



MacBug

Command Reference

1998 Presented By KaZuhiro FuRuhata

- 目 次 -

MacBug について.....	4
FutureBASIC から MacBug を呼び出す方法.....	5
MacBug 編集キー一覧.....	6
MacBug コマンド一覧.....	7
MacBug コマンドリファレンス.....	8

API	8	EXT	15
ATB.....	8	F	16
ATC.....	8	FB.....	16
ATD	8	File	16
ATHC.....	8	FILL	16
ATP	9	FL	17
ATR.....	9	FP	17
ATSS.....	9	FSInfo	17
ATT.....	9	FW.....	17
BR	10	G.....	18
BRC.....	10	Gestalt	18
BRD	10	GT	18
BRM.....	10	GTP	18
BRP	10	HC	19
CS.....	11	HD	19
DB	12	HELP.....	19
DCMD.....	12	HOW	19
DH.....	12	HS	19
DHP.....	12	HT	19
dizy.....	12	HX	20
DL	12	HZ	20
DM	13	ID	21
DMA	13	IDP	21
DP	13	IL	21
Drive	13	ILP.....	21
Drvr	14	IP	21
DSC.....	14	IPP.....	21
DSCX	14	IR.....	22
DV.....	14	IRP	22
DW	14	JumpTable	22
DX.....	14	Leaks	23
EA	15	Lock	23
Echo	15	LOG	23
Error	15	MBUG.....	24
ES.....	15	MC	24

MCC.....	24	SO.....	28
MCD.....	24	SS	28
MList.....	24	ssc.....	29
MR	24	STAT.....	29
Patch.....	25	STOPIF	29
Printf.....	25	StopXPP	29
ProcInfo.....	25	SW.....	29
RAD	26	SWAP.....	29
RB	26	SX.....	29
RD	26	SysTop	30
Registers.....	26	SysSwell.....	30
RN	26	TD	31
RS.....	26	TF	31
S	27	thing	31
SB.....	27	TM.....	31
SC6.....	27	TMP	31
SC7.....	27	UNLOCK	32
SET.....	27	VBL.....	32
SHOW	28	Vol.....	32
SL	28	WH	32
SM.....	28	Where	32

MacsBug Tips	33
MacsBugのエラーメッセージ.....	35

MacsBug について

MacsBugは「デバッガー」と呼ばれるもので、プログラムをマシン語レベルでデバッグできるものです。他にはTMONというデバッガーもあります。

MacsBugはマシン語レベルでのデバッグ作業になるのですが、とりあえず使い方だけでも知っていれば、プログラムミスを見出すのに役に立つでしょう。多くの場合、プログラムを作成して正しく動くはずだが、なぜか正常に動作しない場合に利用すると良いでしょう。

元々スバゲッティプログラムになっているものや、アルゴリズム的ミスがあるものに関しては、どう頑張ってもMacsBugでは解決できません。そのような場合はアルゴリズムを見直す方が先決です。

MacsBugの前にFuture BASICのDebug IIというラインレベルデバッガーを利用するのも良い方法です。こちらはFuture BASIC IIの命令を1ステップ実行しながらチェックすることができますのでMacsBug上で調べるよりもはるかに手軽です。

MacsBugもシステムのバージョンアップ等に伴いバージョンアップが行われていますのでOSバージョンに対応したものをに入れておく必要があります。MacsBugをインストールするのは簡単でMacsBugをシステムフォルダに入れるだけです。

MacsBugについてはMacsBug Referenceという本（日本語版）が出ていますので、金銭的に余裕がある人は購入するのも良いかもしれません（古いバージョンですのでPowerPC関係についてのコマンドや情報はありません）。書籍コードはISBN4-7952-9622-7、金額は9800円となっています。はっきり言って高すぎる金額ですが・・・（内容からすれば1500～1800円程度のもので）Future BASIC II-Jのマニュアルを見ると、もう1冊出ているようですが手元にありませんし翻訳されているかどうか不明です。

MacsBugの簡単な使い方や説明についてはFuture BASIC II-Jマニュアルの382ページ以降に記述されていますので、一読しておくとう良いでしょう。

注意： 私自身、MacsBugについて詳しくありませんので、文中に間違いがあるかもしれません。また不明部分に関しては、記述していない部分もあります。以上、2点あらかじめご了承下さい。

FutureBASICから MacxBug を呼び出す方法

Future BASIC上からMacxBugを利用する事ができ、専用の命令が2つ用意されています。

(1) CALL DEBUGGER

単純にMacxBugを呼び出します。MacxBugからはg[return]と入力する事で復帰できます。

(2) CALL DEBUGSTR(文字列)

指定した文字列をMacxBug側に表示します。これを利用すると、どこで呼び出されたかを簡単に確認する事が出来ます。MacxBugからはg[return]と入力する事で復帰できます。

MacxBugでFuture BASICの関数名を表示させることができます。これを利用するとさらにデバッグ効率が良いになります。関数名を表示させるにはコンパイルオプションを以下のように指定します。

```
COMPILE 0,_MacxBugLabels
```

MacBug 編集キ一ー一覽

	カーソルを左に1つ移動
	カーソルを右に1つ移動
esc	アプリケーション画面とMacBugの画面をトグル切り替え
return	最後の命令の繰り返し
Command + V	コマンドバッファから1つ下のコマンドをコピー
Command + B	コマンドバッファから1つ上のコマンドをコピー
Option +	1つ左の単語へ移動
Option +	1つ右の単語へ移動
Option + Delete	カーソルの左側の単語を消去
Command +	コマンドラインの先頭に移動
Command +	コマンドラインの末尾に移動
Command + Delete	カーソルの左側を消去
Command + D または ;	シンボル表示
Command + E	外部シンボル表示
Command + R	リソース情報シンボルを表示

MacBug コマンド一覧

API	G	SHOW
ATB	Gestalt	SL
ATC	GT	SM
ATD	GTP	SO
ATHC	HC	SS
ATP	HD	ssc
ATR	HELP	STAT
ATSS	HOW	STOPIF
ATT	HS	StopXPP
BR	HT	SW
BRC	HX	SWAP
BRD	HZ	SX
BRM	ID	SysTop
BRP	IDP	SysSwell
CS	IL	TD
DB	ILP	TF
DCMD	IP	thing
DH	IPP	TM
DHP	IR	TMP
dizy	IRP	UNLOCK
DL	JumpTable	VBL
DM	Leaks	Vol
DMA	Lock	WH
DP	LOG	Where
Drive	MBUG	
Drvr	MC	
DSC	MCC	
DSCX	MCD	
DV	MList	
DW	MR	
DX	Patch	
EA	Printf	
Echo	ProclInfo	
Error	RAD	
ES	RB	
EXT	RD	
F	Registers	
FB	RN	
File	RS	
FILL	S	
FL	SB	
FP	SC6	
FSInfo	SC7	
FW	SET	

MacBug コマンドリファレンス

API 【Displays information on a specified API routine】

API (アプリケーションプログラムインターフェース) リストを表示します。

```
API [-s] [-a] [trap [trap|count]]
S:   シングルトラップ表示
A:   オプション (アドレス) 表示
trap:   トラップ名
```

ATB 【A-trap Break】

Aトラップ発生時にMacBugを起動します。

```
ATB [A][trap[trap]][n|expr][';cmd[:cmd]...']
A:   Aトラップがアプリケーションヒープから呼び出されたときのみ起動
trap:   Aトラップの番号を指定
n:   Aトラップが指定回数発生したときに起動
expr:   指定トラップが発生しexprがTRUEの時に起動
cmd:   起動後に実行するコマンド名
```

ATC 【A-trap Clear】

Aトラップ関連コマンド (ATB,ATT,ATHC,ATSS) の設定をクリアします。

```
ATC [trap[trap]]
trap:   Aトラップの番号を指定
```

ATD 【A-trap Display】

Aトラップ関連コマンド (ATB,ATT,ATHC,ATSS) の設定を表示します。

ATHC 【A-trap Heap Check】

指定したAトラップ実行前にヒープをチェックします。ヒープが正常でない場合はエラーメッセージを表示します。

```
ATHC [A][trap[trap]][n|expr]
A:   Aトラップがアプリケーションヒープから呼び出されたときのみ起動
trap:   Aトラップの番号を指定
n:   Aトラップが指定回数発生したときに起動
expr:   指定トラップが発生しexprがTRUEの時に起動
```

ATP 【A-trap Playback】

記録されたトラップの情報を表示します。記録されていない場合はATPコマンドで得た情報を表示します。

ATR 【A-trap Record】

Aトラップの記録を行うか、行わないかを設定します。

ATR [A][ON|OFF]

ATSS 【A-trap Step Spy】

指定範囲のメモリまたは指定アドレスのロングワード(32-bit)のチェックサム(合計)を調べ、不一致の場合にMacrosBugを起動します。

ATSS [A][trap[trap]][n|expr],address1[address2]

A: Aトラップがアプリケーションヒープから呼び出される前にチェックサムを計算
 trap: Aトラップの番号を指定
 n: Aトラップが指定回数発生したときにチェックサムを計算
 expr: 指定トラップが発生しexprがTRUEの時に起動
 address1: address1にあるロングワードのチェックサムを計算。address2を指定した場合はaddress1からaddress2までの範囲のチェックサムを計算

ATT 【A-trap Trace】

Aトラップが発生した場合、プログラムを停止せずにMacrosBugのバッファに書き込みを行います。

ATT [A][trap[trap]][n|expr]

A: Aトラップがアプリケーションヒープから呼び出されたときのみ書き込み
 trap: Aトラップの番号を指定
 n: Aトラップが指定回数発生したときに書き込み
 expr: 指定トラップが発生しexprがTRUEの時に書き込み

BR 【Breakpoint】

ブレークポイントを設定します。

BR address[n|expr][';cmd[:cmd]...']
address: ブレークポイント設定アドレスを指定
n: 指定回数実行後ブレーク
expr: exprがTRUEの時にブレーク
cmd: ブレーク後に実行するコマンド名

BRC 【Breakpoint Clear】

ブレークポイントを解除します。

BRC [address]
address: 指定アドレスのブレークポイントを解除します。アドレスを省略した場合は、全てのブレークポイントを解除します。

BRD 【Breakpoint Display】

ブレークポイントが設定されているアドレスを表示します。

BRM 【Multiple Breakpoints】

名前を指定してブレークポイントを設定します。

BRM name
name: nameを含むルーチンの先頭にブレークポイントを設定します。

BRP 【Breakpoint for PowerPC】

PowerPCコードにブレークポイントを設定します。

BRP address[n|expr][';cmd[:cmd]...']
address: ブレークポイント設定アドレスを指定
n: 指定回数実行後ブレーク
expr: exprがTRUEの時にブレーク
cmd: ブレーク後に実行するコマンド名

CS【Checksum】

チェックサムを計算します。

CS [address1[address2]]

address1: 指定アドレスのチェックサムを計算。address2まで指定した場合は、address1からaddress2までのチェックサムを計算。

DB 【Display Byte】

指定アドレスの1バイトを表示します。

DB [address]

address: 1バイト表示したいアドレス

DCMD 【Show DCMD】

インストールされているDCMDs (modular debugger commands)名を表示します。

DCMD [name]

name: 表示する名前

DH 【Disassemble Hexadecimal】

指定した16進数をアセンブラのニーモニックに変換します。

DH expr

expr: 16進数値

DHP 【Disassemble Hexadecimal of PowerPC】

指定した16進数をPowerPCアセンブラのニーモニックに変換します。

DH expr

expr: 16進数値

dizy 【Check discipline】

disciplineがインストールされているかどうか調べます。

DL 【Display Long】

指定したアドレスのロングワード(32ビット)値を表示します。

DL [address]

address: 表示したいアドレス

DM 【Display Memory】

指定したアドレスから始まるメモリ内容を表示します。

DM [address][bytes][template|type]

address: 表示する先頭アドレス

bytes: 表示するバイト数

temp late:表示に使用するテンプレート

type: 表示に使用するメモリ表示形式。使用できる形式は以下のとおり。

Byte: 1バイト

Word: 2バイト (1ワード)

Long: 4バイト (2ワード)

SignedByte: 符号付1バイト

SignedWord: 符号付2バイト (1ワード)

SignedLong: 符号付4バイト (2ワード)

UnsignedByte: 符号なし1バイト

UnsignedWord: 符号なし2バイト (1ワード)

UnsignedLong: 符号なし4バイト (2ワード)

Boolean: TRUEまたはFALSE

pString: パスカル文字列を表示

cString: C文字列を表示

DM 【Display Memory】

指定したアドレスから始まるメモリ内容をアスキー文字で表示します。

DM [address][bytes]

address: 表示する先頭アドレス

bytes: 表示するバイト数

DP 【Display Page】

指定アドレスから128バイト分メモリ内容を表示します。

DP [address]

address: 表示する先頭アドレス

Drive 【Display info on drives】

ドライブ情報を表示します。

Drive [drvNum|dRefNum]

drvNum: ドライブ番号

Drv [Display Driver]

ドライバーの情報を表示します。

Drv [refNum:inum]

refNum: 参照番号

DSC [Discipline]

Disciplineユーティリティのオン、オフを切り換えます。

DSC [A][X][ON|OFF]

A: Aトラップがアプリケーションヒープから呼び出されたときのみ起動

X: エラー情報を内部に保存

ON: Disciplineオン

OFF: Disciplineオフ

DSCX [Discipline]

Disciplineによるブレークオン、オフを切り換えます。

DSCX [ON|OFF]

ON: Disciplineブレークオン

OFF: Disciplineブレークオフ

DV [Display Version]

MacBugのバージョンを表示します。

DW [Display Word]

指定したアドレスのワード（2バイト）を表示します。

DW [address]

address: 値を表示するアドレス

DX [Debugger Exchange]

ユーザーブレークの切替を行います。

DX [ON|OFF]

ON: ユーザーブレーク有効

OFF: ユーザーブレーク禁止

EA 【Exit to Application】

MacBugを起動したアプリケーションに処理を戻します。

Echo 【Echo the Command】

表示画面にコマンドラインから入力した文字を反映させます。

```
Echo [param]
param:   エコーバックさせるコマンド
```

Error 【Display Error Message】

エラー番号に対応したエラーメッセージを表示します。

```
Error expr
expr:     エラー番号
```

ES 【Exit to Shell】

ファインダーに処理を移します。

EXT 【External symbols for Resource file】

シンボリックファイルからリソースシンボルを読み込みます。

```
EXT [expr [pathname]]
expr:   参照番号
pathname: パス名
```

F 【Find】

指定バイト値を検索します。

Find [B|W|L|P] address bytes exprstring
B: 1 バイト
W: 2 バイト (ワード)
L: 4 バイト (ロングワード)
P: 下位 3 バイト
address: 検索開始アドレス
byte: 検索バイト数
expr: 検索値
string: 文字列

FB 【Find Byte】

指定アドレスから 1 バイト検索します。

FB address byte expr
address: 検索開始アドレス
byte: 検索バイト数
expr: 値

File 【Display file information】

ファイル情報を表示します。

File [fRefNum | "file name" | -t "type"]
frefNum: ファイル参照番号
filename: ファイル名
-t: ファイルタイプ

FILL 【Fill memory】

指定アドレスから指定バイト数だけ指定値で埋めます。

FILL address byte expr
address: 開始アドレス
byte: 埋めるバイト数
expr: 値

FL 【Find Long word】

指定アドレスから 4 バイト (ロングワード) 検索します。

FL address byte expr

address: 検索開始アドレス

byte: 検索バイト数

expr: 値

FP 【Find from Pointer】

ポインタが示すアドレスから 1 バイト検索します。

FP address byte expr

address: 検索開始アドレス

byte: 検索バイト数

expr: 値

FSInfo

ファイルシステム情報を表示します。

FSInfo [fsid | -t]

fsid: ファイルシステムID

-t: 簡略表示

FW 【Find Word】

指定アドレスから 2 バイト (1 ワード) 検索します。

FW address byte expr

address: 検索開始アドレス

byte: 検索バイト数

expr: 値

G【Go】

プログラムを実行します。

G [address]

address: 指定アドレスからプログラムを実行します。省略時は現在のプログラムカウンタの示す位置から実行します。

Gestalt

Macintoshの各種情報をリスト表示します。

Gestalt [-n] [[-s] 'xxxx']

-n: インストールされている部分のみ表示

-s 'xxxx': 指定したもののみ表示

GT【Go Till】

プログラムカウンタが指定アドレスに来るまで実行します。

GT address[:cmd[:cmd]...]

address: 停止アドレス

cmd: 起動後に実行するコマンド名

GTP【Go Till for PowerPC】

プログラムカウンタが指定アドレスに来るまで実行します。PowerPC専用です。

GTP address[:cmd[:cmd]...]

address: 停止アドレス

cmd: 起動後に実行するコマンド名

HC【Heap Check】

ヒープブロックが破壊されているかどうか調べます。

HD【Heap Display】

ヒープブロック情報を表示します。

HD [qualifier]

qualifier: 表示するブロックの種類を指定します。指定できる種類は以下の通りです。

- F: フリーブロック
- N: ノンリロケートブル(再配置不可)ブロック
- R: リロケートブルブロック(再配置可能)ブロック
- L: ロックされているブロック
- P: パージ可能ブロック
- Q: 破壊されている可能性のあるブロック
- RS: リソースブロック
- type:指定タイプのリソースブロック

HELPまたは?【Help】

コマンド情報を表示します。

HELP [cmd|topic]

cmd: コマンド名

topic: 項目名

HOW【Display Break Message】

MacBug起動後に表示されたブレイクメッセージを表示します。

HS【Heap Scramble】

ヒープ内の再配置可能ブロックをAトラップ毎に移動させます。

HS [address]

address: 移動させたいヒープの先頭アドレスを指定。省略時はアプリケーションヒープ。

HT【Heap Totals】

現在のヒープ情報を表示します。

HX【Heap Exchange】

現在のヒープを選択します。

HX [address]

address: ヒープゾーンのアドレスを指定。省略時はヒープを順番に切り替えていきます。

HZ【Heap Zone】

全てのヒープゾーンを一覧表示します。

HZ [address]

address: ヒープの開始アドレス

ID 【Disassemble One Line】

指定アドレスから 1 行逆アセンブル表示を行います。

```
ID [address]
address: 逆アセンブルするアドレス
```

IDP 【Disassemble PowerPC code One Line】

指定アドレスから 1 行分、PowerPCコードとして逆アセンブル表示を行います。

```
IDP [address]
address: 逆アセンブルするアドレス
```

IL 【Disassemble From Address】

指定したアドレスから任意のバイト数だけ逆アセンブルします。

```
IL [address[line]]
address: 逆アセンブル開始行。省略時は現在のプログラムカウンタ
line:    表示する行数 (16進数)
```

ILP 【Disassemble PowerPC code From Address】

指定したアドレスから任意のバイト数だけPowerPCコードとして逆アセンブルします。

```
ILP [address[line]]
address: 逆アセンブル開始行。省略時は現在のプログラムカウンタ
line:    表示する行数 (16進数)
```

IP 【Disassemble Around Address】

指定アドレスから64バイト逆アセンブルします。

```
IP [address]
address: 逆アセンブル開始行を指定。省略時は現在のプログラムカウンタ
```

IPP 【Disassemble Around Address】

指定アドレスから64バイト分、powerPCコードとして逆アセンブルします。

```
IPP [address]
address: 逆アセンブル開始行を指定。省略時は現在のプログラムカウンタ
```

IR 【Disassemble Until End of Procedure】

ルーチンの最後まで（関数処理の最後まで）逆アセンブルします。

IR [address]

address: 逆アセンブル開始行を指定。省略時は現在のプログラムカウンタ

IRP 【Disassemble PowerPC code Until End of Procedure】

ルーチンの最後まで（関数処理の最後まで）PowerPCコードとして逆アセンブルします。

IRP [address]

address: 逆アセンブル開始行を指定。省略時は現在のプログラムカウンタ

JumpTable 【Display Jump Table】

ジャンプテーブル内容を表示します。

Jumtable [expr]

expr: 値

Leaks

スタックがメモリリークを起こしているかどうかチェックします。

Leaks [On|Off|Dump]

On: オン

Off: オフ

Dump: メモリダンプ表示

Lock 【Lock Virtual Memory】

仮想記憶に使用しているメモリをロックします。

LOCK address count

address: ロックするアドレス

count: バイト数

LOG 【Log to a Printer or File】

MacBugの出力をファイルまたはプリンタ（シリアル接続）に送ります。

LOG [pathname|printer]

pathname: ファイル名。パスを指定する事も可能。

printer: プリンタに出力

MBUG 【Display MacsBug Global List】

MacsBugが使用しているグローバルリストを表示します。

MC 【Macro Create】

マクロを定義します。

```
MC name 'expr'expr  
name:   マクロ名  
expr:   展開式
```

MCC 【Macro Clear】

定義したマクロを消去します。

```
MCC [name]  
name:   消去するマクロ名。省略時は定義したマクロ全て。
```

MCD 【Macro Display】

定義されたマクロを表示します。

```
MCD [name]  
name:   マクロ名を指定。省略時は全てのマクロ名。
```

MList

インストールされているメニューリストを表示します。

MR 【Magic Return】

ステップオーバーすべきルーチンに入ってしまった場合、次のAトラップ呼び出し直後まで処理を行います。

```
MR [param]  
param:   リターンアドレスオフセット
```

Patch

パッチリストを表示します。パラメーターがない場合は全てのパッチリストを表示します。

Patch [I|O|T|S|P]

I: 割り込みベクタ
O: OSトラップベクタ
T: TBトラップベクタ
S: 現在のベクタを保存
P: 全てのベクタを印刷

Printf

指定した書式で式を表示します。

Printf "string" arg[arg]...

string: 文字列。文字列中に以下の書式指定子が使えます。

%d: 10進整数
%o: 8進整数
%x: 16進整数
%u: 符号無し10進定数
%c: 1文字
%s: 文字列

ProclInfo

起動しているアプリケーションを表示します。

RAD 【Toggle Register Name Syntax】

アドレスレジスタとデータレジスタの指定方法を切り替えます。

RB 【Reboot】

再起動します。

RD 【Display resource information】

リソース情報を表示します。

```
RD [-c] [-s] [-o] [-f ref#] [-i id] [[-t] 'xxxx'] [-h hndl]
-c:          リソースチェーン表示
-s:          リソースチェーン表示 (フォントリソースは除く)
-o:          リソースのオフセットを表示
-f <refNum>: ファイルのリソースを表示
-i <resID>:   指定リソースのIDのリソースを表示
-t 'xxxx':   指定タイプのリソースのみ表示
-h <handle>: リソースハンドル表示
```

Registers 【Register】

レジスタに直接値を代入またはレジスタの内容を表示します。

```
Registers [=expr:=xpr]
expr:      式。省略時は現在のレジスタ値を表示。
```

RN 【Set Reference Number】

シンボル参照を指定ファイルに限定します。

```
RN [expr]
expr:      ファイル参照番号(16進数)
```

RS 【Restart】

再起動します。

S【Step】

指定した数だけステップ実行します。

S [n|expr]

n: 実行する命令数を指定

expr: 指定条件

SB【Set Byte】

指定アドレスにバイト値を代入します。

SB address value [value]...

address: アドレス

value: 代入する値。文字列は引用符で囲む。

SC6【Stack Crawl(A6)】

スタックフレームの内容を表示します。

SC6 [address [byte]]

address: 現在のフレームアドレスを指定。省略時はA6レジスタ値。

byte: 範囲の上限を指定。省略時はCurStackBase^を使用。

SC7【Stack Crawl(A7)】

リターンアドレスが格納されているスタックアドレスを表示します。

SC7 [address[byte]]

address: 現在のフレームアドレス。省略時はA7レジスタ値。

byte: 範囲の上限を指定。省略時はCurStackBase^を使用。

SET

MacBugの環境を設定します。

SET [option[ON|OFF]]

option: 以下のオプションが指定できます。

SCROLLPROMPT

SUSPENDPROMPT

DITTO

AUTOGP

SIMPLIFIED

SHOW 【Show】

ステータス領域に指定形式で情報を表示します。

SHOW [address|'address'] [L|W|A|LA]

address: 表示するメモリの先頭アドレス

L: ロングワード形式で表示

W: ワード形式で表示

A: アスキー形式で表示

LA: ロングワードとアスキー形式で表示

SL 【Set Long】

指定アドレスにロングワード値を代入します。

SL address value [value]...

address: アドレス

value: 代入する値。文字列は引用符で囲む。

SM 【Set Memory】

指定アドレスに値を代入します。値の大きさによって書き込まれるサイズが変化します。

SL address value [value]...

address: アドレス

value: 代入する値。文字列は引用符で囲む。

SO 【Step Over】

ステップオーバー実行します。

SO iT[n]expr]

n: ステップ実行する命令数

expr: 条件指定

SS 【Step Spy】

命令実行前に指定アドレス、または指定アドレス範囲のチェックサムを求めます。

SS address1[address2]

address1: チェックサム計算する先頭アドレス。address2を指定した場合は、address1からaddress2までの範囲を計算。

SCC

スタックフレームの内容を表示します。

```
scc [address]  
address: スタックフレームアドレス
```

STAT

MacsBugを起動した環境を表示します。

STOPIF

式の値がTRUEになった場合、コマンド処理を行います。

```
STOPIF expr  
expr: 式
```

StopXpp

オープンされているXPPセッションをクローズします。

SW 【Set Word】

指定アドレスにワード値を代入します。

```
SW address value [value]..  
address: アドレス  
value: 代入する値。文字列は引用符で囲む。
```

SWAP 【Swap】

アプリケーションとMacsBug間の画面切替頻度を設定します。

SX 【Symbol Exchange】

アドレスにシンボル使用を許可するかしないかを設定します。

```
SX [ON|OFF]  
ON: シンボル使用を許可  
OFF: シンボル使用を許可しない
```

SysTop 【Install handle】

システムヒープの先頭にロックされたハンドル（4バイト）を追加、削除します。

SysSwell 【Specified System heap size】

システムヒープのサイズを設定します。

SysSwell [bufferSize]

bufferSize: ヒープサイズ（バイト数）

TD 【Display CPU Register】

全てのレジスタ内容を表示します。

TF 【Total Floating-Point Register Display】

全てのFPU レジスタ値を表示します。

thing

情報を表示します。

```
thing ["thingType"]  
thingType:
```

TM 【Total MMU Display】

MMU(メモリマッピングユニット) レジスタ値を表示します。

TMP 【List Templates】

テンプレートリストを表示します。

```
TMP [name]  
name:   テンプレート名
```

UNLOCK 【Unlock Virtual Memory】

仮想記憶に使用しているメモリをアンロックします。

UNLOCK address count

address: アンロックするアドレス

count: バイト数

VBL 【Display VBL task】

VBLタスクを表示します。

Vol 【Display Volume Information】

ボリューム情報を表示します。

WH 【Where】

指定したトラップ、シンボル、アドレスロケーションの情報を表示します。

WH [address!trap]

address: アドレスにある命令のロケーション情報を表示

trap: トラップ番号を指定。トラップのロケーション情報を表示。

Where [address!trap]

アドレスまたはトラップ情報を表示します。

address: アドレス

trap: トラップ番号

MacBug Tips

[1] アプリケーション画面と MacsBug 画面を素早く切り替える

escキーを押すと画面が交互に切り替わります。

[2] 複数コマンドを指定する

(セミコロン) で区切れば複数命令を記述指定できます。

[3] PC 表示領域の行数を変更する

MacsBugをResEditで開きます。mxbiリソースをダブルクリックして開きます。#of PC lines shown という項目の数値が行数ですので、これを変更し保存します。Macintoshを再起動すれば有効になります。

[4] 出力表示領域のバッファサイズを変更する

MacsBugをResEditで開きます。mxbiリソースをダブルクリックして開きます。#Size of history buffer という項目の数値が行数ですので、これを変更し保存します。Macintoshを再起動すれば有効になります。

[5] FKEY (Command+Shift+ 番号)でMacsBug を起動する

SystemファイルをResEditで開きます。(要注意) FKEYリソースを開きます。ResourceメニューからCreate New Resourceを選択し新しいFKEYリソースを作成します。A9FF 4E75と入力してファイルを保存します。Macintoshを再起動すれば有効になります。

[6] 簡単な計算を行う

先頭に#マークを入力後計算式を入力します。四則演算およびカッコが使用できます。10進数で指定する場合は数値の前に#マークを付加します。使用できる演算式を以下に示します。

- + 加算
- 減算
- * 乗算
- / 除算
- MOD 剰余
- @ アドレス参照。@20の場合アドレス\$0020の値を参照する。
- ^ アドレス参照。20^の場合アドレス\$0020の値を参照する。
- ANDまたは& 論理積
- ORまたは! 論理和
- NOTまたは! ビット反転
- XOR 排他的論理和
- << 左シフト
- >> 右シフト
- =または== 等しい
- <>または!= 等しくない
- a < b aはbより小さい
- a > b aはbより大きい
- a <= b aはbより等しいか小さい
- a >= b aはbより等しいか大きい

MacBugのエラーメッセージ

64-bit registers not allowed in expressions

値が64ビット割り当てではない

68881 not installed

コプロセッサ68881がインストールされていない

A6 does not point to a stack frame

A6レジスタがスタックフレームを示していない

Address must be even

アドレスが奇数

Address range must be entered before comparisons

チェックサムアドレスが指定されていない

All step point cleared

全てのステップポイントを消去した

Bad stack:stack pointer must be even and \leq stack base

スタックポインタの値が異常

bkLim does not agree with heap length

ヒープブロック調査後のアドレスが異常

Block length is bad

ブロック長が異常

Count must be greater than zero

カウンタが0以下

Damaged stack: A7 must be even and \leq CurStackBase

スタックポインタが奇数アドレス

Divide zero error

ゼロ除算を行った

Entry will not fit in the table

エントリーテーブルが満杯

Expression evaluation caused data read fault

数式の評価に失敗

Floating point not allowed in expressions

式に浮動小数点は使用できない

Free bytes in heap do not match zone header

フリーブロックサイズの合計が異常

Free master pointer list is bad

マスターポインタリストが異常

Low address must be less than or equal to high address

(Lowメモリ) アドレス範囲が異常

Macro expansion exceeds maximum command line length

マクロがコマンド入力可能文字数を超えた

MacBug code has been changed

MacBug破壊

MacBug stack overflow

MacBugのスタックがオーバーフローした

Master pointer does not point at a block

マスタポインタがヒープブロックを示していない

MMU not installed

MMUがインストールされていない

No blocks of this type found

指定ブロックが見つからない

Nonrelocatable block: Pointer to zone is bad

再配置不可能なブロック(ノンリロケータブルブロック)のブロックヘッダが異常

PC is not inside a procedure

プログラムカウンタシンボル情報が見つからない

Relative handle is bad

リロケータブルブロックのハンドルが異常

Start of link chain does not point to a stack frame

スタックのリンクチェーンがスタックフレームにない

Syntax error

構文エラー。引数不足。コマンド名入力ミス。

Template cannot expand more than 8 levels

テンプレートのネスティング限界(8レベル)を超えた

The address on the stack is not a return address

リターンアドレスがスタック上に見つからない



MacBug Command Reference

1998 Presented By KaZuhiro FuRuhata / OpenSpace