```
// SUV cannot be calculated if any of the specified DICOM attributes are missing or empty or zero
if Corrected Image (0x0028,0x0051) contains ATTN and DECY and Decay Correction (0x0054,0x1102) is START {
        if Units (0x0054,0x1001) are BQML {
                half life = Radionuclide Half Life (0x0018,0x1075) in Radiopharmaceutical Information Sequence (0x0054,0x0016)        // seconds
                if Series Date (0x0008,0x0021) and Time (0x0008,0x0031) are not after Acquisition Date (0x0008,0x0022) and Time (0x0008,0x0032)  {
                        scan Date and Time = Series Date and Time
                }
                else {  // may be post-processed series in which Series Date and Time are date of series creation unrelated to acquisition
                        if  GE private scan Date and Time (0x0009,0x100d,"GEMS_PETD_01") present {
                                scan Date and Time = GE private scan Date and Time (0x0009,0x100d,"GEMS_PETD_01")
                        }
                        else {
                                // else may be Siemens series with altered Series Date and Time
                                // either check earliest of all images in series (for all bed positions) (wrong for case of PETsyngo 3.x multi-injection)
                                scan Date and Time = earliest Acquisition Date (0x0008,0x0022) and Time (0x0008,0x0032)  in all images of series
                                        or
                                // back compute from center (average count rate ) of time window for bed position (frame) in series (reliable in all cases)
                                // Acquisition Date (0x0008,0x0022) and Time (0x0008,0x0032) are the start of the bed position (frame)
                                // Frame Reference Time (0x0054,0x1300) is the offset (ms) from the scan Date and Time we want to the average count rate time
                                if  (Frame Reference Time (0x0054,0x1300) > 0 && Actual Frame Duration (0018,1242) > 0) {
                                        frame duration = Actual Frame Duration (0018,1242) / 1000        // DICOM is in ms; want seconds
                                        decay constant = ln(2) /  half life
```

```
                                        decay during frame = decay constant * frame duration
                                        average count rate time within frame = 1/decay constant * ln(decay during frame / (1 – exp(-decay during frame)))
                                        scan Date and Time = Acquisition Date (0x0008,0x0022) and Time (0x0008,0x0032)
                                            -   Frame Reference Time (0x0054,0x1300) /1000 + average count rate time within frame
                                    }
                            }
                    }
                    if Radiopharmaceutical Start DateTime (0018,1078) is present in Radiopharmaceutical Information Sequence (0x0054,0x0016)  {
                            start Date and Time = Radiopharmaceutical Start DateTime in Radiopharmaceutical Information Sequence (0x0054,0x0016)
                    }
                    else {
                            start Time = Radiopharmaceutical Start Time (0x0018,0x1072) in Radiopharmaceutical Information Sequence (0x0054,0x0016)
                            // start Date is not explicit … assume same as Series Date; but consider spanning midnight
                    }
                    decay Time = scan Time – start Time          // seconds
                    // Radionuclide Total Dose is NOT corrected for residual dose in syringe, which is ignored here …
                    injected Dose = Radionuclide Total Dose (0x0018,0x1074) in Radiopharmaceutical Information Sequence (0x0054,0x0016)      // Bq
                    decayed Dose = injected Dose * pow (2, -decay Time / half life)
                    weight = Patient's Weight (0x0010,0x1030)              // in kg
                    SUVbwScaleFactor = (weight * 1000 / decayed Dose)
}
else if Units (0x0054,0x1001) are CNTS {
```

```
                SUVbwScaleFactor = Philips private scale factor (0x7053,0x1000,“ Philips PET Private Group”)
                // if (0x7053,0x1000) not present, but (0x7053,0x1009) is present, then (0x7053,0x1009) * Rescale Slope
                // scales pixels to Bq/ml, and proceed as if Units are BQML
        }
        else if Units (0x0054,0x1001) are GML {
                SUVbwScaleFactor = 1.0          // assumes that GML indicates SUVbw instead of SUVlbm
        }
}
// Rescale Intercept is required to be 0 for PET, but use it just in case
// Rescale slope may vary per slice (GE), and cannot be assumed to be constant for the entire volume
SUVbw = ((stored pixel value in Pixel Data (0x7FE0,0x0010) * Rescale Slope (0x0028,0x1053) + Rescale Intercept (0x0028,0x1052) ) * SUVbwScaleFactor      // g/ml
```