Coal

操作マニュアル

第　２．２　版

２０２０年１１月１９日

改　訂　履　歴

Ｖｅｒ	改訂年月日	改訂理由	作成者	承認者
1.7	2012.01.24	新規	小林	
1.8	2019.07.06	コマンドラインの実行時オプションで、「オプション番号＝値」の形式をサポートした。	小林	
1.9	2020.03.14	ログの設定でログ種別に、libakx.a のログを設定する文字 x を追加した。	小林	
2.0	2020.03.29	スクリプト名に直接スクリプトの内容を指定する場合に、中括弧({})の外側の改行コード(¥n,¥r)が無視される旨を追記した。	小林	
2.1	2020.10.20	動作環境と文字コードについての説明を追加した。	小林	
	2020.10.21	coal.ctl の説明を追加した。	小林	
2.2	2020.11.19	実行時オプション値の指定方法に、d.d.d.d 形式を追加した。	小林	

目　　次

1. 機能 1
2. コマンドライン 1
3. スクリプトの終了 2
4. 動作環境の設定 3
4.1. 環境変数 3
4.2. 設定ファイル 3
4.2.1. akb.ini 3
4.2.2. dscmndscript 6
4.2.3. coal.ctl 6
5. ログの設定 8
6. デバッグ・オプション 9
7. 動作環境と文字コード 12
7.1. Cygwin 環境 12

1. 機能

Ｃｏａｌ言語スクリプトを実行する。

2. コマンドライン

```
coal [-h|-?|-v|-c] [--{e|p|d|l}[LOG_PARM]] [-s] [-d[ ]DEBUG_OPTIONS] [-CMD_NO]
          [-S[ ]proc_no] [-P[ ]port_no] [-i] [-X] [-x[ ]extension] [-o[ ]EXEC_OPTIONS]
          [-r[out_file]] [-t[ ]max_thread] [-m[level][,[J|E|Other]]] [-D[ ]DEF_PARM]
          script_name [parm-1 parm-2 ...]
```

(1)オプション

(A) -h|-?

usage

(B) -v

バージョン出力。

(C) -c

Copyright 出力。

(D) -s

ログ出力レベルを０にする。初期値は、０。

(E) -CMD_NO

サーバとして起動し、スクリプト実行クライアントツール(mmisc)を使用する場合に指定する。

0x02:デーモンにする

0x04:-Pport_no で指定されたポートで接続要求を受け付け、チャネル０で通信チャネルをオープンする。(単一クライアントのみ接続可能)

0x08:上記で接続要求を受け付けたときに、チャネル０以外で通信チャネルをオープンする。(複数クライアントの接続が可能)

(F) -P port_no

接続要求を受け付けるポート番号またはサービス名。

(G) -S proc_no

プロセス番号を設定する。デフォルトは、２０１。

(H) -i

起動時にコマンド 0x0ffffff0 のスクリプトを実行する。

dscmdscript ファイルに、コマンドとスクリプト・ファイルの対応を設定する。

(I) --{e|p|d|l}[LOG_PARM]

ログ出力を指定する。(4.ログの設定を参照)

(J) -X

拡張子処理を行わず、指定されたファイル名をそのまま使用する。

-X は、どの位置に指定されても、-x の指定より優先する。

本指定は、環境変数の指定より優先する。

(K) -x

extension をデフォルトの拡張子とする。

本指定は、最後に指定されたものが有効となる。また、環境変数の指定より優先する。

デフォルトの拡張子が、'.'のときは、-X と同じに扱いとなる。

(L) -d DEBUG_OPTIONS

デバッグ・オプションを設定する。(5. デバッグ・オプションを参照)

(M) -o EXEC_OPTIONS
実行時オプションを設定する。(言語仕様書の付録の実行時オプションを参照)
オプション番号１の値から順にカンマで区切って指定する。
値が、「オプション番号=値」の形式のときは、そのように設定する。
d.d.d.d形式のときは、dを4バイト整数の各1バイトとして設定する。
Dを省略したときは、0を設定する。dは、下位から詰めて行く。

(N) -r[out_file]
OUTPUTコマンドでパケットに出力されたデータを標準出力またはファイルに出力する。
out_fileが指定されたときに当該ファイルに出力する。

(O) -t max_thread
同時実行数を指定する。0以下、または、253以上のときは、1となる。
デフォルト値は、1。

(P) -m[level][,[J|E|Other]]
ログ出力レベルをlevelに設定する。初期値は、０。
メッセージの言語種別を指定する。cpf.infのMSGLANGを参照。

(Q) -D DEF_PARM
define情報を設定する。この設定は、スクリプト内の設定より優先する。
DEF_PARM: キーワード[(a,b,...)][=定義値]

(2)script_name
(A)ファイル名
実行するスクリプト・ファイル名を指定する。ファイル名には、ディレクトリ名を含んでもよい。
(B)拡張子
ファイル名に拡張子が付いていないときは、デフォルトの拡張子('cl')が付加される。
ただし、拡張子処理は、拡張子オプションに従う。
(C)サーチパス
スクリプト・ファイルは、最初に実行ディレクトリをカレント・ディレクトリとしてサーチされる。
環境変数にSCRIPTPATHが設定されているときは、さらに、このパスをサーチし最初に見つかったファイルを実行する。ファイル名の先頭が、'/'、'./'、'../'ときは、このパスはサーチしない。
また、環境変数にCPF_HOMEが設定されているときは、上記で決定したパスの前に、この指定を付加する。
ただし、上記で決定したパスが絶対パスのときは、この指定を付加しない。
(D)スクリプト内容の直接指定
ファイル名の代わりに、スクリプト内容を中括弧({})で囲って直接指定する。
中括弧({})の外側の空白文字と改行コード(¥n,¥r)は、無視される。
【例】

```
coal "{proc main;print 'Hellow World.';return 0;endproc;}"
```

(3)parm-1 parm-2 ...
スクリプトに渡すパラメータを指定する。
パラメータは全て文字属性となる。

3. スクリプトの終了

以下のとき、スクリプトは終了する。
(1)最初に実行したスクリプトからリターンしたとき。
(2)ユーザまたはシステムエラーが発生したとき。

(3)EXIT コマンドまたは EXIT 関数を実行したとき。
(4)FSHUT モードまたはサスペンドなしの SHUT モードになったとき。

4. 動作環境の設定

4.1. 環境変数

(A)SCRIPTPATH
スクリプト・ファイルをサーチするパスをコロンでつなげて指定する。デフォルトは、未設定。
スクリプト・ファイルは、最初に実行ディレクトリをカレント・ディレクトリとしてサーチされる。
したがって、パス中にカレント・ディレクトリの指定があっても無視される。

(B)SCREXTENSION
スクリプト・ファイルのデフォルトの拡張子を指定する。

(C)AKB_HOME
akb コンフィグ・ファイルの格納ディレクトリを指定する。デフォルトは実行ディレクトリ。

(D)AKB_INI
akb コンフィグ・ファイルのファイル名を指定する。デフォルトは"akb.ini"。
絶対パスで指定した場合は、AKB_HOME は無視される。

4.2. 設定ファイル

4.2.1. akb.ini

詳細は、アプリケーション基盤インタフェース仕様書を参照。

（１）機能
実行パラメータを設定するファイル。

（２）形式
（Ａ）一般形式

```
#
# コメント
#
[セクション名]
キーワード△パラメータのならび # コメント
  ・・・
```

△は、一つ以上空白を表し半角ブランクまたはタブ

（Ｂ）一般規則
（ａ）１レコードは２５５ｂｙｔｅまでを有効とし、２５６ｂｙｔｅ以降は読み捨てる。
（ｂ）シャープ記号(#) 以降はコメントとなる。
（ｃ）セクションは、'[' で始まり、'[' の前で終わる。
（ｄ）キーワードとパラメータは、１行で指定する。複数行に渡たることはできない。
（ｅ）キーワードと最初のパラメータの間には、等号(=) があってもよい。

キーワードの前には、空白があってもよい。

（ｆ）各パラメータは、空白で区切る。

パラメータは、引用符（または２重引用符）で囲むことができる。

引用符の扱いは、一般的な規則（内部に引用符を指定するときは２個連続して指定する等）に従う。

（３）詳細

（Ａ）システム・セクション

```
[AKB_SYSTEM]
LOGDIR△<ログ出力ディレクトリ名>
<ログ名称>△<制御情報>
RWQ_TIMEOUT△<RQueTimeOut>△<WQueTimeOut> (sec)
TIMEOUT△<time1>△<time2>△<time3>△・・・
MSG_TIMEOUT△<MsgTimeOut> (sec)
MSGLANG△<Language>
```

（ａ）LOGDIR

ディレクトリ名はAKB_HOMEからの相対パスまたは絶対パスで指定する。

（ｂ）ログ名称

①<ログ名称>に対応するログの制御情報を設定する。

②<ログ名称>には、ERROR_LOG,PRINT_LOG,DEBUG_LOG,STATI_LOGを指定する。

ERROR_LOG：エラーログ

PRINT_LOG：プリントログ

DEBUG_LOG：デバッグログ

STATI_LOG：統計情報ログ

③制御情報には、「4.ログの設定」のLOG_PARMを空白文字で区切って指定する。

（ｃ）RWQ_TIMEOUT

ReadQueとWriteQueのタイムアウト値(sec)を指定する。

キーワードの設定がないときは、タイムアウト処理を行わない。

両方のタイムアウト値の指定がないときは、通信タイムアウト値が使われる。

WriteQueのタイムアウト値の指定がないときは、ReadQueの値が使われる。

タイムアウト値が負のときは、タイムアウト値は無限とみなされる。

（ｄ）TIMEOUT

通信タイムアウト値を指定する。値のならびは、TIMEOUTセクションと同じ。

（ｅ）MSG_TIMEOUT

送信メッセージのタイムアウト値を指定する。０以下のときは無視される。

デフォルトは、３０秒。

（ｆ）MSGLANG

メッセージの言語種別を指定する。<Language>の先頭１バイトが有効。

デフォルトは、J[apanese]。

・日本語：{J|j}[apanese]

・英　語：{E|e}[nglish]

（Ｂ）通信タイムアウト・セクション

プロセス毎の通信タイムアウト時間(sec)を設定する。デフォルトは１５秒。

```
[TIMEOUT]
プロセス名△タイムアウト時間１△タイムアウト時間２△タイムアウト時間３...
  ・・・
```

（ａ）タイムアウト時間１から順に、read,write,connect,accept,RQue,WQue に対応する。

<RQue>　　:Read　Queのデータ入力保留待ち時間

<WQue>　　:Write Queのデータ出力保留待ち時間

（ｂ）タイムアウト時間が、>= -1 のときは、そのまま設定される。

タイムアウト時間が、== -2 のときは、DEFAULT_TIMEOUT(15)を設定する。

タイムアウト時間が、<= -3 のときは、設定されない。

（注）タイムアウト時間 == -1 は、タイムアウトなしを示す。

タイムアウト時間 ==　0 は、select()に使うときは、そのまま使用する。

alarm()に使うときは、１とする。

（Ｃ）Ｃｏａｌセクション

Ｃｏａｌ固有の設定値を指定する。

```
[COAL]
MSGTIMEOUT△<最大実行時間(秒)>
LRU_SCR_MAX△<スクリプト・キャッシュ数>
OPTIONS△<オプション値の並び>
MSGLANG△<Language>
UTF8_CONV△<変換表ファイル名>
```

（ａ）MSGTIMEOUT

セッションの最大実行時間を秒単位で指定する。デフォルトは、６００秒。

（ｂ）LRU_SCR_MAX

キャッシュするスクリプトの最大数を指定する。デフォルトは、２０。

（ｃ）OPTIONS

実行時オプションを設定する。

空白文字で区切って指定することを除き、コマンドラインと同じ。

（ｄ）MSGLANG

メッセージの言語種別を指定する。<Language>の先頭１バイトが有効。

デフォルトは、J[apanese]。

・日本語：{J|j}[apanese]

・英　語：{E|e}[nglish]

（ｅ）UTF8_CONV

日本語文字コードの変換表ファイル名を指定する。

4.2.2. dscmndscript

（１）機能

スクリプト実行ツール(未サポート)で指定するコマンド番号とスクリプト・ファイル名の対応を設定する。

（２）形式

（A）一般形式

```
#
# コメント
#
コマンド番号 △ スクリプト・ファイル名［△ コメント ］
  ・・・
```

（B）一般規則

（ａ）１レコードは２５５ｂｙｔｅまでを有効とし、２５６ｂｙｔｅ以降は読み捨てる。

（ｂ）シャープ記号(#) 以降はコメントとなる。

（ｃ）コマンド番号は、８桁の１６進数で指定する。

4.2.3. coal.ctl

（１）機能

設定する。

（２）形式

（A）一般形式

```
#
# コメント
#
afhash:[△]<n>
afdump:[△]<memory_file>
nfdump:[△]<nofree_file>
```

（B）一般規則

（a）１レコードは２５５ｂｙｔｅまでを有効とし、２５６ｂｙｔｅ以降は読み捨てる。

（b）シャープ記号(#) 以降はコメントとなる。

（3）詳細

（A）設定値

（a）n

管理するアドレス数の初期値、管理エントリに空きがなくなったときは、nづつ増える。

（b）memory_file

malloc,realloc,free したアドレス等の情報を出力するファイル名。

（c）nofree_file

Nofree実行時にチェックされる未freeアドレスを出力するるファイル名。

（B）一般規則

（a）afhashを指定すると、malloc,realloc,freeしたアドレスを管理する。

（b）afdumpを指定すると、malloc,realloc,freeしたアドレスと実行したソースファイル名、行数をファイルに出力する。

（c）nfdump を指定すると、Nofree 実行時にチェックされる未開放アドレスをファイルに出力する。

Nofree は、で実行する。

・スクリプト実行終了時

・Coal 終了時

Nofree 実行時には、以下を memory_file に出力する。

```
***** Nofree(実行回数)(yyyy/mm/dd hh:mi:ss) *****
```

5. ログの設定

(1)--{e|p|d|l}[LOG_PARM]:

LOG_PARM = [FLAG],[LEVEL],[SIZE_MAX],[FILE_MAX],[OPTION],[FILE],[PRIORITY]

(a)ログ種別の指定

e:エラーログ
p:プリントログ
d:デバッグログ
l:全てのログ
大文字のときは、libakx.a のログを設定する。

(b)パラメータ

FLAG:出力フラグ
0x01:標準出力
0x02:ファイル出力
0x04:標準エラー出力
0x08:syslog 出力
その他は，ＡＫＢインタフェース仕様書を参照。
LEVEL:出力レベル（未使用）
SIZE_MAX:ローテーション・ファイル・サイズ(Kbyte)
>0 のとき，ローテーションする。
FILE_MAX:ローテーション・ファイル数
OPTION:ローテーション・オプション
0x01 ビット: 0 のとき，退避ファイル名をローテーションする。
1 のとき，出力ファイル名をローテーションする。
FILE:出力ファイル名
実行ディレクトリからの相対パスか絶対パスで指定する。
PRIORITY:プライオリティ（未使用）

(c)規則

--d のみの指定のときは，出力フラグの標準エラー出力のみがＯＮになる。
カンマ以降のパラメータを指定しないときは，カンマは不要。

(d)パラメータの初期値

ログ種別	FLAG	LEVEL	SIZE_MAX	FILE_MAX	OPTION	FILE	PRIORITY
エラー	0x0004	0	0	3	0x00	error_log	0
プリント	0x0004	0	0	3	0x00	print_log	0
デバッグ	0x0000	0	0	3	0x00	debug_log	0

6. デバッグ・オプション

(1)-d DEBUG_OPTIONS

DEBUG_OPTIONS := [OPT0],[OPT1],[OPT2],[OPT3]

(a)OPT0

> 0 : ステップ単位で実行する。実行するステップの前で止まり、コマンドを入力する。
0x02: 実行するステップを出力しない。
以下のプロンプトを出力する。

スクリプト名(ステップのライン番号)>

デバッグコマンド一覧

No.	コマンド	機能
1	QUIT	デバッグを終了する
2	THRU	デバッグモードを抜ける
3	NEXT	次のブロックの前まで実行する
4	STEP [n]	n ステップ実行する
5	UNTIL [n]	・n 行まで実行する。 ・n を省略したときは、現在行を超える行まで実行する。 現在行が LOOP の開始行のときは、LOOP を抜けたところまで実行する。
6	LIST 関数名	関数のソースを標準出力に出力する
7	LIST [from][,][n]	・スクリプトの from 行目から n 行を標準出力に出力する。 ・from と n を省略したときは、現在行を出力する。 ・n を省略したときは、from±10 行を出力する。 ・from を省略したときは、現在行から n 行を標準出力に出力する。
8	PRINT 式 ・・・	式の内容を標準出力に出力する
9	LET or SET 代入式	代入式を実行する
10	/	直前の入力を繰り返す

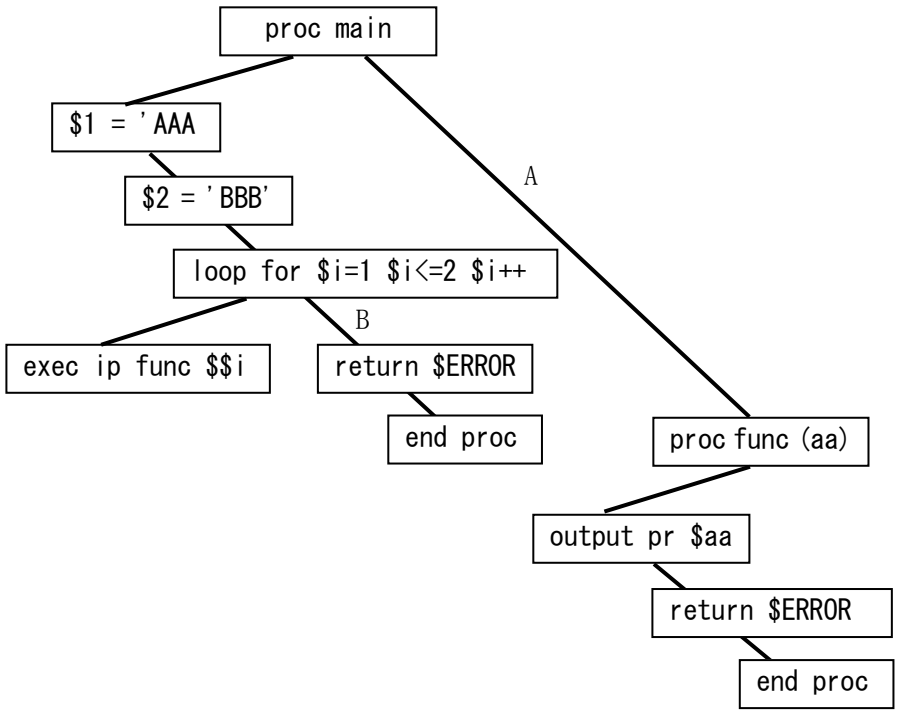
(b)OPT1

0x01: プログラム tree を標準出力に出力する。

【例】

スクリプト	ｔｒｅｅ出力
// proc main; 　　$1 = 'AAA'; 　　$2 = 'BBB'; 　　loop for $i=1 $i<=2 $i++; 　　　　exec ip func $$i; 　　endloop; 　　return $ERROR; end proc; proc func (aa); 　　output pr $aa; 　　return $ERROR; end proc;	[PROC main]->A \| *[LET $1 = 'AAA'] -*[LET $2 = 'BBB'] -[LOOP for $i=1 $i<=2 $i++]->B \| *[EXEC ip func $$i]. B-*[RETURN $ERROR] -*[ENDPROC]. A-[PROC func (aa)]. | *[OUTPUT pr $aa] -*[RETURN $ERROR] -*[ENDPROC].

ｔｒｅｅ構造



（注） ｔｒｅｅ出力の記号（left:下位レベル、right:同レベル）

　　| : left リーフの始まり

　　* : left リーフを持たない leaf

　　- : right リーフの始まり

　　. : right リーフを持たない leaf

　　->英字: 同じレベルでの接続を示す

0x02: このビットは、OPT1 の 0x01 がＯＮのときのみ有効。
このビットが、ＯＦＦのとき、tree 出力して終了する。
ＯＮのとき、終了しない。
0x04: このビットは、OPT1 の 0x01 がＯＮのときのみ有効。
このビットが、ＯＮのとき、ノードのコマンド名の前にレベルを出力する。
ＯＦＦのとき、出力しない。

(c)OPT2

0x01: 1/0 = 毎回コンパイルする。/ ２回目以降はコンパイル結果を使用する。
0x10: リターン値が負のとき、関数名と行数を DEBUGOUT する。

(d)OPT3

未使用

7.動作環境と文字コード

7.1. Cygwin 環境

Cygwin の内部コードは、UTF-8 である。Coal の内部コードも UTF-8 である。
Coal は、S-JIS 環境を前提としている。(LANG=ja_JP.SJIS)
したがって、入出力時には、以下のように文字コードを変換する。
現在は、S-JIS と UTF-8 のみをサポート。

(1)標準入力
Cygwin が S-JIS から UTF-8 に自動変換する。
Coal は、何もしない。

(2)標準入力へのリダイレクト
Cygwin は、何もしない。
Coal は、標準入力時には、何もしないので、S-JIS コードのデータを入力するときは、実行時オプション２１で、入力の文字コードを指定する必要がある。

【例】coal -o21=0x10 test1 < in_file

(3)ファイル入力
Cygwin は、何もしない。
データの文字コードを実行時オプション２１で指定する。
S-JIS のときは、デフォルトのため、指定は不要。

(4)標準出力、標準エラー出力
Cygwin が UTF-8 から S-JIS に自動変換する。
Coal は、何もしない。

(5)標準出力、標準エラー出力のリダイレクト
Cygwin は、何もしない。
Coal は、標準出力、標準エラー出力時には、何もしないので、S-JIS コードのデータを出力するときは、実行時オプション２１で、出力の文字コードを指定する必要がある。

【例】coal -o21=0x100 test2 > out_file

(6)ファイル出力
Cygwin は、何もしない。
データの文字コードを実行時オプション２１で指定する。
S-JIS のときは、デフォルトのため、指定は不要。

(7)スクリプトファイル
S-JIS がデフォルトであり、UTF-8 に自動変換する。
UTF-8 のときは、実行時オプション２１で指定する。

【例】coal -o21=0x2 test3