To describe a voxel geometry, the user must:

1. Assign an organ to each voxel. Each organ is identified by a unique integer ≤ 32767. The numbering does not need to be contiguous, i.e., gaps in the numbering sequence are allowed. One of the organs must have number 0 and plays the role of the medium surrounding the voxels (usually vacuum or air). The assignment is done via a special file where the organ corresponding to each voxel is listed sequentially in Fortran list-oriented format, with the `x` coordinate running faster than `y`, and `y` running faster than `z`. In practice the file is always written by a program similar to the one reported below. The user will need to modify the values of the parameters `DX, DY DZ, NX, NY, NZ` (respectively voxel size and number of voxels for each coordinate), and possibly some other more trivial things (file names, title, reading from the original CT scan file).
   The following program takes also care of recompacting the original organ numbers by eliminating all gaps in the sequence, and writes a translation table to the screen:

   ```
   WRITE(*,'(A,2I10)')' New number, old number: ', NO, IC
   ```

   After having modified the program (assumed to be in a file `writegolem.f`), compile it:

   ```
   $FLUPRO/flutil/fff writegolem.f
   ```

   link it with the FLUKA library:

   ```
   $FLUPRO/flutil/lfluka -o writegolem writegolem.o
   ```

   execute it:

   ```
   ./writegolem
   ```

   The result will be a file `golem.vxl` (or equivalent name chosen by the user) which will be referred to by a special command line in the geometry input (see 2 below).

```
      PROGRAM WRITEGOLEM

      INCLUDE '(DBLPRC)'
      INCLUDE '(DIMPAR)'
      INCLUDE '(IOUNIT)'
*  COLUMNS: FROM LEFT TO RIGHT
*  ROWS: FROM BACK TO FRONT
*  SLICES: FROM TOP TO BOTTOM
      PARAMETER ( DX = 0.208D+00 )
      PARAMETER ( DY = 0.208D+00 )
      PARAMETER ( DZ =-0.8D+00   )
      PARAMETER ( NX = 256 )
      PARAMETER ( NY = 256 )
      PARAMETER ( NZ = 220 )
      DIMENSION GOLEM(NX,NY,NZ)
      INTEGER*2 GOLEM
      CHARACTER TITLE*80
      DIMENSION IREG(1000), KREG(1000)
      INTEGER*2 IREG, KREG
*
      CALL CMSPPR
      DO IC = 1, 1000
         KREG(IC) = 0
      END DO
      OPEN(UNIT=30,FILE='ascii_segm_golem',STATUS='OLD')
      READ(30,*) GOLEM
      NO=0
      MO=0
      DO IZ=1,NZ
         DO IY=1,NY
            DO IX=1,NX
               IF (GOLEM(IX,IY,IZ) .GT. 0) THEN
                  IC = GOLEM(IX,IY,IZ)
                  MO = MAX (MO,IC)
                  DO IR=1,NO
                     IF (IREG(IR) .EQ. IC) GO TO 1000
                  END DO
```

```
                  NO=NO+1
                  IREG(NO)=IC
                  KREG(IC)=NO
                  WRITE(*,'(A,2I10)')' New number, old number: ', NO, IC
 1000             CONTINUE
               END IF
            END DO
         END DO
      END DO
*     NO = number of different organs
*     MO = max. organ number before compacting
      WRITE(*,*) ' NO,MO',NO,MO
      OPEN(UNIT=31,FILE='golem.vxl',STATUS='UNKNOWN',FORM='UNFORMATTED')
      TITLE = 'Golem'
      WRITE(31) TITLE
      WRITE(31) NX,NY,NZ,NO,MO
      WRITE(31) DX,DY,DZ
      WRITE(31) GOLEM
      WRITE(31) (KREG(IC),IC=1,MO)
      STOP
      END
```

Starting from Fluka2011.2b, the voxel files can contain an arbitrary number of extra records of 80 characters each, which are read and interpreted as ordinary input cards. This allows to embed in the voxel files informations such as material definitions, material assignments, correction factor etc, which are often generated by automatic programs out of a CT scan. Flair contains tools for reading CT scans in Dicom format, and automatically generate a voxel file containing the material and correction factor informations according to a Hounsfield number to material/density translation algorithm which can be tuned by the user.

2. Prepare the usual FLUKA input file. The geometry must be written like a normal Combinatorial Geometry input (in any of the allowed formats, as part of the normal input stream or in a separate file), but in addition must include:
   - A VOXELS card as a first line, before the Geometry title card (8.2.2), with the following information:
     WHAT(1), WHAT(2), WHAT(3) = `x, y, z` coordinates chosen as the origin of the "voxel volume", i.e., of a region made of a single `RPP` body (8.2.4.1) which contains all the voxels
     WHAT(4) = index (or name) of the ROT–DEFIni card for an eventual roto/translation of the VOXELs
     WHAT(5), WHAT(6): not used
     SDUM = name of the voxel file (extension will be assumed to be `.vxl`)
   - The usual list of `NB` bodies, not including the `RPP` corresponding to the "voxel volume" (see VOXELS card above). This `RPP` will be generated and added automatically by the code as the `(NB+1)`th body, with one corner in the point indicated in the VOXELS card, and dimensions `NX*DX, NY*DY` and `NZ*DZ` as read from the voxel file.
   - The usual region list of `NR` regions, with the space occupied by body `NB+1` (the "voxel volume") subtracted. In other words, the `NR` regions listed must cover the whole available space, except the space corresponding to the "voxel volume". This is easily obtained by subtracting body `NB+1` in the relevant region definitions, even though this body is not explicitly input at the end of the body list. The code will automatically generate and add several regions:

Name	Number	Description
`VOXEL`	`NR+1`	this is a sort of "cage" for all the voxels. Nothing (energy etc.) should ever be deposited in it: the user shall assign `VACUUM` to it
`VOXEL001`	`NR+2`	containing all voxels belonging to organ number 0. There must be at least 2 of such voxels, but in general they should be many more. Typical material assignment to region `NR+2` is `AIR`.
`VOXEL002`	`NR+3`	corresponding to organ 1