to locate geographical entities.
SST	Sea Surface Temperature
SWETS	Algorithm for the smoothening of VI-profiles, named after the priciple author
SWIR	Shortwave Infrared (1.1-3.0µm), solar/reflective regime but beyond silicium-range. Not to confuse with MIR!
T	Temperature

ABBREV.	MEANING
TAMSAT	Tropical Applications of Meteorology using SATellite and other data
TAP	VITO's Centre of Expertise on Remote Sensing and Earth Observation
TIR	Thermal infrared range of the spectrum, roughly from 7 to 50 µm
TM	Thematic Mapper, HR-sensor on-board of the US LANDSAT-satellite series
UI	User Interface, see GUI
VCI	Vegetation Condition Index
VGT	VEGETATION, LR-sensor on-board the SPOT4/5-satellites
VGT-P	VEGETATION segment, partly pre-processed
VGT-S10	VEGETATION 10-day maximum value composite
VI	Vegetation Index
VIS	Visual range of the spectrum, roughly from 380 to 780 nm
VITO	Vlaamse Instelling voor Technologisch Onderzoek (Flemish Institute for Technological Research), Belgium
VLR	Very Low Resolution (pixels > 1km), for instance MSG-SEVIRI
VPI	Vegetation Productivity Index
WGS84	World Geodetic System 1984
WKT	Well-known text : a text markup language for representing vector geometry objects on a map, spatial reference systems of spatial objects and transformations between spatial reference systems. Format originally defined by OGC.
WMO	World Meteorological Organisation