

ResizeKeys Group

John D. Baker

SHA-256: ec6da2d26618e11d780fd77edcb4c4a17b4e4b12fd878358a960f236ce470f07

June 21, 2025

Contents

ResizeKeys Overview	2
ResizeKeys Interface	2
ResizeKeys Source Code	3
=: Index	29

ResizeKeys Overview

ResizeKeys is a collection of ad hoc utilities I use to *police* print size keys. For years I have been using Python scripts to access my [SmugMug](#) images via SmugMug's spotty API. The keys have been computed using many DPis: mostly 360 and 300.

NOTE: Later versions of my sizing calculations (see `sizekey2`) tend to increase print area. Run the verb `sizekeychanges` for examples. I don't change smaller online keys with matching aspect ratios.

Also, I have tried to maintain consistency between the Python programs and these J scripts but they sometimes produce different keys. The differences are usually size related. Aspect ratios largely match.

ResizeKeys Interface

<code>addprintsizes</code>	<code>[7]</code>	<i>adds additional sizes to size lists</i>
<code>minsizechanges</code>	<code>[12]</code>	<i>number of key differences beteen old and new sizes</i>
<code>moreprintsizes</code>	<code>[14]</code>	<i>add additional print sizes</i>
<code>resizekey</code>	<code>[17]</code>	<i>list mirror_db images getting new size keys with expanded size key list</i>
<code>sizekeychanges</code>	<code>[22]</code>	<i>images with size key changes when size list changes</i>
<code>sizekeycheck</code>	<code>[24]</code>	<i>compare new computed size keys with online keys</i>
<code>ununsized</code>	<code>[26]</code>	<i>resized images with new tables</i>

ResizeKeys Source Code

```
NB.*ResizeKeys s-- utils used when changing print size key lists.
NB.
NB. I have slowly increased the number of print size keys used by
NB. the Python (smugpyter) program over the years. When the size
NB. list changes some existing keys change. This group helps
NB. finds images in (mirror.db) that need inspection.
NB.
NB. Note: size keys have been calculated at a number of DPIs.
NB. Later versions of my sizing calculations tend to increase
NB. print area: run the verb (sizekeychanges) for examples. I
NB. don't change smaller online keys with matching aspect ratios.
NB.
NB. verbatim:
NB.
NB. interface word(s):
NB. -----
NB. addprintsizes - adds additional sizes to size lists
NB. minsizechanges - number of key differences beteen old and new sizes
NB. moreprintsizes - add additional print sizes
NB. resizekey - list mirror_db images getting new size keys with expanded size key list
NB. sizekeychanges - images with size key changes when size list changes
NB. sizekeycheck - compare new computed size keys with online keys
NB. unsized - resized images with new tables
NB.
NB. created: 2020mar04
```

NB. -----
 NB. 25jun15 Darktable aspect ratios added to (SMUGPYTERSIZES)
 NB. 25jun16 (sizekey2) replaces (sizkey)
 NB. 25jun20 (addprintsizes, minsizechanges, ununsized) added
 NB. 25jun21 (shrinkkeys, nszfrkeys, keysfrnsz) added

```
require 'data/sqlite'
coclass 'ResizeKeys'
```

NB. increase print precison
 (9!:11) 12
 NB.*end-header

NB. area adjustment factor
 ADJUSTAREA=: 1

NB. bump resolution to insure excellent print quality
 DPIBUMP=: 120

NB. interface words (IFACEWORDSResizeKeys) group
 IFACEWORDSResizeKeys=: <|. _1 ' addprintsizes minsizechanges moreprintsizes resizekey sizekeychanges sizekey
 >..>check ununsized'

NB. minimum print dpi
 MINDPI=: 72

NB. new expanded smugpyter print size keys

```
NEWSMUGPYTERSIZES=: <;._1 ' 1x1.414214 1x1.618034 1x2.35 1x2.39 1x3.5 1.5x2.121321 1.5x2.427051 1.5x3.525 1
>...>.5x3.585 1.5x5.25 2x2 2x2.5 2x2.65 2x2.828428 2x3 2x3.236068 2x3.5 2x4 2x4.7 2x4.78 2x5 2x7 2.125x2.75 2.2
>...>5x2.25 2.25x4 2.25x5.25 2.5x2.5 2.5x3 2.5x3.25 2.5x3.35 2.5x3.5 2.5x4 2.625x6.125 2.75x2.75 2.75x3.5 2.75x
>...>4 2.75x4.25 2.75x7 3x3 3x3.75 3x3.975 3x4 3x4.242642 3x4.5 3x4.854102 3x5 3x5.25 3x6 3x7 3x7.05 3x7.17 3x7
>...>.5 3x8.125 3x9 3x10.5 3x12 3x15 3x18 3.1875x4.125 3.25x3.25 3.375x6 3.5x3.5 3.5x5 3.5x6 3.75x3.75 3.75x4.5
>...> 3.75x4.875 3.75x5.025 3.75x5.25 3.75x6 3.75x6.75 3.75x8 3.75x8.75 4x4 4x5 4x5.3 4x5.656856 4x6 4x6.472136
>...> 4x7 4x8 4x9.4 4x9.56 4x10 4x14 4x24 4.125x5.25 4.125x6 4.125x6.375 4.125x10.5 4.25x5.5 4.5x4.5 4.5x6 4.5x
>...>7.5 4.5x8 4.5x9 4.5x10.5 4.5x12.1875 4.5x13.5 4.5x18 4.5x22.5 5x5 5x6 5x6.5 5x6.7 5x7 5x8 5x10 5x12.5 5x17
>...>.5 5x30 5.25x7.5 5.25x9 5.25x12.25 5.5x5.5 5.5x7 5.5x8 5.5x8.5 5.5x11 5.5x14 5.5x33 5.625x10.125 5.625x12
>...>6x6 6x7.5 6x7.95 6x8 6x8.485284 6x9 6x9.708204 6x10 6x10.5 6x12 6x14 6x14.1 6x14.34 6x15 6x16.25 6x18 6x21
>...> 6x24 6x30 6x36 6.375x8.25 6.75x12 7x7 7x10 7x12 7x14 7x17.5 7x21 7x24.5 7x28 7x35 7.5x9 7.5x9.75 7.5x10 7
>...>.5x10.05 7.5x10.5 7.5x12 7.5x12.5 7.5x13.5 7.5x16 7.5x17.5 8x8 8x10 8x10.6 8x11.313712 8x12 8x12.944272 8x
>...>14 8x16 8x18.8 8x19.12 8x20 8x24 8x28 8x32 8x40 8x48 8.25x10.5 8.25x12 8.25x12.75 8.25x21 8.5x11 8.5x34 9x
>...>9 9x12 9x13.5 9x15 9x16 9x18 9x21 9x22.5 9x24.375 9x27 9x31.5 9x36 9.5x38 10x10 10x12 10x12.5 10x13 10x13.
>...>4 10x14 10x15 10x16 10x20 10x25 10x30 10x35 10x40 10x50 10x60 10.5x14 10.5x15 10.5x17.5 10.5x18 10.5x24.5
>...>11x11 11x14 11x16 11x16.5 11x17 11x22 11x27.5 11x28 11x33 11x38.5 11x44 11x66 11.25x20.25 11.25x24 12x12 1
>...>2x15 12x15.9 12x16 12x16.970568 12x18 12x19.416408 12x20 12x21 12x24 12x28 12x28.2 12x28.68 12x30 12x32.5
>...>12x36 12x42 12x48 12x60 12x72 12.5x15 12.75x16.5 13.5x24 14x14 14x17.5 14x20 14x21 14x24 14x28 14x35 14x49
>...> 15x18 15x19.5 15x20 15x20.1 15x21 15x24 15x27 15x32 15x35 16x16 16x20 16x21.2 16x22.627424 16x24 16x25.88
>...>8544 16x28 16x32 16x37.6 16x38.24 16x40 16x56 16.5x21 16.5x24 16.5x25.5 16.5x42 17x22 17.5x21 18x18 18x24
>...>18x27 18x30 18x32 18x36 18x42 18x48.75 18x54 18x72 18x90 18x108 20x20 20x24 20x26 20x26.8 20x28 20x30 20x3
>...>2 20x40 21x28 21x30 21x36 22x28 22x32 22x33 22x34 22x56 22.5x40.5 22.5x48 24x32 24x36 24x40 24x65 24x72 24
>...>x96 24x120 24x144 25x25 28x40 28x48 30x30 30x54 30x64'
```

NB. keyword for images that are too small for (SMUGPYTERSIZES)

NOPIXELSKEY=: '0z0'

NB. print key not assigned

`NOPRINTKEY=: '0z9'`

NB. keyword for images that do not match (SMUGPYTERSIZES)

`NORATIOKEY=: '0z1'`

NB. root words (ROOTWORDSResizeKeys) group

`ROOTWORDSResizeKeys=: <;._1 ' IFACEWORDSResizeKeys ROOTWORDSResizeKeys VMDResizeKeys addprintsizes minsizec
>..>hanges resizekey ununsized'`

NB. round off for print surface area (square inches)

`SMUGAREAROUND=: 0.05000000000000000028`

NB. round off for SmugMug aspect ratios - precision must distinguish all ratios

`SMUGASPECTROUND=: 0.00500000000000000001`

NB. default print DPI - set to ThumbsPlus macro value of 360

`SMUGPRINTDPI=: 360`

NB. print sizes matching Jupyter/Python see: 50 list SMUGPYTERSIZES

`SMUGPYTERSIZES=: <;._1 ' 1x1.414214 1x1.618034 1x2.35 1x2.39 1x3.5 2x2 2x2.5 2x2.65 2x2.828428 2x3 2x3.2360
>..>68 2x3.5 2x4 2x4.7 2x4.78 2x5 2x7 2.125x2.75 2.25x4 2.25x5.25 2.5x2.5 2.5x3 2.5x3.25 2.5x3.35 2.5x3.5 2.5x
>..>4 2.75x3.5 2.75x4 2.75x4.25 2.75x7 3x3 3x4 3x5 3x7 3x8.125 3x9 3x12 3x15 3x18 3.5x3.5 3.5x5 3.5x6 3.75x6.7
>..>5 3.75x8 4x4 4x5 4x5.3 4x5.656856 4x6 4x6.472136 4x7 4x8 4x9.4 4x9.56 4x10 4x14 4.25x5.5 4.5x8 4.5x10.5 5x
>..>5 5x6 5x6.5 5x6.7 5x7 5x8 5x10 5x30 5.5x7 5.5x8 5.5x8.5 5.5x14 6x6 6x8 6x10 6x12 6x14 6x15 6x16.25 6x18 6x`

```
>..>21 6x24 6x30 6x36 7x10 7x12 7.5x13.5 7.5x16 8x8 8x10 8x10.6 8x11.313712 8x12 8x12.944272 8x14 8x16 8x18.8
>..>8x19.12 8x20 8x24 8x28 8x32 8x40 8.5x11 9x12 9x15 9x16 9x21 9x36 10x10 10x12 10x13 10x13.4 10x14 10x15 10x
>..>16 10x20 10x25 10x30 10x35 10x40 10x60 11x14 11x16 11x17 11x28 12x12 12x15 12x16 12x18 12x20 12x24 12x28 1
>..>2x30 12x32.5 12x36 12x42 12x48 12x60 12x72 14x20 14x24 15x18 15x27 15x32 16x16 16x20 16x21.2 16x22.627424
>..>16x24 16x25.888544 16x28 16x32 16x37.6 16x38.24 16x40 16x56 17x22 18x24 18x32 18x42 20x20 20x24 20x26 20x2
>..>6.8 20x28 20x30 20x32 20x40 22x28 22x32 22x34 22x56 24x32 24x36 24x40 24x65 24x72 24x96 24x120 24x144 28x4
>..>0 28x48 30x30 30x54 30x64'
```

NB. version, make count, and date

```
VMDResizeKeys=: '0.6.0';23;'21 Jun 2025 21:55:57'
```

```
addprintsizes=: 3 : 0
```

*NB.*addprintsizes v-- adds additional sizes to size lists.*

NB.

NB. monad: addprintsizes uuIgnore

```
'Are you sure? Edit if so!' return.
```

```
os=. SMUGPYTERSIZES
```

```
ns=. NEWSMUGPYTERSIZES
```

```
SMUGPYTERSIZES=: ns
```

```
#NEWSMUGPYTERSIZES=: moreprintsizes ns
```

```
)
```

```
aspgroups=: 3 : 0
```

*NB.*aspgroups v-- split print keys into sorted aspect groups.*

NB.

NB. dyad: bt =. aspgroups blclSizeKeys

NB.

NB. aspgroups NEWSMUGPYTERSIZES

```
g=. /: r=. (%/"1 ,. ]) ".@('x '&charsub)&> y
b=. ~: 0 {"1 r=. g{r
r=. b <:.1 r [ s=. b <:.1 g{y
g=. /:&.> 1&{"1 &.> r
(g {&.> r) ,. g {&.> s
)
```

NB. signal with optional message

```
assert=: 0 0" _ $ 13!:8^:((0: e. ]) `(12" _))
```

```
charsub=: 4 : 0
```

*NB.*charsub v-- single character pair replacements.*

NB.

NB. dyad: clPairs charsub cu

NB.

NB. '-_ \$ ' charsub '\$123 -456 -789'

```
'f t'=. ((#x)$0 1)<@,&a./ .x
t {~ f i. y
)
```



```
dpiarearatio=: 3 : 0
```

```
NB.*dpiarearatio v-- print area and aspect ratio for image dimensions at given DPI.
```

```
NB.
```

```
NB. monad: fa =. dpiarearatio ilWidthHeight
```

```
NB.
```

```
NB.    dpiarearatio 2000 3000
```

```
NB.    dpiarearatio |: 10 # ,: 4888,3256
```

```
NB.
```

```
NB. dyad: fa =. faAspectDP1 dpiarearatio ilWidthHeight
```

```
NB.
```

```
NB.    0.0005 0.05 500 dpiarearatio 2000 3000
```

```
(SMUGASPECTROUND,SMUGAREAROUND,SMUGPRINTDPI) dpiarearatio y
```

```
:
```

```
'aspect area dpi'=. x
```

```
(area round (*/y) % *: dpi) ,: aspect round (<./y) % >./y
```

```
)
```

```
fsd=: 4 : 0
```

```
NB.*fsd v-- fetch sqlite dictionary array.
```

```
NB.
```

```
NB. dyad: bt =. clSql fsd clDb
```

```
NB.
```

```
NB.    trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
```

```
NB.    sql=. 'select ImageKey, OriginalWidth, OriginalHeight, OnlineImageFile, Keywords from OnlineImage'
```

```
NB.    sql fsd trg
```

```
NB. require 'data/sqlite' !(*)=. sqlclose__db sqldict__db sqlopen__psqlite_
d [ sqlclose__db '' [ d=. sqldict__db x [ db=. sqlopen__psqlite_ y
)
```

```
itrgroups=: 3 : 0
```

```
NB.*itrgroups v-- add smaller and larger sizes to aspect groups.
```

```
NB.
```

```
NB. This verb attempts to fill in "holes" in print size keys
```

```
NB. without introducing too many decimal keys.
```

```
NB.
```

```
NB. monad: blft =. itrgroups blclSizeKeys
```

```
NB.
```

```
NB. itrgroups 0 {"1 aspgroups SMUGPYTERSIZES
```

```
NB.
```

```
NB. dyad: blft =. ilLowHigh itrgroups blclSizeKeys
```

```
NB.
```

```
NB. 5 20 itrgroups 0 {"1 aspgroups SMUGPYTERSIZES
```

```
2 30 itrgroups y
```

```
:
```

```
'lw hg'=. x
```

```
r=. }.@{0&f} &.> y
```

```
s=. {{ (-:^(i. 5) 1) *"0 1 y }} &.> r
```

```
t=. {{ (+:^(i. 5) 1) *"0 1 y }} &.> r
```

```
s=. s ,&.> t
```

```
s=. ~.@{/:~}&.> s ,&.> }. "1&.> y
```

```

NB. remove out of bounds sizes
s=. (lw&<:@(0&{"1) &.> s) #&.> s
(hg&>:@(0&{"1) &.> s) #&.> s
)

itrpsizes=: 3 : 0

NB.*itrpsizes v-- insert additional same aspect ratios in sizes list.
NB.
NB. monad: sl =. itrpsizes blsl
NB.
NB. itrpsizes s: ' 3x18 5x30 6x36 10x60 12x72 24x144'
NB.
NB. NB. verb is "powerable"
NB. itrpsizes~: 4 s: ' 3x18 5x30 6x36 10x60 12x72 24x144'

NB. sizes
sz=. _1&".@('x '&charsub)&> 5 s: y

NB. intermediate sizes
iz=. |: -:@+/&> 2 <\"1 |: sz

NB. expand sizes
sz=. ( }: ;(#sz)#<1 0) #~:_1 sz

NB. merge intermediates
sz=. iz (I. 0 = 0{"1 sz)} sz

```

```

NB. reform keys
s: (' x'&charsub)@":&.> <"1 sz
)

keysfrnsz=: 3 : 0

NB.*keysfrnsz v-- keys from numeric sizes.
NB.
NB. monad: blclKeys =. keysfrnsz ftSizes
NB.
NB. keysfrnsz nszfrkeys SMUGPYTERSIZES
NB. keysfrnsz 5 2$0

k=. (' x'&charsub)@":&.> <"1 y

NB. replace any 0x0 keys
(<NOPIXELSKEY) (I. k -:&> <'0x0'})} k
)

minsizechanges=: 3 : 0

NB.*minsizechanges v-- number of key differences beteen old and new sizes.
NB.
NB. monad: ft =. minsizechanges iaCnt
NB.
NB. minsizechanges 50

```

```

NB.
NB. dyad: ft -. iaDpi minsizechanges iaCnt
NB.
NB.   _120 minsizechanges 50
NB.   720 minsizechanges 50

DPIBUMP minsizechanges y
:

NB. sqlite databse
db=. '~MIRROR/mirror.db'
dpi=. x+SMUGPRINTDPI

NB. smaller dpis need smaller areas
allkeys=. 0;dpi;SMUGASPECTROUND;SMUGAREAROUND;<NEWSMUGPYTERSIZES

row=. 0 8$0
sf=. {{ }. (<. -:y) }. i: 1 j. y }} y
for_adjust. sf do.
  ADJUSTAREA=: adjust
  st=. allkeys sizekeycheck db
  ka=. ;1{onkeyarea 0 2 {"1 st
  nr=. adjust,dpi,#st
  row=. row , nr,ka
end.

hd=. ;: 'area_factor dpi diff_cnt equal_area old_greater new_greater old_zero new_zero'

```

```

hd ,: ,.&.> <"1 |: row
)

moreprintsizes=: 3 : 0

NB. *moreprinsizes v-- add additional print sizes
NB.
NB. monad: blcl =. moreprinsizes blclSizeKeys
NB.
NB. 80 list moreprintsizes SMUGPYTERSIZES
NB. 80 list moreprintsizes NEWSMUGPYTERSIZES
NB.
NB. dyad: blcl =. ilMinareaMaxwid moreprinsizes blclSizeKeys
NB.
NB. 80 list 3 40 moreprintsizes NEWSMUGPYTERSIZES

3 30 moreprintsizes y
:
NB. expand sizes
ns=. ~. y , (' x'&charsub)@":&.> <"1 ;"1 ] x itrgroups 0 {"1 aspgroups y

NB. interpolate expanded sizes
sortprintsizes ;(5&s:)@itrpsizes&.> 2 {"1 printsizestable ns
)

nszfrkeys=: 3 : 0

```

*NB.*nszfrkeys v-- numeric sizes from keys.*

NB.

NB. monad: ft =. nszfrkeys blclKeys

NB.

NB. nszfrkeys SMUGPYTERSIZES

```
(_1&".)@('x z '&charsub)&> y
)
```

onkeyarea=: 3 : 0

*NB.*onkeyarea v-- compare old new key areas counts.*

NB.

NB. monad: ft =. onkeyarea blsaKeys

NB.

NB. allkeys=. 0;(DPIBUMP+SMUGPRINTDPI);SMUGASPECTROUND;SMUGAREAROUND;<NEWSMUGPYTERSIZES

NB.

NB. st=: allkeys sizekeycheck '~MIRROR/mirror.db'

NB. onkeyarea 0 2 {"1 st

NB. smoutput y=. (15?#y) { y NB. random check rows

```
onk=. */@(_1&".)@('x z '&charsub)&> 5&s:&> y
```

```
ka=. (+/ =/"1 onk),(+/ </"1 onk),(+/ >/"1 onk), +/0=onk
```

```
(<;_1 ' equal_area old_less old_greater old_unsized new_unsized'),: <"0 ka
)
```

printsizestable=: 3 : 0

```

NB.*printsizestable v-- computes print sizes table.
NB.
NB. monad:  bt =. printsizestable blclPrintSizes
NB.
NB.  printsizestable SMUGPYTERSIZES
NB.  printsizestable <:_1 ' ',reb '10x10 20x30  3x2 2x3 12x8  8x12 10x10  '
NB.  printsizestable SMUGPYTERSIZES
NB.  printsizestable NEWSMUGPYTERSIZES
NB.
NB. dyad:  bt =. faPrecision printsizestable blclPrintSizes
NB.
NB.  0.00005 printsizestable SMUGPYTERSIZES
NB.  0.00005 printsizestable PANORAMICINCHSIZES
NB.  SMUGASPECTROUND printsizestable NEWSMUGPYTERSIZES
NB.
NB.  NB. print sizes from many vendors
NB.  0.00005 printsizestable ALLPRINTSIZES
NB.
NB.  NB. coarse rounding can lead to nonunique aspect ratios
NB.  0.05 printsizestable ALLPRINTSIZES

SMUGASPECTROUND printsizestable y
:
sizes=. sortprintsizes y
ratios=. ".&> 'x% '&charsub&.> sizes
areas=.  ".&> 'x* '&charsub&.> sizes

```



```

NB. sizes with same ratio, eg: 4x5, 8x10, 4x6, 8x12
aspect=. ((~.ratios) i. ratios) </. i. #ratios

'rounded aspect ratios not distinct' assert ~:nubratios=. x round ~.ratios

NB. columns: ratio, areas, print sizes
tab=. (<"0 nubratios) ,. (aspect {&.> <areas) ,. s:&.> aspect {&.> <sizes

NB. return sorted by aspect ratio
(/: ;0 {"1 tab){tab
)

ratiofrkey=: 3 : 0

NB.*ratiofrkey v-- size key from image dimensions.
NB.
NB. monad: sl =. ratiofrkey blsaKeys
NB.
NB. ratiofrkey 0 {"1 sizekeycheck '~MIRROR/mirror.db'

%/"1 ] _1&."1 'x z '&charsub&> 5 s: ;y
)

resizekey=: 3 : 0

NB.*resizekey v-- list mirror_db images getting new size keys
NB. with expanded size key list.

```

```

NB.
NB. monad:  bt =. resizekey clMirrorDb
NB.
NB.  resizekey '~MIRROR/mirror.db'
NB.
NB. dyad:   bt =. blclSizeKeys resizekey clMirrorDb
NB.
NB.  SMUGPYTERSIZES resizekey '~MIRROR/mirror.db'
NB.  (moreprintsizes NEWSMUGPYTERSIZES) resizekey '~MIRROR/mirror.db'

NEWSMUGPYTERSIZES resizekey y
:
NB. fetch images without size keys
sql=. 'select ImageKey, OriginalWidth, OriginalHeight, OnlineImageFile, Keywords from OnlineImage '
sql=. sql, ' where Keywords like "%0z0%" or Keywords like "%0z1%" order by UploadDate desc'

NB. !(*)=. ImageKey OnlineImageFile OriginalWidth OriginalHeight Keywords
(0 {"1 d)=. 1 {"1 d=. sql fsd y

NB. recompute size keys - bump dpi units above 360 standard
parms=. (DPIBUMP + SMUGPRINTDPI);SMUGASPECTROUND;SMUGAREAROUND;<x
r=. (ImageKey ,. OnlineImageFile ,. Keywords) ,. ~ <"0 parms sizekey2 idims=. OriginalWidth,:OriginalHeight

NB. retain images with new size keys
((<"1 |:idims) ,. r) #~ -. (;0 {"1 r) e. s: NOPIXELSKEY;NORATIOKEY
)

NB. round (y) to nearest (x) (e.g. 1000 round 12345)

```

```
round=: [ * [: (<.) 0.5 + %~
```

```
shrinkkeys=: 4 : 0
```

```
NB.*shrinkkeys v-- numeric sizes from keys.
```

```
NB.
```

```
NB. dyad: blclKeys =. fa shrinkkeys blclKeys
```

```
NB.
```

```
NB. 0.5 shrinkkeys SMUGPYTERSIZES
```

```
keysfrnsz x * nszfrkeys y
```

```
)
```

```
sizekey2=: 3 : 0
```

```
NB.*sizekey2 v-- size key from image dimensions.
```

```
NB.
```

```
NB. monad: sl =. sizekey2 ilWidthHeight / itWidthHeight
```

```
NB.
```

```
NB. sizekey2 1544 2501
```

```
NB. sizekey2 5 #"1 ,. 1544 2501
```

```
NB. sizekey2 2000 2000 ,: 3000 3000
```

```
NB.
```

```
NB. db=. '~MIRROR/mirror.db'
```

```
NB. d=. 'select OriginalHeight, OriginalWidth, OnlineImageFile, Keywords from OnlineImage' fsd db
```

```
NB. (0 {"1 d)=. 1 {"1 d
```

```
NB. st=. sizekey2 OriginalWidth,:OriginalHeight
```

```

NB.
NB. dyad:  sl =. (iaDpi ; faPrecision ;< blclPrintSizes) sizekey2 ilWidthHeight / itWidthHeight
NB.
NB.      (720;SMUGASPECTROUND;0.05;<SMUGPYTERSIZES) sizekey2 4000 6000
NB.
NB.      NB. panoramic sizes from sqlite database
NB.      trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.      db=. sqlopen_psqli_ trg
NB.      sqlclose__db '' [ d=. sldict__db PanoramicImageSizes_sql
NB.      (0 {"1 d)=: 1 {"1 d
NB.      OnlineImageFile ,.~ <"0 (200;0.0005;0.05;<PANORAMICINCHSIZES) sizekey2 OriginalWidth,:OriginalHeight

((SMUGPRINTDPI+DPIBUMP);SMUGASPECTROUND;SMUGAREAROUND;<NEWSMUGPYTERSIZES) sizekey2 y
:
'invalid image dimensions' assert (2 = #y) , 0 < ,y

'dpi prec tolr psizes'=. x

'area ratio'=. (prec,tolr,dpi) dpiarearatio y

NB. area controls the size picked for matching
NB. aspect ratios - (ADJUSTAREA) can be used to
NB. to minimize the differences between old sizes
NB. and new sizes - see (minsizechanges)
area=. ADJUSTAREA * area

NB. assume no print keys

```

```
keys=. (#ratio) # s: <NOPRINTKEY
```

NB. aspect ratios from print size table

```
pst=. prec printsizestable psize  
ast=. ;0{"1 pst
```

NB. expand size table by matching aspect ratios

```
trw=. 0 ; (,100000) ; s:<NORATIOKEY  
pst=. (<"0 area),.(<"1 |: y),.(<"0 ratio),.(ast i. ratio){pst,trw
```

NB. not enough area

```
barea=. area < <./&> 4 {"1 pst  
keys=. (s: <NOPIXELSKEY) (I. barea)} keys
```

NB. ratios to far apart - non standard sizes

```
btolr=. ~:/{"1 tolround ;"1 ] 2 3 {"1 pst  
keys=. (s: <NORATIOKEY) (I. btolr)} keys
```

NB. first area greater than needed

```
barea=. I. -.barea  
isize=. (barea { 4 {"1 pst) >"0 1&.> barea { 0 {"1 pst
```

NB. some areas exactly match the last size - not enough pixels

```
ilim=. I. 0 = (isize i.&> 1) < #&> barea { 5 {"1 pst  
keys=. (s: <NOPIXELSKEY) (ilim{barea)} keys  
barea=. barea -. ilim{barea
```

NB. index sizes in bounds

```
isize=. (barea { 4 {"1 pst) >"0 1&.> barea { 0 {"1 pst
ckeys=. (isize i.&> 1) {&> barea {5 {"1 pst
```

NB. return computed keys

```
ckey (barea)} keys
)
```

```
sizekeychanges=: 3 : 0
```

*NB.*sizekeychanges v-- images with size key changes when size
NB. list changes.*

NB.

*NB. NOTE: this verb is similar to (resizekey) but looks at all
NB. images not just the ones with (NOPIXELSKEY, NORATIOKEY) size
NB. keys.*

NB.

```
NB. monad: bt =. sizekeychanges clDb
```

NB.

```
NB. sizekeychanges '~MIRROR/mirror.db'
```

NB.

```
NB. dyad: bt =. blbt sizekeychanges clDb
```

NB.

```
NB. psizes=. ((<moreprintsizes NEWSMUGPYTERSIZES),<NEWSMUGPYTERSIZES),DPIBUMP;0
```

```
NB. psizes=. ((<NEWSMUGPYTERSIZES),<SMUGPYTERSIZES),DPIBUMP;0
```

```
NB. psizes sizekeychanges '~MIRROR/mirror.db'
```

```
(((<NEWSMUGPYTERSIZES),<SMUGPYTERSIZES),DPIBUMP;0) sizekeychanges y
```

```
:
NB. the new list must contain sizes the old one doesn't
'new old dpibump unsized'=. x
'no new size keys' assert 0 < # new -. old

NB. !(*)=. OriginalHeight OriginalWidth OnlineImageFile Keywords
d=. 'select OriginalHeight, OriginalWidth, OnlineImageFile, Keywords from OnlineImage order by UploadDate
>..>desc' fsd y
(0 {"1 d)=. 1 {"1 d

NB. effective dpi
'DPI to small' assert MINDPI <: effdpi=. SMUGPRINTDPI+dpibump

NB. new computed, old computed, current online size keys
ns=. (effdpi;SMUGASPECTROUND;SMUGAREAROUND;<new) sizekey2 OriginalWidth,:OriginalHeight
os=. (SMUGPRINTDPI;SMUGASPECTROUND;SMUGAREAROUND;<old) sizekey2 OriginalWidth,:OriginalHeight
cs=. s: ;1&{&.> <;._1&.> (';'&,)@(-.&' ')&.> Keywords

if. -.unsized do. b=. 1 #~ #cs
else.
  NB. retain rows where the online key is "unsized"
  b=. cs e. s: NOPIXELSKEY;NORATIOKEY;NOPRINTKEY
end.

(<;._1 ' ;EDPI;Old_Table;New_Table;Current_Online;File;Keyword'),(<effdpi) ,"1 b#(5 s: os,.ns,.cs),.OnlineI
>..>mageFile,.Keywords
)
```

```
sizekeycheck=: 3 : 0
```

```
NB.*sizekeycheck v-- compare new computed size keys with online keys.
```

```
NB.
```

```
NB. monad: bt =. sizekeycheck clDb
```

```
NB.
```

```
NB. sizekeycheck '~MIRROR/mirror.db'
```

```
NB.
```

```
NB. dyad: bt =. bl sizekeycheck clDb
```

```
NB.
```

```
NB. (1;300;0.0005;0.0005;<NEWSMUGPYTERSIZES) sizekeycheck '~MIRROR/mirror.db'
```

```
NB.
```

```
NB. NB. show all computed keys that differ from online
```

```
NB. allkeys=. 0;(DPIBUMP+SMUGPRINTDPI);SMUGASPECTROUND;SMUGAREAROUND;<NEWSMUGPYTERSIZES
```

```
NB. allkeys sizekeycheck '~MIRROR/mirror.db'
```

```
(1;(DPIBUMP+SMUGPRINTDPI);SMUGASPECTROUND;SMUGAREAROUND;<NEWSMUGPYTERSIZES) sizekeycheck y
:
```

```
NB. !(*)=. OriginalHeight OriginalWidth OnlineImageFile Keywords
```

```
d=. 'select OriginalHeight, OriginalWidth, OnlineImageFile, Keywords from OnlineImage' fsd y
```

```
(0 {"1 d)=. 1 {"1 d
```

```
NB. new size keys
```

```
ns=. (}.x) sizekey2 idims=. OriginalWidth,:OriginalHeight
```

```
NB. new keys that differ from online key
```



```
st=. (<"0 ns) ,. OnlineImageFile,.Keywords
os=. s: ; 1&{&.> <;._1&.> (';'&,)@(-.&' ')&.> {"1 st
st=. ((<"0 os),.(<"1 |: idims),.st) #~ os ~: ns
```

*NB. NOTE: size keys have been historically computed at many DPIs.
NB. I didn't standardize at 360 DPI until many images already
NB. had assigned size keys. The new (sizekey2) calculation tends to
NB. increase print sizes, 5x7 will go to 10x14. The next step removes
NB. all images that have size keys that have the same aspect ratio.
NB. Resetting thousands of keys is tedious with the SmugMug api and
NB. there are good reasons to stay at 5x7 instead of 10x14. I'm
NB. mainly interested in filling in missing keys and aspect ratio changes.*

```
if. 1 = ;0{x do.
  NB. compute old and new ratios from keys
  or=. ratiofrkey 0 {"1 st
  nr=. ratiofrkey 2 {"1 st
  st=. st #~ bm=. or ~: nr

  NB. if the old ratio is "better" than the new keep it
  sr=. %/"1 /:~"1 > 1 {"1 st
  st=. st #~ (|sr - bm#nr) < |sr - bm#or
end.
)
```

```
sortprintsizes=: 3 : 0
```

*NB.*sortprintsizes v-- sort print sizes as ascending unique Short x Long,*

```

NB.
NB. monad:  sortprintsizes clPrintSizes
NB.
NB.  sortprintsizes SMUGPRINTSIZES
NB.  sortprintsizes <|.1 ' ',reb '10x10 20x30 3x2 2x3 12x8 8x12 10x10 '

(~. ' x'&charsub&.> ":%.> /:~ (/:~)&.> ":%.> 'x ' &charsub&.> y) -.&.> ' '
)

ununsized=: 3 : 0

NB.*ununsized v-- resized images with new tables.
NB.
NB. monad:  bt =. ununsized iaDpibump
NB.
NB.  NB. lowering effective dpi can size the unsized
NB.  ununsized 0      NB. at 360 dpi
NB.  ununsized _60   NB. at 300 dpi
NB.  ununsized _288  NB. at 72 dpi
NB.
NB. dyad:  bt =. pa ununsized iaDpibump
NB.
NB.  NB. shrink any sized keys by dpi bump ratio
NB.  1 ununsized _288

0 ununsized y
:
psizes=. ((<NEWSMUGPYTERSIZES),<SMUGPYTERSIZES),y;1

```

```

st=. psizes sizekeychanges '~MIRROR/mirror.db'
st=. st #~ +./"1 ] 0 = ( 1 2 {"1 st) e. NORATIOKEY;NOPIXELSKEY;NOPRINTKEY
if. 0=x do. st else.
  NB. at lower DPIs shrink keys to maintain print quality
  br=. 0.05 round SMUGPRINTDPI %~ SMUGPRINTDPI + y
  (|: br&shrinkkeys"1 |: }. 1 2 {"1 st) (<{.i.#st);1 2)} st
end.
)

```

NB.POST_ResizeKeys post processor.

```

smoutput IFACE_ResizeKeys=: (0 : 0)
NB. (ResizeKeys) interface word(s): 20250621j215557
NB. -----
NB. addprintsizes      NB. adds additional sizes to size lists
NB. minsizechanges    NB. number of key differences between old and new sizes
NB. moreprintsizes    NB. add additional print sizes
NB. resizekey         NB. list mirror_db images getting new size keys with expanded size key list
NB. sizekeychanges    NB. images with size key changes when size list changes
NB. sizekeycheck      NB. compare new computed size keys with online keys
NB. ununsized         NB. resized images with new tables

NB. typical calls - assumes configured MIRROR folder
resizekey '~MIRROR/mirror.db'
sizekeycheck '~MIRROR/mirror.db'
10 {. sizekeychanges '~MIRROR/mirror.db'

NB. images that might size at 360, 300 and 240 dpi

```

```
ununsized 0
ununsized _60
ununsized _120

NB. shrink smallest key to maintain 360 dpi quality
1 ununsized _288

)

cocurrent 'base'
coinsert 'ResizeKeys'
```

Index

addprintsizes, 7
ADJUSTAREA, 4, 13
aspgroups, 7
assert, 8

charsub, 8

dpiarearatio, 9
DPIBUMP, 4

fsd, 9

IFACE_ResizeKeys, 27
IFACEWORDSResizeKeys, 4
itrgroups, 10
itrpsizes, 11

keysfrnsz, 12

MINDPI, 4
minsizechanges, 12
moreprintsizes, 14

NEWSMUGPYTERSIZES, 5, 7
NOPIXELSKEY, 5
NOPRINTKEY, 6
NORATIOKEY, 6
nszfrkeys, 14

onkeyarea, 15

printsizestable, 15

ratiofrkey, 17
resizekey, 17
ROOTWORDSResizeKeys, 6

round, 19

shrinkkeys, 19
sizekey2, 19
sizekeychanges, 22
sizekeycheck, 24
SMUGAREAROUND, 6
SMUGASPECTROUND, 6
SMUGPRINTDPI, 6
SMUGPYTERSIZES, 6, 7
sortprintsizes, 25

ununsized, 26

VMDResizeKeys, 7