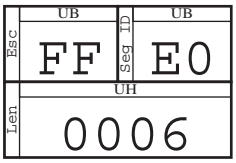
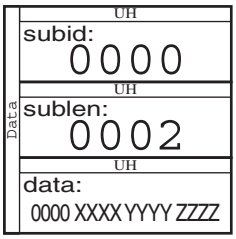
1.0 管理情報セグメント



00



【目的・機能】

このセグメントは、TAD の先頭に 1 つだけ必ず存在し、TAD のバージョン情報を示す。

【用法】

以下のパラメータにより、項目を指定する。項目が複数ある場合は、一つの管理情報セグメントの中で項目を順に並べる。

subid は、この項目の項目 ID を指定する。

sublen は、この項目のデータ本体のバイト数を指定する。必ず偶数となる。

data はデータ本体である。

現在のバージョンでは、以下の項目が定義されている。

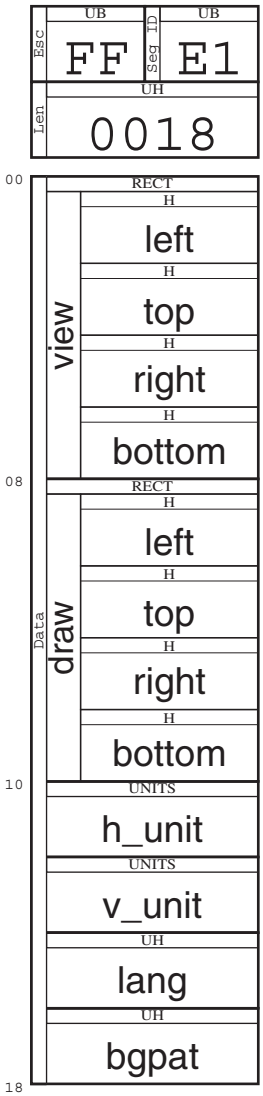
subid = 0、sublen = 2、data = 0x0XYZ の形式により、この TAD の TAD バージョンを示す。例えば data = 0x0120 で、バージョン= 1.20 を示す。現在の最新の TAD バージョンは、1.40 である。

【制限・禁止事項】

通常、このセグメントを解釈する必要はないが、必ず書き出す必要がある。(TAD のバージョンが特別な意味を持つ場合がある。行頭移動指定付箋〔2.1.5〕参照)

未定義の項目 ID を使用してはならない。

1.1 文章開始セグメント



【目的・機能】

このセグメントは、文章終了セグメント〔1.2〕と対になり、文章開始セグメントと文章終了セグメントで囲まれた範囲が文章であることを意味する。特に、管理情報セグメント〔1.0〕の直後に文章開始セグメントがある場合は、全体が文章としてレイアウトされることを意味する。

【用法】

以下のパラメータにより、文章に必要な定義をおこなう。

view は、図形中に文章を配置する領域を指定する。図形中に文章開始セグメント～文章終了セグメントを置く場合にのみ有効で、それ以外の場合は無視される。

draw は、view と対応して、文章を描画するときに使用する領域を指定する。図形中に文章開始セグメント～文章終了セグメントを置く場合にのみ有効である。view と draw の指定により、文章を拡大／縮小して図形中に置くことができる。

h_unit は水平方向の解像度を、v_unit は垂直方向の解像度を、それぞれ指定する。文章中で使用される長さ（用紙の高さや幅など）などは、この解像度で解釈される。

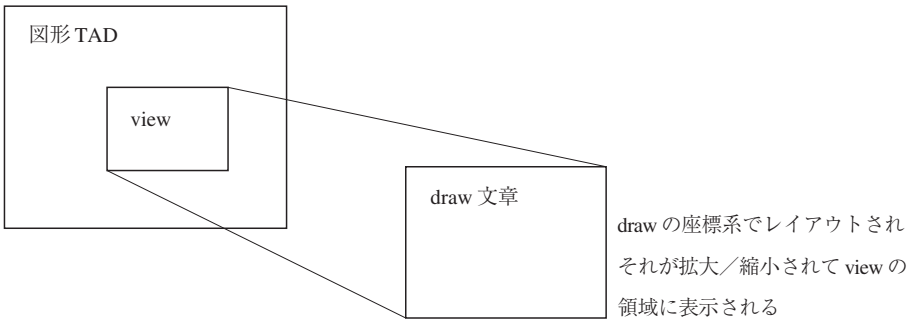
lang は、スクリプトの初期値を指定する（言語指定／多国語処理〔0.2.2〕を参照）。通常は 0x21（システムスクリプト）を指定する。

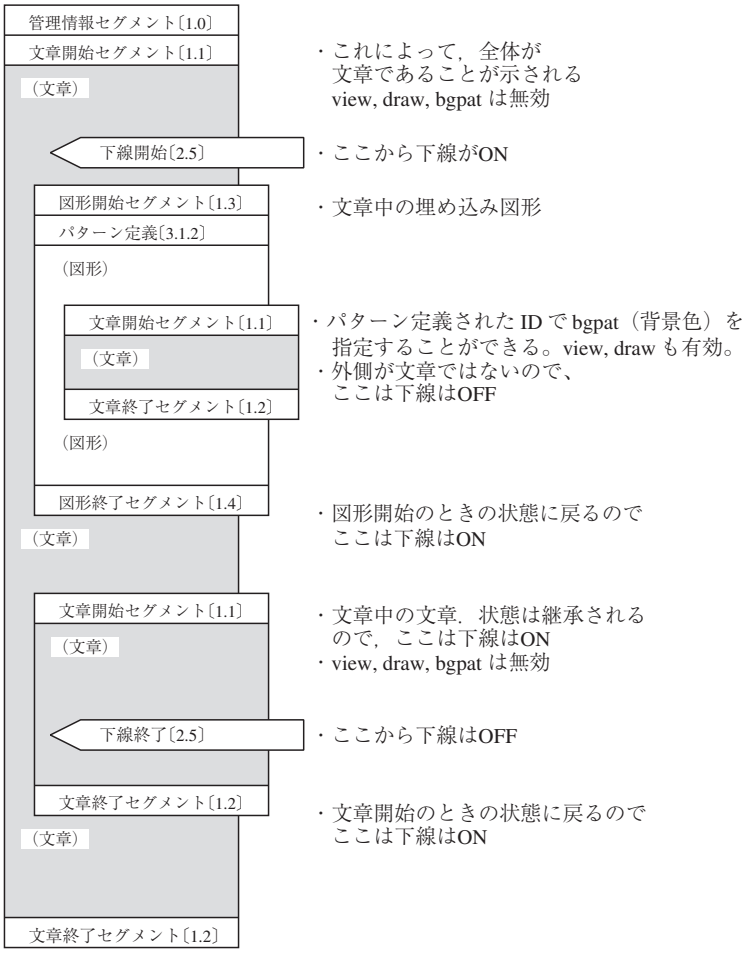
bgpat は、文章の背景となるパターン ID を指定する。図形中に文章開始セグメント～文章終了セグメントを置く場合にのみ有効で、それ以外の場合は無視される。0 の場合は透明となる。

【制限・禁止事項】

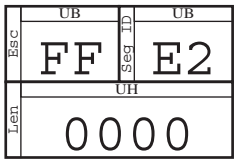
文章開始セグメントと文章終了セグメントに囲まれた範囲の内部に、さらに文章開始セグメント～文章終了セグメントがあってもよいし、図形開始セグメント～図形終了セグメントがあってもよい（TAD の構造〔0.1〕を参照）。

文章中に文章開始セグメントが現れた場合、その時点の現在スクリプトや文字修飾などが保存（push）され、文章終了セグメントが現れたときに復帰（pop）される。そうでない場合は、文章開始セグメントの出現によってパラメータがリセットされる。


図形 TAD
view
draw 文章
draw の座標系でレイアウトされ
それが拡大／縮小されて view の
領域に表示される


管理情報セグメント〔1.0〕
文章開始セグメント〔1.1〕
（文章）
・これによって，全体が
文章であることが示される
view, draw, bgpat は無効
下線開始〔2.5〕
・ここから下線がON
図形開始セグメント〔1.3〕
パターン定義〔3.1.2〕
・文章中の埋め込み図形
（図形）
文章開始セグメント〔1.1〕
（文章）
文章終了セグメント〔1.2〕
・パターン定義された ID で bgpat（背景色）を
指定することができる。view, draw も有効。
・外側が文章ではないので、
ここは下線はOFF
（図形）
図形終了セグメント〔1.4〕
・図形開始のときの状態に戻るので
ここは下線はON
（文章）
文章開始セグメント〔1.1〕
（文章）
・文章中の文章．状態は継承される
ので，ここは下線はON
・view, draw, bgpat は無効
下線終了〔2.5〕
・ここから下線はOFF
文章終了セグメント〔1.2〕
・文章開始のときの状態に戻るので
ここは下線はON
（文章）
文章終了セグメント〔1.2〕

1.2 文章終了セグメント



【目的・機能】

このセグメントは、文章開始セグメント〔1.1〕と対になり、文章開始セグメントと文章終了セグメントで囲まれた範囲が文章であることを意味する。文章開始セグメントを参照。

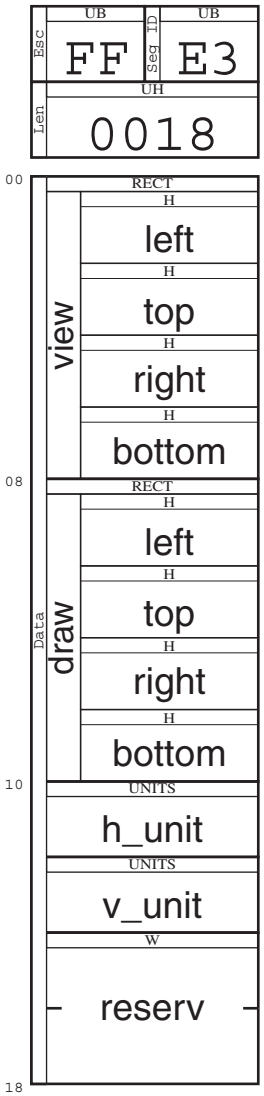
【用法】

len = 0 なので、パラメータはない。

【制限・禁止事項】

文章開始セグメント〔1.1〕を参照。

1.3 図形開始セグメント



【目的・機能】

このセグメントは、図形終了セグメント〔1.4〕と対になり、図形開始セグメントと図形終了セグメントで囲まれた範囲が図形であることを意味する。特に、管理情報セグメント〔1.0〕の直後に図形開始セグメントがある場合は、全体が図形としてレイアウトされることを意味する。

【用法】

以下のパラメータにより、図形に必要な定義をおこなう。

view は、図形を配置する領域を指定する。図形中に図形開始セグメント～図形終了セグメントを置く場合には位置・大きさともに有効、文章中に図形開始セグメント～図形終了セグメントを置く場合には大きさのみが有効で、それ以外の場合は無視される。また、文章中に図形開始セグメント～図形終了セグメントを置く場合に、幅または高さを 0 にすると、そのときの文字サイズに合わせて相似拡大／縮小する機能（文字サイズ比例）がある。横書きの場合は図形の高さが文字サイズと一致するように拡大／縮小され、縦書きの場合は図形の幅が文字サイズと一致するように拡大／縮小される。

draw は、view と対応して、図形を描画するときに使用する領域を指定する。文章中や図形中に図形開始セグメント～図形終了セグメントを置く場合にのみ有効である。view と draw の指定により、図形を拡大／縮小することができる。

h_unit は水平方向の解像度を、v_unit は垂直方向の解像度を、それぞれ指定する。図形中で使用される長さ（用紙の高さや幅など）などは、この解像度で解釈される。

reserv には 0 を指定する。

reserv は TAD ver. 1.20 においては「ratio」（W 型）と規定され、「図形データの場合のみ定義され、描画領域の大きさと、実際の物理的な大きさとの倍率または縮小率を示し、（実際の物理的大きさ）×（倍率）=（描画領域の大きさ）となる。」と定義されていた。

ratio ＞ 0：実際の大きさ×（1／ratio）=図形の大きさ（縮小）

ratio ＜ 0：実際の大きさ×（−ratio）=図形の大きさ（拡大）

ratio = 0：未定義（×1と見なす）

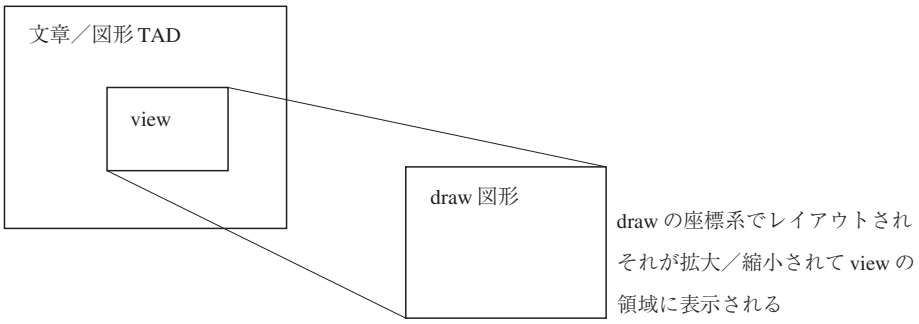
（縦横は等倍率）

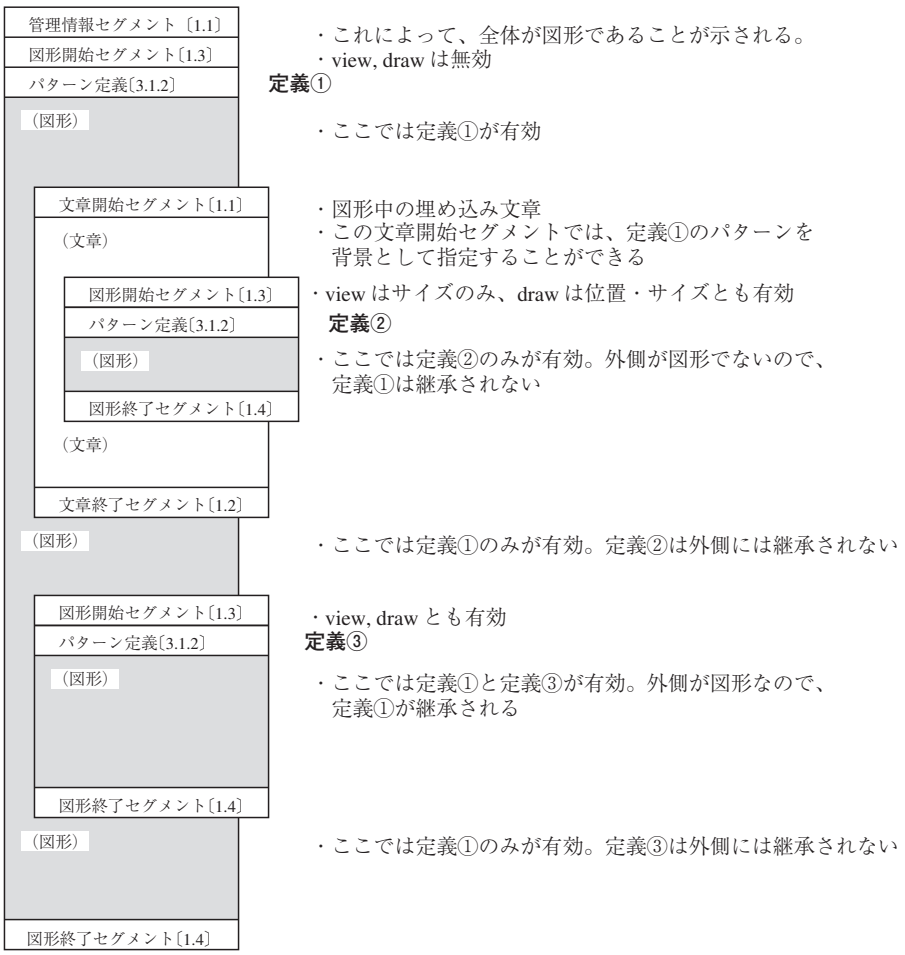
これはあくまでデータ表現上の大きさに対し、物理的な実寸を表すためのものであり、表示上の変化を伴わない。ver 1.21 以降は「reserv」である。

【制限・禁止事項】

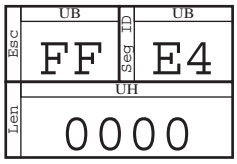
図形開始セグメントと図形終了セグメントに囲まれた範囲の内部に、さらに図形開始セグメント～文章終了セグメントがあってもよいし、図形開始セグメント～図形終了セグメントがあってもよい（TAD の構造〔0.1〕を参照）。

図形中に図形開始セグメントが現れた場合、その時点のデータ定義などが保存（push）され、図形終了セグメントが現れたときに復帰（pop）される。そうでない場合は、図形開始セグメントによってパラメータがリセットされる（TAD の構造〔0.1〕を参照）。





1.4 図形終了セグメント



【目的・機能】

このセグメントは、図形開始セグメント〔1.3〕と対になり、図形開始セグメントと図形終了セグメントで囲まれた範囲が図形であることを意味する。図形開始セグメント〔1.3〕を参照。

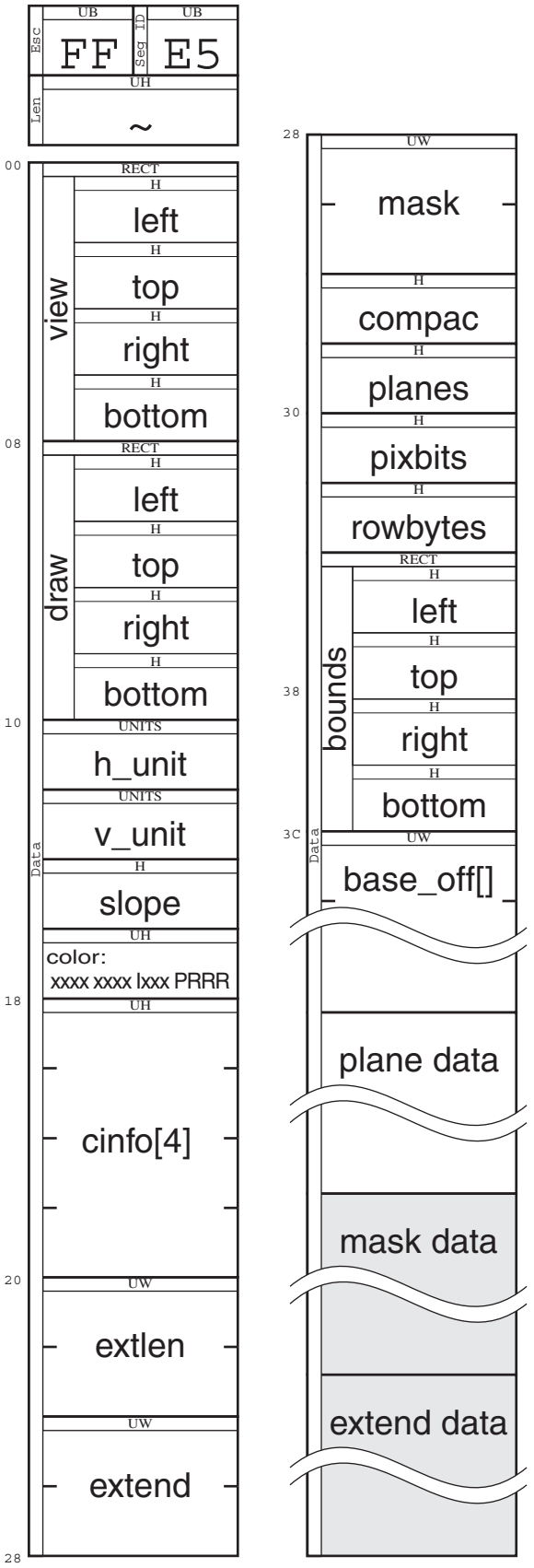
【用法】

len = 0 なので、パラメータはない。

【制限・禁止事項】

図形開始セグメント〔1.3〕を参照。

1.5 画像セグメント（共通）



【目的・機能】

画像セグメントは、ドット（点）の集まり（ビットマップ）によって画像を表現する。たとえば写真などを、TAD 中で表現するために用意されている。ビットマップは一つ一つのドットの色をすべて記述するため、データ量が大きくなりやすい。このため、少しでもデータ量を小さくできるよう、多くの形式が用意されている。

【用法】

color により、ドットの色を数値化する方法を選択することができ、白黒 2 値からフルカラーまでサポートすることができる。カラーについては、画像セグメントでは RGB カラー以外に、CMY カラーで指定することもできる。

color の R = 0 の場合が白黒 2 値、R = 1 の場合、RGB カラーとなり、R = 2 の場合 CMY カラー表現となる。

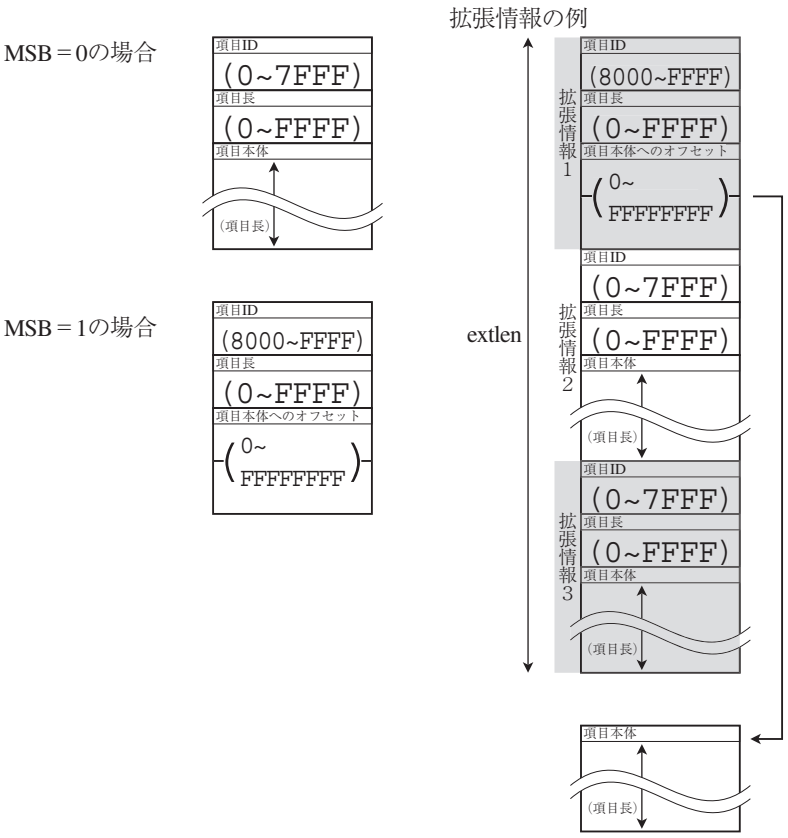
また、color の P = 0 でカラーマップなしの直接方式、P = 1 でカラーマップ方式（パレット方式）となる。

color の I は通常表現／反転表現を指示し、ピクセル値 0 が白、最大値が黒となる通常表現の場合 I = 0、ピクセル値 0 が黒、最大値が白となる反転表現の場合 I = 1 を指定する。

color によってデータの構造がかわるため、本書ではカラー方式ごとにページをわけて解説している。それぞれカラーマップ方式〔1.5a〕、直接白黒方式〔1.5b〕、直接 RGB 方式〔1.5c〕、直接 CMY 方式〔1.5d〕である。

extend data は、画像セグメントの将来の拡張用に用意されている。拡張情報は 1 つ以上の「項目」が並んだ形式を取り、拡張情報へのオフセット（extend）と拡張情報長（extlen）によって示される。項目の終端は特に示されないので、拡張情報長から判断する。拡張情報がない場合は、拡張情報長（extlen）を 0 にする。

図：画像セグメントの拡張情報



拡張情報の「項目」は、項目ID（ext_id：2バイト）、項目長（len：2バイト）、項目本体（data）または項目オフセット（offset）からなる。拡張情報の項目IDのMSB（0x8000）が0の時は、項目に項目本体が含まれるため、項目の総バイト数は項目長+4バイトとなる。項目IDのMSBが1の時は、項目には項目オフセット（4バイト）のみが含まれるため、項目の総バイト数は8バイトとなる。このため未知の項目IDがあっても、その項目のみを読み飛ばすことができる。

現在定義されている項目は、項目ID＝0の「背景色指定」のみである。項目長は4バイトで、内容はCOLORとなっている。画像セグメントではすべてのドットの色が直接指定されるため背景色は意味を持たないが、実装例では画像編集時の情報として利用されている。「背景色指定」を解釈しなくても、表示・印刷には影響はない。

mask dataは、画像の一部を透明にする場合に指定する。透明になった部分は、下に重なっている図形が見えることになる。

maskはマスクへのオフセットを示す。マスクは、planes＝1、pixbits＝0x0101、rowbytesを最適（座標定義による幅が16ドット以下の場合は2バイト、32ドット以下の場合は4バイト、48ドット以下の場合は6バイト）としたプレーンデータである。ビットが0の部分は、画像が透明であることを示し、プレーンデータによるピクセル値は意味を持たない。マスクがない場合（透明部分がない場合）は、マスクオフセット（mask）を0にする。

compac・planes・pixbitsにより、数値化されたドットの色を、どのように実際のバイト列に変換するかが指定される。圧縮形式は、0が無圧縮、1がMH圧縮、2がMR（K＝2）圧縮、3がMR（K＝4）圧縮である。MH圧縮・MR圧縮ではプレーンデータをビット列とみなして圧縮をおこなうが、白黒2値以外では圧縮効果が期待できず、実装例もない。「プレーン数」は、たとえば24ビットの数値を、24ビット単位で書き込まれる一つのプレーンにしたり、8ビット単位で書き込まれる3つのプレーンにしたりという指定をおこなうのに利用される。「ピクセルビット数」は、たとえばプレーンに8ビット単位で書き込むが、実際には下位6ビットのみを使う、といった指定をおこなうのに利用される。実装例では、プレーン数＝4・ピクセルビット数＝1で示される16色形式、プレーン数＝1・

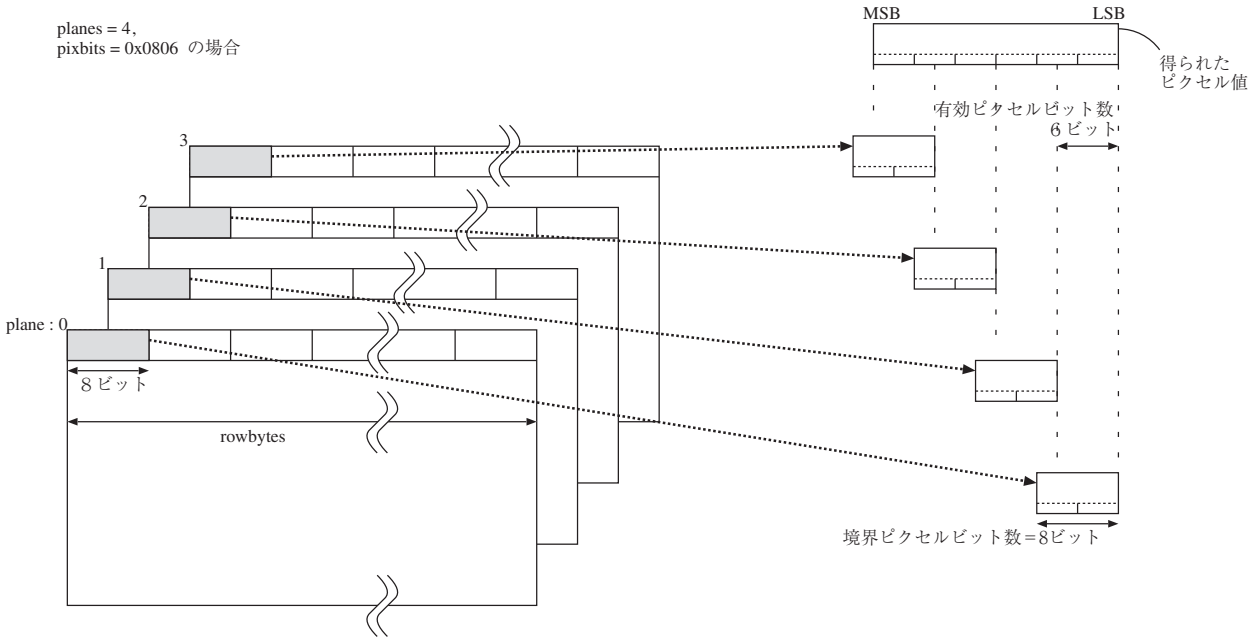
ピクセルビット数= 8 で示される 256 色形式、プレーン数= 1・ピクセルビット数= 16 で示される 65536 色形式などが利用される。

rowbytes は、1 プレーンの 1 行のバイト数を示す。必ず偶数とする。座標定義から決定される最小値より大きくても構わない。

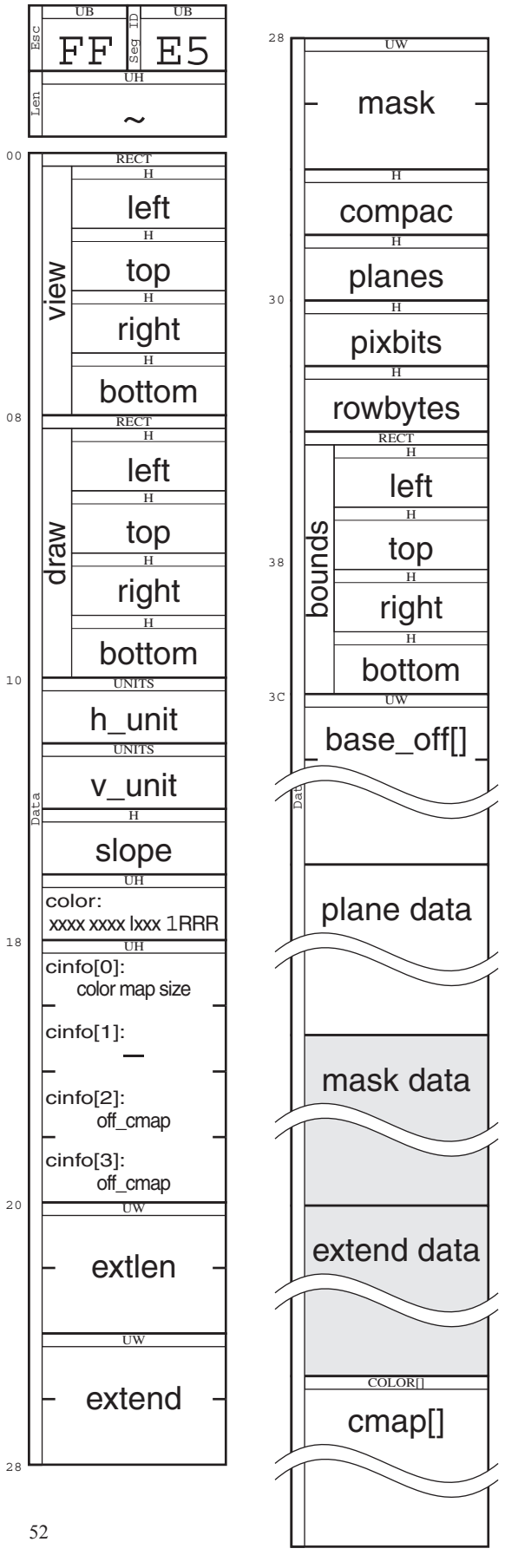
bounds はビットマップの座標の範囲を示す。half-open property で示されるため、right と bottom は、実際の座標よりも 1 だけ大きくする必要がある。各プレーンデータは、rowbytes ×（bounds.bottom−bounds.top）のバイトサイズを持つことになる。bounds と draw が一致していない場合、bounds と draw の重なる領域（AND 領域）のみを描画することを推奨する。

plane data は、各プレーンのデータ（ビット列）である。ピクセルビット数が 8 以下の場合、MSB 側が左のドットを示す。ピクセルビット数が 16・24・32 の場合、準 TAD では下位バイトが先にくる（リトルエンディアン形式）。

なお、プレーンデータ・マスク・拡張情報は、セグメント内であればどこを指していてもよい。順序も任意でよい。ただし、アプリケーションによって更新された場合に、順序などが保持されるとは限らない。また、データが重なっている（複数のプレーンが同じデータを参照したり、プレーンとマスクが同じデータを参照している）場合、保存時には展開されて別々の領域を指すようになる可能性があるので、注意が必要である。



1.5a 画像セグメント(カラーマップ)



【目的・機能】

画像セグメントは、画像（ビットマップ）を表現する。画像セグメントにはカラーマップ方式・直接白黒方式・直接RGB方式･直接CMY方式がある。画像セグメント(共通)〔1.5〕を参照のこと。ここではカラーマップ方式を解説する。

【用法】

文章TAD中または図形TAD中に、画像を置きたい場合に使用する。以下のパラメータにより、画像を表現する。

viewは表示領域を示す。外側が文章の場合、viewで示される大きさの領域が現在位置に確保され、画像が表示される。外側が図形の場合、viewで示される位置・大きさの領域に、画像が表示される。文章TAD中の画像データの場合、viewが空の場合には文字サイズ比例となる(文字サイズ指定付箋〔2.2.2〕を参照)。

drawは描画領域を示す。drawで示される範囲が、viewに対応する形で描画されることになる。後述するboundsの項を参照。drawが空の場合、draw = viewと見なされる。

h_unit・v_unitは無効である。書き出す場合は0を推奨する。

slopeは回転角度を示す。負の値は回転角度が未定義であることを示し、表示時は回転角度= 0として解釈することを推奨する。たとえば60を指定すると、表示領域の左上隅を中心に、60°反時計回りに回転した形で描画されることを示す。

colorはカラー方式を示し、カラーマップの場合は通常は0x89を指定する。0x09を指定することも可能であるが、カラーマップは絶対RGBで指定するため、意味はない。

cinfoは、カラーマップの情報を示す。cinfo[0]がカラーマップのバイト数を示し、cinfo[2]とcinfo[3]とでカラーマップ配列のオフセットを示す。cinfo[2]が上位16ビットを、

cinfo[3] が下位 16 ビットをそれぞれ示す。cinfo[0] が 0 の場合、カラーマップは未定義であることになり、この場合はデータ定義セグメントのカラーマップ定義の指定を利用することを推奨する。cinfo[0] で示せるのは最大 65532（64K−4）バイトであるため、最大 16383 色のカラーマップを定義できる（カラーマップ定義セグメントでは最大 65535 色のカラーマップを定義できる）。

カラーマップ本体は、カラー表現（COLOR）がピクセル値順に並んだ配列である。

extlen は拡張情報の長さを、extend は拡張情報へのオフセットを示す。extlen = 0 の場合は拡張情報がないことを示し、extend は無視される。拡張情報の内容は、画像セグメント（共通）〔1.5〕を参照。

mask はマスクへのオフセットを示す。マスクは、planes = 1、pixbits = 0x0101、rowbytes を最適（座標定義による幅が 16 ドット以下の場合は 2 バイト、32 ドット以下の場合は 4 バイト、48 ドット以下の場合は 6 バイト）としたプレーンデータである。ビットが 0 の部分は、画像が透明であることを示し、プレーンデータによるピクセル値は意味を持たない。マスクがない場合（透明部分がない場合）は、マスクオフセット（mask）を 0 にする。

planes は画像を構成するプレーン数を、pixbits は各プレーンのビット構造を示す。planes と pixbits によって、ピクセルのビット幅（最大 28）が決まる。pixbits の上位 8 ビットは、ビットマップの 1 プレーン上で 1 ピクセルが何ビットになるか（境界ビット数）を指定し、1・2・4・8・16・24・32 のいずれかを指定することを強く推奨する。pixbits の下位 8 ビットは、区切られたビットのうち実際に使用されるビット数（有効ビット数）を指定する。pixbits = 0x2018 の場合、境界ビット数= 32 なので 1 ピクセル= 4 バイトとなり、有効ビット数= 24 により下位 24 ビットが実際に使用されることを示す。

rowbytes はビットマップの一行のバイト数（必ず偶数）を、bounds はビットマップの原点と大きさを示す。ビットマップデータを読み取る場合、幅は rowbytes から、高さは bounds から得る。rowbytes = 30、bounds = {10,20,50,80} とした場合、ビットマップデータ（無圧縮の場合）は、1 行が 30 バイトで、80−20 = 60 行であるから、1 プレーンあたり 1800 バイトとなる。ここで、ビットマップの幅は 50−10 = 40 ドットであるが、もし pixbits = 0x0404（1 ピクセルあたり 4 ビット、つまり 2 ピクセルで 1 バイト）であれば、rowbytes の最適値は 20 となる。しかしこの例のように、bounds と pixbits から計算される最適値よりも rowbytes を大きくして構わなく、bounds から計算される横幅よりも右にあるデータは、読み捨てられることになる。また、このビットマップの左上の点の座標は、bounds によって {10,20} となる。

bounds はビットマップの座標の範囲を示す。half-open property で示されるため、right と bottom は、実際の座標よりも 1 だけ大きくする必要がある。各プレーンデータは、rowbytes ×（bounds.bottom−bounds.top）のバイトサイズを持つことになる。bounds と draw が一致していない場合、bounds と draw の重なる領域（AND 領域）のみを描画することを推奨する。

base_off は、各プレーンの、ビットマップデータのオフセットを示す。base_off が 0 の場合、そのプレーンは未定義であることを示す。

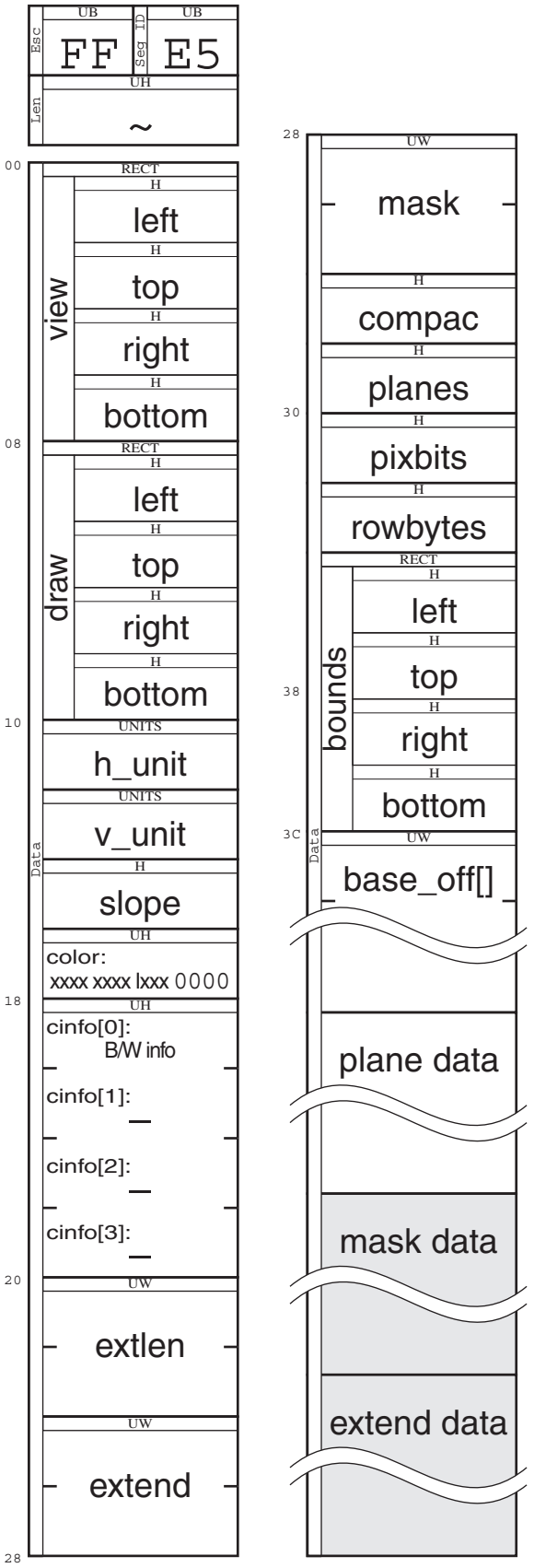
【制限・禁止事項】

（マスクを使用しない場合においては）rowbytes を、bounds と pixbits から計算される必要サイズより大きくしても構わない。ただし、bounds で示される範囲の外については、各プレーンの内容は保持されるとは限らないので注意が必要である。同様に、未使用の領域（どのオフセットからも参照されていない領域）があっても構わないが、これも保持されるとは限らないので注意が必要である。

マスクのビットマップ型式は、画像本体のビットマップ型式と同じであることが要求されているが、これは pixbits ≠ 0x0101 の場合には一意に解釈できない。実装例では、planes = 1、pixbits = 0x0101 とし、rowbytes を最適としたものが採用されているようであるため、本書でもこの方法を推奨する。ただしビットマップ本体の pixbits = 0x0101 で、rowbytes が必要よりも大きい場合には、マスクは（仕様書通りに解釈して）本体と同じ rowbytes で解釈するか、それとも（pixbits ≠ 0x0101 の場合との整合性を重視して）最適な rowbytes を使用するかで混乱が生じる。そこで、マスクを使用する場合は、ビットマップ本体の rowbytes を最適とすることを推奨する。

slope の指定によっては、セグメントの描画範囲の座標が 0 から 32,767 の範囲を越える場合がある。

1.5b 画像セグメント（直接白黒）



【目的・機能】

画像セグメントは、画像（ビットマップ）を表現する。画像セグメントにはカラーマップ方式・直接白黒方式・直接RGB方式･直接CMY方式がある。画像セグメント（共通）〔1.5〕を参照のこと。ここでは直接白黒方式を解説する。

【用法】

文章TAD中または図形TAD中に、画像を置きたい場合に使用する。以下のパラメータにより、画像を表現する。

viewは表示領域を示す。外側が文章の場合、viewで示される大きさの領域が現在位置に確保され、画像が表示される。外側が図形の場合、viewで示される位置・大きさの領域に、画像が表示される。文章TAD中の画像データの場合、viewが空の場合には文字サイズ比例となる（文字サイズ指定付箋〔2.2.2〕を参照）。

drawは描画領域を示す。drawで示される範囲が、viewに対応する形で描画されることになる。後述するboundsの項を参照。drawが空の場合、draw = viewと見なされる。

h_unit・v_unitは無効である。書き出す場合は0を推奨する。

slopeは回転角度を示す。負の値は回転角度が未定義であることを示し、表示時は回転角度= 0として解釈することを推奨する。たとえば60を指定すると、表示領域の左上隅を中心に、60°反時計回りに回転した形で描画されることを示す。

colorはカラー方式を示し、直接白黒の場合は通常は0x80を指定する。0x00を指定することも可能で、この場合は明暗が逆転し、ピクセル値0が最大輝度を示すようになる。

cinfoは、輝度の数値が、ピクセル値のどのビットに対応するかを指定する。cinfo[0]のみを使用し、上位8ビットが

ビット位置、下位 8 ビットがビット幅を示す。たとえば cinfo[0] が 0x0406 の場合、(ピクセル値＞＞ 4) & 0x3f とすることで輝度の値を得ることができる (0x3f はビット表現で 00111111 となり、6 ビット分のマスクとなる)。0x0008 で 8 ビット階調、0x0010 で 16 ビット階調となる。

extlen は拡張情報の長さを、extend は拡張情報へのオフセットを示す。extlen = 0 の場合は拡張情報がないことを示し、extend は無視される。拡張情報の内容は、画像セグメント(共通)〔1.5〕を参照。

mask はマスクへのオフセットを示す。マスクは、planes = 1、pixbits = 0x0101、rowbytes を最適(座標定義による幅が 16 ドット以下の場合は 2 バイト、32 ドット以下の場合は 4 バイト、48 ドット以下の場合は 6 バイト)としたプレーンデータである。ビットが 0 の部分は、画像が透明であることを示し、プレーンデータによるピクセル値は意味を持たない。マスクがない場合 (透明部分がない場合) は、マスクオフセット (mask) を 0 にする。

compac は圧縮型式を示す。0 が無圧縮、1 が MH 圧縮、2 が MR(K = 2) 圧縮、3 が MR(K = 4) 圧縮である。圧縮型式の詳細については、画像セグメント(共通)〔1.5〕を参照。

planes は画像を構成するプレーン数を、pixbits は各プレーンのビット構造を示す。planes と pixbits によって、ピクセルのビット幅(最大 28)が決まる。pixbits の上位 8 ビットは、ビットマップの 1 プレーン上で 1 ピクセルが何ビットになるか(境界ビット数)を指定し、1・2・4・8・16・24・32 のいずれかを指定することを強く推奨する。pixbits の下位 8 ビットは、区切られたビットのうち実際に使用されるビット数(有効ビット数)を指定する。pixbits = 0x2018 の場合、境界ビット数= 32 なので 1 ピクセル= 4 バイトとなり、有効ビット数= 24 により下位 24 ビットが実際に使用されることを示す。

rowbytes はビットマップの一行のバイト数 (必ず偶数) を、bounds はビットマップの原点と大きさを示す。ビットマップデータを読み取る場合、幅は rowbytes から、高さは bounds から得る。rowbytes = 30、bounds = {10,20,50,80} とした場合、ビットマップデータ(無圧縮の場合)は、1 行が 30 バイトで、80−20 = 60 行であるから、1 プレーンあたり 1800 バイトとなる。ここで、ビットマップの幅は 50−10 = 40 ドットであるが、もし pixbits = 0x0404 (1 ピクセルあたり 4 ビット、つまり 2 ピクセルで 1 バイト) であれば、rowbytes の最適値は 20 となる。しかしこの例のように、bounds と pixbits から計算される最適値よりも rowbytes を大きくして構わなく、bounds から計算される横幅よりも右にあるデータは、読み捨てられることになる。また、このビットマップの左上の点の座標は、bounds によって {10,20} となる。

bounds はビットマップの座標の範囲を示す。half-open property で示されるため、right と bottom は、実際の座標よりも 1 だけ大きくする必要がある。各プレーンデータは、rowbytes × (bounds.bottom−bounds.top) のバイトサイズを持つことになる。

boundsとdrawが一致していない場合、boundsとdrawの重なる領域（AND領域）のみを描画することを推奨する。

base_offは、各プレーンの、ビットマップデータのオフセットを示す。base_offが0の場合、そのプレーンは未定義であることを示す。

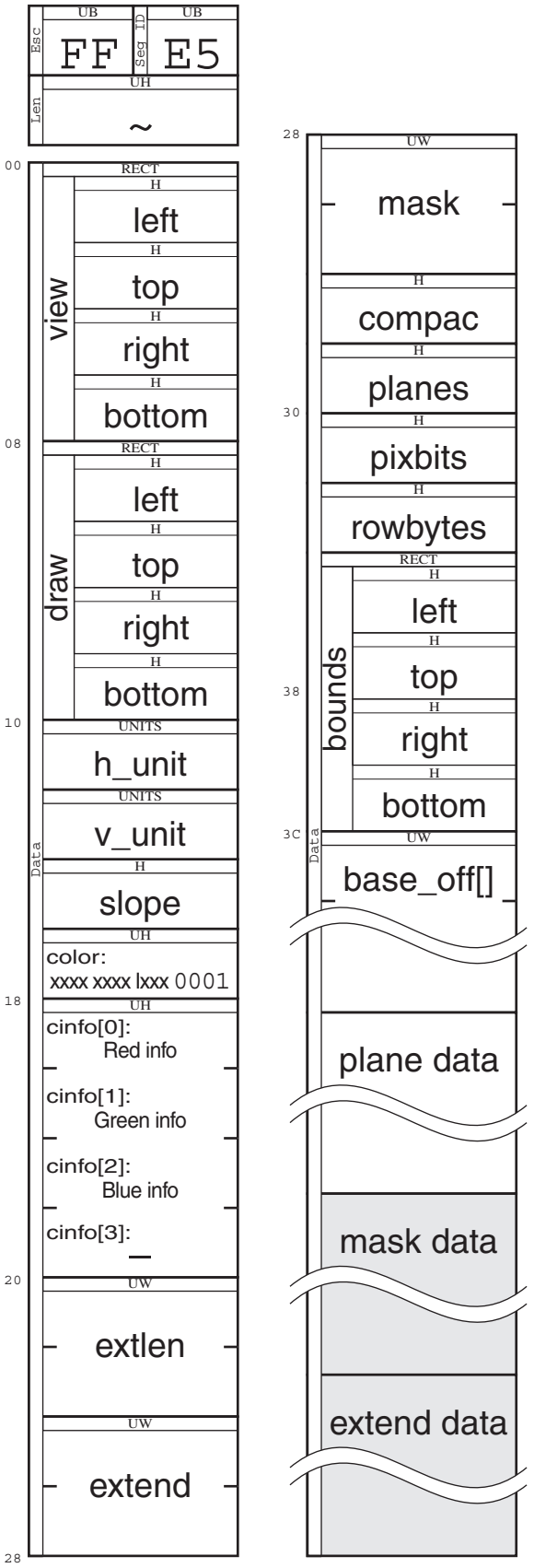
【制限・禁止事項】

（マスクを使用しない場合においては）rowbytesを、boundsとpixbitsから計算される必要サイズより大きくしても構わない。ただし、boundsで示される範囲の外については、各プレーンの内容は保持されるとは限らないので注意が必要である。同様に、未使用の領域（どのオフセットからも参照されていない領域）があっても構わないが、これも保持されるとは限らないので注意が必要である。

マスクのビットマップ型式は、画像本体のビットマップ型式と同じであることが要求されているが、これはpixbits ≠ 0x0101の場合には一意に解釈できない。実装例では、planes = 1、pixbits = 0x0101とし、rowbytesを最適としたものが採用されているようであるため、本書でもこの方法を推奨する。ただしビットマップ本体のpixbits = 0x0101で、rowbytesが必要よりも大きい場合には、マスクは（仕様書通りに解釈して）本体と同じrowbytesで解釈するか、それとも（pixbits ≠ 0x0101の場合との整合性を重視して）最適なrowbytesを使用するかで混乱が生じる。そこで、マスクを使用する場合は、ビットマップ本体のrowbytesを最適とすることを推奨する。

slopeの指定によっては、セグメントの描画範囲の座標が0から32,767の範囲を越える場合がある。

1.5c 画像セグメント（直接RGB）



【目的・機能】

画像セグメントは、画像（ビットマップ）を表現する。画像セグメントにはカラーマップ方式・直接白黒方式・直接RGB方式･直接CMY方式がある。画像セグメント（共通）〔1.5〕を参照のこと。ここでは直接 RGB 方式を解説する。

【用法】

文章 TAD 中または図形 TAD 中に、画像を置きたい場合に使用する。以下のパラメータにより、画像を表現する。

view は表示領域を示す。外側が文章の場合、view で示される大きさの領域が現在位置に確保され、画像が表示される。外側が図形の場合、view で示される位置・大きさの領域に、画像が表示される。文章 TAD 中の画像データの場合、viewが空の場合には文字サイズ比例となる（文字サイズ指定付箋〔2.2.2〕を参照）。

draw は描画領域を示す。draw で示される範囲が、view に対応する形で描画されることになる。後述する bounds の項を参照。draw が空の場合、draw = view と見なされる。

h_unit・v_unit は無効である。書き出す場合は 0 を推奨する。

slope は回転角度を示す。負の値は回転角度が未定義であることを示し、表示時は回転角度= 0 として解釈することを推奨する。たとえば 60 を指定すると、表示領域の左上隅を中心に、60°反時計回りに回転した形で描画されることを示す。

color はカラー方式を示し、直接 RGB の場合は通常は 0x81 を指定する。0x01 を指定することも可能で、この場合は明暗が逆転し、ピクセル値 0 が最大輝度を示すようになる。

cinfo は、RGB の数値が、ピクセル値のどのビットに対応するかを指定する。cinfo[0] は R（赤）に、cinfo[1] は G

（緑）に、cinfo[2] は B（青）に、それぞれ対応する。値の意味はそれぞれ、上位 8 ビットがビット位置、下位 8 ビットがビット幅を示す。たとえば cinfo[0] が 0x0406 の場合、（ピクセル値＞＞ 4）& 0x3f とすることで R（赤）の値を得ることができる（0x3f はビット表現で 00111111 となり、6 ビット分のマスクとなる）。

extlen は拡張情報の長さを、extend は拡張情報へのオフセットを示す。extlen = 0 の場合は拡張情報がないことを示し、extend は無視される。拡張情報の内容は、画像セグメント（共通）〔1.5〕を参照。

mask はマスクへのオフセットを示す。マスクは、planes = 1、pixbits = 0x0101、rowbytes を最適（座標定義による幅が 16 ドット以下の場合は 2 バイト、32 ドット以下の場合は 4 バイト、48 ドット以下の場合は 6 バイト）としたプレーンデータである。ビットが 0 の部分は、画像が透明であることを示し、プレーンデータによるピクセル値は意味を持たない。マスクがない場合（透明部分がない場合）は、マスクオフセット（mask）を 0 にする。

compac は圧縮型式を示す。0 が無圧縮、1 が MH 圧縮、2 が MR（K = 2）圧縮、3 が MR（K = 4）圧縮である。圧縮型式の詳細については、画像セグメント（共通）〔1.5〕を参照。

planes は画像を構成するプレーン数を、pixbits は各プレーンのビット構造を示す。planes と pixbits によって、ピクセルのビット幅（最大 28）が決まる。pixbits の上位 8 ビットは、ビットマップの 1 プレーン上で 1 ピクセルが何ビットになるか（境界ビット数）を指定し、1・2・4・8・16・24・32 のいずれかを指定することを強く推奨する。pixbits の下位 8 ビットは、区切られたビットのうち実際に使用されるビット数（有効ビット数）を指定する。pixbits = 0x2018 の場合、境界ビット数= 32 なので 1 ピクセル= 4 バイトとなり、有効ビット数= 24 により下位 24 ビットが実際に使用されることを示す。

rowbytes はビットマップの一行のバイト数（必ず偶数）を、bounds はビットマップの原点と大きさを示す。ビットマップデータを読み取る場合、幅は rowbytes から、高さは bounds から得る。rowbytes = 30、bounds = {10,20,50,80} とした場合、ビットマップデータ（無圧縮の場合）は、1 行が 30 バイトで、80−20 = 60 行であるから、1 プレーンあたり 1800 バイトとなる。ここで、ビットマップの幅は 50−10 = 40 ドットであるが、もし pixbits = 0x0404（1 ピクセルあたり 4 ビット、つまり 2 ピクセルで 1 バイト）であれば、rowbytes の最適値は 20 となる。しかしこの例のように、bounds と pixbits から計算される最適値よりも rowbytes を大きくして構わなく、bounds から計算される横幅よりも右にあるデータは、読み捨てられることになる。また、このビットマップの左上の点の座標は、bounds によって {10,20} となる。

bounds はビットマップの座標の範囲を示す。half-open property で示されるため、right と bottom は、実際の座標よりも 1 だけ大きくする必要がある。各プレーンデータは、rowbytes ×（bounds.bottom−bounds.top）のバイトサイズを持つことになる。

boundsとdrawが一致していない場合、boundsとdrawの重なる領域（AND領域）のみを描画することを推奨する。

base_offは、各プレーンの、ビットマップデータのオフセットを示す。base_offが0の場合、そのプレーンは未定義であることを示す。

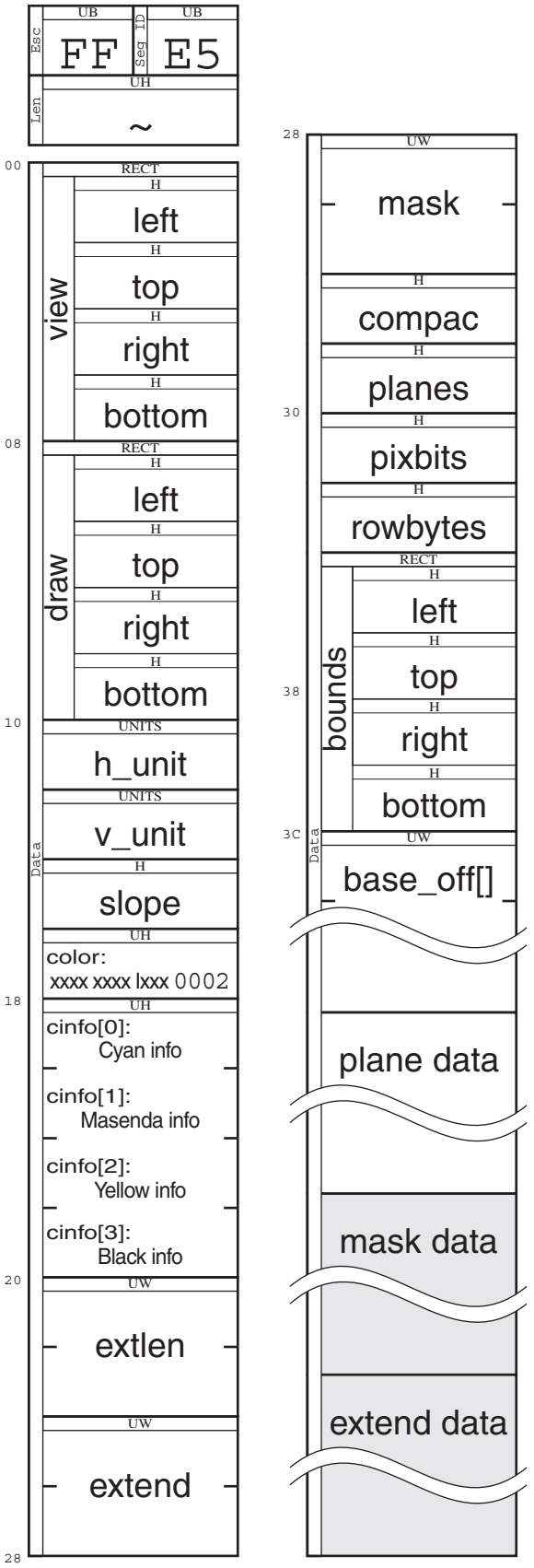
【制限・禁止事項】

（マスクを使用しない場合においては）rowbytesを、boundsとpixbitsから計算される必要サイズより大きくしても構わない。ただし、boundsで示される範囲の外については、各プレーンの内容は保持されるとは限らないので注意が必要である。同様に、未使用の領域（どのオフセットからも参照されていない領域）があっても構わないが、これも保持されるとは限らないので注意が必要である。

マスクのビットマップ型式は、画像本体のビットマップ型式と同じであることが要求されているが、これはpixbits ≠ 0x0101の場合には一意に解釈できない。実装例では、planes = 1、pixbits = 0x0101とし、rowbytesを最適としたものが採用されているようであるため、本書でもこの方法を推奨する。ただしビットマップ本体のpixbits = 0x0101で、rowbytesが必要よりも大きい場合には、マスクは（仕様書通りに解釈して）本体と同じrowbytesで解釈するか、それとも（pixbits ≠ 0x0101の場合との整合性を重視して）最適なrowbytesを使用するかで混乱が生じる。そこで、マスクを使用する場合は、ビットマップ本体のrowbytesを最適とすることを推奨する。

slopeの指定によっては、セグメントの描画範囲の座標が0から32,767の範囲を越える場合がある。

1.5d 画像セグメント(直接CMY)



【目的・機能】

画像セグメントは、画像（ビットマップ）を表現する。画像セグメントにはカラーマップ方式・直接白黒方式・直接RGB方式･直接CMY方式がある。画像セグメント(共通)〔1.5〕を参照のこと。ここでは直接CMY方式を解説する。

【用法】

文章TAD中または図形TAD中に、画像を置きたい場合に使用する。以下のパラメータにより、画像を表現する。

viewは表示領域を示す。外側が文章の場合、viewで示される大きさの領域が現在位置に確保され、画像が表示される。外側が図形の場合、viewで示される位置・大きさの領域に、画像が表示される。文章TAD中の画像データの場合、viewが空の場合には文字サイズ比例となる（文字サイズ指定付箋〔2.2.2〕を参照）。

drawは描画領域を示す。drawで示される範囲が、viewに対応する形で描画されることになる。後述するboundsの項を参照。drawが空の場合、draw = viewと見なされる。

h_unit・v_unitは無効である。書き出す場合は0を推奨する。

slopeは回転角度を示す。負の値は回転角度が未定義であることを示し、表示時は回転角度= 0として解釈することを推奨する。たとえば60を指定すると、表示領域の左上隅を中心に、60°反時計回りに回転した形で描画されることを示す。

colorはカラー方式を示し、直接CMYの場合は通常は0x02を指定する（CMYは、値が大きいほど暗くなるのが一般的と考えられる）。0x82を指定することも可能で、この場合は明暗が逆転し、ピクセル値0が最大濃度を示すようになる。

cinfo は、CMYK の数値が、ピクセル値のどのビットに対応するかを指定する。cinfo[0] は C（シアン／水色）に、cinfo[1] は M（マゼンダ／紫色）に、cinfo[2] は Y（イエロー／黄色）に、cinfo[3] は K（ブラック／黒）に、それぞれ対応する。値の意味はそれぞれ、上位 8 ビットがビット位置、下位 8 ビットがビット幅を示す。たとえば cinfo[0] が 0x0406 の場合、（ピクセル値＞＞ 4）& 0x3f とすることで C（マゼンダ／紫色）の値を得ることができる（0x3f はビット表現で 00111111 となり、6 ビット分のマスクとなる）。

extlen は拡張情報の長さを、extend は拡張情報へのオフセットを示す。extlen = 0 の場合は拡張情報がないことを示し、extend は無視される。拡張情報の内容は、画像セグメント（共通）〔1.5〕を参照。

mask はマスクへのオフセットを示す。マスクは、planes = 1、pixbits = 0x0101、rowbytes を最適（座標定義による幅が 16 ドット以下の場合は 2 バイト、32 ドット以下の場合は 4 バイト、48 ドット以下の場合は 6 バイト）としたプレーンデータである。ビットが 0 の部分は、画像が透明であることを示し、プレーンデータによるピクセル値は意味を持たない。マスクがない場合（透明部分がない場合）は、マスクオフセット（mask）を 0 にする。

compac は圧縮型式を示す。0 が無圧縮、1 が MH 圧縮、2 が MR（K = 2）圧縮、3 が MR（K = 4）圧縮である。圧縮型式の詳細については、画像セグメント（共通）〔1.5〕を参照。

planes は画像を構成するプレーン数を、pixbits は各プレーンのビット構造を示す。planes と pixbits によって、ピクセルのビット幅（最大 28）が決まる。pixbits の上位 8 ビットは、ビットマップ上で 1 ピクセルが何ビットになるか（境界ビット数）を指定し、1・2・4・8・16・24・32 のいずれかを指定することを強く推奨する。pixbits の下位 8 ビットは、区切られたビットのうち実際に使用されるビット数（有効ビット数）を指定する。pixbits = 0x2018 の場合、境界ビット数= 32 なので 1 ピクセル= 4 バイトとなり、有効ビット数= 24 により下位 24 ビットが実際に使用されることを示す。

rowbytes はビットマップの一行のバイト数（必ず偶数）を、bounds はビットマップの原点と大きさを示す。ビットマップデータを読み取る場合、幅は rowbytes から、高さは bounds から得る。rowbytes = 30、bounds = {10,20,50,80} とした場合、ビットマップデータ（無圧縮の場合）は、1 行が 30 バイトで、80−20 = 60 行であるから、1 プレーンあたり 1800 バイトとなる。ここで、ビットマップの幅は 50−10 = 40 ドットであるが、もし pixbits = 0x0404（1 ピクセルあたり 4 ビット、つまり 2 ピクセルで 1 バイト）であれば、rowbytes の最適値は 20 となる。しかしこの例のように、bounds と pixbits から計算される最適値よりも rowbytes を大きくして構わなく、bounds から計算される横幅よりも右にあるデータは、読み捨てられることになる。また、このビットマップの左上の点の座標は、bounds によって {10,20} となる。

boundsはビットマップの座標の範囲を示す。half-open propertyで示されるため、rightとbottomは、実際の座標よりも1だけ大きくする必要がある。各プレーンデータは、rowbytes×（bounds.bottom−bounds.top）のバイトサイズを持つことになる。boundsとdrawが一致していない場合、boundsとdrawの重なる領域（AND領域）のみを描画することを推奨する。

base_offは、各プレーンの、ビットマップデータのオフセットを示す。base_offが0の場合、そのプレーンは未定義であることを示す。

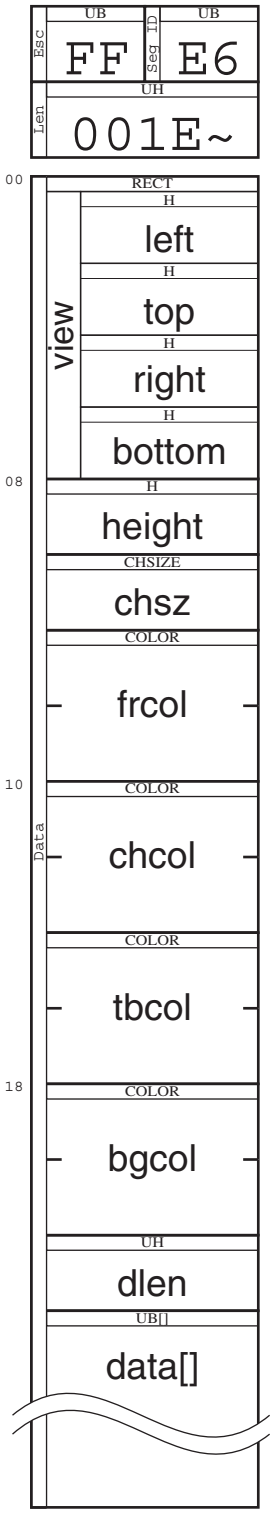
【制限・禁止事項】

（マスクを使用しない場合においては）rowbytesを、boundsとpixbitsから計算される必要サイズより大きくしても構わない。ただし、boundsで示される範囲の外については、各プレーンの内容は保持されるとは限らないので注意が必要である。同様に、未使用の領域（どのオフセットからも参照されていない領域）があっても構わないが、これも保持されるとは限らないので注意が必要である。

マスクのビットマップ型式は、画像本体のビットマップ型式と同じであることが要求されているが、これはpixbits ≠ 0x0101の場合には一意に解釈できない。実装例では、planes = 1、pixbits = 0x0101とし、rowbytesを最適としたものが採用されているようであるため、本書でもこの方法を推奨する。ただしビットマップ本体のpixbits = 0x0101で、rowbytesが必要よりも大きい場合には、マスクは（仕様書通りに解釈して）本体と同じrowbytesで解釈するか、それとも（pixbits ≠ 0x0101の場合との整合性を重視して）最適なrowbytesを使用するかで混乱が生じる。そこで、マスクを使用する場合は、ビットマップ本体のrowbytesを最適とすることを推奨する。

slopeの指定によっては、セグメントの描画範囲の座標が0から32,767の範囲を越える場合がある。

1.6 仮身セグメント



【目的・機能】

このセグメントは、TAD 外のリンクと対になって、仮身を示す。主として、リンクレコードは実身への参照を保持し、仮身セグメントは仮身の表示のされ方を保持する。基本的に BTRON ファイルシステムでのみ有効である。

なお、仮身についてはこのガイドブックでは解説していない。

【用法】

以下のパラメータにより、仮身を指定する。

view は、仮身を表示する領域を指定する。図形中に置く場合には位置・大きさともに有効で、文章中に置く場合には大きさのみが有効である。仮身の高さが十分大きい場合は、この仮身が開いた仮身であることを意味する。ただし、実際の閾値は処理系に依存し、TAD では規定されない。

height は、仮身を開いた場合の高さを示す。−1 の場合はその仮身が開けないことを、0 の場合はデフォールトの高さで開くことを意味する。

chsz は、仮身タイトル（実身名・続柄・データタイプ名）の文字サイズを示す。

frcol は、仮身の枠の色を示す。

chcol は、仮身タイトル（実身名・データタイプ名・更新日時）の文字色を示す。

tbcol は、仮身の背景色を示す。

bgcol は、開いた仮身の表示領域の背景色を示す。

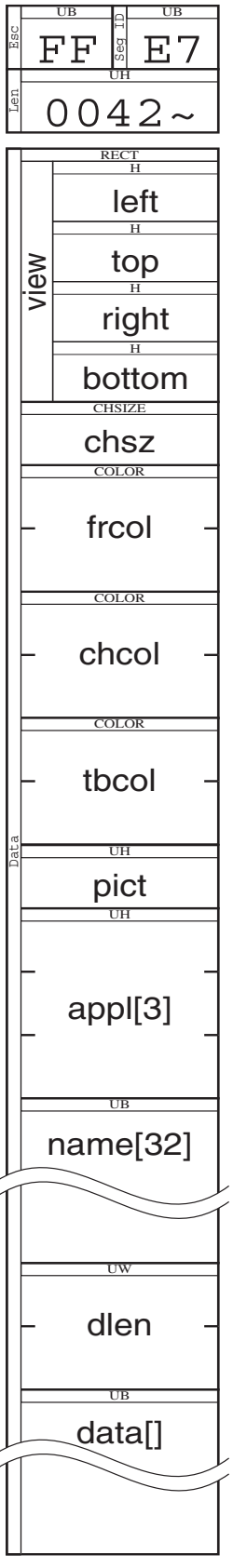
dlen は、固有データのバイト数を示す。

data は、固有データを示す。内容はアプリケーションに依存する。

【制限・禁止事項】

BTRON 仕様書を参照。

1.7 指定付箋セグメント



【目的・機能】

このセグメントは、TAD 中に格納されたアプリケーション固有のデータを示す。このセグメントを使用することで、他のアプリケーションと干渉せずに、アプリケーション依存のデータを保持することができる。

【用法】

以下のパラメータにより、指定付箋を指定する。

view は、指定付箋を表示する領域を指定する。図形中に置く場合には位置・大きさともに有効で、文章中に置く場合には大きさのみが有効である。

chsz は、指定付箋タイトル（指定付箋名）の文字サイズを示す。

frcol は、指定付箋の枠の色を示す。

chcol は、指定付箋タイトル（指定付箋名）の文字色を示す。

tbcol は、指定付箋の背景色を示す。

pict は、指定付箋のピクトグラムを示す。0 はデフォールトを意味する。0 とすることを推奨する。

appl は、アプリケーション ID を示す（アプリケーション ID を含む付箋〔0.3.3〕を参照）。

name は、指定付箋の付箋名を示す。一意表現とする（TRON コードを含むセグメント〔0.3.2〕を参照）。

dlen は、固有データのバイト数を示す 32 ビット符号無しデータである。

data は、固有データを示す。内容はアプリケーションに依存する。

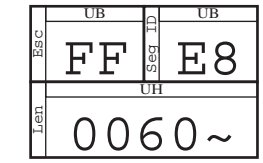
【制限・禁止事項】

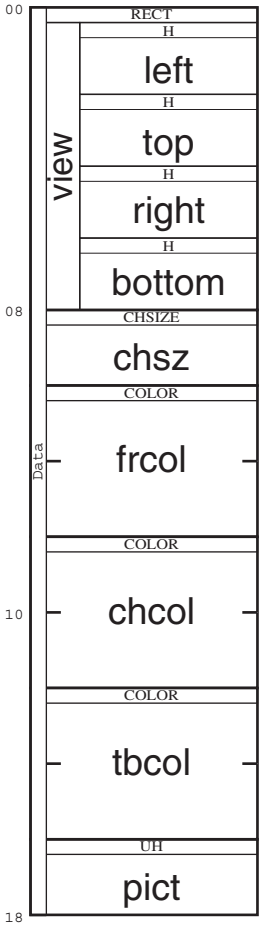
原則として、固有データの内容を知っているアプリケーション（アプリケーション ID が一致するアプリケーション）からしか読み書きできない。ただし他のアプリケーションであっても、指定付箋全体を移動・複写・削除することは可能である。また、他の TAD に移動・複写される可能性もある。

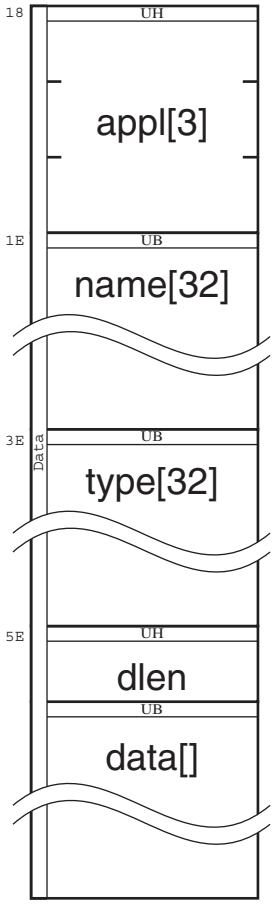
アプリケーション ID は、他のアプリケーションと重複しないように割り当てる必要がある。アプリケーション ID を含む付箋〔0.3.3〕を参照。

解釈できない指定付箋（アプリケーション ID が一致しない指定付箋）は、仮身や機能付箋と同様に表示されると考えられる。

1.8 機能付箋セグメント







【目的・機能】

このセグメントは、アプリケーションを起動するための情報を示す。

【用法】

以下のパラメータにより、機能付箋を指定する。

view は、機能付箋を表示する領域を指定する。図形中に置く場合には位置・大きさともに有効で、文章中に置く場合には大きさのみが有効である。

chsz は、機能付箋タイトル（機能付箋名・データタイプ名）の文字サイズを示す。

frcol は、機能付箋の枠の色を示す。

chcol は、機能付箋タイトル（機能付箋名・データタイプ名）の文字色を示す。

tbcol は、機能付箋の背景色を示す。

pict は、機能付箋のピクトグラムを示す。0 はデフォールトを意味する。0 とすることを推奨する。

appl は、アプリケーション ID を示す（アプリケーション ID を含む付箋〔0.3.2〕を参照）。

name は、機能付箋の付箋名を示す。一意表現とする（TRON コードを含むセグメント〔0.3.2〕を参照）。

type は、機能付箋のデータタイプ名を示す。一意表現とする（TRON コードを含むセグメント〔0.3.2〕を参照）。データタイプ名は、アプリケーション ID の上位 2 ワードに対応したものとなる。

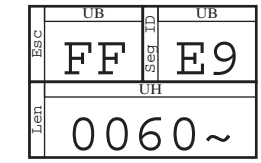
dlen は、固有データのバイト数を示す。

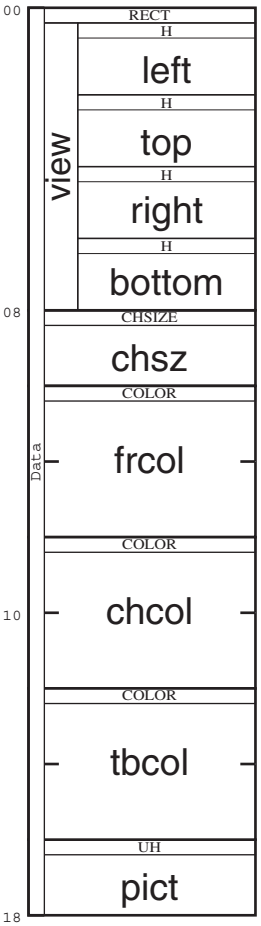
data は、固有データを示す。内容はアプリケーションに依存する。

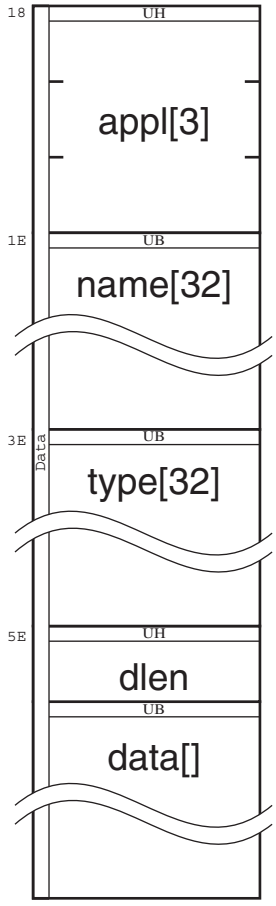
【制限・禁止事項】

BTRON 仕様書を参照。

1.9 設定付箋セグメント







【目的・機能】

システムの設定変更用の付箋である、とされる。

【用法】

仕様書には、変数参照指定付箋（指定）〔2.13.1〕において参照される変数を、実行環境に登録するために利用されるとの記述がある。しかし規格本体内において設定付箋についての記述はセグメントIDが規定されているだけで、実際の構造等は全く不明である。

同時に、用法についても、詳しいことは全く分からない。

左の図は、BTRON OSの仕様書の中の実身／仮身マネージャ仕様書の中の記述等から推測される構造である。

【制限・禁止事項】

仕様書には以下の記述があるが、このセグメントを使用するのに十分な情報はない。このセグメントを使用しないことを推奨する。

［BTRON1 プログラミング標準ハンドブック 付 2−69 4.8.1 変数参照指定付箋］

参照する変数は、特定のIDまたは名前であらかじめユーザーの実行環境に登録されている必要があり、この登録は基本的に設定付箋により行われる。

［BTRON1 プログラミング標準ハンドブック 11−17 1.5.1 付箋の種類］

設定付箋：システムの設定変更用　指定付箋以外は、いずれもアプリケーションプログラムを直接参照し、付箋の処理のために基本的に対応するアプリケーション・プログラムの起動を必要とする。

実身／仮身マネージャでは、指定付箋以外の付箋に関する表示／操作／実行処理のための各種関数を提供している。