

KaracrixBuilderV3 システムマニュアル

18章 リソースファイルの入出力

(章別取扱説明書 v1.00)

株式会社 エスアイ創房

KaracrixBuilder

■改定履歴

第 1.00 版 2008/12/1

■ おことわり

- (1) 本書内容の一部又は全部を、無断で他に転載することは禁止されています。
- (2) 本書内容は、将来予告無く変更する場合があります。

KARACRIX は株式会社エスアイ創房の登録商標です。

KaracrixBuilderV3 システムマニュアル 第 1.00 版 © S.I.Soubou Inc.

目次

18 章	リソースファイルの入出力.....	18-1
18.1	リソースファイルの構成の解説.....	18-1
18.1.1	制御&帳票プログラム・本体ファイルのリソースの構成.....	18-3
18.1.2	制御プログラム・パラメータファイルのリソースの構成.....	18-4
18.1.3	パラメータ付き制御プログラムのリソースの構成.....	18-6
18.1.4	監視パネル&複部品のリソースの構成.....	18-9
18.1.5	ビットマップファイル（絵部品）のリソースの構成.....	18-11
18.1.6	帳票プログラム・パラメータファイルのリソースの構成.....	18-12
18.1.7	帳票フォーマット&複部品のリソースの構成.....	18-14
18.2	制御プログラムファイル.....	18-16
18.2.1	制御プログラムファイルのインポート.....	18-16
18.2.2	制御プログラムファイルのエクスポート.....	18-22
18.3	制御&帳票パラメータファイル.....	18-25
18.3.1	制御&帳票パラメータファイルのインポート.....	18-25
18.3.2	制御&帳票パラメータファイルのエクスポート.....	18-29
18.4	ポイント汎用属性名称のインポート.....	18-30
18.5	監視パネル&複部品ファイル.....	18-37
18.5.1	監視パネル&複部品ファイルのインポート.....	18-37
18.5.2	監視パネルファイルのエクスポート.....	18-44
18.6	ビットマップファイル.....	18-46
18.6.1	ビットマップファイルのインポート.....	18-46
18.6.2	ビットマップファイルのエクスポート.....	18-49
18.7	帳票プログラムファイル.....	18-50
18.7.1	帳票プログラムファイルのインポート.....	18-50
18.7.2	帳票プログラムファイルのエクスポート.....	18-54
18.8	帳票書式ファイル.....	18-56
18.8.1	帳票書式ファイルのインポート.....	18-56
18.8.2	帳票書式ファイルのエクスポート.....	18-59

18章 リソースファイルの入出力

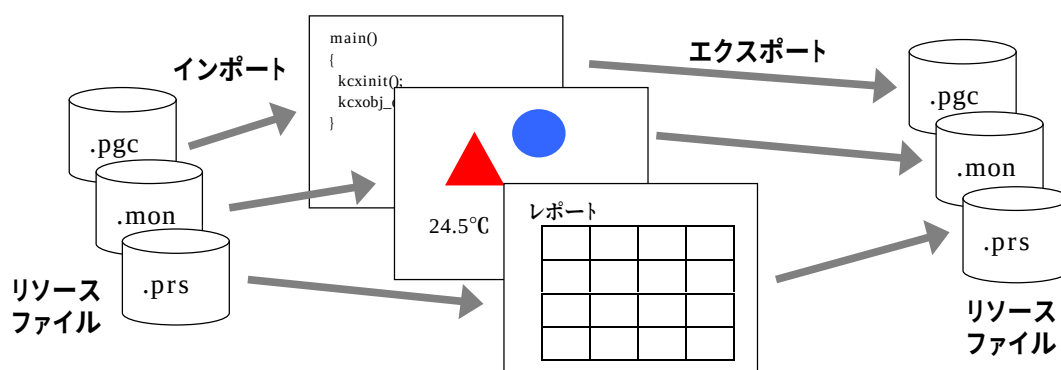
KaracrixBuilder では、作成した各種データを共用資源ファイルとして扱うことができ、リソースファイルと呼んでいます。そして、システムを構築する場合に、これらリソースファイルを取り込んで編集することができます。リソースを KaracrixBuilder に取り込むことをインポートと呼びます。

外部ファイルからインポートできるリソースは、監視パネルファイル・制御プログラムファイル・制御パラメータファイル・帳票書式ファイル・帳票プログラムファイル・CAD 用ビットマップファイルなどです。

また、作成したデータをリソースとして外部ファイルに出力することをエクスポートと呼びます。

18.1 リソースファイルの構成の解説

リソースファイルの概要について以下で解説します。



(1) リソースファイルの種類

KaracrixBuilder で扱えるリソースファイルの一覧を以下の表に示します。

※リソースファイルは、テキストデータですので一般のエディタで参照できます。

表 18.1.1 リソースファイル一覧

リソース名	内容	拡張子	KaracrixBuilder での入出力画面
制御プログラム・本体ファイル	C 言語ソース	.pgc	制御プログラム登録画面
制御プログラム・パラメータファイル	パラメータリスト	.pra	制御パラメータ登録画面
制御プログラム・本体ファイル (汎用属性名称付属部)	汎用属性名称リスト	-	-
監視画面ファイル& 監視画面用複部品ファイル	監視画面 CAD 出力ファイル	.mon	監視パネル管理画面
ビットマップファイル	X Window ビットマップファイル	.xbm	ビットマップ一覧 (CAD 画面内) 画面
帳票プログラム・本体ファイル	C 言語ソース	.pgc	帳票プログラム登録画面
帳票プログラム・パラメータファイル	パラメータリスト	.pra	帳票パラメータ登録画面
帳票フォーマットファイル& 帳票フォーマット用複部品ファイル	帳票 CAD 出力ファイル	.prs	帳票フォーマット登録画面

(2) リソースファイルの入出力に使用するディレクトリ

リソースファイルのインポート及びエクスポートに使用されるディレクトリはデフォルトで“/tmp”です。また、以下の説明でも“/tmp”を例に解説しています。

※インポート、エクスポートディレクトリを変更したい場合には、「24 章システムとファイル仕様 資料 1 syscnf.env」の“externalsrcpath”パラメータを変更して下さい。

(3) リソースファイルの移植時の注意点

KaracrixBuilder システムからエクスポートされたリソースファイルを他の KaracrixBuilder システムにインポートして移植する場合に、リソース中のポイントオブジェクト ID が移植する KaracrixBuilder に登録されていない場合以下のような問題が生じます。

①制御プログラムの場合、オブジェクト ID をオープンする時にエラーとなります。

※di999 という ID が存在しないシステム上で、kcxobj_open("di999")等の関数を記述している時など。

②監視画面の場合、部品の存在を極力残すため同種の他のオブジェクト ID に置き換え (強制変更) いたします。同種の部品が存在しなかった場合、部品は削除されます。

同種とは、Di、Do、Ai、Ao、Pi ポイント種別を指します。

※置き換えの例: di888 という Di 種別の ID は存在するが di999 という Di 種別の ID が存在しないシステム上で、di999 が部品に登録されているリソースでは、di999 は di888 に強制変更されます。同種のものが複数ある場合にはポイント登録の上位より割り当てられます。

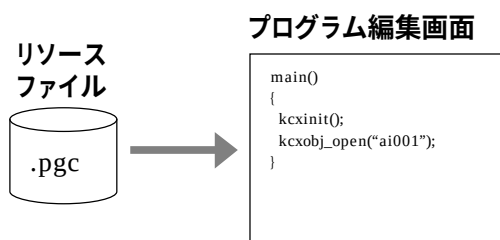
※削除の例: Di 種別の ID が存在しないシステム上で、Di 種別の di999 が部品に登録されているリソースでは、この部品は削除されます。

18.1.1 制御&帳票プログラム・本体ファイルのリソースの構成

制御&帳票プログラムの本体ファイルは、C 言語で記述されたテキストファイルです。下記に簡単なプログラム本体ファイルの例を示します。

```
#include <karacrix.h>
main(argc,argv)
int    argc;
char  *argv[];
{
    kcxinit( argc, argv );
}
```

上記のリソースファイルを「プログラム簡易編集」画面で読み込んだ様子を以下に示します。



No.	プログラム
1	#include <karacrix.h> ↵
2	main(argc, argv) ↵
3	int argc; ↵
4	char *argv[]; ↵
5	{ ↵
6	kcxinit(argc, argv); ↵
7	} ↵
8	
9	
10	

図 18.1.1 制御プログラムリソースの編集表示画面例

18.1.2 制御プログラム・パラメータファイルのリソースの構成

制御パラメータは、制御プログラムが読み込んで条件判断や処理ロジックを変更する場合に使用します。制御パラメータファイルは、「制御パラメータ登録」画面の設定データを以下に示す順序に編成させたテキストデータです。

- ①表示番号[No-1] (0～299:整数)
- ②キーワード (データ文字列)
- ③定義 (データ文字列)
- ④W3 許可 (0,1,2:整数)
- ⑤MB 許可 (0,1,2:整数)
- ⑥キーワード説明 (コメント文字列)
- ⑦備考 (コメント文字列)

上記のパラメータで、制御プログラムから読み取ることができるものは、②キーワードと③定義です。

④と⑤は外部からのアクセス制御用に KaracrixBuilder が使用します。

⑥と⑦は、コンソール画面、Web 画面、携帯端末画面の表示に使用されます。

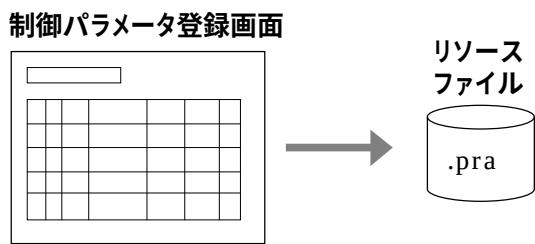
「制御パラメータ登録」画面でパラメータを設定し、そのリソースを出力したファイルの内容例を紹介します。

「制御パラメータ登録」画面で、下記の表示内容でデータを設定した場合、

画面消去 インポート エクスポート					
⑥ CLEAR ⑦ GET ⑦ PUT					
①	②	③	⑥	⑦	④ ⑤
No.	キーワード	定義	入力画面: キーワード説明	入力画面: 備考	W3 MB
1	perm	on	メール許可	on/off	* *
2	ip	BL 192.168.0.200 20000	装置	型	* -
3					
4					

図 18.1.2 「制御パラメータ登録」画面の設定例

このリソースは、以下のように出力されます。



①No の番号-1 ②キーワード ③定義 ④Web 許可 ⑤MB 許可 ⑥キーワード説明 ⑦備考

①	②	③	④⑤	⑥	⑦
0	"perm"	"on"	1,1	"メール許可"	"on/off"
1	"ip"	"BK 192.168.0.200 20000"	1,0	"装置"	"型"

18.1.3 パラメータ付き制御プログラムのリソースの構成

制御プログラムで使用する制御パラメータは、互いに対になって動作する場合がほとんどですので、リソースファイルとしてエクスポートするときに、制御プログラム本体のリソースファイルの中に、制御プログラムパラメータを取込んで一体化してエクスポートできるようになっています。

しかし、この機能の為、制御プログラムのインポート及びエクスポートする場合の操作が少々判りづらくなっています。ここでは、制御プログラムリソースの構成を解説致しますので、操作方法の理解の一助として下さい。

(1) リソースの構成

制御プログラム及びパラメータはそれぞれ、テキスト形式で記述されています。

制御プログラムにパラメータを取込んだ場合とは、制御プログラム部の下方にパラメータを付け足した状態を言います。

但し、パラメータを付け加えただけでは、それぞれの区別がつかなくなりますので、区切りマークも付加されます。

区切りマークは、パラメータ部の上下両端に以下のようなテキスト行を付け加えたものです。

制御プログラム部

/*KCX-PARA OK ←パラメータ部開始行追加（左端スペース不可、OK 間の区切りは半角スペース）

制御パラメータ部

KCX-PARA*/ ←パラメータ部終了行追加（左端スペース不可）

(2) 具体的なファイル内容例

参考のためリソースファイルをテキストエディタで表示した場合の例を以下に示します。

※下記プログラム部とパラメータ部の内容は参考例のため特別な意味はありません。

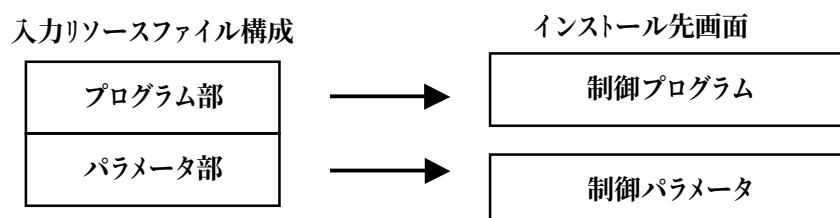
```
#include <karacrix.h>
main(argc,argv)
int   argc;
char  *argv[];
{
    kcxinit( argc, argv );
}
/*KCX-PARA OK
0,"perm","on",1,1,"メール許可","on/off"
1,"ip","RT 192.168.0.200 20000",1,0,"装置","型"
KCX-PARA*/
```

(3) インポート時の画面選択とその結果

パラメータ付き制御プログラムをインポートするときに、パラメータ部を分離して取り込むか否かを選択できます。以下に、その時の動作について説明します。

●「プログラムからパラメータを分離取り込み」を選択した場合

プログラム部は制御プログラム画面へ、パラメータ部は制御パラメータ画面へ分離して取り込まれます。

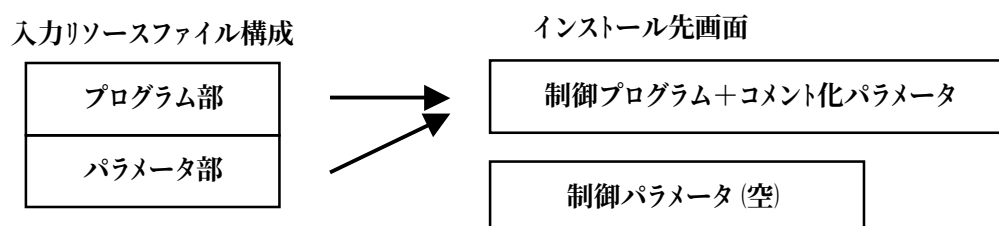


●「プログラムからパラメータを分離取り込み」を拒否した場合

パラメータ部も含め入力ファイルの内容が全て制御プログラム画面へ取り込まれます。取り込まれたパラメータ部は、コメント文として存在することになります。

このコメント化されたパラメータ部が不要な場合には、その部分をプログラム編集画面で削除して下さい。

パラメータ部の取込が不要な場合、リソースファイルをエディタで予め編集しておくとい良いでしょう。

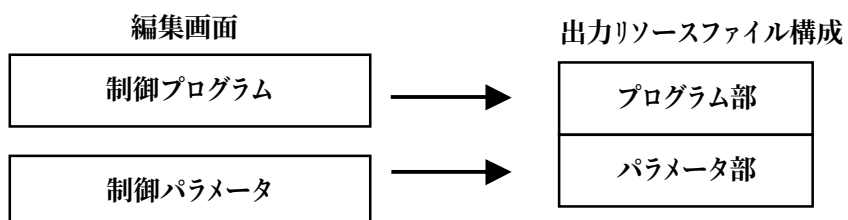


(4)エクスポート時の画面選択とその結果

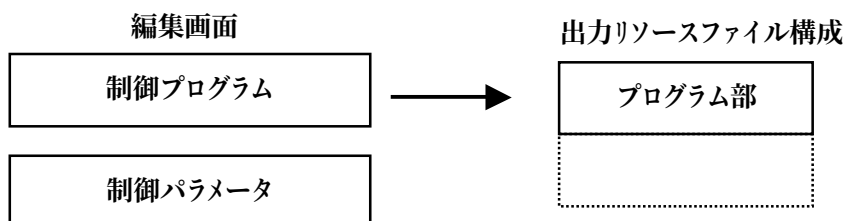
パラメータ付き制御プログラムをエクスポートするときに、プログラム部とパラメータ部を一体化して出力するか否かを選択できます。以下に、その時の動作について説明します。

●「プログラムにパラメータを添付」を選択した場合

作成されているプログラムとパラメータを一体化させてリソースファイルに出力されます。

**●「プログラムにパラメータを添付」を拒否した場合**

作成されているプログラムのみがリソースファイルに出力されます。



18.1.4 監視パネル&複部品のリソースの構成

監視パネルのリソースは、静部品と動部品により構成されています。

静部品は、画面の背景を描くものです。

動部品は、ポイントオブジェクト及びグラフを指定して画面上で変化するものです。

また、複部品は複数の部品を1つの部品として扱うために作成された部分的な監視パネルです。複部品ファイル構成は、監視パネルファイルと全く同じ構成になっています。

以下の例で静部品、動部品を配置したリソースファイルの構成について解説します。

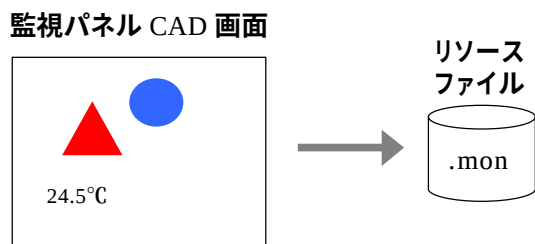
下記画面に示すように静部品1つ、動部品1つを描いてみた例です。

1. (20,20)座標と(100,70)座標を対とする四角形を描画
2. 四角形内の(30,50)座標に数値表示させるため、アナログポイントオブジェクトを配置



図 18.1.3 監視パネル画面作成サンプル

上記の場合のリソースファイルの内容を以下に示します。



stenv、stextent は、静部品用画面環境情報を示すものです。

strectangle は、四角形の描画コマンドを示します。

env9 は、動部品用画面環境情報を示すものです。

mdfig は、アナログポイントの値を表示する描画コマンドを示します。

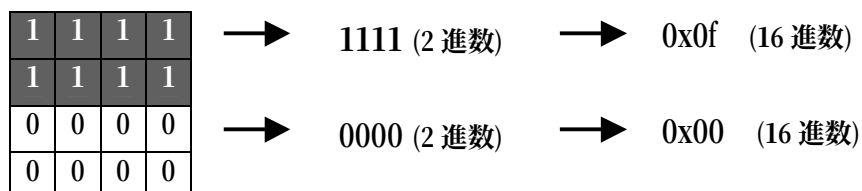
```
0,stenv,      1,1,1,A4,landscape,1.000000,1.000000,1.000000
1,stextent,   20,20,100,70
2,strectangle,0,5.000000,solid,butt,miter,BLACKc,BLACKc,filnon
&,20.000000,20.000000
&,100.000000,70.000000
&,end
&,pat0,0.000000,0.000000,1.000000,1.000000,1.000000,1.000000,BLACKc,WHITEc,0,0
#
0,env9,98,"",offset,88,48,0,0,1104,816,"","","",CursorF_0c,CursorB_0c,color,WHITEc
1,mdfig,0,"ai001",431,"",-1,0,0,0,30,50,0,0,"",KcxMbFont.14x14,fb,2,0
```

18.1.5 ビットマップファイル（絵部品）のリソースの構成

ビットマップ絵部品は、XBM と呼ばれる X Window システムの標準の白黒画像ファイルフォーマットであるビットマップ形式のファイルです。

詳しくは、X Window システムの関連書籍や「XBM」をインターネット検索して参照して下さい。

このファイルは、C 言語の文法で記述された静的文字型変数を用いてビット状態を定義するものです。
 以下のような 4×4 ドットで構成される絵の上半分を ON にして下半分を OFF にした場合、



上記の絵の X Window ビットマップファイルの内容は以下に示されたものになります。
 (ビットマップ名を abc としています)

```
#define abc_width 4
#define abc_height 4
static unsigned char abc_bits[] = { 0x0f, 0x0f, 0x00, 0x00 };
```

18.1.6 帳票プログラム・パラメータファイルのリソースの構成

帳票パラメータは、帳票プログラムが読み込んで条件判断や処理ロジックを変更する場合に使用します。帳票パラメータファイルは、「帳票パラメータ登録」画面の設定データを以下に示す順序に編成させたテキストデータです。制御パラメータとの違いは、携帯端末 (MB) からのアクセスが許可されない (未使用) ことです。

- ①表示番号[No-1] (0～299:整数)
- ②キーワード (データ文字列)
- ③定義 (データ文字列)
- ④W3 許可 (0,1,2:整数)
- ⑤MB 許可 (予備)
- ⑥キーワード説明 (コメント文字列)
- ⑦備考 (コメント文字列)

上記のパラメータで、帳票プログラムから読み取ることができるものは、②キーワードと③定義です。

④は外部からのアクセス制御用に KaracrixBuilder が使用します。

⑥と⑦は、コンソール画面、Web 画面の表示に使用されます。

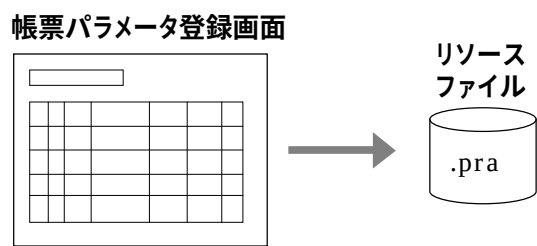
「帳票パラメータ登録」画面でパラメータを設定し、そのリソースを出力したファイルの内容例を紹介します。

「帳票パラメータ登録」画面で、下記の表示内容でデータを設定した場合、

画面消去 インポート エクスポート ⑤ CLEAR GET ⑥ PUT ④					
①	②	③			
No.	キーワード	定義	入力画面: キーワード説明	入力画面: 備考	W3
1	title	タイトル	タイトル	文書名	*
2	year	2008	月報 年	年入力	S
3					
4					

図 18.1.4 「帳票パラメータ登録」画面の設定例

このリソースは、以下のように出力されます。



①No の番号-1 ②キーワード ③定義 ④Web 許可 ⑤MB 許可 ⑥キーワード説明 ⑦備考

①	②	③	④⑤	⑥	⑦
0,"title",	"タイトル",	1,0,"タイトル",	"文書名"		
1,"year",	"2008",	2,0,"月報 年",	"年入力"		

18.1.7 帳票フォーマット&複部品のリソースの構成

帳票フォーマットのリソースは、静部品と原点部品により構成されています。

静部品は、帳票フォーマットを構成する罫線や文字情報などを描くものです。

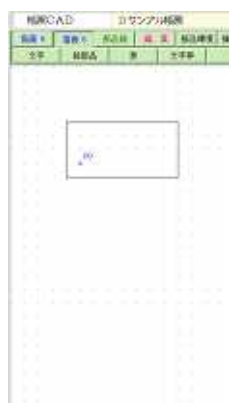
原点部品は、帳票プログラムからデータを印字する時に使用される位置やフォント情報を示すものです。

また、複部品は複数の部品を1つの部品として扱うために作成された部分的な帳票フォーマットです。複部品のファイル構成は、帳票フォーマットファイルと全く同じ構成になっています。

以下の例で静部品、原点部品を配置したリソースファイルの構成について解説します。

下記の画面に示すように静部品1つ、原点部品1つを描いてみた例です。

1. (20.0,20.0)座標と(60.0,40.0)座標を対とする四角形を描画
2. 四角形内の(25.0,35.0)座標に印字出力させるため点原点部品を配置



「帳票フォーマット」表示例



「帳票 CAD」静部品入力例



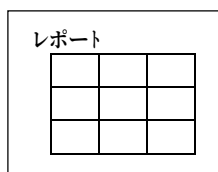
「帳票 CAD」原点部品入力例

図 18.1.5 帳票フォーマット作成サンプル

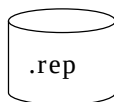
帳票 CAD 画面

上記の場合のリソースファイルの内容を以下に示します。

帳票フォーマット CAD 画面



リソース
ファイル



stenv、stextent は、静部品用環境情報を示すものです。

strectangle は、四角形の描画コマンドを示します。

stopoint は、点原点部品の描画コマンドを示します。

```
0,stenv,      1,1,0,A4,portrait,2.152381,0.267188,0.264648
1,stextent,   20,20,60,40
2,strectangle,0,0.200000,solid,butt,miter,BLACKc,BLACKc,filnon
&,20.000000,20.000000
&,60.000000,40.000000
&,end
&,pat0,0.000000,0.000000,3.000000,3.000000,1.000000,1.000000,BLACKc,WHITEc,0,1
3,stopoint,0,0,KcxVecBzrFont.nxn0,BLACKc,WHITEc,0
&,4.000000,4.000000,0,1,0,0,3.000000,3.000000,0,0,0,0,0,0,0
&,25.000000,35.000000
```

18.2 制御プログラムファイル

制御プログラムファイルは、制御プログラム登録で編集、作成されるプログラムリソースファイルです。

18.2.1 制御プログラムファイルのインポート

(1) 通常の制御プログラムファイル

- ① “/tmp”に取り込む制御プログラムファイルを置きます。この時、拡張子を“*.pgc”としてください。
- ② 「制御プログラム登録」画面を表示して、何も登録されていないリスト上の“プログラム名”欄を選択します。文字入力ダイアログが表示されますので、任意の制御プログラム名を入力して ENT ボタンを押して下さい。（ここでは、「ON/OFF プログラム」と入力）



図 18.2.1 制御プログラム名の登録

- ③ 「外部データを取り込みますか？」という確認ダイアログが表示されますので「YES」を選択すると、インポートファイルが置かれたディレクトリ上 (/tmp) にある制御プログラムファイルのリストが表示されます。



図 18.2.2 外部ファイルのインポート確認ダイアログ

「選択」ダイアログから、インポートする制御プログラムファイルを選択します。

(ここでは、“on_off.pgc”を選択)

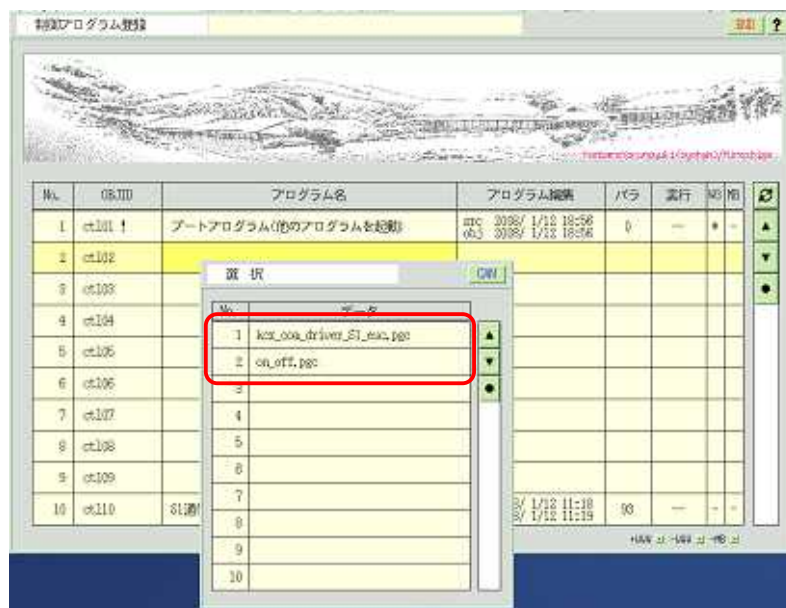


図 18.2.3 外部ファイルの一覧表示

④インポートする制御プログラムファイルを選択します。インポートされると図のような画面表示になります。



図 18.2.4 インポート直後の管理画面表示

- ⑤“プログラム編集”欄を選択すると編集画面でインポートされたプログラムを確認することができます。



図 18.2.5 インポートプログラムの編集

- ⑥インポートが正常に行われた後は、①でディレクトリ (/tmp) に置いたファイルは削除してかまいません。

(2) パラメータ付き制御プログラムファイル

制御パラメータを使用した制御プログラムをインポートすることにより、プログラムソースと同時に定義パラメータも KaracrixBuilder に取り込むことができます。

- ① “/tmp”に取り込むパラメータ付き制御プログラムファイルを置きます。この時、拡張子を“*.pgc”としてください。
- ② 「制御プログラム登録」画面を表示して、何も登録されていないリスト上の“プログラム名”欄を選択します。文字入力ダイアログが表示されますので、任意の制御プログラム名を入力して ENT ボタンを押して下さい。



図 18.2.6 制御プログラム名の登録

- ③ 「外部データを取り込みますか？」という確認ダイアログが表示されますので「YES」を選択すると、インポートファイルが置かれたディレクトリ (/tmp) 上にある制御プログラムファイルのリストが表示されます。



図 18.2.7 外部ファイルのインポート確認ダイアログ

「選択」ダイアログから、インポートする制御プログラムファイルを選択します。

(ここでは、“on_off_para.pgc”を選択)



図 18.2.8 外部ファイルの一覧表示

- ④「プログラムからパラメータを分離取り込みますか?」という確認ダイアログが表示されますので「YES」を選択すると、選択した制御プログラムがインポートされ、添付されているパラメータ定義情報も同時にインポートされます。



図 18.2.9 インポート直後の管理画面表示

- ⑤インポートされると図のような画面表示になります。「パラ」欄に“1”の値が表示されています。この場合パラメータ定義が1つあることを示しています。



№	OBJ	プログラム名	プログラム編集	パラ	実行	実行B	
1	et101	ブートプログラム(他のプログラムを起動)	src 2008/1/12 18:56 obj 2008/1/12 18:56	0	—	*	▲
2	et102	(H/PC)プログラム	src 2008/1/19 23:22 obj / / 1	0		-	▼
3	et103	(H/PC)プログラム2	src 2008/1/20 00:08 obj / / 1	1		-	●
4	et104						
5	et105						
6	et106						
7	et107						
8	et108						
9	et109						
10	et110	通信ドライバ	src 2008/1/12 11:18 obj 2008/1/12 11:19	99	—	-	

図 18.2.10 インポート直後の管理画面表示

- “パラ”欄をクリックすると「制御パラメータ登録」画面が表示されます。
パラメータ定義情報が確認できます。



№	キーワード	定義	入力画面: キーワード選択	入力画面: 備考	PC	実行B	
1	src	1	各々4時間	異なっています	*	*	▲
2							▼
3							●
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							

図 18.2.11 インポートされたパラメータ定義情報

- ⑥“プログラム編集”欄を選択すると編集画面でインポートされたプログラムを確認することができます。
⑦インポートが正常に行われた後は、①でディレクトリ (/tmp) に置いたファイルは削除してかまいません。

18.2.2 制御プログラムファイルのエクスポート

(1) 通常の制御プログラムファイル

「制御プログラム登録」画面に登録されている制御プログラムをエクスポートする場合には、以下の手順で行ないます。

- ① 「制御プログラム登録」画面を表示してリストスクロールボタンでエクスポートする制御プログラムの登録行を表示して下さい。次に、 ボタンを押して“PUT”モードを選択して、エクスポートする制御プログラムの登録行に対応する  ボタンを押して下さい。以下のように、エクスポートファイル名の「文字入力」ダイアログが表示されます。



図 18.2.12 エクスポートファイル名の編集

- ② ファイル名には、デフォルトの名前 (プログラムの OBJID) が設定されていますので、変更する場合にはこの画面で編集することができます。 (“ .pgc ” の拡張子を入力する必要はありません)

“ENT” ボタンを押すと、ファイル名を変更した場合も、変更しない場合にも、表示されている名前に “.pgc” の拡張子が付加されてファイルが作成出力されます。

※ ファイル名に “.pgc” 拡張子を付けて指定した場合、システムによる “.pgc” 拡張子の自動付加は行ないません。

(2)パラメータ定義情報付き制御プログラムファイル

「制御プログラム登録」画面に登録されている“パラメータ定義情報付き制御プログラム”をエクスポートする場合には、以下の手順で行ないます。

- ①「制御プログラム登録」画面を表示してリストスクロールボタンでエクスポートする制御プログラムの登録行を表示して下さい。次に、 ボタンを押して“PUT”モードを選択して、エクスポートする制御プログラムの登録行に対応する  ボタンを押して下さい。以下のように、エクスポートファイル名の「文字入力」ダイアログが表示されます。



図 18.2.13 エクスポートファイル名の編集

- ②ファイル名には、デフォルトの名前(プログラムの OBJID)が設定されていますので、変更する場合にはこの画面で編集することができます。(“.pgc”の拡張子を入力する必要はありません)
“ENT”ボタンを押すと、「プログラムにパラメータを添付しますか？」と確認ダイアログが表示されます。ここで、“YES”を選択すると、入力された名前に“.pgc”の拡張子が付加されてファイルが作成出力されます。



図 18.2.14 パラメータ添付の確認ダイアログ

[参考] パラメータ定義情報付き制御プログラムサンプル

“パラメータ定義情報付き制御プログラム”をエクスポートした場合に作成されるリソースファイルのサンプルを以下に示します。

(例) ON/OFF プログラム (パラメータ付き)

プログラムの概要は、do001 (デジタルポイント) を、パラメータに定義された (キーワード: sec、定義値: 1) WAIT 時間 (単位: 秒) の値を取得して、その WAIT 時間のインターバルで ON、OFF を繰り返す内容です。

```

/*インクルードファイル*/
#include <karacrix.h>
/* ポイント(do001)を1秒おきに ON/OFF させるプログラムの開始 */
main( argc, argv )
int  argc;
char *argv[];
{
    int  objid,status; /* 整数変数宣言 */
    int  waitsec;
    char gettext[128];
    char *keyword [BUFSIZ];
    char *paradata[BUFSIZ];
    /*KARACRIX ライブラリの初期設定*/
    kcxinit( argc, argv );
    /*ポイントオブジェクト ID の取得*/
    objid = kcxobj_open( "do001" );
    /*パラメータセットの取得*/
    keywords = kcxprg_para_data_get( 0, keyword, paradata, BUFSIZ );
    if( keywords < 0 ){
        /*パラメータ無い時は動作終了*/
        exit(0);
    }
    /*パラメータ値の取得*/
    for(i=0;i<keywords;i++){
        if( strcmp( keyword[i], "sec" ) == 0 ){
            strcpy( gettext, paradata[i] );
        }
    }
    waitsec = strtol( gettext, (char**)NULL, 10 );
    if( waitsec <=0 ) waitsec = 1;
    /*永久ループ(1秒毎)*/
    status = 0;
    for(;;){
        if( status == 0 ) status = 1;
        else          status = 0;
        kcxobj_sndistat_tokcx( objid, status );
        sleep( waitsec );
    }
}
/* FILE_END */
/*KCX-PARA OK
#!v3.0 KaracrixBuilder.
#<< 制御プログラム 変数 >>
# no(i),"key(c)","data(c)",web(i),keitai(i),"prefix(c)","suffix(c)"
#
0,"sec","1",1,1,"待ち時間","単位秒です"
#
KCX-PARA*/

```

パラメータ定義エリア

18.3 制御&帳票パラメータファイル

パラメータファイルは、プログラムから使用するパラメータを定義したリソースファイルです。

パラメータは、制御パラメータの場合も帳票パラメータの場合も扱いが同じですので、本章では制御パラメータを使用した場合の例で解説します。

制御パラメータは、制御パラメータ付きプログラムをインポートすることで取り込むことができますが、パラメータファイル単独でもインポート、エクスポートが可能です。この機能により、プログラム本体とパラメータ部を別個に取り込むことが可能です。

18.3.1 制御&帳票パラメータファイルのインポート

①“/tmp”に取り込む制御パラメータファイルを置きます。この時、拡張子を“*.pra”としてください。

②「制御プログラム登録」画面を表示して、制御パラメータをインポートしたい制御プログラムの“パラ”欄を選択して「制御パラメータ登録」画面を表示します。

この時、“パラ”欄に“0”が表示されている場合には、そのプログラムにはパラメータが未登録であることを示しています。この場合は、③の手順へ進んで下さい。



No.	ObjID	プログラム名	プログラム編集	パラ	実行	MS	MB
1	ct101	ブートプログラム(他のプログラムを起動)	src 2008/04/6 17:04 obj 2008/04/6 17:04	0	実行	*	-
2	ct102	制御プログラム1	src 2008/04/6 17:01 obj / /	0		*	-
3	ct103						
4	ct104						
5	ct105						
6	ct106						
7	ct107						
8	ct108						
9	ct109						
10	ct110						

図 18.3.1 制御パラメータが未登録の例

“パラ”欄に“1”以上の数字が表示されている場合には、プログラムに既存のパラメータが登録されていることを示しており、そのままではインポートできません。



No.	ObjID	プログラム名	プログラム編集	パラ	実行	MS	MB
1	ct101	ブートプログラム(他のプログラムを起動)	src 2008/04/6 17:04 obj 2008/04/6 17:04	0	実行	*	-
2	ct102	制御プログラム1	src 2008/04/6 17:01 obj / /	8		*	-
3	ct103						
4	ct104						

図 18.3.2 制御パラメータが既登録の例

この場合は、“パラ”欄を選択して「制御パラメータ登録」画面を表示して画面消去(CLEAR)ボタンを押すと確認ダイアログが表示されますので「YES」を選択して既存のパラメータを一度消去して下さい。



図 18.3.3 既存のパラメータの削除

既存パラメータを消去すると、インポート(GET)ボタンが有効になり選択できるようになります。



図 18.3.4 既存のパラメータの削除後のインポートボタン

これで、③の手順へ進んで下さい。

- ③インポート(GET)ボタンを押すと、「外部データを取り込みますか？」という確認ダイアログが表示されますので「YES」を選択して下さい。



図 18.3.5 外部ファイルのインポート確認ダイアログ

注) データファイルが、存在しない場合は、以下のメッセージが表示されます。
(/tmp ディレクトリに、“*.pra”ファイルが存在しない場合)



- ④インポートファイルが置かれたディレクトリ上にある制御パラメータファイルのリストが表示されます。



図 18.3.6 外部ファイルの一覧表示

- ⑤インポートする制御パラメータファイルを選択します。インポートされると図のように制御パラメータが表示されます。



No.	キーワード	定義	入力画面: キーワード説明	入力画面: 備考	MS	PR
1	timeout	000	通信タイムアウト		-	-
2	devicno	kanocrix0	装置名		-	-
3	mail_port	00	メール送信許可	on,off	-	-
4	mail_address1	abc@nippou.xx	メールアドレス1	通知先	-	-
5	mail_address2	xyz@japan.yy	メールアドレス2	通知先	-	-
6	hi_limit	9999	上限値		-	-
7						
8						
9						
10						
11						
12						
13						

図 18.3.7 インポートされた制御パラメータを表示

- ⑥“ENT”ボタンを押して「制御プログラム登録」画面に戻ると、“パラ”欄にインポートされた制御パラメータの個数が表示されます。“CAN”ボタンを押した場合は、インポート(画面編集)はキャンセルされます。



No.	OBJID	プログラム名	プログラム編集	パラ	実行	MS	PR
1	ct101	ブートプログラム(他のプログラムを起動)	src 2008/09/4 17:04 obj 2008/09/4 17:04	0	実行	*	-
2	ct102	制御プログラム1	src 2008/09/6 11:01 obj / / :	6		*	-
3	ct103						
4	ct104						
5	ct105						
6	ct106						
7	ct107						
8	ct108						
9	ct109						
10	ct110						

図 18.3.8 インポート直後の管理画面表示

- ⑦インポートが正常に行われた後は、①でディレクトリ(/tmp)に置いたファイルは削除してかまいません。

18.3.2 制御&帳票パラメータファイルのエクスポート

制御&帳票パラメータを使用している制御&帳票プログラムのパラメータのみをエクスポートする場合には、以下の手順で行ないます。

ファイルが作成される場所はデフォルトで“/tmp”です。拡張子は、“*.pra”になります。

- ①「制御プログラム登録」画面を表示して、制御パラメータをエクスポートしたい制御プログラムの“パラ”欄を選択して「制御パラメータ登録」画面を表示します。



図 18.3.9 制御パラメータのインポート

- ②エクスポート(PUT)ボタンを押すと、以下のように、エクスポートファイル名の「文字入力」ダイアログが表示されます。



図 18.3.10 エクスポートファイル名の編集

- ③ファイル名には、デフォルトの名前(プログラムの OBJID)が設定されていますので、変更する場合にはこの画面で編集することができます。 (“pra”の拡張子を入力する必要はありません)

“ENT”ボタンを押すと入力された名前に“pra”の拡張子が付加されてファイルが作成出力されます。

18.4 ポイント汎用属性名称のインポート

制御プログラムは、制御に使用するグローバル的なパラメータとして“制御パラメータ”を用いることができますが、ポイント毎の個別に使用するパラメータを使用する場合には、ポイントの汎用属性値を用います。しかし、汎用属性の名称は、デフォルトでは“整数 1～8”、“実数 1～8”、“文字 1～4”という名称になっておりデータを設定する時に判かりにくいいため、この名称を使用するパラメータの意味に合わせて変更する必要があります。

この汎用属性の名称を変更する手段として、制御プログラムリソースファイルに、ポイント汎用属性名称を交換させる事を目的に付加させることで行います。但し現在のところ、本仕組みはインポートのみ有効で、画面操作によるエクスポートの機能はありません。制御プログラムに付加されている汎用属性名称の文字列をシステムにインポートする時に使用できます。従って、汎用属性名称を制御プログラムに付加させる場合、テキストエディタ等を使用して手動で予め編集しておく必要があります。

ポイント汎用属性名称は、ファイルとして\$KARACRIX/usr/ustbl/のディレクトリ内に分離されてインポートされます。例えば、属性名称テーブル名(後述)が“ABC123”の場合、kcxitem_objmem.ABC123.tbl というファイル名でディレクトリに移されます。 (\$KARACRIX/usr/ustbl/ kcxitem_objmem.ABC123.tbl)

インポートした属性名称テーブル内の名称をポイント汎用属性名称に割り付けるには、以下の関数を使用してポイントごとに、プログラムによって使用する属性テーブル名を設定します。

・kcxobj_item_table_cwt(objid, "ABC123"); (「23 章 KCX ライブラリファレンス」参照)

なおデフォルトの状態に戻すときは属性名称テーブル名に空("")を下記のように指定します。

・kcxobj_item_table_cwt(objid, "");

(1) リソースの構成

制御パラメータの時と同様に、制御プログラム部、制御パラメータ部の下方にポイント汎用属性名称部を置き、区切りマークを挿入します。区切りマークは、ポイント汎用属性名称部の上下両端に以下のようなテキスト行を付け加えたものです。

制御プログラム部

/*KCX-PARA OK

制御パラメータ部

KCX-PARA*/

/*KCX-ITEM OK 属性名称テーブル名 ← 追加行 (左端スペース不可、OK 間の区切りはスペース)

ポイント汎用属性名称部

KCX-ITEM*/

← 追加行 (左端スペース不可)

1. 属性名称テーブル名 属性名称テーブル名は、英数文字(8文字以内)で記述します。

2. ポイント汎用属性名称部 以下に示すデータ並びで記述します。

フォーマット

1 行目	①ポイント種別指定
2 行目以降	②設定番号,00000,③汎用属性指定,④汎用属性型指定,⑤設定値,⑥設定項目,⑦説明
最終行	end

①ポイント種別指定

汎用属性名称はポイント種別毎に、まとめて記述します。属性名称テーブルが割り付けられたポイントの属性表示は、そのポイント種別に対応する属性名称のものに置き換わります。

なお、同じポイント種別で異なる属性名称を指定したい場合には、属性名称テーブルを分ける必要があります。

KcxOBJtp_POINT_DI:	デジタル入力ポイント用定義
KcxOBJtp_POINT_DO:	デジタル出力ポイント用定義
KcxOBJtp_POINT_PI:	カウンタ入力ポイント用定義
KcxOBJtp_POINT_AI:	アナログ入力ポイント用定義
KcxOBJtp_POINT_AO:	アナログ出力ポイント用定義

②設定番号

1 から連番で指定します。(番号右の 00000 は将来予約のマスクコードとなっています)

③汎用属性指定

KcxTBLtp_OBJPOINT_AttButInt?Val:	整数属性 1～8 (? : 0～7)
KcxTBLtp_OBJPOINT_AttButFlt?Val:	実数属性 1～8 (? : 0～7)
KcxTBLtp_OBJPOINT_AttButText?Val:	文字属性 1～4 (? : 0～3)

④汎用属性型指定 予備(将来対応:属性の設定値)

⑤設定値 予備(将来対応:属性の設定値)

両設定は将来対応で予約されています。現在は下記に示すよう固定されています。

整数値:	i,0
実数値:	f,0.0
文字:	c,""

⑥設定項目 汎用属性の名称を記述します。

⑦説明 汎用属性の説明を記述します。

以下に、実際に使用しているサンプルを紹介します。

属性名称テーブルを新規に作成する場合には、このサンプルをたたき台に⑥設定項目と⑦説明の部分を編集し下記”kcxsamp”という属性名称テーブル名と競合しない別の名称に変更して使用して下さい。

```

/*KCX-ITEM OK kcxsamp
KcxOBJtp_POINT_DI
  1,00000,KcxTBLtp.OBJPOINT_AttButInt0Val,i,0, "メール送信先 1","メールアドレスが格納されている番地番号"
  2,00000,KcxTBLtp.OBJPOINT_AttButInt1Val,i,0, "メール送信先 2","メールアドレスが格納されている番地番号"
  3,00000,KcxTBLtp.OBJPOINT_AttButInt2Val,i,0, "メール送信先 3","メールアドレスが格納されている番地番号"
  4,00000,KcxTBLtp.OBJPOINT_AttButInt3Val,i,0, "整数 属性 4", ""
  5,00000,KcxTBLtp.OBJPOINT_AttButInt4Val,i,0, "整数 属性 5", ""
  6,00000,KcxTBLtp.OBJPOINT_AttButInt5Val,i,0, "整数 属性 6", ""
  7,00000,KcxTBLtp.OBJPOINT_AttButInt6Val,i,0, "整数 属性 7", ""
  8,00000,KcxTBLtp.OBJPOINT_AttButInt7Val,i,0, "整数 属性 8", ""
  9,00000,KcxTBLtp.OBJPOINT_AttButFlt0Val,f,0.0, "実数 属性 1", ""
  10,00000,KcxTBLtp.OBJPOINT_AttButFlt1Val,f,0.0, "実数 属性 2", ""
  11,00000,KcxTBLtp.OBJPOINT_AttButFlt2Val,f,0.0, "実数 属性 3", ""
  12,00000,KcxTBLtp.OBJPOINT_AttButFlt3Val,f,0.0, "実数 属性 4", ""
  13,00000,KcxTBLtp.OBJPOINT_AttButFlt4Val,f,0.0, "実数 属性 5", ""
  14,00000,KcxTBLtp.OBJPOINT_AttButFlt5Val,f,0.0, "実数 属性 6", ""
  15,00000,KcxTBLtp.OBJPOINT_AttButFlt6Val,f,0.0, "実数 属性 7", ""
  16,00000,KcxTBLtp.OBJPOINT_AttButFlt7Val,f,0.0, "実数 属性 8", ""
  17,00000,KcxTBLtp.OBJPOINT_AttButText0Val,c,"", "メール送信文", "メール補足文を書き入れる"
  18,00000,KcxTBLtp.OBJPOINT_AttButText1Val,c,"", "文字 属性 2", ""
  19,00000,KcxTBLtp.OBJPOINT_AttButText2Val,c,"", "文字 属性 3", ""
  20,00000,KcxTBLtp.OBJPOINT_AttButText3Val,c,"", "文字 属性 4", ""
end
KcxOBJtp_POINT_DO
  1,00000,KcxTBLtp.OBJPOINT_AttButInt0Val,i,0, "整数 属性 1", ""
  2,00000,KcxTBLtp.OBJPOINT_AttButInt1Val,i,0, "整数 属性 2", ""
  3,00000,KcxTBLtp.OBJPOINT_AttButInt2Val,i,0, "整数 属性 3", ""
  4,00000,KcxTBLtp.OBJPOINT_AttButInt3Val,i,0, "整数 属性 4", ""
  5,00000,KcxTBLtp.OBJPOINT_AttButInt4Val,i,0, "整数 属性 5", ""
  6,00000,KcxTBLtp.OBJPOINT_AttButInt5Val,i,0, "整数 属性 6", ""
  7,00000,KcxTBLtp.OBJPOINT_AttButInt6Val,i,0, "整数 属性 7", ""
  8,00000,KcxTBLtp.OBJPOINT_AttButInt7Val,i,0, "整数 属性 8", ""
  9,00000,KcxTBLtp.OBJPOINT_AttButFlt0Val,f,0.0, "実数 属性 1", ""
  10,00000,KcxTBLtp.OBJPOINT_AttButFlt1Val,f,0.0, "実数 属性 2", ""
  11,00000,KcxTBLtp.OBJPOINT_AttButFlt2Val,f,0.0, "実数 属性 3", ""
  12,00000,KcxTBLtp.OBJPOINT_AttButFlt3Val,f,0.0, "実数 属性 4", ""
  13,00000,KcxTBLtp.OBJPOINT_AttButFlt4Val,f,0.0, "実数 属性 5", ""
  14,00000,KcxTBLtp.OBJPOINT_AttButFlt5Val,f,0.0, "実数 属性 6", ""
  15,00000,KcxTBLtp.OBJPOINT_AttButFlt6Val,f,0.0, "実数 属性 7", ""
  16,00000,KcxTBLtp.OBJPOINT_AttButFlt7Val,f,0.0, "実数 属性 8", ""
  17,00000,KcxTBLtp.OBJPOINT_AttButText0Val,c,"", "文字 属性 1", ""
  18,00000,KcxTBLtp.OBJPOINT_AttButText1Val,c,"", "文字 属性 2", ""
  19,00000,KcxTBLtp.OBJPOINT_AttButText2Val,c,"", "文字 属性 3", ""
  20,00000,KcxTBLtp.OBJPOINT_AttButText3Val,c,"", "文字 属性 4", ""
end
KcxOBJtp_POINT_PI
  1,00000,KcxTBLtp.OBJPOINT_AttButInt0Val,i,0, "整数 属性 1", ""
  2,00000,KcxTBLtp.OBJPOINT_AttButInt1Val,i,0, "整数 属性 2", ""
  3,00000,KcxTBLtp.OBJPOINT_AttButInt2Val,i,0, "整数 属性 3", ""
  4,00000,KcxTBLtp.OBJPOINT_AttButInt3Val,i,0, "整数 属性 4", ""
  5,00000,KcxTBLtp.OBJPOINT_AttButInt4Val,i,0, "整数 属性 5", ""
  6,00000,KcxTBLtp.OBJPOINT_AttButInt5Val,i,0, "整数 属性 6", ""
  7,00000,KcxTBLtp.OBJPOINT_AttButInt6Val,i,0, "整数 属性 7", ""
  8,00000,KcxTBLtp.OBJPOINT_AttButInt7Val,i,0, "整数 属性 8", ""

```

```

    9,00000,KcxTBLtp_OBJPOINT_AttButFlt0Val,f,0.0, "実数 属性 1", ""
    10,00000,KcxTBLtp_OBJPOINT_AttButFlt1Val,f,0.0, "実数 属性 2", ""
    11,00000,KcxTBLtp_OBJPOINT_AttButFlt2Val,f,0.0, "実数 属性 3", ""
    12,00000,KcxTBLtp_OBJPOINT_AttButFlt3Val,f,0.0, "実数 属性 4", ""
    13,00000,KcxTBLtp_OBJPOINT_AttButFlt4Val,f,0.0, "実数 属性 5", ""
    14,00000,KcxTBLtp_OBJPOINT_AttButFlt5Val,f,0.0, "実数 属性 6", ""
    15,00000,KcxTBLtp_OBJPOINT_AttButFlt6Val,f,0.0, "実数 属性 7", ""
    16,00000,KcxTBLtp_OBJPOINT_AttButFlt7Val,f,0.0, "実数 属性 8", ""
    17,00000,KcxTBLtp_OBJPOINT_AttButText0Val,c,"", "文字 属性 1", ""
    18,00000,KcxTBLtp_OBJPOINT_AttButText1Val,c,"", "文字 属性 2", ""
    19,00000,KcxTBLtp_OBJPOINT_AttButText2Val,c,"", "文字 属性 3", ""
    20,00000,KcxTBLtp_OBJPOINT_AttButText3Val,c,"", "文字 属性 4", ""
end
KcxOBJtp_POINT_AI
    1,00000,KcxTBLtp_OBJPOINT_AttButInt0Val,i,0, "メール送信先 1", "メールアドレスが格納されている番地番号"
    2,00000,KcxTBLtp_OBJPOINT_AttButInt1Val,i,0, "メール送信先 2", "メールアドレスが格納されている番地番号"
    3,00000,KcxTBLtp_OBJPOINT_AttButInt2Val,i,0, "メール送信先 3", "メールアドレスが格納されている番地番号"
    4,00000,KcxTBLtp_OBJPOINT_AttButInt3Val,i,0, "整数 属性 4", ""
    5,00000,KcxTBLtp_OBJPOINT_AttButInt4Val,i,0, "整数 属性 5", ""
    6,00000,KcxTBLtp_OBJPOINT_AttButInt5Val,i,0, "0=スケール演算 1=ABC演算", "AD正規化値を単位系に変換する方法の選択"
    7,00000,KcxTBLtp_OBJPOINT_AttButInt6Val,i,0, "整数 属性 7", ""
    8,00000,KcxTBLtp_OBJPOINT_AttButInt7Val,i,0, "整数 属性 8", ""
    9,00000,KcxTBLtp_OBJPOINT_AttButFlt0Val,f,0.0, "A:Y=[A*(X+B)+C]", "AD正規化 X(1-0)値を単位系に変換する計算値"
    10,00000,KcxTBLtp_OBJPOINT_AttButFlt1Val,f,0.0, "B:Y=[A*(X+B)+C]", "AD正規化 X(1-0)値を単位系に変換する計算値"
    11,00000,KcxTBLtp_OBJPOINT_AttButFlt2Val,f,0.0, "C:Y=[A*(X+B)+C]", "AD正規化 X(1-0)値を単位系に変換する計算値"
    12,00000,KcxTBLtp_OBJPOINT_AttButFlt3Val,f,0.0, "実数 属性 4", ""
    13,00000,KcxTBLtp_OBJPOINT_AttButFlt4Val,f,0.0, "実数 属性 5", ""
    14,00000,KcxTBLtp_OBJPOINT_AttButFlt5Val,f,0.0, "実数 属性 6", ""
    15,00000,KcxTBLtp_OBJPOINT_AttButFlt6Val,f,0.0, "実数 属性 7", ""
    16,00000,KcxTBLtp_OBJPOINT_AttButFlt7Val,f,0.0, "実数 属性 8", ""
    17,00000,KcxTBLtp_OBJPOINT_AttButText0Val,c,"", "メール送信文", "メール補足文を書き入れる"
    18,00000,KcxTBLtp_OBJPOINT_AttButText1Val,c,"", "文字 属性 2", ""
    19,00000,KcxTBLtp_OBJPOINT_AttButText2Val,c,"", "文字 属性 3", ""
    20,00000,KcxTBLtp_OBJPOINT_AttButText3Val,c,"", "文字 属性 4", ""
end
KcxOBJtp_POINT_AO
    1,00000,KcxTBLtp_OBJPOINT_AttButInt0Val,i,0, "整数 属性 1", ""
    2,00000,KcxTBLtp_OBJPOINT_AttButInt1Val,i,0, "整数 属性 2", ""
    3,00000,KcxTBLtp_OBJPOINT_AttButInt2Val,i,0, "整数 属性 3", ""
    4,00000,KcxTBLtp_OBJPOINT_AttButInt3Val,i,0, "整数 属性 4", ""
    5,00000,KcxTBLtp_OBJPOINT_AttButInt4Val,i,0, "整数 属性 5", ""
    6,00000,KcxTBLtp_OBJPOINT_AttButInt5Val,i,0, "整数 属性 6", ""
    7,00000,KcxTBLtp_OBJPOINT_AttButInt6Val,i,0, "整数 属性 7", ""
    8,00000,KcxTBLtp_OBJPOINT_AttButInt7Val,i,0, "整数 属性 8", ""
    9,00000,KcxTBLtp_OBJPOINT_AttButFlt0Val,f,0.0, "実数 属性 1", ""
    10,00000,KcxTBLtp_OBJPOINT_AttButFlt1Val,f,0.0, "実数 属性 2", ""
    11,00000,KcxTBLtp_OBJPOINT_AttButFlt2Val,f,0.0, "実数 属性 3", ""
    12,00000,KcxTBLtp_OBJPOINT_AttButFlt3Val,f,0.0, "実数 属性 4", ""
    13,00000,KcxTBLtp_OBJPOINT_AttButFlt4Val,f,0.0, "実数 属性 5", ""
    14,00000,KcxTBLtp_OBJPOINT_AttButFlt5Val,f,0.0, "実数 属性 6", ""
    15,00000,KcxTBLtp_OBJPOINT_AttButFlt6Val,f,0.0, "実数 属性 7", ""
    16,00000,KcxTBLtp_OBJPOINT_AttButFlt7Val,f,0.0, "実数 属性 8", ""
    17,00000,KcxTBLtp_OBJPOINT_AttButText0Val,c,"", "文字 属性 1", ""
    18,00000,KcxTBLtp_OBJPOINT_AttButText1Val,c,"", "文字 属性 2", ""
    19,00000,KcxTBLtp_OBJPOINT_AttButText2Val,c,"", "文字 属性 3", ""
    20,00000,KcxTBLtp_OBJPOINT_AttButText3Val,c,"", "文字 属性 4", ""
end
KCX-ITEM*/

```

(2) 実際のリソースファイル内容例

参考のためリソースファイルをテキストエディタで表示した場合の例を以下に示します。

プログラム部、制御パラメータ部、ポイント汎用属性名称部が確認できます。

※下記プログラム部、パラメータ部、ポイント汎用属性名称部の内容は参考例のため特に意味はありません。

```
#include <karacrix.h>
main(argc,argv)
int   argc;
char  *argv[];
{
    kcxinit( argc, argv );
}
/*KCX-PARA  OK
0,"perm","on",1,1,"メール許可","on/off"
1,"ip","RT 192.168.0.200 20000",1,0,"装置","型"
KCX-PARA*/
/*KCX-ITEM  OK   abc123
KcxOBJtp_POINT_DI
    1,00000,KcxTBLtp_OBJPOINT_AttButInt0Val,i,0,"メール送信先 1","メールアドレスが格納されている番地番号"
    2,00000,KcxTBLtp_OBJPOINT_AttButText0Val,c,"","メール送信文","メール補足文を書き入れる"
end
KCX-ITEM*/
```

(3) プログラムでの使用例

以下のサンプルプログラムは、デジタルポイント(di001)の汎用属性名称をインポートするプログラムの例です。手順を以下に示します。

- ①このプログラムを「制御プログラム登録」画面でインポートすると、制御プログラムの内容と同時に、“abc123”の属性名称テーブル名でポイント汎用属性名称ファイル(kcxitem_objmem.abc123.tbl)も分離インポートされます。 (\$KARACRIX/usr/ustbl/ kcxitem_objmem.abc123.tbl)
- ②この制御プログラムをコンパイルして実行すると、kcxobj_item_table_cwt 関数に指定した“di001”ポイントの汎用属性名称に、“abc123”属性名称テーブル名で指定されるポイント汎用属性名称ファイルから⑤設定項目⑥説明を読み込んで画面表示するようになります。

```
#include <karacrix.h>

main( argc, argv )
int  argc;
char *argv[];
{
    int objid;

    /* KCX ライブラリの初期化 */
    kcxinit( argc, argv );

    /* ポイントを開く */
    objid = kcxobj_open( "di001" );

    /* ポイント(di001)の汎用属性名称テーブル名の割り付け */
    kcxobj_item_table_cwt( objid, "abc123" );

}

/*KCX-ITEM OK abc123
KcxOBJtp_POINT_DI
    1,00000,KcxTBLtp_OBJPOINT_AttButInt0Val,i,0,    "メール送信先 1","メールアドレスが格納されている番地番号"
    2,00000,KcxTBLtp_OBJPOINT_AttButFlt0Val,f,31.5, "監視上限値","上限値を超えたらメール送信"
    3,00000,KcxTBLtp_OBJPOINT_AttButText0Val,c,"",  "メール送信文", "メール補足文を書き入れる"
end
KCX-ITEM*/
```

読込まれた、“di001”の汎用属性設定の表示を以下に示します。

No.	汎用属性名	属性値	単位
1	メール送信先 1	メールアドレスが格納されている番地番号	0
2	整数 属性 2		0
3	整数 属性 3		0
4	整数 属性 4		0
5	整数 属性 5		0
6	整数 属性 6		0
7	整数 属性 7	「オブジェクト動的変更」画面非表示項目	0
8	整数 属性 8	「オブジェクト動的変更」画面非表示項目	0
9	監視上限値	上限値を超えたらメール送信	31.5
10	実数 属性 2		0.0
11	実数 属性 3		0.0
12	実数 属性 4		0.0

図 18.4.1 インポートされたポイント汎用属性名称 (di001)

18.4 ポイント汎用属性名称のインポート

13	実数 属性 5		△ 0
14	実数 属性 6		△ 0
15	実数 属性 7	「オブジェクト動的変更」画面非表示項目	△ 0
16	実数 属性 8	「オブジェクト動的変更」画面非表示項目	△ 0
③	メール送信文	メール送信文を書き入れる	□
18	文字 属性 1		□
19	文字 属性 3	「オブジェクト動的変更」画面非表示項目	□
20	文字 属性 4	「オブジェクト動的変更」画面非表示項目	□
21			
22			
23			
24			

(1. 「ポイント属性」→「ポイント属性一時変更」)

(2. 「ポイント登録」→「ポイント属性設定」)

図 18.4.2 インポートされたポイント汎用属性名称 (コンソール画面)

OBJID: d001
ポイント名: 窓ゼンサリ

① 監視許可: off

② ポイント属性: 1: 0, 2: 1000, 3: 0.000, 4: 0.000, 5: 0.000, 6: 0.000

③ メール送信文: 文字 属性 2

(「ポイント」→「接点入力」→「ポイントの属性設定」)

図 18.4.3 インポートされたポイント汎用属性名称 (Web 画面)

18.5 監視パネル&複部品ファイル

監視パネルファイルは、監視パネル CAD で作成される静部品、動部品を配置したリソースファイルです。また、複部品は複数の部品を1つの部品として扱うために作成された部分的な監視パネルです。複部品ファイル構成は、監視パネルファイルと全く同じ構成になっています。

以下、監視パネルファイルを例に説明します。

18.5.1 監視パネル&複部品ファイルのインポート

- ①“/tmp”ディレクトリに取り込む監視パネルファイルを置きます。この時、拡張子を“*.mrs”としてください。
- ②「監視パネルメニュー」→「監視パネル管理」ボタンを押して「監視パネル管理」画面を表示して、何も登録されていないリスト上の“監視パネル名”欄を選択して下さい。文字入力ダイアログが表示されますので、任意の監視パネル名称を入力して ENT ボタンを押して下さい。

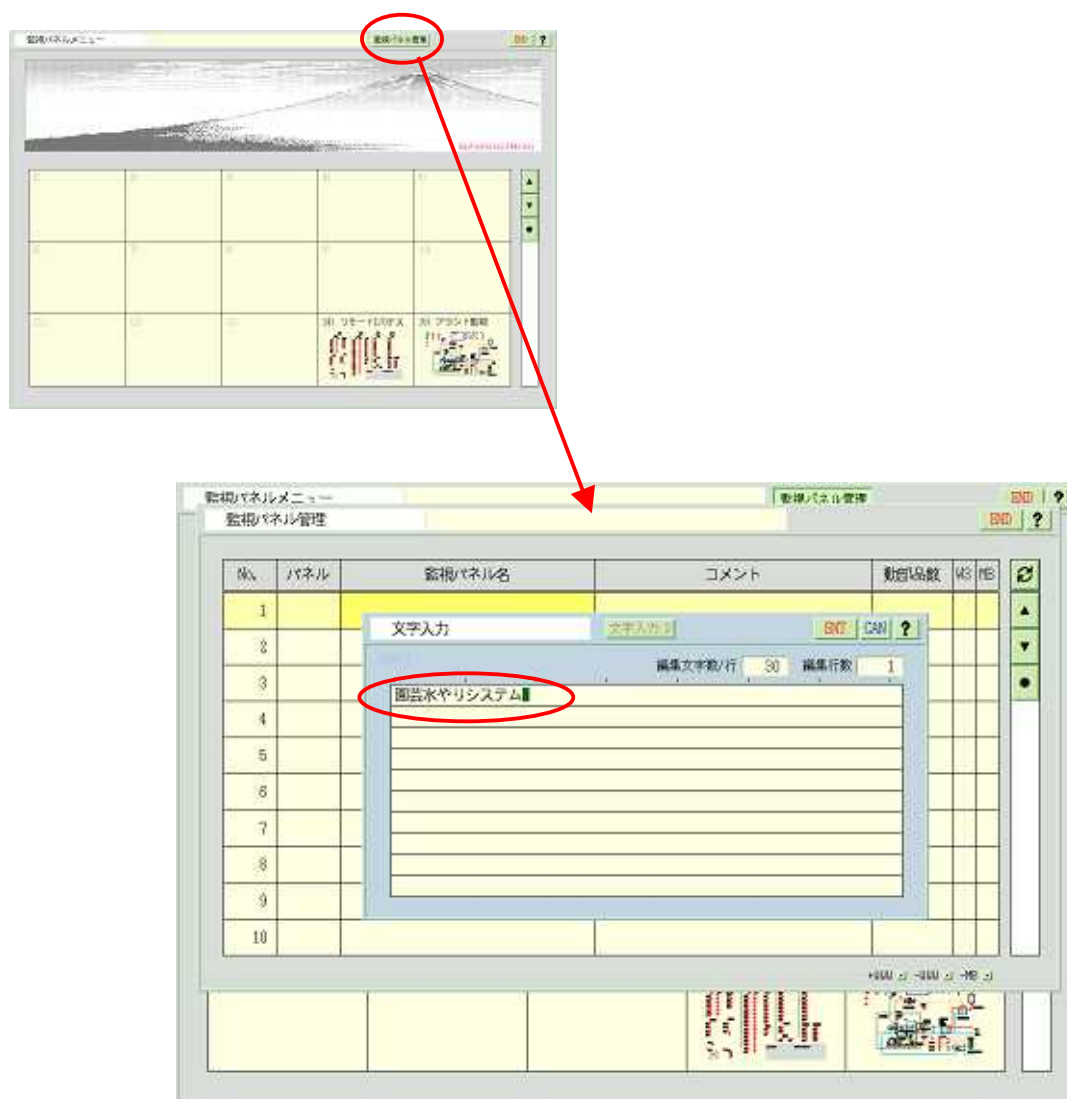


図 18.5.1 監視パネル名称の決定

- ③「外部データを取り込みますか?」という確認ダイアログが表示されますので「YES」を選択すると、インポートファイルが置かれたディレクトリ (/tmp) 上にある監視パネルファイルのリストが表示されます。



図 18.5.2 外部ファイルのインポート確認ダイアログ

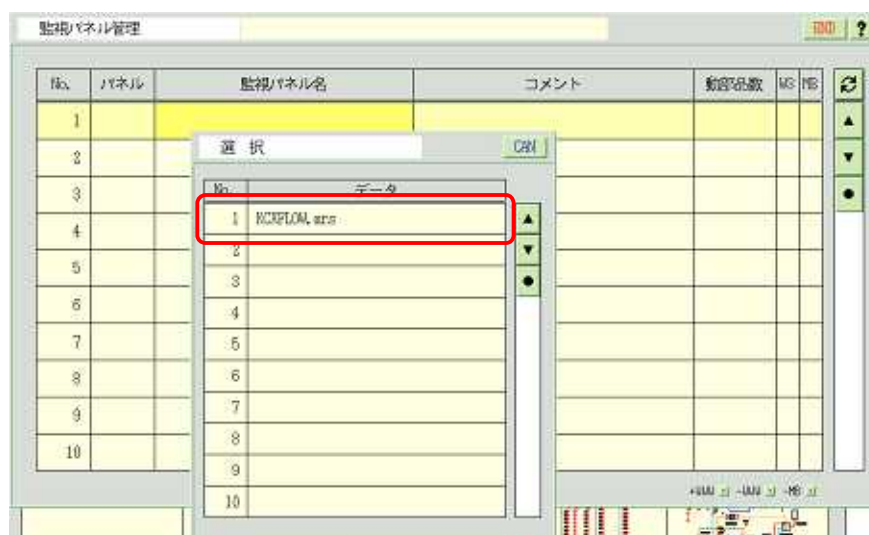


図 18.5.3 外部ファイルの一覧表示

- ④監視パネルファイルのリストから、インポートする監視パネルファイルを選択します。ここで、インポートされる監視パネルファイルの中でビットマップ画像ファイルが使用されている場合、次のように「ビットマップ書込指定」画面が表示されます。

このとき、KaracrixBuilder で管理しているビットマップファイルと同じ名前のビットマップファイルがある場合には、次のように「書込」欄に“上書拒否”がデフォルトで表示されます。

新規ビットマップ画像の場合には、“-”が表示されます。

「ビットマップ書込指定」画面について以下で解説します。



図 18.5.4 「ビットマップ書込指定」画面表示

- No.: グラフの登録番号です。
- グループ名: KaracrixBuilder で管理されているビットマップ画像のグループ名です。既存のビットマップ画像の場合は、変更することはできません。新規ビットマップ画像の場合は、ここでグループを変更することが出来ます。
- 仕様: 画像サイズと画像の一部が表示されます。(w:幅, h:高さ)
- 書込: -: 無条件にインポートします。(新規ビットマップ画像の場合)
上書拒否: 上書きを拒否します。(既存ビットマップ画像の場合)
上書: 上書きを許可します。(既存ビットマップ画像の場合)
- ☐ *OVR ボタン: 既存のビットマップを一括して上書きにします。
- ☐ *ORG ボタン: 既存のビットマップを一括して上書拒否にします。

ここで、インポートするビットマップファイルの扱いを選択することができます。

1. 新規ビットマップファイルの場合

“グループ名”欄には、デフォルトで作業用のグループが選択されていますが、次のようにグループ選択ダイアログを表示して変更することができますので、ここで、インポートするグループを選択して下さい。



図 18.5.5 「グループ」選択ダイアログ表示

2. 全ての既存ビットマップファイルの上書きを拒否する場合

デフォルトの設定では、“書込”欄が“上書拒否”に設定されています。“ENT”ボタンで終了して下さい。

3. 一部の既存ビットマップファイルの上書きを許可する場合

上書きをするビットマップの行の“書込”欄を選択して下さい。次のように「上書き (更新) しますか？」とメッセージがダイアログ画面に表示されますので、「ON」を選択して下さい。

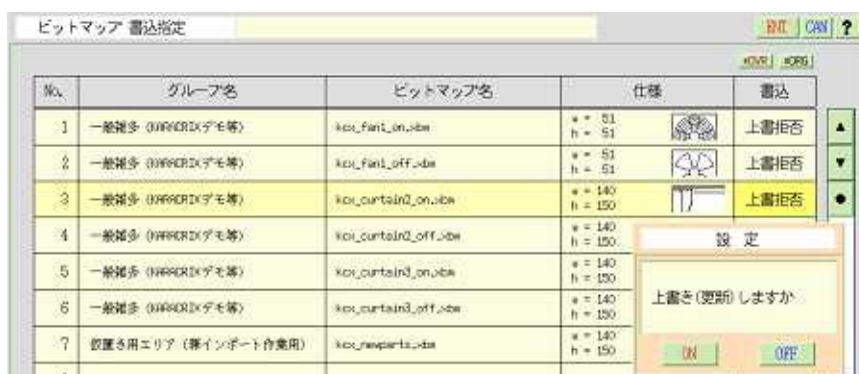


図 18.5.6 一部ビットマップの上書きを許可する場合

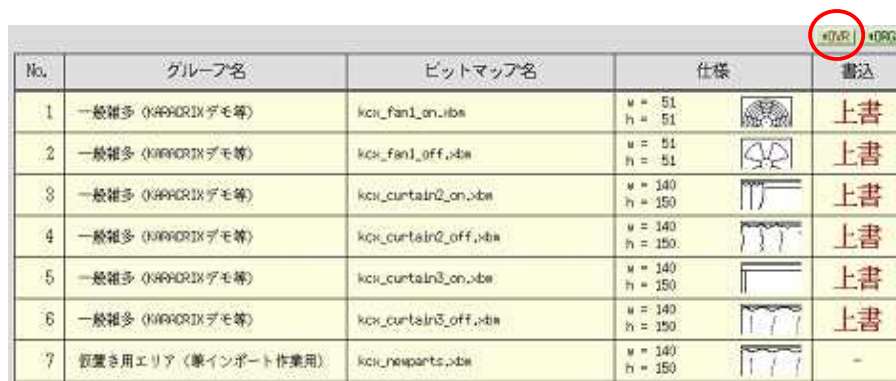
指定したビットマップファイルの“書込”欄が“上書”と表示されます。“ENT”ボタンで終了すると指定したビットマップファイルは上書きされてインポートされます。

3	一般雑多 (KPRORIXデモ等)	koi_curtain2_on.xbm	w = 140 h = 150	上書
---	-------------------	---------------------	--------------------	----

図 18.5.7 上書き許可表示

4. 全ての既存ビットマップファイルの上書きを許可する場合

“*OVR”ボタンを押して下さい。表示されている全ての既存ビットマップの“書込”欄に“上書”が表示されます。“ENT”ボタンで終了すると全てのビットマップファイルは上書きされてインポートされます。



No.	グループ名	ビットマップ名	仕様	書込
1	一般雑多 (KARACRIX デモ等)	kcx_fan1_on.xbm	w = 51 h = 51	上書
2	一般雑多 (KARACRIX デモ等)	kcx_fan1_off.xbm	w = 51 h = 51	上書
3	一般雑多 (KARACRIX デモ等)	kcx_curtain2_on.xbm	w = 140 h = 150	上書
4	一般雑多 (KARACRIX デモ等)	kcx_curtain2_off.xbm	w = 140 h = 150	上書
5	一般雑多 (KARACRIX デモ等)	kcx_curtain3_on.xbm	w = 140 h = 150	上書
6	一般雑多 (KARACRIX デモ等)	kcx_curtain3_off.xbm	w = 140 h = 150	上書
7	設置き用エリア (単インポート作業用)	kcx_newparts.xbm	w = 140 h = 150	-

図 18.5.8 既存ビットマップの一括上書き許可

- ⑤インポートされると図のような空のパネルアイコンと監視パネル名のみの画面表示になります。これは、監視パネルメニューで表示するアイコン画像がまだ作成されていないためです。“END”ボタンを押して「監視パネルメニュー」画面に戻して下さい。



No.	パネル	監視パネル名	コメント	動作保証	OK	NG
1		消防水やりシステム		?	*	*
2						
3						
4						
5						
6						
7						
8						
9						
10						

図 18.5.9 インポート直後の管理画面表示

- ⑥「監視パネルメニュー」画面に戻ると図のようなメニュー表示となっています。インポートした監視パネルファイルがアイコンメニューにまだ反映されませんので、これを反映させるために以下の作業を行なって下さい。

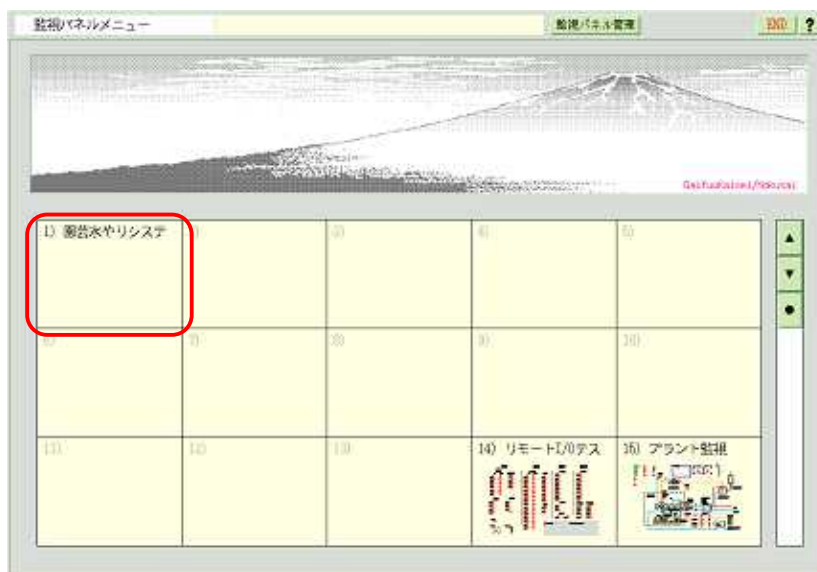


図 18.5.10 インポート直後のメニュー画面表示

- ⑦登録した“パネルメニュー”欄を選択して「監視パネル」画面を呼出し、さらに「監視パネル CAD」画面を呼出して下さい。監視パネル CAD が表示されたら“ENT”ボタンを押して「監視パネル」画面に戻ります。この時にアイコンイメージが CAD によって作られます。そして、さらに“END”ボタンで「監視パネルメニュー」画面に戻って下さい。アイコンメニューにインポートされたイメージ画面が反映されます。インポートが正常に行われた後は、①でディレクトリ(/tmp)に置いたファイルは削除してかまいません。



図 18.5.11 監視パネル CAD の呼出し

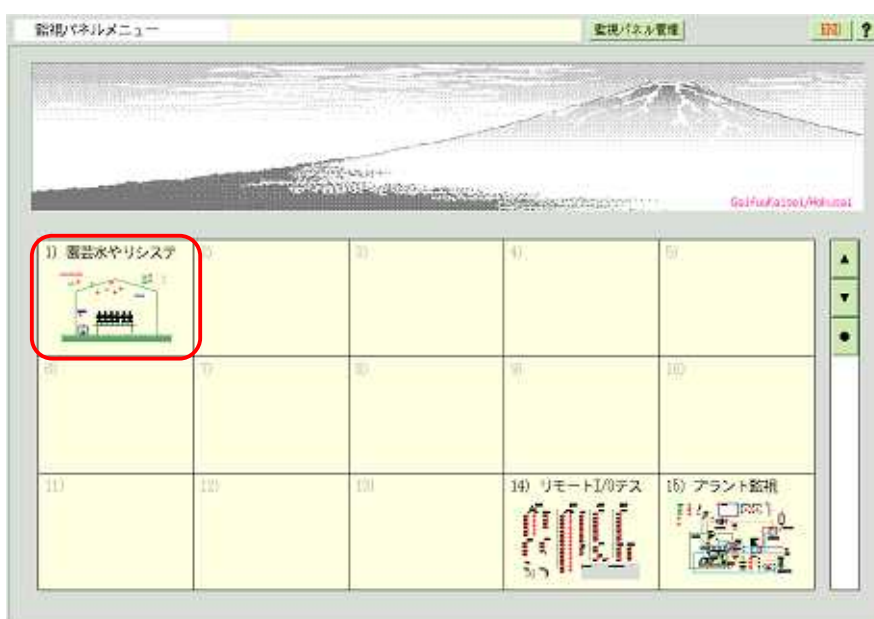


図 18.5.12 監視パネルメニューに反映された様子

18.5.2 監視パネルファイルのエクスポート

「監視パネルメニュー」画面に登録されている監視パネルをエクスポートする場合には、以下の手順で行ないます。

ファイルが作成される場所はデフォルトで“/tmp”です。拡張子は、“*.mrs”になります。

- ①「監視パネルメニュー」画面を表示して下さい。例として、下図 No.15 の「プラント監視」パネルをエクスポートしてみます。ここで、「監視パネル管理」ボタンを押して「監視パネル管理」画面を表示して下さい。

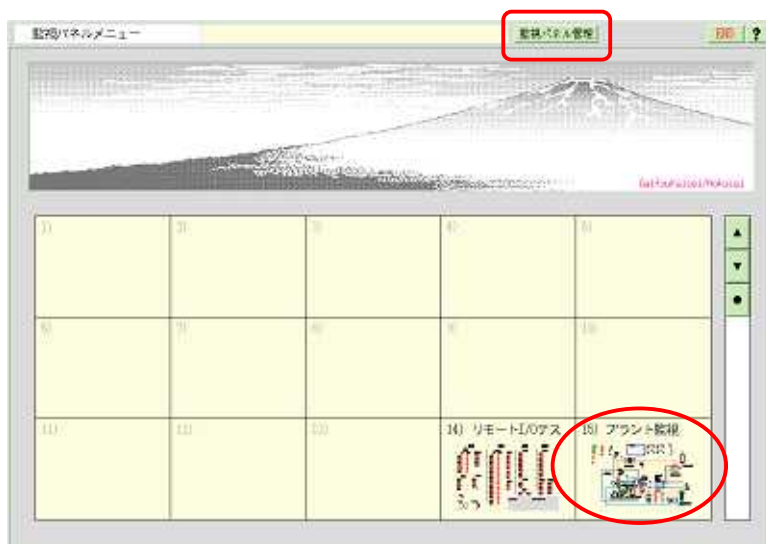


図 18.5.13 エクスポートする監視パネル

- ②リストスクロールボタンでエクスポートする監視パネルの登録行表示して下さい。(ここでは、No.15)



図 18.5.14 エクスポートする監視パネル登録行を表示

- ③ここで、ボタンを押して“PUT”のモードを選択して、エクスポートする監視パネルの登録行に対応するボタンを押して下さい。以下のように、エクスポートファイル名の「文字入力」ダイアログが表示されます。



図 18.5.15 エクスポートファイル名の編集

- ④ファイル名には、デフォルトの名前(自動生成された適当な名前)が設定されていますので、変更する場合にはこの画面で編集することができます。 (“.mrs”の拡張子を入力する必要はありません)
- “ENT”ボタンを押すと、ファイル名を変更した場合も変更しない場合にも、表示されている名前に “.mrs”の拡張子が付加されてファイルが作成出力されます。

18.6 ビットマップファイル

ビットマップファイルは、監視パネル CAD、帳票フォーマット CAD で使用する白黒画像ファイルリソースファイルです。

18.6.1 ビットマップファイルのインポート

- ①“/tmp”に取り込むビットマップファイルを置きます。この時、拡張子を“*.xbm”としてください。
- ②登録したい「ビットマップグループ」画面の“ビットマップ登録数”欄をクリックして「ビットマップ (絵部品) 一覧」画面を表示して、何も登録されていないリスト上の「ビットマップ名」欄を選択します。

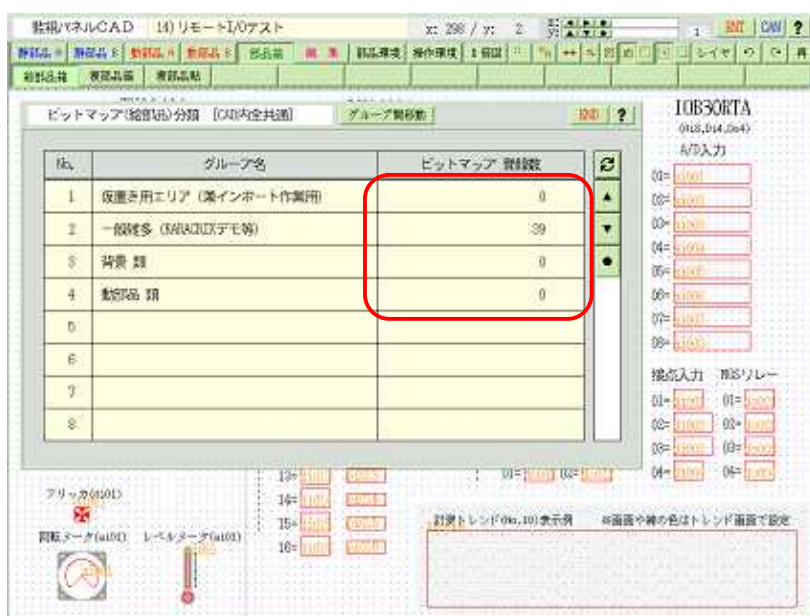


図 18.6.1 ビットマップグループ画面

- ③「文字入力」ダイアログが表示されますのでビットマップ名を入力してください。

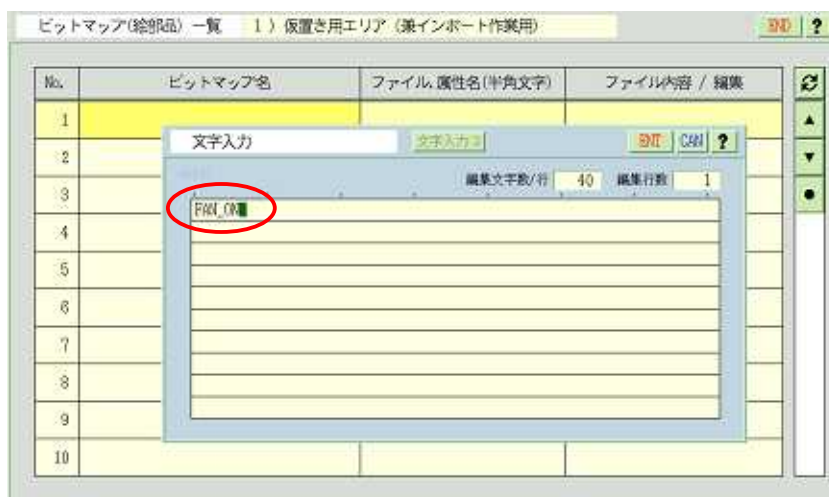


図 18.6.2 外部ファイルのインポート確認ダイアログ

- ④「ファイル.属性名(半角文字) 欄」をクリックすると「外部データを取り込みますか?」という確認ダイアログが表示されますので「YES」を選択すると、先ほど置いたディレクトリ(/tmp)上にあるビットマップファイルのリストが表示されます。



図 18.6.3 外部ファイルのインポート確認ダイアログ



図 18.6.4 ビットマップファイルの一覧表示

- ⑤ここで目的のファイルを選択するとビットマップファイルがインポートされます。



図 18.6.5 ビットマップファイルの一覧表示

⑥“ファイル内容/編集”欄を選択するとインポートされたビットマップイメージを確認することができます。

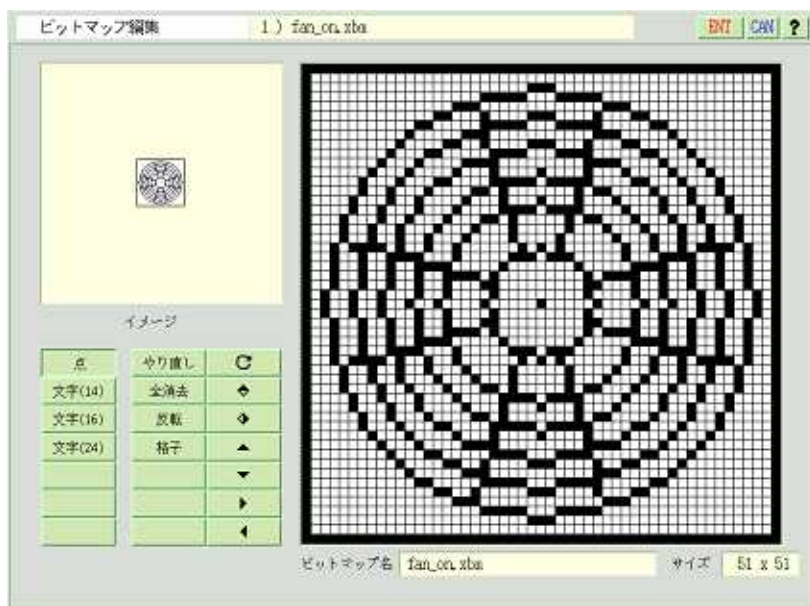


図 18.6.6 ビットマップイメージの表示

⑦インポートが正常に行われた後は、①でディレクトリに置いたファイル(/tmp)は削除してかまいません。

18.6.2 ビットマップファイルのエクスポート

「ビットマップリスト」画面に登録されているビットマップをエクスポートする場合には、以下の手順で行ないます。

ファイルが作成される場所はデフォルトで“/tmp”です。

なお、ビットマップファイルの場合に限り、拡張子は自動的に付加されませんのでご注意ください。

- ①「ビットマップリスト」画面を表示してリストスクロールボタンでエクスポートするビットマップの登録行を表示して下さい。次に、ボタンを押して“PUT”モードを選択して、エクスポートするビットマップの登録行に対応するボタンを押して下さい。以下のように、エクスポートファイル名の「文字入力」ダイアログが表示されます。



図 18.6.7 エクスポートファイル名の編集

- ②ファイル名には、デフォルトの名前が設定されていますが、再度インポートする目的でエクスポートする場合、名前を必ず変更しておいてください（同名のファイルをインポートすることができないため）。また、ビットマップファイルの場合、拡張子として“.xbm”を付加しておくことを推奨します。“ENT”ボタンを押すとファイルが作成出力されます。

18.7 帳票プログラムファイル

帳票プログラムファイルは、帳票印刷用プログラムのリソースファイルです。

18.7.1 帳票プログラムファイルのインポート

- ① “/tmp”に取り込む帳票プログラムファイルを置きます。この時、拡張子を“*.pgc”としてください。
- ② 「帳票プログラム登録」画面を表示して、何も登録されていないリスト上の“取込書式”欄を選択して「フォーマット選択」ダイアログで使用する帳票書式を選択します。



図 18.7.1 帳票フォーマットの選択

- ③ 次に“文書名”欄を選択します。「文字入力」ダイアログが表示されますので、任意の文書名を入力して ENT ボタンを押して下さい。



図 18.7.2 帳票プログラム名の登録

- ④「外部データを取り込みますか？」という確認ダイアログが表示されます。



図 18.7.3 外部ファイルのインポート確認ダイアログ

- ⑤ここで「YES」を選択すると、インポートファイルが置かれたディレクトリ(/tmp)上にある帳票プログラムファイルのリストが表示されます。



図 18.7.4 外部ファイルの一覧表示

- ⑥インポートする帳票プログラムファイルを選択します。ここで、インポートするプログラムにパラメータが添付されている場合には、「プログラムからパラメータを分離取り込みますか?」という確認ダイアログが表示されますので「YES」を選択すると、選択した帳票プログラムがインポートされ、添付されているパラメータ定義情報も同時にインポートされます。



図 18.7.5 パラメータ取込み確認ダイアログ

- ⑦インポートされると下図のような画面表示になります。

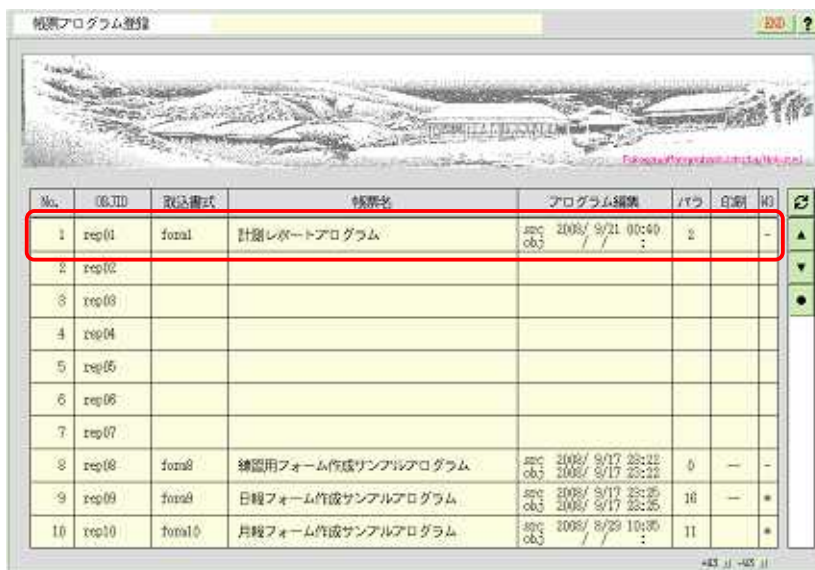


図 18.7.6 インポート直後の管理画面表示

⑧“プログラム編集”欄を選択すると編集画面でインポートされたプログラムを確認することができます。



図 18.7.7 インポートプログラムの表示

⑨また、パラメータも同時にインポートした場合には、“パラ”欄を選択するとパラメータ登録画面でインポートされたパラメータを確認することができます。



図 18.7.8 インポートパラメータの表示

⑩インポートが正常に行われた後は、①でディレクトリ(/tmp)に置いたファイルは削除してかまいません。

18.7.2 帳票プログラムファイルのエクスポート

「帳票プログラム登録」画面に登録されている帳票プログラムをエクスポートする場合には、以下の手順で行ないます。

ファイルが作成される場所はデフォルトで“/tmp”です。拡張子は、“*.pgc”になります。

- ①「帳票プログラム登録」画面を表示してエクスポートする帳票プログラムの登録行を表示して下さい。ここでは、下図 No.1 の「計測レポートプログラム」をエクスポートする例で説明します。

No.	OBJID	取込書式	帳票名	プログラム編集	パラ	印刷	MS
1	rep01	focal	計測レポートプログラム	apc obj 2008/ 9/21 00:00	2	—	—
2	rep02						
3	rep03						
4	rep04						
5	rep05						
6	rep06						
7	rep07						
8	rep08	form8	補償用フォーム作成サンプルプログラム	apc obj 2008/ 9/27 23:22	0	—	—
9	rep09	form9	日報フォーム作成サンプルプログラム	apc obj 2008/ 9/27 23:25	10	—	*
10	rep10	form10	月報フォーム作成サンプルプログラム	apc obj 2008/ 8/29 18:35	13	—	*

図 18.7.9 エクスポートする帳票プログラムを表示

- ②次に、 ボタンを数回押して“PUT”モードを選択して、エクスポートする帳票プログラムの登録行に対応する ボタンを押して下さい。以下のように、エクスポートファイル名の「文字入力」ダイアログが表示されます。

No.	OBJID	取込書式	帳票名	プログラム編集	パラ	印刷	MS
1	rep01	focal	計測レポートプログラム	apc obj 2008/ 9/21 00:00	2	—	—
2	rep02						
3	rep03						
4	rep04						
5	rep05						
6	rep06						
7	rep07						
8	rep08	form8	補償用フォーム作成サンプルプログラム	apc obj 2008/ 9/27 23:22	0	—	—
9	rep09	form9	日報フォーム作成サンプルプログラム	apc obj 2008/ 9/27 23:25	10	—	*
10	rep10	form10	月報フォーム作成サンプルプログラム	apc obj 2008/ 8/29 18:35	13	—	*

図 18.7.10 エクスポートファイル名の編集

③ファイル名には、デフォルトの名前(プログラムの OBJID)が設定されていますので、変更する場合にはこの画面で編集することができます。(".pgc"の拡張子を入力する必要はありません)

"ENT"ボタンを押すと、ファイル名を変更した場合も、変更しない場合にも、表示されている名前に".pgc"の拡張子が付加されてファイルが作成されます。

パラメータを使用していない帳票プログラムの場合以上でエクスポート作業は終了です。

パラメータを使用している帳票プログラムの場合は、以下の様に、「プログラムにパラメータを添付しますか?」と確認ダイアログが表示されます。ここで、「YES」を選択すると、パラメータを付加して帳票プログラムがエクスポートされます。



図 18.7.11 パラメータ添付の確認ダイアログ

18.8 帳票書式ファイル

帳票書式ファイルは、帳票フォーマット CAD で作成された帳票フォーマットリソースファイルです。

18.8.1 帳票書式ファイルのインポート

- ① “/tmp”に取り込む監視パネルファイルを置きます。この時、拡張子を“*.prs”としてください。
- ② 「帳票フォーマット登録」画面を表示して、何も登録されていないリスト上の“フォーマット名”欄を選択します。「文字入力」ダイアログが表示されますので、任意のフォーマット名を入力して ENT ボタンを押して下さい。



図 18.8.1 帳票書式名の登録

- ③ 「外部データを取り込みますか？」という確認ダイアログが表示されますので「YES」を選択すると、インポートファイルが置かれたディレクトリ上にある帳票書式ファイルのリストが表示されます。



図 18.8.2 外部ファイルのインポート確認ダイアログ

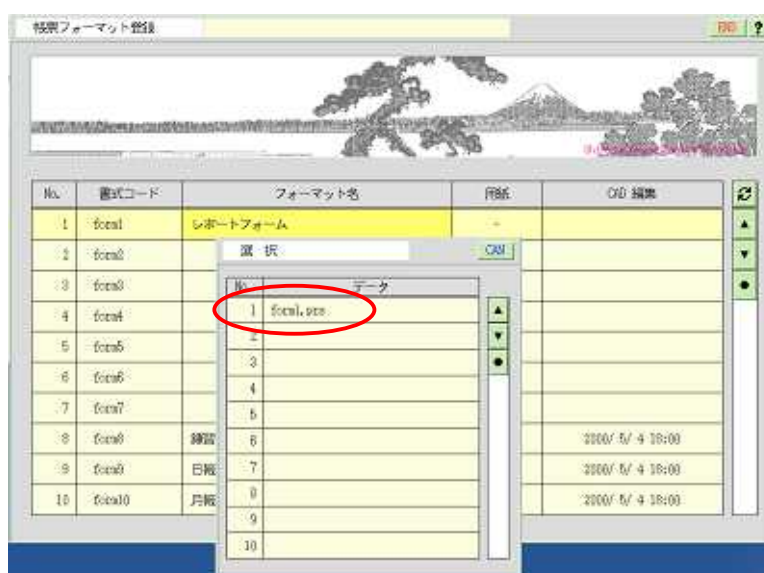


図 18.8.3 帳票書式ファイルの一覧表示

- ④インポートする帳票書式ファイルを選択すると、帳票プログラムで使用する原点情報ファイルを作成するために、「帳票 CAD」画面が表示されます。「帳票 CAD」画面で“ENT”ボタンを押してインポート作業を終了して下さい。

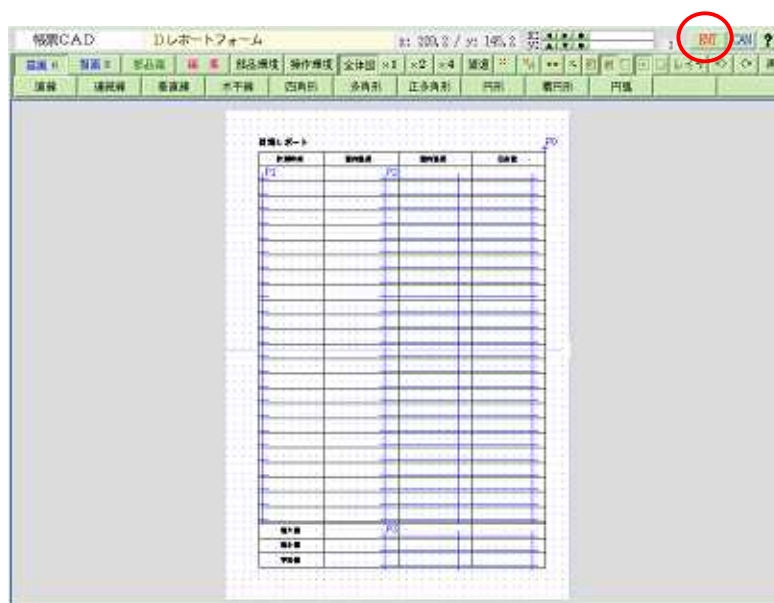


図 18.8.4 インポートした帳票書式を登録

- ⑤インポートされると図のような画面表示になります。“CAD 編集”欄にインポートされた日時が表示されます。



No.	書式コード	フォーマット名	用紙	CAD 編集
1	forma1	レポートフォーム	A4・たて	2008/ 9/23 01:28
2	forma2			
3	forma3			
4	forma4			
5	forma5			
6	forma6			
7	forma7			
8	forma8	雑誌用フォーム	A4・たて	2010/ 5/ 4 18:00
9	forma9	日報(サンパル)フォーム	A4・たて	2010/ 5/ 4 18:00
10	forma10	月報(サンパル)フォーム	A4・たて	2010/ 5/ 4 18:00

図 18.8.5 インポート直後の管理画面表示

- ⑥インポートが正常に行われた後は、①でディレクトリ(/tmp)に置いたファイルは削除して構いません。

18.8.2 帳票書式ファイルのエクスポート

「帳票フォーマット登録」画面に登録されている帳票書式をエクスポートする場合には、以下の手順で行ないます。

ファイルが作成される場所はデフォルトで“/tmp”です。拡張子は、“*.prs”になります。

- ①「帳票フォーマット登録」画面を表示してエクスポートする帳票書式の登録行を表示して下さい。ここでは、下図 No.1 の「レポートフォーム」書式をエクスポートする例で説明します。



No.	書式コード	フォーマット名	所属	作成 編集
1	form1	レポートフォーム	総・たて	2008/ 9/21 01:28
2	form2			
3	form3			
4	form4			
5	form5			
6	form6			
7	form7			
8	form8	練習用フォーム	総・たて	2008/ 6/ 4 18:00
9	form9	日報(サンダル)フォーム	総・たて	2008/ 5/ 4 18:00
10	form10	月報(サンダル)フォーム	総・たて	2008/ 5/ 4 18:00

図 18.8.6 帳票初期ファイル登録画面表示

- ②次に、 ボタンを数回押して“PUT”モードを選択して、エクスポートする帳票書式の登録行に対応する  ボタンを押して下さい。以下のように、エクスポートファイル名の「文字入力」ダイアログが表示されます。



No.	書式コード	フォーマット名	所属	作成 編集	PUT	Export
1	form1	レポートフォーム	総・たて	2008/ 9/21 01:28		
2	form2					
3	form3					
4	form4					
5	form5					
6	form6					
7	form7					
8	form8	練習用				
9	form9	日報(サ				
10	form10	月報(サ				

文字入力

form1

編集文字数/行 20 編集行数 1

図 18.8.7 エクスポートファイル名の編集

- ③ファイル名には、デフォルトの名前(書式コード)が設定されていますので、変更する場合にはこの画面で編集することができます。“ENT”ボタンを押すと、ファイル名を変更した場合も、変更しない場合にも、表示されている名前に“.prs”の拡張子が付加されてファイルが作成出力されます。

[インポート・エクスポート手順早見表]

●リソースファイルのインポート手順

インポートファイル	取り込む画面	外部ファイルを置くディレクトリ	インポート後の処理
監視パネルファイル	監視パネルメニュー ↓ 監視パネル管理	/tmp/*.mrs	①監視パネルメニューから取り込んだパネルを呼び出す。 ②監視パネルから CAD を呼び出して“ENT”ボタンで終了する。
制御プログラムファイル	制御プログラム登録	/tmp/*.pgc	なし
制御パラメータファイル	制御プログラム登録 ↓ 制御パラメータ登録	/tmp/*.pra	①登録画面で“ENT”ボタンで終了する。
汎用属性名称付属部	制御プログラム登録	-	-
帳票書式ファイル	帳票フォーマット登録	/tmp/*.prs	①CAD 画面で“ENT”ボタンで終了する。
帳票プログラムファイル	帳票プログラム登録	/tmp/*.pgc	なし
CAD 用 ビットマップファイル	部品箱 ↓ 絵部品箱 ↓ ビットマップ (絵部品) 分類 ↓ ビットマップ (絵部品) 一覧	/tmp/*.xbm	なし

※/tmp は、インポート及びエクスポートするデフォルトのディレクトリです。

●リソースファイルのエクスポート手順

エクスポートファイル	出力する画面	ファイルを出力するディレクトリ	ファイル名の変更
監視パネルファイル	監視パネルメニュー ↓ 監視パネル管理	/tmp/*.mrs	○
制御プログラムファイル	制御プログラム登録	/tmp/*.pgc	○
制御パラメータファイル	制御プログラム登録 ↓ 制御パラメータ登録	/tmp/*.pra	○
帳票書式ファイル	帳票フォーマット登録	/tmp/*.prs	○
帳票プログラムファイル	帳票プログラム登録	/tmp/*.pgc	○
CAD 用 ビットマップファイル	部品箱 ↓ 絵部品箱 ↓ ビットマップ (絵部品) 分類 ↓ ビットマップ (絵部品) 一覧	/tmp/*.xbm	○ ※ファイル名に全角文字は使用できません。 ※ファイル名の長さは、拡張子と合わせて 23 文字以内にする必要があります。

※/tmp は、インポート及びエクスポートするデフォルトのディレクトリです。

