[metaChart]
設定ファイルの文法

[metaChart]は paperChart のための解釈機械(interpretive machine)です.
paperChart は[metaChart]によって解釈実行される仮想機械記述ファイルの集合体です.

各設定ファイルの具体的な解説は「設定ファイルの書き方.pdf」をご覧ください.
メニューファイルの書き方と MA2.exe の使い方は「メニューファイルの書き方.pdf」をご覧ください.

辞書の文法

設定ファイルは，項目名とその値の並びです．この paperChart システムの動作を規定する辞書のようなものです．paperChart システムの各実行モジュールは起動時にこの辞書を読み，そこに定められた動作をするように作られています．設定ファイルを読むのは各ソフトの起動時に一回だけです．起動してから設定ファイルの内容を変更してもソフトの動作には反映されません．ご使用になる施設にあわせて，この辞書を利用者ご自身で書き換えていただきます．

例：

```
parameters
{
    NBP // 非観血血圧
    {
        systolic
        {
            symbol = color 192 0 0;
            symbol = polygon -14 -14 0 0 14 -14;
        }
        diastolic
        {
            symbol = color 192 0 0;
            symbol = polygon -14 14 0 0 14 14;
        }
        alternative_mean
        {
            symbol = color 192 0 0;
            symbol = polygon -14 -14 0 0 14 -14;
            symbol = polygon -14 14 0 0 14 14;
        }
        ordinate = hipress;
        option = median;
    }
    HR  // 心拍数
    {
        symbol = color 0 192 0;
        symbol = polygon -6 0 0 -6 6 0 0 6;
        ordinate = hipress;
        option = median;
    }
}
```

この例では，まず parameters という大項目名があり，半角大括弧({と})で括られた中に NBP や HR などの項目名があり，NBP の中にはさらに systolic や diastolic などの項目名が並びます．
項目名は多重に入れ子にすることができます．
半角イコール(=)の左辺には最終的な項目名が並び，右辺にはその値が並びます．
値の最後は半角セミコロン(;)で終わります．それぞれの名前や値は空白やタブ，改行で区切られます．
改行は単なる区切りであって，それ以上の特別な意味を持ちません．各自が見やすいように改行してください．
2つの半角スラッシュ(//)から行末までは単なるコメント(読む人のための覚書き)で，[metaChart]はこれを読み飛ばします．
一続きの値や名前に空白文字が含まれるときは，その名前や値全体を半角ダブルクォーテーションで囲んでください．
値や名前の中に改行やタブなどの制御文字(0x00～0x1f)を含めることはできません．漢字はシフトJISコードのみ使用できます．

例：″salty dog″

同様に値や名前が;や=や{や}を含むときにも，その語全体を半角ダブルクォーテーションで囲んでください．語の中に半角ダブルクォーテーション自体を含めたいときには，半角ダブルクォーテーションを2つ続けて書いてください．

例：″a″″b″″c″
は a″b″c と解釈されます．名前や値は，たとえ半角ダブルクォーテーションで囲んでも，複数行にわたって書くことはできません．

例：
```
大項目名１
{
    中項目名１
    {
        小項目名１ = 値１ 値２ 値3;
        小項目名２ = 値４ 値５ 値6;
        …
    }
    中項目名２
    {
        小項目名３ = 値７ 値８ 値9;
        …
    }
    …
}
```

文法上は全てを1行に続けて，
大項目名１{中項目名１{小項目名１= 値１ 値２ 値3; 小項目名２ = 値４ 値５ 値6;} …
と書いても構いません．
インデントは読みやすいように，各自で工夫してください．
ただし，一行あたりの文字数は半角で 1000 字，全角で 500 字以内にしてください．
設定ファイルを編集するときは「メモ帳」アプリケーションを使ってください．MSWORD 等を使うと，自動校正で文字が置き換えられたりして，不都合が生じます．

色指定の書き方

色の指定は3原色（Red/Green/Blue）をそれぞれ0～255 の数値で指定してください．
たとえば xx_color = 255 255 255 ; という書き方は白を表します．また，0 0 0 は黒，255 0 0 は赤，0 255 0 は緑，0 0 255 は青です．

図形（symbol）の書き方

各種図形（血圧や心拍数の記号など）は
symbol = 動作 座標あるいは色指定 ;
のように書いてください．
例：
```
symbol = color    255 0 0 ;
symbol = line     -6 -6 6 -6 6 6 -6 6 -6 -6 ;
symbol = color    0 0 255 ;
symbol = line     -6 -6 6 6;
symbol = line     -6 6 6 -6 ;
```
これは，記号をプロットする座標の中心から上下左右に6づつ（つまり幅高さそれぞれ１２）の赤い□の内側に青い×を書いたものです．ここでいう“座標”は，後述する dspcnf.txt の screen 節の中の std_width と std_height です．NV のウィンドウはパソコン画面上での大きさが何ドットであっても，縦 std_height ドット，横 std_width ドットの広さとして取り扱います．したがって，この座標系で書かれた symbol はウィンドウの伸縮に合わせてサイズが伸び縮みします（**標準座標**と略します）．X 座標は右方向が＋，Y 座標は下方向が＋です．symbol=color…は描画前景色を指定します．いちど色を指定すると，つぎに別の色を指定するまで同じ色で図形を描きます．symbol で背景色を指定することはできません．背景色は，そのシンボルがどこに描かれるかによって，それぞれ異なるものです．

color　色指定（RGB） ;
　色指定は上記の RGB 値です．

line　X 座標 Y 座標 X 座標 Y 座標 … ;
　それぞれの座標を結ぶ一連の折れ線を書きます．必ず X 座標と Y 座標の２つの数の組を書いてくださ

い.
つまり数値は必ず偶数個になるはずです．座標の数は 30 組以内です.
一筆書きできない線は上記の×の例のように複数の line 文に分けてください.

polygon X 座標 Y 座標 X 座標 Y 座標 … ;
例： symbol = polygon -2 -2 2 -2 2 2 -2 2 ; //原点を中心に上下左右それぞれ２の塗りつぶし四角形
それぞれの座標を結ぶ折れ線で囲まれた範囲を指定色で塗りつぶします．線分が閉じられていないときは，最初と最後の座標を結ぶ直線で図形を閉じます.

hcirc 横半径 縦半径 [中心の横座標 中心の縦座標] ;
例： symbol = hcirc 12 8 0 0 ; //原点を中心に縦半径８，横半径８の中空楕円
“hcirc”は hollow circle の意味で，中抜き円（楕円）を書きます．中心の座標を省略した場合は，円の中心はシンボルの中心になります.

scirc 横半径 縦半径 [中心の横座標 中心の縦座標] ;
例： symbol = scirc 12 8 0 0 ; //原点を中心に縦半径８，横半径８の塗りつぶし楕円
“scirc”は solid cicle の意味で，塗りつぶし円（楕円）を書きます．中心の座標を省略した場合は，円の中心はシンボルの中心になります.

glyph 文字列 フォントの高さ（ドット数） [P][B][I] ;
例： symbol = glyph "rock & roll" 40 P I ; //rock & roll の文字，フォント高さ 40，可変幅斜体
文字自体をシンボルとして使用します． 文字列に空白や＝や;や{や}を含む場合は，文字列をﾀﾞﾌﾞﾙｸｫｰﾃｰｼｮﾝで囲んでください．フォントの高さは先の std_height で表しますので，画面の大きさに合わせて文字も伸縮します．P はプロポーショナル（可変幅）フォント，B はボールド（太字），I はイタリック（斜体）です．何も指定しないと等幅の正立体で書きます.

bitmap ﾋﾞｯﾄﾏｯﾌﾟ ﾌｧｲﾙ名 ;
例： symbol = bitmap heart.bmp ; //CONF¥heart.bmp をシンボルに
photoshop などで作った BMP ファイルをシンボルとして使用します．シンボルの大きさは画面の大きさに合わせて伸縮します．画面上のサイズは縦横それぞれ，ﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙのﾄﾞｯﾄ数×ｳｨﾝﾄﾞｳのﾄﾞｯﾄ数÷std_ﾄﾞｯﾄ数です．ﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙは CONF フォルダに入れてください．ﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙ中の黒ﾄﾞｯﾄ（RGB=0,0,0）は透過色になります．つまり画面上の背景色と置き換えられます．画面上に黒色を表示したいときは代わりに RGB=1,1,1 などをお使いください.

width 太さの数字 ;
例： symbol = width 3 ;
印刷時のみ有効．画面表示では無視します．線の太さを指定します．この文以後の線の太さは，この値になります．この数値は実際のプリンタ上でのドット数になります.

計算式の文法

計算式は各種のサマリーの値を麻酔チャートに記入する際に使用します．たとえば患者年齢や退室時血圧，心肺前後の輸液量などです.
（ 計算式は calcnf.txt 以外のﾌｧｲﾙ内では必ず **calc{ と }** で囲まれた節の中に書いてください． calcnf.txt 内では **calc{ }** で囲む必要はありません．）
例:

```
年令値      = age（'生年月日'）;
年令文字列 = 年令値 < 0.07671 ? 年令値 * 365 + 0.5 # 1 $ '#日'                    :
             年令値 < 0.5     ? 年令値 * 52.14 + 0.5 # 1 $ '#週'                  :
             年令値 < 4       ? 年令値 # 1 $ '#才' @ 年令値 * 12 % 12 # 1 $ '#月' :
                                年令値 # 1 $ '#才' ;
```

age（）は生年月日と名づけられた患者属性データ文字列と現在（モニタ開始日時）から年令を求め，年令値という名前をつけます．この年令値が1ヶ月未満なら“X日”，半年未満なら“X週”，4歳未満なら“X才X月”，それ以上なら“X才”という表記に変換し，年令文字列と名づけます．＝と;は前述の辞書表記と同じものです．式の末尾には必ず半角セミコロンを書いてください.

式の右辺はいったん辞書構文解析により，“age”と“(”，“’生年月日’”，“)”の4つの文字列値として格納されますが，最終的には再度つなぎ合わせて，計算構文解析を行います．つまり右辺の空白をすべて取り除いて，
年令値=age(’生年月日’)；
という1つの辞書式として書いても計算は同じように実行されます．

- **欠損値と暫定値，確定値**

手術時間（手術開始から終了までの時間）について考えてください．手術開始の直後に手術が終了したなら，手術時間は0分あるいは0秒であり，麻酔チャートの手術時間の欄には0分と書かれるべきですが，もし手術自体が行われていない，つまり手術開始も終了も記録されていなければ，チャートの手術時間の欄は空白であるべきです．また，手術終了だけが記録され開始時刻がなければ，ICUかどこかで開胸して，手術室になだれ込んで手術を終えたと考えるべきでしょう．このばあい，手術時間は手術室に入ってから手術終了までと（麻酔チャート上は）するべきでしょう．
本システムでは，1.値が存在しない（欠損値），2.手術開始か終了の片方だけしかない場合にとりあえずモニタ開始から手術終了まで，あるいは手術開始からモニタ終了までを手術時間とする（暫定値），3.手術開始から手術終了までの時間を手術時間とする（確定値），の3種類の値を扱います．

注意：欠損値は“0”ではありません．0はあくまでもゼロという値であり，値が無い（欠損値）状態とは違います．

問題：挿管から抜管までの間，毎時15ml のセボフルレンを消費します．13時0分からモニタが開始され，14時0分に抜管の記載がありましたが，挿管の記載はありません．何 ml のセボフルレンを消費したでしょうか？ 答えは，チャート記載の不備はともかく，「暫定値として15ml」です．

欠損値と確定値（あるいは暫定値）の乗算（×），除算（／），剰余算（%）の答えは欠損値です．
確定値（あるいは暫定値）と欠損値の加減算は欠損値を確定値の0と置き換えて扱います．
つまり １＋欠損値＝1， 欠損値－１＝ －1です．
暫定値と確定値の四則計算の答えは暫定値です．確定値同士の計算の答えは，もちろん確定値です．
式の中に書かれる定数あるいは定数文字列はいずれも確定値として取り扱われます．

- **文字定数の書き方**

定数文字列は半角シングルクォーテーション（’）で括ってください．

定数文字列の中に ＝{ }； などを含んでいてもシングルクォーテーションの外側をさらにダブルクォーテーションで囲む必要はありません．空白文字を含む変数名はダブルクォーテーションで括ることで使用できますが，シングルクォーテーションとの包含関係がややこしくなりますので，おすすめしません．

- **時刻値の内部表現**

絶対時刻はグリニッジ標準時 1970 年1月1日午前0時0分0秒からの秒数で表します（いわゆる unix time）．経過時間（何時何分から何時何分まで）も秒数で表します．

- **書式指定文字列**

本システムでは数値を文字列に変換するために書式指定文字列を使用します．

- **数値の書式指定：**

各桁と小数点の位置を#と. で表してください．左側の空白文字は必要（数値の桁数）に応じて数字をつめます．小数点の左側に#を連ねた場合，ゼロ詰めになります．＋は正の数にも符号を付記することを意味します．
書式の桁数が短すぎても数値の上位桁の切り詰めは起きません．下位桁は四捨五入します．
#を後述の切捨て演算子と混同しないでください．
例：

’#’	1.5⇒”2”	0.47⇒”0”	-1⇒”-1”
’#.#’	1.5⇒”1.5”	0.47⇒”0.5”	-1⇒”-1.0”
’ #.#’	1.5⇒” 1.5”	128⇒”128.0”	-1⇒” -1.0”
’###.#’	1.5⇒”001.5”	128⇒”128.0”	-1⇒”-001.0”
’ +#.##’	1.5⇒” +1.50”	128⇒”+128.00”	-1⇒” -1.00”

- **時刻の書式指定：**

アルファベット小文字**y**で年，**m** で月，**d** で日，**w** で曜，大文字 **H** で時，**M** で分，**S** で秒を表します．
大文字小文字の区別にご注意ください．
y は4つ並べて4桁年号，2つで下2桁年号です．いずれも西暦表示です．年号の1桁や3桁表示はありません．

元号つきの日付を西暦に変換することは era 関数でできますし，age 関数で生年月日から年齢を求めることはできますが，その逆の計算手段はサポートしていません．
m は1つで1桁表示，2つで2桁（上位0詰め）表示，3つ"**mmm**"で3文字（Jan Feb Mar …）表示です．
w は1つで数字表示（日=0，月=1，火=2，… 土=6）．3つ"**www**"で3文字（sun mon tue …）表示です．
d，**H**，**M**，**S** は，いづれも1つで1桁表示，2つで2桁（上位0詰め）表示です．
それ以外の文字は単に転写されるだけです．

'yyyy/mm/dd(www)-HH:MM:SS'	2005/03/08(tue)-09:32:06
'mmm/dd/''yy HH:MM:SS'	Mar/08/'05 09:32:06

↑ 注意：シングルクォーテーション2連続でひとつのシングルクォーテーション文字自体を表します．

'yy/ m/ d(w)- H: M: S'	05/ 3/ 8(2)- 9:32: 6

↑ 注意：0は日曜，2は火曜，・・・6は土曜です．

'yyyy 年 mm 月 dd 日 HH 時 MM 分 SS 秒'	2005 年 03 月 08 日 09 時 32 分 06 秒

（曜日を数字で表す方法は曜日ごとの分類データを Excel へ持ち込んで大小順並べ替えをするときなどにお使いください．）

- **算術演算子**：優先度の高い順に示します．

括弧 ( )	カッコ内の式を先に評価します．
単項負号 -	符号を反転します．例：-3.14
べき乗 ^	左辺の値を右辺値乗します．左辺が負の値の場合，答えは欠損値になります． 2^64 以上の値も欠損値になります（大きすぎて計算できません）． 例：2^3 ⇒ 8，0^0 ⇒ 1，(-2)^3 ⇒ 欠損値
乗算 *，除算 /， 剰余 %	左右両項のどちらかまたは両方が欠損値の場合，答えは欠損値です．除算と剰余は右項が0のばあいも欠損値を返します．（一般のプログラム言語と異なり，ゼロ除算例外は起こしません．）例：数値*数値 数値/数値 数値%数値
加算 +，減算 -	左右両項のどちらかが欠損値の場合，その値を0とみなして計算しますが，両項とも欠損値の場合，こたえは0ではなく欠損値になります． 例：数値+数値 数値-数値
切り捨て #	floor(左項/右項)*右項を返します．右項が0のときは欠損値を返します．左右両項のどちらかまたは両方が欠損値のときも，欠損値を返します． 例：32#10 ⇒ 30
書式 $	左項の数値を右項の文字列書式にしたがって文字列に変換します．返す値は文字列で，数値ではありません． 例：3.14159265358979323846264 33 $ '+#.##' ⇒ "+3.14"

- **文字列演算子**：現在のところ，これと後述する substr 関数（文字列切り出し）の2つだけです．

文字列結合 @	左項と右項の文字列を結合します． 例：'神' @ '戸市' ⇒ '神戸市'

- **比較演算子**：数値同士あるいは文字列同士の大小比較を行います．数値は，差が 0.000000000001 未満のときは等しいとみなします．文字列の大小比較はシフトJISコード順です．比較結果が真のときは0を返します．偽のときは欠損値を返します．1<a<=b<3 という書き方もできます．比較演算の連接は全ての比較結果の論理積を意味します．つまり，先の例は 1<a かつ a<=b かつ b<3 という意味です．比較演算子同士には優先順位はありません．左から順に評価します．

等しい	==
大なり	>
小なり	<
等しいか大なり	>= =>
等しいか小なり	<= =<
等しくない	!= <> ><

- **論理演算子**：引数が実値（欠損値以外）か，または"空文字列でない"ならば真，欠損値か空文字列なら偽として，論理演算を行います．優先順位は否定，積，和の順です．

単項否定 ！	例：！ 年齢 >= 20
論理積 &	例：年齢 < 20 & 性別 == 男
論理和 |	例：年齢 < 20 & 性別 == 男 | 年齢 >= 20 & 性別 != 男 ⇒ 未成年男性または成年非男性

論理積は，左項が真なら右項の値を，左項が偽なら欠損値を返します．
例：性別 & '性別入力済み' ⇒性別が'男'や'女'なら'性別入力済み'を，そうでなければ空を返します．
論理和は，左項が真なら左項の値を，左項が偽なら右項の値を返します．
例：性別 | '不明' ⇒性別が'男'や'女'ならその文字列を，性別が空文字列なら'不明'を返します．

- **条件演算子**:条件式の答えが真なら？の次の式，答えが偽なら：の次の式の答えを返します.

<条件式>？<真のときの式>：<偽のときの式>
例: 年齢 > 15 ？'内科診察' : '小児科診察'

- **関数**:

関数名の命名規則: 関数名は，left，drug，time などの語素を連結して作られます．関数名の語素解析の都合上，2つめ以降の語素は，それぞれの先頭の文字を大文字にしてください． 語素の間に空白は入れないでください.
例: leftSystolicVitalValue(…)← ○　　left systolic vital value(…)← ×
語素の順序はどうでも構いません． 例: valueVitalLeftSystolic(…)← ○
それぞれの語素は3文字までに省略することができます． 例: volume，volum，volu，vol は全てOKです.
したがって，lefSysVitVal などと書くこともできます.
引数について: 1番目の引数は記号やバイタルサイン，薬剤などの名前文字定数に限ります．1番目の引数に式を書くことはできません．2番目以降の引数は式です．たとえば remarkValue の記号名は定数ですが副記号名は式です.
ただし，age()，era()は1番目の引数にも式を書くことができます.

remarkValue(記号名文字列，副記号名式，序数式) あるいは **remarkValue(記号名文字列，序数式)**
remarkTime(記号名文字列，序数式)
記号が入力された時刻や記載コメント文字列を返します．開始と終了がある記号(気管挿管や手術など)では begin と end をつけて，beginRemarkValue や endRemarkTime などとします．記号名は'気管挿管'や'手術'などをシングルクォーテーションで括って書いてください.
remarkValue には副記号名を付けることができます．これは'気管挿管'に対して'ｻｲｽﾞ'や'深さ'などです.
副記号名を省略した場合は該当する記号につけられたコメント文全体を返します．返す文字列が数値として解釈できる場合(チューブ深さが 22cm のばあいには 22 が)，数値としても返されます．この数値は算術演算の対象になります．文字列だけの答えを算術演算の対象としたときには欠損値として扱います.
remarkTime は，記号が記載されている時刻を返します．時刻は先に「時刻の内部表現」で述べた絶対時刻です.
序数は，記号が複数回記載された場合の「何番目」を表す数です．例えば remarkTime('硬麻穿刺',1) は「1回目の硬麻穿刺時刻」を意味します．逆に -1 は「最後から数えて一回目」，つまり2回穿刺したばあい2回目，3回穿刺したばあいは3回目になります．たとえば paperChart でチャートに印刷する気管チューブの太さは「最後に挿管されたチューブの太さ」を使用しています．序数0は beginRemarkTime のばあいには「最も早く開始された…」を表し，endRemarkTime のばあいには「最も遅く終了した…」を表します.
手術時間として endRemarkTime('手術',0)-beginRemarkTime('手術',0) を使用するか，
(endRemarkTime('手術',1)-beginRemarkTime('手術',1))+(endRemarkTime('手術',2)-beginRemarkTime('手術',2))+(endRemarkTime('手術',3)-beginRemarkTime('手術',3))+(endRemarkTime('手術',4)-beginRemarkTime('手術',4))
を使用するかは利用者の判断によって計算式が書き換えられるべきです．手術が4回未満であった場合，存在しない手術の手術時間は欠損値になります．先に述べた(欠損値を0と扱う)加算の規則により存在する手術の手術時間のみが加算されます．まさか1麻酔記録中に手術の数が4回を越えることは無いでしょう.
開始終了のある記号に begin も end も付けなかったばあい，begin とみなします.
記号名も副記号名も後で述べる remcnf.txt の書き方で決まる名前です.
記号名として'logging'という言葉を使った場合，[metaChart]のデータ収集開始とデータ収集終了を意味します．つまり，logging は予約語です.
beginRemarkTime('logging',0)はデータ収集開始(モニタ開始)時刻，endRemarkTime('logging',0)はデータ収集終了(モニタ停止)時刻です．logging の序数は何を書いても結果は同じですが，序数を省略しないでください．モニタ途中で「モニタ停止」⇒「モニタ再開」しても endRemarkTime('logging',0)-beginRemarkTime('logging',0)の値は変わりません.

vitalValue(バイタルサイン名文字列，左端時刻式，右端時刻式)
vitalTime(バイタルサイン名文字列，左端時刻式，右端時刻式)
vitalUnit(バイタルサイン名文字列，左端時刻式，右端時刻式)
外部機器から自動的に収集されるバイタルサインの数値，時刻，単位をそれぞれ返します．バイタルサイン名は定数文字列です．'NBP'，'HR'，'SpO2'など，シングルクォーテーションで括って書いてください．左端時刻と右端時刻は，少なくとも，どちらか一方は絶対時刻を指定してください.
例えば leftSystolicVitalValue('NBP',beginRemarkTime('手術',0),300) は，「手術開始から300秒以内で最も早く(左側:left)測定された NBP の収縮期値」を返します．相対時刻には負の秒数を指定することもできます.
rightDiastolicVitalValue('NBP',-300,beginRemarkTime('logging',0)) は，退室(モニタ停止)前 300 秒以

内で最も遅く(right)測定された NBP の拡張期血圧を返します．実際の退室時血圧の計算式は calcnf.txt をご覧ください．

vitalUnit 関数は文字列(mmHg, kPa など)を返します．ハートモニタからのデータに単位が付いていないばあい，あるいは与えられた時刻区間内に測定が行われていない場合，この関数は空文字列を返します．

vitalValue 関数，vitalTime 関数には以下の修飾子を付けることができます．vitalUnit 関数では付けても無視します．

systolic 収縮期値を返します．測定値が収縮/拡張/平均の3つ組でない場合，欠損値を返します．

diastolic 拡張期値を返します．測定値が収縮/拡張/平均の3つ組でない場合，欠損値を返します．
diastolic も systolic も付けないと，平均値と解釈します．もちろん SpO_2 など測定値が3つ組でないものには，どちらもつける必要もありません．

minimum vitalValue では，与えられた時刻区間のバイタルサインの 20 秒毎の移動平均の最低値を返します．vitalTime では，その最低値が記録された時刻を返します．

maximum vitalValue では，与えられた時刻区間のバイタルサインの 20 秒毎の移動平均の最高値を返します．vitalTime では，その最高値が記録された時刻を返します．
maximum も minimum も指定されない vitalTime は常に欠損値を返します．

left vitalValue では，与えられた時刻区間内で最も早く(左側)測定された値を返します．vitalTime では，その値が記録された時刻を返します．当該区間内で一度も測定されなかった場合，欠損値を返します．

right vitalValue では，与えられた時刻区間内で最も遅く(右側)測定された値を返します．vitalTime では，その値が記録された時刻を返します．当該区間内で一度も測定されなかった場合，欠損値を返します．

注意：

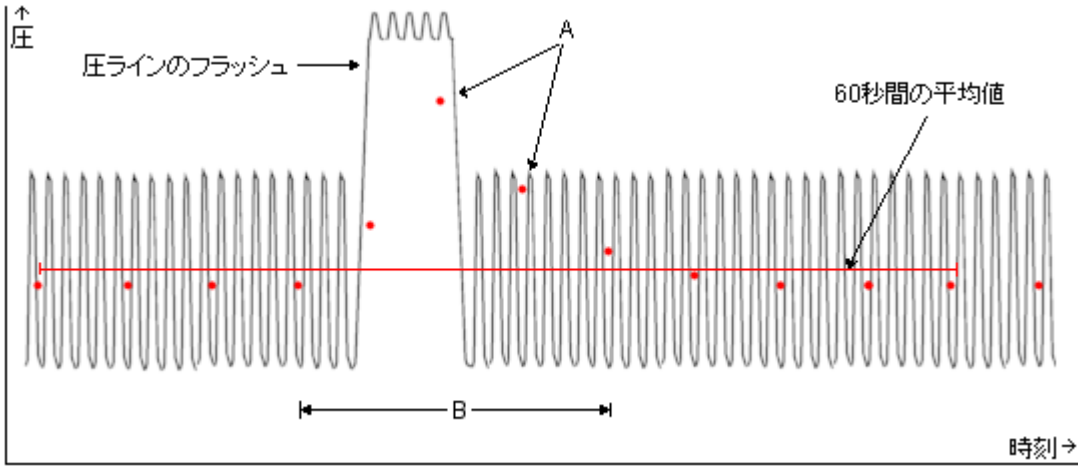
minimum, maximum, left, right のいずれも指定しないときは，与えられた時刻区間のバイタルサインの平均値を求めます．ただし，ここで言う平均値とは“平均血圧”の“平均”ではありません．たとえば systolicVitalValue と書くと，与えられた時刻区間内の時系列(たとえば5秒間隔)で得られた収縮期血圧値(複数個)の平均値を求めます．minimumSystolicVitalValue と書くと，時系列で得られた収縮期血圧値の中で最も低い数値という意味になります．当然のことですが，与えられた時刻区間内に1ヶ所しか測定値が存在しない場合(非観血血圧など)は，その値そのものを返します．与えられた区間に1度も測定が行われていなかった場合は，欠損値を返します．

NBP や bolusCO，血液ガスデータなどの測定間隔は，原則として不定期です．また HR や SpO_2 などの連続測定パラメータの測定間隔(ハートモニタからのデータ送出間隔)は，5秒から2分30秒程度までハートモニタの機種により様々です．

動脈圧フラッシュやゼロ点校正などのアーチファクトの除去について：

次の図の灰色線は観血圧波形，赤点は 5 秒ごとにハートモニタから送られてくる平均血圧値です．

データは 20 秒毎(データの受信タイミングにもよるが，おおよそ4点づつ)の区間平均を用います．“maximum”を指定したときは，この 20 秒毎の区間平均の最高値を，また“minimum”を指定すると 20 秒毎の区間平均の最低値を算出します．何も指定しない(平均値)ときは，単純に各赤点の平均値です．ただし，それぞれの区間の両側 20 秒間も加えた合計 60 秒間の平均値の上 30%または下 50%の範囲から逸脱した値(下図 A)を含む 20 秒区間(下図 B)のデータは計算から除外します．この方法は圧波形の鈍りによる不正確な計測値や加圧バッグ内圧の低下によるダラダラフラッシュの影響を除外することはできません．圧ラインフラッシュは高圧で短時間に行ってください．ゼロ点校正も 10～20 秒以内でお願いします．圧ラインを使用しないときはトランスデューサとハートモニタの間の電線をはずしてください．そうしないと「血圧ゼロ」という値が麻酔記録に持続的に送られてしまいます．



計算に要する時間について：

バイタルサインデータ(???.wna ファイル)を用いた計算には患者属性データ(???.txt ファイル)を用いた計算より時間がかかります． FM.exe の一覧表作成機能を用いて大量の症例(たとえば合計数千時間分)のバイタルサイン関連

の計算の行う場合には，計算自体にも数時間を要することがあります．計算に要する時間は CPU の計算速度とディスクのアクセス速度で決まります．一般的な目安を示すことはできません．頻繁に行う計算はあらかじめ calcnf.txt の output 節に記載しておいてください．そうすれば一覧表作成作業が早くなります．

mnemonicDosis（薬剤名文字列，左端時刻式，右端時刻式） または **mnemonicDosis（薬剤名文字列）**
mnemonicVolume（薬剤名文字列，左端時刻式，右端時刻式） または **mnemonicVolume（薬剤名文字列）**
codeDosis（薬剤コード文字列，左端時刻式，右端時刻式） または **codeDosis（薬剤コード文字列）**
codeVolume（薬剤コード文字列，左端時刻式，右端時刻式） または **codeVolume（薬剤コード文字列）**
drugGroupVolume（薬剤グループ名文字列，左端時刻式，右端時刻式）
または **drugGroupVolume（薬剤グループ名文字列）**
drugUnit（薬剤名文字列）
mnemonicDosis は薬剤の溶質の量（単位はμg，mg，mEq など）の合計を返します．
mnemonicVolume は薬剤の溶媒の量（単位は常にml）の合計を返します．
codeDosis と codeVolume は薬剤名称のかわりに JSA 麻酔台帳の薬剤コードで薬剤を指定します．
JSA 麻酔台帳薬剤コードは現在のところ数字ですが，将来サブ分類とか言って'0101-A'などとされても良いように，
本システムでは文字列として取り扱います．'0101'というふうにシングルクォーテーションで括って書いてください．
drugGroupVolume はそのグループに属する薬剤の合計量（単位は常に ml）を返します．
drugUnit はその薬剤の溶質の単位（μg，mg，mEq など）の文字列を返します．
左右両端の時刻はバイタルサインと同じですが，点滴投与の薬剤は左右両時刻の範囲内で開始量と終了量（あるいは途中〆）が記載されていなければなりません．点滴の場合，ポンプと異なり，投与時間による比例配分はしません．開始量あるいは終了量のどちらかが不明の場合，投与量は0mlとみなします．

demograph（患者情報項目名文字列）
患者情報項目名に対応する値（多くは文字列）を返します．患者情報項目とは NV.exe の画面の「記載…」ボタンで入力する項目です．返す文字列が数値として解釈できる場合，数値としても返されます．この数値は算術演算の対象になります．文字列だけの答えを算術演算の対象としたときには欠損値として扱います．
[metaChart]は関数名を頭3文字（dem だけ！）で識別しますので，正しい綴り(demography)でも通ります．ソフトを作るときの単なるミスで綴りの間違いを引きずっているだけです．（プログラム内部を全部直すのも面倒だし・・・）

age（年月日項目名文字列）
年月日項目を生年月日情報とみなして，その日付から現在（モニタ開始時）の年齢を計算し，年を単位とする数値として返します．1ヶ月は 1/12 歳です．1週は 1/52 歳です．

era(年月日項目名文字列，書式指定式)
年月日を書式指定で示される書式に変換します．
例：era('生年月日','yyyy-mm-dd')
⇒ 患者生年月日が元号で書かれていても西暦で書かれていても日本麻酔科学会の“JSA麻酔台帳”の年月日書式に変換します．

age( ) と era( ) で読み取ることのできる年月日書式は次のとおりです．これは前述の“書くほうの書式”とは異なります．
数字は全て半角で．
19950507
1995 5 7
1995/05/07
1995/5/7
5/7/1995 <-5月7日です．7月5日（ヨーロッパ風）は使えません．
1995 年 5 月 7 日
1995 年 05 月 07 日
あるいは元号記号に続けて，年数2桁以内で上記の書式． H7/5/7 平成 070507 など…
元号記号なしの2桁年（2桁西暦）は認識しません．JAN,FEB,MAR… 等も使えません．
年と月，月と日の間の区切り記号は数字以外なら何でも可．“/”“,”“.”空白“年”“ねん”など….
区切り記号なしの場合は必ず，元号つきの年は2桁，西暦年は4桁，月と日は2桁です．
年齢計算はこの生年月日からモニタ開始までの年数を実数で返します．つまり6ヶ月は 0.5 年，1 日は約 0.00274 年という風に計算します．ちなみにうるう年の 1 日は 0.002732 年ですが，誤差は無視しています．
元号記号は必要に応じて calcnf.txt の gengou 節に追加してください．

substr(区切り文字列, 序数, 全体文字列)
全体文字列を区切り文字列ごとに切り分けた時の序数番目の部分文字列を返します. ただし序数は 0 から始まります.
たとえば, substr(' ', 0, 'one two three four five six') は“one”を返し,
たとえば, substr(' ', 4, 'one two three four five six') は“five”を返します.
区切り文字列(上の例では空白文字)自体は答えに含まれません. 区切り文字列は複数文字でもかまいません.
たとえば, substr('e ', 2, 'one two three four five six') は“four fiv”を返します. (eと空白が区切り)
しかし, substr(' ', 6, 'one two three four five six') の答えは欠損値です. (空白が5つしかないので)
さらに, 全体文字列中に区切り文字列が無いときも, 答えは欠損値になります.
たとえば, substr('xyz', 0,'one two three four five six') の答えは欠損値です.
区切り文字列は必ず文字定数にしてください. 区切り文字列として式や式の名前(後述)を書くことはできません.

pageValue( )
現在, 画面表示あるいは印刷しようとしている麻酔記録のページ数(第？ページ)を現在の画面表示あるいは印刷の時間幅で計算して返します. 引数はありません. 最初が第1ページです.

pageTime( )
現在, 画面表示あるいは印刷しようとしている麻酔記録のページの左端時刻を返します. 引数はありません.

- **名前 = 式 ;**

等号の右辺を計算した結果の値に, 左辺の名前をつけます. ただしこれは束縛(binding)であって代入(assignment)ではありませんので, **この値 = この値 + 1;** というように, 自分自身を式の一部として参照することはできません. めぐりめぐって自分自身を参照しているような場合も, NV.exe 起動時にエラーを表示して終了します.