

A d v a n c e C A D

ユーティリティ・マニュアル

Advance CAD software version 22

ユーティリティ・マニュアル

Advance CAD software version 22

2025 年 1 月 15 日 第1版

Copyright © 1986-2025 伊藤忠テクノソリューションズ株式会社
〒105-6950 東京都港区虎ノ門 4-1-1 神谷町トラストタワー

本書の内容の一部または全部を無断転載することを禁止します。
本書の内容に関しては将来予告無しに変更することがあります。
本書は将来の開発による変更を前提としています。本書は現時点でできる限り正確に記述するよう心がけました。しかし弊社は提供した資料に基づくいかなる損害の責任も負いません。また将来の開発により生ずる変更によるいかなる損害についても責任を負いません。

MS, MS-DOS, Windows、Windows NT、Windows 2000、Windows XP、Windows Vista、Windows 7、Windows 10、Windows 11、Visual C++ および Microsoft は Microsoft Corporation の商標または登録商標です。

SolidWorks および SolidWorks のロゴは SolidWorks 社の登録商標です。

FlexNet Publisher は Revenera Software 社の登録商標です。

libtiff の著作権は以下のとおりです。

Copyright (c) 1988-1996 Sam Leffler

Copyright (c) 1991-1996 Silicon Graphics, Inc.

各会社名、各製品名は各社の商標または登録商標です。

はじめに

本書は Advance CAD システム標準のユーティリティコマンドとプログラムについて説明します。
本書の構成はつぎのとおりです。

第 1 章	APG パラメトリック図形の登録と使い方
第 2 章	関係アイテム 関係アイテムの使い方
第 3 章	特性データ アイテムに付ける非図形情報（特性データ）の使い方
第 4 章	パーツ パーツ（特性データ、関係アイテムを持つ APG）の作成と使い方
第 5 章	モデル検索 ファイルの検索方法
第 6 章	I G E S IGES プリプロセッサ／ポストプロセッサについて
第 7 章	S X F SXF ファイルを介して他の CAD/CAM システムとデータ交換をする方法
第 8 章	共通データファイル Advance CAD の共通データファイルを介して他の CAD システムとデータ 交換する方法
第 9 章	イメージファイルの作成 技術文章の挿入図などの作成方法
第 10 章	マクロ マクロの作成方法
第 11 章	同時設計（コンカレント） 同時設計機能の使い方
第 12 章	モデルファイルダウンコンバータ モデルファイルを下位バージョンのモデルファイルに変換する方法

● 技術的なお問い合わせ先

Advance CAD の技術的なご質問は下記で受付けております。
Advance CAD ソフトウェア保守契約に加入されているお客様に限らせていただきます。

----- Advance CAD サポートサービス -----

E-mail : acad_support@ctc-g.co.jp

----- 受付時間 : 平日 9:00 ～ 17:30 -----



目次

第 1 章	APG	1
1.1	APG テキストファイル	1
1.2	APG で使う図形の作成	3
1.2.1	APG で解析できるアイテム	3
1.2.2	図形と寸法を作成するときの注意	5
1.2.3	APG パラメータファイル	8
1.3	コマンド	9
1.3.1	APG ファイルを作成する	9
1.3.2	APG ファイルからの図形作成	10
1.3.3	APG ファイルから図形をドラッグモードで作成する	12
1.3.4	一覧図から選択する	15
1.3.5	APG アイテムを更新する	15
1.3.6	APG 登録時のデバッグ	17
1.3.7	既存の寸法値を変更し図形を再作成する	18
1.3.8	APG ファイル作成時の図形寸法をモデル中に作成する	19
1.3.9	APG 条件設定	20
1.3.10	APG 条件を設定する	20
1.3.11	交点／接点を定義する	21
1.3.12	中心点を定義する	22
1.3.13	中間点を定義する	22
1.3.14	最短距離点を定義する	23
1.3.15	平行線を定義する	24
1.3.16	2 点間線を定義する	24
1.3.17	接線を定義する	25
1.3.18	中心点と半径を定義する	26
1.3.19	接円を定義する	26
1.3.20	中心点と接図形を定義する	28
1.3.21	中心点と円上点を定義する	29
1.3.22	円上の 3 点を定義する	29
1.3.23	3 接円を定義する	30
1.3.24	APG 設定条件についての情報を表示する	31
1.3.25	APG 設定条件を削除する	32
1.3.26	APG 設定条件の白色表示を消す	33
1.4	APG バッチ配置	34
1.4.1	APG パラメータファイルの作成方法	34
1.4.2	APG バッチコントロールファイル	39
1.4.3	バッチ配置する	42
1.5	APG 実行例	43
第 2 章	関係アイテム (associate item)	51
2.1	コマンド	52
2.1.1	関係アイテムの部品配置のマトリックス表	52
2.1.2	関係アイテムを作る	53
2.1.3	関係アイテムにメンバを追加する	54
2.1.4	関係アイテムからメンバを除去する	55
2.1.5	関係アイテムを分解する	55
2.1.6	関係アイテムを削除する	56
2.1.7	関係アイテムの名前を変更する	56
2.1.8	関係アイテムの確認	56
2.1.9	関係アイテムをリストする	57
2.1.10	関係アイテムをアクティブリストにのせる	57
2.1.11	関係アイテムを利用して部品を配置する	58
2.1.12	関係アイテムからサブモデルを作る	59

目次

2.1.13 関係アイテムの属性を修正する	61
2.1.14 関係アイテムの位置を移動する	61
2.1.15 関係アイテムのメンバアイテムを伸縮する	62
2.1.16 関係アイテムのメンバアイテムを縮小／拡大する	63
第 3 章 特性データ	65
3.1 特性ファイル	66
3.1.1 特性ファイルの作成	68
3.1.2 インプットファイル	69
3.1.3 テンプレートファイル	69
3.1.4 特性ファイルチェッカー	77
3.2 コマンド	80
3.2.1 特性データの設定	81
3.2.1.1 特性ファイルを指定する	81
3.2.1.2 特性ファイルのレコード番号を指定する	81
3.2.1.3 特性データテーブルを設定する	81
3.2.1.4 アイテムに付いている特性データを特性データテーブルにコピーする	82
3.2.1.5 特性データテーブルを削除する	83
3.2.2 特性データテーブルを表示する	83
3.2.2.1 アイテムに特性データを付ける	83
3.2.2.2 アイテムの特性データを削除する	84
3.2.2.3 アイテムの特性データを変更する	84
3.2.2.4 アイテムの特性データを更新する	85
3.2.2.5 特性データをコピーする	85
3.2.3 アイテムに付けた特性データを確認する	85
3.2.3.1 アイテムの特性データを画面に表示する	85
3.2.3.2 特性レコード番号使用一覧表	86
3.2.3.3 特性データの世代を 1 世代更新する	86
3.2.3.4 特性データをもとにアイテム検索をする	86
3.2.3.5 アイテムの特性レコード番号を検索する	88
3.2.4 特性データの表示	88
3.2.4.1 アイテムの特性データをジェネラルノートで表示する	89
3.2.4.2 アイテムの特性データを TAG 名で表示する	90
3.2.4.3 アイテムの特性データをリファランスノートで表示する	90
3.2.4.4 特性データの表示をやめる（非表示にする）	91
3.2.5 部品表（リスト）の作成	91
3.2.5.1 アイテムの特性データをファイルに出力する	91
3.2.5.2 部品表（特性集計表）を作る	92
3.2.6 特性データの確認	95
3.2.6.1 同一特性データを持ったシンボルを確認する	95
第 4 章 パーツ	97
4.1 パーツの定義	98
4.1.1 パーツに必要なファイル	98
4.1.2 パーツコントロールファイル	99
4.1.3 パーツ定義ファイルの作成	99
4.1.4 パーツ定義ファイル例	102
4.2 パーツセット	107
4.3 パーツセットの定義	108
4.3.1 パーツセットに必要なファイル	108
4.3.2 パーツセットコントロールファイル	109
4.3.3 パーツセット定義ファイルの作成	109

4.3.4 パーツセット定義ファイル例	112
4.4 コマンド	114
4.4.1 パーツを配置する	114
4.4.2 パーツセットを配置する	117
4.4.3 パーツ・パーツセットをグループ化する	119
4.4.4 パーツ・パーツセットを更新する	119
4.4.5 パーツ・パーツセットを削除する	119
4.4.6 パーツ・パーツセットを確認する	120
4.4.7 パーツ・パーツセットをサイズ変更する	121
4.4.8 パーツ・パーツセットを移動する	121
4.4.9 パーツに寸法を付ける	122
4.4.10 パーツ・パーツセットのグループを解除する	122
4.4.11 パーツ・パーツセットをアクティブリストにのせる	122
4.4.12 パーツセットを解除する	123
4.4.13 パーツ・パーツセットのクリーンナップ	123
4.4.14 パーツ・パーツセットのアソシエイトリスト	124
4.4.15 穴形状を配置する	124
4.4.16 穴形状を作成する	126
 第 5 章 モデル検索	 129
5.1 主な機能	130
5.2 準備	132
5.2.1 コンフィグレーションファイルの確認	132
5.2.2 モデル検食用ディレクトリ定義ファイルを作成する	133
5.2.2.1 ACADM.DM.SET ファイルを作成する	133
5.2.2.2 ACADLOCALMDM.SET ファイルを作成する	139
5.2.2.3 ACADOTHERMDM.SET ファイルを作成する	140
5.2.2.4 PLOTMDM.SET ファイルを作成する	140
5.2.3 モデル管理ファイルを作成する	141
5.2.4 モデル情報ファイルを作成する	142
5.2.5 表示用テンプレートファイルの作成	143
5.2.5.1 モデル表示用を作成する	143
5.2.5.2 シンボル表示用を作成する	147
5.2.6 管理番号のグループを指定するファイルを作成する (ACADM.DM.MGN)	148
5.2.7 管理番号の階層構造を指定するファイルを作成する (ACADM.DM.GRP)	149
5.2.8 リスト出力テンプレートファイルの作成	150
5.2.9 ページタイトルテンプレートファイルの作成	152
5.3 コマンド	154
5.3.1 モデルの検索	154
5.3.1.1 モデルを検索する	159
5.3.1.2 検索結果のモデルファイル情報を表示する	162
5.3.1.3 モデルのページタイトル情報を表示する	162
5.3.1.4 検索後のモデル内のピクチャを画面に表示する	163
5.3.1.5 モデルを読み込む	163
5.3.1.6 モデルの管理番号及び名前を変更する	164
5.3.1.7 モデルの所有者名を変更する	165
5.3.1.8 モデルタイトルを変更する	165
5.3.1.9 モデルの書込み属性を変更する	166
5.3.1.10 モデルを削除する	166
5.3.1.11 モデルの検索結果の一覧をプリンタおよびファイルに出力する	167
5.3.2 シンボル検索	168
5.3.2.1 シンボルを検索する	169
5.3.2.2 シンボルを削除する	170
5.3.3 APG 検索	171
5.3.3.1 APG ファイルを検索する	171

目次

5.3.3.2 APG ファイルを削除する	172
5.3.4 旧モデル検索	173
5.3.4.1 旧モデルを検索する	174
5.3.4.2 旧モデルを現モデルに復元する	175
5.3.4.3 旧モデルを削除する	175
5.3.4.4 旧モデルを廃棄する	175
5.3.5 廃棄モデル検索	176
5.3.5.1 廃棄モデルを検索する	176
5.3.5.2 廃棄モデルを現モデルに復元する	177
5.3.5.3 廃棄モデルを削除する	178
5.3.6 旧シンボル検索	178
5.3.6.1 旧シンボルを検索する	179
5.3.6.2 旧シンボルを削除する	179
5.3.7 旧 APG 検索	180
5.3.7.1 旧 APG ファイルを検索する	180
5.3.7.2 旧 APG ファイルを削除する	181
5.3.8 ファイル検索	181
5.3.8.1 ファイル種類別に検索する	182
5.3.8.2 ファイルを削除する	183
5.3.9 モデル登録	183
5.3.9.1 未登録のモデルを検索する	184
5.3.9.2 未登録のモデルを管理ファイルに登録する	184
5.3.10 親子関係検索	185
5.3.10.1 親子関係を検索する	186
5.3.11 親子関係作成	187
5.3.11.1 親子関係を作成する	188
5.3.12 モデル情報作成	188
5.3.12.1 新しいモデル情報ファイル作成する	189
5.3.13 モデルファイルのダンプ	190
5.3.13.1 モデル検索後モデルダンプをする	191
5.3.14 モデルファイルのロード	193
5.3.14.1 モデルロードするために検索をする	194
5.3.15 ダンプしたカセットの管理	195
5.3.15.1 カセット内のモデル名の一覧を確認する	196
5.3.15.2 カセットを削除する	197
5.3.15.3 カセットテープの書込み可能レコード数を求める	197
5.3.16 モデルのプロット出力	198
5.3.16.1 検索結果のモデルをプロット出力する	199
5.3.17 プロットファイルの出力	199
5.3.17.1 作図ファイルを検索する	201
5.3.17.2 作図ファイルを出力する	201
5.3.17.3 作図ファイルの名前を変更する	202
5.3.17.4 作図ファイルの書込み属性を変更する	202
5.3.17.5 作図ファイルを削除する	202
5.3.18 ダンプファイルのロード	202
5.3.18.1 ダンプしたモデルを指定したディレクトリにロードする	203
5.3.19 管理ファイル作成	204
5.3.19.1 モデル管理ファイルを新規作成する	205
5.3.20 検索結果出力ファイルの形式	206
5.3.21 マクロ作成例	207
第 6 章 I G E S	211
6.1 コマンド	212
6.1.1 IGES ファイルを読み込む	212
6.1.2 アクティブモデルを IGES ファイルに書き込む	213

6.1.3 ポンチ絵を作成する	215
6.2 Advance CAD — IGES	217
6.2.1 IGES ファイルの読み込み	217
6.2.2 IGES ファイルの作成	222
6.2.3 読み込み／書き込み可能な IGES エンティティ	227
6.2.4 日本語文字列の取り扱い	228
6.2.5 アローヘッドの種類	230
6.2.6 dimension entity	230
6.2.7 general note entity	231
6.2.8 IGES/WRITE で出力するグローバル部のパラメーター一覧	236
6.3 IGES 出力の磁気テープへの入出力	238

第 7 章 SXF

7.1 国土交通省 C A D 製図基準 (案) への対応および S X F 変換を円滑におこなうために	240
7.1.1 概要	240
7.1.2 初期設定	240
7.1.2.1 コンフィグレーションファイルとディレクトリの作成	241
7.1.2.2 サンプルファイルのコピー	242
7.1.2.3 標準定数ファイルに線種の既定値を登録	243
7.1.2.4 各種設定ファイルの作成	243
7.1.2.5 製図基準ファイルの絞り込みと自動問い合わせ	245
7.1.3 C A D 製図基準 (案) での運用	245
7.1.3.1 製図基準を設定する	245
7.1.3.2 図面種別を設定する	246
7.1.3.3 SXF ファイルを読み込む	246
7.1.3.4 責任主体 (ライフサイクル) を変更する	246
7.1.3.5 SXF ファイルを出力する	247
7.1.3.6 その他の関連コマンド	247
7.2 コマンド	248
7.2.1 製図基準テンプレートファイルを使用して環境を設定	249
7.2.2 製図基準を設定	250
7.2.3 図面種別を設定	250
7.2.4 SXF ファイル読み込み	250
7.2.5 SXF ファイル書き込み	254
7.2.6 ポイントマーカーを作成	258
7.2.7 ユーザ定義ハッチング作成	258
7.2.8 パターンハッチング作成	259
7.2.9 塗りハッチング作成	260
7.2.10 既定義ハッチング作成	260
7.2.11 ユーザ定義色設定 (ダイアログモードでのみ動作)	260
7.2.12 ユーザ定義線種設定 (ダイアログモードでのみ動作)	261
7.2.13 ユーザ定義線幅設定 (ダイアログモードでのみ動作)	262
7.2.14 部分図一覧表示 (ダイアログモードでのみ動作)	263
7.2.15 部分図の倍率設定 (ダイアログモードでのみ動作)	263
7.2.16 各名称の一覧を表示する (ダイアログモードでのみ動作)	264
7.2.17 複合図形 (作図グループ・作図部品) 配置 (ダイアログモードのみ)	267
7.2.18 複合図形 (作図グループ・作図部品) 作成・編集 (ダイアログモードのみ)	269
7.2.19 複合図形 (作図部品) 配置情報修正 (ダイアログモードのみ)	272
7.2.20 複合図形 (作図グループ・作図部品) 削除	273
7.2.21 複合図形 (作図グループ・作図部品) 分解	273
7.2.22 図形を移動・回転・縮小拡大する	273
7.2.23 表示順	274
7.2.24 表題欄設定 (ダイアログモードでのみ動作)	275
7.2.25 表題欄属性 (ダイアログモードでのみ動作)	276

目次

7.2.26 属性設定（ダイアログモードでのみ動作）	278
7.2.27 等高線作成	278
7.2.28 レイヤの表示・選択マスク設定（ダイアログモードでのみ動作）	278
7.2.29 図形の確認（ダイアログモードでのみ動作）	281
7.2.30 画面再表示	284
7.2.31 矩形領域を拡大表示	284
7.2.32 拡大表示	285
7.2.33 縮小表示	285
7.2.34 全体表示	285
7.2.35 図形領域全体表示	285
7.2.36 表示位置移動	286
7.2.37 図面印刷（ダイアログモードでのみ動作）	286
7.2.38 画面印刷（ダイアログモードでのみ動作）	288
7.2.39 図面印刷プレビュー（ダイアログモードでのみ動作）	289
7.2.40 画面印刷プレビュー（ダイアログモードでのみ動作）	291
7.2.41 属性付加機構（ダイアログモードでのみ動作）	293
7.2.42 確認機能（ダイアログモードでのみ動作）	298
7.2.43 オプション設定（ダイアログモードでのみ動作）	301
7.3 変換テーブルファイル	302
7.3.1 書式	302
7.3.2 SXF → Advance CAD 変換パラメータ	303
7.3.3 SXF → Advance CAD 変換パラメータ既定値	305
7.3.4 Advance CAD → SXF 変換パラメータ	307
7.3.5 Advance CAD → SXF 交換パラメータ既定値	310
7.4 製図基準テンプレートファイル	316
7.4.1 前提条件	316
7.4.2 テンプレートファイルの場所	316
7.4.3 テンプレートの書式	316
7.4.4 運用	320
7.5 SXF 既定義線種・線幅・色コード（参考資料）	321
7.5.1 既定義線種コード	321
7.5.2 既定義線幅コード	321
7.5.3 既定義色コード	321
第 8 章 共通データファイル	323
8.1 共通データファイル バージョン 10.0	324
8.1.1 共通データファイルの構造	324
8.1.2 バージョン番号ブロック	324
8.1.3 モデル データブロック	325
8.1.4 ピクチャデータブロック	326
8.1.5 ドローイング データブック	327
8.1.6 アイテム データブロック	328
8.2 参考：旧バージョンの共通データファイルの書式	332
8.2.1 参考：共通データファイル バージョン 1.0	332
8.2.2 参考：共通データファイル バージョン 2.0	337
8.2.3 参考：共通データファイル バージョン 3.0	342
8.2.4 参考：共通データファイル バージョン 4.0	347
8.2.5 参考：共通データファイル バージョン 5.0	352
8.2.6 参考：共通データファイル バージョン 6.0	357
8.2.7 参考：共通データファイル バージョン 7.0	363
8.2.8 参考：共通データファイル バージョン 8.0	369
8.2.9 参考：共通データファイル バージョン 9.0	375

第 9 章 イメージファイル	381
9.1 イメージファイルの作成	382
9.1.1 ラスターファイルを作成する	382
9.1.2 ラスターファイルを作成する	383
9.1.3 図面配置状態のラスターファイルを作る	384
9.1.4 Adobe Illustrator で取り込める形式の EPS ファイルの出力を行う	384
第 10 章 マクロ	387
10.1 マクロとは何か	387
10.2 マクロファイルの書式	389
10.3 言語部	390
10.3.1 識別子 (名前)	390
10.3.2 予約語	390
10.3.3 変数	390
10.3.3.1 組み込み関数名と変数名の区別	390
10.3.3.2 内部変数と外部変数	390
10.3.3.3 外部変数の宣言	391
10.3.4 データの型および定数	391
10.3.4.1 数値型	391
10.3.4.2 文字列型	391
10.3.4.3 座標型, デジタイズ点型, テンポラリポイント型	393
10.3.4.4 スクリーン座標型	393
10.3.4.5 カーソル位置型	394
10.3.4.6 アイテム識別子型	394
10.3.4.7 コマンドエンド型、バックスペースキー型、スペースキー型	394
10.3.5 組み込み定数	395
10.3.6 配列	395
10.3.6.1 一次元配列	395
10.3.6.2 二次元配列	396
10.3.7 演算子	397
10.3.7.1 単項演算子 ' + ' 絶対値	398
10.3.7.2 単項演算子 ' - ' 符号反転	398
10.3.7.3 単項演算子が続くときはもっとも右から順に処理する。	398
10.3.7.4 法演算子	398
10.3.7.5 関係演算子、等値演算子	398
10.3.7.6 加算 ' + ' 演算子	398
10.3.7.7 条件演算子 a ? b : c	398
10.3.7.8 代入演算子 ' += ' , ' -= ' , ' *= ' , ' /= ' , ' %= '	399
10.3.7.9 複代入文	399
10.3.7.10 座標型・デジタイズ点型の演算子	399
10.3.7.11 カンマ演算子	399
10.3.8 式と文	399
10.3.9 フロー制御文	400
10.3.9.1 while 文	400
10.3.9.2 do 文	400
10.3.9.3 for 文	400
10.3.9.4 条件文	401
10.3.9.5 switch 文	401
10.3.9.6 continue 文	401
10.3.9.7 break 文	402
10.3.9.8 goto 文、ラベル	402
10.3.9.9 return 文	402
10.3.9.10 exit 文	402
10.3.9.11 echo 文	403

目次

10.3.10 組み込み関数	403
10.3.10.1 算術関数	403
10.3.10.2 文字列	404
10.3.10.3 変数テーブル	409
10.3.10.4 会話型入力関数	410
10.3.10.5 座標・デジタイズ点・スクリーン座標・カーソル位置型・テンポラリポイント型 データの生成	412
10.3.10.6 配列要素を並べ替える関数	413
10.3.10.7 アイテム識別子を返す関数	413
10.3.10.8 ファイル入出力機能	414
10.3.10.9 ファイル属性を調べる	416
10.3.10.10 テンポラリウィンドウ	416
10.3.10.11 アイテム数を得る関数	419
10.3.10.12 モデル変更の有無を調べる関数	420
10.3.10.13 マスクの状態を得る関数	421
10.3.10.14 カラー割付情報を得る関数	422
10.3.10.15 図面配置されているウィンドウ数を得る関数	422
10.3.10.16 ピクチャマトリックスを得る関数	422
10.3.10.17 製図用定義などの現在値を得る関数	423
10.3.10.18 環境変数の内容を得る関数	429
10.3.10.19 menu 関数	429
10.3.11 制限	429
10.4 コマンドストリーム部	431
10.5 マクロの実行	441
10.5.1 実行コマンド	441
10.5.2 ダイアログボックス	442
10.5.3 図形表示を抑止する	442
10.5.4 情報表示	443
10.5.5 メニュー／メッセージの表示	443
10.5.6 ダイアログモードを切り替え	444
10.5.7 マクロの実行結果を1度のアンドウで元に戻す	444
10.5.8 アテンションキー入力を無効にする	444
10.5.9 クリップボード内文字列操作マクロ関数	445
10.5.10 マクロのデバッグ	445
10.5.11 マクロライブラリ	447
10.5.12 マクロライブラリ作成ツール（参考：UNIX 版のみ）	448
10.5.13 セッションファイルからマクロを作る（参考例）	455
10.5.14 マクロの例	456
第 11 章 同時設計（コンカレント）	461
11.1 概要	461
11.2 機能説明	465
11.2.1 キャンバスモデル	465
11.2.2 同時設計作業	466
11.2.3 表示制御	467
11.2.4 キャンバスモデルの保存／呼び出し	467
11.2.5 コンカレントファイルとキャンバスモデルの整合	468
11.2.6 ペアレント情報	468
11.2.7 ペアレント情報の回復	469
11.3 準備	470
11.3.1 メニューファイルの修正	470
11.3.2 コンカレントファイルのディレクトリ	470
11.3.3 ユーザキャンバスモデルファイルのディレクトリ	470
11.3.4 ユーザ名の登録	470
11.3.5 モデル定数の設定	470

目次

11.3.6 制約.....	471
11.3.7 処理速度	472
11.4 コマンド.....	473
11.4.1 コンカレントファイルを新規作成する.....	474
11.4.2 コンカレントを終了する	474
11.4.3 アクティブメンバーの選択を中止する.....	474
11.4.4 ペアレントを呼び出す.....	474
11.4.5 ユーザキャンバスモデルを読み込む	475
11.4.6 ユーザキャンバスモデルを保存する	475
11.4.7 コンカレントファイルを再作成する	476
11.4.8 キャンバスモデルにメンバーを追加する	476
11.4.9 実データをメンバーに変換する	477
11.4.10 メンバーを削除する	477
11.4.11 全メンバーを更新する.....	478
11.4.12 アクティブにするメンバーを選択する.....	478
11.4.13 アクティブメンバーを保存する	479
11.4.14 保存するメンバーを指定する	480
11.4.15 ペアレント情報を表示する.....	480
11.4.16 メンバーの表示／非表示を変更する	481
11.4.17 メンバーアイテムをサブモデルに変換する	482
11.4.18 サブモデルをメンバーに変換する.....	482
11.4.19 表示／非表示の条件を表示コントロールファイルに保存する.....	482
11.4.20 表示コントロールファイルをロードする	483
11.4.21 マスターコンカレントファイルを更新する	483
11.4.22 メンバー表示のクラス／アイテム表示マスクを変更.....	483
11.4.23 コンカレント情報のベリファイ	484
11.5 同時設計作業中のメンバアイテムの選択	484
11.5.1 メンバアイテムの選択の可否を切り換え	485

第 12 章 モデルダウンコンバータ 487

12.1 ダウンコンバータ	488
12.1.1 使用方法	488
12.1.1.1 コマンド入力に変換する場合.....	488
12.1.1.2 ダイアログに変換する場合.....	488
12.1.2 制限事項	494

目次

第 1 章 A P G

● 概要

類似形状を作図するときは、大きさが変わるところの寸法値だけを指示して作図できると便利です。通常これを行うにはプログラミングが必要でした。

Advance CAD のパラメトリック機能はプログラミングなしで形状のパラメタライズを行うことができます。面倒なプログラミングの代りに形状と寸法を解析して自動的にパラメタライズを実行する機能がありますので、誰でもパラメトリック機能を使うことができます。この機能を APG (Automatically Parametrized Geometry) と呼んでいます。

APG の使い方はつぎのとおりです。

- (1) 形状を作成する
図形を作り寸法を記入する。寸法が変わるところを変数に置き換える。計算式を含めることができる。
- (2) APG 登録
図形と寸法アイテムの相関関係を調べパラメタライズし、APG ファイルを作成する。
- (3) 配置
登録した APG に寸法を指定して図形を作る。作図後に寸法を修正して形状を作り直すこともできる。

APG は長さ寸法、半径寸法だけでなく角度寸法や座標寸法もパラメタライズできます。

角度のパラメタライズはきわめて有効です。

また、単純に寸法を変えるのではなく、接線や接円を正しく処理します。たとえば接円の半径を変更したならば、その円弧に接していた線分は正しく円弧に接します。さらに接円の半径を 0 とすれば接円なしの形状ができます。このように APG は入力形状を分析して、接線・接円・交点などの関係を自動的に調べあげ、その関係を保持するようにパラメタライズします。

APG を利用して標準部品や類似設計を行うことができます。

1.1 APG テキストファイル

Advance CAD バージョン 12 までは APG ファイルはバイナリファイルだったため、異機種間でデータの互換性がありませんでした。バージョン 13 からは、異機種間の APG の移動を簡単にするために APG ファイルをテキストファイルで出力することができます。バージョン 18 からは、テキストファイルのみが使用できます。

古い APG バイナリファイル (拡張子 APG) を APG テキストファイル (拡張子 APX) に変換するには、プログラム `apgconv` を使います。`apgconv` の変換対象となる APG ファイルのバージョンは 4.0 以上です。

例

```
prompt% apgconv    APG ファイル名 <cr>
```

プログラム `apgconv` で変換した APG テキストファイルがうまく使えないなら、APG を再作成しなければなりません。APG を作成したときのモデルファイルがあれば簡単です。そのようなモデルファイルがない場合は、APG バイナリファイルから APG 作成時のデータを復元する方法があります。APGORG コマンドで APG バイナリファイルを読み込み、そのモデルを保存します。これは APG バイナリファイルをサポートする Advance CAD バージョン 17 以下で作業します。このモデルファイルがあれば APG を再作成できます。

APG ファイルのデフォルトディレクトリは ACAD.SET のキーワード `#APG#` で指示します。APG テキストファイルを使用するには キーワード `#APG#` の拡張子を `APX` にします。

例

```
#APG#  "/acad/files/"  !.APG!   APG file
↓
#APG#  "/acad/files/"  !.APX!   APG file
```

APG テキストファイルの先頭にある下記の行は APG のバージョン番号を記述しています。
/Revision [502] Advance CAD APX

カギカッコの中の 502 がバージョン番号です（少数点がない 3 桁の数です）。
以下は Advance CAD と APG のバージョン番号の対応表です。

```
V20-      5.02
V13-19    5.01
V12       4.07
V11       4.06
```

下位バージョンの Advance CAD で作成した APG ファイルは、そのまま上位バージョンの Advance CAD で使用できます。その逆は避けてください。

Advance CAD バージョン 20 以降の APG ファイルの変更は以下の通りです。ファイルの書式は変わりませんが、項目の値の範囲を変更しています。

1) アイテム属性の上限値を拡張

幾何要素、スプライン、シンボル、寸法、注記レコードのアイテム属性の上限を APG のアイテム属性にも適応しました。

属性	以前の上限	新しい上限
クラス	255	256
レビジョン	127	256
線種	31	63
線幅	7	16

2) テキストフォントの上限値を拡張

テキストフォントの上限を APG のテキストフォントにも適応しました。

3) テキストとマークの角度

以前は角度は度単位でしたが少数点以下も出力するようになりました。マークが引き出し線の矢であれば、矢の向きは引き出し線に合わせますので、この角度は無視します。

1.2 APG で使う図形の作成

1.2.1 APG で解析できるアイテム

APG で解析できるアイテムは次のとおりです。

● 図形アイテム

各アイテムが以下の条件で決定できるときに限ります。

- (1) 点
 - ・ 原点 (ORG)
 - ・ 交点 (PIN) line - line, arc - arc, line - arc
 - ・ 接点 arc - line, arc - arc
 - ・ 円弧中心点 (PCR)
 - ・ ベクトル点 (PVR)
- (2) 線分
 - ・ 2 点を結ぶ直線 (LBP)
 - ・ 1 点とベクトルで定義される直線 (LTP)
 - ・ 接線 (LTAN) arc - vector, arc - point, arc - arc
- (3) 円／円弧
 - ・ 中心点と半径 (CRC)
 - ・ 中心点と円弧上の 1 点 (CCC)
 - ・ 円弧上の 3 点 (CTP)
 - ・ 接円 (FIL) point - line, point - arc, line - line, line - arc, arc - arc
 - ・ 3 接円 (C3C) point - point - line, line - line - line, point - point - arc, line - line - arc, point - line - line, line - arc - arc, point - line - arc, arc - arc - arc, point - arc - arc
- (4) 自由曲線
 - ・ 構成点だけがパラメタライズされます。
- (5) スtring
- (6) 図形の属性
元の図形が持っている属性（線種・線幅・クラス番号・レビジョン番号）はそのまま出力されます。

● 寸法アイテム

APG で使用できる寸法アイテムは次のとおりです。ただし APG 作成時の寸法形状バランスは保持されません。累進寸法とオーディネイト寸法は使用できません。
また、寸法要素では元のレビジョン属性は保持されません。

- (1) 長さ寸法 (Linear Dimension)
 - ・ 単一寸法
 - ・ 直列寸法
 - ・ 並列寸法
 - ・ 片寄せ寸法
- (2) 角度寸法 (Anguler Dimension)
 - ・ 単一寸法
 - ・ 直列寸法
 - ・ 並列寸法

- 寸法変数 ... 256

● 接円弧

APG 配置で作成された接円弧が崩れてしまうときは、登録した接円弧の定義がうまくできていません。このようなときは『TRACE』を使用して図形の定義が正しいかどうか調べてください。

1.2.2 図形と寸法を作成するときの注意

● 図形の原点

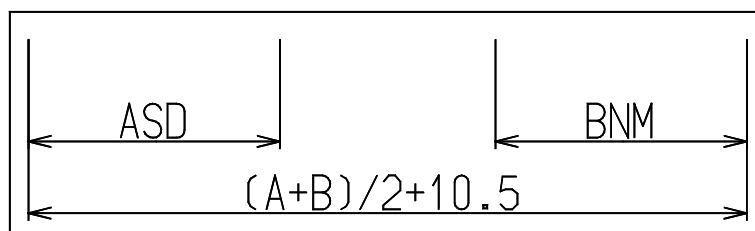
原点は APG 登録時にも指定できるためどこにあってもかまいませんが、モデル原点とするのが簡単です。

● 図形と端点の決定

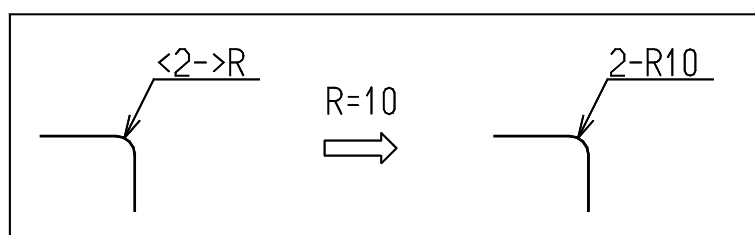
線分は無限直線、円／円弧は円として扱います。また交点／接点などは無限直線や円との交点／接点として定義されます。

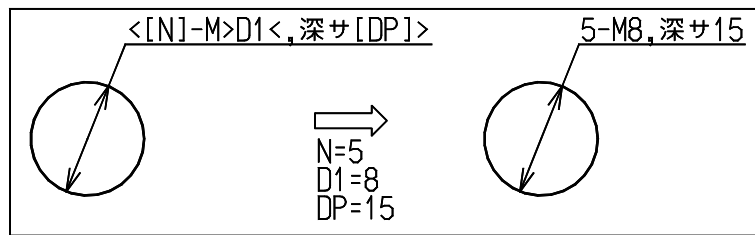
● パラメータテキスト

- 寸法テキストを入力する場合に先頭文字が [や < などの特別な文字で始まるときは、文字列入力であることを明示する必要があります。寸法テキストをダブルクォーツ (") で囲んで入力する方法が安全です。
- 寸法テキストが単純な変数名でない場合は、計算式とみなして処理しますので、計算式は [] で囲んでなくてもかまいません。



- 寸法テキスト中の < > で囲まれた文字列は、寸法数値の前文字、後文字として処理されます。< > で囲まれた文字列の中に [] で囲まれた変数があれば計算結果に置き換えます。





- 寸法テキスト中の変数名はパラメータとして使用されます。ただし以下の制限があります。

パラメータの文字数	64 文字以内
パラメータの変数名	6 文字以内。大文字の英文字 (A ~ Z) で始まること。
パラメータの数	最大 256 個

- 計算式は Advance CAD の計算式の機能がそのまま使用できます。演算子および関数はつぎのとおりです。

+ - * / ** () %
SQRT SIN COS TAN ASIN ACOS ATAN INT LOG LOG10 DMS

● センター振り分けの寸法

一点鎖線 (線種 3 の線分) は中心線で特別な意味があります。

寸法と角度寸法の寸法線の中心点が一点鎖線の線分上にあるとき、その寸法はセンター振分として処理します。

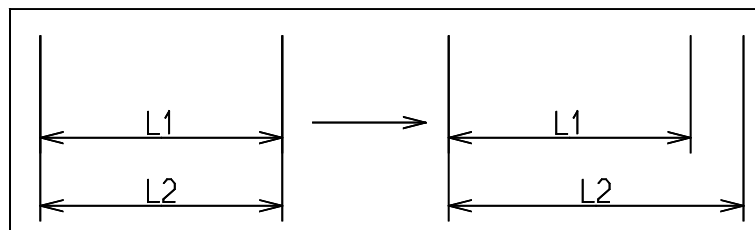
● 円弧半径の決定

円弧半径に対するパラメータは、つぎのいずれかで設定されます。

- 半径寸法の引出し線の端点が指す円弧に対して、半径寸法パラメータがセットされたものとしします。
- (1) で決定された円弧と同一半径を持つ円弧はすべてそれと同じパラメータがセットされたものとしします。
- (1) または (2) で 半径が決定できない円弧は、「3 点を通る円弧」または「中心点と円上の 1 点で決まる円弧」のどちらかでパラメタライズします。
- 以上のどれでも決定できない円弧の半径はパラメタライズされず、そのまま使用されます。

● 寸法の重なり

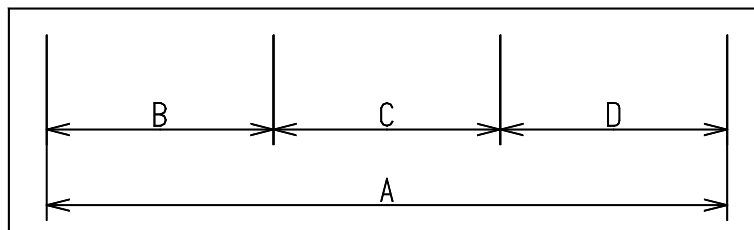
下図のように同じ寸法があると、どちらの寸法を参照するかわからないので、寸法が重ならないようにしてください。



● 重複寸法

下図のような重複寸法を作成したとき、パラメータの値によっては結果が保証されません。

$A = B + C + D$ の場合は問題ありませんが、
 $A < B + C + D$ の場合はどのような図形になるか保証できません。

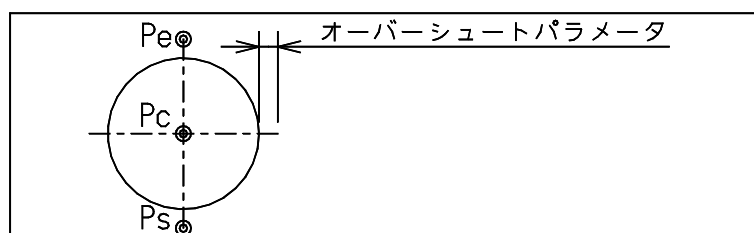


この場合は A を $B+C+D$ にするなど何かひとつを計算式にするべきです。
 また重複寸法のまま APG 登録をすると、どれかが未使用寸法となりエラー表示されます。

● 円中心線

パラメタライズできる円中心線の条件は、以下のとおりです。

- ・ 線種／線幅定数内の円中心線バンドルテーブルで定義されている線種番号と同じ線種番号を持つ線分であること。
- ・ 線分の中心点 (P_c) が円の中心点を通っていること。
- ・ 線分の長さが円中心線を作成した時と同一のパラメータ値であること。



● パラメタライズ時の許容差の設定

パラメタライズ時の許容差の設定は演算定数の最小長さ (MINDST) を参照しています。
 単精度浮動小数のデータベースを持つ CAD (たとえば CADAM や ME10 などの CAD) からデータを取り込んだ場合、図形と寸法引出線との間に誤差があり、有効精度が Advance CAD 側でパラメタライズできないケースがあります。このような場合には演算定数の最小長さを 0.1 程度に設定してからパラメタライズしてください。

● クラス番号と APG 出力の関係

クラス 255 で作成した図形と寸法は、寸法表示を ONN と指示して APG 図形配置を行なっても出力されません。

クラス 254 で作成した図形と寸法は、SKIP を指示すれば APG 図形配置を行なっても出力されません。

クラス 251 で作成した寸法は、APG 登録する時には無視します。ただし APG 配置時にその寸法に対応する値を計算し、その値が変数の場合にカリキュレータにその値を入力します。
 クラス 251 で使用できる寸法は、長さ寸法、角度寸法、半径寸法、直径寸法、座標寸法です。

1.2.3 APG パラメータファイル

● APG パラメータファイルの作成方法

APG で図形を作成するとき、あらかじめパラメータの設定値を記述しておくファイルをパラメータファイルとよびます。同じパラメータを何度も使用するとき便利です。パラメータファイルは以下の規約にしたがってテキストエディタで作成します。

APG パラメータ ファイルのディレクトリとファイル拡張子は、ファイル ACAD.SET のキーワード **#APG:PAR#** で指定します。

たとえば指定がつぎのようになっていれば、ディレクトリ **/usr/acad/files** にファイル拡張子 **.API** をつけて作成します。

```
#APG:PAR#   APG parameter file  "/usr/acad/files/"      !.API!
```

第 1 カラムがスラッシュの行はコメント行です。

1 行に 1 つのパラメータだけを記述します。左辺が変数名、右辺は値 (数値または文字定数) または計算式です。

パラメータは上から順に実行されるので、計算式の変数を先に記述します。

つぎは、APG パラメータファイルの作成例です。

```
/
/      filename:  TEST.API
/
ANG = 63.435
BOTTOM = 134.164
C = 180.000
D = 140.000
E = BOTTOM*0.5+C*SIN(ANG)
H = 30.000
/
/ end of data
```

APG パラメータファイルには、マクロの if 文を使用できます。

```
if (type == 1) {
    a = 10 ;
    b = 20 ;
} else {
    a = 20 ;
    b = 30 ;
}
```

1.3 コマンド

● コマンド一覧

コマンド名	機能
APGANAL	APG ファイルを作成する
APGPARAM	APG ファイルから図形を作成する
APGSET	APG ファイルから図形をドラッグモードで作成する
APG / DSP	APG を一覧図から選択する
APGREGEN	APG アイテムを更新する
APGTRACE	APG 登録時のデバッグ
APGADJUST	寸法値を変更し図形を再作成する
APGORG	APG ファイル作成時の図形寸法を作成する
APG __ PRESET	APG ファイル作成前に図形に定義条件を設定する
APG __ PREVER	図形に設定した定義条件の情報を表示する
APG __ PREDEL	図形に設定した定義条件を削除する
APG __ PRECLR	定義条件の白色表示部分を消す
APGBATCH	バッチ配置する

1.3.1 APG ファイルを作成する

【メニュー】

〔パラメトリック〕 → 〔登録〕

【構文】

APGANAL [APGA/FILE] name [[APGA/ORG] P]	$\left[\begin{array}{c} \text{USEACT} \\ \text{CUR} \end{array} \right] \langle \text{CE} \rangle$
--	---

ファイル名を指示する。

APGA/FILE name : APG ファイル名を入力する。

APG 原点を指示する。

APGA/ORG P : APG 原点をテンポラリポイントで入力する。省略するとモデル原点を使用する。

APG ファイルに登録する図形を選択する。

USEACT : アクティブリスト中のアイテムを選択する。

CUR : カレントピクチャにあるアイテムを選択する。(省略時)

登録を開始する。

<CE> : アイテムを解析し APG ファイルを作成する。

登録が終了すると、つぎのようなメッセージが表示される。

APG パラメタライズ終了 output-file-name

APG 未決定図形 m

APG 未決定寸法 n

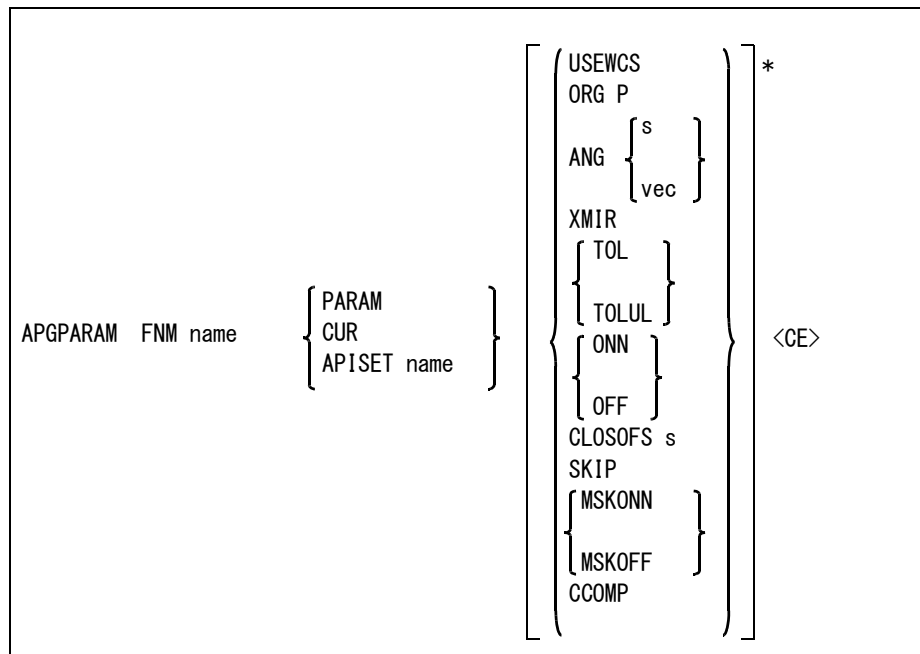
ここで m, n はアイテム数である。未決定図形数 m と未決定寸法 n はパラメタライズできなかったアイテム数で、画面では白色表示になっている。正しくパラメタライズできたときは、共に 0 になる。未決定となったアイテムを修正して APG 登録をしなければならない。

1.3.2 APG ファイルからの図形作成

【メニュー】

〔パラメトリック〕 → 〔配置〕

【構文】



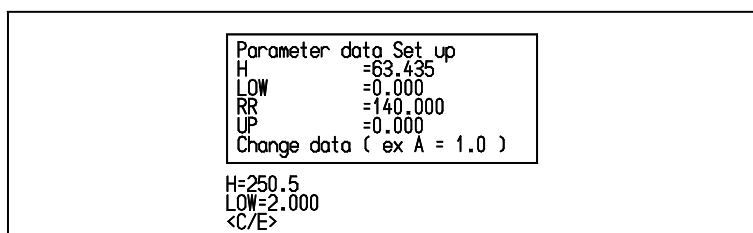
ファイル名を指示する。

FNM name : APG ファイル名を入力する。配置中に APG ファイルを再登録した場合は、ファイル名を入力し直すこと。(FNM の代わりに旧バージョンの APGP/NAME でも可)

寸法パラメータをつぎのいずれかの方法で指示する。

PARAM : この APG の変数名と現在値の一覧が画面に表示される。変更するパラメータの上をデジタイズする。または APG の形状と寸法も同時に表示されるので寸法をデジタイズしてもよい。続いて新しい値を入力する。変数名をキイインすることもできる。” 変数名=値 ” の形式で入力する。

変数名入力の代りに <CE> を入力すると終了する。



- CUR : APG ファイルを作成したときの寸法をそのまま使用する。
 APISET name : APG パラメータファイル名を入力する。パラメータファイルを読み込みそこに記述されたパラメータ値を使用する。

補助座標系 (WCS) を利用するとき指示する。省略時はカレント座標系に準ずる。

- USEWCS : 補助座標系を利用する。配置角度に補助座標系の + X 軸の角度、配置位置に補助座標系の原点を設定する。この指定を直接解除する方法はない。補助座標を解除し (WCS/OFF) もう一度入力するとカレント座標系を利用する。

配置位置を指示する。

- ORG P : APG を配置する位置をテンポラリポイントで入力する。省略するとモデル原点に配置する。

配置角度を指示する。

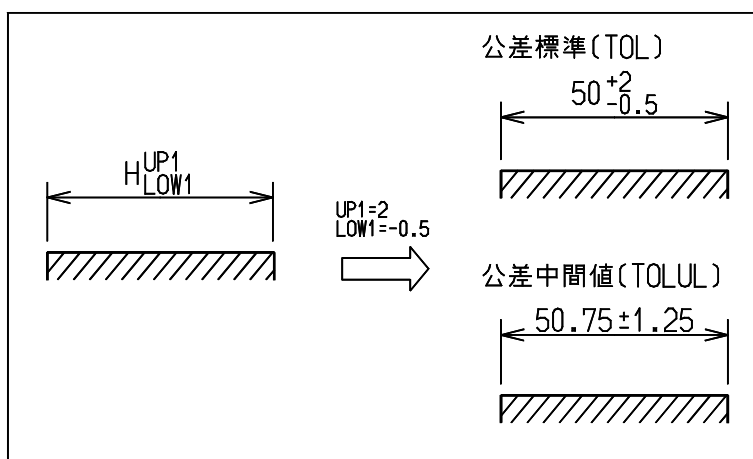
- ANG s, vec : 角度を数値かベクトルで入力する。補助座標系 (WCS) に基づいた角度を入力するなら ANG @DAs のようにする。これは補助座標系の + X 軸からの角度 s である。表示はカレント座標系 + X 軸からの角度である。省略すると 0。

APG を反転して配置するかどうか指示する。省略時は反転しない。

- XMIR : 補助座標系の Y 軸に対して反転する。もう一度入力すると反転しない。
 YMIR : 補助座標系の X 軸に対して反転する。もう一度入力すると反転しない。

寸法公差の処理方法を指示する。

- TOL : 寸法公差をそのまま表示する。(省略時)
 TOLUL : 寸法値は中間値 (寸法値 + (上公差 + 下公差) × 0.5) となり、公差は ± 表示になる。たとえば上公差を 2 下公差を -0.5 にした場合は下図のようになる。



寸法アイテムを作るかどうか指示する。

ONN : 寸法アイテムを作成する。
 OFF : 寸法アイテムを作成しない。(省略時)

クラスを変更するかどうか指示する。省略時は変更しない。

CLSOFSS : クラス加算値を入力する。元の図形のクラスに指定した数を加算したクラスを使う。クラス 255 は常に表示しないクラスであり、対象外である。クラス 254 は、SSKIP しないでアイテムを作成するなら対象になる。

クラス 254 のアイテムを作成するかどうか指示する。

SKIP : クラス 254 のアイテムを作成しない。もう一度入力すると作成になる。

選択マスクを使用するかどうか指示する。

MSKONN : 選択マスクを使用する。一時選択マスクとアクティブピクチャの表示マスクを参照する。寸法アイテムが選択されないと、ONN を指定しても寸法アイテムは作成しない。

MSKOFF : 選択マスクを使用しない。全アイテムを作成する。
 (省略時)

APG 配置してできたアイテムを APG アイテムにするかどうか指示する。省略時は APG アイテムにならない。

CCOMP : APG で作成される図形を APG アイテムにする。もう一度入力すると APG アイテムにしない。

APG を配置する。

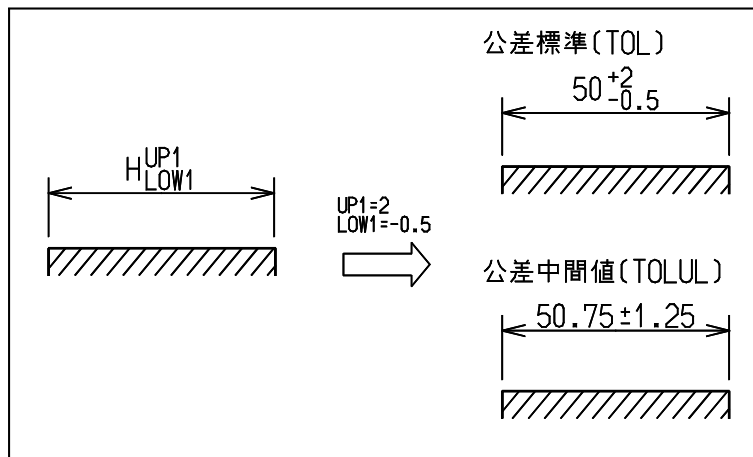
<CE> : APG 配置を実行すると、APG ファイル名に拡張子 .APP がついたファイルができる。このファイルには APG 配置を実行したときの変数の値が記録されている。

1.3.3 APG ファイルから図形をドラッグモードで作成する

【メニュー】

〔パラメトリック〕 → 〔SET〕

TOLUL : 寸法値は中間値 (寸法値 + (上公差 + 下公差) × 0.5) となり、公差は ± 表示になる。たとえば上公差を 2 下公差を -0.5 にした場合は下図のようになる。



クラス 254 のアイテムを作成するかどうか指示する。

SKIP : クラス 254 のアイテムを作成しない。もう一度入力すると作成になる。

選択マスクを使用するかどうか指示する。省略時は全アイテムを作成する。

MSKONN : 選択マスクを使用する。一時選択マスクとアクティブピクチャの表示マスクを参照する。寸法アイテムが選択されないと、ONN を指定しても寸法アイテムは作成しない。

MSKOFF : 選択マスクを使用しない。全アイテムを作成する。
(省略時)

クラスを変更するかどうか指示する。省略時は変更しない。

CLSOFSS : クラス加算値を入力する。元の図形のクラスに指定した数を加算したクラスを使う。クラス 255 は常に表示しないクラスであり、対象外である。クラス 254 は、SKIP しないでアイテムを作成するなら対象になる。

APG 配置してできたアイテムを APG アイテムにするかどうか指示する。省略時は APG アイテムにならない。

CCOMP : APG で作成される図形を APG アイテムにする。もう一度入力すると APG アイテムにしない。

ドラッグするデータのタイプを指示する。

Advance CAD バージョン 4.4 よりも前に作成した APG ファイルは、APGSET で配置できないことがある。このようなときは、APG ファイルを登録しなおすこと。

寸法アイテムを作る (ONN) が指定されても、ドラッグするデータには寸法アイテムは含めない。

DRGAUTO : 図形の外形だけをドラッグする。図形の形状に依存するが、外形がうまく取れなかったり、全く取れないこともある。

DRGITM : 図形全部をドラッグする。

DRGBOX : 図形全部を包みこむ最小矩形をドラッグする。

配置角度を指示する。

ANG s : 角度を数値で入力する。

ANG vec : 角度をベクトルで入力する。

APG を反転して配置するかどうか指示する。省略時は反転しない。

XMIR : 補助座標系の Y 軸に対して反転する。もう一度入力すると反転しない。

YMIR : 補助座標系の X 軸に対して反転する。もう一度入力すると反転しない。

APG を配置する。

- <CE> : このとき APG ファイル名に拡張子 (.APP) がついたファイルができる。このファイルには APG 配置を行ったときの変数の値が記録されている。
- P : APG 配置位置をテンポラリポイントで入力する。

1.3.4 一覧図から選択する

【メニュー】

〔パラメトリック〕 → 〔表示〕

【構文】

APG/DSP [BRK s] filename

表示数を変更するときに指定する。

- BRK s : 1 画面に表示する APG 数を入力する。s は 1 ～ 256 の整数。

表示する APG ファイル名を指定する。

- filename : 表示する APG ファイル名を入力する。ワイルドカードで指定できる。
指定した APG の形状が表示される。
APG 形状をマウスで選択すると、APG ファイル名が動作中のコマンドに渡される。

1.3.5 APG アイテムを更新する

【メニュー】

〔パラメトリック〕 → 〔更新〕

【構文】

APGREGEN ISapg	$\left[\begin{array}{l} \left(\begin{array}{l} \text{NME name} \\ \text{ORG p} \\ \text{ANG ang} \\ \text{XMIR} \\ \text{YMIR} \\ \left\{ \begin{array}{l} \text{PARAM} \\ \text{PARAMF name} \end{array} \right\} \\ \text{CUR} \end{array} \right) \\ \left\{ \begin{array}{l} \text{ONN} \\ \text{OFF} \end{array} \right\} \\ \text{SKIP} \\ \text{CCOMP} \\ \left\{ \begin{array}{l} \text{TOL} \\ \text{TOLUL} \end{array} \right\} \\ \text{CLSOFs s} \end{array} \right]$	+
		<CE>

更新する APG アイテムを指示する。

- ISapg : APG アイテムを選択する。

変更後の APG 名を指示する。別の APG にする。

- NME name : APG ファイル名を入力する。

配置原点を変更する。

ORG P : 配置点をテンポラリポイントで入力する。

配置角度を変更する。

ANG ang : 配置角を度で入力する。

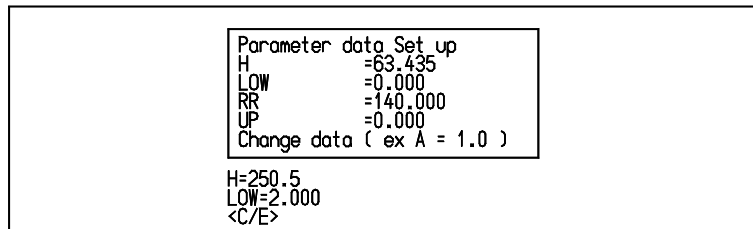
反転を指示する。

XMIR : 図形を APG 作成時の X 軸に対してミラーする。もう一度 XMIR を入力するとキャンセルする。

YMIR : 配置時の図形を APG 作成時の Y 軸に対してミラーする。もう一度 YMIR を入力するとキャンセルする。

寸法パラメータをつぎのいずれかの方法で指示する。

PARAM : この APG の変数名と現在値の一覧が画面に表示される。変更するパラメータの上をデジタイズする。または APG の形状と寸法も同時に表示されるので寸法をデジタイズしてもよい。続いて新しい値を入力する。変数名をキイインすることもできる。” 変数名=値 ” の形式で入力する。変数名入力の代りに <CE> を入力すると終了する。



PARAMF name : APG パラメータファイル名を入力する。そのパラメータファイルに記述されたパラメータ値を使用する。

CUR : APG 配置時の寸法を使用する。

寸法アイテムを作るかどうか指示する。

ONN : 寸法アイテムを作成する。

OFF : 寸法アイテムを作成しない。(省略時)

クラス 254 のアイテムを作成するかどうか指示する。

SKIP : クラス 254 のアイテムを作成しない。もう一度入力すると作成になる。

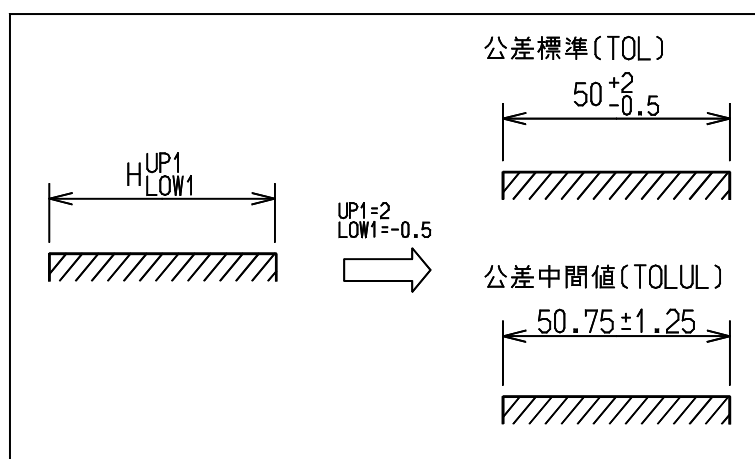
APG 配置してできたアイテムを APG アイテムにするかどうか指示する。省略時は APG アイテムになる。

CCOMP : 入力すると APG アイテムにしない。もう一度入力すると APG アイテムにする。

寸法公差の処理方法を指示する。

TOL : 寸法公差をそのまま表示する。

TOLUL : 寸法値は中間値（寸法値 + (上公差 + 下公差) × 0.5）となり、公差は ± 表示になる。たとえば上公差を 2 下公差を -0.5 にした場合は下図のようになる。



クラスを変更するかどうか指示する。

CLS0FS s : クラス加算数を入力する。元の図形のクラスに指定した数を加算したクラスを使う。

APG アイテムの更新をする。

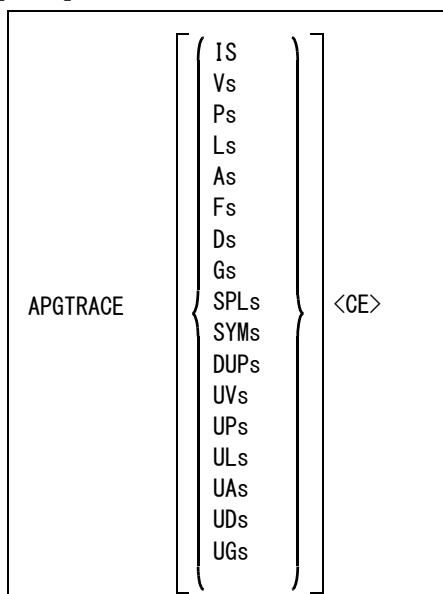
<CE> : APG の更新を実行する。この後、ここで設定した条件（修飾子）はすべてリセットされる。

1.3.6 APG 登録時のデバッグ

【メニュー】

〔パラメトリック〕 → 〔TRACE〕

【構文】



APG 登録時に未決定アイテムが残ったとき、その原因を調べるのに使用する。登録直後に実行すること。

調べたい図形の番号をつぎのいずれかの方法で指定する。Ps は P1, P2 ... を表す。他も同様。

IS	: 調べたいアイテムを選択する。
Vs	: ベクトルを表示する。
Ps	: 点を表示する。
Ls	: 線分を表示する。
As	: 円／円弧を表示する。
Fs	: APG が解析された順に表示する。
Ds	: 寸法を表示する。
Gs	: セグメントを表示する。
SPLs	: スプラインを表示する。
SYMs	: シンボル、サブモデルを表示する。
DUPs	: 使用していない寸法を表示する。

調べたい未決定図形の番号をつぎのいずれかの方法で指定する。

UVs	: 未決定なベクトルを表示する。
UPs	: 未決定な点を表示する。
ULs	: 未決定な線分を表示する。
UAs	: 未決定な円／円弧を表示する。
UDs	: 未決定な寸法を表示する。
UGs	: 未決定なセグメントを表示する。

調べたい図形の番号を入力すると、対応する図形とそれに関係する図形が白く表示される。そしてその図形の決定条件がメッセージウインドウに表示される。

例. “L1” と入力すると直線 #1 のデータをつぎのように表示する。

```
Lin 1      Pnt 2
LBP        Pnt 1
```

```
Lin 1 : この直線アイテムが APG に登録されたときの番号
LBP   : アイテムを作成したコマンド
Pnt 2 : 始点の番号
Pnt 1 : 終点の番号
```

例. “D1” と入力すると寸法 # のデータをつぎのように表示する。

```
Dim 1      Lin 4
Ldim        Lin 2
fixd
```

```
Dim 1 : この寸法が APG に登録されたときの番号
Ldim  : 寸法を作成したコマンド
fixd  : APG に登録されていることをしめす。
Lin 4  : 寸法の参照アイテム
Lin 2  : 寸法の参照アイテム
```

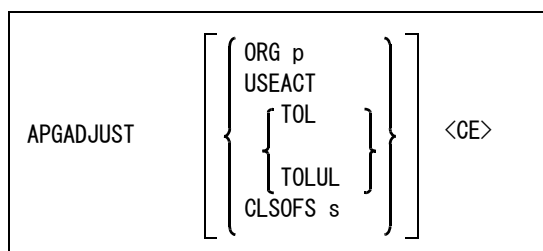
APG 登録で未定義になったものは、番号と "*****" が表示される。

1.3.7 既存の寸法値を変更し図形を再作成する

【メニュー】

〔パラメトリック〕 → 〔寸法調整〕

【構文】



寸法値を変更するには『寸法修正』を選ぶ。

基準点を指示する。

ORG P : 基準点をテンポラリポイントで入力する。

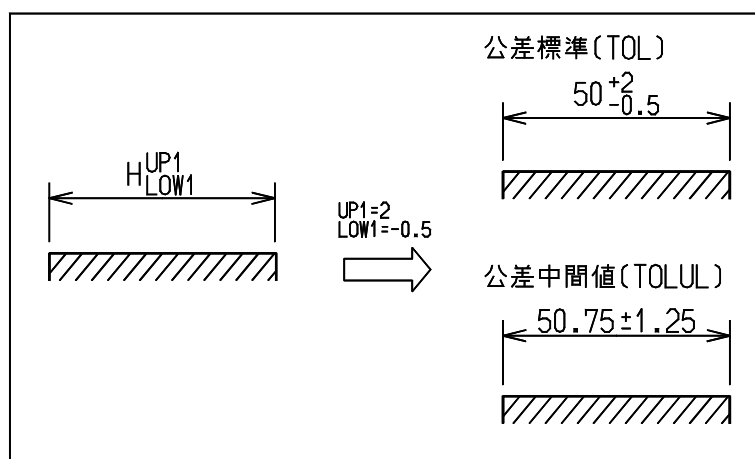
再作成するアイテムを指示する。省略するとアクティブピクチャのアイテムを使う。

USEACT : アクティブリスト中の図形、寸法を調整する。

寸法公差の処理方法を指示する。

TOL : 寸法公差をそのまま表示する。(省略時)

TOLUL : 寸法値は中間値 (寸法値 + (上公差 + 下公差) × 0.5) となり、公差は ± 表示になる。たとえば上公差を 2 下公差を -0.5 にした場合は下図のようになる。



クラスを変更するかどうか指示する。省略時は変更しない。

CLSOFs s : クラス加算数を入力する。元の図形のクラスに指定した数を加算したクラスを使う。

コマンドを実行する。

⟨CE⟩ : APGADJUST を実行する。

1.3.8 APG ファイル作成時の図形寸法をモデル中に作成する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕

【構文】

APGORG name <CE>

ファイル名を指示する。

name : APG ファイル名を入力する。

コマンドを実行する。

<CE> : APGORG を実行する。

このコマンドは APG ファイル作成時のモデルがなくなった場合に使用すると便利。

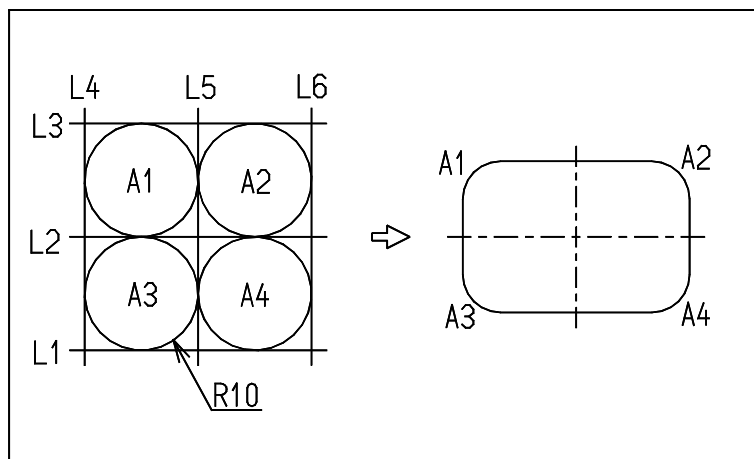
1.3.9 APG 条件設定

この節では、図形に対して定義条件をあらかじめ設定する手順を説明します。

APG 登録の時、アイテムの定義形式は自動決定されます。そのため、オペレータの意図とは異なる定義形式で、APG ファイルに図形が登録されてしまうことがあります。

APG_PRESET コマンドでは、特定の図形に対してあらかじめ定義条件を設定しておくことができます。したがって、図形に明示的に定義付けたい場合や、意に添わない定義が自動決定されるのを防ぎたいときに、このコマンドを使用します。

下図で、円 A1 を線分 L3 と L4 の接円として作図したとします。しかし、A1 は他に円 A2・A3 線分 L2・L5 にも接しています。APG の自動決定では A1 は最初に決定された 2 つの図形間の接円として定義されるため、オペレータの考えた L3・L4 の接円になるとは限りません。また、半径定義が無い場合には A1 は L2・L3・L4・L5 のいずれか 3 つに接する 3 接円として定義されてしまいます。それを防ぐために、A1 は半径 10 の L3・L4 の接円であることを、明示的に定義することが必要となります。



1.3.10 APG 条件を設定する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕

【構文】

```
APG_PRESET s [ IS ]+ <CE>
```

アイテムに定義形式を設定する。

s : APG プリセット番号を指定する。各プリセット番号、対応する設定条件は以下の通り。

番号	設定条件
200番台	点 条件
202	交点／接点
203	円中心点
208	直線の間接点
210	最短距離点
300番台	線 条件
302	平行線
303	2点間線
304	接線
400番台	円 条件
401	中心点と半径
402	接円
403	中心点と接図形
404	中心点と円上点
405	円上の3点
407	3接円

アイテムを指示する。

IS : 定義条件を設定するアイテムと、設定に必要な図形・寸法線などのサポートアイテムを指示する。

1.3.11 交点／接点を定義する

【メニュー】

〔パラメトリック〕→〔条件設定〕→〔交点／接点〕

【構文】

```
APG_PRESET 202 ISpoint { ISline1 } { ISline2 } { ISarc1 } { ISarc2 } <CE>
```

プリセット番号を指定する。

202 : 交点／接点を定義する番号を入力する。

条件を設定するアイテムを指示する。

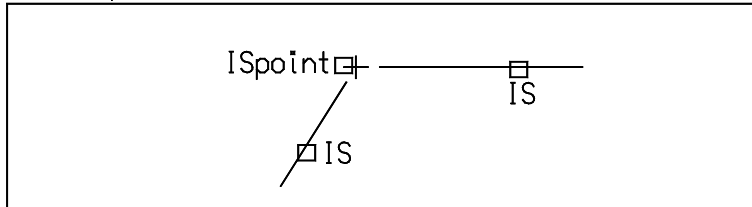
ISpoint : 交点／接点を定義する点を指示する。

設定に必要なアイテムを指示する。

ISline : 定義条件を満たす線分を指示する。

ISarc : 定義条件を満たす円弧を指示する。

202 ISpoint IS1 IS2 <CE>



1.3.12 中心点を定義する

【メニュー】

〔パラメトリック〕→〔条件設定〕→〔円中心点〕

【構文】

APG_PRESET 203 ISpoint ISarc <CE>

プリセット番号を指定する。

203 : 円中心点を定義する番号を入力する。

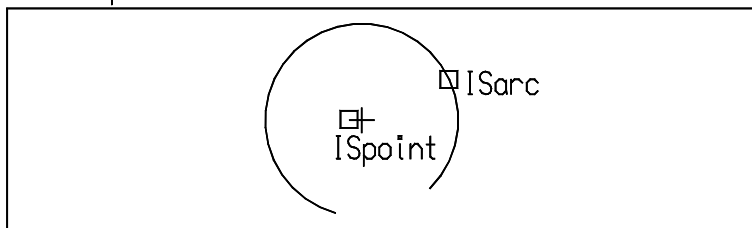
条件を設定するアイテムを指示する。

ISpoint : 円中心点を定義する点を指示する。

設定に必要なアイテムを指示する。

ISarc : 定義条件を満たす円弧を指示する。

203 ISpoint ISarc <CE>



1.3.13 中間点を定義する

【メニュー】

〔パラメトリック〕→〔条件設定〕→〔中間点〕

【構文】

```
APG_PRESET 208 ISpoint ISline <CE>
```

プリセット番号を指定する。

208 : 中間点を定義する番号を入力する。

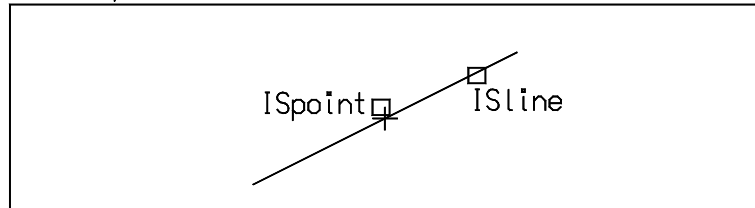
条件を設定するアイテムを指示する。

ISpoint : 中間点を定義する点を指示する。

設定に必要なアイテムを指示する。

ISline : 定義条件を満たす線分を指示する。

```
208 ISpoint ISline <CE>
```



1.3.14 最短距離点を定義する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕 → 〔最短距離点〕

【構文】

```
APG_PRESET 210 ISpoint ISgeom1 ISgeom2 <CE>
```

プリセット番号を指定する。

210 : 最短距離点を定義する番号を入力する。

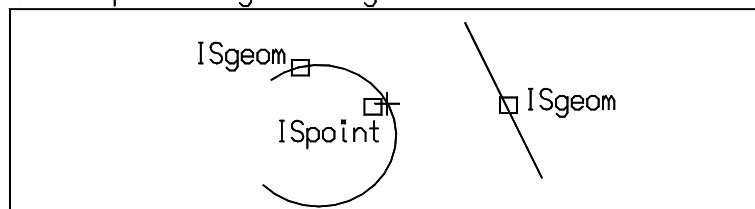
条件を設定するアイテムを指示する。

ISpoint : 最短距離点を定義する点を指示する。

設定に必要なアイテムを指示する。

ISgeom : 定義条件を満たすアイテムを指示する。

```
210 ISpoint ISgeom1 ISgeom2 <CE>
```



注) ISgeom1 は ISpoint が乗る図形（上図では円弧）を指示すること。
ISgeom2（上図では線分）を先に指示すると、ISgeom2 上の点を最短距離点として定義することになってしまう。

1.3.15 平行線を定義する

【メニュー】

〔パラメトリック〕→〔条件設定〕→〔最短距離点〕

【構文】

```
APG_PRESET 302 ISline1 ISline2 ISdim <CE>
```

プリセット番号を指定する。

302 : 平行線を定義する番号を入力する。

条件を設定するアイテムを指示する。

ISline1 : 平行線を定義する線分を指示する。

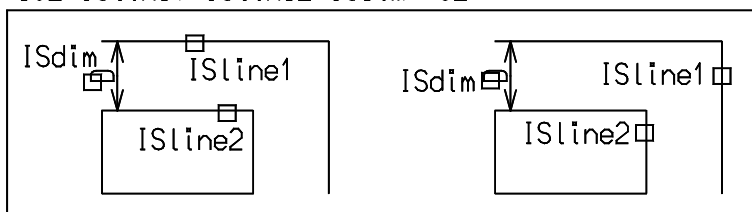
設定に必要なアイテムを指示する。

ISline2 : 定義条件を満たす線分を指示する。

オフセットする寸法値を指示する。

ISdim : オフセットする距離に対応する寸法を指示する。

302 ISline1 ISline2 ISdim <CE>



注) 図右の例のように、ISdim は定義する線分を直接指していなくても使用できる。
また ISline1 は ISline2 から ISdim 分オフセットした線分として定義されるため、
順序に注意して指示すること。

1.3.16 2点間線を定義する

【メニュー】

〔パラメトリック〕→〔条件設定〕→〔2点間線〕

【構文】

```
APG_PRESET 303 ISline ISpoint1 ISpoint2 <CE>
```

プリセット番号を指定する。

303 : 2点間線を定義する番号を入力する。

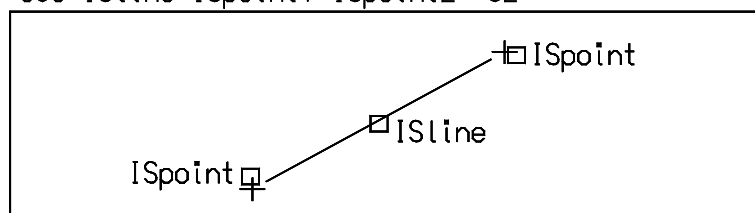
条件を設定するアイテムを指示する。

ISline : 2点間線を定義する線分を指示する。

設定に必要なアイテムを指示する。

ISpoint : 定義条件を満たす点を指示する。

303 ISline ISpoint1 ISpoint2 <CE>



1.3.17 接線を定義する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕 → 〔接線〕

【構文】

APG_PRESET 304 ISline ISarc ISgeom <CE>

プリセット番号を指定する。

304 : 接線を定義する番号を入力する。

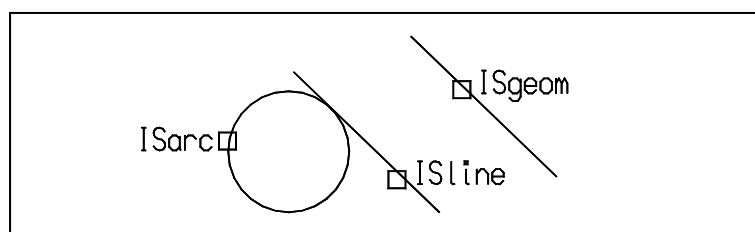
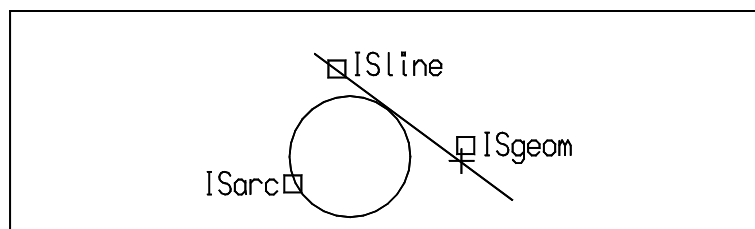
条件を設定するアイテムを指示する。

ISline : 接線を定義する線分を指示する。

設定に必要なアイテムを指示する。

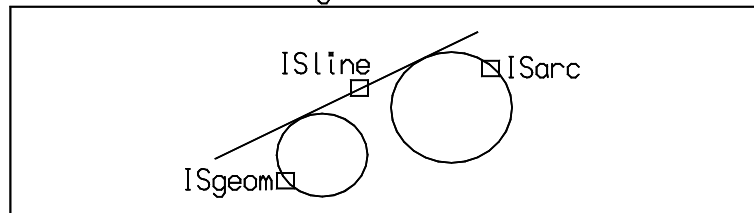
ISarc : 定義するアイテムに接する円弧を指示する。

ISgeom : 定義条件を満たすアイテムを指示する。



つぎの例では、ISgeom のベクトルを ISline (接線) のベクトルとして使用する。

```
304 ISline ISarc ISgeom <CE>
```



1.3.18 中心点と半径を定義する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕 → 〔中心点と半径〕

【構文】

```
APG_PRESET 401 ISarc ISpoint ISdim <CE>
```

プリセット番号を指定する。

401 : 中心点と半径を定義する番号を入力する。

条件を設定するアイテムを指示する。

ISarc : 中心点と半径を定義する円弧を指示する。

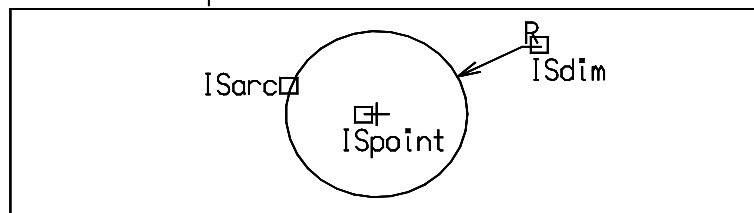
設定に必要なアイテムを指示する。

ISpoint : 定義条件を満たす点を指示する。

半径寸法を指示する。

ISdim : 半径値に対応する寸法を指示する。

```
401 ISarc ISpoint ISdim <CE>
```



注) ISdim は ISarc で指示した円弧を直接指していなくてもよい。

1.3.19 接円を定義する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕 → 〔接円〕

【構文】

```
APG_PRESET 402 ISarc ISgeom1 ISgeom2 ISdim <CE>
```

プリセット番号を指定する。

402 : 接円を定義する番号を入力する。

条件を設定するアイテムを指示する。

ISarc : 接円を定義する円弧を指示する。

設定に必要なアイテムを指示する。

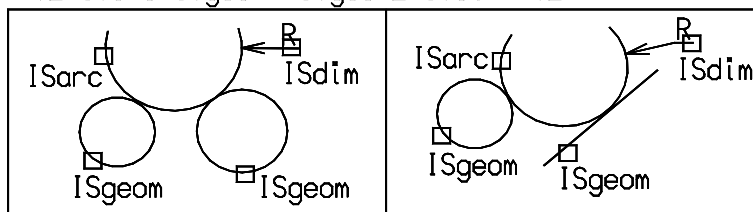
ISgeom : 定義条件を満たすアイテムを指示する。

半径寸法を指示する。

ISdim : 接円の半径値に対応する寸法を指示する。

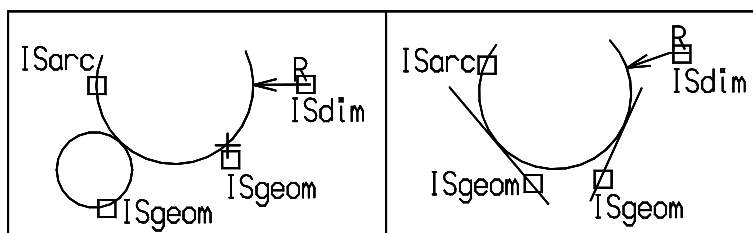
ISgeom の組み合わせは次の通り。指示する順序は問わない。

402 ISarc ISgeom1 ISgeom2 ISdim <CE>



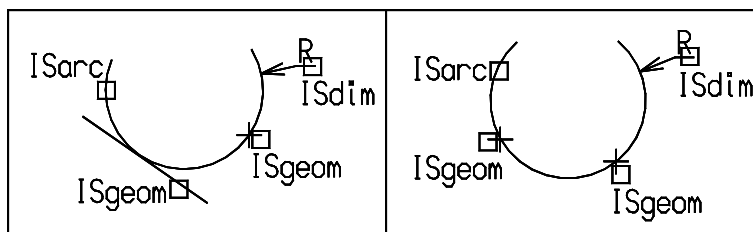
arc - arc

arc - line



arc - point

line - line



line - point

point - point

注) ISdim は ISarc で指示した円弧を直接指していなくてもよい。

1.3.20 中心点と接図形を定義する

【メニュー】

〔パラメトリック〕→〔条件設定〕→〔中心点と接図〕

【構文】

```
APG_PRESET 403 ISarc ISpoint ISgeom <CE>
```

プリセット番号を指定する。

403 : 中心点と接図形を定義する番号を入力する。

条件を設定するアイテムを指示する。

ISarc : 中心点と接図形を定義する円弧を指示する。

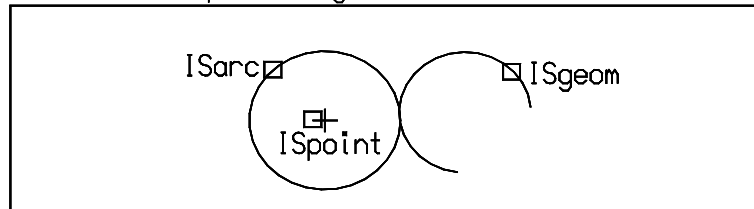
設定に必要なアイテムを指示する。

ISpoint : 円弧の中心点となる点を指示する。

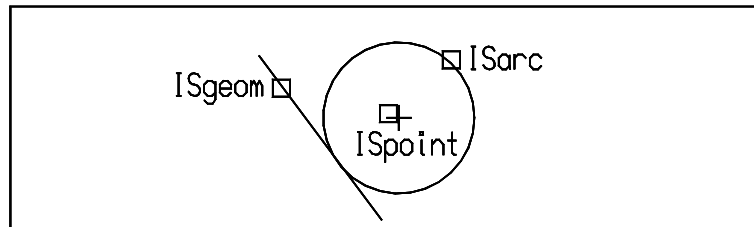
ISgeom : 定義条件を満たすアイテムを指示する。

ISgeom で指示できるアイテムは次の通り。

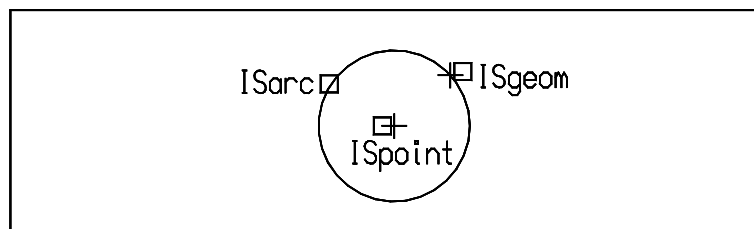
403 ISarc ISpoint ISgeom <CE>



arc



line



point

1.3.21 中心点と円上点を定義する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕 → 〔中心点と円上点〕

【構文】

```
APG_PRESET 404 ISarc ISpoint1 ISpoint2 <CE>
```

プリセット番号を指定する。

404 : 中心点と円上点を定義する番号を入力する。

条件を設定するアイテムを指示する。

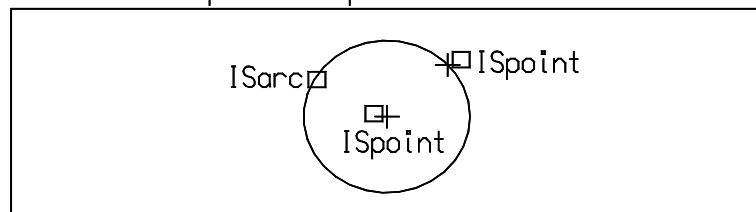
ISarc : 中心点と円上点を定義する円弧を指示する。

設定に必要なアイテムを指示する。

ISpoint : 円弧の中心点となる点を指示する。

ISpoint : 円上の点を指示する。

```
404 ISarc ISpoint1 ISpoint2 <CE>
```



1.3.22 円上の3点を定義する

【メニュー】

〔パラメトリック〕 → 〔条件設定〕 → 〔円上の3点〕

【構文】

```
APG_PRESET 405 ISarc ISpoint1 ISpoint2 ISpoint3 <CE>
```

プリセット番号を指定する。

405 : 円上の3点を定義する番号を入力する。

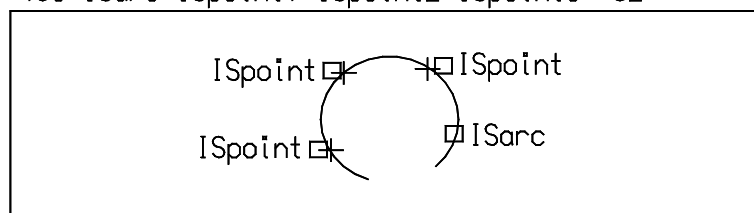
条件を設定するアイテムを指示する。

ISarc : 円上の3点を定義する円弧を指示する。

設定に必要なアイテムを指示する。

ISpoint : 円上の点を指示する。

405 ISarc ISpoint1 ISpoint2 ISpoint3 <CE>



1.3.23 3接円を定義する

【メニュー】

〔パラメトリック〕→〔条件設定〕→〔円上の3点〕

【構文】

APG_PRESET 407 ISarc ISgeom1 ISgeom2 ISgeom3 <CE>

プリセット番号を指定する。

407 : 3接円を定義する番号を入力する。

条件を設定するアイテムを指示する。

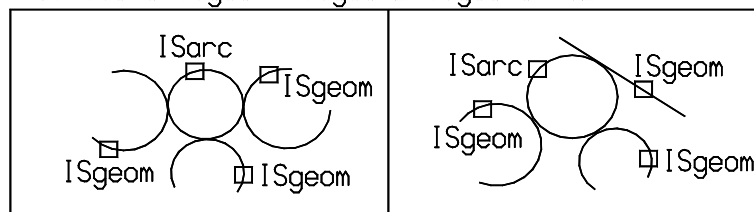
ISarc : 3接円を定義する円弧を指示する。

設定に必要なアイテムを指示する。

ISgeom : 定義条件を満たすアイテムを指示する。

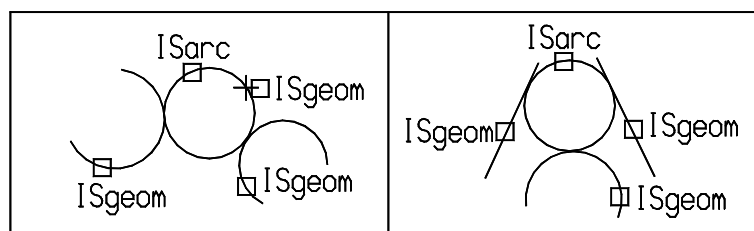
ISgeom の組み合わせは次の通り。指示する順序は問わない。

407 ISarc ISgeom1 ISgeom2 ISgeom3 <CE>



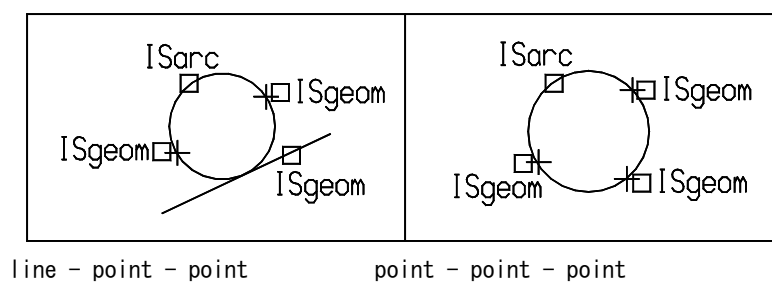
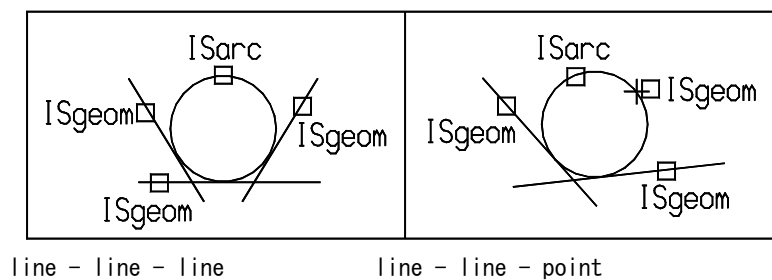
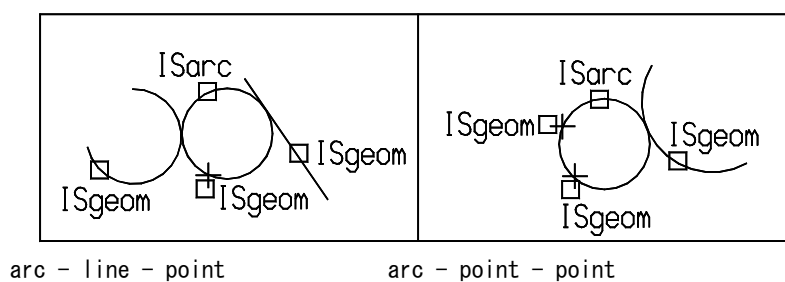
arc - arc - arc

arc - arc - line

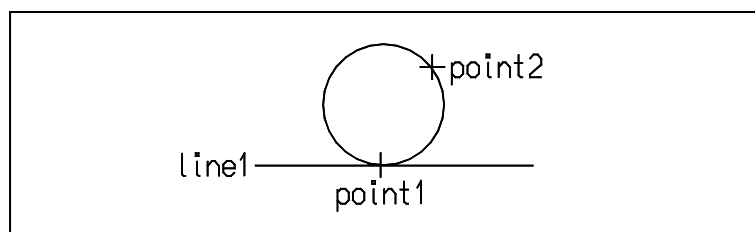


arc - arc - point

arc - line - line



下図のように線分 (line1) と線分上の点 (point1) と他の点 (point2) での3接円定義も可能。



1.3.24 APG 設定条件についての情報を表示する

【メニュー】

〔パラメトリック〕 → 〔条件 VER〕

【構文】

$$\text{APG_PREVER} \left\{ \begin{array}{l} \text{IS} \\ s \end{array} \right\} [\text{<CE> }]^*$$

アイテムを指示する。

IS : 指示したアイテムを白色表示し、定義条件をメッセージエリアに表示する。

設定条件を指示する。

s : 調べたい定義条件のプリセット番号を指定する。
 0 を入力すると、条件を定義した全アイテムを白色表示し、定義条件をメッセージエリアに表示する。
 200 を入力すると、点条件を定義した全アイテムを白色表示し、定義条件をメッセージエリアに表示する。
 300 を入力すると、線条件を定義した全アイテムを白色表示し、定義条件をメッセージエリアに表示する。
 400 を入力すると、円条件を定義した全アイテムを白色表示し、定義条件をメッセージエリアに表示する。
 上記以外のプリセット番号を入力すると、各条件を定義したアイテムを白色表示し、定義条件をメッセージエリアに表示する。

情報を順次得る。

<CE> : キーを押すごとに条件に対応するアイテムを順次白色表示し、定義条件をメッセージエリアに表示する。
 実線が表示されている図形は条件を設定したアイテムであり、点線表示のものはサポート（設定に必要な）アイテムである。

1.3.25 APG 設定条件を削除する

【メニュー】

〔パラメトリック〕 → 〔条件 DEL〕

【構文】

$$\text{APG_PREDEL} \left\{ \begin{array}{l} \text{IS} \\ s \end{array} \right\} \left\{ \begin{array}{l} [\text{<CE>}]^+ \text{ CUR} \\ \text{ALL} \end{array} \right\}$$

アイテムを指示する。

IS : 指示したアイテムの定義条件を削除する。

設定条件を指示する。

s : 削除する定義条件のプリセット番号を指定する。
 0 を入力すると、条件を定義した全アイテムを対象とする。
 200 を入力すると、点条件を定義した全アイテムを対象とする。
 300 を入力すると、線条件を定義した全アイテムを対象とする。
 400 を入力すると、円条件を定義した全アイテムを対象とする。
 上記以外のプリセット番号 を入力すると、各条件を定義したアイテムを対象とする。

設定条件を削除する。

<CE> CUR : 現在画面で白色表示しているアイテムに設定されている定義条件を 1 つずつ削除する。

- ALL : <CE> を押すごとに条件に対応するアイテムが順次白色表示される。
 : <CE> 押した直後に CUR を指示すると定義条件が削除され、続けて
 : <CE> を押すとそのアイテムはスキップして次の対象アイテムを白色表示する。
 : 対象アイテムの定義条件を全て削除する。

1.3.26 APG 設定条件の白色表示を消す

【メニュー】

〔パラメトリック〕 → 〔APG_PRECLR〕

【構文】

APG_PRECLR

注意

- APG_PRESET で設定した条件は絶対的なものです。そのため、他の方法であれば決定されるアイテムであっても、APG_PRESET 条件が満たされない限り、定義されません。すなわち、対象アイテムを定義するアイテムが未決定であると、対象アイテムも自動的に未決定となってしまいます。
- APG_PRESET 条件が設定できるアイテムは実データだけです。仮想点を対象とすることはできないので、条件設定のときには必ず、実際のアイテムとしての点を作成し、それを使用してください。
- スtringアイテム、複合アイテム等、かたまりとなっているデータは定義できません。したがってStringアイテム、複合アイテム中のセグメントを定義する場合、セグメントはクラス 255 のアイテムとして自動的に作成されます。条件設定は、その自動作成されたアイテムに対して行うこととなります。
- 1つのアイテムに設定できるのは1つの定義条件だけです。既に定義設定されているアイテムに条件を追加しようとするとエラーとなります。
- APG_PRESET は APG 登録する図形のアイテムすべてに対して定義する必要はありません。条件設定されていない図形は APG 登録の規則に基づいて定義されます。
- APG 登録時に APG 条件設定した図形は、必ずその定義条件を保持します。ただし、APGTRACE のメッセージと異なることがあります。これは、APG_PRESET ではベクトルを定義させていないためです。

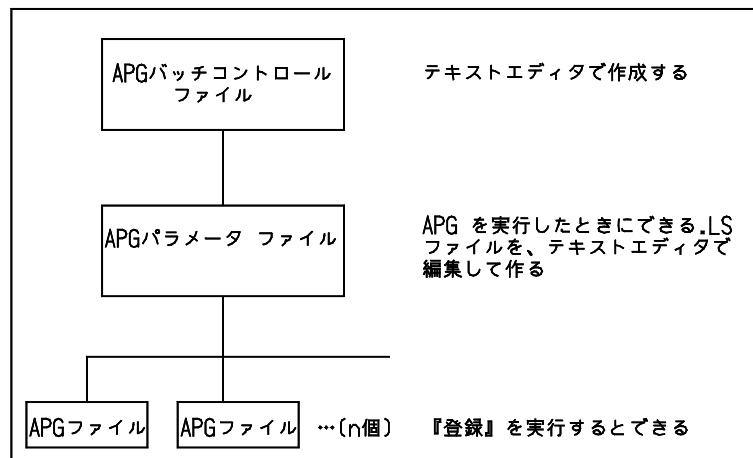
1.4 APG バッチ配置

本節では、APG をバッチで行う手順を説明します。

部品などの図形を最小単位で APG ファイル化しておき、これらの図形をアセンブルするまでの過程をバッチで行うのが APG バッチ配置です。

たとえば小さい部品 A, B, C, D を組み合わせて部品 E の図面を作る場合、つぎの手順で行います。最初に部品 A, B, C, D の APG 登録を行います。つぎに部品 A, B, C, D で使用するパラメータ全部を 1 つの APG パラメータファイルに記述します。そして部品 A, B, C, D をどのように配置するか、つまり APG 図形配置をどう行うかを、APG バッチコントロールファイルに記述します。最後に APG バッチ配置コマンドを実行します。

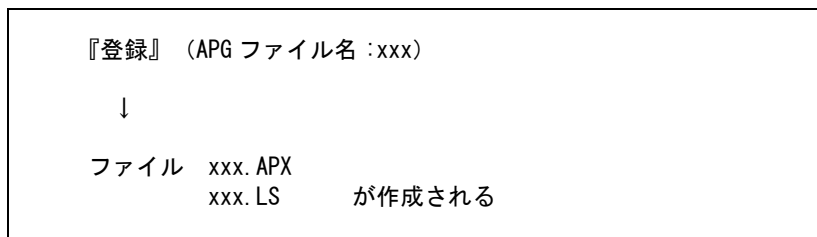
次図は APG バッチ配置のファイルの関係を示したものです。



1.4.1 APG パラメータファイルの作成方法

APG バッチで使用するパラメータ ファイルの作成規約・作成方法は、インタラクティブ処理のときにオプションで利用できるパラメータファイルと同じです。パラメータファイルは、なにもない状態からテキストエディタで作成してもよいのですが、つぎの方法で作成すると簡単です。

『登録』を実行すると、拡張子が .APX と .LS の 2 つのファイルが作成されます。



xxx.APX はパラメトリック図形ファイルです。

xxx.LS は設定した変数名と値が記録されたテキストファイルです。リスト1は.LS ファイルの例です。

この.LS ファイルをパラメータ ファイルを作るときの土台にします。つまり、バッチで配置したいいくつかの APG ファイルの.LS ファイルを1つのファイルにまとめ、重複するパラメータを省いてパラメータファイルにします。

```
xxx.LS
yyy.LS } → エディタで編集 → batchxxx.API
zzz.LS }
```

リスト2はパラメータ ファイルの例です。これは、サンプル図1～8の APG ファイルを作成したときにできた.LS ファイルをテキストエディタで編集したものです。

リスト1 (.LS)

```
*** parametric data output list ***
parameter      28  vector      4  point      72
extension      24  dimension   38  geometry   57  tangent    56
spline         0
fixup          207
BASE   =    100.000
BL     =     40.349
BR     =     25.489
DR1    =     25.291
DR2    =     32.246
DR3    =     38.048
KL     =     40.000
KR     =     30.000
RKL    =     55.000
RKR    =     15.000
RL1    =     24.745
RL2    =     30.968
RL3    =     34.270
SBOL   =     70.000
SBOLD  =      6.000
SBOLW  =     12.000
SBOLX  =      8.000
T      =      1.500
ZR     =      2.800
```

リスト2 (APG パラメータファイル)

```
/
/
/      APG Parameter file sample
/
BASE   =    150.000
RL1    =      20.
RL2    =     22.5
RL3    =     31.
RR1    =     10.
RR2    =     12.
```

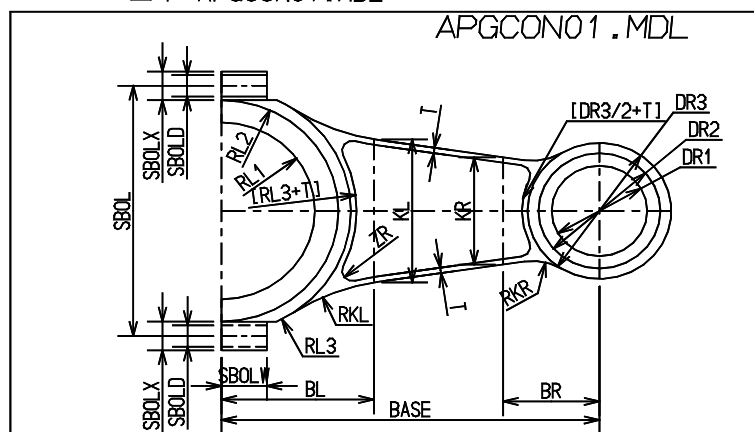
file: APGKUMIZU.API

```

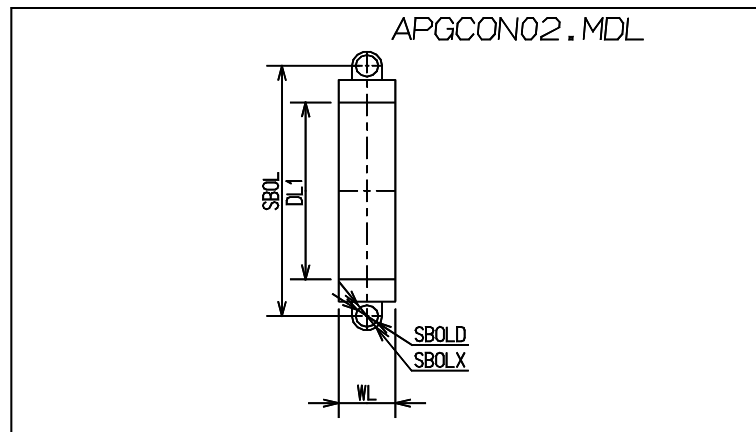
RR3      =      14.
DL1      =      40
DR1      =      20
DR2      =      24
DR3      =      28
/
/ Conection kubi portion
/
KL       =      20.0
KR       =      12.0
BL       =      32.5
BR       =      20
RKL      =      25.000
RKR      =      10.000
T        =      2.200
ZR       =      3.000
/
/ top view and Reft side view
/
WL       =      15.000
WR       =      16.000
TUP      =      7.300
TTR      =      2.200
/
/ Bolt portion
/
SBOL     =      60.000
SBOLW    =      10.000
SBOLD    =      5.000
SBOLX    =      8.000
BTOLER   =      0.175
BHEAD    =      3.0
SL       =      3.000
KUBL     =      4.000
WASH     =      0.100
NEJIW    =      0.75
/

```

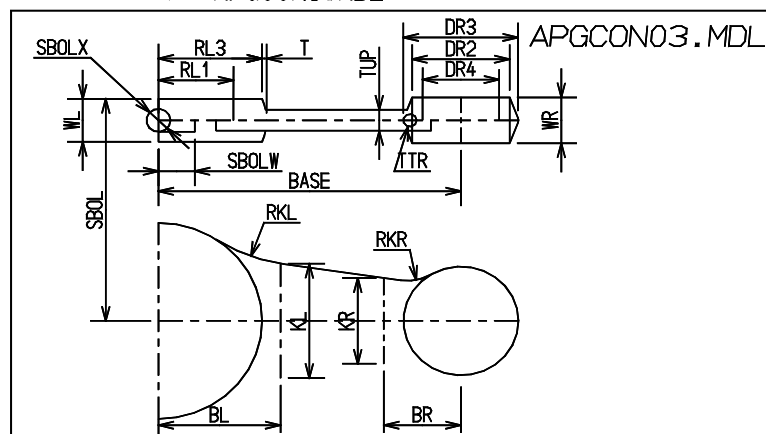
サンプル図 1 APGCON01.MDL



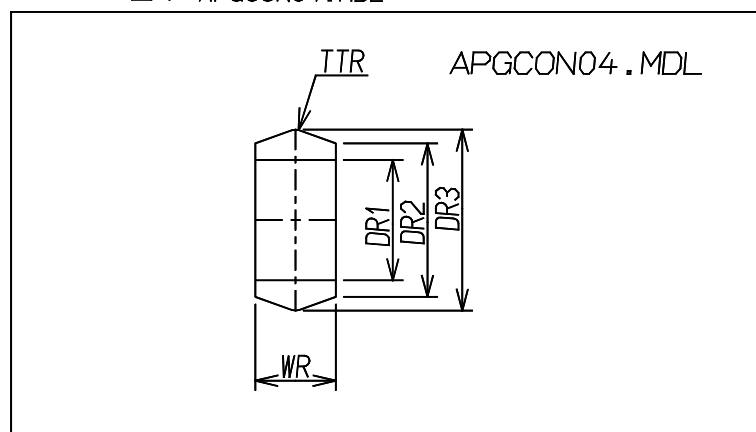
サンプル図 2 APGCON02.MDL



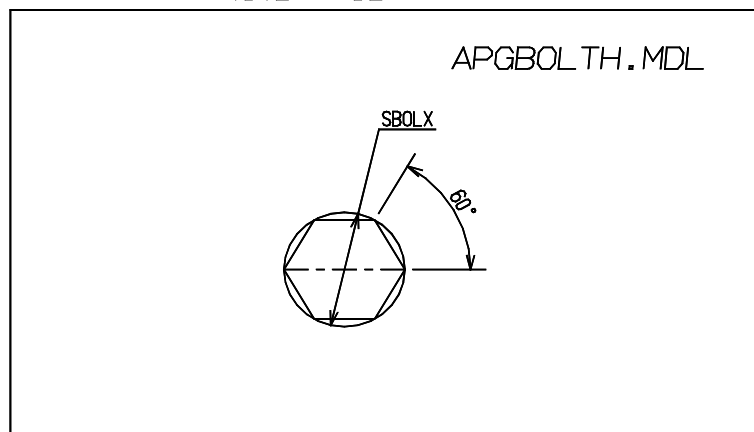
サンプル図 3 APGCON03.MDL



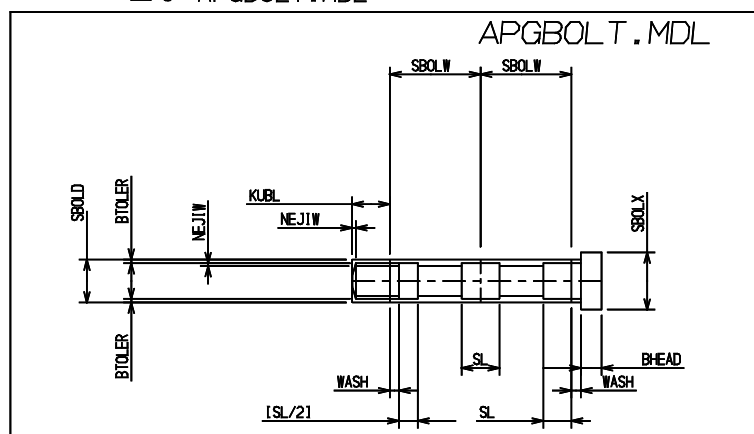
サンプル図 4 APGCON04.MDL



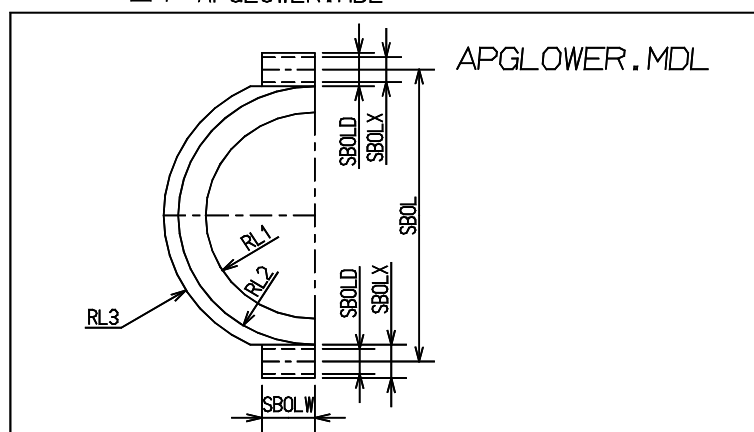
サンプル図 5 APGBOLTH.MDL



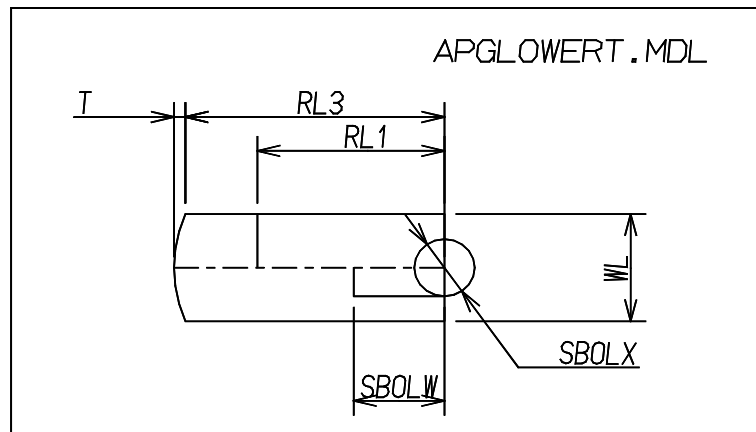
サンプル図 6 APGBOLT.MDL



サンプル図 7 APGLOWER.MDL



サンプル図 8 APGLOWERT.MDL



1.4.2 APG バッチコントロールファイル

リスト 3 の APGKUMIZU.APC を参考にして APG バッチコントロール ファイル (.APC) の作成方法を説明します。

● 作成規約

APG バッチコントロール ファイルは、ACAD.SET のキーワード #APG:APC# で指示したディレクトリに、指示した拡張子で作成します。デフォルトの拡張子は .APC です。

1 カラム目の / (スラッシュ) はコメント行を表します。

1 カラム目からの \$\$ はコントロール キーワードの設定を表します。API ファイルの \$\$ とは異なるので注意してください。

キーワード間は = (イコール) または , (カンマ) で区切ります。スペースとタブを含めてもかまいません。

● コントロールキーワード

1 カラム目から \$\$ で始まる次の英数字をコントロール キーワードと呼びます。

コントロール キーワードは 1 つの APG コントロール ファイルの中に 1 度だけ、つぎの順番で記述します。

```

$$ CONDITION
$$ PIC
$$ GO
$$ END

```

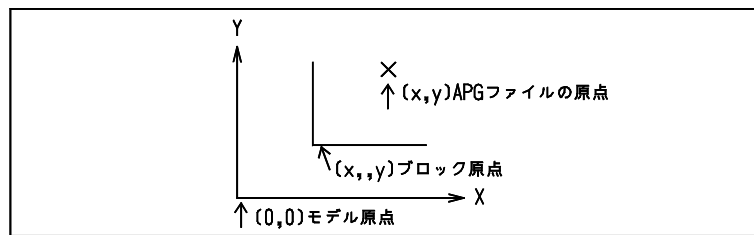
各コントロール キーワード中に以下に説明するキーワードのうち必要なものを記述し、APG バッチ処理の条件を設定します。

● ブロック

おたがいに関係を持っていない処理のかたまりをブロックと呼びます。

ブロックはおのこの原点を持っています。APG ファイルは、指定されたブロックの原点をモデルの原点と見なします。コントロール キーワード \$\$ CONDITION で指定できる 座標値 (APG ファイルの原点) は、このブロック原点からの相対座標値ということになります。

各ブロックの原点はコントロール キーワード `$$PIC` の X および Y で設定します。



またブロックごとに図形を出力するピクチャを指示できます (`$$PIC` の `OUTPIC`)。さらに指定したピクチャ + n 番のピクチャに出力することもできます (`$$GO` の `OFP`)。

● コントロール キーワードの種類と内容

n は任意の英数字です。

`$$ CONDITION`

条件設定を行います。

<code>APG=n</code>	:	呼び込む APG ファイル名。最大 40 文字。
<code>BLK=n</code>	:	出力ブロック番号。省略すると 1 になります。ブロック番号は 1 以上の整数で指示します。
<code>X=n</code>	:	X 原点座標値。省略すると 0 になります。
<code>Y=n</code>	:	Y 原点座標値。省略すると 0 になります。
<code>ROT=n</code>	:	回転角度 (degree)。省略すると 0 になります。
<code>CLS=n</code>	:	クラス番号。省略するとオリジナルクラス番号を使用します。
<code>FNT=n</code>	:	線種番号。省略するとオリジナルクラス番号を使用します。
<code>LWT=n</code>	:	線幅番号。省略すると線幅 1 を使用します。
<code>SKIP</code>	:	クラス 254 で作成したアイテムは出力しません。省略可。
<code>DIM</code>	:	寸法線を出力します。省略可。
<code>CMP</code>	:	作成される図形を APG アイテムにします。省略可。

`$$ PIC`

`$$ CONDITION` で設定したブロックについて設定します。`$$ CONDITION` で指定したブロックについて必ず記述しなければなりません。

<code>BLK=n</code>	:	ブロック番号
<code>X=n</code>	:	ブロック原点の X 座標値を指定します。
<code>Y=n</code>	:	ブロック原点の Y 座標値を指定します。
<code>OUTPIC=n</code>	:	そのブロックを指定してある APG ファイルを出力するピクチャの番号。

`$$ GO`

バッチ配置開始

<code>API=n</code>	:	実行する APG パラメータファイル名
<code>OFP=n</code>	:	<code>OUTPIC</code> のオフセット値。 <code>\$\$ PIC</code> の <code>OUTPIC</code> キーで指示した番号 + n 番のピクチャに指定図形を出力します。
<code>OFC=n</code>	:	出力図形および寸法線データのクラス番号のオフセット値。 <code>\$\$ CONDITION</code> の <code>CLS</code> キーで指示したクラス番号 + n 番のクラスで指定図形を作成します。

`$$ END`

終了

リスト3 (APG バッチコントロールファイル)

```

/
/
/      Apg batch mode sample control file
/
/      For Conecting rod
/
/ ( Main key word )
/
/ Col 1      /      comment line
/           $$      control
/           CONDITION      condition set
/           PIC          output BLK origine
/           GO           APG batch process execution start
/           END          Batch JOB end
/
/ $$ CONDITION key word
/           key word separator = ,
/           sp will skip    TAB will skip
/
/                                     Default Value
/
/      APG      Input APG file name      max 40 ch
/      BLK      Output Block number      1
/      X        APG data set origine x    0
/      Y        APG data set origine y    0
/      ROT      Rotation angle ( Degree )  0
/      CLS      Output class number      if omit use original class
/      FNT      Output font number      if omit use original font
/      LWT      Output line weight      if omit use 1
/      SKIP     Skip CLS number
/      DIM      Dimension output ( ACAD inside only )
/      CMP      Create Composit item
/
/ $$ PIC block offset condition
/      BLK      block data number
/      X        Offset x
/      Y        Offset y
/      OUTPIC   output actual pic number
/
/ $$ GO      key word
/      API      Input APG parameter file
/      OFP      OUTPIC number offset value
/      OFC      Output geom/dimension data class number offset value
/
/
/ Using APG file
/      APGCON01      Conecting rod body Front view
/      APGCON02      Left view
/      APGCON03      Top view
/      APGCON04      Right view
/      APGLOWER      Lower position Front view
/      APGLOWERT      Lower position Top view
/      APGBOLT       Bolt side view
/      APGBOLTH      Bolt Top view
/
/
/ $$ CONDITION
APG= APGCON01 , BLK=1 , X=0 , Y=0      , ROT=0, CLS=1 , SKIP
APG= APGLOWER , BLK=1 , X=0 , Y=0      , ROT=0, CLS=1 , SKIP
APG= APGCON02 , BLK=1 , X=-SBOL , Y=0   , ROT=0, CLS=1 , SKIP
APG= APGBOLTH , BLK=2 , X=0 , Y=0      , ROT=0, CLS=1
APG= APGCON03 , BLK=1 , X=0 , Y=SBOL    , ROT=0, CLS=1 , SKIP

```

```
APG= APGLOWERT, BLK=1 , X=0 , Y=SBOL , ROT=0, CLS=1 , SKIP
APG= APGCON04 , BLK=1 , X=[BASE+DR3*1.2], Y=0 , ROT=0, CLS=1 , SKIP
APG= APGBOLT , BLK=3 , X=0 , Y=0 , ROT=0, CLS=1
APG= APGBOLT , BLK=4 , X=0 , Y=0 , ROT=0, CLS=1 , SKIP
/
/ PIC condition
/
$$ PIC
    BLK = 1 , X= 0 , Y= 0 , OUTPIC = 1
    BLK = 2 , X=-SBOL , Y= [ SBOL/2] , OUTPIC = 1
    BLK = 3 , X= 0 , Y= [ SBOL/2] , OUTPIC = 1
    BLK = 4 , X= 0 , Y= [-SBOL/2] , OUTPIC = 1
/
/ Execution
/
$$ GO
    API = APGKUMIZU , OFP = 0 , OFC = 0
$$ END
```

1.4.3 バッチ配置する

【メニュー】

〔パラメトリック〕 → 〔バッチ〕

【構文】

APGBATCH file <CE>

file : APG バッチコントロールファイル名を入力する。
<CE> : APG バッチ処理を開始する。

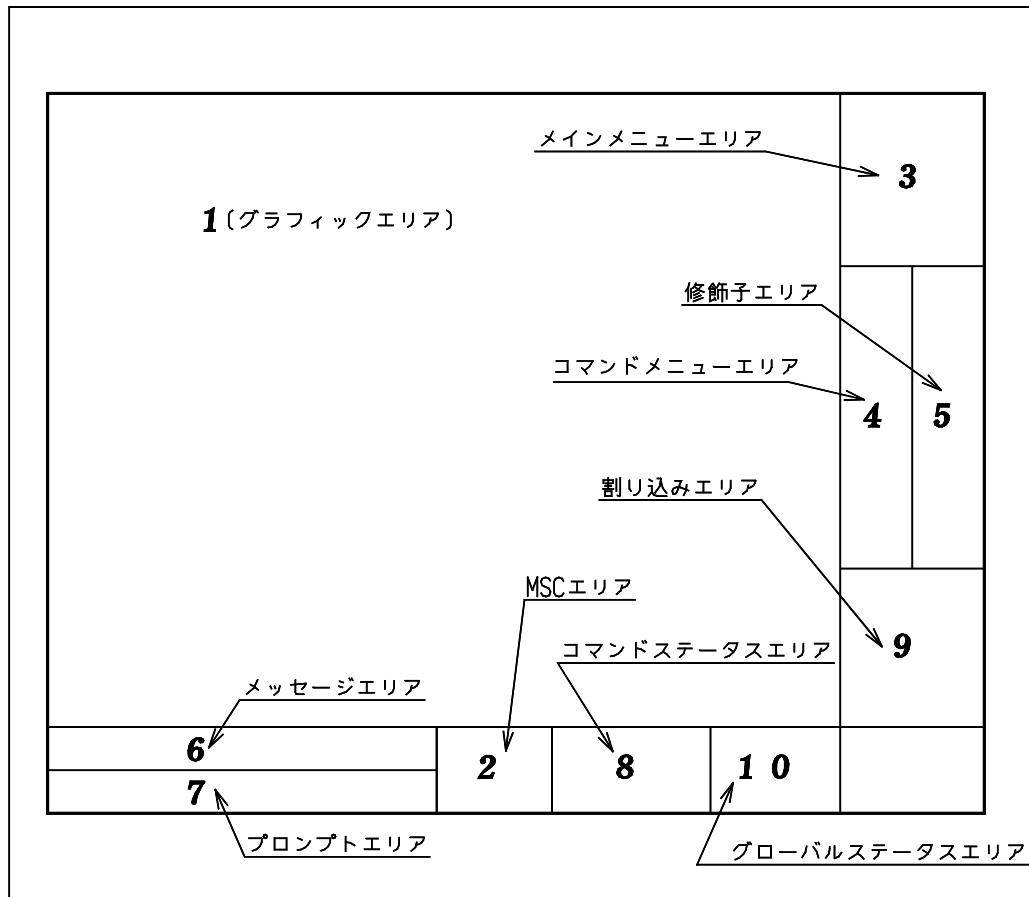
1.5 APG 実行例

この節では APG 登録と配置を、作成例を使って説明します。

説明の便宜上、画面の各部分を下図のように呼びます。

なお、10 グローバルステータスエリア中のテンポラリポイントを作成するコマンドが配置してあるところを特に「テンポラリポイント指示エリア」と呼びます。

『と』で囲んであるのはコマンド名またはメッセージです。

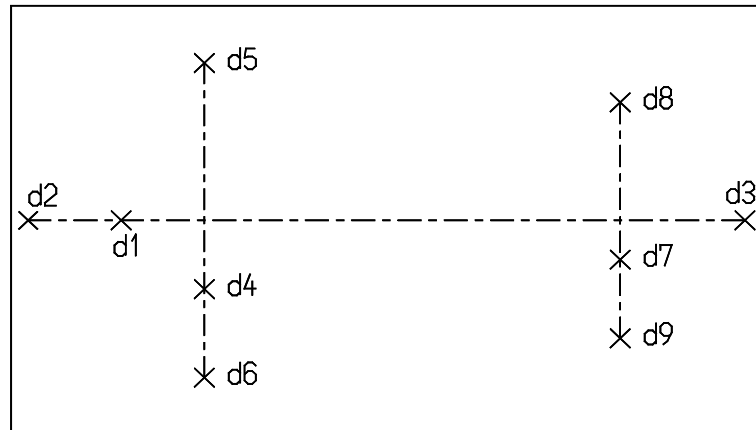


- (1) クラス 255 に設定する
デフォルトのクラスは 100 です。
クラス 255 で作成したアイテムは、APG を実行したとき出力しません。センター線分や補助寸法のように、APG 登録に必要でも APG 配置に不要なアイテムはクラス 255 で作成します。なおクラス 254 で作成すれば、出力させるかさせないかを APG 配置時に指定できます。
- (2) 線種を 3 に設定する
APG では、線種 3 の線分をセンターラインとして認識します。
線種 3 はデフォルトでは 1 点鎖線です。
- (3) メインメニューの「直線」を選択する。
すぐ下のコマンドメニューに直線を作成する各種のコマンドが表示されます。

- (4) コマンドメニューの〔水平線〕を選択する。
 センターラインにする水平線を作成します。『水平線』を選択すると、メッセージエリアに『LHL 基準点を入力』というメッセージが表示されます。
 最初の『LHL』はコマンド名です。つまり、キーボードから "LHL" とタイプしてもこれと同じコマンドを実行できます。日本語の部分は水平線を作成する方法を指示しています。このようにコマンドを選択したとき表示されるメッセージは、最初にコマンド名を表示し、続けてコマンドの実行方法を表示します。
 水平線の基準点をデジタイズするか、座標値を指示するかして水平線を作成します。ここではキーボードから "0,0" <CE> と入力します。
- (5) コマンドメニューの〔垂直線〕を選択する。
 センターラインにする垂直線を作成します。
 ここではキーボードから "0,0" "100,0" <CE> と入力します。
- (6) 全体を表示する
 初期状態では座標値 0,0 はグラフィックエリアの中心にあります。
 水平線および垂直線はグラフィックエリアからはみだしてしまう大きさで作成されますので、全体を表示するコマンドを実行します。
- 通常 MSC エリアには、現在の作業座標軸の方向および、つぎの2色の枠が表示されていますので参考にしてください。
 緑色の枠は現在表示している画面の範囲を示しています。
 赤い枠（ゾーン）は図面配置を行うときに図面に配置する範囲を示しています。
- (7) 作業しやすい範囲を表示する
 割り込みコマンドの『ズーム』メニューより『縮小』、『ズーム__倍率』コマンドなどで作業に適した範囲を画面に表示します。
- (8) トリムコマンドでセンターラインの長さを調整する
 トリムコマンドで範囲を指示するにはいくつかの方法があります。
 トリミングしたいアイテムと別のアイテムとの交点を求め、その位置までアイテムをトリミングできます。これが図形間のトリムです。図形間のトリムは『両端－図形』または『片側－図形』コマンドで行います。
 またポイントで範囲を指示すると、トリミングするアイテムに点を投影した位置までアイテムをトリミングできます。これがポイント間のトリムです。
- テンポラリポイント指示エリアの点をどれか選択すると点を指示できます。なおテンポラリポイント指示エリアの『自動』は、デジタイズした位置とアイテムとの距離を計算し、自動的に作成すべき点を選びます。選ぶ順は交点、端点、投影点、DIG 点の順になります。
 ここではテンポラリポイント指示エリアの『自動』を選択します。
 割り込みコマンドの『トリム』メニューより『トリム（一つずつ） 両端－点』を選択し、
 d1 d2 d3 d4 d5 d6 d7 d8 d9
 の順にデジタイズします。
 d1 で水平線がピックされ、d2, d3 点間の長さになるように修正されます。

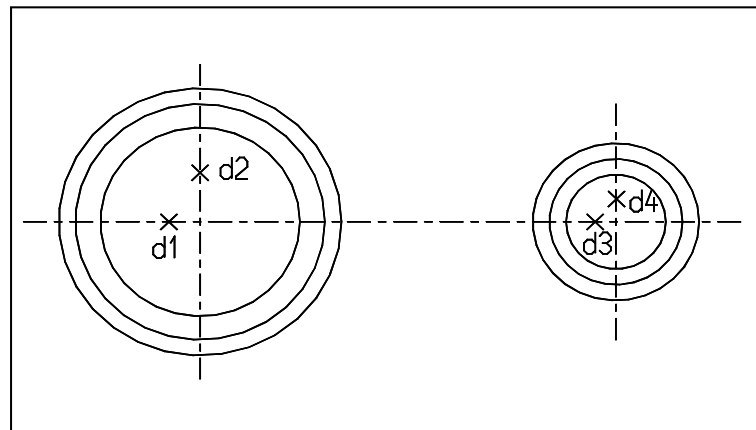
d4, d7 もおのの垂直線がピックされ、d5, d6 間、d8, d9 間の長さになるように修正されます。

(8) トリムコマンドでセンターラインの長さを調整する



- (9) クラスおよび線種を設定する
カレントクラスを 100 に設定します。
線種は 1 に設定します。
- (10) 円を作成する
メインメニューエリアの『円弧』メニューを選択し、円を作成します。
先に作成したセンターラインとセンターラインの交点を円の中心点にします。
テンポラリ ポイント指示エリアの『交点』を選択します。
d1, d2 の直線を指示して半径 24, 30, 34 の円を作成します。つぎに、d3, d4 の直線を指示して半径 12, 16, 20 の円を作成します。

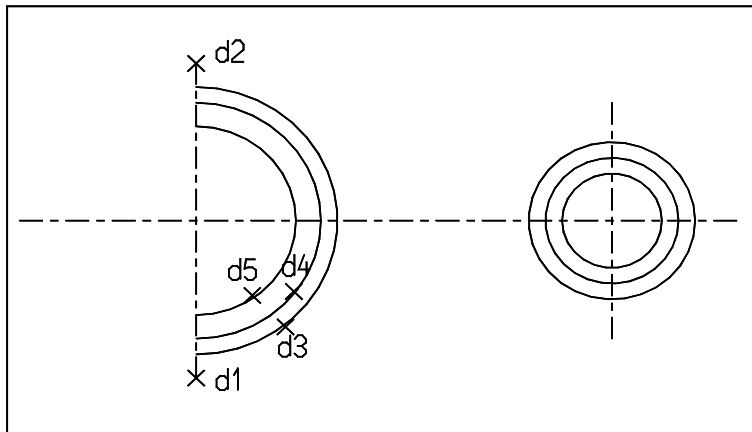
(10) 円を作成する



- (11) 円をトリムする
再度トリムコマンドを使用して、左側の円をトリムします。

トリムメニューの『マルチトリム 両端一図形』を選択し、d1, d2 をピックして範囲を設定します。また d3, d4, d5 をピックしてトリムする円を選択します。

(11) 円をトリムする



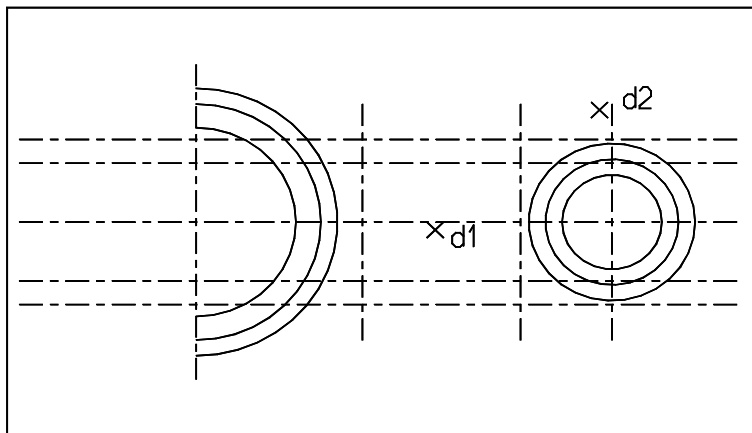
(12) 線分をオフセット

クラスを 255 に、線種を 3 に設定します。

オフセットコマンドまたは『直線』メニューの平行線コマンドを使用しセンター直線をオフセットします。

オフセット距離 15 を入力後、d1 の基準線をピックします。つぎにモディファイヤ『全候補』(ALL) を入力すると反対側にもオフセットされます。つづけてオフセット距離を 21 に変更し、再度 d1 の基準線をピックします。また『全候補』(ALL) を入力すると反対側にオフセットラインが作成されます。d2 の基準線にオフセットラインを作成するには、オフセット距離 22, 60 と順次入力します。

(12) 線分のオフセット



(13) 線分の作成およびトリム

クラスを 100、線種を 1 に設定します。

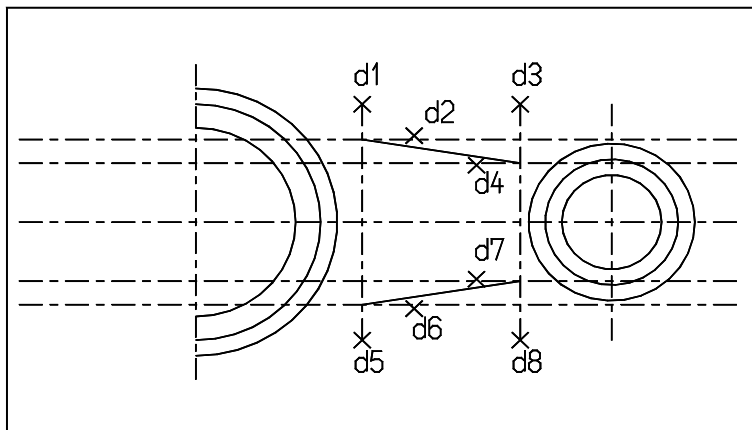
オフセットした直線の交点を使用して直線を作成し、不要な線分を削除します。

アイテム間で線分をトリムします。

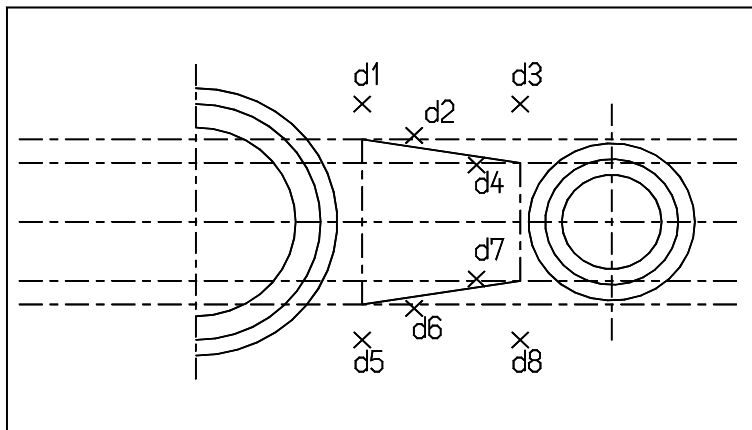
『2点間線』コマンドを使用します。d1, d2 および d3, d4 をピックしてそれぞれの交点間にラインを

作成します。d5, d6 および d7, d8 をピックすると同様に処理できます。
『トリム—両端図形』を使用して、d1 の線分を d2 と d6 で、d3 の線分を d4 と d7 でトリムします。

(13) 線分の作成およびトリム



(13) 線分の作成およびトリム(トリム実行後)



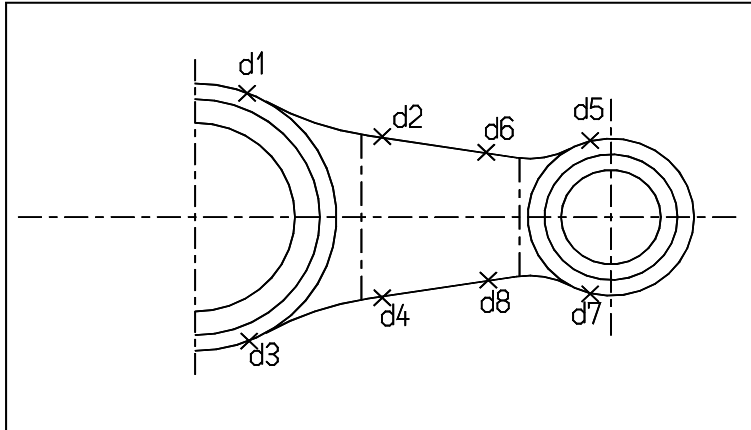
- (14) フィレットを作成する
フィレットコマンドを使用し円弧とラインの間にフィレットを作成します。
円弧側はトリムを実行しないでライン側はトリムを実行するように指示します。
2番目にピックしたアイテムはトリムしたいので、『STM』（セカンドトリム）モディファイアを指示します。

まず、「(12) 線分のオフセット」で作成したセンターライン以外の水平線を削除します。
それから、フィレットの大きさを半径値で指示します。左側の円弧には 70 の大きさを指示して、d1 で円弧をピックし、メニューのモディファイア『STM』（セカンドトリム）をヒットします。

d2 でラインをピックすると円弧はトリムされずにライン間のフィレットが作成されます。下部も同様に実行します。
つぎにフィレットの大きさを半径値 21 の大きさを指示して、d5 で円をピックし、メニューのモディファイア『STM』（セカンドトリム）をヒットします。

d6 でラインをピックすると左側の円弧と同じように円はトリムされずにライン間のフィレットが作成されます。下部も同様に実行します。

〔14〕 フィレットを作成する

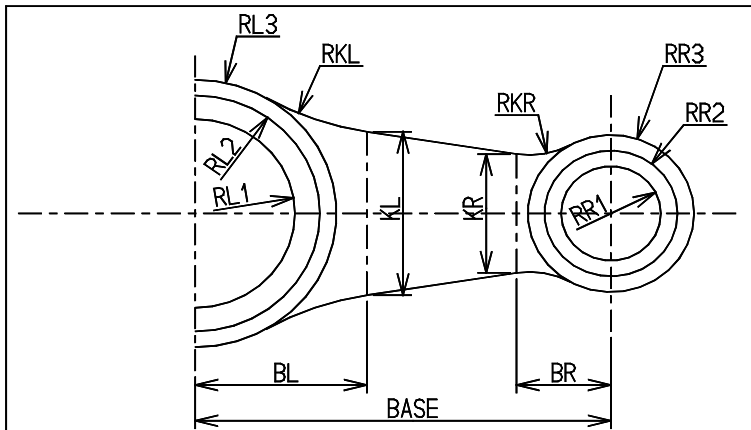


〔15〕 寸法アイテムを作成する

APG を使用するので、寸法値は変数におきかえてください。変数 KL および KR は線種 3 に対する寸法値ですので、水平のセンターラインの振分けとして認識されます。

『寸法 1』の『水平』『垂直』『半径』で寸法を作成します。
テンポラリポイントを端点にします。

〔15〕 寸法アイテムを作成する



〔16〕 APG 登録

メインメニューエリアの『APG』を選択します。
サブメニューエリアの『登録』を選択すると次のメッセージを表示します。メッセージに従って必要な情報を入力します。

- (1) ファイル名を入力
- (2) 原点を指定。デフォルトは (0, 0) です。
- (3) APG 出力

なお上記以外のオプションも設定できますがここでは省略します。
<CE>を入力すると、APG 登録を開始します。そのとき画面上ではアイテムが白く表示されます。
コマンドステータスエリアには、APG ファイル名と未決定寸法および未決定図形の数が表示されます。
未決定数が 0 なら APG ファイル作成は成功です。0 以外のときは寸法入力に誤りがあります。そのとき

は未決定部分が白く表示されたままの状態になります。このときは図形と寸法を再設定し、『登録』を行います。

補足：この例では未決定図形が3となります。円中心線である水平の一点鎖線（1本）と垂直の一点鎖線（2本）が白くなります。これらの3本の線分は長さが未決定です。対策は二通りあります。ひとつは、それらの長さを決定できるような寸法を追加することです。もうひとつの方法は、未決定線分をトリムして、その端点が決定可能な点になるようにします。たとえば、水平の一点鎖線の端点は左右の円の中心にします。垂直の一点鎖線は最も外側の円の直径になるようにトリムします。

(17) APG 配置

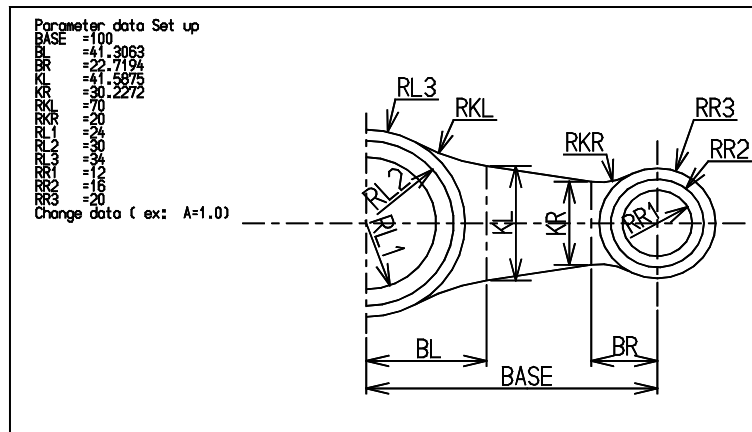
サブメニューエリアの『配置』を選択します。コマンドエリアに表示されるパラメータを設定してください。

最初にファイル名を指示します。

- (1) ファイル名入力
- (2) APG 原点指定。デフォルトは (0, 0) です。
- (3) パラメータ設定

パラメータセットを指示するとグラフィックエリアの左上に変数および現在値と図形が白色で表示されます。変更する変数を選択し値を入力します。

(17) APG 配置



たとえば、BASE = 130, KL = 45, RKL = 72, RL3 = 36 に変数の設定が終了したら、〈CE〉を入力します。あらかじめ値を APG パラメータファイル (.API) に設定しておく、そのファイル名を指示すれば、上記の操作をしなくてもパラメータファイルの値を自動的に呼出して値を設定します。

- (4) 寸法出力 ONN/OFF
ON/OFF どちらかを選択します。（デフォルトは OFF）
- (5) SKIP 254
クラス 254 のデータを出力しません。

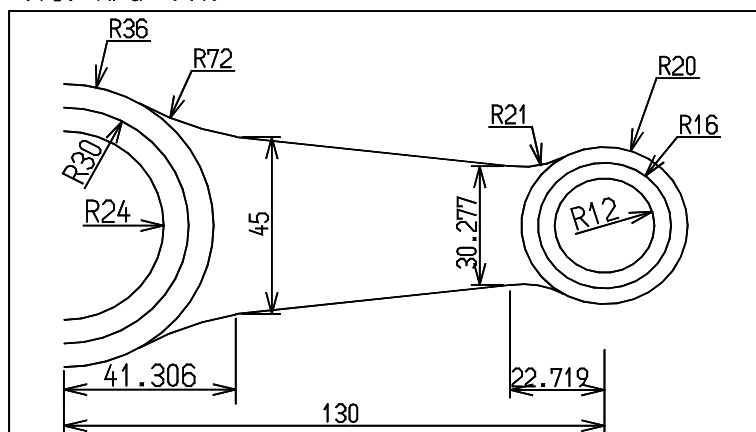
各種パラメータの設定が終了したら〈CE〉を入力して下さい。

(18) APG を実行する

下図に APG ファイル APG001 (.APX) の実行結果を示します。

変数におきかえられていた寸法は実寸で表示されます。クラス番号 254 の垂直線は作成されていません。

(18) APGを実行する



第 2 章 関係アイテム (associate item)

● 概要

関係アイテムは (associate item) アイテムのグループを定義するアイテムです。

複合アイテムやシンボルも複数のアイテムの集合ですが、これらは物理的に 1 つに固まっています。それに対して関係アイテムは、アイテムの論理的な固まりと言った方がよいでしょう。関係アイテムは自分の子供のアイテムがどれであることを記録しています。子供のアイテムは関係アイテムのメンバアイテムといい、独立しています。

逆にメンバアイテムも自分が属する関係アイテムがどれであることを記録しており、関係アイテムからメンバアイテム、メンバアイテムから関係アイテムがお互いわかるようになっています。

関係アイテムはメンバアイテムがどれであることを記録しているだけなので、関係アイテム自身は何も表示する図形データを持っていません。このため関係アイテムを指示するには関係アイテムの名前を入力するか、関係アイテムのメンバアイテムをピックするかどちらかです。

したがって、関係アイテムの名前は重要でモデル内では、必ずユニークでなければなりません。32 文字以内で、英文字で始まり英数字が続く名前にしてください。特に名前を指定しなければ、自動的にユニークな名前が付きます。

関係アイテムは最大 4096 のメンバアイテムを持つことができます。1 つの関係アイテムは異なるピクチャのアイテムをメンバとして持つことができます。この異なるピクチャのアイテムをメンバとして持てることは、たいへん重要です。関係アイテムの位置を移動するコマンドを使うと関係アイテムのメンバが移動しますが、このとき各メンバアイテムはピクチャのマトリクスに応じた移動をします。たとえば、X Y 平面上で @DX50@DY20 の移動をすると Z Y 平面のピクチャの図形は @DX50 の移動をします。

このように 1 つの関係アイテムのメンバアイテムはそれぞれ 3 次元モデルのように移動します。本来、関係アイテムは 2 次元 CAD システムで擬似 3 次元の設計ができるように考案されたアイテムです。

部品の 6 面図（平面、正面、下面、背面、右側面、左側面）をそれぞれ 6 つのピクチャで作図し、全体を 1 つの関係アイテムとします。そしてそれをモデルとして保存します。このようなモデルをいくつかサブモデルとして配置し組み立てます。このとき同じサブモデルを平面や側面に配置すると、いずれの場合も 6 つの射影図の関係が矛盾のないように配置する機能が必要になります。そしてサブモデルを移動するとき、3 次元的に矛盾なく射影図を移動する機能も必要です。これらの機能を実現するために Advance CAD では関係アイテムと関係アイテム用のコマンドがあります。

しかし、2 次元の図形で 3 次元を構築するため制限があります。

直交平面への射影以外は対応できないことがその 1 つです。それから、使用するピクチャの射影方向をすべて合わせておかねばなりません。

標準ではピクチャ 1 が平面図、ピクチャ 2 が正面図、以下ピクチャ 7 まで設定してありますが、このピクチャマトリクスを全モデルで統一しておかないと正しく機能しません。

Ver20 以降『APG 登録』でエラーとなったアイテムは配置原点を指定しても原点移動されなくなりました。

Ver19 以前と同じように原点を移動させたい場合は、修飾子 EIMV を使用して移動可能な状態にしてからコマンドを実行してください。

システム管理者の手引き『第 2 章 . 起動と環境設定』 - 『2.2 コンフィグレーションファイル』 - 『2.2.6 その他の指定』 参照

2.1 コマンド

● コマンド一覧

コマンド名	機能
ASC __ NEW	関係アイテムを作る
ASC __ ADD	関係アイテムにメンバを追加する
ASC __ REL	関係アイテムからメンバを除去する
ASC __ BRK	関係アイテムを分解する
ASC __ DEL	関係アイテムを削除する
ASC __ CNG	関係アイテムの名前を変更する
ASC __ VER	関係アイテムの確認
ASC / LIST	関係アイテムをリストする
ACT / ASC	関係アイテムをアクティブリストにのせる
ASC __ SET	関係アイテムを利用して部品を配置する
ASC __ SUB	関係アイテムからサブモデルを作る
ASC __ MOD	関係アイテムの属性を修正する
ASC __ MOVE	関係アイテムの位置を移動する
ASC __ STRETCH	関係アイテムのメンバアイテムを伸縮する
ASC __ SCALE	関係アイテムのメンバアイテムを縮小／拡大する

2.1.1 関係アイテムの部品配置のマトリックス表

```

/                                     file:ASSOCIPIC.TBL
/
/      部品配置に使用する、元のピクチャとアクティブモデルのピクチャ
/      の対応テーブル
/
/ ピクチャ名の定義。ピクチャ名は2文字とする。
$$ PIC
      PL  = 1
      FR  = 2
      RS  = 3
      LS  = 4
      BT  = 5
      RR  = 6
      I1  = 7
/
/      配置角度    0
/
$$ ANGLE = 0
/      平面      正面      右側面    左側面    下面      背面

```

```

/
/      PL      FR      RS      LS      BT      RR
/
PL = FR / 0 BT / 0 RS / 270 LS / 90 RR / 180 PL / 180 I1 / 0
FR = PL / 0 FR / 0 RS / 0 LS / 0 BT / 0 RR / 0 ** / 0
RS = PL / 90 LS / 0 FR / 0 RR / 0 BT / 270 RS / 0 ** / 0
LS = PL / 270 RS / 0 RR / 0 FR / 0 BT / 90 LS / 0 ** / 0
BT = RR / 180 PL / 0 RS / 90 LS / 270 FR / 0 BT / 180 ** / 0
RR = PL / 180 RR / 0 LS / 0 RS / 0 BT / 180 FR / 0 ** / 0
/
/      配置角度  90
/
$$ ANGLE = 90
/      平面      正面      右側面      左側面      下面      背面
/      PL      FR      RS      LS      BT      RR
/
PL = FR / 90 LS / 90 BT / 0 PL / 180 RR / 90 RS / 270
FR = RS / 90 FR / 90 BT / 90 PL / 90 LS / 90 RR / 270
RS = RS / 180 PL / 90 FR / 90 RR / 270 LS / 0 BT / 90
LS = RS / 0 BT / 90 RR / 270 FR / 90 LS / 180 PL / 90
BT = RR / 90 RS / 90 BT / 180 PL / 0 FR / 90 LS / 270
RR = RS / 270 RR / 270 PL / 90 BT / 90 LS / 270 FR / 90
/
/      配置角度  180
/
$$ ANGLE = 180
/      平面      正面      右側面      左側面      下面      背面
/      PL      FR      RS      LS      BT      RR
/
PL = FR / 180 PL / 180 LS / 90 RS / 270 RR / 0 BT / 0
FR = BT / 180 FR / 180 LS / 180 RS / 180 PL / 180 RR / 180
RS = BT / 270 RS / 180 FR / 180 RR / 180 PL / 90 LS / 180
LS = BT / 90 LS / 180 RR / 180 FR / 180 PL / 270 RS / 180
BT = RR / 0 BT / 180 LS / 270 RS / 90 FR / 180 PL / 0
RR = BT / 0 RR / 180 RS / 180 LS / 180 PL / 0 FR / 180
/
/      配置角度  270
/
$$ ANGLE = 270
/      平面      正面      右側面      左側面      下面      背面
/      PL      FR      RS      LS      BT      RR
/
PL = FR / 270 RS / 270 PL / 180 BT / 0 RR / 270 LS / 90
FR = LS / 270 FR / 270 PL / 270 BT / 270 RS / 270 RR / 90
RS = LS / 0 BT / 270 FR / 270 RR / 90 RS / 180 PL / 270
LS = LS / 180 PL / 270 RR / 90 FR / 270 RS / 0 BT / 270
BT = RR / 270 LS / 270 PL / 0 BT / 180 FR / 270 RS / 90
RR = LS / 90 RR / 90 BT / 270 PL / 270 RS / 90 FR / 270

```

2.1.2 関係アイテムを作る

【メニュー】

〔アソシエイト〕 → 〔作成〕

【構文】

ASC_NEW	$\left\{ \begin{array}{l} [\text{NME}] \text{ name} \\ \text{ONN} \end{array} \right\} \left\{ \begin{array}{l} [\text{IS}]+ \\ \text{USEACT} \end{array} \right\} \left[\text{GORG} \left\{ \begin{array}{l} \text{d1} [\text{d2}] \\ \text{X, Y, Z} \\ \text{X, Y} \end{array} \right\} \right] \langle \text{CE} \rangle$
---------	---

複数のアイテムをグループ関係にする。

関係アイテムの名前をつぎのいずれかの方法で指示する。

- NME name : 関係アイテムの名前を文字列で入力する。同じ名前の関係アイテムがすでにあるとエラーになる。
- ONN : 関係アイテムの名前を重複しないように自動的に決定する。

関係アイテムのメンバとするアイテムをつぎのいずれかで指示する。

- IS : アイテムを1つずつ選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- USEACT : アクティブリストを使う。

関係アイテムの配置位置を指示する。

- GORG d1 [d2] : d1 と d2 は別のピクチャでデジタイズする。この2つの2次元点座標から1つの3次元点座標を決定し、それが関係アイテムの配置位置となる。省略時は配置ピクチャは2、配置位置は (0, 0, 0) となる。
- GORG X, Y, Z : XYZ 座標値を数値入力して3次元点座標を決定する。3つの値が“,,” (カンマ) で区切られた場合に有効となる。たとえば、10, 20, 30 と入力すると、X=10, Y=20, Z=30 の点が配置位置とみなされる。
- GORG X, Y : カレントピクチャの座標値を入力して3次元点座標を決定する。

コマンドを実行する。

- 〈CE〉 : 関係アイテムを作成する。

2.1.3 関係アイテムにメンバを追加する

【メニュー】

〔アソシエイト〕 → 〔追加〕

【構文】

ASC_ADD	$\left\{ \begin{array}{l} \text{ISasc} \\ [\text{NME}] \text{ name} \end{array} \right\} \left\{ \begin{array}{l} [\text{IS}]+ \\ \text{USEACT} \end{array} \right\} \langle \text{CE} \rangle$
---------	---

関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- NME name : 関係アイテムの名前を文字列で入力する。選択された関係アイテムは白色表示される。

関係アイテムのメンバとして追加するアイテムをつぎのいずれかで指示する。

- IS : 追加するアイテムを1つずつ選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- USEACT : アクティブリスト中のアイテムを関係アイテムに追加する。

コマンドを実行する。

- 〈CE〉 : 関係アイテムのメンバを追加する。

2.1.4 関係アイテムからメンバを除去する

【メニュー】

〔アソシエイト〕 → 〔除去〕

【構文】

ASC_REL	$\left\{ \begin{array}{l} \text{ISasc} \\ [\text{NME}] \text{ name} \end{array} \right\} \left\{ \begin{array}{l} [\text{IS}]^+ \\ \text{USEACT} \end{array} \right\} \langle \text{CE} \rangle$
---------	--

関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- NME name : 関係アイテムの名前を文字列で入力する。選択された関係アイテムは白色表示される。

関係アイテムから除去するメンバアイテムをつぎのいずれかで指示する。

- IS : 取り除くメンバアイテムを1つずつ選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- USEACT : アクティブリスト中のアイテムを関係アイテムから取り除く。取り除かれたメンバアイテムは白色から元の色に戻る。バックスペースでキャンセルできる。取り除かれたアイテムは関係アイテムのメンバではなくなっただけで、削除はされない。

コマンドを実行する。

- 〈CE〉 : 関係アイテムのメンバを削除する。

2.1.5 関係アイテムを分解する

【メニュー】

〔アソシエイト〕 → 〔分解〕

【構文】

ASC_BRK	$\left\{ \begin{array}{l} \text{ISasc} \\ [\text{NME}] \text{ name} \end{array} \right\} \langle \text{CE} \rangle$
---------	---

関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- NME name : 関係アイテムの名前を文字列で入力する。選択された関係アイテムは白色表示される。

コマンドを実行する。

- 〈CE〉 : 関係アイテムを指示しただけでは分解しない。〈CE〉を入力した時点ではじめて分解する。関係アイテムのメンバアイテムは残るが、関係アイテム自身は削除され、その関係アイテムの名前もなくなる。

2.1.6 関係アイテムを削除する

【メニュー】

〔アソシエイト〕 → 〔削除〕

【構文】

ASC_DEL	$\left\{ \begin{array}{l} \text{ISasc} \\ \text{[NME] name} \end{array} \right\} \langle \text{CE} \rangle$
---------	---

関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- NME name : 関係アイテムの名前を文字列で入力する。選択された関係アイテムは白色表示される。

コマンドを実行する。

- 〈CE〉 : 関係アイテムを指示しただけでは削除しない。〈CE〉を入力した時点ではじめて削除する。関係アイテムと関係アイテムのメンバアイテムをすべて削除する。

2.1.7 関係アイテムの名前を変更する

【メニュー】

〔アソシエイト〕 → 〔名称変更〕

【構文】

ASC_CNG	$\left\{ \begin{array}{l} \text{ISasc} \\ \text{[NME] name} \end{array} \right\} \left[\begin{array}{l} \text{[NME] name2} \end{array} \right] \langle \text{CE} \rangle$
---------	--

関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- NME name1 : 関係アイテムの名前を文字列で入力する。選択された関係アイテムは白色表示される。

新しい名前を指示する。

- NME name2 : 新しい名前を入力する。

コマンドを実行する。

- 〈CE〉 : 関係アイテムの名前を変更する。

2.1.8 関係アイテムの確認

【メニュー】

〔アソシエイト〕 → 〔確認〕

【構文】

ASC_VER	$\left\{ \begin{array}{l} \text{ISasc} \\ [\text{NME}] \text{ name} \end{array} \right\}$
---------	---

部品配置で作成された関係アイテムについて、配置時の情報を表示する。配置ピクチャにピクチャ名が付いていればピクチャ名も表示する。

情報を表示する関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての〈SP〉入力で次候補アイテムに切り替わる。
- NME name : 関係アイテムの名前を文字列で入力する。選択された関係アイテムは白色表示される。

このコマンドで表示される内容は、システムレジスタに保持される。

以下の情報がシステムレジスタに保持される。

- ・ 関係アイテムの名前
- ・ サブモデル名
- ・ 配置位置
- ・ 配置ピクチャ
- ・ 配置角度

2.1.9 関係アイテムをリストする

【メニュー】

〔アソシエイト〕 → 〔リスト〕

【構文】

ASC/LIST name <CE>

指定された関係アイテムの名前を表示する。

関係アイテムがメンバとしてサブモデルを含んでいるとき、そのサブモデル名も表示する。

表示する関係アイテムの名前を指定する。

- name : 表示する関係アイテムの名前を文字列で入力する。ワイルドカードを指定できる。

コマンドを終了する。

- <CE> : コマンドを終了する。

2.1.10 関係アイテムをアクティブリストにのせる

【メニュー】

〔ACT 選択〕 → 〔ASC〕

【構文】

ACT/ASC	$\left\{ \begin{array}{l} \text{ISasc} \\ [\text{NME}] \text{ name} \end{array} \right\}$	<CE>
---------	---	------

関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。
 NME name : 関係アイテムの名前を文字列で入力する。選択された関係アイテムは白色表示される。

コマンドを実行する。

- <CE> : 関係アイテムをアクティブリストにのせる。

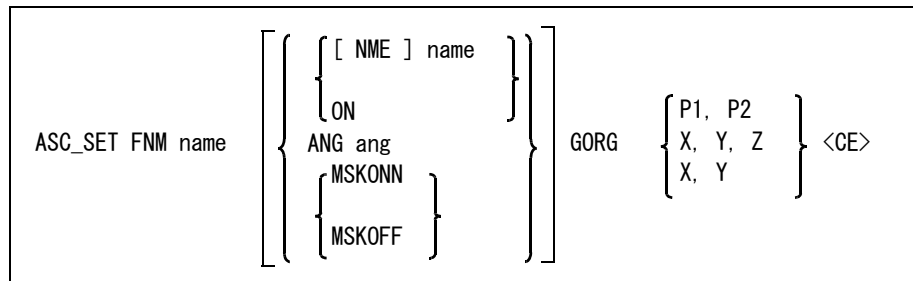
関係アイテムをアクティブリストにのせた場合、関係アイテム自身とそのメンバアイテムがアクティブリストにのる。

2.1.11 関係アイテムを利用して部品を配置する

【メニュー】

〔アソシエイト〕 → 〔部品配置〕

【構文】



サブモデル名を指示する。

- FNM name : 配置するサブモデル名を入力する。このサブモデルは関係アイテムで作られていないなければならない。

関係アイテムの名前をつぎのいずれかの方法で指示する。配置されたサブモデルの関係アイテムの名前はここで指定した名前に変更される。

- NME name : 関係アイテムの名前を文字列で入力する。
 ONN : 関係アイテム名を自動決定にする。(省略時)

配置角度を指示する。

- ANG ang : 配置角度は 0°、90°、180°、270° だけである。上記以外を入力しても無効。省略時は 0°。

選択マスクを使用するかどうか指示する。

- MSKONN : 選択マスクで選択可能なアイテムだけを対象とする。
 MSKOFF : 選択マスクを参照しない。(省略時)

配置位置を指示する。

- GORG P1 P2 : P1 と P2 は異なるピクチャでの点でなければならない。この2つの2次元点座標から1つの3次元点座標を決定する。P1 点を入力するとき使ったピクチャにサブモデルの正面図が表示されるようにサブモデルは3次的に回転して配置される。
 GORG X, Y, Z : XYZ 座標値を数値入力して3次元点座標を決定する。3つの値が“,“(カンマ)で区切られた場合に有効となる。
 GORG X, Y : カレントピクチャの座標値を入力して3次元点座標を決定する。

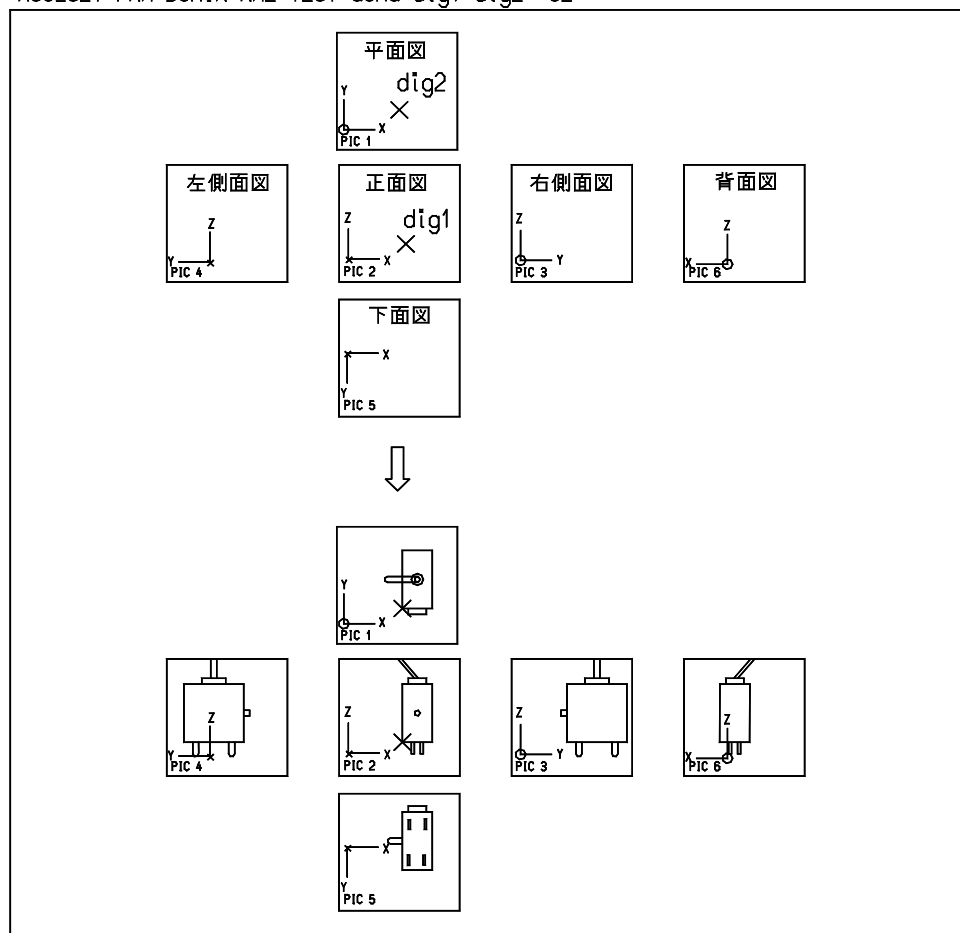
コマンドを実行する。

- <CE> : 部品を配置する。

例. スイッチボックスの投影図を配置すると下図のようになる。

正面図をピクチャ (2) に配置する。

ASC_SET FNM BUHIN NME TEST GORG dig1 dig2 <CE>

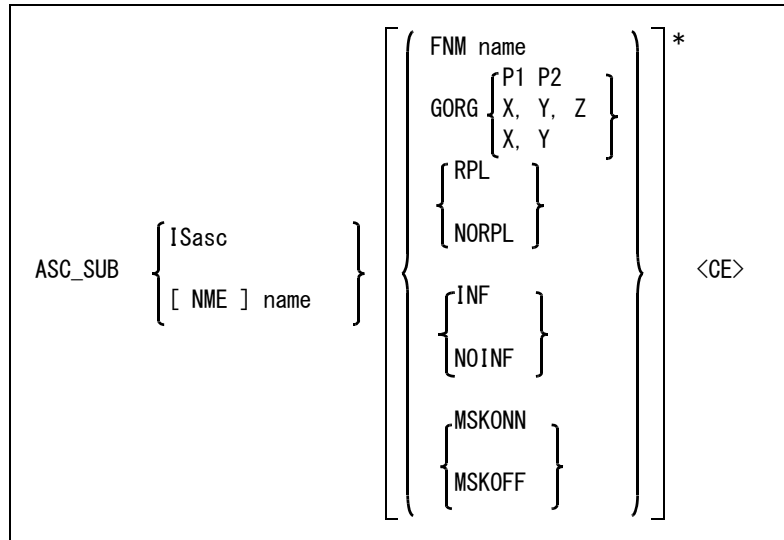


2.1.12 関係アイテムからサブモデルを作る

【メニュー】

〔アソシエイト〕 → 〔部品サブ出力〕

【構文】



図形要素が属しているピクチャごとに出力される。
各々のピクチャの原点は関係アイテムの 3 D 原点から計算した 2 D 原点が使用される。

関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。
- NME name : 関係アイテムの名前を入力する。選択された関係アイテムは白色表示される。

サブモデル名を指示する。

- FNM name : 選択した関係アイテムにすでにサブモデル名がついているときは省略できる。

原点を指示する。

- GORG P1 P2 : P1 と P2 は異なるピクチャでの点でなければならない。この 2 つの 2 次元点座標から 1 つの 3 次元点座標を決定する。
- GORG X, Y, Z : XYZ 座標値を数値入力して 3 次元点座標を決定する。3 つの値が“,“(カンマ)で区切られた場合に有効となる。
- GORG X, Y : カレントピクチャの座標値を入力して 3 次元点座標を決定する。

関係アイテムからサブモデルを出力したのち、関係アイテムを削除 (ASC_DEL) し、もう一度サブモデルを配置して関係アイテムを作り直す (ASC_SET) かどうか指示する。

- RPL : 関係アイテムに変換する。(省略時)
- NORPL : 関係アイテムに変換しない。

モデル情報ファイルを出力するかどうか指示する。

- INF : 情報ファイルを出力する。
- NOINF : 情報ファイルを出力しない。

選択マスクを使用するかどうか指示する。

- MSKONN : 選択マスクで選択可能なアイテムだけを対象とする。
- MSKOFF : 選択マスクを参照しない。(省略時)

コマンドを実行する。

- <CE> : サブモデルを作成する。

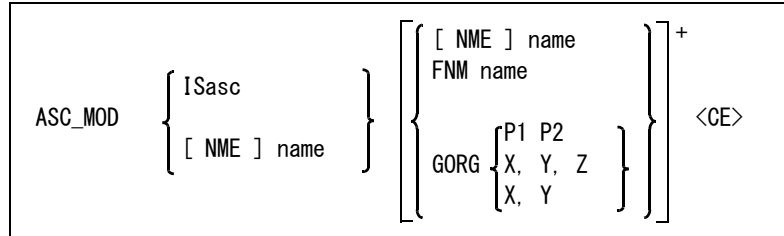
配置ピクチャ、配置角度、配置位置を変更して、ASC_SET した関係アイテムを ASC_SUB すると、元のサブモデル自身を登録する。つまり、配置ピクチャ、配置角度、配置位置を変更前 (配置位置 PIC2, 配置角度 0°, 配置位置 0,0,0) の状態に戻して登録する。

2.1.13 関係アイテムの属性を修正する

【メニュー】

〔アソシエイト〕 → 〔部品修正〕

【構文】



関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての<SP>入力で次候補アイテムに切り替わる。
- NME name : 関係アイテムの名前を入力する。選択された関係アイテムは白色表示される。

関係アイテムの名前、またはサブモデル名を変更する。

- NME name : 新しい関係アイテム名を入力する。
- FNM name : 新しいサブモデル名を入力する。

関係アイテムの配置位置を変更する。

- GORG P1 P2 : 関係アイテムの配置位置を変更する。P1 と P2 は異なるピクチャでの点でなければならない。この2つの2次元点座標から1つの3次元点座標を決定する。
- GORG X, Y, Z : XYZ 座標値を数値入力して3次元点座標を決定する。3つの値が“,,”(カンマ)で区切られた場合に有効となる。
- GORG X, Y : カレントピクチャの座標値を入力して3次元点座標を決定する。

コマンドを実行する。

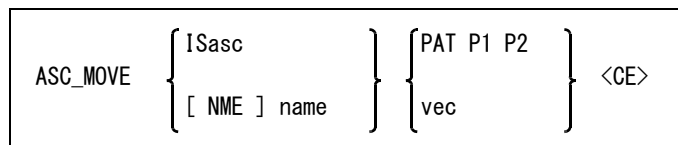
- <CE> : 関係アイテムの属性を修正する。

2.1.14 関係アイテムの位置を移動する

【メニュー】

〔アソシエイト〕 → 〔部品移動〕

【構文】



関係アイテムをつぎのいずれかの方法で指示する。

- ISasc : 関係アイテムのメンバアイテムを選択する。次候補アイテムがある場合は、続けての<SP>入力で次候補アイテムに切り替わる。
- NME name : 関係アイテムの名前を入力する。選択された関係アイテムは白色表示される。

移動量をつぎのいずれかで指示する。

- PAT P1 P2 : 移動量を2点で指示する。2点は同じピクチャ上で指示しなければならない。
- vec : 移動量をベクトルで指示する。指示したベクトル量はアクティブピクチャ上で計算される。

コマンドを実行する。

<CE> : メンバアイテムを移動する。関係アイテム自身は図形データを持たないので移動には関係しない。

2.1.15 関係アイテムのメンバアイテムを伸縮する

【メニュー】

〔アソシエイト〕 → 〔伸縮〕

【構文】

ASC_STRETCH	$\left\{ \begin{array}{l} \text{ISasc} \\ \text{[NME] name} \end{array} \right\} \left\{ \begin{array}{l} \text{PIC_ALL} \\ \text{PIC_SPC s} \end{array} \right\} \text{ORG} \left\{ \begin{array}{l} \text{X, Y, Z} \\ \text{P} \\ \text{X, Y} \end{array} \right\} \left\{ \begin{array}{l} \text{DIR_X} \\ \text{DIR_Y} \\ \text{DIR_I_Z} \end{array} \right\}$
	$\left\{ \begin{array}{l} \text{DIR_P s} \\ \text{DIR_M s} \\ \text{DIR_PM s} \\ \text{DIR_O} \end{array} \right\} \left\{ \begin{array}{l} \text{D_ONN} \\ \text{D_OFF} \end{array} \right\} \text{<CE>}$

このコマンドは X-Y, X-Z, Y-Z のピクチャだけに有効。

関係アイテムをつぎのいずれかの方法で指示する。

ISasc : 関係アイテムのメンバアイテムを選択する。
NME name : 関係アイテムの名前を入力する。選択された関係アイテムは白色表示される。

伸縮の対象とするピクチャを指示する。

PIC_ALL : 全てのピクチャの関係アイテムを伸縮の対象にする。
PIC_SPC s : 指定したピクチャの関係アイテムを伸縮の対象にする。

伸縮の基準点をつぎのいずれかで指示する。

ORG X, Y, Z : XYZ の座標値を数値入力し全ピクチャの基準点にする。
ORG P : デジタイズされたピクチャの位置を伸縮の基準点とする。
ORG X, Y : カレントピクチャの座標値を入力し基準点にする。

伸縮方向を指示する。

DIR_X : X 軸方向へ伸縮させたいとき指定する。
DIR_Y : Y 軸方向へ伸縮させたいとき指定する。
DIR_Z : Z 軸方向へ伸縮させたいとき指定する。

伸縮量を指示する。

DIR_P s : 指定した軸の基準点から＋方向にあるアイテムを伸縮する。
DIR_M s : 指定した軸の基準点から－方向にあるアイテムを伸縮する。
DIR_PM s : 指定した軸の基準点から＋－方向にあるアイテムを伸縮する。
DIR_O : 指定したアイテムの伸縮を取り消す。(数字のゼロ)

寸法アイテムの寸法値を更新するかどうか指示する。

D_ONN : 寸法値を更新する。
D_OFF : 寸法値を更新しない。(省略時)

コマンドを実行する。

<CE> : メンバアイテムを伸縮する。

2.1.16 関係アイテムのメンバアイテムを縮小／拡大する

【メニュー】

〔アソシエイト〕→〔縮小拡大〕

【構文】

ASC_SCALE	$\left\{ \begin{array}{l} \text{ISasc} \\ \text{[NME] name} \end{array} \right\} \left\{ \begin{array}{l} \text{PIC_ALL} \\ \text{PIC_SPC s} \end{array} \right\} [\text{ORG P}] + \left\{ \begin{array}{l} \text{MIR_X} \\ \text{MIR_Y} \\ \text{MIR_Z} \end{array} \right\}$
	$\left\{ \begin{array}{l} \text{SCL_X s} \\ \text{SCL_Y s} \\ \text{SCL_Z s} \\ \text{SCL_XYZ s} \\ \text{SCL_XYZ s1 s2 s3} \end{array} \right\} \left\{ \begin{array}{l} \text{D_ONN} \\ \text{D_OFF} \end{array} \right\} \text{<CE>}$

このコマンドは X-Y, X-Z, Y-Z のピクチャだけに有効。

関係アイテムをつぎのいずれかの方法で指示する。

ISasc : 関係アイテムのメンバアイテムを選択する。
NME name : 関係アイテムの名前を入力する。選択された関係アイテムは白色表示される。

縮小／拡大の対象となるピクチャを指示する。

PIC_ALL : 全てのピクチャの関係アイテムを伸縮の対象にする。
PIC_SPC s : 指定されたピクチャの関係アイテムを伸縮の対象にする。

縮小／拡大の基準点を指示する。

ORG P : 縮小拡大の基準点を指示する。

軸に対しての反転を指示する。

MIR_X : 基準点を中心に X 反転する。
MIR_Y : 基準点を中心に Y 反転する。
MIR_Z : 基準点を中心に Z 反転する。

各軸方向の倍率をつぎのいずれかの方法で指示する。

SCL_X s : X 軸方向の倍率を入力する。
SCL_Y s : Y 軸方向の倍率を入力する。
SCL_Z s : Z 軸方向の倍率を入力する。
SCL_XYZ s : X Y Z 軸方向への均一の倍率を数値で指定する。
SCL_XYZ s1 s2 s3 : X Y Z 軸方向への倍率をそれぞれ数値で指定する。

寸法アイテムの寸法値を更新するかどうか指示する。

D_ONN : 寸法値を更新する。
D_OFF : 寸法値を更新しない。(省略時)

コマンドを実行する。

<CE> : メンバアイテムを縮小／拡大する。

第3章 特性データ

● 概要

特性データは、アイテムに付ける非図形情報です。非図形情報というのは、品名、部品番号、材質、数量などの文字列または数値データのことです。

Advance CAD ではこれらのデータを特性データと呼んでいます。

特性データではこれらのデータをアイテムに付加しておき、部品表を作成したり、ラベルを表示したりすることができます。特性データは図形データではないので、通常は画面には表示しません。

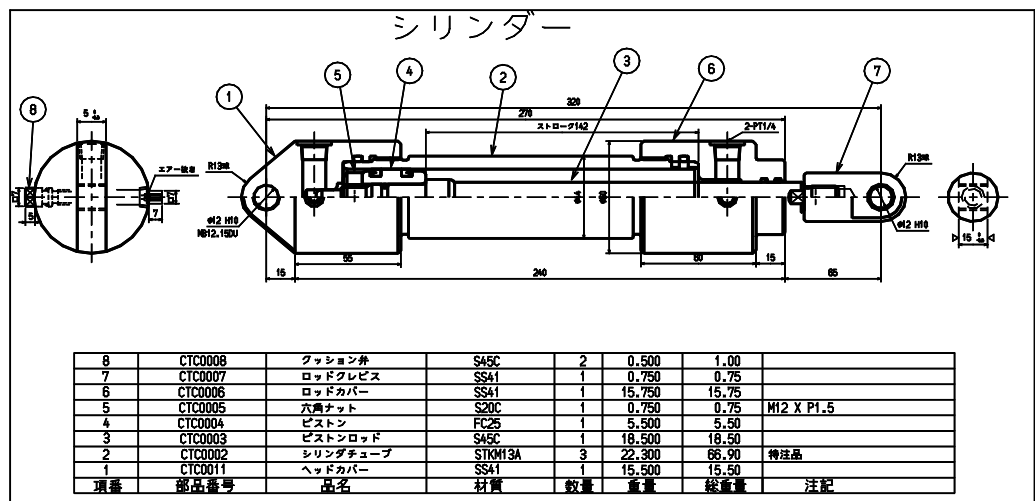
特性データの品名や部品番号などの項目は自由に変更できます。Advance CAD では一例として標準部品表とよばれる特性データを用意しています。

この特性データを変更して使用してもかまいませんし、また別の特性データを定義して使用してもかまいません。特性データの項目を定義しているのが特性ファイルです。

最初に特性ファイルの作り方を説明します。

つぎにこの特性ファイルを使って実際にアイテムに特性データを付加したり、集計表を作るコマンドを説明します。

特性データを利用したオプションソフトウェアとして、シーケンス図、P & I D、およびパーツがあります。

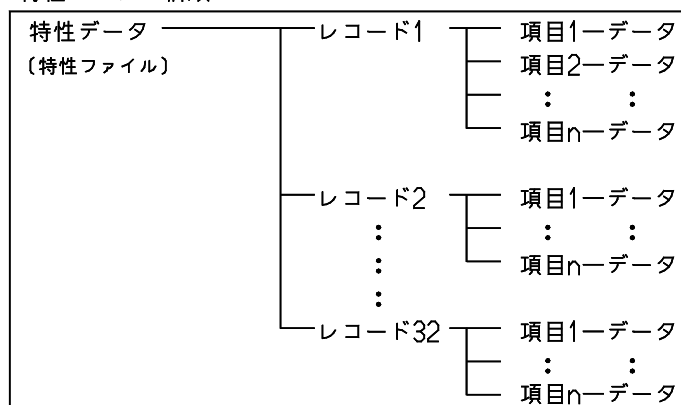


3.1 特性ファイル

特性データの項目として何を含めるかを定義しているのが特性ファイルです。
特性ファイルは複数作成しておくことができますが、1つのモデルでは1つの特性ファイルしか使えません。

1つの特性ファイルには最高 32 のレコードを持つことができます。
各レコードには最高 256 個の項目を設定することができます。各項目は文字列または数値のいずれかです。

特性データの構成



● 標準部品表の構成

特性ファイルの構成例として、標準部品表の構成を下記に示します。

レコード番号	レコード名	項目番号 項目
1	部材データ	1 ラベル頭文字 2 ラベル番号 3 ラベル副番 4 部品番号 5 品名 6 材質 7 数量 8 重量 9 総重量 10 注記

レコード番号	レコード名	項目番号 項目
2	部品 データ	1 番号 2 追加記号 3 PARTS NAME 4 品名 5 備考
3	部品データ	1 N o . 2 N A M E 3 M A T 4 D W G、N O 5 Q t y 6 R e f .
4	新機材配置図	1 番号 2 機器名 3 型式番号 4 メーカー名 5 電圧 (V) 6 電力 7 接続 C P U 8 使用部署 9 注記 10 個数
5	機材配置図	1 番号 2 設置機器名 3 接続 C P U 4 使用部署 5 使用場所 6 電圧／電力

レコード番号	レコード名	項目番号 項目
		7 帳簿番号 8 帳簿区分 9 注記
10	穴形状データ	1 穴タイプ 2 記号 3 直径 4 はめあい 5 加工タイプ 6 数量 7 表示個数 8 深さ 9 深さ2 10 面取り幅 11 注記 12 穴タイプ番号

3.1.1 特性ファイルの作成

特性データを定義する特性ファイルを作成するためには、「テンプレートファイル」と「インプットファイル」という2種類のファイルを用意する必要があります。これらのファイルはテキストエディタで作成します。

テンプレートファイルには1レコード分の項目内容を定義します。特性ファイルに複数のレコードを含めるときは、レコードの数だけテンプレートファイルが必要です。テンプレートファイルのディレクトリと拡張子は、ファイル ACAD.SET のキーワード #SPEC:ASC# の行で指定します。

インプットファイルには、1つの特性データファイルに含めるテンプレートファイルの名前を記述します。最高 32 のテンプレートファイルを指定できます。

インプットファイルのディレクトリと拡張子は、ファイル ACAD.SET のキーワード #SPEC:INP# の行で指定します。

特性データの定義に関係する ファイル ACAD.SET のキーワードはつぎの4つです。キーワード (#SPEC:xxx# の部分) 以外は変更できます。

```
#SPEC:ASC#      "~/spec/"      !.TMP!  Spec data ASCII file
#SPEC:INP#      "~/spec/"      !.INP!  Spec file definition
#SPEC:OUT#      "~/files/"     !.OUT!  Fromto data DB format out
```

```
#SPEC:TXT#      "~/files/"      !.TXT!  Spec data DB format out
```

3.1.2 インプットファイル

インプットファイルの名前は "SPEC" で始まり続いて 4 桁の番号からなります。
この 4 桁の番号は特性ファイル番号で、"0001" から "9999" までの番号を使用できます。この番号は Advance CAD のコマンドで特性ファイルを指定するとき使います。

● 作成規約

インプットファイルは以下で説明するに従ってテキストエディタで作ります。1 行は 255 文字以内でなければなりません。

1 カラム目がスラッシュ '/' の行はコメント行です。

テンプレートファイル名記述行は、下記の書式で記述します。

+ (レコード番号) !テンプレートファイル名!

1 カラム目をプラス '+' で始め、レコード番号 (1 ~ 32) とテンプレートファイル名を記述します。
レコード番号は '(' と ')' で囲みます。

テンプレートファイル名は '!' で囲みます。スペースを入れてはいけません。

例

```

/                                     file:SPEC0001.INP
/                                     Advance CAD  特性データインプットファイル
/
+   ( 1 ) !SPECGEN01.TMP!           Spec Cat. number 1  General buhin
+   ( 2 ) !SPECGEN02.TMP!           Spec Cat. number 2  General parts
+   ( 3 ) !SPECGEN03.TMP!           Spec Cat. number 3  Shiyoukiki
+   ( 4 ) !SPECGEN04.TMP!           Spec Cat. number 4  New machine layout
+   ( 5 ) !SPECGEN05.TMP!           Spec Cat. number 5  Machine layout
+   (10 ) !SPECGEN10.TMP!           Spec Cat. number 10 Hole data

```

3.1.3 テンプレートファイル

● 作成規約

テンプレートファイルは以下で説明するに従ってテキストエディタで作ります。

1 カラム目がスラッシュ '/' の行はコメント行です。

以下に示す 2 つのアスタリスク '**' で始まる行は、節の始まりを示します。
各節の詳細は後述します。

** bias	特性データの世代 (履歴) 管理する世代数を記述する。
** menu/koumoku	項目名、データタイプ、文字数、などを記述する。
** calculation	計算式で記述する。
** label	特性データをジェネラルノート、TAG 名またはリファレンスノートとして表示するときの書式を記述する。

** sort	プリント、プロット時の順番を記述する。
** print	プリントする項目および表示位置、表示方法を記述する。
** print title	プリントする項目名およびその位置を記述する。
** plot	モデル中に部品表を作成するときの項目および表示位置、表示方法を記述する。
** plot title	モデル中に部品表を作成するときの項目名およびその位置を記述する。
** end	テンプレートファイルの定義終了の意味をあらわす。

(1) **** bias**

アイテムに付けた特性データの履歴を残したいときに定義します。
世代管理 (表示・保存) の項目数と世代数を記述します。省略された場合は1世代とします。

```
** bias    ( n, m )
```

n	:	1 世代の項目数を表す。menu/koumoku で設定された項目数とする。
m	:	保存する世代数を表す。最大 5 世代まで保存できる。

(2) ** menu/koumoku

項目の定義を記述します。

T < i, j > "レコード名"

T : レコード名を記述する行をあらわす。
i, j : "レコード名" を表示する位置を指示する。i は行、j は列。
レコード名 : 画面に表示する文字列をダブルクォーツ '""' で囲む。

Kn < i, 1 > "項目名" datatype, cnt = n, form = nm, wd = 1,

countup, total,	tlb = " ファイル名 "
	menu = " メニューページ名 "

Kn	:	特性データの項目を表す。n は項目番号。 K だけで n を付けしないと自動符番するが、項目番号を付ける方がよい。 項目は最大 255 番まで定義できるが、画面に表示できるのは 30 番までである。会 話型で入力／修正できるのは 30 番までなので、31 番以降は表示位置、項目名を 指定してはならない。また、全項目のデータの和が 4096 バイト以下でなければな らない。文字列は文字データ数（漢字は 2 バイト）、整数は $2 \times$ データ数、実数 は $4 \times$ データ数、倍精度実数は $8 \times$ データ数がバイト数。
i, 1	:	画面表示する位置を指示する。i は行、1 は列。 指定がなければ表示しない。画面に表示できるのは 30 行までである。
項目名	:	画面に表示する文字列をダブルクォーツ '""' で囲んで記入する。

次の各項目は全体を '(' と ')' で囲み、項目間をカンマ ';' で区切って並べます。不要な項目は省略できますが、**data type** と **cnt** は必ず記述してください。

data type

character: 文字列。
integer : 16 bit 整数。
real : 32 bit 実数。
double : 64 bit 倍精度実数。

$$\text{cnt} = n$$

データ数。文字列のときは最長で 127 文字以下。
(character, cnt = 4) は最長 4 文字の文字列。
(integer, cnt = 1) は整数データ 1 つ。

```

form = nnm
  整数と実数の表示形式を指定する。
    nn      : 全体のカラム数 (1 ~ 12)
    m       : 小数点以下の桁数 (0 ~ 7, 9)
  データの制限は以下の通り。
    integer: 50
    real    : 84
    double  : 127
  整数データで m = 9 は特別で、先頭部の空白の代りに 0 を置く。
  整数 35 はつぎのように表示する。
    ( integer, cnt = 1, form = 50 ) の場合   □□□35
    ( integer, cnt = 1, form = 59 ) の場合   00035
wd = 1
  特性データバッファでのこのデータの格納位置を指定する。
  記述がなければプログラムが自動的に計算する。
  これはユーザがプログラミングインタフェースで特性データバッファを直接参照しなければ不要。
countup
  アイテムに特性データを付加するたびに自動的にカウントアップする項目に指定する。ラベルの番号
  などに使用する。
total
  部品表や集計表などの作成時に合計を取りたい項目に指定する。
tlb = "ファイル名"
  テキストファイル名を指定する。ファイル名をダブルクォーツ (") で囲む。
  指定したテキストファイルのデータを画面に表示し、その中から選択されたデータを項目に設定す
  る。
  ディレクトリおよびファイル拡張子は ACAD.SET のキーワード #TEXT# で定義する。
menu = "メニューページ名"
  メニューページ名を指定する。メニューページ名をダブルクォーツで囲む。
  メニューページ名についてはマニュアル「システム管理者の手引き」を参照すること。
  tlb と menu は同時に指定できない。

```

以下はテキストファイルの例である。

```

/ file name : SPECTLB01.TXT
鉄
ステンレス
アルミ
ニッケル
特殊鋼

```

以下は ** menu / koumoku の設定例です。

```

** menu / koumoku
T  <1,1>    " 部材データ "
K1  <3,1>    " ラベル頭文字 " ( character, cnt= 2 )
K2  <4,1>    " ラベル番号 "   ( integer, cnt = 1, form = 30, countup )
      :
K9  <11,1>   " 総重量 "       ( double, cnt = 1, form = 82 , total )
K10 <12,1>   " 注記 "         ( character, cnt = 20 , tlb = "DATA001" )

```

部材データ	
(1) ラベル頭文字	
(2) ラベル番号	0
(3) ラベル副番号	
(4) 部品番号	
(5) 品名	
(6) 材質	
(7) 数量	0
(8) 重量	0
(9) 総重量	0
(10) 注記	

(3) ** calculation

項目に設定するデータを計算式からもとめる場合に記述します。
計算式は乗算 (mult)、除算 (devide)、加算 (add)、減算 (sub) のいずれかです。

```
mult ( K1 = [Km,Kn] ) : Km × Kn を K1 にセットする。
devide ( K1 = [Km,Kn] ) : Km ÷ Kn を K1 にセットする。
add ( K1 = [Km,Kn] ) : Km + Kn を K1 にセットする。
sub ( K1 = [Km,Kn] ) : Km - Kn を K1 にセットする。
```

計算式の記述例です。

```
mult ( K9 = [K7,K8] )
K7 が数量、K8 が重量のとき、K9 は数量 × 重量で計算される総重量となります。
```

(4) ** label

特性データをジェネラルノート、TAG 名またはリファランスノートとして表示するときの書式を記述します。

```
GNT [format] : ジェネラルノートで表示する。
TAG [format] : TAG 名で表示する。
RFN [format] : リファランスノートで表示する。
```

[] には表示する項目名を記述します。2つ以上の項目を表示する場合は項目名をカンマ',' またはスラッシュ '/' で区切って並べます。スラッシュ '/' は改行をあらわします。
表示する項目に文字列を含める場合は文字列をシングルクォーツ'で囲みます。

たとえば、TAG 名で表示するとき、項目1のあとにハイフン '-' を付け、つづいて項目2,3を付けて1つの文字列としたい場合は、下記のようにします。

```
TAG [K1, '-', K2, K3]
```

以下は、** label の例です。

```
GNT [K1, '-', K2, K3/K4/K5/K6/K7, ' 個 ', K8, ' Kg Total ', K9, ' Kg' /K10]
TAG [K1, '-', K2, K3]
RFN [K1, K2, K3]
```

(5) ** sort

部品票 (集計表) 作成時のソート (並べかえ) の順番を指定します。項目名をカンマ',' で区切って並べます。昇順に並べかえます。

以下は、** sort の例です。

```
sort [K1, K2, K3, K4, K5, K6]
項目 K1, K2, K3, K4, K5, K6 の順にソートされます。
```

(6) ** print

集計表をファイルに出力するときの出力する項目、その位置、形式を記述します。

```
data [ format ] ( line = l, col = m )
```

data [format] : [] に出力する項目名を記述する。2 つ以上の項目名を繋げて出力する場合は文字列をカンマ ',' で区切る。項目間に文字列を含める場合はシングルクォート ' で囲む。

line = n : 配置する行数 (1 ~) を指定する。

col = m : データの配置位置をカラムで指定する。1 行は 256 カラム以内。

たとえば、data [K1,"×",K2," ",K10," 個"] (line = 1,col = 11) と設定してある場合に K1=5, K2=3, K10=15 が入ったとします。この場合は、1 行目の 11 カラム目から、「5×3 15 個」という文字列が入ります。

以下は、** print の例です。

```
data [K1, '-', K2, K3] ( line = 1, col = 1 )
data [K4] ( line = 1, col = 6 )
:
data [K10] ( line = 1, col = 70 )
```

(7) ** print title

集計表をファイルに出力するときのタイトルを記述します。これは **print の節で記述した項目に合わせなければなりません。

```
text " 文字列 " ( line = l, col = m )
```

text " 文字列 " : 文字列をダブルクォーツ '""' で囲んで記述する。テキストにカリキュレータ変数、システムレジスタを参照することができる。

line = n : 配置する行数 (1 ~) を指定する。

col = m : データの配置位置をカラムで指定する。1 行は 132 カラム以内。

以下は、** print title の例です。

```
/
/ Spec data category number 1
/
:
** print title
text "[#CURMDLNAME]" ( line=1, col= 2 )
text "[A]" ( line=1, col= 7 )
text " 品名 " ( line=1, col=19 )
text " 材質 " ( line=1, col=35 )
text " 数量 " ( line=1, col=47 )
text " 重量 " ( line=1, col=53 )
text " 総重量 " ( line=1, col=61 )
:
** end
```

(8) ** plot

部品表 (集計表) を作画するときの項目、位置、形式を記述します。

```
next ( tate < x, y > )
```

または

```
next ( yoko < x, y > )
```

next : 表の表示方法および次行のデータの表示位置をあらわす。
 tate : 表を縦表示にすると指定する。
 yoko : 表を横表示にすると指定する。

x, y : 表の原点からの相対位置

tate 表示の場合は y 値が次のデータの表示位置として +y 移動する。

yoko 表示の場合は x 値が次のデータの表示位置として +x 移動する。

なお x,y 値はすべて実際にプロッタで出力する実寸値とする。

すなわち、<270.0, 5.0> の場合モデルの縮尺値に関係なくプロッタ上の位置として定義すること。

```
data [ format ] ( < x, y >, h=th, adj=xx, area=< bw, bh >,
```

```
cls=yyy, rev=rrr, lwt=zz)
```

data [format] : [] に出力する項目名を記述する。2 つ以上の項目名を繋げて出力する場合は文字列をカンマ ' , ' で区切る。項目間に文字列を含める場合はシングルクォート ' ' で囲む。

x, y : データの表示原点。(表の原点からの相対位置)

h = th : 文字の大きさ (省略時は 3)

adj = xx : データのテキスト原点をつぎのいずれかで指定する。
 ll : 左下 (省略時)
 lc : 左中央
 lu : 左上
 cl : 中央下
 cc : 真中
 cu : 中央上
 rl : 右下
 rc : 右中央
 ru : 右上

area = <bw, bh> : テキストを表示するゾーンを指定する。
 テキストがゾーンをはみ出した場合、ゾーンに収まるように調整する。省略可。
 テキスト列がゾーンより大きい場合は計算式
 文字数 × 文字幅 ≤ ゾーン
 により文字の大きさを決定する。

cls = yyy : クラス番号を指定する。(省略時は現在のクラス番号)

rev = rrr : レビジョン番号を指定する。(省略時は現在のレビジョン番号)

lwt = zz : 線幅番号を指定する。(省略時は現在の線幅番号)

```
line ( < xs, ys >, < xe, ye >, linetype, cls=yyy, rev=rrr, lft=tt, lwt=zz)
```

枠線を引きます。

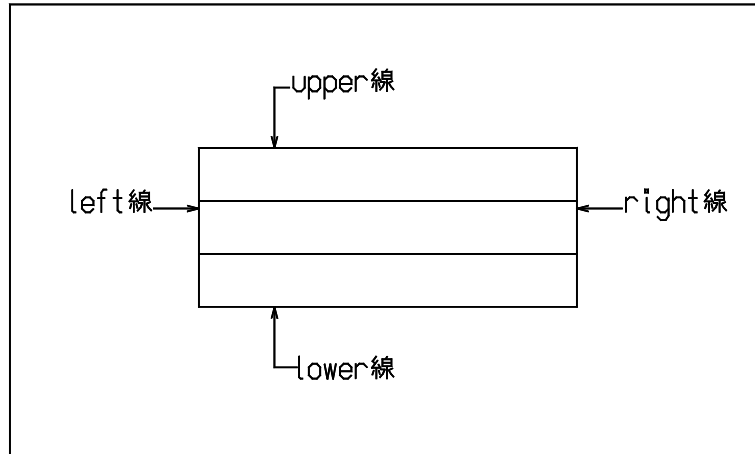
<xs, ys>, <xe, ye> : 線分の始点と終点の座標

linetype : 表の外枠線のときだけ指定する。つぎのいずれかを記入する。
 left line
 right line

```

                                lower line
                                upper line
cls = yyy                      : クラス番号を指定する。(省略時は現在のクラス番号)
rev = rrr                      : レビジョン番号を指定する。(省略時は現在のレビジョン番号)
lft = tt                       : 線種番号を指定する。(省略時は現在の線種番号)
lwt = zz                       : 線幅番号を指定する。(省略時は現在の線幅番号)

```



以下は ** plot の例です。

```

next ( tate, <275.0, 5.0> )
data [K1, '-', K2, K3] ( <0.0, 0.0>, h = 3.0, area = <20.0, 5.0> )
data [K4]                ( <20.0, 0.0>, h = 3.0, area = <40.0, 5.0> )
:
data [K10]                ( <215.0, 0.0>, h = 3.0, area = <60.0, 5.0> )
line ( <0.0, 0.0>, <0.0, 5.0>, left line )
:
line ( <0.0, 0.0>, <275.0, 0.0>, lower line )
line ( <0.0, 5.0>, <275.0, 5.0>, upper line )

```

(9) ** plot title

集計表を作画するときの表のタイトル、およびタイトルを囲む枠を記述します。

```
text " 文字列 " ( < x, y >, h=th, area=< bw, bh >, cls=yyy, rev=rrr, lwt=zz)
```

```

text " 文字列 " : 文字列をダブルクォーツ '""' で区切って記述する。
                  テキストにカリキュレータ変数、システムレジスタを参照することができ
                  る。
x, y             : データの表示原点。(表の原点からの相対位置)
h = th          : 文字の大きさ (省略時は 3)
area = <bw, bh> : テキストを表示するゾーンを指定する。
                  テキストがゾーンをはみ出した場合、ゾーンに収まるように調整する。省略
                  可。
                  テキスト列がゾーンより大きい場合は計算式
                  文字数 × 文字高さ × ゾーン
                  により文字の大きさを決める。
cls = yyy       : クラス番号を指定する。(省略時は現在のクラス番号)
rev = rrr       : レビジョン番号を指定する。(省略時は現在のレビジョン番号)
lwt = zz        : 線幅を指定する。(省略時は現在の線幅)

```

以下は text の設定例です。

```
** plot title
```

```

data [K1, '-', K2, K3] ( <0.0,0.0>, h = 3.0, area = <20.0,5.0> )
text "[#CURMDLNAME]" ( <20.0,0.0>, h = 3.0, area = <40.0,5.0> )
text "[A]" ( <60.0,0.0>, h = 3.0, area = <50.0,5.0> )
:
data [K10] ( <215.0,0.0>, h = 3.0, area = <60.0,5.0> )
:
** end

```

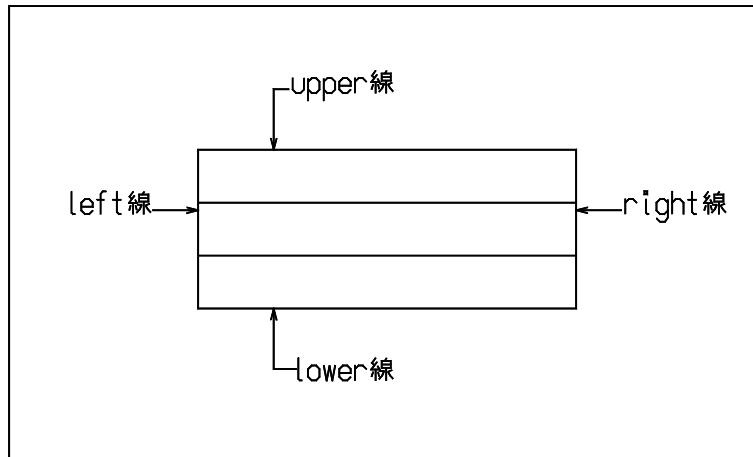
```
line ( < xs, ys >, < xe, ye >, linetype, cls=yyy, rev=rrr, lft=tt, lwt=zz)
```

枠線を引きます。

```

<xs, ys>, <xe, ye> : 線分の始点と終点の座標
linetype           : 表の外枠線のときだけ指定する。つぎのいずれかを記入する。
                    left line
                    right line
                    lower line
                    upper line
cls = yyy          : クラス番号を指定する。(省略時は現在のクラス番号)
rev = rrr          : レビジョン番号を指定する。(省略時は現在のレビジョン番号)
lft = tt           : 線種番号を指定する。(省略時は現在の線種番号)
lwt = zz           : 線幅を指定する。(省略時は現在の線幅)

```



以下は text および line の設定例です。

```

text " 項番 " ( <3.0,0.0>, h = 3.5, area = <10.0,5.0> )
text " 部品番号 " ( <22.0,0.0>, h = 3.5, area = <25.0,5.0> )
:
text " 重量 " ( <167.0,0.0>, h = 3.5, area = <10.0,5.0> )
text " 総重量 " ( <192.0,0.0>, h = 3.5, area = <10.0,5.0> )
text " 注記 " ( <217.0,0.0>, h = 3.5, area = <40.0,5.0> )
line ( <0.0,0.0>, <0.0,5.0>, left line )
line ( <20.0,0.0>, <20.0,5.0> )
:
line ( <275.0,0.0>, <275.0,5.0>, right line )
line ( <0.0,0.0>, <275.0,0.0>, lower line )
line ( <0.0,5.0>, <2715.0,5.0>, upper line )

```

(10) ** end
定義終了を示します。

3.1.4 特性ファイルチェッカー

プログラム `speccheck` を使用すると特性ファイルの内容を確認できます。新しい特性ファイルを作成した場合は、必ず `speccheck` を使用して内容を確認してください。

`speccheck` はコマンドラインより使用します。

`speccheck` の構文は下記のようになります。

```
prompt% speccheck [ -nSPECNUMBER ] [ -pACAD.SET ] <C/R>
```

`-nSPECNUMBER` : `-n` に続けて特性ファイルの番号を指定する。省略すると `<C/R>` を押した後に下記のように番号入力を促す。
Enter file number SPECn. INP (n=1-9999):
`-pACAD.SET` : `-p` に続けて、使用する `ACAD.SET` のディレクトリと名前を指定する。省略すると現在のディレクトリの `ACAD.SET` が使用される。

例えば、`SPEC0001.INP` を使用して定義された特性ファイルの内容を確認する場合は、次のように入力します。

```
prompt% speccheck <C/R>
Enter file number SPECn. INP (n=1-9999): 1 <C/R>
```

あるいは

```
prompt% speccheck -n1 <C/R>
```

0001 は 1 と入力しても同じです。`SPEC` は固定なので、この場合は 1 とだけ入力します。また、`speccheck` の実行結果をファイルに出力するには下記のようにします。

この場合、特性ファイル番号は 101 で、結果はファイル `log` へ出力されます。

```
prompt% speccheck -n101 > log <C/R>
```

テンプレートファイルの例)

```

/
/
/          Spec new data category number 1
/
** bias ( 10 , 5 )
/
** menu / koumoku
T  < 1,1>      " 部材データ "
K1  < 3,1>      " ラベル頭文字 " ( character , cnt = 2 )
K2  < 4,1>      " ラベル番号 "   ( integer   , cnt = 1 , form=30 , countup )
K3  < 5,1>      " ラベル副番 "   ( character , cnt = 4 )
K4  < 6,1>      " 部品番号 "     ( character , cnt =12 )
K5  < 7,1>      " 品名 "         ( character , cnt =16 )
K6  < 8,1>      " 材質 "         ( character , cnt =12 , tlb = "NEJI001" )
K7  < 9,1>      " 数量 "         ( integer   , cnt = 1 , form=40 , total )
K8  <10,1>      " 重量 "         ( double    , cnt = 1 , form=83 )
K9  <11,1>      " 総重量 "       ( double    , cnt = 1 , form=82 , total )
K10 <12,1>      " 注記 "         ( character , cnt =20 )
/
** calculation
mult ( K9 = [K7,K8] )
/
/ Output
/
** label
tag [ K1, '-', K2, K3 ]

```

```

rfn [ K1,K2,K3 ]
gnt [ K1,'-',K2,K3 / K4 / K5 / K6 / K7,' 個 ',K8,' Kg Total',K9,'Kg' / K10 ]
** sort
sort [ K1,K2,K3,K4,K5,K6 ]
** print
data [K1,'-',K2,K3]      ( line=1 , col= 1 )
data [K4]                ( line=1 , col= 6 )
data [K5]                ( line=1 , col=18 )
data [K6]                ( line=1 , col=34 )
data [K7]                ( line=1 , col=46 )
data [K8]                ( line=1 , col=52 )
data [K9]                ( line=1 , col=60 )
data [K10]               ( line=1 , col=70 )
** print title
text  " 項番 "           ( line=1 , col= 2 )
text  " 部品番号 "       ( line=1 , col= 7 )
text  " 品名 "           ( line=1 , col=19 )
text  " 材質 "           ( line=1 , col=35 )
text  " 数量 "           ( line=1 , col=47 )
text  " 重量 "           ( line=1 , col=53 )
text  " 総重量 "        ( line=1 , col=61 )
text  " 注記 "           ( line=1 , col=71 )
** plot
next  ( tate , < 275.0 , 5.0 > )
data [K1,'-',K2,K3]      ( < 1.0 , 0.0 > , h=3.0 , area = < 20.0 , 5.0 > )
data [K4]                ( < 21.0 , 0.0 > , h=3.0 , area = < 40.0 , 5.0 > )
data [K5]                ( < 61.0 , 0.0 > , h=3.0 , area = < 50.0 , 5.0 > )
data [K6]                ( < 111.0 , 0.0 > , h=3.0 , area = < 40.0 , 5.0 > )
data [K7]                ( < 151.0 , 0.0 > , h=3.0 , area = < 15.0 , 5.0 > )
data [K8]                ( < 166.0 , 0.0 > , h=3.0 , area = < 25.0 , 5.0 > )
data [K9]                ( < 191.0 , 0.0 > , h=3.0 , area = < 25.0 , 5.0 > )
data [K10]               ( < 216.0 , 0.0 > , h=3.0 , area = < 60.0 , 5.0 > )
line  ( < 0.0 , 0.0 > , < 0.0 , 5.0 > , left line )
line  ( < 20.0 , 0.0 > , < 20.0 , 5.0 > )
line  ( < 60.0 , 0.0 > , < 60.0 , 5.0 > )
line  ( < 110.0 , 0.0 > , < 110.0 , 5.0 > )
line  ( < 150.0 , 0.0 > , < 150.0 , 5.0 > )
line  ( < 165.0 , 0.0 > , < 165.0 , 5.0 > )
line  ( < 190.0 , 0.0 > , < 190.0 , 5.0 > )
line  ( < 215.0 , 0.0 > , < 215.0 , 5.0 > )
line  ( < 275.0 , 0.0 > , < 275.0 , 5.0 > , right line )
line  ( < 0.0 , 0.0 > , < 275.0 , 0.0 > , lower line )
line  ( < 0.0 , 5.0 > , < 275.0 , 5.0 > , upper line )
** plot title
next  ( tate , < 275.0 , 5.0 > )
text  " 項番 "           ( < 3.0 , 0.0 > , h=3.5 , area = < 10.0 , 5.0 > )
text  " 部品番号 "       ( < 22.0 , 0.0 > , h=3.5 , area = < 25.0 , 5.0 > )
text  " 品名 "           ( < 62.0 , 0.0 > , h=3.5 , area = < 30.0 , 5.0 > )
text  " 材質 "           ( < 112.0 , 0.0 > , h=3.5 , area = < 25.0 , 5.0 > )
text  " 数量 "           ( < 152.0 , 0.0 > , h=3.5 , area = < 10.0 , 5.0 > )
text  " 重量 "           ( < 167.0 , 0.0 > , h=3.5 , area = < 10.0 , 5.0 > )
text  " 総重量 "        ( < 192.0 , 0.0 > , h=3.5 , area = < 10.0 , 5.0 > )
text  " 注記 "           ( < 217.0 , 0.0 > , h=3.5 , area = < 40.0 , 5.0 > )
line  ( < 0.0 , 0.0 > , < 0.0 , 5.0 > , left line )
line  ( < 20.0 , 0.0 > , < 20.0 , 5.0 > )
line  ( < 60.0 , 0.0 > , < 60.0 , 5.0 > )
line  ( < 110.0 , 0.0 > , < 110.0 , 5.0 > )
line  ( < 150.0 , 0.0 > , < 150.0 , 5.0 > )
line  ( < 165.0 , 0.0 > , < 165.0 , 5.0 > )
line  ( < 190.0 , 0.0 > , < 190.0 , 5.0 > )
line  ( < 215.0 , 0.0 > , < 215.0 , 5.0 > )
line  ( < 215.0 , 0.0 > , < 215.0 , 5.0 > )

```

```
line      ( < 275.0 , 0.0 > , < 275.0 , 5.0 > , right line )
line      ( < 0.0 , 0.0 > , < 275.0 , 0.0 > , lower line )
line      ( < 0.0 , 5.0 > , < 275.0 , 5.0 > , upper line )
** end
```

3.2 コマンド

特性データを扱うコマンドには以下のものがあります。

- (1) 最初に特性ファイルとレコード番号を指示します。特性ファイルは特性データとしてアイテムに付ける項目を宣言したものです。

<code>SPEC __ FLN</code>	特性ファイルを指定する
<code>SPEC __ NUM</code>	レコード番号を指定する

- (2) 特性データテーブルにデータを設定します。ここで設定した特性データテーブルの内容をアイテムに付加することになります。

<code>SPEC __ INP</code>	特性データテーブルを設定する
<code>SPEC __ GET</code>	アイテムの特性データを特性データテーブルにコピーする
<code>SPEC __ CLR</code>	特性データテーブルを削除する
<code>SPEC __ DSP</code>	特性データテーブルを表示する

- (3) 特性データテーブルの特性データをアイテムに付加します。

<code>SPEC __ ADD</code>	アイテムに特性データを付ける
<code>SPEC __ DEL</code>	アイテムの特性データを削除する
<code>SPEC __ MOD</code>	アイテムの特性データを変更する
<code>SPEC __ UPD</code>	アイテムの特性データを更新する
<code>SPEC __ COPY</code>	特性データをコピーする
<code>SPEC __ VER</code>	アイテムの特性データを表示する
<code>SPC __ CATVER</code>	モデルの特性レコード番号とレコード使用数を表示
<code>SPEC __ BIAS</code>	特性データの世代を1世代更新する
<code>SPEC __ SEL</code>	特性データを元にアイテム検索をする
<code>SPEC __ NODATA</code>	アイテムの特性レコード番号を検索する

- (4) 特性データの表示
特性データをジェネラルノート、TAG 名またはリファランスノートのいずれかの形式で図面上に表示します。

<code>SPEC __ GNT</code>	アイテムの特性データをジェネラルノートで表示する
<code>SPEC __ TAG</code>	アイテムの特性データとタグ名を表示する
<code>SPEC __ RFN</code>	アイテムの特性データをリファランスノートで表示する
<code>SPEC __ ERS</code>	特性データを非表示にする

- (5) 特性データの集計表
アイテムの特性データを集めて表を作成します。

`SPEC__WRT` アイテムの特性データをファイル出力する
`SPEC__OUT` 部品表を作成する

- (6) 特性データの確認
特性データの重複を確認する。

`SPEC__CHK` 同一特性データを持ったシンボルを確認する

3.2.1 特性データの設定

3.2.1.1 特性ファイルを指定する

【メニュー】

〔特性データ〕 → 〔ファイル番号〕

【構文】

`SPEC_FLN s <CE>`

`s` : 特性ファイルの番号を入力する。
たとえば1と入力すれば、特性ファイル名は `SPEC0001`、2ならば `SPEC0002` になる。ひとつのモデル中では1つの特性ファイルだけを使用する。

3.2.1.2 特性ファイルのレコード番号を指定する

【メニュー】

〔特性データ〕 → 〔レコード番号〕

【構文】

`SPEC_NUM s <CE>`

`s` : 特性ファイルを指定するとそのファイル中に定義されている特性データレコードが表示されるので、その番号を入力するか画面上をデジタイズしてレコード番号を指定する。

3.2.1.3 特性データテーブルを設定する

【メニュー】

〔特性データ〕 → 〔特性設定〕

【構文】

SPEC_INP [FNM filename] $\left[\begin{array}{c} \left\{ \begin{array}{c} s \\ d \end{array} \right\} \text{ data} \end{array} \right]^+ <CE>$
--

設定値をテキストファイルから読み込み一括設定する場合

FNM filename : ファイル名を入力する。
 設定されていたデータはクリアされる。
 ファイルのディレクトリおよびファイル拡張子は ACAD.SET のキーワード #SPEC:TXT# で定義する。

各項目ごとに設定する場合、項目番号をつぎのいずれかで指示する。

s : 項目番号を入力する。
 d : 項目をデジタイズする。

データを入力する。

data : データを入力する。
 特殊文字列として "NIL" が入力されるとデータがクリアされる。

● テキストファイルの書式

コマンド SPEC_INP および SPEC_MOD で指定するテキストファイルは、以下の規約に従ってテキストエディタで作成してください。

またはコマンド『Item 特性出力』(SPEC_WRT) で出力されたファイルでも拡張子を変更すればテキストファイルとして使用することもできます。

- コメント行
 1 カラム目がスラッシュ '/' の行はコメント行です。
- テキストファイルのデータ記述行

cat = 1	:	レコード番号 (1 ~ 32) を記述します。
2 = 1	:	項目番号とデータを記述します。この例は項目 2 のデータが 1 になる。 データが文字列の場合はシングルクォーツ ' ' で囲む。 データの文字数はテンプレートファイルの menu/koumoku で定義された数までとする。
end	:	テキストファイルの終りを指示する。

例

```

/ file : NEJ101.TXT
/
cat = 1
2 = 1
4 = 'CTC0002'
5 = ' シリンダチューブ '
6 = 'STKM13A'
7 = 3
8 = 22
10 = ' 特注品 '
end
  
```

3.2.1.4 アイテムに付いている特性データを特性データテーブルにコピーする

【メニュー】

〔特性データ〕 → 〔特性設定〕 → 〔図形より〕

【構文】

SPEC_GET [IS]+

IS : 特性データを持つアイテムを選択する。

3.2.1.5 特性データテーブルを削除する

【メニュー】

〔特性データ〕 → 〔特性設定〕 → 〔クリア〕

【構文】

SPEC_CLR

● 特性データテーブルを表示する

【メニュー】

〔特性データ〕 → 〔特性設定〕 → 〔テーブル表示〕

【構文】

SPEC_DSP

3.2.2 特性データテーブルを表示する

3.2.2.1 アイテムに特性データを付ける

【メニュー】

〔特性データ〕 → 〔特性付加〕

【構文】

SPEC_ADD [MULT] [IS]+ <GE>

同一レコードの特性データが既に付いているときは更新されない。
特性データテーブルに値を設定しないまま特性データを付けると、空の特性データが付くことがある。

異なる特性データを付加したいとき指示する。

MULT : 特性データが既に付いているアイテムに、異なる特性データを付加したいとき指示する。

特性データを付けるアイテムを指示する。

IS : 特性データを付けるアイテムを選択する。特性データテーブルの値がアイテムに付加される。

3.2.2.2 アイテムの特性データを削除する

【メニュー】

〔特性データ〕 → 〔削除〕

【構文】

SPEC_DEL	[DSP]	$\left\{ \begin{array}{l} [IS]^+ \\ USEACT \end{array} \right\}$	<CE>
----------	---------	--	------

現在設定されているレコード番号の特性データを削除する。

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

削除するアイテムをつぎのいずれかで指示する。

IS : 特性データを削除するアイテムを選択する。

USEACT : アクティブリスト中のアイテムから特性データを削除する。

3.2.2.3 アイテムの特性データを変更する

【メニュー】

〔特性データ〕 → 〔修正〕

【構文】

SPEC_MOD	[DSP]	IS	[FNM filename]	$\left[\left\{ \begin{array}{l} s \\ d \end{array} \right\} \text{data} \right]^+$	<CE>
----------	---------	----	------------------	---	------

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

変更するアイテムを指示する。

IS : アイテムを選択すると、画面にそのアイテムの特性データが表示される。

変更値をテキストファイルから読み込み一括変更する場合。

FNM filename : ファイル名を入力する。
設定されていたデータはクリアされる。
ファイルのディレクトリおよびファイル拡張子は ACAD.SET のキーワード #SPEC:TXT# で定義する。

各項目ごとに変更する場合、項目番号をつぎのいずれかで指示する。

s : 項目番号を入力する。
世代管理をしている場合は $s = \text{項目数} * (\text{世代数} - 1) + \text{項目番号}$ とする。
たとえば、3世代の項目 5番を変更する場合は
 $10 * (3 - 1) + 5 = 25$ になる。
d : 項目をデジタイズする。

新しいデータを入力する。

data : 新しい特性データを入力する。
特殊文字列として "NIL" が入力されるとデータがクリアされる。

テキストファイルの書式はコマンド SPEC_INP の項目を参照のこと。

3.2.2.4 アイテムの特性データを更新する

【メニュー】

〔特性データ〕 → 〔更新〕

【構文】

```
SPEC_UPD [ DSP ] [ IS ]+ <CE>
```

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

特性データを更新するアイテムを指示する。

IS : 特性データを更新するアイテムを選択する。特性データテーブルの値をアイテムに付け直す。

3.2.2.5 特性データをコピーする

【メニュー】

〔特性データ〕 → 〔コピー〕

【構文】

```
SPEC_COPY IS1 IS2 <CE>
```

既存のアイテムについている特性データを別のアイテムにコピーする。

コピー対象アイテムを指示する。

IS1 : 特性データを持つアイテムを選択する。

IS2 : 特性データを付けるアイテムを選択する。指定したアイテムに特性データが付いているときは、この特性データは消去されて新しい特性データが付く。

3.2.3 アイテムに付けた特性データを確認する

3.2.3.1 アイテムの特性データを画面に表示する

【メニュー】

〔特性データ〕 → 〔VER〕

【構文】

```
SPEC_VER [ DSP ] IS <CE>
```

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

アイテムを指示する。

IS : 特性データを持つアイテムを選択する。

3.2.3.2 特性レコード番号使用一覧表

【メニュー】

〔特性データ〕 → 〔データ数表示〕

【構文】

SPC_CATVER

モデルで使用している特性レコード番号とそのレコードの使用数を画面に表示する。

3.2.3.3 特性データの世代を1世代更新する

【メニュー】

〔特性データ〕 → 〔世代更新〕

【構文】

SPEC_BIAS IS <CE>

世代を更新するアイテムを選択する。

IS : 世代を更新するアイテムを選択する。
 世代の更新ができる特性データはテンプレートファイルに bias の定義があるレコード番号の特性データなる。
 世代の更新は1世代目のデータを2世代目にコピーする。2世代以降にもデータがある場合は1世代ずつ更新され、5世代目のデータは消去される。
 これは世代の登録数が最大5世代となっているため。

3.2.3.4 特性データをもとにアイテム検索をする

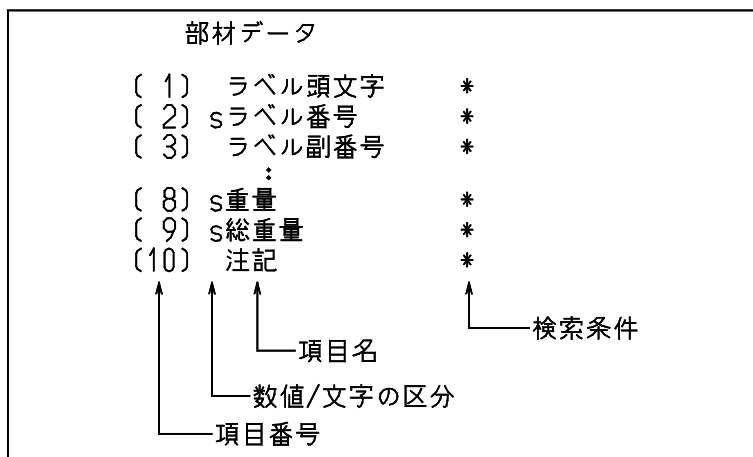
【メニュー】

〔特性データ〕 → 〔次ページ〕 → 〔特性選択〕

【構文】

SPEC_SEL [FNM filename] $\left[\left[\begin{matrix} s \\ d \end{matrix} \right] \text{data} \right]^+ \left\{ \begin{matrix} \text{ALLPIC} \\ \text{USEACT} \end{matrix} \right\} [\text{INF filename}] \text{ <CE>}$

コマンド『特性選択』(SPEC_SEL) が選択されると画面に現在設定されている特性レコードの項目が表示される。



数値／文字の区分に S が表示されている項目は、数値データであることを表している。

S がなければ文字データである。

検索条件に * がついている項目は、検索されない。数値または文字列が表示されている項目だけを検索条件とする。

検索内容は次の方法で指定する。

数値データの項目の場合は次の条件式を指定する。

S=1 この項目の値が 1 であるならばの意味

S>=1&&s<10 この項目の値が 1 以上 10 未満ならばの意味

ここでの S はその項目の値をあらわす変数である。

文字データの項目の場合にはつぎの 3 つの条件のいずれかを使用する。

- 文字列のパターンを指定する。
パターンには * (ワイルドカード) が使用できる。* は 0 文字以上の任意の文字列に相当する。
たとえば、A* は A で始まる文字列と一致する。また *AB* ならば文字列中に AB を含む文字列と一致する。
- 文字列を範囲で指定する。
?>"DD"&&?<"FF" というような条件式で指定する。? はその項目の文字列をあらわす変数で、この式は "DD" より後で "FF" より前の文字列ならばという意味になる。結果としては "DE" や "FA" などが当てはまる。"AA" や "FG" は当てはまらない。なお文字列はダブルクォーツ "" で囲む。
- 文字列を設定していない空レコードを捜すときには空であることを示す特別な文字 'NIL' を指定する。

検索条件をテキストファイルから読み込み一括設定する。

FNM filename : ファイル名を入力する。
 設定されていたデータはクリアされる。
 ファイルのディレクトリおよびファイル拡張子は ACAD.SET のキーワード #SPEC:TXT# で定義する。

各項目ごとに設定する場合は項目番号を次のいずれかで指示する。

テキストファイルにより設定された項目にあらたに条件を追加することもできる。

s : 追加する項目番号を入力する。
d : 追加する項目をデジタイズで指示する。

データを入力する。

data : データを入力する。

検索するアイテムをつぎのいずれかで指示する。

ALLPIC : モデル中の全てのアイテムを検索の対象にする。(省略可)
USEACT : アクティブリスト中のアイテムを検索の対象にする。

設定された検索条件のデータをテキストファイルに出力する。

INF filename : ファイル名を入力する。
ファイルのディレクトリおよびファイル拡張子は ACAD.SET のキーワード
#SPEC:OUT# で定義する。

検索を実行する。

<CE> : 検索項目の設定が終了し <CE> を入力すると、検索条件を満たす特性記録を持つアイテムを捜し出してアクティブリストにのせる。

3.2.3.5 アイテムの特性記録番号を検索する

【メニュー】

〔特性データ〕 → 次ページ → 〔無特性検索〕

【構文】

SPEC_NODATA <CE>

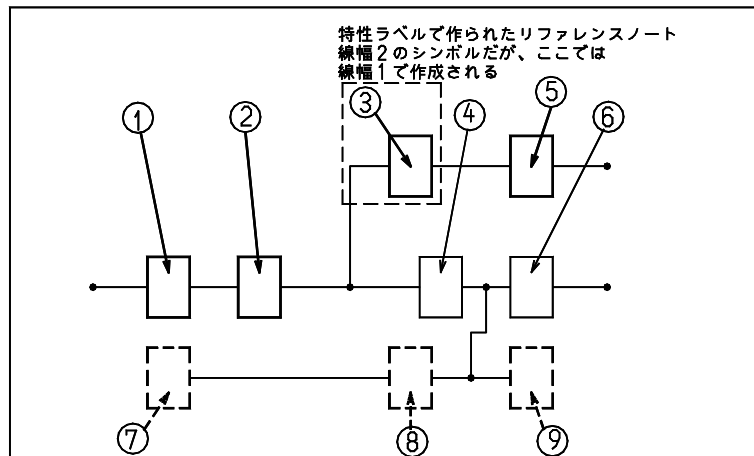
アクティブリスト上のアイテムを調べて、現在の記録番号の特性データが付いていないアイテムだけをアクティブリストに残す。

3.2.4 特性データの表示

アイテムに付いている特性データをジェネラルノート、TAG 名またはリファレンスノートとして表示します。ジェネラルノート、TAG 名、リファレンスノートのいずれか1つだけです。たとえばジェネラルノートを表示させているとき、TAG 名を表示させるとジェネラルノートは削除されます。

ジェネラルノート、TAG 名、リファレンスノートはアイテムの一部分です。特性データを修正すればジェネラルノート、TAG 名、リファレンスノートも更新されます。文字列の修正コマンド(TEXT/REP)で文字列だけを変更してしまうと特性データとの整合性がとれなくなりますので注意してください。

ジェネラルノート、TAG 名、リファレンスノートの文字列および引出線は、いつでも線種 1・線幅 1 で表示します。



3.2.4.1 アイテムの特性データをジェネラルノートで表示する

【メニュー】

〔特性データ〕 → 〔GNT〕

【構文】

```
SPEC_GNT [ DSP ] [ PAR ] [ MULT ] [ IS [ P ]+ <CE> ]+
```

表示する文字列の内容はテンプレートファイルで指定する。

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

文字列をラインセグメントに平行に表示する。

PAR : ラインセグメントの持つ角度に平行な位置に文字列を表示する。

ジェネラルノートを複数表示する。

MULT : ジェネラルノートを複数表示したいときに指定する。最大 32ヶ所まで。

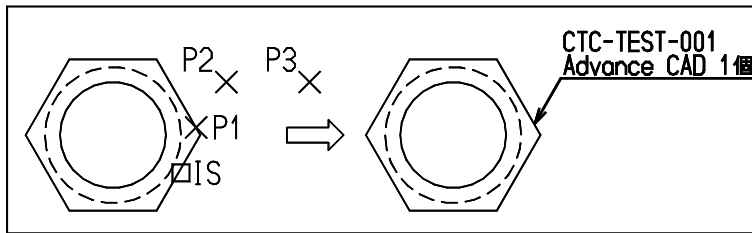
アイテムを指示する。

IS : 特性データを持つアイテムを選択する。

ジェネラルノートの位置を指示する。

P : ジェネラルノートの位置をテンポラリポイントで入力する。
点を入力しなければアイテム選択した位置に文字列を書く。アイテムがシンボルであればシンボル原点に文字列を書く。1 点だけのときはその位置に文字列を書く。2 点以上入力すれば引出線付きの文字列を書く。

SPEC_GNT IS P1 P2 P3 <CE>



3.2.4.2 アイテムの特性データを TAG 名で表示する

【メニュー】

〔特性データ〕 → 〔TAG〕

【構文】

SPEC_TAG [DSP] [MULT] [IS [P]+ <CE>]+

表示する TAG 名の内容はテンプレートファイルで指定する。

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

表示方法を指示する。

MULT : TAG 名を複数表示したいときに指定する。最大 32ヶ所まで。

アイテムを指示する。

IS : 特性データを持つアイテムを選択する。

TAG 名の表示位置を指示する。

P : TAG 名の表示位置をテンポラリポイントで入力する。
 1 点だけのときはその位置に文字列を書く。2 点以上入力すれば引出線付きの文字列を書く。点を入力しなければアイテム選択した位置に文字列を書く。シンボルであればシンボル原点に文字列を書く。

3.2.4.3 アイテムの特性データをリファランスノートで表示する

【メニュー】

〔特性データ〕 → 〔RFN〕

【構文】

SPEC_RFN [DSP] [MULT] [IS [P]+ <CE>]+

表示するリファランスノートの内容はテンプレートファイルで指定する。

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

表示方法を指示する。

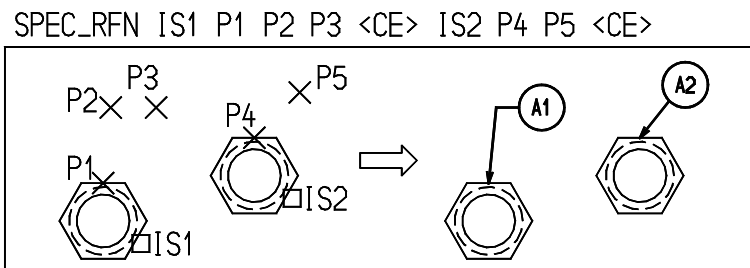
MULT : リファレンスノートを複数表示したいときに指定する。最大 32ヶ所まで。

アイテムを指示する。

IS : 特性データを持つアイテムを選択する。

リファレンスノートの表示位置を指示する。

P : リファレンスノートの表示位置をテンポラリポイントで入力する。
1点だけのときはその位置に文字列を書く。2点以上入力すれば引出線付きの文字列を書く。点を入力しなければアイテム選択した位置に文字列を書く。シンボルであればシンボル原点に文字列を書く。



3.2.4.4 特性データの表示をやめる (非表示にする)

【メニュー】

〔特性データ〕 → 〔テキスト消去〕

【構文】

SPEC_ERS [DSP]	$\left[\begin{array}{l} [IS]^+ \\ USEACT \end{array} \right]^+ <CE>$
------------------	---

特性データをジェネラルノート、TAG 名あるいはリファレンスノートで表示したものを除去する。
特性データは削除されない。

アイテムに付加されている複数の特性データを一覧で表示するとき指示する。

DSP : 複数の特性データを一覧で表示したいとき指示する。

特性データの表示をやめるアイテムをつぎのいずれかの方法で指示する。

IS : GNT, TAG あるいは RFN の表示をやめるアイテムを選択する。

USEACT : アクティブリスト中のアイテムを対象にする。

3.2.5 部品表 (リスト) の作成

3.2.5.1 アイテムの特性データをファイルに出力する

【メニュー】

〔特性データ〕 → 〔Item 特性出力〕

【構文】

SPEC_WRT filename	$\left\{ \begin{array}{l} [IS]+ \langle GE \rangle \\ USEACT \end{array} \right\}$
-------------------	--

出力ファイル名を指示する

filename : ファイル名を入力する。
 ファイルのディレクトリおよびファイル拡張子は ACAD.SET のキーワード
 #SPEC:OUT# で定義する。

対象とするアイテムをつぎのいずれかの方法で指示する。

IS : 特性データを持つアイテムを選択する。
 USEACT : アクティブリスト中の特性データを持つアイテムを対象とする。

例

```

/
** item = 'S000103'      : アイテム名
  cat = 1                : レコード番号
  2 = 3.000000000000     : 項目 #2
  4 = 'CTC003'           : " #4
  5 = 'ピストンロッド'   : " #5
  6 = 'S45C'             : " #6
  7 = 1.000000000000     : " #7
  8 = 18.500000000000    : " #8
  9 = 18.500000000000    : " #9
  end                    :
** end item              : データの終り

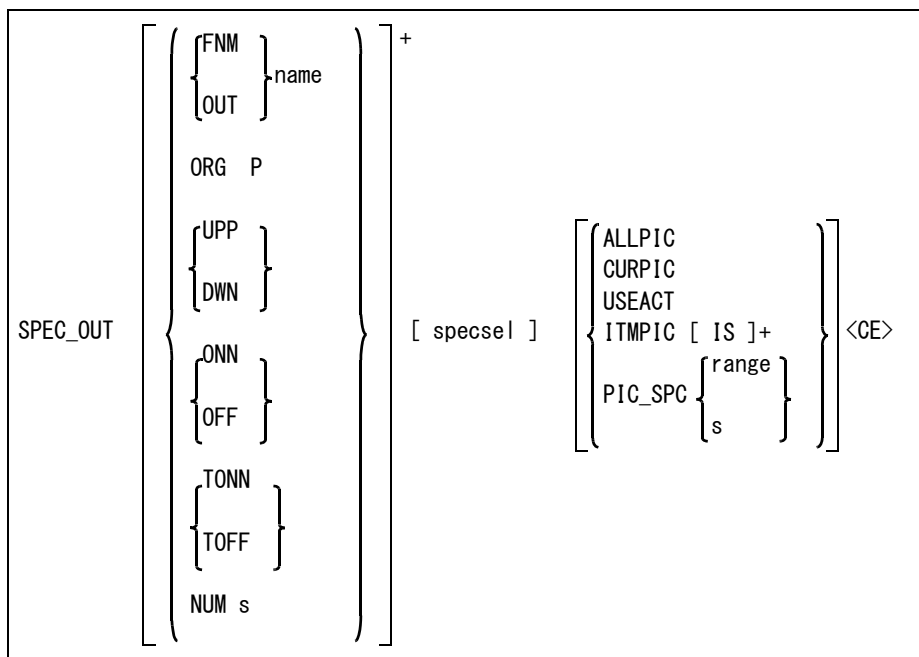
```

3.2.5.2 部品表 (特性集計表) を作る

【メニュー】

〔特性データ〕 → 〔部品表〕

【構文】



集計表をファイルに出力したいときにファイル名を指示する。

- FNM name : 出力ファイル名を入力する。デフォルトのファイル拡張子は .OUT となる。
画面とファイルに部品表を作る。
- OUT name : 出力ファイル名を入力する。デフォルトのファイル拡張子は .OUT となる。
ファイルに部品表を作る。画面には表示しない。

表の位置を指示する。

- ORG P : 表の原点をテンポラリポイントで入力する。省略するとモデル原点に作表する。作表原点は作表終了時に次に書き込み可能な場所に設定されるので、アイテムを選択して1行ずつ付け加えることができる。

並べる方向を指示する。

- UPP : 下から上へ並べる。(省略時)
- DWN : 上から下へ並べる。

タイトルを表示するかどうか指示する。

- ONN : タイトルを表示する。(省略時)
- OFF : タイトルを表示しない。

SPEC_OUT UPP <CE>

下から上		
9	TEMPERATURE	TEMP
8	TEST PERIOD	
7	TEST NAME	
6	TEST	
5	BUFFER TIME	
4	FILE FILTER	
3	FILE UNIT	
2	FILE LINE	
1	FILE UNIT	
0	FILE FILTER	
00	PARTS NAME	00

SPEC_OUT DWN <CE>

上から下		
番号	項目名	単位
1	HEATED PRESSURE 加熱圧力	F-1
2	VELOCITY 速度	
3	VELOCITY 速度	
4	VELOCITY UNIT 速度単位	
5	VELOCITY 速度	
6	VELOCITY 速度	
7	VELOCITY 速度	
8	VELOCITY 速度	
9	VELOCITY 速度	

アイテム単位・タイトルなし

2	VELOCITY 速度	
5	VELOCITY 速度	
9	VELOCITY 速度	
6	VELOCITY 速度	

トータル計算を表示するかどうか指示する。

TONN : トータルを表示する。(省略時)
TOFF : トータルを表示しない。

表にするレコード番号を指示する。

NUM s : レコード番号を入力する。

作表の対象アイテムを選択定義する。

specsel : 選択定義の検索条件を指示する。詳細は下記のとおり。

【specsel の部分の構文】

SELONN FNM filename	$\left[\begin{array}{c} s \\ d \end{array} \right] \text{data}$	$\left[\begin{array}{c} s \\ d \end{array} \right] \text{data}$	INF filename	SELOFF
---------------------	--	--	--------------	--------

検索条件をテキストファイルから読み込み一括設定する

FNM filename : ファイル名を入力する。設定されていたデータはクリアされる。ファイルのディレクトリおよびファイル拡張子は ACAD.SET のキーワード #SPEC:TXT# で定義する。

各項目ごとに設定する場合は項目番号をつぎのいずれかで指示する。

テキストファイルにより設定された項目にあらたに条件を追加することもできる。

s : 追加する項目番号を入力する。
d : 追加する項目をデジタイズで指示する。

データを入力する

data : データを入力する。

設定された検索条件のデータをテキストファイルに出力する。

INF filename : ファイル名を入力する。ファイルのディレクトリおよびファイル拡張子は ACAD.SET のキーワード #SPEC:OUT# で定義する。

検索を実行する。

SELOFF : 検索項目の設定が終了し SELOFF を入力すると、検索条件を満たす特性レコードを持つアイテムを捜し出してアクティブリストにのせる。

作表の対象アイテムをつぎのいずれかの方法で指示する。

ALLPIC : モデルのすべてのアイテム。
 CURPIC : アクティブピクチャのすべてのアイテム。
 USEACT : アクティブリスト内のアイテム。
 ITMPIC IS : 選択したアイテム。
 PIC_SPC range : 範囲で指定されたピクチャのアイテム。range はピクチャ番号の範囲。
 1-10 ピクチャ1から10まで
 1-10, 14 ピクチャ1から10までとピクチャ14
 PIC_SPC s : 指定されたピクチャのアイテム。s はピクチャ番号。

作表開始を指示する。

<CE> : 作表開始を実行する。

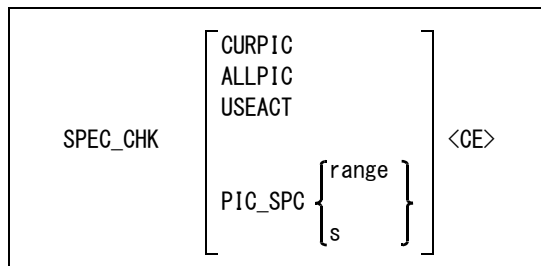
3.2.6 特性データの確認

3.2.6.1 同一特性データを持ったシンボルを確認する

【メニュー】

〔特性データ〕→〔特性チェック〕

【構文】



検索の対象となるシンボルを次のいずれかの方法で指定する。

CURPIC : アクティブ・ピクチャのシンボル。
 ALLPIC : 全てのピクチャのシンボル。
 USEACT : アクティブリスト内のシンボル。
 PIC_SPC range : 範囲で指定されたピクチャのシンボル。range はピクチャ番号の範囲。
 1-10 ピクチャ1から10まで
 1-10, 14 ピクチャ1から10までとピクチャ14
 PIC_SPC s : 番号で指定されたピクチャのシンボル。

検索を開始を指示する。

<CE> : 検索を開始する。

第4章 パーツ

● 概要

パーツは、Advance CAD で作成したパラメトリック図形（APG）とパーツ情報ファイル（パーツ定義ファイル）をもとにパーツ形状または穴形状を配置できます。パーツにはあらかじめ特性データを定義してあるので部品表を作成することもできます。この機能を利用すると設計製図において、部品作成の労力を軽減でき、作業の効率アップが可能です。

● パーツ機能の特長

- (1) APG の場合はサイズ変更の度にパラメータを設定する必要があります。
パーツではあらかじめサイズごとのパラメータをパーツ定義ファイルに登録しますので、サイズ名を指示するだけですみます。
- (2) サブモデル、シンボルの場合はサイズ別にサブモデルファイル、シンボルファイルを登録する必要があります。パーツではパラメータデータを APG に代入し使用しますので、APG ファイルを投影図ごとに1つずつ作成するだけで使用できます。
- (3) 特性データを付加することができます。
パーツ定義ファイルに特性データを定義することによって、自動的にパーツに特性データを付加します。
- (4) 未定義のパラメータを実行時に設定できます。
同一のパーツでも実行時に未定義パラメータを変更することによって形状を変更できます。
- (5) 既存のパーツから同一サイズのパーツを呼び出せます。
- (6) 各投影図のパーツを簡単に呼び出せます。
- (7) 投影図上に複数の同一パーツ（サイズ名とパーツ名が同じ）をグループ化することにより、特性データのトータル数を1つにすることができます。
- (8) サイズ別にファイルを作成する必要がありませんので使用ディスク量が減少します。

パーツの各ファイルは基本的にオプションで有償になります。ただし、サンプルとしてボルト／ナットのパーツデータを標準リリースいたします。

4.1 パーツの定義

Advance CAD においてパーツを使用する場合、パーツの情報はパーツ単位のパーツマスターファイルに含まれています。

パーツの情報を追加，更新，新規作成した場合にはパーツマスターファイルを新たに作成しなければなりません。

4.1.1 パーツに必要なファイル

パーツに必要なファイルは次の4つです。

- (1) パーツ A P G 形状ファイル
パーツ、穴形状の A P G ファイルです。1つのパーツに対してパーツ用、穴形状用の各投影図にあった A P G を作成する。
- (2) パーツ定義ファイル
 - ・ パーツの投影図に対応するそれぞれの A P G ファイル名
 - ・ 寸法表のサイズに対応した値を A P G の変数に代入するためのリスト
 - ・ パーツに付ける特性データの内容
- (3) パーツコントロールファイル
パーツ定義ファイル名を登録する。
- (4) パーツマスターファイル
Advance CAD でパーツを使用するときに参照されるファイルである。
パーツマスターファイルはプログラム partsgen で作成する。

プログラム

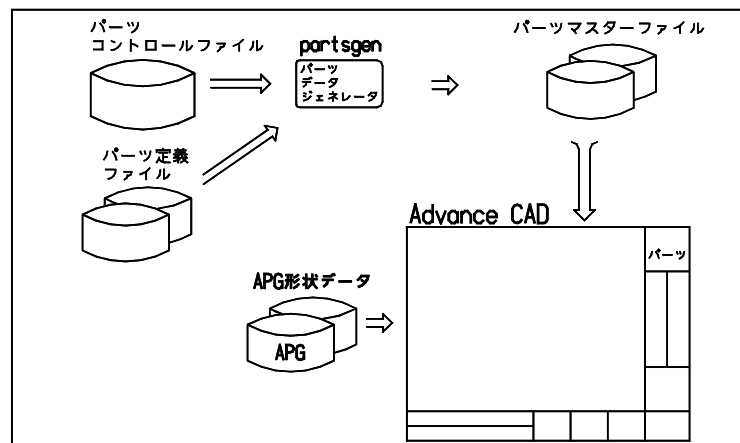
```
prompt% partsgen <CR>
```

Parts Input Control File Name : パーツコントロールファイル名 <CR>

各ファイルのディレクトリ名とファイル拡張子は、ファイル ACAD.SET で指定します。ACAD.SET のキーワードはそれぞれ以下の通りです。

- ・ パーツ A P G 形状ファイル
#PARTS:APG# ""~/parts/apg/" !.APX! Parts APG file
- ・ パーツ定義ファイル
#PARTS:INP# ""~/parts/prt/" !.INP! Parts master input file
- ・ パーツコントロールファイル
#PARTS:CTL# ""~/parts/prt/" !.CTL! Parts control file
- ・ パーツマスターファイル
#PARTS:BIN# ""~/parts/prt/" !.PRT! Parts master DB file

パーツ概念図



4.1.2 パーツコントロールファイル

パーツコントロールファイル名は英字で始まる大文字で作成します。
 パーツコントロールファイルには、パーツ定義ファイル名だけを記述します。

ファイルは以下の記述でテキストエディタで作成します。

- コメント行
 1 カラム目がスラッシュ (/) またはスペースで始まっている行はコメント行。
- パーツ定義ファイル名の記述行
 1 カラム目がスラッシュ (/) またはスペース以外で始まる行は、パーツ定義ファイル名を示す。
 ファイル名にはファイル拡張子は付けない。

例

```

/           : コメント行
/ TEST     : コメント行
/           : コメント行
PART1      : パーツ定義ファイル名
PART2      : パーツ定義ファイル名
PART3      : パーツ定義ファイル名
  
```

4.1.3 パーツ定義ファイルの作成

パーツ定義ファイルは各パーツごとに個別に作成します。ファイル名は英字で始まる大文字で作成します。

ファイルにはパーツ名、特性データ情報、APG 情報、共通データ情報、サイズリスト見出し情報、パーツサイズ情報を記述します。

ファイルは以下の記述でテキストエディタで作成します。

1 カラム目がスラッシュ (/) またはスペースで始まっている行はコメント行を表します。

\$\$ で始まる行は節の始まりを示します。各節で使用する記号 !! <> "" (), は定義内容の区切りを表しているのを省略しないでください。

パーツ定義ファイルでは、パーツ定義ファイル中に使用する数値・文字列を定数として定義できます。定数を定義できるのは \$\$PARTS キーワード後から \$\$END キーワードまでです。定義した定数は中括弧 ("{}") で定義名を囲むことにより定数値に展開されます。また、定数は式使用可の時のみ使用できます。

定数の定義方法

t	!const!	(value)
---	---------	---------

const	:	定数名（空白を含まない文字列）
value	:	定数値（整数／実数／文字列）

キーワード

\$\$PARTS	:	パーツ名定義キーワード
\$\$SPEC	:	特性データ定義キーワード
\$\$APG	:	APG 定義開始キーワード
\$\$COMMON	:	共通データ定義開始キーワード
\$\$TITLE	:	サイズリスト見出し文字列キーワード
\$\$SIZE	:	パーツサイズ別定義開始キーワード
\$\$END	:	定義終了キーワード

(1) パーツ名定義

\$\$PARTS	!parts_name!	<class#, apg_f>	"comment"
-----------	--------------	-----------------	-----------

parts_name	:	パーツ名を英数字で記述する。
class#	:	配置クラス番号。（省略可）
apg_f	:	APG 配置フラグ。（省略可）
		0 : ASC 配置（省略時）
		1 : APG 配置
comment	:	パーツのタイトル名。（最大 64 文字、省略可）

(2) 特性データの定義

\$\$SPEC	<spec_file#, spec_rec#>
----------	-------------------------

spec_file#	:	特性データのコントロールファイル番号（整数）
spec_rec#	:	特性データのレコード番号（整数）

(3) パーツ、穴形状の APG ファイル名の定義

\$\$APG		
\$TOP	!top_apg_name!	"hole_apg_name"
\$BOTTOM	!bottom_apg_name!	"hole_apg_name"
\$FRONT	!front_apg_name!	"hole_apg_name"
\$BACK	!back_apg_name!	"hole_apg_name"
\$RIGHT	!right_apg_name!	"hole_apg_name"
\$LEFT	!left_apg_name!	"hole_apg_name"

*_apg_name	:	各投影図に対応したパーツの APG ファイル名（英数字）
hole_apg_name	:	各投影図に対応する穴形状の APG ファイル名（英数字）ファイル名は最大 32 文字まで。

(4) 共通データ定義

特性データの各項目内容、APGのパラメータを記述する。

```
$$COMMON      "size_name"
s <item_number,type> "value"
p !param! (value) <undef,type> "text"
```

size_name : パーツ共通データサイズ名（省略可）
 サイズリストがない時に、仮のサイズ名として使用される。
 省略時はサイズ名が“-”（ハイフン）となる。
 カリキュレータ変数を大括弧（“[]”）で囲むことにより、配置時にカリキュレータ変数値がサイズ名と置き換わる。

s = 特性データ定義

item_number : 特性データの項目番号（整数）
 type : 項目データタイプ（整数）
 1 : 数値データ
 0 : 文字列データ
 value : 値（整数／実数／文字列）
 カリキュレータ変数を大括弧（“[]”）で囲むことにより、配置時にカリキュレータ変数値がパーツ情報と置き換わる。
 以下のキーワードを中括弧（“{}”）で囲むことにより、配置時にカレントパーツ情報と置き換わる。
 NAME : パーツ名
 SIZE : サイズ名

p = APG データ定義

param : APG パラメータ名（英数字）
 value : APG パラメータ数値（整数／実数／式）
 値未定義の時は、未定義パラメータのデフォルト値とする。（省略可）
 undef : 値未定義フラグ（整数、省略可）
 0 : 値定義済み（省略時）
 1 : 値未定義
 type : パラメータタイプ（整数、省略可）
 0 : 距離（省略時）
 1 : 角度
 text : プロンプト文字列（省略可）
 省略時はパラメータ名を使用

(5) サイズリスト見出し文字列定義（省略可）

```
$$TITLE      "strings"
```

strings : サイズリスト見出し文字列（最大 64 文字）

(1) パーツサイズ名ソート（省略可）

```
$$LSORT
$$CSORT
$$NSORT
```

\$\$LSORT : サイズ名の長さとお名前によるソート（省略時）
 \$\$CSORT : サイズ名のみによるソート
 \$\$NSORT : 定義順によるソートなし

(6) パーツのサイズ別定義

```

$$$SIZE      "size_name"      !addtext!
s <item_number,type> "value"
p !param! (value) <undef,type> "text"

```

size_name : パーツサイズ名 (最大 32 文字)
 addtext : 追加文字列 (最大 32 文字)
 s, p は (4) 共通データの定義と同様。

(7) 定義終了

```

$$$END

```

4.1.4 パーツ定義ファイル例

例 1

```

/
/      PARTS Input filename : NEJI-B1180.INP
/
$$$PARTS !PARTS01! <100>
/
$$$SPEC <001,001>
/
$$$APG
$TOP      !P1TOP!
$BOTTOM   !P1BOT!
$FRONT     !P1FRT!
$BACK      !P1BAK!
$RIGHT     !P1RIT!
$LEFT      !P1LFT!
/
$$$COMMON
s <1,0> "parts01"
s <2,1> "10.0"
p !W! (100.0) "幅"
/
$$$TITLE "サイズ (幅 × 高さ)"
/
$$$SIZE !80X70! "10 - 50"
s <3,1> "80.0"
s <4,1> "70.0"
p !A! (80.0) <0>
p !B! (70.0) <0>
p !H! (100.0) <1,0> "高さ"
p !ANG! (90.0) <1,1> "角度"
/
$$$SIZE !100X100!
s <3,1> "100.0"
s <4,1> "100.0"
p !A! (100.0)
p !B! (100.0)
p !H! (100.0) <1> "高さ"
/
$$$END

```

例 2 - 1

J I S の六角ナットの寸法を各サイズの A P G パラメータデータとして使用する。また、ネジの呼びをサイズとして使用する。

ネジの呼び	M1.6	M2	M2.5	M3	(M3.5)	M4	M5
DA	1.6	2	2.5	3	3.5	4	5
DW	2.4	3.1	4.1	4.6	5.1	5.9	6.9
E	3.41	4.32	5.45	6.01	6.58	7.66	8.79
M	1.3	1.6	2	2.4	2.8	3.2	4.7
MM	0.84	1.08	1.4	1.72	2.04	2.32	3.52
S	3.2	4	5	5.5	6	7	8

ネジの呼び	M5	M6	M8	M10	M12	(M14)	M16
DA	5	6	8	10	12	14	16
DW	6.9	8.9	11.6	14.6	16.6	19.6	22.5
E	8.79	11.05	14.38	17.77	20.03	23.35	26.75
M	4.7	5.2	6.8	8.4	10.8	12.8	14.8
MM	3.52	3.92	5.15	6.43	8.3	9.68	11.28
S	8	10	13	16	18	21	24

例 2 - 2

例 2 - 1 のデータでパーツ定義ファイルを作成するための記述例

```

$$PARTS    !NUTS!      ..... パーツ名を定義する。
                                "$$PARTS" の後に定義したいパーツ名を !! で囲む。
$$APG      ..... ここから下は A P G ファイル名の定義であることを示す。
$TOP       !NUTS-XY!    ..... 平面図に NUTS-XY. APX という A P G を使用することを示す。
$FRONT     !NUTS-XZ!    ..... 正面図に NUTS-XZ. APX という A P G を使用することを示す。
$RIGHT     !NUTS-YZ!    ..... 右側面図に NUTS-YZ. APX という A P G を使用することを示す。

```

他にも \$BOTTOM、\$BACK、\$LEFT の後に !! で A P G ファイル名を囲むことによって各投影図の A P G データを定義できる。

```

$$COMMON      ..... パーツ "NUTS" には、各サイズ共通のデータがないため共通データ定義は省略する。

$$TITLE       " 呼び寸法 " ..... サイズリストを表示するときに表示するタイトルを示す。

$$SIZE        " M 2 "      ..... ここからは各サイズ別の A P G パラメータ特性データを定義することを示す。

```

各サイズ別と各サイズ共通の A P G パラメータと特性データの定義方法は同じ。

以下サイズ "M 2 " のパラメータ定義。
 パラメータ名は A P G のパラメータ名と同じ。

```
p !DA!      (2. 00000) <0> " ネジの内径 "
               ..... DA に 2. 00000 を定義する。
               ..... 値未定義フラグに 0 を定義する。プロンプト文字列を定義する
               ..... ことにより再設定ができる。

p !DW!      (3. 10000) ..... DW に 3. 10000 を定義する。
p !E!       (4. 32000) ..... E に 4. 32000 を定義する。
p !M!       (1. 60000) ..... M に 1. 60000 を定義する。
p !MM!      (1. 08000) ..... MM に 1. 08000 を定義する。
p !S!       (4. 00000) ..... S に 4. 00000 を定義する。

$$SIZE      "M 2. 5 " ..... サイズ別指定の繰り返し
:           :
:           :

$END        ..... パーツの定義の終了を示す。
```

例 2 - 3

例 2 - 1 のデータをもとに作成したパーツ定義ファイル

```
/
/ 規格 NO.  NUTS  名称  ナット
/
$PARTS !NUTS! " ナット "
$APG
$TOP    !NUTS-XY!
$FRONT  !NUTS-XZ!
$RIGHT  !NUTS-YZ!
$$TITLE " サイズ "
$$SIZE  "M1. 6"
p !DA!  (      1. 60000)
p !DW!  (      2. 40000)
p !E!   (      3. 41000)
p !M!   (      1. 30000)
p !MM!  (      0. 84000)
p !S!   (      3. 20000)
$$SIZE  "M2"
p !DA!  (      2. 00000)
p !DW!  (      3. 10000)
p !E!   (      4. 32000)
p !M!   (      1. 60000)
p !MM!  (      1. 08000)
p !S!   (      4. 00000)
$$SIZE  "M2. 5"
p !DA!  (      2. 50000)
p !DW!  (      4. 10000)
p !E!   (      5. 45000)
p !M!   (      2. 00000)
p !MM!  (      1. 40000)
p !S!   (      5. 00000)
$$SIZE  "M3"
p !DA!  (      3. 00000)
p !DW!  (      4. 60000)
p !E!   (      6. 01000)
p !M!   (      2. 40000)
p !MM!  (      1. 72000)
```

```

p !S! ( 5. 50000)
$$$SIZE "(M3. 5)"
p !DA! ( 3. 50000)
p !DW! ( 5. 10000)
p !E! ( 6. 58000)
p !M! ( 2. 80000)
p !MM! ( 2. 04000)
p !S! ( 6. 00000)
$$$SIZE "M4"
p !DA! ( 4. 00000)
p !DW! ( 5. 90000)
p !E! ( 7. 66000)
p !M! ( 3. 20000)
p !MM! ( 2. 32000)
p !S! ( 7. 00000)
$$$SIZE "M5"
p !DA! ( 5. 00000)
p !DW! ( 6. 90000)
p !E! ( 8. 79000)
p !M! ( 4. 70000)
p !MM! ( 3. 52000)
p !S! ( 8. 00000)
$$$SIZE "M6"
p !DA! ( 6. 00000)
p !DW! ( 8. 90000)
p !E! ( 11. 05000)
p !M! ( 5. 20000)
p !MM! ( 3. 92000)
p !S! ( 10. 00000)
$$$SIZE "M8"
p !DA! ( 8. 00000)
p !DW! ( 11. 60000)
p !E! ( 14. 38000)
p !M! ( 6. 80000)
p !MM! ( 5. 15000)
p !S! ( 13. 00000)
$$$SIZE "M10"
p !DA! ( 10. 00000)
p !DW! ( 14. 60000)
p !E! ( 17. 77000)
p !M! ( 8. 40000)
p !MM! ( 6. 43000)
p !S! ( 16. 00000)
$$$SIZE "M12"
p !DA! ( 12. 00000)
p !DW! ( 16. 60000)
p !E! ( 20. 03000)
p !M! ( 10. 80000)
p !MM! ( 8. 30000)
p !S! ( 18. 00000)
$$$SIZE "(M14)"
p !DA! ( 14. 00000)
p !DW! ( 19. 60000)
p !E! ( 23. 35000)
p !M! ( 12. 80000)
p !MM! ( 9. 68000)
p !S! ( 21. 00000)
$$$SIZE "M16"
p !DA! ( 16. 00000)
p !DW! ( 22. 50000)
p !E! ( 26. 75000)
p !M! ( 14. 80000)

```

```
p !MM! (      11.28000)
p !S!   (      24.00000)
$$END
```

4.2 パーツセット

● パーツセット概要

パーツセットは、パーツセット情報ファイル(パーツセット定義ファイル)をもとに複数の単品部品(パーツ)をひとつの組にしたユニットです。

パーツセットはユニットとして配置できるので特性データも別に定義できます。

● パーツセット機能の特長

- (1) パーツセットではユニットのメンバーになるパーツをパーツセット定義ファイルに登録する必要があります。関係ある複数個のパーツ配置をするとき、パーツセット配置で簡単にできます。
- (2) パーツセットに特性データを付加することができます。
パーツとパーツセットは個々に特性データを持つので各々の部品表を作成することができます。
- (3) パーツセット配置もパーツ配置と同手順なので、簡単に形状が呼び出せます。
- (4) 配置された複数の同一パーツセット(サイズ名とパーツセット名が同じ)をグループ化することにより、特性データのトータル数を1つにすることができます。
また、パーツセットに含まれるパーツはパーツセットのグループ化により自動的にグループ化されます。

4.3 パーツセットの定義

Advance CAD においてパーツセットを使用する場合、パーツセットの情報はパーツセットマスターファイルに含まれています。
パーツセットの情報を追加・更新・新規作成した場合にはパーツセットマスターファイルを新たに作成しなければなりません。

4.3.1 パーツセットに必要なファイル

パーツセットに必要なファイルは次の3つです。(パーツ配置に必要なパーツマスターファイル・APG ファイルも必要です)

- (1) パーツセット定義ファイル
 - 配置するパーツ名およびサイズ名
 - パーツセットに付ける特性データの内容
- (2) パーツセットコントロールファイル
パーツセット定義ファイル名を登録する。
- (3) パーツセットマスターファイル
Advance CAD でパーツセットを使用するときに参照されるファイルである。
パーツセットマスターファイルはプログラム psetgen で作成する。

プログラム psetgen は、Advance CAD 起動ディレクトリで実行する。

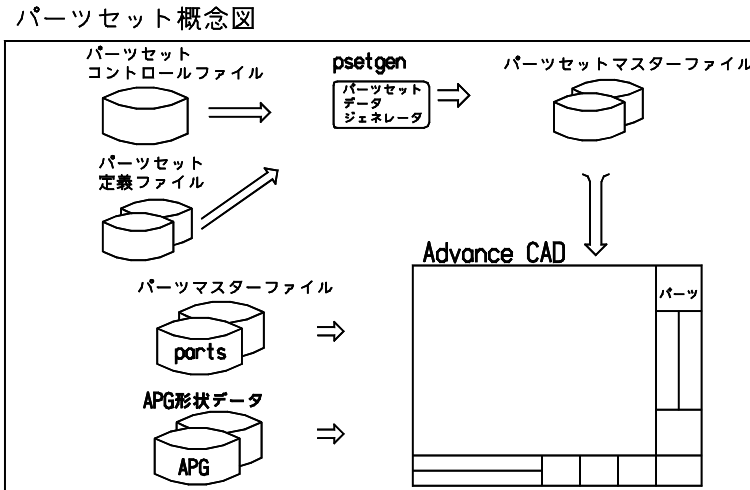
```
prompt% psetgen <CR>
```

Parts Set Input Control File Name : パーツセットコントロールファイル名 <CR>

- (4) パーツマスターファイル・APG ファイル
Advance CAD でパーツを配置するとき参照されるファイルである。

各ファイルのディレクトリ名と拡張子は、ファイル ACAD.SET で指定する。
ACAD.SET のキーワードはそれぞれ以下の通り。

- パーツセット定義ファイル
#PSET:INP# "~/parts/pset/" !.INP! Parts Set master input file
- パーツセットコントロールファイル
#PSET:CTL# "~/parts/pset/" !.CTL! Parts Set master control file
- パーツセットマスターファイル
#PSET:BIN# "~/parts/pset/" !.PST! Parts Set master DB file



4.3.2 パーツセットコントロールファイル

パーツセットコントロールファイル名は英字で始まる大文字で作成します。
 パーツセットコントロールファイルには、パーツセット定義ファイル名だけを記述します。

ファイルは以下の記述でテキストエディタで作成します。

コメント行

1 カラム目がスラッシュ (/) またはスペースで始まっている行はコメント行。

パーツセット定義ファイル名の記述

1 カラム目がスラッシュ (/) またはスペース以外で始まる行はパーツセット定義ファイル名を示す。ファイル名にはファイル拡張子は付けない。

例

/	: コメント行
/ TEST	: コメント行
/	: コメント行
PSET1	: パーツセット定義ファイル名
PSET2	: パーツセット定義ファイル名
PSET3	: パーツセット定義ファイル名

4.3.3 パーツセット定義ファイルの作成

パーツセット定義ファイルは各パーツセットごとに個別に作成します。ファイル名は英字で始まる大文字で作成します。

ファイルにはパーツセット名、特性データ情報、共通データ情報、サイズ情報、パーツ情報を記述します。

ファイルは以下の記述でテキストエディタで作成します。

1 カラム目がスラッシュ (/) またはスペースで始まっている行はコメント行を表します。

\$\$ で始まる行は節の始まりを示します。各節で使用する記号 !! <> () は定義内容の区切りを表しているなので省略しないでください。

パーツセット定義ファイルでは、パーツセット定義ファイル中に使用する数値・文字列を定数として定義できます。定数を定義できるのは \$\$PSET キーワード後から \$\$END キーワードまでです。定義した定数は中括弧 ("{}") で定義名を囲むことにより定数値に展開されます。また、定数は式使用可の時のみ使用できます。

定数の定義方法

t	!const! (value)
---	-----------------

const: 定数名 (空白を含まない文字列)

value: 定数値 (整数／実数／文字列)

キーワード

\$\$PSET	: パーツセット名定義キーワード
\$\$SPEC	: 特性データ定義キーワード
\$\$VIEW	: V I E W 情報キーワード
\$\$PARTS	: パーツ名定義キーワード
\$\$COMMON	: 共通データ定義開始キーワード
\$\$TITLE	: サイズリスト見出し文字列定義キーワード
\$\$SIZE	: パーツセットサイズ別定義開始キーワード
\$\$END	: 定義終了キーワード

(1) パーツセット名定義

\$\$PSET	!pset_name! <class#> "comment"
----------	--------------------------------

pset_name	: パーツセット名を英数字で記述する。
class#	: 配置クラス番号。(省略可)
comment	: パーツセットのタイトル名。(最大 64 文字、省略可)

(2) 特性データの定義

\$\$SPEC	<spec_file#, spec_rec#>
----------	-------------------------

spec_file#	: 特性データのコントロールファイル番号 (整数)
spec_rec#	: 特性データのレコード番号 (整数)

(3) V I E W 情報定義

\$\$VIEW	!vname!		
\$TOP	!view_top!	<mirr_top,	ang_top>
\$BOTTOM	!view_bottom!	<mirr_bottom,	ang_bottom>
\$FRONT	!view_front!	<mirr_front,	ang_front>
\$BACK	!view_back!	<mirr_back,	ang_back>
\$RIGHT	!view_right!	<mirr_right,	ang_right>
\$LEFT	!view_left!	<mirr_left,	ang_left>

vname	: V I E W 情報定義名 V I E W 情報定義名は 32 文字まで。 V I E W 情報数は 128 まで。
view_*	: 投影図定義 (省略可) top : 平面図 bottom : 底面図 front : 正面図 back : 背面図 right : 右側面図

left : 左側面図
 unuse : 使用しない
 mirr_* : 配置ミラー (整数/実数/式)・(省略可)
 00 : ミラーなし
 01 : Y ミラー
 10 : X ミラー
 11 : X Y ミラー
 ang_* : 配置角度 (整数/実数/式)・(省略可)

(4) パーツ名定義

```
$$PARTS num !parts_name1! "vname1"
```

num : パーツ番号
 parts_name* : パーツファイル名
 ファイル名は最大 32 文字まで。
 パーツ名定義は 128 まで。
 vname* : デフォルト V I E W 情報名 (省略可)

(5) 共通データ定義

```
$$COMMON "size_name"
s <item#, type> "value"
p <parts#, arr> "size" [xorg, yorg, zorg, mirr, ang] !vname!
a !param! (value) <undef, type> "text"
```

size_name : パーツセット共通データサイズ名 (省略可)
 サイズリストがない時に、仮のサイズ名として使用される。
 省略時はサイズ名が "-" (ハイフン) となる。
 カリキュレータ変数を大括弧 ("[] ") で囲むことにより、配置時にカリキュレータ変数値がサイズ名と置き換わる。

s = 特性データ定義

item# : 特性データの項目番号 (整数)
 type : 項目データタイプ (整数)
 0 : 値定義済み (省略時)
 1 : 値未定義
 value : 値 (整数/実数/文字列)
 カリキュレータ変数を大括弧 ("[] ") で囲むことにより、配置時にカリキュレータ変数値がパーツセット情報と置き換わる。
 以下のキーワードを中括弧 ("{ } ") で囲むことにより、配置時にカレントパーツセット情報と置き換わる。
 NAME : パーツセット名
 SIZE : サイズ名

p = パーツデータ定義

parts# : パーツ番号 (整数)
 arr : 配置フラグ (整数/実数/式)・(省略可)
 0 : 配置する (省略時)
 1 : 配置しない
 size : パーツサイズ名 (省略可)
 省略時はパーツセットサイズ名を使用する。
 xorg : X 方向原点オフセット
 yorg : Y 方向原点オフセット
 zorg : Z 方向原点オフセット
 mirr : ミラー配置
 00 : ミラーなし
 01 : Y ミラー
 10 : X ミラー
 11 : X Y ミラー
 xorg, yorg, zorg, ang, mirr は省略時は 0 とする位置パラメタである。(整数/

実数／式）・（省略可）
 ang, mirr は V I E W 情報を定義したときは使用されない。（デフォルト
 V I E W 情報も含む）

ang : 配置角度
 vname : V I E W 情報名（省略可）

a = パラメータデータ定義

param : パラメータ名（英数字）
 value : パラメータ数値（整数／実数／式）
 値未定義の時は、未定義パラメータのデフォルト値とする。（省略可）

undef : 値未定義フラグ（整数、省略可）
 オペレーションで設定可能な未定義変数は最大64個まで有効。ただし、パーツと
 パーツセットで設定できる未定義変数の合計を32個以内にする。こと。
 0 : 値定義済み（省略時）
 1 : 値未定義

type : パラメータタイプ（整数、省略可）
 0 : 距離（省略時）
 1 : 角度

text : プロンプト文字列（省略可）
 省略時はパラメータ名を使用

(6) サイズリスト見出し文字列定義

```
$$TITLE "strings"
```

strings : サイズリスト追加見出し文字列（最大 64 文字）

(7) パーツセットサイズ名ソート

```
$$LSORT  
$$CSORT  
$$NSORT
```

\$\$LSORT : サイズ名の長さで名前によるソート（省略時）
 \$\$CSORT : サイズ名のみによるソート
 \$\$NSORT : 定義順によるソートなし

(8) パーツセットサイズ別定義

```
$$$SIZE "size_name" !addtext!  
s <item#, type> "value"  
p <parts#, arr> "size" [xorg, yorg, zorg, mirr, ang] !vname!  
a !param! (value) <undef, type> "text"
```

size_name : パーツセットサイズ名（最大 32 文字）
 addtext : 追加文字列（最大 32 文字）

s, p, a は (5) 共通データ定義と同様。

(9) 定義終了

```
$$END
```

4.3.4 パーツセット定義ファイル例

例 1
/

```

/      PSET Input filename : NBSET.INP
/
$$PSET !NBSET! "ナット・ボルトセット"
/
t      !n!      (0)
t      !xon!    (10)
t      !yon!    (01)
t      !xyon!   (11)
/
$$VIEW !BOLTS!
$BOTTOM !top!
$BACK   !front!      <{xyon}>
/
$$VIEW !NUTS!
$TOP    !front!      <{yon}>
$BOTTOM !front!      <{yon}>
$FRONT  !top!        <{yon}>
$BACK   !top!        <{xon}>
$LEFT   !right!      <{xon}>
/
$$PARTS 1      !B1180_S_C_X_X! "BOLTS" / Bolts
$$PARTS 2      !B1163_X_X_X_X! "NUTS" / Nuts
/
$$NSORT
/
/$$COMMON
a !XOFF!      <1, 0> "距離"
p      <1>
$$TITLE "Size List"
$$SIZE "M5x010"
p      <2> "M5" [XOFF]
$$SIZE "M5x012"
p      <2> "M5" [XOFF]
$$SIZE "M5x016"
p      <2> "M5" [XOFF]
$$SIZE "M5x020"
p      <2> "M5" [XOFF]
$$SIZE "M5x025"
p      <2> "M5" [XOFF]
$$SIZE "M5x030"
p      <2> "M5" [XOFF]
$$SIZE "M5x035"
p      <2> "M5" [XOFF]
$$END

```

4.4 コマンド

● コマンド一覧

コマンド名	機能
PARTS	パーツを配置する
PSET	パーツセットを配置する
PRTGRP	パーツ・パーツセットをグループにする
PRTUPD	パーツ・パーツセットを更新する
PRTDEL	パーツ・パーツセットを削除する
PVER	パーツ・パーツセットを確認する
PSIZE	パーツ・パーツセットのサイズ変更をする
PMOVE	パーツ・パーツセットを移動する
PDIM	パーツに寸法を追加する
PBREAK	パーツ・パーツセットのグループを解除する
PACT	パーツ・パーツセットをアクティブリストにのせる
PSETBRK	パーツセットを分解しパーツにする
MKHOLE	パーツから穴形状を作成する
HOLE	穴形状を配置する
PASCCLR	パーツ・パーツセットのアソシエイトクリーンアップ
PASCLIST	パーツ・パーツセットのアソシエイト情報を表示する

Windows 版ではダイアログボックスを使ってパーツ・パーツセット名の入力、サイズの入力、パラメータの設定、情報の表示を行うことができます。ダイアログボックスを使用しない場合は従来通りサブウィンドウを使用します。
ダイアログボックスの使用についてはマニュアル『コマンドリファレンス』11.1 モデル定数の DLGMODE を参照してください。

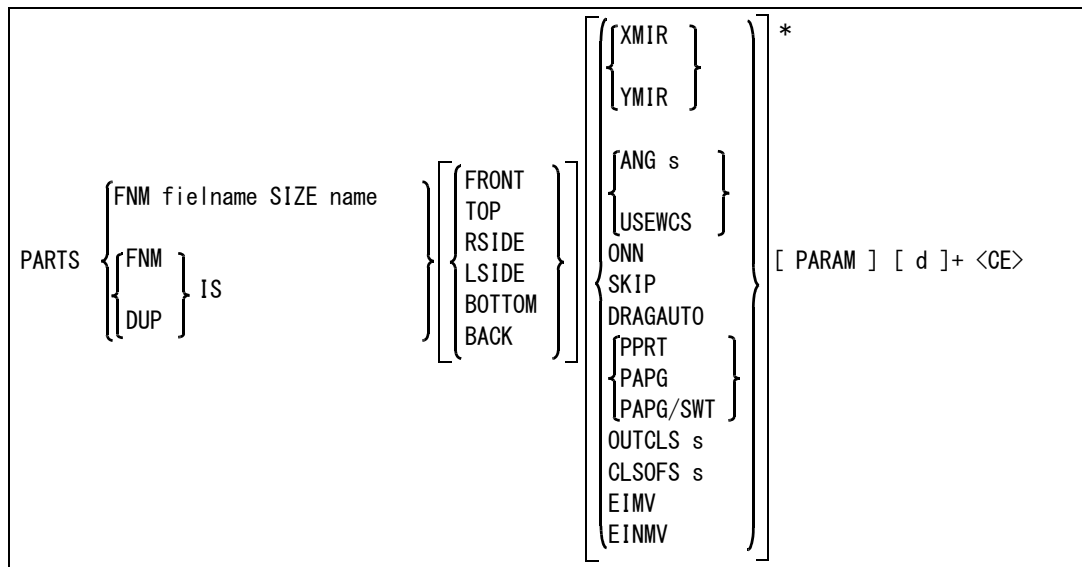
Ver20 以降『APG 登録』でエラーとなったアイテムは『パーツ』や『セット』で配置原点を指定しても原点移動されなくなりました。
Ver19 以前と同じように原点を移動させたい場合は、修飾子 EIMV を使用して移動可能な状態にしてからコマンドを実行してください。
システム管理者の手引き『第2章. 起動と環境設定』 - 『2.2 コンフィグレーションファイル』 - 『2.2.6 その他の指定』 参照

4.4.1 パーツを配置する

【メニュー】

[パーツ] → [パーツ]

【構文】



配置するパーツ名をつぎのいずれかで指定する。

- FNM filename : 配置するパーツ名を入力する。ワイルドカードが使用できる。Windows 版ではワイルドカードを入力した場合ダイアログボックスを使って名前が選択できます。
- FNM IS : 既に配置されているパーツを選択する。選択されたパーツと同じパーツ名、同じサイズのパーツを配置する。
- DUP IS : 既に配置されているパーツを選択する。選択されたパーツと同じパーツ名、同じサイズ、同じパラメータ値のパーツを配置する。

配置するパーツのサイズを指定する。

- SIZE name : 配置するパーツのサイズ名を入力する。ワイルドカードが使用できる。Windows 版ではワイルドカードを入力した場合ダイアログボックスを使ってサイズが選択できます。

配置するパーツの投影図を指定する。

- TOP : 平面図
- FRONT : 正面図
- RSIDE : 右側面図
- LSIDE : 左側面図
- BOTTOM : 下面図
- BACK : 背面図

パーツを反転して配置したいときに指定する。

- XMIR : X 反転の指示、または解除スイッチ。
- YMIR : Y 反転の指示、または解除スイッチ。

配置角度を指定する。

- ANG s : 数値で角度を指定する。
- USEWCS : 補助座標の X 軸の角度を配置角度とする。

配置したパーツに寸法を付ける。

- ONN: 寸法アイテムの作成を指示、または解除スイッチ。

クラス 254 のアイテムを作成しないための指示をする。

- SKIP : クラス 254 のアイテムを作成、または解除スイッチ。

ドラッグングの方法を指示する。

DRGAUTO : 配置するときにパーツをドラッグする。
指定されると外形、ボックス、図形全体とドラッグ方法が変更される。

配置するパーツのクラスを指示する。省略されるとカレントクラスになる。

OUTCLS s : 数値を入力する。
CLSFS s : 加算数を入力する。カレントクラス、または指定されているクラスに加算する。

エラーが発生した場合の配置動作を指示する。

EIMV : エラーアイテムを移動する。Ver19 以前の動作
EINMV : エラーアイテムを移動しない（初期値）Ver20 以降の動作

配置するパーツのアイテム構成を指示する。

PPRT : アソシエイトアイテムと A P G アイテムで構成されたパーツを配置する。特性データはアソシエイトアイテムに付加される。この配置の場合、編集コマンド (COPY など) を実行すると配置された A P G アイテムのみ対象となり、パーツアイテムと認識されない。
PAPG : A P G アイテムのみで構成されたパーツを配置する。特性データは A P G アイテムに付加される。この配置の場合、パーツのグループ化 (PRTGRP) の対象にはならないが、編集コマンド (COPY など) で作成されたアイテムはパーツアイテムと認識される。
PAPG/SWT : 配置するパーツのアイテム構成スイッチ。

パーツの配置を開始する。

パーツ定義ファイルの中に未定義パラメータが設定されているパーツマスターを使用した場合、未定義パラメータ入力を開始します。

PARAM : パーツ名を IS で選択した場合、その選択されたパーツの配置時パラメータ情報を未定義パラメータに設定します。
Windows 版でダイアログボックスを使用した時このオプションは無効になります。

未定義パラメータは下記のようにサブウインドウに表示される。

未定義名 リスト 1: H寸法 = 未定義 2: L寸法 = 未定義 未定義名を選択し値入力

変更したいパラメータをピック、パラメータ項番又はプロンプト名の入力を行い距離又は角度を次のように入力する。

- キーボード入力の場合
250.5<CR>
- 距離入力の場合
スペースキーを入力し図形表示に戻し、図形上の始点と終点をデジタイズする。
- 角度入力の場合
スペースキーを入力し図形表示に戻し、図形上の中心点、始点、終点を順にデジタイズする。

パラメータ入力後、画面のプロンプトとその変数の値が再表示される。

<CR> を押すとパラメータ入力終了する。

Windows 版でダイアログボックスを使用する場合

未定義パラメータはダイアログボックスに表示されます。

変更したいパラメータのテキストフィールドに距離又は角度を入力します。
 この場合図形上のデジタイズによって距離又は角度の入力できません。
 ダイアログボックスを終了するとパラメータ入力が終了します。

パーツを配置する位置を指示する。

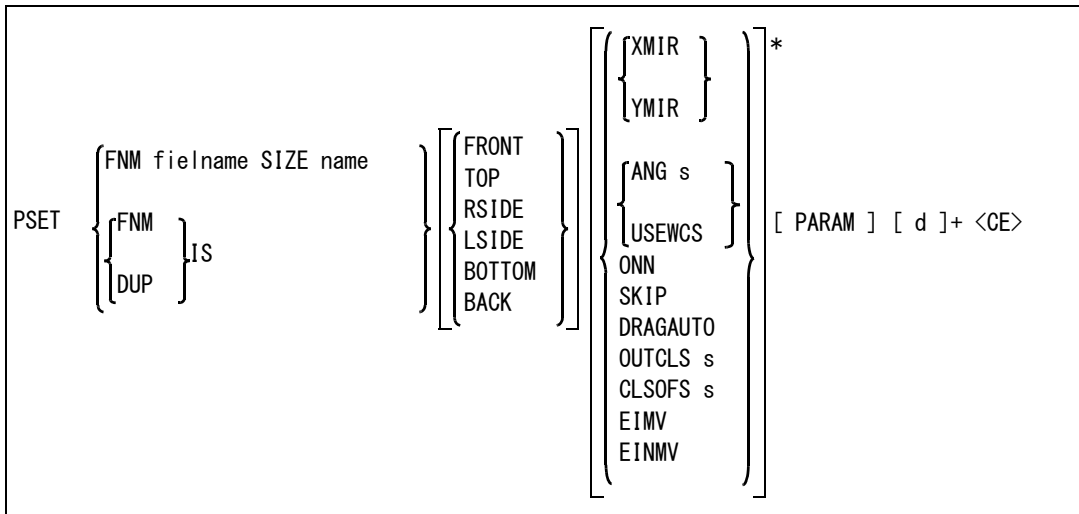
d : 任意の位置をデジタイズする。

4.4.2 パーツセットを配置する

【メニュー】

〔パーツ〕 → 〔セット〕

【構文】



配置するパーツセット名をつぎのいずれかで指定する。

- FNM filename : 配置するパーツセット名を入力する。ワイルドカードが使用できる。Windows 版ではワイルドカードを入力した場合ダイアログボックスを使って名前が選択できます。
- FNM IS : 既に配置されているパーツセットを選択する。選択されたパーツセットと同じパーツセット名、同じサイズのパーツセットを配置する。
- DUP IS : 既に配置されているパーツセットを選択する。選択されたパーツセットと同じパーツセット名、同じサイズ、同じパラメータ値のパーツセットを配置する。

配置するパーツセットのサイズを指定する。

- SIZE name : 配置するパーツセットのサイズ名を入力する。ワイルドカードが使用できる。Windows 版ではワイルドカードを入力した場合ダイアログボックスを使ってサイズが選択できます。

配置するパーツセットの投影図を指定する。

- TOP : 平面図
- FRONT : 正面図
- RSIDE : 右側面図
- LSIDE : 左側面図
- BOTTOM : 下面図
- BACK : 背面図

パーツセットを反転して配置したいときに指定する。

- XMIR : X 反転の指示、または解除スイッチ。
- YMIR : Y 反転の指示、または解除スイッチ。

配置角度を指定する。

ANG s : 数値で角度を指定する。
USEWCS : 補助座標の X 軸の角度を配置角度とする。

配置したパーツセットに寸法を付ける。

ONN : 寸法アイテムの作成を指示、または解除スイッチ。

クラス 254 のアイテムを作成しないための指示をする。

SKIP : クラス 254 のアイテムを作成、または解除スイッチ。

ドラッグングの方法を指示する。

DRGAUTO : 配置するときにパーツセットをドラッグングする。指定されると外形、ボックス、図形全体とドラッグング方法が変更される。

配置するパーツセットのクラスを指示する。省略されるとカレントクラスになる。

OUTCLS s : 数値を入力する。
CLS0FS s : 加算数を入力する。カレントクラス、または指定されているクラスに加算する。

エラーが発生した場合の配置動作を指示する。

EIMV : エラーアイテムを移動する。Ver19 以前の動作
EINMV : エラーアイテムを移動しない（初期値）Ver20 以降の動作

パーツセットの配置を開始する。

パーツセット定義ファイルの中に未定義パラメータが設定されているパーツセットマスターを使用した場合、未定義パラメータ入力を開始します。

PARAM : パーツセット名を IS で選択した場合、その選択されたパーツセットの配置時パラメータ情報を未定義パラメータに設定します。Windows 版でダイアログボックスを使用した時このオプションは無効になります。

ダイアログボックスを使用しない場合

従来通りサブウインドウに未定義パラメータが表示されます。変更したいパラメータをピックアップ、パラメータ項番又はプロンプト名の入力を行い距離又は角度を入力します。<CE>を入力するとパラメータ入力が終了します。

未定義パラメータは下記のようにサブウインドウに表示される。

未定義名 リスト	
1: H寸法	= 未定義
2: L寸法	= 未定義
未定義名を選択し値入力	

変更したいパラメータをピックアップ、パラメータ項番又はプロンプト名の入力を行い距離又は角度を次のように入力する。

- キーボード入力の場合
250.5<CR>
- 距離入力の場合
スペースキーを入力し図形表示に戻し、図形上の始点と終点をデジタイズする。
- 角度入力の場合
スペースキーを入力し図形表示に戻し、図形上の中心点、始点、終点を順にデジタイズする。

パラメータ入力後、画面のプロンプトとその変数の値が再表示される。
<CR> を押すとパラメータ入力が終了する。

Windows 版でダイアログボックスを使用する場合

未定義パラメータはダイアログボックスに表示されます。
変更したいパラメータのテキストフィールドに距離又は角度を入力します。
この場合図形上のデジタイズによって距離又は角度の入力はできません。
ダイアログボックスを終了するとパラメータ入力 that 終了します。

パーツセットの配置する位置を指示する。
d : 任意の位置をデジタイズする。

4.4.3 パーツ・パーツセットをグループ化する

【メニュー】

〔パーツ〕 → 〔グループ化〕

【構文】

```
PRTGRP IS [ ISauto ]+ <CE>
```

パーツ・パーツセットは投影図にそれぞれ配置された同一部品に関して、アソシエイトによりグループ関係を持つことができる。

IS : グループの基準となるアイテムを指示する。
ISauto : グループに含めるアイテムを選択する。

4.4.4 パーツ・パーツセットを更新する

【メニュー】

〔パーツ〕 → 〔更新〕

【構文】

```
PRTUPD { [ ITMPIC ] [ ISauto ]+  
ALL  
USEACT  
}
```

部品形状の変更若しくは特性データの変更を行った場合、このコマンドを実行することにより部品が更新される。指定された同一部品をすべて更新する。

ITMPIC : ALL , USEACT の指定を解除する。

更新するパーツ・パーツセットを以下のどれかで指定する。

ISauto : 選択したパーツ・パーツセットを更新する。
ALL : 全てのパーツ・パーツセットを更新する。
USEACT : アクティブリスト中のパーツ・パーツセットアイテムを更新する。

このコマンドは以下の時に使用して下さい。

- PARTS , PSET コマンドを使用したとき「パートマスターが更新されています」のメッセージが出たとき

- PVER コマンドで『DB との対応』が NG となっており、下に DB の日付が出たとき

4.4.5 パーツ・パーツセットを削除する

【メニユ一】

〔パーツ〕 → 〔削除〕

【構文】

$$\text{PRTDEL} \left\{ \left[\begin{array}{c} \text{PTOP} \\ \text{PGRP} \end{array} \right] \left[\begin{array}{c} \text{PALL} \\ \text{PPRT} \\ \text{PITM} \end{array} \right] \left\{ \begin{array}{l} [\text{ISauto}] + \\ \text{USEACT} \\ \left\{ \begin{array}{c} \text{CURPIC} \\ \text{ALLPIC} \end{array} \right\} \end{array} \right\} \left[\begin{array}{c} \text{ONN} \\ \text{OFF} \end{array} \right] \right\} \langle \text{CE} \rangle$$

グループ化されたパーツ・パーツセットの処理方法を指示する。

PTOP : 指定したパーツ・パーツセットをグループ化単位で削除する。
PGRP : 指定したパーツ・パーツセットを配置単位で削除する。

削除対象アイテムを以下のどれかで指示する。

PALL : アイテム全てを削除する。(パーツ・パーツセットも含まれる)
PPRT : パーツ・パーツセットだけを削除する。
PITM : パーツ・パーツセット以外のアイテムを削除する。

削除するパーツ・パーツセットを以下のどれかで指定する。

ISauto	:	削除するアイテムを選択する。
ALLPIC	:	モデル全体のアイテムを削除する。
CURPIC	:	現在のピクチャのアイテムを削除する。
USEACT	:	アクティブリスト中のアイテムを削除する。

ピクチャ指定で削除する場合、選択マスクを使うかどうか指定する。

ONN : 選択マスク (CLS/SEL, ITM/SEL) で選択可能なアイテムだけを削除する。
OFF : 全てのアイテムを削除する。

4.4.6 パーツ・パーツセットを確認する

【メニユ一】

〔パーツ〕→〔ベリファイ〕

【構文】

$$\text{PVER} \left[\left\{ \begin{array}{c} \text{PSETLIST} \\ \text{SPECLIST} \end{array} \right\} \right] \left\{ \begin{array}{c} \text{IS} \\ \text{USEACT} \end{array} \right\} \langle \text{CE} \rangle$$

選択されたパーツまたはパーツセットが白色表示される。グループ化されている場合は、同じグループ内のパーツ・パーツセットも白色表示される。

パーツセットの情報表示時にパーツセットの中のパーツ情報も表示したい時に指定する。(パーツセットを選択したときのみ有効)

PSETLIST : パーツセット中のパーツも表示したいときに指定する。

特性データを表示したいときに指定する。

SPECLIST : パーツ・パーツセットに定義されている特性データを表示したいときに指定する。
複数のパーツ・パーツセットが選択された場合、最初のパーツ・パーツセットの特性データだけを表示します。

確認するパーツ・パーツセットを以下のどれかで指定する。

IS : 確認するパーツ・パーツセットを選択する。
USEACT : アクティブリスト中のパーツ・パーツセットを確認する。

※ パーツ・パーツセットの確認では、次の情報が表示される。

カテゴリー	グループ化されているパーツ数
パーツ・パーツセットタイトル	配置クラス パーツマスター作成日付
パーツ・パーツセット名	特性ファイル番号 DB との対応
サイズ名	特性レコード番号

ダイアログボックスを使用しない場合

サブウインドウにパーツ・パーツセットの情報が表示されます。
<CE> が入力されると表示を終了します。

Windows 版でダイアログボックスを使用する場合

ダイアログボックスのリストにパーツ・パーツセットの情報が表示されます。
ダイアログボックスを終了すると表示を終了します。

4.4.7 パーツ・パーツセットをサイズ変更する

【メニュー】

〔パーツ〕 → 〔サイズ変更〕

【構文】

PSIZE IS1 { name IS2 } <CE>

サイズを変更するパーツ・パーツセットを選択する。

IS1 : パーツ・パーツセットを選択する。

変更後のサイズを次のいずれかで指示する。

name : サイズ名を入力する。ワイルドカードが指定できる。
IS2 : 配置済みのパーツ・パーツセットをピックアップする。
ピックアップしたパーツ・パーツセットと同じサイズになる。

4.4.8 パーツ・パーツセットを移動する

【メニュー】

〔パーツ〕 → 〔移動〕

【構文】

PMOVE	ISauto	[vec]	<CE>
-------	--------	---------	------

- ISauto : 移動するパーツ・パーツセットを選択する。
- vec : 移動量をベクトルで入力し指示する。省略すると前回指定した値を使用する。
- <CE> : 移動を開始する。

4.4.9 パーツに寸法を付ける

【メニュー】

〔パーツ〕 → 〔寸法付け〕

【構文】

PDIM	$\left\{ \begin{array}{l} [ITMPIC] \text{ ISauto} \\ \text{ALL} \\ \text{USEACT} \end{array} \right\}$
------	--

既に配置してあるパーツに寸法を付ける。

- ITMPIC : ALL , USEACT の指定を解除する。

寸法をつけるパーツを以下のどれかで指定する。

- ISauto : 寸法を付けるパーツを選択する。
- ALL : 全パーツに寸法線を付ける。
- USEACT : アクティブリスト中のパーツに寸法線を付ける。

単品パーツのみに寸法を付けます。グループ化されたパーツ・パーツセットであっても選択されたパーツにだけ寸法が付きます。

4.4.10 パーツ・パーツセットのグループを解除する

【メニュー】

〔パーツ〕 → 〔グループ解除〕

【構文】

PBREAK	IS [ISauto]*	<CE>
--------	--------------	------

- IS : グループ化されているパーツ・パーツセットを選択する。
- ISauto : 指示したグループから除外するパーツ・パーツセットを選択する。

4.4.11 パーツ・パーツセットをアクティブリストにのせる

【メニュー】

〔パーツ〕 → 〔PACT〕

【構文】

PACT	$\left[\begin{array}{l} \text{ALLPIC} \\ \text{CURPIC} \\ [\text{ITMPIC}] \end{array} \right] [\text{ISauot}] + \left[\begin{array}{l} \text{ALL} \\ \text{PPRTLST} \\ \text{PSETLST} \end{array} \right] \left[\begin{array}{l} \text{PALL} \\ \text{PPRT} \\ \text{PPST} \\ \text{PTOP} \end{array} \right] \langle \text{CE} \rangle$
------	---

パーツ・パーツセットのアソシエイトアイテムをアクティブリストにのせる。

アクティブリストにのせるパーツ・パーツセットの存在するピクチャを指示する。

ITMPIC : ALLPIC, CURPIC の指定を解除する。

パーツ・パーツセットの存在するピクチャを以下のどれかで指定する。

ALLPIC : モデル全体のパーツ・パーツセットをのせる。
 CURPIC : 現在のピクチャのパーツ・パーツセットをのせる。
 ISauto : アクティブリストにのせるパーツ・パーツセットを選択する。

選択対象アイテムを以下のどれかで指示する。

ALL : パーツ・パーツセットを両方対象とする。
 PPRTLST : パーツのみを対象とする。
 PSETLST : パーツセットのみを対象とする。

選択されたパーツ・パーツセットのアソシエイトアイテムを以下の処理方法で指示してアクティブリストにのせる。

PALL : 全てのアソシエイトアイテムをのせる。
 パーツ : パーツのアソシエイトアイテムをのせる。
 パーツセット : パーツ・パーツセットのアソシエイトアイテムをのせる。
 PPRT : パーツのアソシエイトアイテムをのせる。
 パーツ : パーツのアソシエイトアイテムをのせる。
 パーツセット : パーツのアソシエイトアイテムのみをのせる。
 PSET : パーツセットのアソシエイトアイテムをのせる。
 パーツ : アソシエイトアイテムをアクティブリストにのせない。
 パーツセット : パーツセットのアソシエイトアイテムのみをのせる。
 PTOP : 一番上位のアソシエイトアイテムをのせる。
 パーツ : パーツのアソシエイトアイテムをのせる。
 パーツセット : パーツセットのアソシエイトアイテムのみをのせる。

4.4.12 パーツセットを解除する

【メニュー】

〔パーツ〕 → 〔セット解除〕

【構文】

PSETBRK IS <CE>

パーツセットを分解しパーツにする。

IS : パーツセットを選択する。

4.4.13 パーツ・パーツセットのクリーンナップ

【メニュー】

〔パーツ〕 → 〔ASC クリア〕

【構文】

PASCCLR	$\left\{ \begin{array}{c} Y \\ N \end{array} \right\}$	<CE>
---------	--	------

現モデルのアソシエイト（パーツ・パーツセット関係）をチェックし、正常なつながりを持っていないアソシエイトの修復・削除を行う。

"Y <CE>" をキーボードから入力すると、クリーンナップを開始する。

"N <CE>" をキーボードから入力すると、コマンドをキャンセルする。

配置されている APG アイテムにアソシエイトが付いていれば、パーツ・パーツセットになる。付いていない場合、APG はパーツ・パーツセットではなく APG アイテムになる。

4.4.14 パーツ・パーツセットのアソシエイトリスト

【メニュー】

〔パーツ〕 → 〔ASC リスト〕

【構文】

PASCLIST	[IS]	<CE>
----------	--------	------

モデル中のアソシエイトアイテム（パーツ・パーツセット）のアソシエイト情報を表示する。

以下の情報を表示する。

- ・ パーツ・パーツセット名・サイズサフィックス（アソシエイト名）
- ・ パーツ・パーツセットセット名・サイズサフィックス（アソシエイト名）
- ・ アソシエイトカテゴリー番号
- ・ アソシエイトアイテム ID ポインタ
- ・ アソシエイトされているアイテムの ID ポインタ

ダイアログボックスを使用しない場合

サブウィンドウにモデル中のアソシエイトアイテム（パーツ・パーツセット）のアソシエイト情報が表示されます。

<CE> が入力されると表示を終了します。

Windows 版でダイアログボックスを使用する場合

ダイアログボックスのリストにモデル中のアソシエイトアイテム（パーツ・パーツセット）のアソシエイト情報が表示されます。

ダイアログボックスを終了すると表示を終了します。

アソシエイトアイテムの ID ポインタは - をつけて表現される。

4.4.15 穴形状を配置する

【メニュー】

〔パーツ〕 → 〔穴形状配置〕

$$\text{HOLE} \left\{ \begin{array}{l} \text{FNM filename SIZE name} \\ \text{FNM IS} \end{array} \right\} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{FRONT} \\ \text{TOP} \\ \text{RSIDE} \\ \text{LSIDE} \\ \text{BOTTOM} \\ \text{BACK} \end{array} \right\} \end{array} \right\} \left[\begin{array}{l} \left\{ \begin{array}{l} \text{XMIR} \\ \text{YMIR} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{ANG s} \\ \text{USEWCS} \\ \text{SKIP} \\ \text{DRAGAUTO} \\ \text{INFNT s} \end{array} \right\} \end{array} \right]$$
$$\left[\begin{array}{l} \text{ADD [ISauto]}+ \\ \text{REL [ISauto]}+ \end{array} \right] \quad [\text{PARAM}] \text{ d } \langle \text{CE} \rangle$$

このコマンドはパーツの穴形状のみを作成する。
パーツとプレートが重なるようにパーツを配置した場合は、自動的にプレートをクリッピング範囲を形成するアイテムとみなす。
閉じていないプレートの場合は、アウトラインが作成できないため、穴形状は付加されない。
クリッピングされるアイテムは、デジタイズした点に配置される穴形状アイテムである。
なお、穴形状は、アソシエーションによりグループ関係を持っているため、各プレートと同じクラス番号が自動的に付けられる。

配置するパーツ名を次のどちらかで指定する。

FNM filename	:	配置するパーツ名を入力する。ワイルドカードが使用できる。Windows 版ではワイルドカードを入力した場合ダイアログボックスを使って名前が選択できます。
FNM IS	:	指示された穴形状からパーツ名を指定する。サイズも指示された穴形状と同サイズになる。

配置する穴形状のサイズを指定する。

SIZE name : 配置するパーツのサイズ名を入力する。ワイルドカードが使用できる。Windows NT 版ではワイルドカードを入力した場合ダイアログボックスを使ってサイズが選択できます。

配置する穴形状の投影図を指定する。

TOP	:	平面図
FRONT	:	正面図
RSIDE	:	右側面図
LSIDE	:	左側面図
BOTTOM	:	下面図
BACK	:	背面図

穴形状を反転して配置したいときに指定する。

XMIR : X 反転の指示、または解除スイッチ。
YMIR : Y 反転の指示、または解除スイッチ。

配置角度を指定する。

ANG s : 数値で角度を指定する。
USEWCS : 補助座標の X 軸の角度を配置角度とする。

クラス 254 のアイテムを作成しないための指示をする。

SKIP : クラス 254 のアイテムを作成、または解除スイッチ。

ドラッグングを指示する。

DRGAUTO : 配置するときにパーツをドラッグングする。
指定されると、外形、ボックス、図形全体とドラッグング方法が変更される。

陰線処理を指定する。指定をすると、プレートに穴形状でクリップする。クリップされた内側部分の線種を変更できる。陰線処理では閉じていないアイテムも対象になる。

INFNT s : 陰線処理後の線種を入力する。
指定されると、プレートの実線部分が陰線処理の対象になる。
0 または 1 を入力すると、陰線処理は実行されない。
-1 を入力すると、クリップされた線は見えなくなる。
プレートの始終点を含む部分をクリップするとプレートが閉じた形状にならない場合があるので注意。

クリッピング処理方法を指示する。

ADD ISauto : 穴形状でクリッピング処理をするプレートを指定する。
指定されたプレートに穴形状が重なっていても穴形状を作成する。
REL ISauto : 穴形状でクリッピング処理をしないプレートを指定する。
指定されたプレートに穴形状が重なっていても穴形状を作成しない。

穴形状の配置を開始する。

穴形状を作成するパーツ定義ファイルの中に未定義パラメータが設定されているパーツマスターを使用した場合、未定義パラメータ入力を開始します。

PARAM : 穴形状名を IS で選択した場合、その選択された穴形状の配置時パラメータ情報を未定義パラメータに設定します。Windows 版でダイアログボックスを使用した時このオプションは無効になります。

ダイアログボックスを使用しない場合

未定義パラメータはサブウインドウに表示されます。
変更したいパラメータのピック、パラメータ項番またはプロンプト名の入力を行い、距離または角度を入力します。

Windows 版でダイアログボックスを使用する場合

未定義パラメータはダイアログボックスに表示されます。
変更したいパラメータのテキストフィールドに距離又は角度を入力します。
この場合図形上のデジタイズによって距離又は角度の入力はできません。
ダイアログボックスを終了するとパラメータ入力が終了します。

穴形状を配置する位置を指示する。

d : 任意の位置をデジタイズする。

4.4.16 穴形状を作成する

【メニュー】

〔パーツ〕 → 〔穴形状作成〕

【構文】

MKHOLE	$\left\{ \begin{array}{l} [\text{ISauto}]+ \\ \text{USEACT} \end{array} \right\} [\text{INFNT } s] \left[\begin{array}{l} \text{ADD } [\text{ISauto}]+ \\ \text{REL } [\text{ISauto}]+ \end{array} \right] \langle \text{CE} \rangle$
--------	--

指定された穴形状パーツをもつパーツを基に穴形状を作成する。

穴形状を作成するパーツを次のどちらかで指定する。

- ISauto : 穴形状を作成するパーツ（ピン等）を指定する。
- USEACT : アクティブリスト中のすべてのパーツの穴形状を作成する。

陰線処理を指定する。指定をすると、プレートに穴形状でクリップする。クリップされた内側部分の線種を変更できる。陰線処理では閉じていないアイテムも対象になる。

- INFNT s : 陰線処理後の線種を入力する。
指定されると、プレートの実線部分が陰線処理の対象になる。
0 または 1 を入力すると、陰線処理は実行されない。
-1 を入力すると、クリップされた線は見えなくなる。
プレートの始終点を含む部分をクリップするとプレートが閉じた形状にならない場合があるので注意。

クリッピング処理方法を指示する。

- ADD ISauto : 穴形状でクリッピング処理をするプレートを指定する。
指定されたプレートに穴形状が重なっていても穴形状を作成する。
- REL ISauto : 穴形状でクリッピング処理をしないプレートを指定する。
指定されたプレートに穴形状が重なっていても穴形状を作成しない。

穴形状を作成する。

- ⟨CE⟩ : 穴形状の作成を開始する。

第 5 章 モデル検索

● 概要

モデル検索ユーティリティは、Advance CAD で作成されたモデルファイル、シンボルファイルなどのファイルを検索、管理します。

この機能を使用すると、ファイル名の検索／変更／削除／保護／読み込みなどを Advance CAD 上で行えます。

さらにファイルの管理機能として、ファイルに管理番号と所有者名を設定できます。

管理番号とは、モデル、シンボル、APG ファイルを格納するディレクトリに番号を付けたものです。したがって、ファイルがいくつかのディレクトリに分かれていてもディレクトリごとの検索が可能です。

所有者名の設定があるファイルは、属性を変更するときに所有者名を指定する必要がありますので、モデル保護をよりいっそう強化できます。

なお所有者名は画面に表示されず、"Owner" という文字列で所有者名の設定があることを表します。もしアカウンティングが設定されていれば、モデル、シンボル、APG の所有者名は、モデル、シンボル、APG のファイルを保存したときにアカウンティングのユーザ名と同じ名前が付けられます。

注) マクロから使用の場合、ダイアログボックスでは情報の受け渡しが行えないため、マクロから使用する場合はサブウィンドウ表示にしてからモデル検索を行ってください。

サブウィンドウ表示にするにはモデル定数設定で DLGMODE 0 (サブウィンドウ表示) にします。詳しくはマニュアル「コマンドリファレンス」11.1 モデル定数をご覧ください。

5.1 主な機能

- モデルファイルについて、それぞれ以下のことが行えます。
 - (1) ファイル名, 作成日, 更新日, 呼出し日, 管理番号, 所有者名, モデルタイトルなどを指定してモデルを検索し、画面に表示する。
 - (2) ファイル名, 管理番号, 所有者名を変更する。
 - (3) 書込み許可／禁止を設定する。
 - (4) ファイルを削除する。
 - (5) 検索結果をリスト出力する。
 - (6) モデルファイルの読み込み、登録する。
 - (7) ラスターイメージの表示。
 - (8) モデルデータの表示。
- シンボル, APG ファイルについて、それぞれ以下のことが行えます。
 - (1) ファイル名, 作成日, 更新日, 管理番号, 所有者名などを指定してシンボル, APG を検索し、画面に表示する。
 - (2) ファイル名, 管理番号, 所有者名を変更する。
 - (3) 書込み許可／禁止を設定する。
 - (4) ファイルを削除する。
 - (5) 検索結果をリスト出力する。
- 旧モデル, 旧シンボル, 旧 APG ファイルの検索
 - (1) ファイル名, 作成日, 更新日, 管理番号, 所有者名などを指定して検索し、画面に表示する。
 - (2) ファイルを復元・削除・廃棄（旧モデルのみ）する。
 - (3) 検索結果をリスト出力する。
- IGES, 共通, CADAM, DXF, 作画, マクロファイルについて、それぞれ以下のことが行えます。
 - (1) ファイル名, 作成日を指定して検索し、画面に表示する。
 - (2) ファイル名を変更する。
 - (3) 書込み許可／禁止を設定する。
 - (4) ファイルを削除する。
 - (5) リストを出力する。
- モデルの親子関係やモデルが参照しているサブモデル, シンボル, 部品名を一覧表示します。また逆に子供図をもっている、サブモデル, シンボル, 部品名を参照しているモデル名を一覧表示します。

親子関係とは

組立図などを作成する場合、モデルは組立図の型番号をモデルタイトルで持つことができますが、図面内に配置した形状によってはその形状だけの決まった型番号を持っていることがあります。モデル検索ではこのような形状の型番号を子供名として登録することができ、子供名を参照しているモデルの検索ができます。

部品名とは

関係アイテム（アソシエイトアイテム）につけられている名前、つまりアソシエイト名のことです。

所有者名とは

所有者名は一種のパスワード（暗号）です。

所有者名が設定されているファイルでは、ファイルの削除／ファイルの書込保護／ファイル名の変更／所有者名の変更をするときに、所有者名をキー入力しなければなりません。その入力された所有者名がファイルの所有者名と一致しなければ、それらの処理は行われません。

アカウンティングが設定されていると、モデル, シンボル, APG ファイルを保存したときに自動的にアカウンティングのユーザ名をセットします。

注) アカウントユーザ名が“SYSTEM”という名前で Advance CAD を使用する場合は、所有者名の問合せは行いません。

● 検索条件の文字列でつぎの指定ができます。

文字列にワイルドカード *、? と || [] (OR)、<> (～以外) が使用できる。

- * : 任意の文字列
- ? : 任意の 1 文字
- || : または (OR) 条件
- [] : または (OR) 条件
- <> : 指定された条件以外のもの

例)

- A* : A で始まるモデル名。
 - A*||B* : A または B で始まるモデル名。
 - [AB]* : A または B で始まるモデル名。(同上)
 - A*||B*<> : A および B 以外の文字で始まるモデル名。
 - [AB]*<> : A および B 以外の文字で始まるモデル名。(同上)
- “<>” は文字列の先頭か最後に入力する。
 “<>” の指定がある場合に“||”は AND として働く。

5.2 準備

5.2.1 コンフィグレーションファイルの確認

モデル検索するためにはコンフィグレーションファイル (デフォルト: ACAD.SET) に次のキーワードが必要です。

(1) 管理ファイルを作成するディレクトリの定義

```
#MODEL:MANAGE#           !.MDM!           Modelmanagement file
```

モデル検索を行うファイルのディレクトリ定義ファイル (ACADMMDM.SET) のあるディレクトリを指定します。

ファイル名は ACADMMDM.SET で固定です。

ここで指定したディレクトリに、検索のための管理ファイルを作成します。

定義がなければ、起動ディレクトリに作成します。

```
#MODEL:LOCALMG#          !.LMG!           Local Model management file
```

ローカルなモデル検索を行うファイルのディレクトリ定義ファイルのあるディレクトリを指定します。

ファイル名は ACADLOCALMDM.SET で固定です。

ここで指定したディレクトリに、検索のための管理ファイルを作成します。

(2) 検索用の表示テンプレートファイルがあるディレクトリの定義

```
#MDLIN:DISP#             "~/maint/"           !.MDT!           Model manage Disp template
```

モデル検索結果の表示フォーマットのテンプレートファイルのディレクトリを指定します。ファイル名は自由に付けることができます。複数のテンプレートファイルから選択して使用することができます。

(3) 管理番号のグループファイルがあるディレクトリの定義

```
#MDLIN:MGNO#             !.MGN!           Model manage Group mg_no.
```

モデルの管理番号のグループを指定するファイルのディレクトリを指定します。

ファイル名は ACADMMDM で固定です。

(4) リスト出力するファイルを格納するディレクトリの定義

```
#MDLIN:LIST#             !.MLS!           Model manage list file
```

検索結果をファイル出力する時のディレクトリを指定します。

定義がなければ、起動ディレクトリに作成します。

(5) リスト出力するテンプレートファイルがあるディレクトリの定義

```
#MDLIN:TEMP#             "~/maint/"           !.MTP!           Model manage print template
```

モデル検索結果のリスト出力フォーマットのテンプレートファイルのディレクトリを指定します。
 ファイル名は自由に付けることができます。複数のテンプレートファイルから選択して使用することができます。
 定義がなければ、起動ディレクトリにある拡張子 .MTP が付いているファイルを参照します。

(6) 検索するモデル名のリストファイルがあるディレクトリの定義

```
#MDLINF:MDLNAM#      "~/work/"      !.NAM!      Model Name List File
```

モデル検索の検索条件でモデル名のリストファイルを使用する場合にリストファイルのあるディレクトリを指定します。

(7) リスト出力するプリンタの定義

```
#PRINTER#
```

キーワードが定義されていないとプリンタに出力されません。
 また、出力には `aprint_w.bat` というバッチファイルを使用します。
`aprint_w.bat` は、ローカルにつながっているプリンタを使用します。`aprint_w.bat` は、プリンタ出力モジュール `xrfprint.exe` を使用します。

(8) バックグラウンドジョブのログ

```
#MDLINF:LOG#          "XRF"          !.LOG!          Model manage Log file
```

このキーワードを設定すると、プリント出力、プロッタ出力のバックグラウンドジョブのログを取ることができます。

注) 注意

モデル検索でファイル名の変更／削除／保護などを行ったあとは、Windows コマンドでファイル名の変更／削除／保護などをしないでください。誤って行なってしまった場合には、もう一度管理ファイルを作成し直してください。

5.2.2 モデル検索用ディレクトリ定義ファイルを作成する

5.2.2.1 ACADMDM.SET ファイルを作成する

モデル検索で検索対象にするモデル、シンボル、APG、親子関係は、**ACADMDM.SET** ファイルで定義します。このファイルはホストおよびディレクトリごとに管理番号を付けてディレクトリ単位で検索が可能です。したがって管理番号が違っていれば、モデル名が同一でも検索ができます。

ファイル **ACADMDM.SET** を置くディレクトリは、ファイル **ACAD.SET** のキーワード **#MODEL:MANAGE#** で定義します。

ACADMDM.SET に定義するキーワードは次の通りです。

```
#<CE>:EXIT#          検索で <CE> を終了にする。  
                     このキーワードがない場合 <CE> は検索開始となる。
```

#LIMITFREE#	Advance CAD バージョン 15 からのモデルタイトル及び頁タイトルの上限拡張に伴い、このキーワードを記述するとモデル検索でも上限拡張を有効にします。ただしバージョン 14 以下のモデル検索との互換性はありません。 このキーワードの記述がない場合はバージョン 14 以下と互換性があり、バージョン 14 までの制限に収まるように制限を越えた部分を無視して検索を行います。
#MDMACCESS# #MDMNOACCESS#	モデル検索結果の一覧表示で管理ファイルを参照する。 モデル検索結果の一覧表示で管理ファイルを参照しない。 #MDMNOACCESS# の場合には一覧表示で管理ファイルの競合は起こりません。“他のユーザが管理ファイルを使用中です”で待たされることがなくなります。 #MDMNOACCESS# とするとモデルタイトルは最初の 50 バイト分のみを表示します。それ以上の文字数を表示する場合は #MDMACCESS# を指定して下さい。
#MODEL:xxx#	モデルファイルを保存してあるディレクトリ。 xxx : 管理番号 (1 ~ 899)
#MODELINF:xxx#	モデル情報ファイルを保存してある、またはモデル検索で情報ファイルを作成するディレクトリ。 xxx : 管理番号 (1 ~ 999)
#SYMBOL:xxx#	シンボルファイルを保存してあるディレクトリ。 xxx : 管理番号 (1 ~ 899)
#APG:xxx#	APG ファイルを保存してあるディレクトリ。 xxx : 管理番号 (1 ~ 899)
#PASSWORD#	“password” 廃棄モデル検索を使用する場合のパスワードを設定します。 password:『管理ファイル』の修飾コマンド『パスワード』で得られた意味をなさない文字列を設定します。 このキーワードが設定されていなければ『廃棄モデル検索』は使用できません。
#DISCARDMDL#	“path” !.ext! 廃棄モデルを格納するディレクトリ。 path:パス名 ext:ファイル拡張子 このキーワードが設定されていなければ『廃棄モデル検索』は使用できません。
#MODELSAVE:xxx#	モデルの削除をしたときモデル名を保存しておくディレクトリ。 1 回目の削除ではファイルが削除され名前は残っています。 2 回目の削除でファイルに関する情報が全て消去されます。 xxx : 管理番号 (1 ~ 899)
#HISTORY:xxx#	検索履歴ファイルを保存しておくディレクトリ。 xxx : 管理番号 (1 ~ 899)
#MODELREF:xxx#	親子関係ファイルを保存しておくディレクトリ。 xxx : 管理番号 (1 ~ 899)
#HISTORYMAX:xxx#	検索履歴ファイルの最大世代数を指定します。 xxx : 世代数 このキーワードが指定されなかった場合は世代数は 10 になります。
#RASTER:xxx#	ラスターファイルを保存してあるディレクトリ。ファイル名はモデルファイル名と同じにしてください。 xxx : 管理番号 (1 ~ 999)
#FILESYSYSTEM#	“path_name”

#COMMENT#	"comment" 装置3のファイルシステムとして使用できるドライブを指定します。 ハードディスク・MO等が指定できます。フロッピーは指定できません。 装置3では、モデルダンプの新規・追加・上書きができます。 path_name : ドライブ名とパス名を指定します。 comment : 装置の注釈 例) #FILESYSTEM# "F:/" #COMMENT# "MO 640M"
#SYMLIST:xxx#	シンボルリストを保存してあるディレクトリ。 xxx : 管理番号 (1 ~ 899) 親子関係の検索でモデルが参照しているピクチャ番号表示したい場合に設定してください。モデルのシンボルリストを出力する必要があります。
#SUBLIST:xxx#	サブモデルリストを保存してあるディレクトリ。 xxx : 管理番号 (1 ~ 899) 親子関係の検索でモデルが参照しているピクチャ番号表示したい場合に設定してください。モデルのサブモデルリストを出力する必要があります。
#MODELDATE:xxx#	モデル最新呼び出し日を格納するディレクトリ。 xxx : 管理番号 (1 ~ 899) 注) このキーワードの指定がなければ、呼び出し日は更新日と同一になります。
#TABLEBASE#	検索できるモデル件数の制限はなくなりますが、一覧表示できるモデルは 3500 件までにになります。この指定をした場合は、絞り込み検索でモデル件数を 3500 以内に絞り込んで一覧表示してください。 キーワード #FILEBASE# を指定した場合よりも検索時間が短くなります。
#FILEBASE#	検索できるモデル件数と一覧表示できる制限はなくなりますが、ワークファイルにデータを書き出して、そのファイルの読み書きをするため、時間がかかります。 注) キーワード #TABLEBASE#, #FILEBASE# の指定がない場合は、検索できるモデル件数と一覧表示できる件数は 3500 件までです。
#COMMENT#	"XXXX" キーワード #MODEL:xxx#, #SYMBOL:xxx#, #APG:xxx# の直後に入れてください。管理番号の注釈を指定します。 "XXXX":注釈。(半角文字で最大 72 文字以内、全角文字で最大 36 文字以内) 例) ACADMDM.SET の内容 #MODEL:001# "/home/acad/files/" !MDL! #COMMENT# "共通モデル"
#BACKUP#	"directory" ファイルシステムにモデルダンプを行う場合の出力先のディレクトリ。
INCLUDE	"file_name" file_name で指定されたファイルから読み込みを行います。 読み込みが終了すると次の行から処理を続けます。 このキーワードは ACADMDM.SET 内でのみ使用できます。
#PAGETITLE#	ページタイトルでの検索を行います。このキーワードがない場合はページタイトルでの検索は行えません。
#MGNODIALOG#	モデル検索の最初に表示するダイアログを管理番号の選択ダイアログにする。

ホスト名指定

ディレクトリ名の前にホスト名を指定することができます。ホスト名を指定するとそのホストが使用可能状態にあるかないかを調べてから検索処理を行います。使用可能状態にない場合は、管理番号を表示するとその管理番号のところに "マウントされていません" と表示されます。

ホスト名の指定がなくマウント先のマシンが使用可能状態でないときに、モデル検索を起動させると待ち状態になります。

例. #MODEL:001# "CTC:/home/acad/files/" !MDL! (CTC がホスト名)

注1) 管理ファイル作成時、"ACADMDM.SET にエラーがあります" というメッセージが表示された場合は、Advance CAD を止めて、起動ディレクトリのファイル Model_Kensaku.lis を参照してください。

注2) ディレクトリ名は、モデル保存等をするときのディレクトリ名と一致していなければいけません。

注意事項

モデル検索の途中で何らかの理由で Advance CAD が異常終了した後、再度モデル検索 (モデル保存、呼び出し) を行おうとすると "他のユーザが管理ファイルを使用中です" とメッセージが表示され、次の処理に進めなくなる場合があります。これは、管理ファイルの同時使用を回避するためのロックファイルが残ったままになっているためです。

ACAD.SET のキーワード #MODEL:MANAGE# で指定したディレクトリ内のロックファイル "FILESTATUSxxx.MDM" を削除してください。

ロックファイル (FILESTATUSxxx.MDM) の管理方法は次のようになっています。

1. 管理ファイルのオープンで既にロックファイルがある場合、
ロックファイルが自分自身で作成したものであればロックしない。
”他のユーザが管理ファイルを使用中です” とはしない。
2. ロックファイルがある場合は、
”他のユーザが管理ファイルを使用中です” とメッセージ表示し
ロックファイルの内容 (ホスト名、プロセス ID、日付) を表示します。
3. Advance CAD 起動時起動ディレクトリにファイル
AdvanceCAD_Start.INF を作成し終了時に削除するようになっています。
ファイルの内容はホスト名とプロセス ID です。

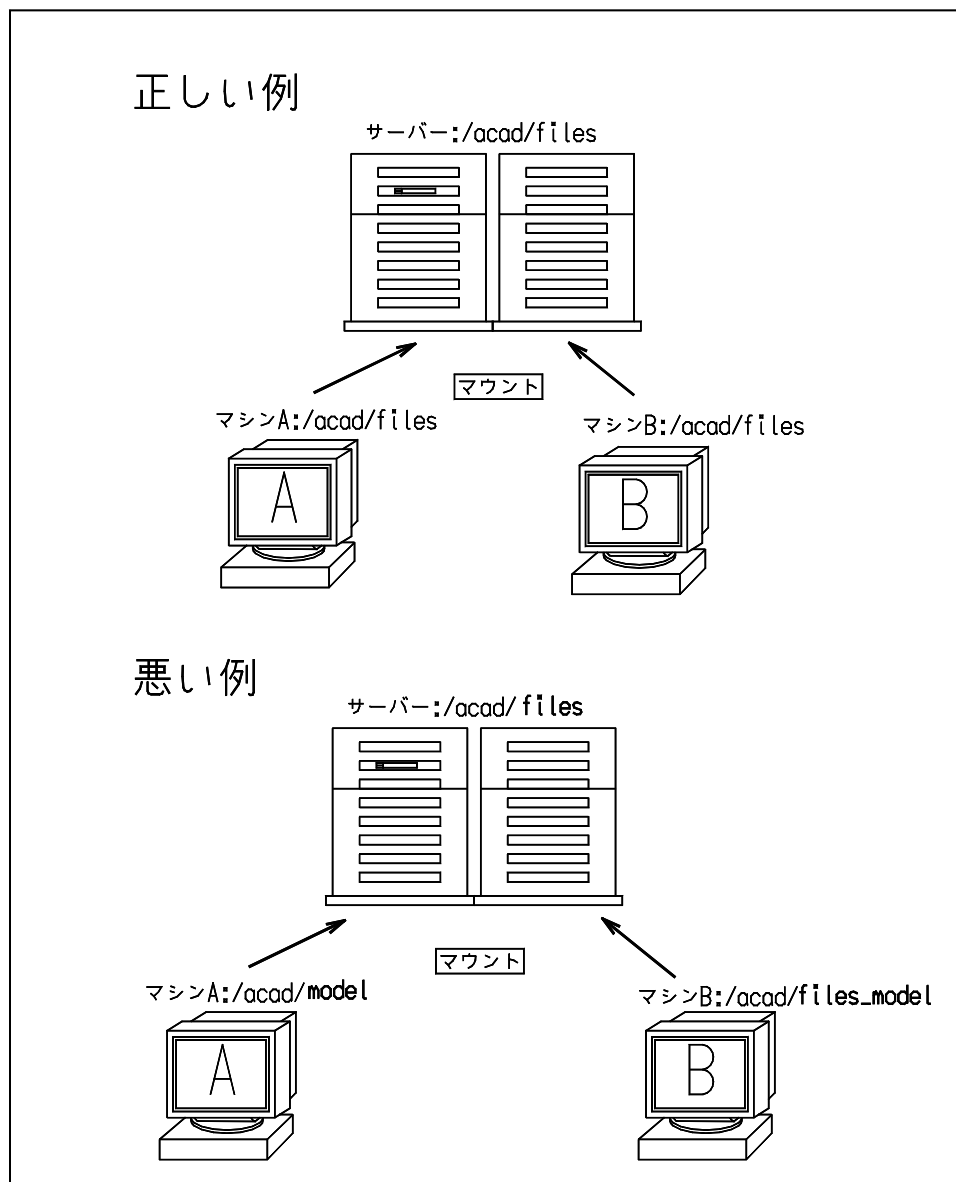
起動時の処理。

起動時にファイル AdvanceCAD_Start.INF が存在すれば
前回異常終了したとみなし自分自身が作成したロックファイルが
残っているかを調べ残っていれば削除します。

終了時の処理。

自分自身が作成したロックファイルが残っているかを調べ残っていれば
削除します。

4. モデル検索終了時 (ダイアログを閉じるとき) に自分自身が作成した
ロックファイルが残っていれば削除します。(ダイアログモードのみ)



● ACADMDM.SET のファイルフォーマット

#キーワード# "ディレクトリ" ! 拡張子! 注釈

- 第1カラムを/(スラッシュ)で始めると、その行はコメントです。
- 1行は80文字以内です。
- #キーワード#
キーワードでは、モデル・シンボル・APGなどの区別および管理番号を指定します。キーワードは、モデル・シンボル・APGおよびローカルを合わせて10000まで指定できます。

モデル情報ファイル(#MODELINF:xxx#)のキーワードには2種類あります。

- (1) モデル情報ファイルがモデルファイルと同じディレクトリにあれば、このキーワードは必要ありません。なお、モデル情報ファイルのキーワードを指定する場合は、モデルの管理番号とモデル情報の管理番号を一致させてください。

- (2) モデルがない場合は、キーワードの管理番号を 900 ～ 999 にします。この機能を使うと、手書きや他のソフトウェアで作成した図面の情報だけを Advance CAD のモデルタイトルに入力できるので、それらを含めてモデル検索が行えます。

注) ディレクトリ名の指定で ~ が使えますが、使用する場合は注意してください。複数のマシンで ACADMDM.SET を共有する場合、~ が示すディレクトリが各マシンで異なる場合は正しい管理が行えません。

5.2.2.2 ACADLOCALMDM.SET ファイルを作成する

ローカルなモデル検索を行うモデル・ファイルのディレクトリを指定します。
ローカルとはプライベートで使用しているディレクトリや ACADMDM.SET で指定した共用のディレクトリ以外にあるモデルファイルなどです。

#MODEL:xxxx# モデルファイルを保存してあるディレクトリ。
 xxxx: 管理番号 (1001 ～ 1999)

#COMMENT# "XXXX" キーワード #MODEL:xxxx# の直後に入れてください。管理番号の注釈を指定します。
 "XXXX" : 注釈。(半角文字で最大 72 文字以内、全角文字で最大 36 文字以内)

例) ACADLOCALMDM.SET の内容
 #MODEL:0001# "/home/acad/files/local/" !MDL!
 #COMMENT# "ローカルモデル"

#MODELINF:xxxx# モデル情報ファイルを保存してあるディレクトリ。
 xxxx : 管理番号 (1001 ～ 1999)

#MODELSAVE:xxxx# モデルの削除のときモデル名を保存しておくディレクトリ。1 回目の削除ではファイルは削除されますがモデル名は残っています。2 回目の削除でモデルに関するファイルが全て削除されます。
 xxxx : 管理番号 (1001 ～ 1999)

#HISTORY:xxxx# モデル情報ファイルの履歴を保存しておくディレクトリ。
 xxxx : 管理番号 (1001 ～ 1999)

#MODELREF:xxxx# 親子関係の情報を保存しておくディレクトリ。
 xxxx: 管理番号 (1001 ～ 1999)

#SYMLIST:xxx# シンボルリストを保存してあるディレクトリ。
 xxx : 管理番号 (1001 ～ 1999)

#SUBLIST:xxx# サブモデルリストを保存してあるディレクトリ。
 xxx : 管理番号 (1001 ～ 1999)

#MODELDATE:xxx# モデル最新呼び出し日を格納するディレクトリ。
 xxx : 管理番号 (1001 ～ 1999)

● ACADLOCALMDM.SET のファイルフォーマット

キーワード# "ディレクトリ" ! 拡張子! 注釈

- ・ 第 1 カラムを / (スラッシュ) で始めると、その行はコメントです。
- ・ 1 行は 80 文字以内です。

- # キーワード #
 キーワードでは、モデルの区別および管理番号を指定します。
 管理番号は、1001 ～ 1999 までです。
 キーワードは、ローカルおよびマスターを合わせて 10000 まで指定できます。

5.2.2.3 ACADOTHERMDM.SET ファイルを作成する

他のユーザのローカルモデルの検索を行う場合に定義します。
 ACAD.SET の #MODEL:LOCALMG# で指定したディレクトリにファイル名
 ACADOTHERMDM.SET で作成します。

```
#OTHER:nnnn#      "host_name:directory_name"! .ext!
                  他のローカルモデルを検索するディレクトリを指定する。
                  OTHER:キーワード
                  nnnn          :   管理番号 (2001 ～ 2999)
                  directory_name:
                                他のユーザの ACADLOCALMDM.SET があるディレクトリ名
                  ext          :   ファイル拡張子 (. LMG)

#COMMENT#         "comment"
                  他のローカルモデルの注釈を記述する。
                  COMMENT:キーワード
                  comment       :   注釈を記述する
```

(例)

```
/ File Name : ACADOTHERMDM.SET
/
#OTHER:2001# "ctc1:D:/acad/files/user01/" !. LMG!
#COMMENT#   "U S E R 0 1 のモデル"
/
#OTHER:2002# "ctc2:D:/acad/files/user02/" !. LMG!
#COMMENT#   "U S E R 0 2 のモデル"
/
/ Eof
```

- 他のローカルモデル検索での制約
 - モデルの検索、表示、呼出しのみが使用できる。
 - 管理ファイルの読み込みは行なうが書き込みは行なわない。
 モデルを呼出し修正後同じディレクトリに格納しても管理ファイルには反映されない。
 - 他のユーザのローカル管理ディレクトリ及びモデルファイル格納ディレクトリは、ネットワークドライブの割り当てでマウントされている事を条件にする。
 - 他のユーザのローカル管理ファイル ACADLOCALMDM.SET の管理番号 1001 - 1900 までを検索対象とする。

管理番号の区分けは、次のようになります。

```
マスター      1 - 999
ローカル    1001 - 1999
その他      2001 - 2999
```

5.2.2.4 PLOTMDM.SET ファイルを作成する

『モデル作図』『作図ファイル』機能を使用する場合に作図出力するプロッタを定義します。
 ACAD.SET の #MODEL:MANAGE# で指定したディレクトリにファイル名 PLOTMDM.SET で作成します。

● PLOTMDM.SET のファイルフォーマット

- 1 カラム目が / : 注釈行
- 1 カラム目が空白 : プロッタのパラメータ
1つのパラメータは複数行にまたがってはいけない
- 1 カラム目が上記以外 : プロッタ名
- 1 行 : 1 2 8 バイト以内で記述

プロッタのパラメータ

HOST=xxx : プロッタが接続されている計算機のホスト名
ローカルプロッタの場合は不要

CMT="xxx" : プロッタの説明 (" で囲んで記入)

CMD=xxx : プロット出力実行モジュール名

PRM=xxx : プロット出力パラメータファイル名

OPT=xxx : プロット出力オプション

QUE="xxx" : プロット出力コマンド (" で囲んで記入)

WAIT : プロット出力完了待ちをする
プロット出力完了プロンプトを出力する

NOWAIT : プロット出力完了待ちをしない
プロット出力完了プロンプトを出力しない

PIPE="XXX" : パイプに続くコマンドを記述します。

プロット出力実行モジュールに引き渡すパラメータ

-s -p -r -w -m -sort -t

に付いては、プロッタマニュアルを参照。

例)

```
Local:a304
CMT="Canon A304 LBP ( A3 or A4 )" WAIT
CMD=C:/acad/exe/ap_LIPS
PRM=A304.PRM OPT= OPT="-jplotter_name"
-s0.7 -pC:/acad/uenv/ACAD.SET -r -m0.0 -w0.1 -sort -tTITLE
```

5.2.3 モデル管理ファイルを作成する

モデル、シンボルなどから情報を参照してモデル検索用のファイルを作成します。
モデル管理ファイルがないと、モデル検索コマンドは使えません。

管理ファイルの作成には、次の3種類の作成方法があります。

全部 : ACADMDM.SET と ACADLOCALMDM.SET で指定したディレクトリをもとに管理ファイルを作成します。
モデル検索機能を使用する前に一度は、必ずこの指定でモデル管理ファイルを作成する必要があります。

マスター: ACADMDM.SET で指定したディレクトリをもとに管理ファイルを作成します。

ローカル: ACADLOCALMDM.SET で指定したディレクトリをもとに管理ファイルを作成します。

『マスター』を実行すると ACAD.SET のキーワード #MODEL:MANAGE# で定義されたディレクトリに次のファイルが作成されます。

MODELFILE.MDM	: モデル (サブモデル) に関する情報。
MODELTITLE.MDM	: モデルタイトルに関する情報。
PAGETITLE.MDM	: ページタイトルに関する情報。
MODELDATE.MDM	: モデル最新呼出し日に関する情報。
SUBMDLINf.MDM	: モデルが参照しているサブモデルに関する情報。

SYMBOLINF. MDM	: モデルが参照しているシンボルに関する情報。
PARTINF. MDM	: モデルが参照している部品（関係アイテム）に関する情報。
REFINDEX. MDM	: 親モデルが参照している子供モデル名のインデックス。
REFMODEL. MDM	: 親モデルが参照している子供モデル名。
SYMBOLFILE. MDM	: シンボルに関する情報。
APGFILE. MDM	: APG に関する情報。
OLDMODEL. MDM	: 旧モデル（.OLD）に関する情報。
OLDSYMBOL. MDM	: 旧シンボル（.OLD）に関する情報。
OLDAPG. MDM	: 旧 APG（.OLD）に関する情報。
DISCARDMODEL. MDM	: 廃棄モデルに関する情報。
MDLNAMEyyyymmddhhmmss. MDM	: カセット内のモデル名の管理。yyyymmddhhmmss はこのファイルを作成したときの年月日時分秒
MODEL_LIST_TEMP. MDM	: モデル表示テンプレートファイル名の保存。
MODEL_DISPLAY_xxx. MDM	: モデル表示に関する情報の保存。
TITLEBIT. MDM	: 管理ファイル MODELTITLE. MDM の使用状況に関する情報。
PAGEBIT. MDM	: 管理ファイル PAGETITLE. MDM の使用状況に関する情報。
MODEL_SORT_KEY. MDM	: 一覧表示のソートキーに関する情報。

これらのファイルは一度作成されると、『モデルの保存』、『サブモデルの作成』、『シンボルの作成』、『APG 登録』を行うたびに自動的に更新されます。

『ローカル』を実行すると ACAD.SET のキーワード #MODEL:LOCALMG# で定義されたディレクトリに次のファイルが作成されます。

MODELFILE. LMG	: モデル（サブモデル）に関する情報。
MODELTITLE. LMG	: モデルタイトルに関する情報。
PAGETITLE. LMG	: ページタイトルに関する情報。
MODELDATE. LMG	: モデル最新呼出し日に関する情報。
SUBMDLINf. LMG	: モデルが参照しているサブモデルに関する情報。
SYMBOLINF. LMG	: モデルが参照しているシンボルに関する情報。
PARTINF. LMG	: モデルが参照している部品（関係アイテム）に関する情報。
REFINDEX. LMG	: モデルが参照している子供モデル名のインデックス。
REFMODEL. LMG	: モデルが参照している子供モデル名。
OLDMODEL. LMG	: 旧モデル（.OLD）に関する情報。
TITLEBIT. LMG	: 管理ファイル MODELTITLE. LMG の使用状況に関する情報。
PAGEBIT. LMG	: 管理ファイル PAGETITLE. LMG の使用状況に関する情報。

これらのファイルは、一度作成されると『モデルの保存』を行うたびに自動的に更新されます。

5.2.4 モデル情報ファイルを作成する

『親子関係』を検索したいときは、そのモデルを保存するときにモデル情報ファイル(.MDI)を作成してください。

『モデル』
『保存』を押す。(MODEL/WRITE)
『情報出力する』(INF)を押す。

モデル情報ファイルがないとコマンド『親子関係』で参照することはできません。
その他のコマンドの実行には差し支えありません。

モデル情報ファイルについては、コマンドリファレンス『モデルの保存』を参照してください。

シンボルリストを作成する。

『親子関係』でシンボルが参照されているモデルのピクチャ番号を表示するために使用します。

このファイルがない場合は、ピクチャ番号の表示は行いません。
ピクチャ番号の表示を行う場合はモデル情報ファイルとこのファイルを作成してください。

5.2.5 表示用テンプレートファイルの作成

サブウインドウ表示の場合のみ有効です。

5.2.5.1 モデル表示用を作成する

モデル検索で検索結果を画面に表示するフォーマットを指定するファイルの作成 (モデルの検索結果に対してのみ有効)

複数のテンプレートファイルを持つことができます。『テンプレ選択』でテンプレートファイルを選択することができます。テンプレートファイルが選択されなければ標準のフォーマットで表示します。またテンプレートファイル名 "CANCEL" が選択された場合に標準のフォーマットで表示します。

モデルダンプの一覧表示は、テンプレートファイル MODEL_DUMP_LIST.MDT が存在すれば、そのテンプレートにもとづいて表示します。ファイルが存在しない場合は、『モデル』の『テンプレ選択』で選択されたテンプレートにもとづいて表示します。

ファイルのフォーマット

(1) 制限値

最大桁数

MAXCOL col

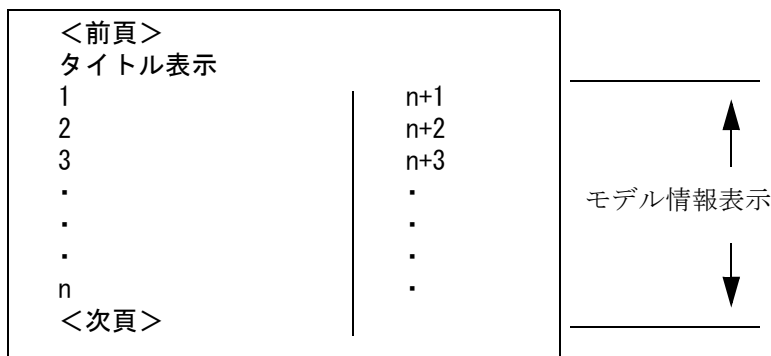
col : 1 行の最大桁数。(1 文字の間隔 = 画面の X サイズ / col) (最大 120)

画面分割数

SCREEN n

n : 画面の分割数。(1 ~ 4) 省略時は 1。

分割すると以下のように表示される。(例:2 分割の場合)



(2) タイトルの定義

タイトルは行の先頭から指定された行数分出力し、改ページ毎に出力する。

TITLE [l, c, col] "comment", v(m)

l : 行番号。(1 ~)
 c : 表示開始桁の位置。
 col : 色番号。(0 ~ 15) 省略時は 0。
 0 : 一時図形色。
 1 ~ 15 : 色番号 1 ~ 15。
 comment : 注釈文字列。(省略可)
 v(m) : 変数名。(省略可) 下記の中から選択。
 (m) は出力桁数を () で囲んで指定する。

v = DATE : 現在の日付を表示 (8 桁) YY/MM/DD
 TIME : 現在の時刻を表示 (8 桁) HH/MM/SS
 PAGE : ページ番号を表示 (4 桁)
 MAXPAGE : 最大ページ数を表示 (4 桁)
 COUNT : モデル数を表示 (4 桁)
 KEYMGNO : 検索対象になっている管理番号を表示 (1 ~ 60 桁)
 KEYMGNM : 検索対象になっている管理番号のグループ名を表示 (1 ~ 60 桁)
 (管理番号がグループ化されている場合のみ)
 KEYNAME : 検索対象になっているモデル名を表示 (1 ~ 80 桁)
 KEYCDATE : 検索対象になっている作成日を表示 (8 桁)
 KEYUPDATE : 検索対象になっている更新日を表示 (8 桁)
 KEYADATE : 検索対象になっている最新呼出し日を表示 (8 桁)
 KEYOWNER : 検索対象になっている所有者名を表示 (10 桁)
 KEYTTLnnn : 検索対象になっているモデルタイトルを表示 (1 ~ 80 桁)
 nnn はモデルタイトルの項目番号 (1 ~ 201)

(3) データの定義

検索結果のモデル情報の表示フォーマットを指定する。

DATA [l, c, col] "comment", v(m)

l : 行番号。(1 ~)
 c : 表示開始桁の位置。
 col : 色番号。(0 ~ 15) 省略時は 0。
 0 : 一時図形色。
 1 ~ 15 : 色番号 1 ~ 15。

comment: 注釈文字列。(省略可)

v(m) : 変数名。(省略可) 下記の中から選択。

(m) は出力桁数を () で囲んで指定する。

```

v = SEQNO      : 順序番号 (5 桁)
  SUBMODEL     : サブモデルフラグ (1 桁)
  MDLNAME      : モデル名 (32 桁)
  MDLPATH      : モデルパス名 (32 桁)
  MGPATH       : フルパスモデル名 (32 桁)
  READ         : 読み込み属性 (1 桁)
  WRITE        : 書き込み属性 (1 桁)
  MGNO         : 管理番号 (5 桁)
  CDATE        : 作成日 (8 桁)
  CDATE (10)   : 作成日 (10 桁) YYYY/MM/DD
  UDATE        : 更新日 (8 桁)
  UDATE (10)   : 更新日 (10 桁) YYYY/MM/DD
  ADATE        : 呼出し日 (8 桁)
  ADATE (10)   : 呼出し日 (10 桁) YYYY/MM/DD
  OWNER        : 所有者名設定の有無 (1 桁)
  OLD          : 旧モデルの有無 (1 桁)
  CURRENT      : 現モデルの有無 (1 桁)
  SIZE         : モデルのサイズ (バイト長) (8 桁)
  RASTER       : ラスターファイルの有無 (1 桁)
  CHILD        : 子供定義の有無 (1 桁)
  TTLnnn       : モデルタイトル (1 ~ 256 桁)
                  nnn はモデルタイトル項目番号
                  (1 ~ 201)

```

例 1

```

/ 検索表示用テンプレートファイル (1)
/ Define Limit
MAXCOL      90
/
/ Define List Title
TITLE [ 1,  1] "モデル名一覧"
TITLE [ 1, 71] "ページ", PAGE (4)
TITLE [ 1, 79] "/"
TITLE [ 1, 80] MAXPAGE (4)
/
TITLE [ 2,  5] "検索条件"
TITLE [ 2, 71] "個数", COUNT (4)
/
TITLE [ 3, 11] "管理番号" : ", KEYMGNO (40)
TITLE [ 4, 11] "名前"      : ", KEYNAME (32)
TITLE [ 5, 11] "作成日"    : ", KEYCDATE (20)
TITLE [ 6, 11] "更新日"    : ", KEYUDATE (20)
TITLE [ 7, 11] "所有者名"  : ", KEYOWNER (10)
TITLE [ 8, 11] "主タイトル" : ", KEYTTL201 (40)
/
TITLE [ 9,  1]
/
TITLE [10,  7] "モデル名"
TITLE [10, 41] "読書"
TITLE [10, 45] "管理番号"
TITLE [10, 51] "作成日"
TITLE [10, 59] "更新日"
TITLE [10, 68] "所有者名"
TITLE [10, 74] "旧"
TITLE [10, 78] "カセット"
TITLE [11,  1]

```

```

/
/ Define List Model Item
DATA [ 1, 1] SEQNO (5)
DATA [ 1, 7] SUBMODEL
DATA [ 1, 9] MDLNAME (30)
DATA [ 1, 41] READ
DATA [ 1, 42] WRITE
DATA [ 1, 45] MGNO (5)
DATA [ 1, 51] CDATE
DATA [ 1, 59] UDATE
DATA [ 1, 70] OWNER
DATA [ 1, 74] OLD
/
DATA [ 2, 11] TTL201 (30)
DATA [ 2, 41] TTL001 (20)
DATA [ 2, 61] TTL002 (20)
/

```

例 2

```

/ 検索表示用テンプレートファイル (2)
/ Define Limit
MAXCOL 90
/
/ Define List Title
TITLE [ 1, 1] "モデル名一覧"
TITLE [ 1, 71] "ページ ", PAGE (4)
TITLE [ 1, 79] "/"
TITLE [ 1, 80] MAXPAGE (4)
/
TITLE [ 2, 5] "検索条件"
TITLE [ 2, 71] "個数 ", COUNT (4)
/
TITLE [ 2, 18] "管理名" : ", KEYMGNM (60)
TITLE [ 3, 18] "名前" : ", KEYNAME (32)
TITLE [ 4, 18] "作成日" : ", KEYCDATE (20)
TITLE [ 5, 18] "更新日" : ", KEYUPDATE (20)
TITLE [ 6, 18] "所有者名" : ", KEYOWNER (10)
TITLE [ 7, 18] "主タイトル" : ", KEYTTL201 (40)
/
TITLE [ 8, 1] "モデル名"
TITLE [ 8, 22] "主タイトル"
TITLE [ 8, 54] "読"
TITLE [ 8, 56] "書"
TITLE [ 8, 58] "管理番号"
TITLE [ 8, 65] "作成日"
TITLE [ 8, 72] "更新日"
TITLE [ 8, 78] "旧"
TITLE [ 8, 80] "カセット"
/
/ Define List Model Item
DATA [ 1, 1] SUBMODEL
DATA [ 1, 2] MDLNAME (20)
DATA [ 1, 22] TTL201 (30)
DATA [ 1, 54] READ
DATA [ 1, 56] WRITE
DATA [ 1, 59] MGNO (5)
DATA [ 1, 64] CDATE
DATA [ 1, 71] UDATE
DATA [ 1, 78] OLD
/

```

5.2.5.2 シンボル表示用を作成する

モデル検索で検索結果を画面に表示するフォーマットを指定するファイルの作成 (シンボルの検索結果に対してのみ有効)

ファイル名 MDLINF_SYMBOL_LIST.MDT で作成してください。このファイルがない場合は、標準のフォーマットで表示します。

ファイルのフォーマット

- (1) 制限値
最大桁数

MAXCOL col

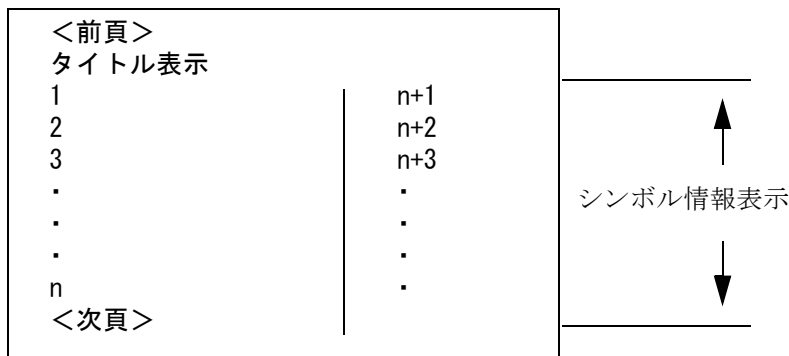
col : 1 行の最大桁数。(1 文字の間隔 = 画面の X サイズ / col) (最大 120)

画面分割数

SCREEN n

n : 画面の分割数。(1 ~ 4) 省略時は 1。

分割すると以下のように表示される。(例: 2 分割の場合)



- (2) タイトルの定義
タイトルは行の先頭から指定された行数分出力し、改ページ毎に出力する。

TITLE [l, c, col] "comment", v(m)

- l : 行番号。(1 ~)
c : 表示開始桁の位置。
col : 色番号。(0 ~ 15) 省略時は 0。
0 : 一時図形色。
1 ~ 15 : 色番号 1 ~ 15。
comment: 注釈文字列。(省略可)
v(m) : 変数名。(省略可) 下記の中から選択。
(m) は出力桁数を () で囲んで指定する。
- | | |
|----------|---------------------------------|
| v = DATE | : 現在の日付を表示 (8 桁) YY/MM/DD |
| TIME | : 現在の時刻を表示 (8 桁) HH/MM/SS |
| PAGE | : ページ番号を表示 (4 桁) |
| MAXPAGE | : 最大ページ数を表示 (4 桁) |
| COUNT | : シンボル数を表示 (4 桁) |
| KEYMGNO | : 検索対象になっている管理番号を表示 (1 ~ 60 桁) |
| KEYNAME | : 検索対象になっているシンボル名を表示 (1 ~ 80 桁) |
| KEYCDATE | : 検索対象になっている作成日を表示 (8 桁) |

KEYUPDATE : 検索対象になっている更新日を表示 (8 桁)
 KEYOWNER : 検索対象になっている所有者名を表示 (10 桁)

(3) データの定義

検索結果のシンボル情報の表示フォーマットを指定する。

```
DATA [l, c, col] "comment", v (m)
```

l : 行番号。(1 ~)
 c : 表示開始桁の位置。
 col : 色番号。(0 ~ 15) 省略時は 0。
 0 : 一時図形色。
 1 ~ 15 : 色番号 1 ~ 15。
 comment : 注釈文字列。(省略可)
 v (m) : 変数名。(省略可) 下記の中から選択。
 (m) は出力桁数を () で囲んで指定する。

v = SEQNO	:	序番号 (5 桁)
NAME	:	シンボル名 (32 桁)
PATH	:	シンボルパス名 (32 桁)
READ	:	読込み属性 (1 桁)
WRITE	:	書込み属性 (1 桁)
MGNO	:	管理番号 (5 桁)
CDATE	:	作成日 (8 桁)
DATE (10)	:	" (10 桁) YYYY/MM/DD
UPDATE	:	更新日 (8 桁)
UPDATE (10)	:	" (10 桁) YYYY/MM/DD
OWNER	:	所有者名設定の有無 (1 桁)
OLD	:	旧シンボルの有無 (1 桁)
CURRENT	:	現モデルの有無 (1 桁)

5.2.6 管理番号のグループを指定するファイルを作成する (ACADM.DM.MGN)

モデルの管理番号をグループ化し名前を付ける。
 グループ化することによりモデル検索時に、管理番号をグループごとに表示して検索を簡潔にすることができる。

```
group_name = mg_no , .....
```

group_name : グループ名。(1 ~ 20 バイト, 半角で 20 文字, 全角で 10 文字)
 mg_no : 管理番号。(ACADM.DM.SET, ACADLOCALMDM.SET に定義されていること)

例)

```
/File Name : ACADM.DM.MGN
/
/Grouping Model Mangement Number
/
/Form : Group_name = Management#1 [.....]
/
/
共用__1      =    1,   2,   3
共用__2      =    4,   5,   6
共用__3      =    7,   8,   9
プライベート = 1001
モデル情報   =  901, 902
/
```

接続行

mg_no ,

管理番号が 1 行で納まらない場合に、1 行目を空白にして残りの管理番号を定義する。

5.2.7 管理番号の階層構造を指定するファイルを作成する (ACADMDM.GRP)

モデルの管理番号を階層構造化する。
階層構造化することによりモデル検索時に、管理番号を階層表示して検索を簡潔にすることができる。

<GroupDefinition>	: 定義の始まり
</GroupDefinition>	: 定義の終わり
<Group name="group_name">	: グループの始まり
</Group>	: グループの終わり
<ManagementNo no=mg_no/>	: グループに属する管理番号を指定

group_name : グループ名。(1 ~ 20 バイト, 半角で 20 文字, 全角で 10 文字)
mg_no : 管理番号。(ACADMDM.SET, ACADLOCALMDM.SET に定義されていること)

注) グループ内にグループがあってもいいですがグループと管理番号の混在は許されません。

例)

```

/File Name : ACADMDM.GRP
/
<GroupDefinition>
  <Group name="グループ 1">
    <ManagementNo no=1/>
    <ManagementNo no=2/>
    <ManagementNo no=3/>
  </Group>
  <Group name="グループ 2">
    <Group name="グループ 2-1">
      <ManagementNo no=211/>
      <ManagementNo no=212/>
      <ManagementNo no=213/>
    </Group>
    <Group name="グループ 2-2">
      <ManagementNo no=221/>
      <ManagementNo no=222/>
      <ManagementNo no=224/>
    </Group>
  </Group>
</GroupDefinition>
/
/

```

5.2.8 リスト出力テンプレートファイルの作成

コマンド『リスト出力』(LPR)で出力されるファイルはテンプレートファイルで定義されたフォーマットで出力されます。

定義がない場合は標準のフォーマットで出力されます。または、コマンド『テンプレート』でテンプレートファイルが選択できます。

テンプレートファイル名の頭が CSV_ で始まるテンプレートファイルでのリスト出力は CSV 形式での出力になります。出力ファイルのファイル拡張子は .CSV になります。

テンプレートのキーワード "FORM" "MAXCOL" 及びキーワード "TITLE" "DATA" 内の項目 [l,c] は無視します。これらのキーワード及び項目は、なくても構いません。

モデルダンプのテンプレートファイルは、MODEL_DUMP_LIST.MTP で作成してください。ファイルが存在しない場合は、標準のフォーマットで出力します。

(1) フォームの指定

```
FORM form
```

```
form = TATE      : 縦方向  字の大きさ  普通
      tate       :      "      "      小
      YOKO       : 横方向      "      普通
      yoko       :      "      "      小
```

(2) 制限値

(a) 最大行数 (指定がなければ 55 行)

```
MAXLIN line
```

line = 1 ページの行数を指定します。

行数 = タイトル行数 + データ行数

指定された行数になったとき改ページが行われる。

(b) 最大桁数

```
MAXCOL col
```

col = 1 行の最大桁数。(最大 200 桁)

(3) タイトル 1 の定義

タイトルは行の先頭から指定された行数分出力し、改ページごとに出力する。

```
TITLE [l,c] "comment", v (m)
```

```
l      : 行番号。(1 ~ )
c      : 開始桁の位置。
comment : 注釈文字は " " で囲む。(省略可)
v (m)  : 変数名。下記の 10 種類から選択する。
        (m) は出力桁数を () で囲んで指定する。省略された場合は下記 () 内の既定値を使用する。
          v = DATE: 現在の日付を出力 (8 桁)  YY/MM/DD
              TIME  : 現在の時刻を出力 (8 桁)  HH:MM:SS
              PAGE  : ページを出力 (4 桁)
              MAXPAGE : 最大ページ数を表示 (4 桁)
              COUNT  : モデルの数を出力 (4 桁)
              KEYMGNO : 検索対象になっている管理番号を出力 (1 ~ 60 桁)
              KEYNEME : 検索対象になっている名前を出力 (1 ~ 80 桁)
              KEYCDATE : 検索対象になっている作成日を出力 (8 桁)
              KEYUDATE : 検索対象になっている更新日を出力 (8 桁)
              KEYADATE : 検索対象になっている呼出し日を出力 (8 桁)
              KEYOWNER : 検索対象になっている所有者名を出力 (10 桁)
```

KEYTTLnnn : " モデルタイトルを出力 (1 ~ 80 桁)
nnn は項目番号 (1 ~ 201)

(4) データの定義

検索結果をこのフォーマットに置き換えて出力される。
タイトル 1 が出力された後、データが出力される。

DATA [l, c] "comment", v (m)

l : 行番号。(1 ~)
c : 開始桁の位置。
comment : 注釈文字は " " で囲む。(省略可)
v (m) : 変数名。下記の 14 種類から選択する。
(m) は出力桁数を () で囲んで指定する。省略された場合は下記 () 内の既定値を使用する。

v = SEQNO:順序番号 (5 桁)

SUBMODEL	: サブモデルフラグ (1 桁)
MDLNEME	: モデル名 (32 桁)
MDLPATH	: モデルパス名 (32 桁)
MGPATH	: フルパスモデル名 (32 桁)
READ	: 読み込み属性 (1 桁)
WRITE	: 書き込み属性 (1 桁)
MGNO	: 管理番号 (4 桁)
CDATE	: 作成日 (8 桁)
CDATE (10)	: 作成日 (10 桁) YYYY/MM/DD
UPDATE	: 更新日 (8 桁)
UPDATE (10)	: 更新日 (10 桁) YYYY/MM/DD
ADATE	: 呼出し日 (8 桁)
ADATE (10)	: 呼出し日 (10 桁) YYYY/MM/DD
OWNER	: 所有者名設定の有無 (1 桁)
OLD	: 旧モデルの有無 (1 桁)
CURRENT	: 現モデルの有無 (1 桁)
SIZE	: モデルのサイズ (バイト長) (8 桁)
CHILD	: 子供定義の有無 (1 桁)
TTLnnn	: モデルタイトル (1 ~ 256 桁)

nnn は項目番号 (1 ~ 201)

(5) タイトル 2 の定義

データが出力された後に出力するタイトルです。(省略可)

TRAIL [l, c] "comment", v (m)

内容は (3) タイトル 1 の定義と同様

例) 検索結果リスト出力用テンプレートファイル

```

/
/ Define Limit
MAXLIN 55
MAXCOL 80
/
/ Define List Title
TITLE [ 1, 1] "Model Name List"
TITLE [ 1, 31] "Date ", DATE (8)
TITLE [ 1, 51] "Time ", TIME (8)
TITLE [ 1, 71] "Page ", PAGE (4)
/
TITLE [ 2, 1]
/
TITLE [ 3, 11] "Manegement Number : ", KEYMGNO (40)
TITLE [ 4, 11] "Name : ", KEYNAME (32)
TITLE [ 5, 11] "Creation Date : ", KEYCDATE (20)

```

```

TITLE [ 6, 11] "Updating Date      : ", KEYUPDATE (20)
TITLE [ 7, 11] "Owner Name        : ", KEYOWNER (10)
TITLE [ 8, 11] "Main Title         : ", KEYTTL201 (40)
/
TITLE [ 9,  1]
/
TITLE [10,  7] "  Model Name"
TITLE [10, 41] "R/W"
TITLE [10, 45] "No."
TITLE [10, 50] "Creation"
TITLE [10, 59] "Updating"
TITLE [10, 68] "Owner"
TITLE [10, 74] "Old"
/
TITLE [11, 48] "    Date"
TITLE [11, 58] "    Date"
/
/ Define List Model Item
DATA [ 1,  1] SEQNO (5)
DATA [ 1,  7] SUBMODEL
DATA [ 1,  9] MDLNAME (30)
DATA [ 1, 41] READ
DATA [ 1, 43] WRITE
DATA [ 1, 45] MGNO (4)
DATA [ 1, 50] CDATE
DATA [ 1, 59] UDATE
DATA [ 1, 70] OWNER
DATA [ 1, 75] OLD
/
DATA [ 2, 11] TTL201 (30)
DATA [ 2, 41] TTL001 (20)
DATA [ 2, 61] TTL002 (20)
/

```

5.2.9 ページタイトルテンプレートファイルの作成

ページタイトルで検索する時の項目（各ページ共通）を設定します。
このテンプレートファイルがない場合は、全項目共通の検索を行います。

次の書式で項目を設定。

(item_no) "comment"

item_no : 項目の番号。(1～255)
comment : 項目の説明。

この書式にあわない行は、すべて注釈とみなします。

例)

```

/ File name : PAGETITLE.TMP
/
/           Sample Page Title Template for Model Kensaku
/
(  1) " 項目 1  "
(  2) " 項目 2  "
(  3) " 項目 3  "
(  4) " 項目 4  "
/
/ Eof

```


5.3 コマンド

5.3.1 モデルの検索

● コマンド一覧

モデル検索メニュー

メニュー表示	コマンド	機能
モデル検索	MDLDISP	検索条件の設定・変更メニューを表示する。
検索条件		
設定・変更		
表示		
情報表示	INFDISP	モデルの情報を表示する。
モデル表示	MDLDRAW	モデル表示するモデル名を選択する。
表示 G O	DRAWGO	モデル表示を開始する。
モデル読込		
読込	MDLREAD	指示したモデルを読み込む。
変更		
変更		属性変更メニューを表示する。
モデル削除		
削除	MDLDELETE	指示したモデルを削除する。
全部	ALL	全モデルを削除する。
ソートキー		
設定・変更		ソートキーの設定・変更メニューを表示する。
表示制御		
設定・変更		表示制御の設定・変更メニューを表示する。
リスト出力		
リスト出力	LPR	検索結果をプリンタに出力する。

検索条件メニュー

メニュー表示	コマンド	機能
モデル検索	MDLDISP	

メニュー表示	コマンド	機能
検索条件		
管理番号	MDLMGNO	モデルの管理番号を指定して検索する。
サイズ	MDLSIZE	管理番号別にモデルの総容量を表示する。
名前	MDLNAME	モデル名を指定して検索する。
ファイル	MDLFILE	モデル名リストファイルのモデル名で検索する。
作成日	MDLDATE	モデルの作成日を指定して検索する。
更新日	MDLUPDATE	モデルの更新日を指定して検索する。
呼出し日	MDLACDATE	モデルの最新呼出し日を指定して検索する。
所有者	MDLOWNER	モデルの所有者名を指定して検索する。
タイトル	MDLTITLE	モデルタイトル検索メニューを表示する。
頁タイトル	PAGETITLE	頁タイトルを指定してモデルを検索する。
現 なし	MDLCUR	現モデルのないモデルを検索する。
旧 あり	MDLOLD	旧モデルがあるモデルを検索する。
ダンプ済	MDLCST	ダンプ済モデルを検索する。
前回条件	USELAST	前回設定した検索条件を使用する
検索		
絞込 GO	CONTINUE	検索の絞り込み処理を開始する。
次頁		モデル検索メニューを表示する。

変更メニュー

メニュー表示	コマンド	機能
モデル検索		
変更		
名前	CHGMGNO	管理番号及びモデル名を変更する。
複製	DUPONOFF	変更前のモデルを残す。
所有者	CHGOWNER	所有者名を変更する。
情報変更		
情報変更	INFUPDATE	モデルタイトルを変更する。
保護		
書込み	WRITEONOFF	モデルへの書込みスイッチ。

メニュー表示	コマンド	機能
検索条件		
設定・変更		検索条件の設定・変更メニューを表示する。
前頁		モデル検索メニューを表示する。

表示制御メニュー

メニュー表示	コマンド	機能
モデル検索		
表示制御		
テンプレ選択	MDLTMP	テンプレートファイルを指定する。
絞込 G O	CONTINUE	検索の絞り込み処理を開始する。
モデル表示		
カラー表示	ACTPLN	カラーでモデルを表示する。
単色表示	TMPPLN	単色でモデルを表示する。
表示 P I C	DSPPIC	モデルの表示ピクチャおよび画面の分割数を指定する。
全ピクチャ	ALLPIC	モデルの全ピクチャを表示する。
S L O 合せ	CURSLO	現在の S L O に合わせてモデルを表示する。
ラスタ表示		
ラスタ表示	RASDISP	ラスタイメージを表示する。
検索条件		
設定・変更		検索条件の設定・変更メニューを表示する。
前頁		モデル検索メニューを表示する。

ソートキーメニュー

メニュー表示	コマンド	機能
モデル検索		
ソートキー		
名前	SORTNAME	モデル名をアルファベット順で表示する。
作成日	SORTDATE	モデル名を作成日順に表示する。
更新日	SORTUPDATE	モデル名を更新日順に表示する。
呼出し日	SORTACDATE	モデル名を最新呼出し日順に表示する。

メニュー表示	コマンド	機能
タイトル	SORTTITLE	モデルタイトルで並べて表示する。
昇順／降順	SORTAS_DES	検索名の順序を昇順または降順に並べ換えるスイッチ。
検索条件 設定・変更		検索条件の設定・変更メニューを表示する。
前頁		モデル検索メニューを表示する。

読込メニュー

メニュー表示	コマンド	機能
モデル検索 読込み		指示したモデルを読み込む。
新規モード	NEW	アクティブモデルを放棄し、読み込む。(省略時)
追加モード	ADD	アクティブモデルに追加読み込みする。
読込 P I C	RDPIC	指定されたピクチャを読み込む。省略すると全ピクチャを読み込む。
格納先 P I C	OUTPIC	格納先ピクチャを指定する。指定したピクチャに格納する。省略すると各々のピクチャに格納する。
選択マスク	MSKONN	現在のアイテム、クラス選択マスクで選択可能なアイテムだけを読み込む。
選択マスク	MSKOFF	選択マスクを参照しない。(省略時)
前頁		モデル検索メニューを表示する。

モデルタイトル検索メニュー

メニュー表示	コマンド	機能
モデル検索 項目別	TTLEACH	各項目別に検索条件を設定する。
全項目共通	TTLALL	全項目に共通な検索条件を設定する。
設定	TTLSET	検索条件を設定する。
文章呼出	TLB	テキストファイルを呼出し検索条件を設定する。
取消	TTLCAN	検索条件を取り消す。
検索 G O	TTLGO	検索処理を開始する。

メニュー表示	コマンド	機能
絞込 G O	TTLCONTINUE	検索の絞り込み処理を開始する。

5.3.1.1 モデルを検索する

【メニユ一】

〔モデル検索〕→〔モデル〕

【構文】

MDLDISP	{ { D_ONN } { D_OFF } { ONN } { OFF } { SORTNAME { SORTDATE SORTUPDATE SORTACDATE } { sortword } } }	[SORTAS_DES]	{ mgnword MDLNAME name MDLFILE name MDLDATE date MDLUPDATE date MDLAGDATE date MDLOWNER owner MDLCSTNAME name MDLCUR MDLOLD MDLCST ttlword ptlword USELAST }	* }

[MDLTMP name] { <CE>
CONTINUE }

- ※ sortword はタイトルソートキーを参照。
- ※ mgnword は管理番号検索を参照。
- ※ ttlword はタイトル検索を参照。
- ※ ptlword は頁タイトル検索を参照。

検索結果を画面に一覧表示するかどうかを指示する。

D_ONN : 一覧表示をする。(省略時)
D_OFF : 一覧表示をしない。

検索結果をファイルに出力するかどうかを指示する。

ONN : 検索結果を出力する。
OFF : 検索結果を出力しない。(省略時)

検索結果の一覧をつぎのいずれかの方法で並べて表示する。

SORTNAME	:	モデル名で並べて表示する。(省略時)
SORTDETE	:	モデルの作成日で並べて表示する。
SORTUPDETE	:	モデルの更新日で並べて表示する。
SORTACDETE	:	モデルの呼出し日で並べて表示する。

検索結果の一覧をつぎの方法で並べて表示する。

sortword : モデルのタイトルで並べて表示する。ソートキーとして指定できるモデルタイトルは最高 5 項目で、使用できる文字数はモデルタイトル 1 項目につき半角 20 文字まで。構文は以下のとおり。

【sortword の部分の構文】

SORTTITLE	$\left[\left\{ \begin{array}{l} d \\ \text{key-no} \end{array} \right\} \text{ttl-no} \right]^*$
-----------	---

ソート対象となるタイトルのキー番号をつぎのいずれかの方法で指示する。

d : ソート対象となるタイトルのキー番号をピックする。
key-no : ソート対象となるタイトルのキー番号 (1 ~ 5) を入力する。

ソート対象となるタイトル項目番号を指示する。

ttl-no : ソート対象となるモデルタイトル項目番号 (1 ~ 201) を入力する。
"0" を入力すると取消になる。

検索結果の一覧表示の並びを昇順または降順に指示する。

SORTAS_DES : 昇順を降順にする。もう一度入力すると降順を昇順にする。省略時は昇順に並べる。

検索対象をつぎのいずれかの方法で指示する。

mgnoword : モデルを管理番号で検索するとき指示する。省略時は全管理番号が検索対象になる。管理番号は複数指示できる。

【mgnoword の部分の構文】

MDLMGNO [MDLSIZE] no

管理番号別にモデルのサイズを表示したいときに指示する。

MDLSIZE : 管理番号別にモデルの総容量を表示する。

モデルの管理番号を指示する。

no : 管理番号を入力する。

MDLNAME name : モデル名を入力する。省略時は全モデル名が検索対象になる。モデル名にはワイルドカード文字 * ? と || [] (OR) ・ <> (～以外) が使用できる。

例)

A* : A で始まるモデル名。

A*||B* : A または B で始まるモデル名。

[AB]* : A または B で始まるモデル名。(同上)

A*||B*<> : A および B 以外の文字で始まるモデル名。

[AB]*<> : A および B 以外の文字で始まるモデル名。(同上)

"<>" は文字列の先頭か最後に入力する。"<>" の指定がある場合 "||" は AND として働く。

MDLFILE name : モデル名が登録されたリストファイルのファイル名を入力する。リストファイルに載っているモデル名が検索対象になる。
この検索条件が選択されると他の検索条件は無視され、実行後この検索条件は解除される。

リストファイルで指定できるモデル名のは最高 8000 まで。

リストファイルの内容は、モデル名またはモデルパス名で登録すること。

MDLDATE date : モデル作成日を入力する。日付 date は yyyy/mm/dd と入力する。

例) 1994/01/01

MDLUPDATE date : モデル更新日を入力する。日付 date は yyyy/mm/dd と入力する。

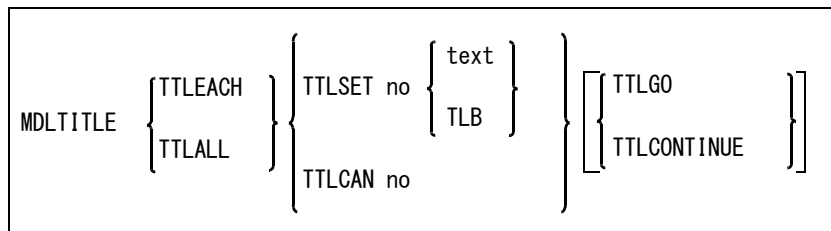
例) 1994/01/01

MDLACDATE date : モデル最新呼出し日を入力する。日付 date は yyyy/mm/dd と入力する。例) 1994/01/01

MDLOWNER owner : モデルの所有者名を入力する。

MDLCUR	: 現モデルが存在しないモデルを検索対象とする。
MDLOLD	: 旧モデルが存在するモデルを検索対象とする。
ttlword	: モデルをタイトルで検索するとき指示する。モデルタイトルに記述されたメニューが画面に表示される。
ptlword	: モデルをページタイトルで検索するとき指示する。
USELAST	: 最後に検索を行ったときの検索条件を使用する。

【ttlword の部分の構文】



検索対象になるモデルタイトルを指示する。

TTL EACH	: 各項目別に検索条件を指示する。
TTL ALL	: 全項目に共通な検索条件を指示する。

検索条件の項目を指示する。

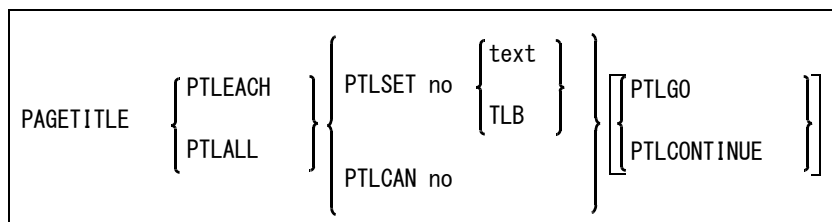
TTL SET no text	: 検索対象のモデルタイトル項目番号と検索文字列を入力する。複数の項目を指示できる。検索文字列を省略すると全文字列が対象になる。検索文字列にはワイルドカード文字 * ? と OR () ・ NIL ・ -NIL 、大小比較 < <= > >=) が使用できる。
(1)	OR () は "A B" で A または B を検索する。
(2)	NIL はモデルタイトルの項目に設定のないものを検索する。
(3)	-NIL はモデルタイトルの項目に何かの設定があるものを検索する。
(4)	< , <= , > , >= はモデルタイトルを数字だけで検索する。数値変換できない文字列が含まれている場合は検索の対象外となる。
	1, +1, 1.0, -1.0 などは数値とみなす。
例)	<= 100 : 100 に等しいか、100 より小さい。
	> 1 < 100 : 1 より大きく 100 より小さい。

TTL SET no TLB	: テンプレートファイルから検索文字列を選択し入力する。複数の項目を指示できる。
TTL CAN no	: TTL SET で指示したモデルタイトル項目番号を取り消す。

つぎのいずれかの方法で検索を開始する。

TTL GO	: 検索を開始する。
TTL CONTINUE	: 絞り込み検索を開始する。

【ptlword の部分の構文】



検索対象になるページタイトルを指示する。

TTL EACH	: 各項目別に検索条件を指示する。
TTL ALL	: 全項目に共通な検索条件を指示する。

検索条件の項目を指示する。

PTL SET no text	: 検索対象のページタイトル項目番号と検索文字列を入力する。複数の項目を指示できる。検索文字列を省略すると全文字列が対象になる。検索文字列にはワイルドカード文字 * ? と OR () が使用できる。
PTL SET no TLB	: テンプレートファイルから検索文字列を選択し入力する。複数の項目を指示できる。
PTL CAN no	: PTL SET で指示したページタイトル項目番号を取り消す。

つぎのいずれかの方法で検索を開始する。

PTLGO : 検索を開始する。
PTLCONTINUE : 絞り込み検索を開始する。

表示形式を規定するテンプレートファイルを指示する。

MDLTMP name : ファイル名を入力する。省略すると標準の形式を用いる。テンプレートファイルは、ACAD.SET のキーワード #MDLINP:DISP# で定義したディレクトリになければならない。つぎに変更されるまで表示フォーマットは有効。ただし、“CANCEL.MDT”を選択すると指示されていたテンプレートファイルのフォーマットがキャンセルされ、標準のテンプレートファイルのフォーマットが用いられる。

つぎのいずれかの方法で検索を開始する。

<CE> : 検索を開始する。
CONTINUE : 絞り込み検索を開始する。

5.3.1.2 検索結果のモデルファイル情報を表示する

【メニュー】

〔モデル検索〕 → 〔モデル〕 → 〔情報表示〕

【構文】

INFDISP	$\left\{ \begin{array}{l} \text{filename} \\ d \\ s \end{array} \right\}$
---------	---

モデルの情報はシステムレジスタに保存される。

モデル名をつぎのいずれかで指示する。

filename : モデル名を入力する。
d : モデル一覧表のモデル名をピックアップする。
s : モデル一覧表の番号を入力する。

5.3.1.3 モデルのページタイトル情報を表示する

【メニュー】

〔モデル検索〕 → 〔モデル〕 → 〔頁タイトル〕

【構文】

PTLDISP	$\left\{ \begin{array}{l} \text{filename} \\ d \\ s \end{array} \right\}$
---------	---

モデル名をつぎのいずれかで指示する。

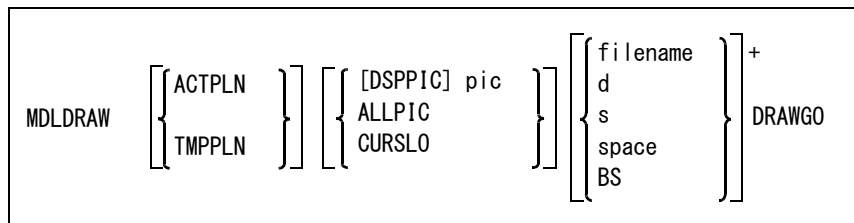
filename : モデル名を入力する。
d : モデル一覧表のモデル名をピックアップする。
s : モデル一覧表の番号を入力する。

5.3.1.4 検索後のモデル内のピクチャを画面に表示する

【メニュー】

〔モデル検索〕→〔モデル〕→〔モデル表示〕

【構文】



表示形式を指示する。設定はつぎに変更されるまで有効。

表示色をつぎのいずれかで選択する。

ACTPLN : モデルをカラーで表示する。アクティブモデルは一時的に表示をやめる。

TMPPLN : テンポラリ図形の表示色で画面に重ね書きする。

表示するピクチャをつぎのいずれかで指示する。

DSPPIC pic : ピクチャ番号を入力する。2通りの指定方法がある。

(1) ピクチャ番号を指定する。画面の分割数は変更しない。

(2) ピクチャ番号と画面の X 方向の分割数 (1 ~ 8) と文字 "X"、Y 方向の分割数 (1 ~ 8) で指定する。画面の分割数は次に変更されるまで有効。

例 "1,2X4" (表示ピクチャは "1"、画面の X 方向分割数は "2"、画面の Y 方向分割数は "4" になる。)

ALLPIC : モデル中のすべてのピクチャを表示する。画面をピクチャ数で自動的に分割しそれぞれのピクチャを表示する。

CURSLO : 現在のスクリーンレイアウトに合わせてピクチャを表示する。ただし複数のビューポートに同じピクチャを表示している場合は最大面積のビューポートにだけ表示する。

モデルをつぎのいずれかの方法で選択する。

filename : モデル名を入力する

d : モデル一覧表のモデル名をピックする

s : モデル一覧表の番号を入力する

モデルを選択するために表示を切り替えるとき指示する。

space : 次ページを表示する。

BS : 前のページを表示する。

モデルの表示を開始する。

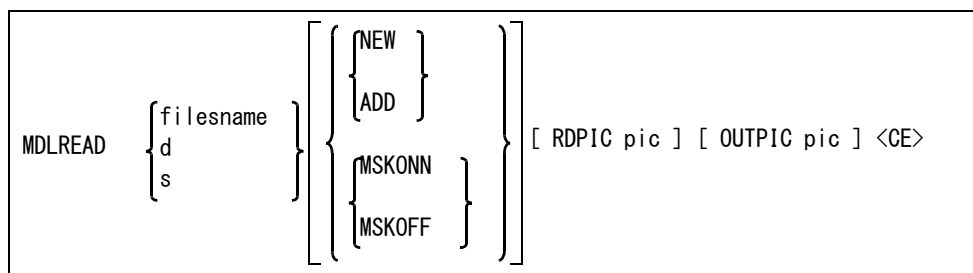
DRAWGO : モデルの表示を開始する。

5.3.1.5 モデルを読み込む

【メニュー】

〔モデル検索〕→〔モデル〕→〔読込〕

【構文】



モデル名をつぎのいずれかで指示する。

filename : モデル名を入力する。
 d : モデル一覧表のモデル名をピックアップする。
 s : モデル一覧表の番号を入力する。

注) 指定したモデルにロックファイルが作成されている場合、ワーニングメッセージが表示されます。

追加読み込みするかどうかを指示する。

NEW : モデルを新規に読み込む。(省略時)
 ADD : モデルを追加読み込みする。

選択マスクを参照するかどうかを指示する。

MSKONN : 現在のアイテム・クラス・レビジョン選択マスクで選択可能な有効アイテムだけを読み込む。
 MSKOFF : 選択マスクを参照しないで読み込む。(省略時)

読み込みピクチャを指定する。

RDPIC pic : 読み込むピクチャの番号を入力する。省略するか0を入力すると、モデルの全ピクチャを読み込む。

格納先ピクチャを指定する。

OUTPIC pic : 格納先ピクチャを番号で入力する。省略時はそれぞれのピクチャに格納する。

コマンドを実行する。

<CE> : モデルを読み込む。

5.3.1.6 モデルの管理番号及び名前を変更する

【メニュー】

〔モデル検索〕→〔モデル〕→〔変更〕→〔名前〕

【構文】

CHGMGNO	$\left\{ \begin{array}{l} d \\ no \end{array} \right\}$	[DUPONOFF]	$\left\{ \begin{array}{l} filename \\ d \\ s \end{array} \right\}$	[owner]	new_name <CE>
---------	---	--------------	--	-----------	---------------

新しいモデル管理番号を指示する。

d : モデルの管理番号をピックアップする。
 no : モデルの管理番号を入力する。

変更前のモデルを残すか残さないかを指示する。

DUPONOFF : 変更前モデルを保存する。省略時は変更前モデルを残さない。

管理番号を変更したいモデル名をつぎのいずれかで指示する。

filename : モデル名を入力する。
 d : モデル一覧表のモデル名をピックアップする。
 s : モデル一覧表の番号を入力する。

モデルに所有者名が設定されている場合に、所有者名を指示する。

owner : 所有者名を入力する。

新しいモデル名を指示する。

new_name : 新しいモデル名を入力する。

コマンドを実行する。

<CE> : モデルの管理番号及び名前を変更する。

5.3.1.7 モデルの所有者名を変更する

【メニュー】

〔モデル検索〕→〔モデル〕→〔変更〕→〔所有者名〕

【構文】

CHGOWNER	$\left\{ \begin{array}{l} \text{filename} \\ d \\ s \end{array} \right\}$	[owner] new_owner
----------	---	---------------------

所有者名を変更すると、モデルは更新され旧モデル(.OLD)が作成される。

所有者名を変更したいモデル名をつぎのいずれかの方法で指示する。

filename : モデル名を入力する。
 d : モデル一覧表のモデル名をピックアップする。
 s : モデル一覧表の番号を入力する。

所有者名を指定する。

owner : モデルに所有者名が設定されている場合は、現在の所有者名を入力する。
 new_owner : 新しい所有者名を入力する。英数字で 10 文字まで指定できる。所有者名を解除したいときは、"*"(アスタリスク)を入力する。

5.3.1.8 モデルタイトルを変更する

【メニュー】

〔モデル検索〕→〔モデル〕→〔変更〕→〔情報変更〕

【構文】

INFUPDATE	$\left\{ \begin{array}{l} \text{filename} \\ d \\ s \end{array} \right\}$	$\left\{ \begin{array}{l} \text{TTLSET no } \left\{ \begin{array}{l} \text{text} \\ \text{TLB} \end{array} \right\} \\ \text{TTLCAN no} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{TTLWRITE} \\ \text{TTLUPDATE} \end{array} \right\}$	<CE>
-----------	---	---	---	------

「書込み禁止」の情報ファイルは変更できません。

モデルタイトルを変更したいモデル名をつぎのいずれかの方法で指示する。

filename : モデル名を入力する。
 d : モデル一覧表のモデル名をピックアップする。
 s : モデル一覧表の番号を入力する。

検索条件の項目をつぎのいずれかで指示する。

TTLSET no text : 検索対象のモデルタイトル項目番号と検索文字列を入力する。複数の項目を指示できる。また、検索文字列を省略すると全文字列が対象になる。

TTLSET no TLB : テンプレートファイルから検索文字列を選択し指示する。複数の項目を指示できる。
 TTLCAN no : TTLSET で指示したモデルタイトル項目番号を取り消す。

情報ファイルの履歴を残すかどうかをつぎのいずれかの方法で指示する。

TTLWRITE : 履歴を残さずに情報ファイルを書き換える。
TTLUPDATE : 履歴を残し、情報ファイルを書き換える。

コマンドを実行する。

<CE> : モデルタイトルを変更する。

5.3.1.9 モデルの書込み属性を変更する

【メニュー】

〔モデル検索〕→〔モデル〕→〔変更〕→〔書込み〕

【構文】

WRITEONOFF	$\left\{ \begin{array}{l} \text{filename} \\ \text{d} \\ \text{s} \end{array} \right\} \text{ [owner]}$
------------	---

モデルの書込み属性を書込み可または書込み不可に変更する。
現在書込み可なら不可に、現在書込み不可なら可に変更する。

書込み属性を変更したいモデル名をつぎのいずれかで指示する。

filename : モデル名を入力する。
d : モデル一覧表のモデル名をピックする。
s : モデル一覧表の番号を入力する。

モデルに所有者名が設定されている場合に、所有者名を指示する。

owner : 所有者名を入力する。

5.3.1.10 モデルを削除する

【メニュー】

〔モデル検索〕→〔モデル〕→〔削除〕

【構文】

MDLDELETE	$\left\{ \left[\left\{ \begin{array}{l} \text{filename} \\ \text{d} \\ \text{s} \end{array} \right\} \right]^+ \right\} \text{ [owner] } \text{ <CE>}$ ALL
-----------	---

1 回目の削除はモデルファイルとモデル情報ファイルを削除する。
管理ファイル上には情報は保持される。管理番号が負の値になる。2 回目の削除は、モデルに関する情報を全て削除する。

削除されたモデルは、旧モデル (.OLD) になる。

「書込み禁止」になっている場合は削除できません。

削除するモデル名をつぎのいずれかで指示する。複数のモデルを選択可。

filename : モデル名を入力する。

- d : モデル一覧表のモデル名をピックアップする。
s : モデル一覧表の番号を入力する。

モデルを一括で削除したい場合に指示する。

- ALL : 全モデルを削除対象にする。

モデルに所有者名が設定されている場合に、所有者名を指示する。

- owner : 所有者名を入力する。
ALL の場合、所有者名が設定されていないモデルおよび所有者名が一致したモデルが削除対象になる。

コマンドを実行する

- <CE> : モデルを削除する。

5.3.1.11 モデルの検索結果の一覧をプリンタおよびファイルに出力する

【メニュー】

〔モデル検索〕 → 〔モデル〕 → 〔リスト出力〕

【構文】

LPR LPRNAME [filename1]	$\left[\begin{array}{c} \left\{ \begin{array}{c} \text{LPRONN} \\ \text{LPROFF} \end{array} \right\} \end{array} \right]$	$\left[\begin{array}{c} \text{LPRTMP} \left\{ \begin{array}{c} d \\ \text{filename2} \end{array} \right\} \end{array} \right]$	GO
---------------------------	--	---	----

出力ファイルを指示する。

- LPRNAME filename1
: 出力ファイル名を入力する。出力ファイルは、ACAD.SET のキーワード #MDLINP:LIST# で定義したディレクトリに作成する。

検索結果の一覧をプリンタ出力するかどうかを指示する。

- LPRONN : プリンタへ出力する。
LPROFF : プリンタへ出力しない。(省略時)

出力テンプレートファイル名を選択するときに指示する。

- LPRTMP : 出力テンプレートファイル名の選択をする。

出力テンプレートファイル名を指示する。

- d : 出力テンプレートファイル名をピックアップする。
filename2 : 出力テンプレートファイル名を入力する。出力テンプレートファイル名は、ACAD.SET のキーワード #MDLINP:TEMP# で定義したディレクトリになければならない。省略すると標準のフォーマットで出力する。

出力を開始する。

- GO : 出力を開始する。

5.3.2 シンボル検索

● 機能

シンボルファイルの一覧を表示し、さらに名前の変更、ファイルの保護、削除を行う。

● コマンド一覧

メニュー表示	コマンド	機能
シンボル検索	SYMDISP	
検索条件		
管理番号	SYMMGNO	シンボルの管理番号を指定して検索する。
名前	SYMNAME	シンボル名を指定して検索する。
作成日	SYMDATE	シンボルの作成日を指定して検索する。
更新日	SYMUPDATE	シンボルの更新日を指定して検索する。
所有者	SYMOWNER	シンボルの所有者名を指定して検索する。
前回条件	USELAST	前回設定した検索条件を使用する。
検索		
絞込 G O	CONTINUE	検索の絞り込み処理を開始する。
表示		
選択	SYMDRAW	シンボル表示するシンボル名を選択する。
表示 G O	DRAWGO	シンボル表示を開始する。
表示条件		
カラー表示	ACTPLN	カラーでシンボルを表示する。
単色表示	TMPPLN	単色でシンボルを表示する。
画面分割数	DSPPIC	シンボルの表示ピクチャおよび画面の分割数を指定する。
次頁		次ページのメニューを表示する。

メニュー表示	コマンド	機能
シンボル検索	SYMDISP	
変更		
名前	CHGNAME	シンボル名を変更する。
所有者	CHGOWNER	所有者名を変更する。
保護		
書込み	WRITEONOFF	シンボルの書込みスイッチ。
削除		
削除	SYMDELETE	シンボルを削除する。
全部	ALL	全シンボルを削除する。

メニュー表示	コマンド	機能
リスト出力		
リスト出力	LPR	検索結果をプリンタに出力する。
前頁		前ページのメニューを表示する。

★ 一覧表の表示についての注意

表示用テンプレートファイル MDLINF_SYMBOL_LIST.MDT を作成することにより、任意のフォーマットで表示させることができます。表示用テンプレートファイルについては、『5.2.5 表示用テンプレートファイルの作成』の(B)を参照してください。

5.3.2.1 シンボルを検索する

【メニュー】

〔モデル検索〕→〔シンボル〕

【構文】

SYMDISP	$\left[\begin{array}{l} \left\{ \begin{array}{l} D_ONN \\ D_OFF \end{array} \right\} \\ \left\{ \begin{array}{l} ONN \\ OFF \end{array} \right\} \end{array} \right]$	$\left[\begin{array}{l} SYMMGNO \text{ no} \\ SYMNAME \text{ name} \\ SYMDATE \text{ date} \\ SYMUPDATE \text{ date} \\ SYMOWNER \text{ owner} \\ USELAST \end{array} \right]$	$\left\{ \begin{array}{l} \langle CE \rangle \\ CONTINUE \end{array} \right\}$
---------	--	---	--

検索結果を画面に一覧表示するかどうかを指示する。

D_ONN : 一覧表示をする。(省略時)
D_OFF : 一覧表示をしない。

検索結果をファイルに出力するかどうかを指示する。

ONN : 検索結果を出力する。
OFF : 検索結果を出力しない。(省略時)

検索対象をつぎのいずれかの方法で指示する。

SYMMGNO no : 管理番号を入力する。省略時は全管理番号が検索対象になる。管理番号は複数指示できる。
SYMNAME name : シンボル名を入力する。省略時は全シンボル名が検索対象になる。
SYMDATE date : シンボル作成日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 1994/01/01
SYMUPDATE date : シンボル更新日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 1994/01/01
SYMOWNER owner : シンボルの所有者名を入力する。
USELAST : 最後に検索を行ったときの検索条件を使用する。

つぎのいずれかの方法で検索を開始する。

<CE> : 検索を開始する。
CONTINUE : 絞り込み検索を開始する。

● シンボル名を変更する

※ モデル検索の「モデル名の変更」を参照のこと。

● シンボルの所有者名を変更する

※ モデル検索の「モデルの所有者名を変更」を参照のこと。

● シンボルの書込み属性を変更する

※ モデル検索の「モデルの書込み属性を変更」を参照のこと。

5.3.2.2 シンボルを削除する

【メニュー】

〔モデル検索〕→〔シンボル〕→〔削除〕

【構文】

SYMDELETE	$\left\{ \begin{array}{l} \left[\begin{array}{l} \text{filename} \\ d \\ s \end{array} \right]^+ \\ ALL \end{array} \right\} [\text{owner}] \langle CE \rangle$
-----------	--

削除されたシンボルは、旧シンボル (.OLD) になる。

「書込み禁止」になっている場合は削除できません。

削除するシンボル名をつぎのいずれかで指示する。複数のシンボルを選択可。

filename: シンボル名を入力する。

d : シンボル一覧表のシンボル名をピックする。

s : シンボル一覧表の番号を入力する。

シンボルを一括で削除したい場合に指示する。

ALL : 全シンボルを削除対象にする。

シンボルに所有者名が設定されている場合に、所有者名を指示する。

owner : 所有者名を入力する。

ALL の場合、所有者名が設定されていないシンボルおよび所有者名が一致したシンボルが削除対象になる。

コマンドを実行する。

〈CE〉 : シンボルを削除する。

● シンボルの検索結果の一覧をプリンタおよびファイルに出力する

※ モデル検索の「モデルの検索結果の一覧をプリンタおよびファイルに出力」を参照のこと。
テンプレートファイルは使用できない。

5.3.3 APG 検索

● 機能

APG ファイルの一覧を表示し、さらに名前の変更、ファイルの保護、削除を行う。

● コマンド一覧

メニュー表示	コマンド	機能
APG 検索	APGDISP	

メニュー表示	コマンド	機能
検索条件		
管理番号	APGMGNO	A P G ファイルの管理番号を指定して検索する。
名前	APGNAME	A P G ファイル名を指定して検索する。
作成日	APGDATE	A P G ファイルの作成日を指定して検索する。
更新日	APGUPDATE	A P G ファイルの更新日を指定して検索する。
所有者	APGOWNER	A P G ファイルの所有者名を指定して検索する。
前回条件	USELAST	前回設定した検索条件を使用する。
絞込 G O	CONTINUE	検索の絞り込み処理を開始する。
変更		
名前	CHGNAME	A P G ファイル名を変更する。
所有者	CHGOWNER	所有者名を変更する。
保護		
書込み	WRITEONOFF	A P G ファイルの書込みスイッチ。
削除		
削除	APGDELETE	A P G ファイルを削除する。
全部	ALL	全 A P G ファイルを削除する。
リスト出力		
リスト出力	LPR	検索結果をプリンタに出力する。

5.3.3.1 A P G ファイルを検索する

【メニュー】

〔モデル検索〕 → 〔APG〕

【構文】

APGDISP	$\left[\left\{ \begin{array}{l} \text{D_ONN} \\ \text{D_OFF} \end{array} \right\} \right]$	$\left[\left\{ \begin{array}{l} \text{APGMGNO no} \\ \text{APGNAME name} \\ \text{APGDATE date} \\ \text{APGUPDATE date} \\ \text{APGOWNER owner} \\ \text{USELAST} \end{array} \right\} \right]$	$\left\{ \begin{array}{l} \text{<CE>} \\ \text{CONTINUE} \end{array} \right\}$
	$\left[\left\{ \begin{array}{l} \text{ONN} \\ \text{OFF} \end{array} \right\} \right]$		

検索結果を画面に一覧表示するかどうかを指示する。

D_ONN : 一覧表示をする。(省略時)
D_OFF : 一覧表示をしない。

検索結果をファイルに出力するかどうかを指示する。

ONN : 検索結果を出力する。
OFF : 検索結果を出力しない。(省略時)

検索対象をつぎのいずれかの方法で指示する。

APGMGNO no : 管理番号を入力する。省略時は全管理番号が検索対象になる。管理番号は複数指示できる。
APGNAME name : A P G ファイル名を入力する。省略時は全 A P G ファイル名が検索対象になる。
APGDATE date : A P G ファイル作成日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 1994/01/01
APGUPDATE date : A P G ファイル更新日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 1994/01/01
APGOWNER owner : A P G ファイルの所有者名を入力する。
USELAST : 最後に検索を行ったときの検索条件を使用する。

つぎのいずれかの方法で検索を開始する。

<CE> : 検索を開始する。
CONTINUE : 絞り込み検索を開始する。

- A P G ファイル名を変更する
※ モデル検索の「モデル名の変更」を参照のこと。
- A P G ファイルの所有者名を変更する
※ モデル検索の「モデルの所有者名を変更」を参照のこと。
- A P G ファイルの書込み属性を変更する
※ モデル検索の「モデルの書込み属性を変更」を参照のこと。

5.3.3.2 A P G ファイルを削除する

【メニュー】

〔モデル検索〕 → 〔APG〕 → 〔削除〕

【構文】

APGDELETE	$\left\{ \begin{array}{l} \left[\begin{array}{l} \text{filename} \\ d \\ s \end{array} \right] + \\ \text{ALL} \end{array} \right\} \left[\text{owner} \right] \text{ <CE>}$
-----------	--

削除された A P G ファイルは、旧 A P G ファイル(.OLD)になる。
「書込み禁止」になっている場合は削除できません。

削除する A P G ファイル名をつぎのいずれかで指示する。複数の A P G ファイルを選択可。

filename : A P G ファイル名を入力する。
d : A P G ファイル一覧表の A P G ファイル名をピックする。
s : A P G ファイル一覧表の番号を入力する。

A P G ファイルを一括で削除したい場合に指示する。

ALL : 全 A P G ファイルを削除対象にする。

A P G ファイルに所有者名が設定されている場合に、所有者名を指示する。

owner : 所有者名を入力する。
ALL の場合、所有者名が設定されていない A P G ファイルおよび所有者名が一致した A P G
ファイルが削除対象になる。

コマンドを実行する

<CE> : A P G ファイルを削除する。

- A P G ファイルの検索結果の一覧をプリンタおよびファイルに出力する
 - ※ モデル検索の「モデルの検索結果の一覧をプリンタおよびファイルに出力」を参照のこと。テンプレートファイルは使用できない。

5.3.4 旧モデル検索

● 機能

旧モデルファイルの一覧を表示し、さらに以下の機能を実行します。

● コマンド一覧

メニュー表示	コマンド	機能
旧モデル検索	MDL2DISP	
検索条件		
管理番号	MDLMGNO	旧モデルの管理番号を指定して検索する。
名前	MDLNAME	旧モデル名を指定して検索する。
作成日	MDLDATE	旧モデルの作成日を指定して検索する。
更新日	MDLUPDATE	旧モデルの更新日を指定して検索する。
所有者	MDLOWNER	旧モデルの所有者名を指定して検索する。
前回条件	USELAST	前回設定した検索条件を使用する
検索		
絞込 G O	CONTINUE	検索の絞り込み処理を開始する。
復元		
復元	RECOVER	旧モデルを現世代モデルに復元する。
削除		
削除	MDLDELETE	旧モデルを削除する。
廃棄		
廃棄	DISCARD	旧モデルを廃棄する。

注) コンフィグレーションファイル ACAD.SET でキーワード #MODEL:OLD# が設定されている場合は、旧モデルの検索で検索することはできません。

5.3.4.1 旧モデルを検索する

【メニュー】

〔モデル検索〕 → 〔旧モデル〕

【構文】

MDL2DISP	$\left[\begin{array}{l} \left\{ \begin{array}{l} \text{D_ONN} \\ \text{D_OFF} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{ONN} \\ \text{OFF} \end{array} \right\} \end{array} \right] \left[\begin{array}{l} \left\{ \begin{array}{l} \text{MDLMGNO no} \\ \text{MDLNAME name} \\ \text{MDLDATE date} \\ \text{MDLUPDATE date} \\ \text{MDLOWNER owner} \\ \text{USELAST} \end{array} \right\}^* \\ \left\{ \begin{array}{l} \text{<CE>} \\ \text{CONTINUE} \end{array} \right\} \end{array} \right]$
----------	--

検索結果を画面に一覧表示するかどうかを指示する。

D_ONN : 一覧表示をする。(省略時)
D_OFF : 一覧表示をしない。

検索結果をファイルに出力するかどうかを指示する。

ONN : 検索結果を出力する。
OFF : 検索結果を出力しない。(省略時)

検索対象をつぎのいずれかの方法で指示する。

MDLMGNO no : 管理番号を入力する。省略時は全管理番号が検索対象になる。管理番号は複数指示できる。
MDLNAME name : 旧モデル名を入力する。省略時は全旧モデル名が検索対象になる。
MDLDATE date : 旧モデル作成日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 1994/01/01
MDLUPDATE date : 旧モデル更新日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 1994/01/01
MDLOWNER owner : 旧モデルの所有者名を入力する。
USELAST : 最後に検索を行ったときの検索条件を使用する。

つぎのいずれかの方法で検索を開始する。

<CE> : 検索を開始する。
CONTINUE : 絞り込み検索を開始する。

5.3.4.2 旧モデルを現モデルに復元する

【メニュー】

〔モデル検索〕 → 〔旧モデル〕 → 〔復元〕

【構文】

RECOVER	$\left\{ \begin{array}{l} \text{filename} \\ \text{d} \\ \text{s} \end{array} \right\} [\text{owner}] \text{<CE>}$
---------	--

現モデルが存在しているときは旧モデルと現モデルが入れ代わる。
現モデルが「書き込み禁止」になっている場合は復元できません。

復元するモデル名をつぎのいずれかで指示する。

filename : モデル名を入力する。
 d : モデル一覧表のモデル名をピックアップする。
 s : モデル一覧表の番号を入力する。

モデルに所有者名が設定されている場合に、所有者名を指示する。

owner : 所有者名を入力する。

コマンドを実行する。

<CE> : 旧モデルを現モデルに復元する。

5.3.4.3 旧モデルを削除する

【メニュー】

〔モデル検索〕→〔旧モデル〕→〔削除〕

【構文】

MDLDELETE	$\left\{ \begin{array}{l} \left[\begin{array}{l} \text{filename} \\ d \\ s \end{array} \right] + \\ \text{ALL} \end{array} \right\}$	[owner] <CE>
-----------	---	----------------

※ モデル検索の「モデルを削除」を参照のこと。
 削除された旧モデル (.OLD) はディスクから消える。

5.3.4.4 旧モデルを廃棄する

【メニュー】

〔モデル検索〕→〔旧モデル〕→〔廃棄〕

【構文】

DISCARD	$\left\{ \begin{array}{l} \left[\begin{array}{l} \text{filename} \\ d \\ s \end{array} \right] + \\ \text{ALL} \end{array} \right\}$	[owner] <CE>
---------	---	----------------

※ モデル検索の「モデルを削除」を参照のこと。

5.3.5 廃棄モデル検索

● 機能

廃棄モデルファイルの一覧を表示し、さらに以下の機能を実行します。

● コマンド一覧

メニュー表示	コマンド	機能
廃棄モデル検索	MDL3DISP	
検索条件		
名前	MDLNAME	廃棄モデル名を指定して検索する。
作成日	MDLDATE	廃棄モデルの作成日を指定して検索する。
更新日	MDLUPDATE	廃棄モデルの更新日を指定して検索する。
廃棄日	MDLACDATE	廃棄モデルの廃棄日を指定して検索する。
検索		
検索 GO	LISTGO	検索を開始する。
絞込 GO	CONTINUE	検索の絞り込み処理を開始する。
情報表示		
表示	INFDISP	モデル情報を表示する。
復元		
復元	RECOVER	廃棄モデルを現モデルに復元する。
削除		
削除	MDLDELETE	廃棄モデルを削除する。
全部	ALL	全廃棄モデルを削除する。

5.3.5.1 廃棄モデルを検索する

【メニュー】

〔モデル検索〕 → 〔廃棄モデル〕

【構文】

MDL3DISP password	$\left[\begin{array}{c} \left\{ \begin{array}{c} D_ONN \\ D_OFF \end{array} \right\} \\ \left\{ \begin{array}{c} ONN \\ OFF \end{array} \right\} \end{array} \right] \left[\begin{array}{c} \left\{ \begin{array}{c} MDLNAME \text{ name} \\ MDLDATE \text{ date} \\ MDLUPDATE \text{ date} \\ MDLACDATE \text{ date} \end{array} \right\} * \left\{ \begin{array}{c} <CE> \\ LISTGO \\ CONTINUE \end{array} \right\} \end{array} \right]$
-------------------	--

パスワードを入力する。

password : パスワードを入力する。

検索結果を画面に一覧表示するかどうかを指示する。

D_ONN : 一覧表示をする。(省略時)

D_OFF : 一覧表示をしない。

検索結果をファイルに出力するかどうかを指示する。

ONN : 検索結果を出力する。
OFF : 検索結果を出力しない。(省略時)

検索対象をつぎのいずれかの方法で指示する。

MDLNAME name : 廃棄モデル名を入力する。省略時は全旧モデル名が検索対象になる。
MDLDATE date : 廃棄モデル作成日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 2000/08/01
MDLUPDATE date : 廃棄モデル更新日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 2000/08/01
MDLACDATE date : 廃棄モデル廃棄日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 2000/08/01

つぎのいずれかの方法で検索を開始する。

<CE> : 検索を開始する。
LISTGO : 検索を開始する。
CONTINUE : 絞り込み検索を開始する。

5.3.5.2 廃棄モデルを現モデルに復元する

【メニュー】

〔モデル検索〕→〔廃棄モデル〕→〔復元〕

【構文】

RECOVER	mgno	$\left\{ \begin{array}{l} \text{filename} \\ d \\ s \end{array} \right\}$	[owner] <CE>
---------	------	---	----------------

現モデルが存在しているときは廃棄モデルと現モデルが入れ代わる。
現モデルが「書込み禁止」になっている場合は復元できません。

復元先の管理番号を指定する。

mgno : 管理番号。

復元するモデル名をつぎのいずれかで指示する。

filename : モデル名を入力する。
d : モデル一覧表のモデル名をピックする。
s : モデル一覧表の番号を入力する。

モデルに所有者名が設定されている場合に、所有者名を指示する。

owner : 所有者名を入力する。

コマンドを実行する。

<CE> : 廃棄モデルを現モデルに復元する。

5.3.5.3 廃棄モデルを削除する

【メニュー】

〔モデル検索〕→〔廃棄モデル〕→〔削除〕

【構文】

MDLDELETE	$\left\{ \begin{array}{l} \left[\begin{array}{l} \text{filename} \\ d \\ s \end{array} \right]^+ \\ \text{ALL} \end{array} \right\} \left[\text{owner} \right] \langle \text{CE} \rangle$
-----------	---

※ コマンド引数等はモデル検索の「モデルを削除」を参照のこと。
削除された旧モデル (.OLD) はディスクから消える。

5.3.6 旧シンボル検索

- 機能
旧シンボルファイルの一覧を表示し、さらに以下の機能を実行します。
- コマンド一覧

メニュー表示	コマンド	機能
旧シンボル検索	SYM2DISP	
検索条件		
管理番号	SYMMGNO	旧シンボルの管理番号を指定して検索する。
名前	SYMNAME	旧シンボル名を指定して検索する。
作成日	SYMDATE	旧シンボルの作成日を指定して検索する。
更新日	SYMUPDATE	旧シンボルの更新日を指定して検索する。
所有者	SYMOWNER	旧シンボルの所有者名を指定して検索する。
前回条件	USELAST	前回設定した検索条件を使用する。
検索		
絞込 G O	CONTINUE	検索の絞り込み処理を開始する。
復元		
復元	RECOVER	旧シンボルを現世代シンボルに復元する。
削除		
削除	SYMDELETE	旧シンボルを削除する。
廃棄		
廃棄	DISCARD	旧モデルを廃棄する。

5.3.6.1 旧シンボルを検索する

【メニュー】

〔モデル検索〕 → 〔旧シンボル〕

【構文】

SYM2DISP	<table><tr><td><table><tr><td>D_ONN</td></tr><tr><td>D_OFF</td></tr></table></td><td><table><tr><td>ONN</td></tr><tr><td>OFF</td></tr></table></td></tr></table>	<table><tr><td>D_ONN</td></tr><tr><td>D_OFF</td></tr></table>	D_ONN	D_OFF	<table><tr><td>ONN</td></tr><tr><td>OFF</td></tr></table>	ONN	OFF	<table><tr><td><table><tr><td>SYMMGNO no</td></tr><tr><td>SYMNAME name</td></tr><tr><td>SYMDATE date</td></tr><tr><td>SYMUPDATE date</td></tr><tr><td>SYMOWNER owner</td></tr><tr><td>USELAST</td></tr></table></td><td>*</td><td><table><tr><td><CE></td></tr><tr><td>CONTINUE</td></tr></table></td></tr></table>	<table><tr><td>SYMMGNO no</td></tr><tr><td>SYMNAME name</td></tr><tr><td>SYMDATE date</td></tr><tr><td>SYMUPDATE date</td></tr><tr><td>SYMOWNER owner</td></tr><tr><td>USELAST</td></tr></table>	SYMMGNO no	SYMNAME name	SYMDATE date	SYMUPDATE date	SYMOWNER owner	USELAST	*	<table><tr><td><CE></td></tr><tr><td>CONTINUE</td></tr></table>	<CE>	CONTINUE
<table><tr><td>D_ONN</td></tr><tr><td>D_OFF</td></tr></table>	D_ONN	D_OFF	<table><tr><td>ONN</td></tr><tr><td>OFF</td></tr></table>	ONN	OFF														
D_ONN																			
D_OFF																			
ONN																			
OFF																			
<table><tr><td>SYMMGNO no</td></tr><tr><td>SYMNAME name</td></tr><tr><td>SYMDATE date</td></tr><tr><td>SYMUPDATE date</td></tr><tr><td>SYMOWNER owner</td></tr><tr><td>USELAST</td></tr></table>	SYMMGNO no	SYMNAME name	SYMDATE date	SYMUPDATE date	SYMOWNER owner	USELAST	*	<table><tr><td><CE></td></tr><tr><td>CONTINUE</td></tr></table>	<CE>	CONTINUE									
SYMMGNO no																			
SYMNAME name																			
SYMDATE date																			
SYMUPDATE date																			
SYMOWNER owner																			
USELAST																			
<CE>																			
CONTINUE																			

※ コマンド引数との説明はシンボル検索の「シンボルを検索」を参照のこと。

● 旧シンボルを現シンボルに復元する

※ 旧モデル検索の「旧モデルを現モデルに復元」を参照のこと。

5.3.6.2 旧シンボルを削除する

【メニュー】

〔モデル検索〕 → 〔旧シンボル〕 → 〔削除〕

【構文】

SYMDELETE	$\left\{ \left[\left\{ \begin{array}{c} \text{filename} \\ d \\ s \end{array} \right\}^+ \right] \right\}$	[owner] <CE>
	ALL	

※ コマンド引数等はシンボル検索の「シンボルを削除」を参照のこと。
削除された旧シンボル (. OLD) はディスクから消える。

5.3.7 旧 APG 検索

● 機能

旧 A P G ファイルの一覧を表示し、さらに以下の機能を実行します。

● コマンド一覧

メニュー表示	コマンド	機能
旧 A P G 検索	APG2DISP	
検索条件		
管理番号	APGMGNO	旧 A P G ファイルの管理番号を指定して検索する。
名前	APGNAME	旧 A P G ファイル名を指定して検索する。

メニュー表示 作成日	コマンド APGDATE	機能 旧 A P G ファイルの作成日を指定して検索する。
更新日	APGUPDATE	旧 A P G ファイルの更新日を指定して検索する。
所有者	APGOWNER	旧 A P G ファイルの所有者名を指定して検索する。
前回条件	USELAST	前回設定した検索条件を使用する。
検索 絞込 G O	CONTINUE	検索の絞り込み処理を開始する。
復元 復元	RECOVER	旧 A P G ファイルを現世代 A P G ファイルに復元する。
削除 削除	APGDELETE	旧 A P G ファイルを削除する。

5.3.7.1 旧 A P G ファイルを検索する

【メニュー】

〔モデル検索〕→〔旧 A P G〕

【構文】

APG2DISP	$\left[\left[\left\{ \begin{array}{l} \text{D_ONN} \\ \text{D_OFF} \end{array} \right\} \right] \left\{ \begin{array}{l} \text{ONN} \\ \text{OFF} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{APGMGNO no} \\ \text{APGNAME name} \\ \text{APGDATE date} \\ \text{APGUPDATE date} \\ \text{APGOWNER owner} \\ \text{USELAST} \end{array} \right\} \right]^* \left\{ \begin{array}{l} \text{<CE>} \\ \text{CONTINUE} \end{array} \right\}$
----------	---

※ A P G ファイル検索の「A P G ファイルを検索」を参照のこと。

● 旧 A P G ファイルを現 A P G ファイルに復元する

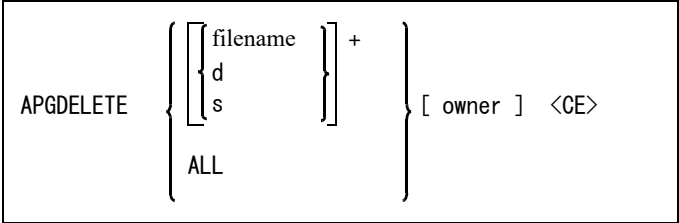
※ 旧モデル検索の「旧モデルを現モデルに復元」を参照のこと。

5.3.7.2 旧 A P G ファイルを削除する

【メニュー】

〔モデル検索〕→〔旧 A P G〕→〔削除〕

【構文】



※ A P G ファイル検索の「A P G ファイルを削除」を参照のこと。
 削除された旧 A P G ファイル (.OLD) はディスクから消える。

5.3.8 ファイル検索

- 機能
各ファイルの一覧を表示し、さらに以下の機能を実行します。
- コマンド一覧

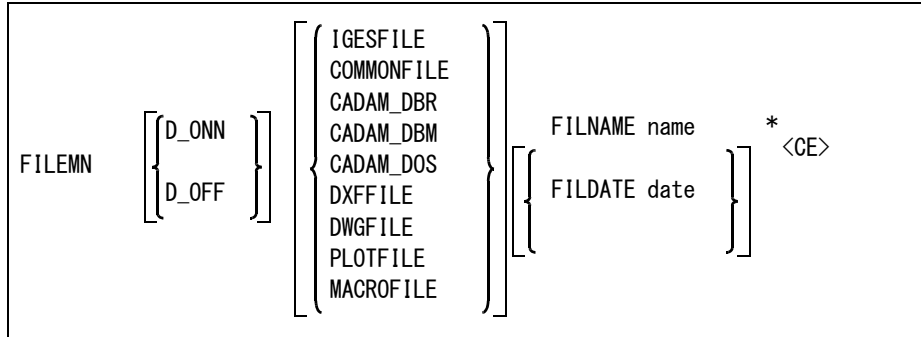
メニュー表示	コマンド	機能
ファイル種別	FILEMN	
IGES	IGESFILE	IGES データファイルを検索する。
COMMON	COMMONFILE	共通データファイルを検索する。
CADAM IF_R	CADAM_IFR	Regular CADAM 中間ファイルを検索する。
CADAM IF_M	CADAM_IFM	Micro CADAM "
CADAM DB_R	CADAM_DBR	Regular CADAM 図形ファイルを検索する。
CADAM DB_M	CADAM_DBM	Micro CADAM "
CADAM DOS	CADAM_DOS	Micro CADAM の DOS ファイルを検索する。
DXF	DXFFILE	DXF ファイルを検索する。
PLOT	PLOTFILE	プロットファイルを検索する。
MACRO	MACROFILE	マクロファイルを検索する。
検索		
名前	FILNAME	ファイル名を指定して検索する。
作成日	FILDATE	ファイルの作成日を指定して検索する。
名前変更	CHGNAME	ファイル名を変更する。
書込み保護	WRITEONOFF	ファイルの書込みスイッチ。
削除	FILDELETE	ファイルを削除する。
リスト出力	LPR	検索結果をプリンタに出力する。

5.3.8.1 ファイル種類別に検索する

【メニュー】

〔モデル検索〕→〔ファイル〕

【構文】



検索結果を画面に一覧表示するかどうかを指示する。

D_ONN : 一覧表示をする。(省略時)
D_OFF : 一覧表示をしない。

検索対象ファイルの種類をつぎのいずれかで指示する。

IGESFILE : IGES データファイルを指定する。
COMMONILE : 共通データファイルを指定する。
CADAM_DBR : Regular CADAM 図形ファイルを指定する。
CADAM_DBM : Micro CADAM " "
CADAM_DOS : Micro CADAM の DOS ファイルを指定する。
DXFFILE : DXF ファイルを指定する。
DWGFILE : DWG ファイルを指定する。
PLOTFILE : プロットファイルを指定する。
MACROFILE : マクロファイルを指定する。

検索対象をつぎのいずれかの方法で指示する。

FILNAME name : ファイル名を入力する。省略時は全ファイル名が検索対象になる。
FILDATE date : ファイル作成日を入力する。日付 date は yyyy/mm/dd と入力する。
例) 1994/01/01

検索を開始する。

<CE> : 検索を開始する。

- ファイル名を変更する
※ モデル検索の「モデル名の変更」を参照のこと。
- ファイルの書き込み属性を変更する
※ モデル検索の「モデルの書き込み属性を変更」を参照のこと。
- ファイルの読み込み属性を変更する
※ モデル検索の「モデルの読み込み属性を変更」を参照のこと。

5.3.8.2 ファイルを削除する

【メニュー】

〔モデル検索〕→〔ファイル〕→〔削除〕

【構文】

```
FILDELETE d <CE>
```

「書込み禁止」になっている場合は削除できません。

削除するファイル名を指示する。

d : ファイル一覧表のファイル名をピックアップする。

コマンドを実行する

<CE> : ファイルを削除する。

● ファイルの検索結果の一覧をプリンタおよびファイルに出力する

※ モデル検索の「モデルの検索結果の一覧をプリンタおよびファイルに出力」を参照のこと。テンプレートファイルは使用できない。

5.3.9 モデル登録

● 機能

未登録のモデル名を管理ファイルに登録する。

● コマンド一覧

メニュー表示	コマンド	機能
モデル登録	MDLCATALOG	
管理番号	MDLMGNO	登録するモデルの管理番号を指定する。
検索条件		
名前	MDLNAME	モデル名を指定する。(指定がなければ全未登録モデル)
検索		
絞り込 G O	CONTINUE	検索の絞り込み処理を開始する。
登録		
選択	MDLREGIST	選択したモデルを登録する。
全モデル	ALL	全モデルを登録する。

5.3.9.1 未登録のモデルを検索する

【メニュー】

〔モデル検索〕 → 〔モデル登録〕

【構文】

MDLCATALOG	MDLMGNO	$\left\{ \begin{array}{c} d \\ no \end{array} \right\}$	[MDLNAME name]	$\left\{ \begin{array}{c} \langle CE \rangle \\ CONTINUE \end{array} \right\}$
------------	---------	---	------------------	--

登録するモデルの管理番号をつぎのいずれかの方法で指示する。

MDLMGNO no : 管理番号を入力する。
MDLMGNO d : 一覧表の管理番号をピックアップする。

検索条件を指示する。

MDLNAME name : モデル名を入力する。省略時は指定された管理番号をもつ全未登録モデル名が検索対象になる。

つぎのいずれかの方法で検索を開始する。

<CE> : 検索を開始する。
CONTINUE : 絞り込み検索を開始する。

5.3.9.2 未登録のモデルを管理ファイルに登録する

【メニュー】

〔モデル検索〕 → 〔モデル登録〕 → 〔登録〕 / 〔全モデル〕

【構文】

$\left\{ \begin{array}{c} MDLREGIST \\ ALL \end{array} \right\}$	$\left\{ \begin{array}{c} \text{filename} \\ d \\ s \end{array} \right\}$	<CE>
--	---	------

登録するモデルをつぎのいずれかで指示する。

MDLREGIST filename : モデル名を入力する。
MDLREGIST d : モデル一覧表のモデル名をピックアップする。
MDLREGIST s : モデル一覧表の番号を入力する。
ALL : 全モデルを選択。

コマンドを実行する

<CE> : モデルに登録する。

5.3.10 親子関係検索

● 機能

モデルが参照している子供名、サブモデル名、シンボル名および部品名の一覧を表示する。
またはサブモデル、シンボルおよび部品を参照しているモデル名の一覧を表示する。

● コマンド一覧

メニュー表示	コマンド	機能
子供検索	REFLIST	

メニュー表示	コマンド	機能
管理番号	MDLMGNO	親の管理番号を指定して検索する。
サブモデル	REFSUB	モデルが参照しているサブモデル名の一覧を表示する。
シンボル	REFSYM	モデルが参照しているシンボル名の一覧を表示する。
部品	REFPART	モデルが参照している部品名の一覧を表示する。
子供	REFCHILD	親子関係作成で作成された子供名の一覧を表示する。
参照リスト	REFDISP	指定されたモデルが参照しているサブモデル名・シンボル名または部品名を表示する。
モデル情報	INFDISP	指定されたモデルの情報を表示する。
親検索	REFLIST	
管理番号	MDLMGNO	親の管理番号を指定して検索する。
サブモデル	XREFSUB	モデルが参照しているサブモデル名の一覧を表示する。
シンボル	XREFSYM	モデルが参照しているシンボル名の一覧を表示する。
部品	XREFPART	モデルが参照している部品名の一覧を表示する。
親	XREFCHILD	親子関係作成で作成された子供名を参照している親名の一覧を表示する。
一覧表		
モデル	MDLLIST	モデル名の一覧を表示する。
シンボル	SYMLIST	シンボル名の一覧を表示する。
リスト出力	LPR	検索結果をプリンタに出力する。

★ 親子関係検索をするための注意 (必要ファイル)

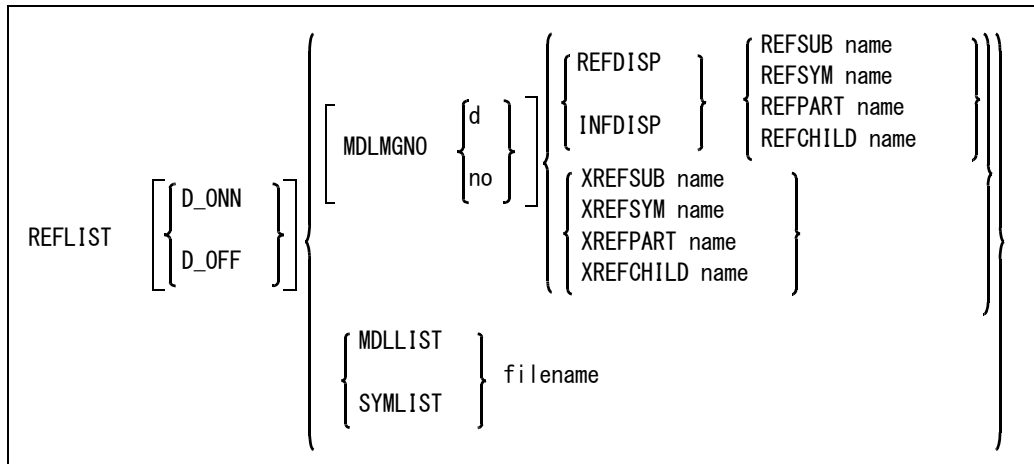
- (1) モデル情報ファイル (.MDI)
 コマンド『モデル__保存』(MODEL/WRITE) で情報ファイル (.MDI) 出力でモデル保存します。

5.3.10.1 親子関係を検索する

【メニュー】

〔モデル検索〕 → 〔親子関係検索〕

【構文】



検索結果を画面に一覧表示するかどうかを指示する。

D_ONN : 一覧表示をする。(省略時)
D_OFF : 一覧表示をしない。

検索するモデルの管理番号をつぎのいずれかの方法で指示する。

MDLMGNO no : 管理番号を入力する。省略時は全管理番号が検索対象になる。管理番号は複数指示できる。
MDLMGNO d : 一覧表の管理番号をピックアップする。

検索結果の表示方法をつぎのいずれかで指示する。

REFDISP : モデルが参照しているファイル名の一覧表示をする。(省略時)
INFDISP : モデルの情報を表示をする。

子供検索対象をつぎのいずれかの方法で指示する。

REFSUB name : モデル名を入力する。省略時は全モデル名が検索対象になる。指示されたモデルが参照しているサブモデル名を表示する。
REFSYM name : モデル名を入力する。省略時は全モデル名が検索対象になる。指示されたモデルが参照しているシンボル名を表示する。
REFPART name : モデル名を入力する。省略時は全モデル名が検索対象になる。指示されたモデルが参照している部品名を表示する。
REFCHILD name : モデル名を入力する。省略時は全モデル名が検索対象になる。指示された親モデルが参照している子供モデル名を表示する。

親検索対象をつぎのいずれかの方法で指示する。

XREFSUB name : サブモデル名を入力する。省略時は全サブモデル名が検索対象になる。指示されたサブモデルを参照しているモデル名が表示される。
XREFSYM name : シンボル名を入力する。省略時は全シンボル名が検索対象になる。指示されたシンボルを参照しているモデル名が表示される。
XREFPART name : 部品名を入力する。省略時は全モデル名が検索対象になる。指示された部品名を参照しているモデル名が表示される。
XREFCHILD name : 子供モデル名を入力する。省略時は全モデル名が検索対象になる。指示された子供モデルを参照している親モデル名が表示される。

表示するファイルの種類をつぎのいずれかの方法で指示する。

MDLLIST filename : モデル名の一覧を表示する。
SYMLIST filename : シンボル名の一覧を表示する。

● 親子関係検索の検索結果の一覧をプリンタおよびファイルに出力する

※ モデル検索の「モデルの検索結果の一覧をプリンタおよびファイルに出力」を参照のこと。
テンプレートファイルは使用できない。

5.3.11 親子関係作成

- 機能
親図が参照している子図 (子供) を登録する
- コマンド一覧

メニュー表示	コマンド	機能
モデル検索	CREMDLREF	モデル検索を参照。
管理番号	MDLMGNO	
名前	MDLNAME	
作成日	MDLDATE	
更新日	MDLUPDATE	
所有者	MDLOWNER	
タイトル	MDLTITLE	
絞込 G O	CONTINUE	
情報表示	INFDISP	
子供登録	REFCREATE	
登録	REFSET	子供名を登録する。
変更	REFCHANGE	子供名を変更する。
取消	REFCANCEL	子供名を取り消す。
圧縮	REFCOMP	取り消された子供名を整理し、アルファベット順に並べ換える
書き込み	REFWRITE	管理ファイルに書き込む。

5.3.11.1 親子関係を作成する

【メニュー】

〔モデル検索〕 → 〔親子関係作成〕

【構文】

REFCREATE	mdlisp	$\left. \begin{array}{l} \text{filename} \\ d \\ s \end{array} \right\}$	$\left. \begin{array}{l} \text{REFSET name} \\ \text{REFCHANGE } d \text{ name2} \\ \text{REFCANCEL } d \\ \text{REFCOMP} \end{array} \right\}$	REFWRITE <CE>
-----------	--------	--	---	---------------

※ mdlisp は「モデル検索」を参照。

作成したいモデルをつぎのいずれかで指示する。

filename : モデル名を入力する。
d : モデル一覧表のモデル名をピックアップする。

s : モデル一覧表の番号を入力する。

作成したい項目をつぎのいずれかで指示する。

REFSET name : 子供名を入力する。名前は 32 文字まで登録できる。登録数は 70 個まで設定できる。
 REFCHANGE d : 変更したい子供名をピックする。
 name2 : 新しい子供名を入力する。
 REFCANCEL d : 取り消す子供名をピックする。
 REFCOMP : 取り消された子供名を取り除き、子供名をアルファベット順に並べ換える。

管理ファイルに書き込みを開始する。

REFWRITE : 親子関係を管理ファイルに書き込む。

5.3.12 モデル情報作成

● 機能

手書や他のソフトウェアで作成された図面の情報を Advance CAD のモデルタイトルの項目に入力してモデル情報ファイル (.MDI) を作成します。

● コマンド一覧

メニュー表示	コマンド	機能
管理番号	MDIMGNO	モデル情報ファイルの管理番号を入力する。
読み込み	TTLREAD	すでに作成されているモデル情報ファイルを読み込む。
書込み		
変更	TTLWRITE	履歴を残さずに情報ファイルを書き換える。
更新	TTLUPDATE	履歴を残し、情報ファイルを書き換える。
タイトル		
設定	TTLSET	検索条件を設定する。
文章呼出	TLB	テキストファイルを呼出し検索条件を設定する。
取消	TTLCAN	検索条件を取り消す。
初期設定	MDLTITLE	モデルタイトルの全項目内容を削除する。(クリア)

5.3.12.1 新しいモデル情報ファイル作成する

【メニュー】

〔モデル検索〕 → 〔情報作成〕

【構文】

CREMDLINF MDIMGNO {no
d} [TTLREAD name1] {TTLSET no{text
TLB}
TTLCAN no} {TTLWRITE name2
TTLUPDATE name2} <CE>

作成するモデル情報ファイルの管理番号をつぎのいずれかの方法で指示する。

MDIMGNO no : 管理番号を入力する。省略時は全管理番号が検索対象になる。管理番号は複数指示できる。

MDIMGNO d : 一覧表の管理番号をピックする。

すでに作成されているモデル情報ファイルを読み込むときに指示する。

TTLREAD : すでに作成されているモデル情報ファイルを読み込む。

作成するモデル情報ファイル名を指示する。

name1 : 作成するモデル情報ファイル名を入力する。

作成条件の項目を指示する。

TTLSET no text : モデルタイトル項目番号と文字列を入力する。

TTLSET no TLB : テンプレートファイルから文字列を選択し入力する。

TTLCAN no : TTLSET で指示したモデルタイトル項目番号を取り消す。

MDLTITLE : モデルタイトルの全項目内容を削除する。

つぎのいずれかの方法で書き込みを指示する。

TTLWRITE name2 : 履歴を残さずに情報ファイルを書き換える。

TTLUPDATE name2 : 履歴を残し、情報ファイルを書き換える。

コマンドを実行する

<CE> : モデル情報ファイルを作成する。

5.3.13 モデルのプロット出力

- 機能
モデル検索の検索結果のモデルを選択してプロット出力する。
- コマンド一覧

メニュー表示	コマンド	機能
モデル作図	MDLPLOT	

メニュー表示	コマンド	機能
検索		
管理番号	MDLMGNO	モデル検索を参照。
名前	MDLNAME	
作成日	MDLDATE	
更新日	MDLUPDATE	
呼出し日	MDLACDATE	
所有者名	MDLOWNER	
タイトル	MDLTITLE	
頁タイトル	PAGETITLE	
前回条件	USELAST	
絞込 GO	CONTINUE	
出力		
オンライン	PLTONN	オンラインプロットを行う。
オフライン	PLTOFF	オフラインプロットを行う。
コピー枚数	NUM	コピー枚数を指定する。
指定&選択	SELECT	出力ページを指定してからモデルを選択する。
指定&全部	ALL	出力ページを指定して全モデルを選択する。
選択&指定	SELECT1	モデルを選択してから出力ページを指定する。
出力 GO	GO	出力を開始する。

注) ACAD.SET の #MODEL:MANAGE# で指定したディレクトリにプロッタ定義ファイル PLOTMDM.SET と ACAD.SET の #MDLIN:DISP# で指定したディレクトリに表示用テンプレートファイル MODEL_PLOT.MDT が必要です。以下の手順で作成します。

- 1) 出力するプロッタを PLOTMDM.SET に登録する。
 - 2) 表示用テンプレートファイル MODEL_PLOT.MDT をコピーする。
- PLOTMDM.SET, MODEL_PLOT.MDT は、ディレクトリ sample/PLOT にあります。

5.3.13.1 検索結果のモデルをプロット出力する

【メニュー】

〔モデル検索〕 → 〔モデル作図〕

【構文】

```
MDLPLOT  mdlisp  { PLTONN name } [NUM n] { SELECT page <CE> sel
                  { PLTOFF      }          { ALL page
                  { SELECT1 sel page <CE> } } GO
```

※ mdlisp は「モデル検索」を参照。

プロットの方法を指定する。

PLTONN name : オンラインプロットを行う。
出力するプロッタを指定する。(プロッタは、PLOTMDM.SET に登録しておく必要があります。)

PLTOFF : オフラインプロットを行う。

出力枚数を指定する。

NUM n : 出力枚数を入力する。指定がなければ 1。(1 ~ 999)

出力モデル、ページを指定する。

SELECT : 出力ページを指定してから出力モデルを選択する。
ALL : 出力ページを指定してから全モデルを選択する。
SELECT1 : モデルを選択してから出力ページを指定する。
page : 出力ページを指定する。
= 0 : 全ページを出力／取り消し
= n : ページ n を出力／取り消し
既に出力指定されているページを指定した場合は、取り消しとなる。

<CE> : ページ指定の終了。
sel : 出力モデルを選択する。

プロット出力を開始する。

GO : 出力を開始する。
プロット出力完了待ちをする場合は、1 モデル毎にプロット完了待ちのプロンプトを出力する。
<CE> 入力により次のモデルをプロットする。

5.3.14 プロットファイルの出力

● 機能

オフライン出力で作成したプロットファイルをプロッタへ出力する。

● コマンド一覧

メニュー表示	コマンド	機能
作図ファイル	PLOTMN	
検索		
名前	FILNAME	名前を指定して検索する。
作成日	FILDATE	作成日を指定して検索する。
作図日	PLTDATE	作図日を指定して検索する。
出力済	PLTONN	出力済みファイルを検索する。
未出力	PLTOFF	未出力ファイルを検索する。

メニュー表示	コマンド	機能
出力指示		
プロッタ	SELDEV	プロッタを選択する。
選択	SELECT	出力ファイルを選択する。
全部	ALL	全ファイルを選択する。
出力 G O	GO	出力を開始する。
リスト出力		
ファイル名	LPRNAME	出力ファイル名の一覧をリスト出力するファイル名を指定する。
P R 出力		
する	LPRONN	出力ファイル名の一覧をリスト出力する。
しない	LPROFF	出力ファイル名の一覧をリスト出力しない。
次頁		

メニュー表示	コマンド	機能
作図ファイル		
検索		
名前	FILNAME	} 前頁を参照。
作成日	FILDATE	
作図日	PLTDATE	
出力済	PLTONN	
未出力	PLTOFF	
変更		
名前変更	CHGNAME	ファイル名を変更する。
書込み保護	WRITEONOFF	ファイルの書き込み属性を変更する。
読込み保護	READONOFF	ファイルの読み込み属性を変更する。
削除		
削除	SELECT1	削除するファイルを選択する。
全部	FILDELETE	全ファイルを削除対象とする。
前頁		

- 注) ACAD.SET の #MODEL:MANAGE# で指定したディレクトリにプロッタ定義ファイル PLOTMDM.SET と表示用テンプレートファイル PLOT_FILE_MN.MDT が必要です。以下の手順で作成します。
- 1) 出力するプロッタを PLOTMDM.SET に登録する。

- 2) 表示用テンプレートファイル PLOT_FILE_MN.MDT をコピーする。
 PLOTMDM.SET, PLOT_FILE_MN.MDT は、ディレクトリ sample/PLOT にあります。

5.3.14.1 作図ファイルを検索する

【メニュー】

〔モデル検索〕 → 〔作図ファイル〕

【構文】

PLOTMN	$\left\{ \begin{array}{l} \text{FILNAME name} \\ \text{FILDATE date} \\ \text{PLTDATE date} \\ \text{PLTONN} \\ \text{PLTOFF} \end{array} \right\}$	<CE>
--------	---	------

検索条件を指定する。

FILNAME name : 検索する作図ファイルの名前を入力する。
 FILDATE date : 検索する作図ファイルの作成日を入力する。
 PLTDATE date : 検索する作図ファイルの作図日を入力する。
 PLTONN : 出力済の作図ファイルを検索する。
 PLTOFF : 未出力の作図ファイルを検索する。

ファイルの検索を開始する。

<CE> : 検索を開始する。

5.3.14.2 作図ファイルを出力する

【メニュー】

〔モデル検索〕 → 〔作図ファイル〕 → 〔プロッタ〕

【構文】

SELDEV	$\left\{ \begin{array}{l} d \\ s \end{array} \right\}$	$\left\{ \begin{array}{l} \text{SELECT } d \\ \text{ALL} \end{array} \right\}$	GO
--------	--	--	----

出力装置を選択する。

d : 装置名をピックする。
 s : 装置名の順序番号。

作図ファイルを選択する。

SELECT d : 出力する作図ファイルを選択する。
 ALL : 全作図ファイルを選択する。

作図ファイルを出力する。

GO : 出力を開始する。

5.3.14.3 作図ファイルの名前を変更する

【メニュー】

〔モデル検索〕 → 〔作図ファイル〕 → 〔名前変更〕

【構文】

```
CHGNAME d new_name
```

ファイルの名前を変更する。

d new_name: 選択した作図ファイルの名前を変更する。

5.3.14.4 作図ファイルの書き込み属性を変更する

【メニュー】

〔モデル検索〕 → 〔作図ファイル〕 → 〔書き込み保護〕

【構文】

```
WRITEONOFF d
```

ファイルの書き込み属性を変更する。

d : 選択した作図ファイルの書き込み属性を変更する。

5.3.14.5 作図ファイルを削除する

【メニュー】

〔モデル検索〕 → 〔作図ファイル〕 → 〔削除〕

【構文】

```
FILDELETE [ SELECT1 d ] <CE>
```

削除するファイルを選択する。

SELECT1 d : 削除する作図ファイルを選択する。省略すると全作図ファイルを削除する。

ファイルを削除する。

<CE> : 作図ファイルの削除を開始する。

5.3.15 ダンプファイルのロード

● 機能

『モデルダンプ』コマンドで作成したテープまたはフロッピーディスク内のモデルファイルを、指定したディレクトリにロードする。

● メニュー一覧

メニュー表示	コマンド	機能
T A R	TAR	ダンプファイルをロードする。
入力装置選択		
装置選択	SELDEV	入力装置を選択する。
ファイル番号		
番号	CSTNO	入力するファイル番号を指定する。
最後	CSTLASTFL	最後のファイル番号を入力する。
全部	CSTALLFL	全ファイル番号を入力する。
格納先		
管理番号	MDLMGNO	ファイルを格納する管理番号を指定する。
D I R 名	MDLFILE	ファイルを格納するディレクトリを指定する。
上書する	WONN	格納先に同じファイルが存在しても上書きする。
上書しない	WOFF	格納先に同じファイルが存在した場合は上書きしない。
ファイル一覧		
一覧作成	LISTGO	入力するファイル名の一覧を表示する。
入力ファイル		
選択	LOADMDL	ロードするファイルを選択する。
全部選択	LOADALL	全ファイルをロードする。
全部取消	CANCEL	選択を取り消す。
T A R 実行		
G O	LOADGO	ロードを開始する。

5.3.15.1 ダンプしたモデルを指定したディレクトリにロードする

【メニュー】

〔モデル検索〕→〔T A R〕

【構文】

TAR [SELDEV sel]	$\left[\begin{array}{l} \text{CSTNO no} \\ \text{CSTLASTFL} \\ \text{CSTALL} \end{array} \right]$	$\left[\begin{array}{l} \text{MDLMGNO sel} \\ \text{MDLFILE d} \end{array} \right]$	$\left[\begin{array}{l} \text{WONN} \\ \text{WOFF} \end{array} \right]$	LISTGO	$\left[\begin{array}{l} \text{LOADMDL sel} \\ \text{LOADALL} \\ \text{CANCEL} \end{array} \right]$	LOADGO
------------------	--	--	--	--------	---	--------

入力装置を選択する。

SELDEV sel : 入力装置を選択する。

入力メディアの何番目のファイルのデータをロードするのかを指定する。

CSTNO no : ファイル番号を指定する。(既定値: 1)
 CSTLASTFL : 最後のファイルをロードする。
 CSTALLFL : 全ファイルをロードする。

格納先ディレクトリを指定する。

MDLMGNO sel : 管理番号の一覧を表示し選択する。
 MDLFILE dir : ディレクトリ名を入力する。

ファイルを上書きするかどうか指定する。

WONN : 同じファイルが格納先にあっても上書きする。
 WOFF : 同じファイルが格納先にある場合は、ロードしない。(既定値)

ファイル名の一覧を表示する

LISTGO : 入力メディアのファイル名の一覧を表示する。

ロードするファイル名を指定する。

LOADMDL sel : ロードするファイルを選択する。
 LOADALL : 全ファイルをロードする。
 CANCEL : 選択を取り消す。

ロードを開始する。

LOADGO : ロードを開始する。

5.3.16 管理ファイル作成

● 機能

モデル管理ファイルを新規作成する。

モデル管理ファイルには以下の情報が登録される。

- | | | |
|--|---|---|
| <ul style="list-style-type: none"> モデル名リスト サブモデル参照リスト シンボル参照リスト | } | モデルファイル、モデル情報ファイルから |
| <ul style="list-style-type: none"> 親子インデックス 子供名のリスト シンボル名リスト A P G ファイル名リスト 旧モデル名リスト 旧シンボル名リスト 旧 A P G ファイル名リスト | } | シンボルファイルから
A P G ファイルから
旧モデルファイルから
旧シンボルファイルから
旧 A P G ファイルから |

メニュー表示	コマンド	機能
管理ファイル	CREMANAGE	

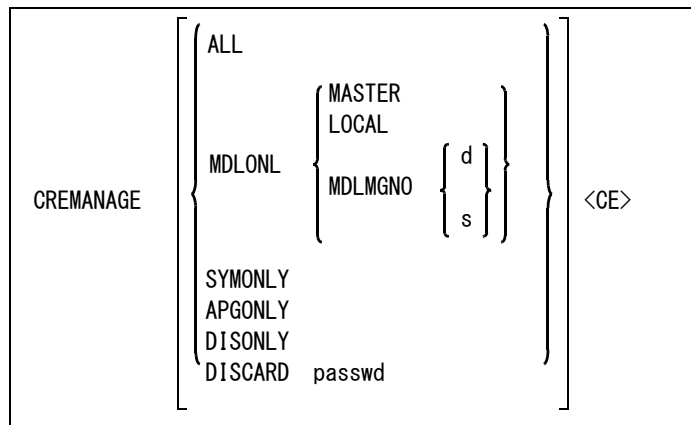
メニュー表示	コマンド	機能
全部	ALL	管理ファイルを新規作成する。
モデル	MDLONLY	モデルの部分について管理ファイルを作成する。
マスター	MASTER	モデルのマスターの管理ファイルを新規作成する。
ローカル	LOCAL	モデルのローカルの管理ファイルを新規作成する。
管理番号	MDLMGNO	モデルの指定した管理番号の管理ファイルを再作成する。
シンボル	SYMONLY	シンボルの管理ファイルを新規作成する。
A P G	APGONLY	A P G の管理ファイルを新規作成する。
廃棄モデル	DISONLY	廃棄モデルの管理ファイルを新規作成する。
パスワード	DISCARD	ACADMMDM.SET に設定するパスワードを求める。

5.3.16.1 モデル管理ファイルを新規作成する

【メニュー】

〔モデル検索〕 → 〔管理ファイル〕

【構文】



作成する管理ファイルをつぎのいずれか選択する。

- ALL : モデル、シンボル、A P G、廃棄モデルファイルの管理ファイルを新規作成する。
モデル検索機能を使用するにあたり、必ず 1 度は行うこと。
- MDLONLY : モデルについてのみ管理ファイルを作成する。
- SYMONLY : シンボルについてのみ管理ファイルを作成する。シンボルに関する管理ファイルを新規作成する。
- APGONLY : A P G ファイルについてのみ管理ファイルを作成する。A P G ファイルに関する管理ファイルを新規作成する。
- DISONLY : 廃棄モデルファイルについてのみ管理ファイルを作成する。廃棄モデルに関する管理ファイルを新規作成する。

モデルのみ (MDLONLY) を選択した場合、作成するモデルの管理ファイルをつぎのいずれかを選択する。

- MASTER : モデルのマスター管理ファイルを作成する。モデルのマスターに関する管理ファイルを新規作成する。
- LOCAL : モデルのローカル管理ファイルを作成する。モデルのローカルに関する管理ファイルを新規作成する。
- MDLMGNO : モデルの指定した管理番号の管理ファイルを作成する。モデルの指定された管理番号に関する部分のみ管理ファイルを再作成する。

モデルの管理番号をつぎのいずれかで指示する。

- d : 一覧表の管理番号をピックアップする。
- s : 一覧表の管理番号を入力する。

コマンドを実行する。

- <CE> : モデル管理ファイルが作成される。

注) 指定された部分以外の管理ファイルについては、作成前の状態を保っています。
既にモデル管理ファイルがあるときにモデル、シンボル、APG ファイルを保存すると、それぞれ管理ファイルに追加登録されます。

パスワードを求める。

- DISCARD password
- password : 意味のある文字列。
結果として意味不明の文字列が表示される。
この文字列を ACADMDM.SET の #PASSWORD# に設定する。

5.3.17 検索結果出力ファイルの形式

E モデル検索では、ファイル出力指定を ONN にすると、ACAD.SET のキーワード #MDLINF:LIST# で定義したディレクトリに以下のファイルが作成されます。

1. MDLINF_MODEL_INF.MLS (モデルファイルの情報)
2. MDLINF_MODEL_TITLE.MLS (モデルタイトルの一覧)
3. MDLINF_MODEL_LIST.MLS (検索後に検出されたモデル名の一覧)
4. MDLINF_REF_LIST.MLS (親子関係の検索後に検出されたモデル名の一覧)

各ファイルは以下のような形式になっています。

1. モデルファイルの情報

【出力フォーマット】		
Model Name	: model_name	モデル名
Model Path Name	: model_full_pathname	モデルのフルパス名
Management Number	: nnnn	管理番号
Read Protection	: n	読み込み属性
Write Protection	: n	書き込み属性
Owner Name	: n	所有者名設定の有無
Creation Date	: yyyy/mm/dd/ hh:mm:ss	作成日
Updating Date	: yyyy/mm/dd/ hh:mm:ss	更新日
Old Model	: n	旧モデルの有無
Casset Name	: xxxxxxxxxxxxxxxxxxxxxx	カセット名

2. モデルタイトルの一覧

【出力フォーマット】 <pre> title_no title ... title_no title </pre>	モデルタイトルの項目番号と内容
--	-----------------

設定されている項目番号についてのみ出力される。

3. 検索で検出されたモデル名の一覧

【出力フォーマット】 <pre> mdlcount models seq_no mdl_full_pathname ... seq_no mdl_full_pathname </pre>	モデル名の総数 順序番号とモデルのフルパス名
--	---------------------------

4. 親子関係の検索で検出されたモデル名の一覧

【出力フォーマット】 <pre> mdlcount Reference seq_no mdl_name ... seq_no mdl_name </pre>	モデル名の総数 順序番号とモデル名
---	----------------------

5.3.18 マクロ作成例

以下はモデル検索のマクロ例です。マクロ作成の参考にしてください。

注) このマクロは、サブウインドウ表示モードでの例です。

(1) 指定したサブモデルを参照しているモデルを調べるマクロ

```

/*
*/
echo "モデル名を入力" ;
mname = gettxt( 1 ) ;
if ( vtype(mname) != 4 ) { exit ; }
/*
*/
MDLDISP          /* モデル検索 */
D_OFF            /* 画面表示をしない */
ONN              /* モデルタイトルをファイルに出力する */
MDLNAME [name]   /* モデル名を指定 */
LISTGO           /* 検索開始 */
INFDISP          /* モデルタイトルをレジスターにセット */
<CE>
if ( vtype(#MMGMDLCNT) != 3 ) {
    echo "モデルが存在しません" ;
    exit ; }
/* */
if ( #MMGMDLCNT != 1 ) {
    echo "モデルが複数存在します" ;
    exit ; }
/* */
REFLIST          /* 親子関係検索 */
ONN              /* 親のモデル名をファイルに出力する */
XREFSUB [#MMGMDLNAME] /* サブモデル名を指定（親検索） */
<CE>
/* */
if ( vtype(#MMGMDLCNT) != 3 ) {
    echo "親モデルが存在しません" ;
    exit ; }
/* */
echo " 正常終了" ;
/* */

```

(2) モデルの書き込み属性を書き込み不可にするマクロ

```

/*
*/
echo " モデル名を入力 (ワイルドカード)" ;
mname = gettxt( 1 ) ;
if ( vtype(mname) != 4 ) { exit ; }
/*
*/
MDLDISP                                /* モデル検索                                */
MDLMGNO 1                               /* 管理番号を指定                                */
MDLNAME [mname]                         /* モデル名を指定                                */
MDLTITLE TTLSET 10 " 名前 "            /* 作成者名を指定                                */
LISTGO                                  /* 検索開始                                    */
if ( vtype(#MMGMDLCNT) != 3 ) {
    echo " モデルが存在しません " ;
    exit ; }
/* */
nmdl = #MMGMDLCNT ;
for ( i = 1 ; i <= nmdl ; i = i+1 ) {
    INFDISP [ i ]                        /* モデル情報を取り出す                        */
    write = #MMGWRITE ;                 /* 書き込み属性を保存                        */
    iowner = #MMGOWNER ;                /* 所有者名の有無を保存                      */
    CONTINUE                            /* 一覧表示に戻る                          */
    if ( iowner == 0 ) {
        if ( iwrite == 0 ) { WRITEONOFF [i] }
    } else {
        if ( iwrite == 0 ) { WRITEONOFF [i] "owner_name" }
    }
}
/* */
echo " 正常終了 " ;
/* */

```

第 6 章 I G E S

- 概要

IGES (initial graphics exchange specification) ファイルを介して、他の CAD/CAM システムと図形データの変換ができます。

Advance CAD の IGES は、U.S.DEPARTMENT OF COMMERCE National Bureau of Standard の IGES (Initial Graphics Exchange Specification) Version 5.0 に準拠しています。

6.1 コマンド

● コマンド一覧

コマンド名	機能
IGES / READ	IGES ファイルを読み込む
IGES / WRITE	IGES ファイルを書き込む
IGES / PUNCH	ポンチ絵を作成

6.1.1 IGES ファイルを読み込む

【メニュー】

〔データ変換1〕 → IGES〔読み込み〕

【構文】

```
IGES/READ filename s1 s2 s3 s4 <CE>
```

IGES ファイル名を指示する。

filename : ファイル名にファイル拡張子を与えなければ、デフォルト拡張子 .IGF と見なす。指定したファイルが存在しなければ再度ファイル名入力に戻る。

確認モードを指示する。

s : IGES ファイル読み込みの進捗状況を表示するかどうか指定する。
 0 または <CE> では何も表示しない。
 1 であれば処理している Directory Entry (DE) ポインタを表示する。
 表示に時間がかかるので多少遅くなる。DE ポインタとは、そのエンティティの Directory Entry Section での番号である。

投影モードを指示する。

s2 : IGES ファイル中の 3 次元図形を 2 次元投影するかどうかを指定する。
 0 または <CE> では XY 平面だけに投影する。
 1 であれば drawing entity に配置された view entity 全てに対して投影する。
 このオプションは、3 次元 CAD システムで作成された IGES ファイルを読み込むとき、ひとつの 3 次元形状を複数のビューに投影した形状を得たいときに指定する。2 次元 CAD システムで作成した IGES ファイル、3 次元 CAD システムで作成した IGES ファイルでも 2 次元の図形だけのときは、指定しないこと。

漢字形式を指示する。

s3 : 漢字の形式を番号 (0 ~ 4) を入力する。
 <CE> と入力すると、漢字形式 0 を指定したことになる。

シンボル登録を指示する。

s4 : IGES subfigure Definition (#308) をシンボルとして登録するときの方法を指示する。
 0 または <CE> を指示すると、同名のシンボルファイルがないときだけシンボル登録する。
 1 を指示すると、既に同名のシンボルファイルがあるときにはそれを削除し、新しくシンボル登録を行う。
 2 を指示すると、Subfigure を分解して配置する。

IGES/READ のメッセージウインドウ

IGES ファイル名	
(1)	(2)
(3)	

- (1) は、IGES ファイル読み込み時に処理しているエンティティの DE ポインタを表示。
 (2) は、IGES ファイルから Advance CAD アイテムを作成するとき、処理中のエンティティの DE ポインタを表示。
 (3) は、エラーの時発生したエンティティの DE ポインタを表示。
 (1), (2) は、確認モードが 1 (DE 番号表示) のときだけ表示される。

例

ファイル名を入力 EXAMPLE
 確認モード 0 = なし, 1 = DE 番号表示 <CE>
 ビュー投影 0 = なし, 1 = 投影する <CE>
 漢字形式 (0, 1, 2, 3) <CE>
 シンボル 0 = なければ登録, 1 = 全て再登録 2 = 分解配置する

- 注) 自由曲線の変換を通過点のみで近似をしていましたが余りにも図形が違ってしまう場合があります。通過点の間に何点か補間点を挿入し近似して変換することができます。

ACAD.SET に次のキーワードを設定してください。

#IGES:NOEXPD#

このキーワードがあると補間点を挿入しないで変換します。

#IGES:EXPD# "n"

このキーワードがあると補間点を挿入し分割して変換します。

n : 分割数 (1 ~ 40)
 最長セグメントの分割数。

これらのキーワードがない場合各セグメントを 20 分割して変換します。

6.1.2 アクティブモデルを IGES ファイルに書き込む

【メニュー】

〔データ変換 1〕 → IGES 〔書込み〕

【構文】

IGES/WRITE filename t1 t2 t3	$\left\{ \begin{array}{l} \text{ALL} \\ \text{CUR} \\ \text{USEACT} \\ \left\{ \begin{array}{l} \text{drawnam} \\ \text{s1} \end{array} \right\} \text{s2} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{PICMTXOFF} \\ \text{PICMTXONN} \end{array} \right\} \text{s3 s4 s5 <CE>}$
------------------------------	--	---

IGES ファイル名を指示する。

filename : ファイル名にファイル拡張子を与えなければ、デフォルト拡張子 .IGF が付く。
 入力したファイル名と同じ名前のファイルが既に存在すれば、「重ね書き Yes (Y)
 No (N) ? 」というメッセージが表示される。
 'Y' を入力すると次に進む。それ以外の入力では再度ファイル名入力に戻る。

表題を入力する。

t1 : IGES ファイルの Start section に記入する表題を文字列で入力する。
 80 文字以内で漢字は使用できない。省略するときは <CE> だけを入力する。

作成者名と組織名を入力する。

t2 および t3 : 作成者名 (t2)、組織名 (t3) を文字列で入力する。それぞれ 80 文字以内で漢字
 は使用できない。省略するときは <CE> だけを入力する。
 ここで入力した作成者名、組織名が、IGES ファイルの Global section のパラメー
 タとして組み込まれる。

IGES ファイルに出力するアイテムをつぎのいずれかで指示する。

ALL : モデル全体を IGES ファイルに出力する。
 CUR : カレント ピクチャのアイテムだけを出力する。
 USEACT : アクティブリスト中のアイテムだけを出力する。
 drawname または s1 : ドローイングレイアウトを出力するとき指定する。
 ドローイングレイアウトはドローイングレイアウト名またはドローイングレイアウト
 番号のどちらかを入力する。
 ドローイング出力のときは、各々のピクチャのアイテムタイプ、クラス、レビジョ
 ン、線種、線幅のディスプレイマスクを参照し、表示可能状態のアイテムだけを出力
 する。
 ドローイングレイアウトで用いている図面枠シンボルも出力する。
 図面枠シンボル中のキーワードを含むジェネラルノートは、キーワードを適切な文
 字列に変換した後出力する。
 図面枠シンボル中の画面表示だけのデータ（これはレビジョン番号で区分けされて
 いる）は、出力しない。
 PICMTXOFF : ピクチャのローテーションマトリックスを考慮しないで出力する。
 PICMTXONN : ピクチャのローテーションマトリックスを考慮して出力する。
 ピクチャ指定で出力する場合のみ指定可能。
 正面図 (X - Z 平面) のピクチャに描かれている図形は、3次元座標の X - Z
 に変換する。変換できるアイテムは、点・直線・円・円弧・自由曲線・ストリン
 グ。複合アイテム・シンボル・サブモデル内の変換できるアイテムも同様。
 s2 : ドローイングレイアウト出力を指定したとき、ドローイング座標に変換する (=1)
 しないか (=0) を指示する。そのドローイングで参照しているアイテムだけを選
 択して出力する。
 0 を入力すると、ドローイング座標に変換しない。出力はつぎのようになる。
 ・ ドローイングレイアウトは、IGES ドローイングエンティティとして出力する。
 ・ ドローイングに配置してあるピクチャは、IGES ビューエンティティとして出力
 する。
 ・ ドローイングレイアウトに用いている図面枠シンボルも出力する。
 図面枠シンボルから生成した IGES エンティティは、IGES ドローイングエンティ
 ティが直接参照するエンティティとして出力する。
 1 を入力すると、ドローイング座標に変換する。このモードでは座標はドローイン
 グ座標に変換して出力する。IGES ドローイングエンティティとビューエンティ
 ティを処理しない IGES ポストプロセッサで IGES ファイルを読み込むと、ドロー
 イングレイアウトに配置したビューのエンティティがすべて重なってしまう。これ
 を避けたいとき使用する。出力はつぎのようになる。
 ・ ドローイングレイアウトにピクチャを配置した状態をそのまま IGES 出力する。
 ・ ピクチャのゾーンからはずれたアイテムは出力しない。しかし一部だけゾーンに
 入っているアイテムは出力する。アイテムのピクチャゾーンに入っている部分だ
 けを出力することはできない。
 ・ ドローイングレイアウトに用いている図面枠シンボルも出力する。
 <CE> : 上記のいずれも入力しないで省略する場合で、以下の規則がある。
 ・ これ以前に ALL, CUR, または USEACT を指示していてここでは <CE> と入力
 したときは、それ以前に入力した ALL, CUR, または USEACT が有効になる。

- これ以前に ALL, CUR, または USEACT の指示がなく <CE> と入力したときは、モデル全体 (ALL) を指示したと同じになる。

漢字形式を指示する。

s3 : 漢字の形式を番号 (0 ~ 4) を入力する。<CE> と入力すると、漢字形式 0 になる。

円錐曲線を表わす自由曲線の出力形式

s4 : Advance CAD では円錐曲線は自由曲線で表現している。円錐曲線を表わす自由曲線を IGES 出力するとき、IGES parametric spline curve entity (112) とするか、conic arc entity (104) とするか選択できる。
0 または <CE> を入力すると、自由曲線のまま出力する。
1 を入力すると、円錐曲線に変換して出力する。

JAMA-IS V1.0 への変換スイッチ

s5 : JAMA-IS V1.0 へ変換するかどうかを指示する。
0 または <CE> を入力すると、従来どおりの IGES を出力する。
1 を入力すると、自動車工業会 IGES SUBSET V1.0 (JAMA-IS V1.0) に沿った内容で IGES ファイルを出力する。

s5 で、JAMA-IS で出力した場合と通常との出力の違いは下記のとおり。

- (1) JAMA-IS 項目番号 2.4.1 スタート部の 1 ライン目に JAMA-IS 使用の判別情報を入れる。スタート部の 1 ライン目に JAMA-IS 使用の判別情報がはいる。
- (2) JAMA-IS 項目番号 2.4.3.4 ディレクトリ部のフィールド No.4 線種には、0 ~ 5 の値をセットする。6 ~ 63 は 1 にセットする。
- (3) JAMA-IS 項目番号 2.4.3.13 Type 314 (color Definition Entity) を出力しない。
ディレクトリ部 No.13 色に、IGES で定義している色のうち近いものを選び、その番号をセットする。
- (4) JAMA-IS 項目番号 2.4.3.19 ディレクトリ部のフィールド No.19 サブスプリクトは Type126, 128 のみ使用し、それ以外の Entity では使用しない。0 をセットする。なお、Advance CAD は Type126, 128 を出力しない。
- (5) JAMA-IS 項目番号 3, 7, 9 Type 116 Point Entity のディレクトリ部のフィールド No. 4 線種には、常に 0 をセットする。

例

ファイル名を入力	EXAMPLE
タイトル	<CE>
作成者	TNI
組織	CTC
出力ドローイング名, ALL, CUR または USEACT	TEST
漢字形式 (0, 1, 2, 3)	<CE>
自由曲線→円錐曲線変換 0= しない, 1= する	<CE>
JAMA-IS 変換 0= しない, 1= する	<CE>
設定完了ならば <CE> を入力	<CE>

6.1.3 ポンチ絵を作成する

図面配置の状態を指定されたピクチャに 1 / 1 の倍率で再現する。

【メニュー】

〔データ変換 1〕 → IGES 〔ポンチ絵〕

【構文】

IGES/PUNCH	PAGE page	OUTPIC pic	<table><tr><td>NEW</td></tr><tr><td>ADD</td></tr></table>	NEW	ADD	<CE>
NEW						
ADD						

PAGE page : 図面配置のページ番号を指定
OUTPIC pic : 図面配置状態を再現するピクチャ番号を指定
NEW : 元図を残さない
ADD : 元図を残す（初期値）
<CE> : 処理を開始する

6.2 Advance CAD – IGES

6.2.1 IGES ファイルの読み込み

- データ フォーム

Advance CAD IGES プロセッサは固定長 ASCII フォームの IGES ファイルだけを処理します。圧縮 ASCII フォーム、バイナリ フォームの IGES ファイルを読み込むには、あらかじめ固定長 ASCII フォームに変換する必要があります。

- 読み込み可能な IGES エンティティ

つぎの IGES エンティティは、つぎの Advance CAD アイテムタイプに変換されます。下記以外のエンティティは無視されます。

IGES Entity	Advance CAD アイテム
point	点
line	線分
circular arc	円／円弧
conic arc	自由曲線
parametric spline curve	自由曲線
linear spline (ctype=1)	ストリングアイテム
copious data (ORM# 1, 2, 3, 11, 12, 13)	ストリングアイテム
composit curve	ストリングアイテム
offset curve	自由曲線
general note	ジェネラルテキスト (ジェネラルノート)
general label	ジェネラルテキスト (ジェネラルラベル)
general symbol	ジェネラルテキスト (リファレンスノート、ラベル)
point dimension	ジェネラルテキスト (リファレンスノート)
section (copious data FORM# 31 ~ 38)	複合アイテム
centerline(copious data FORM# 20, 21)	一点鎖線の (複数の) 線分
Simple closed area (copious data FORM# 63)	ストリング
linear dimension	長さ寸法、累進寸法
ordinate dimension	オーディネイト寸法
angular dimension	角度寸法

IGES Entity	Advance CAD アイテム
diameter dimension	直径寸法
radius dimension	半径寸法
sectioned area	ハッチング
subfiger definition	シンボルファイル
singular subfigure instance	シンボルインスタンス
drawing	ドローイングレイアウト

● 次の制限数を越える IGES ファイルは Advance CAD に読み込めません。

ビューの数 : 253 以内

エンティティの数 : 262143 以内

エンティティ数の多い IGES ファイルを読み込んだ時、モデルのアイテム数の制限を超えた時は読み込み失敗となります。モデルのアイテム数の上限はコンフィギュレーションファイル ACAD.SET で設定されたアイテム数の上限 (MAXID) に依存します。

IGES ディレクトリエン트리セクションのシーケンス番号が D524286 以内の IGES ファイルが読み込み可能です。IGES ファイルのターミネイトセクションを見ると、各セクションの行数がわかります。ターミネイトセクションは、UNIX の tail コマンドで IGES ファイルの最後の行を表示させると見ることができます。

```
prompt% tail -l name.IGF
s      2G      3D      302P      233      T      1
```

この例では、各セクションの行数は以下のとおりです。

```
Start セクション      : 2
Global セクション     : 3
Directory Entry セクション : 302
Parameter Data セクション : 233
```

● IGES view entity は、Advance CAD のピクチャに対応します。

複数のビューを含むときは、それぞれひとつのピクチャとなります。最高 256 のピクチャを作ります。view entity が IGES ファイル中に現われた順にピクチャ 1, 2, ... と対応づけます。

IGES entity の directory entry の view pointer が、ある view を指示しているとき、そのエンティティはその view に対応したピクチャに作られます。

IGES entity の directory entry の view pointer が 0 であると、ピクチャ番号が決まりません。また drawing entity が直接参照している annotation entity がありますが、Advance CAD ではドローイングレイアウトにはアイテムを作れませんから、これも表示するピクチャ番号を決めてやる必要があります。

これらの場合は、最も若い、使用していないピクチャを使います。たとえば IGES ファイル中に 3 つの view entity があったときは、4 番目のピクチャを使います。

view entity がひとつもないときは、1 番目のピクチャを使います。

IGES View entity が name property entity (FORM#15) を持つとき、その名前をピクチャ名とします。ただし、先頭の 256 文字が有効です。ビューのスケールをピクチャスケールに設定します。

● アイテム属性

ピクチャ	: VIEW 番号が 1 ~ 256 以外の場合はピクチャ 1 とします。
クラス	: LEVEL 番号が 1 ~ 256 以外の場合はクラス 1 とします。
線幅	: LINE WEIGHT 番号が 1 ~ 15 以外の場合は線太さ 1 とします。
線種	: LINE FONT PATTERN 番号が 1 ~ 63 以外の場合は線種 1 とします。
カラー	: レビジョン番号 1 ~ 15 とします。

● IGES エンティティの順序

IGES の general label、general symbol や dimension entity は、いくつかのエンティティで構成されます。他のエンティティを構成する要素となるエンティティは、Subordinate Entity と呼ばれています。Advance CAD - IGES では SubordinateEntity はそれを構成する独立なエンティティ (Independent Entity) の直前に出力します。そしてこの場合、subordinate switch は physically dependent (01) とします。

たとえば Linear dimension (Entity type #266) は、

- ひとつの General note (Entity type #212)
- 2つの Leader (Entity type #214)
- 2つの Witness line (Entity type #106)

で構成されます。これらは、Linear dimension の直前に出力します。

しかし、他の CAD システムで作成した IGES ファイルは必ずしもそうではありません。IGES ファイル読み込みのときは、エンティティの出現順序には依存しません。

● Parametric spline curve

最大セグメント数は 512 です。

● conic arc (円錐曲線)

Advance CAD では、conic arc はそれを近似する自由曲線として取込みます。

● 漢字テキストも処理します。

● ordinate dimension entity

寸法値テキストは、寸法値をあらわす数字のテキストでなければなりません。他の英文字で置き換えてあると、寸法は作成されません。

● singular subfiger instance entity

subfiger definition から Advance CAD のシンボルを作り、シンボルを配置します。同じ名前のシンボルがすでにある場合はシンボル作成は行ないません。

● drawing entity

drawing entity から Advance CAD のドローイングレイアウトを作成します。

ドローイングレイアウト名は 'IGESxx' です。xx はドローイングページ番号 (01 ~ 256) になります。drawing entity 中の参照 view entity は、ドローイングレイアウトにピクチャを配置することになります。多くの 2 次元 CAD では drawing entity では 1 つも view entity を持たず、annotation entity だけを持つものもあります。このときは、空のドローイングレイアウトができます。IGES Drawing entity が name property entity (FORM#15) を持つとき、その名前をドローイングレイアウト名とします。ただし、先頭の 256 文字が有効です。

● drawing entity で直接参照している annotation entity も読み込みます。

Advance CAD のドローイングレイアウトは、ピクチャの配置だけが可能で、そこに注記や寸法を記入することはできません。したがって drawing entity で直接参照している annotation

entity は、ドローイングレイアウトではなく、通常のピクチャに表示します。「IGES view entity は、Advance CAD のピクチャに対応します」の項を参照してください。

● 3次元図形の投影

ビュー投影の指示がないときは、すべて XY 平面に投影します。

ビュー投影の指示があったときは、drawing entity に配置された view entity すべてに対して投影します。

たとえば drawign entity が 4 つの view entity を参照しているとします。その 4 つの view が XY 平面 (TOP)、XZ 平面 (FRNT)、YZ 平面 (SIDE)、および ISO ビューを定義しているとします。このとき 1 つの 3 次元図形は 4 つのビューに投影されます。つまり、Advance CAD では 4 つのビューに対応する各々のピクチャに現われることになり、1 つの 3 次元図形が 4 つの 2 次元図形に変換されます。

投影する 3 次元図形はつぎのものです。

- IGES エンティティ番号が 100 番台

Point
Line
Circular Arc
Conic Arc
Parametric Spline Curve
Composit Curve
Copious data (FORM=1, 2, 3, 11, 12, 13)

- エンティティのビューポイントが 0 であること。エンティティのビューポイントがビューエンティティを指すとき、そのエンティティはそのビューにだけ表示されるものなので、ビューポイントによって指定されたビューにだけ、投影します。

- View Visible (associativity entity FORM#3)

IGES Directory entry の 6 番目のパラメータ (View) が Views visible associativity instance を指すときの処理

3 次元 CAD システムでは、ある図形をすべてのビューに表示するのではなく、2 つ以上の特定のビューにだけ表示させる場合があります。たとえば、立方体をアイソメで見たとき、隠れて見えない稜線を非表示にしたとします。その稜線は、アイソメビューには表われませんが、他のビューでは表示されます。このように、1 つの図形を特定のビューにだけ表示させるとき、IGES では Viewsvisible associatibity instance を使用します。

Advance CAD では、Views visible associativity を指す Geometric entity があった場合は、それを指定された複数のビューに表示します。これは、IGES/READ コマンドのビュー投影オプションには関係なく行います。

ビュー投影する、ビュー投影しないのどちらを選んでも、Views visible associativity で指定したビューに表示します。

● 処理方法

IGES ファイルをシーケンシャルに読み込んで処理をするのは制約が大きいので、IGES/READ の処理は 2 段階に分けてあります。

- (1) 最初に IGES ファイルを読み込みます。
この時点で Advance CAD で処理できないエンティティ、たとえば曲面などは捨ててしまいます。
- (2) 次に Advance CAD アイテムを作成します。

アイテム作成については、つぎのような制限があります。

- 独立しているエンティティだけを処理します。
つまり、エンティティステータスの subordinate switch が independent (00) または logically dependent (02) のものだけを処理します。

subordinate switch がそれ以外のものは、他のエンティティの子どもですので Advance CAD のアイテムにはなりません。

- 非表示アイテムは処理しません。
独立しているアイテムでも、エンティティステータスの blank flag が 0 でないものは処理しません。
- 矢印 (leader entity) と寸法補助線 (copious data entity, form#40)
これらは、独立しているエンティティでも処理しません。IGES プロセッサによってはリファランスラベルを独立したジェネラルノートエンティティ、円弧エンティティ、矢印エンティティの 3 つに分解して出力するものもあります。Advance CAD には矢印だけのアイテムは存在しないので、独立した矢印エンティティは無視します。また、寸法補助線も寸法アイテムが参照すべきエンティティなので、それが独立したエンティティでも無視します。

● 線幅のピッチ

IGES ではグローバルセクションに最大線幅番号とそのときの最大幅を持っています。線幅はつぎのように計算します。

$$\text{線の幅} = \text{線幅番号} \times (\text{最大幅} \div \text{最大線幅})$$

この最大線幅番号は CAD システムによってさまざまです。Advance CAD は 15 ですが、他のシステムでは 3, 5, 8, 10 のものから 10000, 32767 という大きな値のものまであります。

最大線幅番号が小さいシステムの線幅は Advance CAD の線幅と 1 対 1 に対応付けることができるので問題ありません。しかし大きな最大線幅番号を持つシステムの IGES ファイルを読み込んだときは線幅 1 ～ 15 以外はすべて線幅 1 になってしまいます。また線幅 1 ～ 15 もピッチが小さいため線幅が出ません。

たとえば最大線幅番号 32767, 最大幅 32.767 では、線幅ピッチは 0.001 です。線幅番号が 15 でも 0.015 です。これでは線幅が出ません。線幅番号 500 で幅 0.5, 1000 で幅 1 の太さの線になります。また最大値 32767 では太さ 32.767 になりますが通常では 500 ～ 2000 くらいの線幅番号を使うはずで、そこでその IGES ファイル中で使用している線幅の最大値を調べ、それがほぼ Advance CAD の線幅 15 に当たるように対応づけることにします。

● 色

Advance CAD と IGES の色の扱いの違いは、つぎようになります。

- 色の数
IGES で規定している色は 8 色で、番号 1 ～ 8 です。それ以外の色は色定義エンティティを使用して、色の三原色の強度を指定する方法です。したがって IGES では同時に使用できる色の数には制限がありません。Advance CAD では 15 色に制限されています。
- アイテムの色属性
IGES では色エンティティは独立した属性です。しかし Advance CAD では、色はクラス、アイテムタイプ、レビジョンのいずれかと対応を付ける方式であり、色は他の属性に従属しています。
- 色定義テーブル
IGES では色定義エンティティを使って任意の色を定義できます。もしひとつの IGES ファイル中で Advance CAD の上限である 15 色以内の色を使用していたとしても、Advance CAD の色定義テーブルは変更できません。Advance CAD の色定義テーブルは各モデルから独立しており、色定義テーブルを変更すると、すべてのモデルに影響があるからです。

以上を考慮して、つぎのようにします。

- IGES ファイル中の色 (規定色および色定義エンティティで定義した色) は、Advance CAD の 15 色のうちいちばん近い色に割り当てる。
- 色番号はそのままレビジョン番号に設定する。
IGES/READ で読み込んだアイテムのレビジョン番号は、1 ～ 15 の値になります。
- IGES/READ 後
CLR/REV コマンドで色 1 とレビジョン 1、色 2 とレビジョン 2 というように、色割付を行ってください。

● **Associativity Instance(#402, Form#1, 7, 14, 15)**

IGES ファイル中に Associativity Instance(#402, Form#1, 7, 14, 15) があるとき、このエンティティが参照しているエンティティの Subordinate Entity Switch が 02 以外だと、参照されているエンティティを処理しませんでした。02 以外のときも 02 と見なして処理するようにしました。

6.2.2 IGES ファイルの作成

● **IGES に出力可能なアイテム**

つぎの Advance CAD アイテムタイプは、以下の IGES エンティティに変換されます。これ以外のタイプのアイテムは出力されません。

Advance CAD アイテム	IGES エンティティ
ポイント	point
ライン	line
円／円弧	circular arc
自由曲線	parametric spline curve (ctype=3)
円錐曲線を近似している自由曲線	Conic arc
ストリングアイテム	composit curve または copious data (FORM#11)
ジェネラルテキスト (ジェネラルノート)	general note
ジェネラルテキスト (ジェネラルラベル)	general label
ジェネラルテキスト (リファレンスノート、ラベル)	general symbol
ハッチング	section (copious data FORM# 31)
塗潰し	composit curve または copious data (FORM#11)
長さ寸法、累進寸法	linear dimension
角度寸法	angular dimension
直径寸法	diameter dimension
半径寸法	radius dimension
点座標寸法	general label

以下の Advance CAD アイテムは、アイテムに含まれる要素をそれぞれ独立した entity として出力します。

幾何公差
弧長寸法

マークアイテム (溶接記号、面の肌記号、単独のマークアイテム)
 切断線 (テキストアイテム)
 複合アイテム
 APG アイテム
 シンボルインスタンス
 サブモデルインスタンス
 ドローイングレイアウト drawing
 ピクチャ view

- 1つのピクチャを1つの view として出力します。view # にはピクチャ番号 (1-256) が入ります。

ビュー エンティティとともに viewing box を規定する plan entity を出力します。

● カラー番号

Advance CAD では赤・緑・青の3色の強さで色を指定します。赤・緑・青のそれぞれが 0 から 255 の範囲の値を取ります。赤緑青の強さ (RGB) とそれに対応する IGES の色番号を示します。

赤 (R)		緑 (G)		青 (B)		色名	IGES 色 番号
0	+	0	+	0	=	黒 (Black)	(1)
255	+	0	+	0	=	赤 (Red)	(2)
0	+	255	+	0	=	緑 (Green)	(3)
0	+	0	+	255	=	青 (Blue)	(4)
255	+	255	+	0	=	黄色 (Yellow)	(5)
255	+	0	+	255	=	深紅 (Magenta)	(6)
0	+	255	+	255	=	青緑 (Cyan)	(7)
255	+	255	+	255	=	白 (White)	(8)

上記以外の色は IGES には規定されていません。規定外の色は IGES では色定義エンティティ (Color Definition Entity 314) を使って RGB を指定できるようになっています。

色が IGES で規定している8色のどれかであれば、IGES ディレクトリエントリの13番目のフィールド色番号にはその色番号 (1-8) が入ります。

規定外の色であれば、その色を定義する IGES Color Definition Entity を指すポインタ (負の値で示される) が入ります。

IGES の Color Definition Entity 自身のディレクトリエントリの色番号は、IGES 規定の8色のうちの一番似通った色の番号を設定しています。

- 線分だけからなるストリングアイテムは copious data (FORM# 11) で出力します。
- 漢字テキストも処理します。テキストは複数行でもかまいません。
- 寸法値テキストがない寸法アイテムは出力できません。

- テキストに Ymirror, Xmirror の両方を設定したものは no-mirror とします。
- general note entity の文字数
 - ・ 1つのジェネラルノート アイテムに含まれる文字数が 1024 文字以内のアイテムだけが変換されます。
 - ・ 寸法テキストは、64 文字以内
 - ・ 寸法公差テキストは 32 文字以内
- general label entity
 - ・ リーダの数は 16 個以内
 - ・ リーダの点数は 32 点以内
 のものだけ変換されます。
- 部分線種・部分線幅・部分ブランクを行ったセグメントを含むストリングアイテム
IGES コンポジットカーブエンティティとして出力します。非表示の線分・円弧エンティティのブランクングステータスは '01' となります。
- 部分線種・部分線幅・部分ブランクを行ったラインアイテムと円／円弧アイテム
IGES コンポジットカーブとして出力します。非表示の線分・円弧エンティティのブランクングステータスは、'01' となります。
- Hierarchy
IGES 寸法エンティティとコンポジットカーブエンティティの Hierarchy は '01' とします。
- 結合アイテム (複合アイテム・APG アイテム・シンボルインスタンス・サブモデルインスタンス)
それに含まれるアイテムを別々の IGES entity として出力します。たとえば、線分と円弧を含む複合アイテムは、Line entity と Circulararc entity として、それぞれ独立した entity として出力します。
このときメンバアイテムの属性 (クラス、線種、線太さおよびこれらにより決まる色番号) は、モデル定数 (RVP/MODEL) の以下のパラメータを参照して決定します。
 - 結合アイテムの [表示の制御及びカラー] の表示
 - 結合アイテムの [線幅] の表示
 - 結合アイテムの [線種] の表示
 つまり画面に表示されている状態を反映する属性で出力します。
- 通常シンボルアイテムのベーステキスト・ノードテキストは表示されないので、IGES ファイルには出力しません。ただし、ベーステキスト・ノードテキストでも、表示されているものは、General Note entity として出力します。
- IGES Global Section Parameter #20
IGES Global Section Parameter #20 Approximate max:min coordinate value 各 IGES ファイルごとに最大座標の近似値を算出し、設定します。いままでこのパラメータには 10^{12} を出力していました。
- 以下のアイテムも IGES ファイルに出力できます。
 - ・ 幾何公差
 - ・ マークアイテム (溶接記号、面の肌記号、単独のマークアイテム)
 - ・ 切断線 (テキストアイテム)

これらのアイテムは複数の IGES entity に分解して IGES ファイルに出力されます。

文字列 → General Note entity
 引出線 → Leader entity
 マーク → マークを分解して折れ線 (Copious data entity) や円／円弧 (Circular arc entity) で出力

● Global section #17 Size of maximum line width in units

最大線幅 = 15*2*(ラインウェイトピッチ)

● Advance CAD Version 5.02 まで

Advance CAD version 5.02 までは、Advance CAD の IGES ファイルは図形要素の Directory Entry Status Field が '01' (表記) となっていました。IGES プロセッサにもありますが、これが '00' (図形) でないと読み込めないことがありました。

これでは Entity use flag の '00', '01' を判別している IGES プロセッサでは不都合なので、IGES version 5 の文書に従うように変更しました。

IGES version 5.0 (September 1990) 文書によると、Directory Entry の Status field は次のようになっています。

- '*' は、このエンティティに対しては適用しないことを表わす。'00' とする。
- '??' はそのフィールドで利用できる適切な値を設定することを表わす。
- 数字 ('00' や '02') が記入してあるところはその通りにしなければならない。

Status number field

- 1-2 Blanked status
 - 00 Visible
 - 01 Blanked
- 3-4 Subordinate entity switch
 - 00 Independent
 - 01 Physically dependent
 - 02 Logically dependent
 - 03 Both '01' and '02'
- 5-6 Entity use flag
 - 00 Geometry
 - 01 Annotation
 - 02 Definition
 - 03 Other
 - 04 Logical/Oisutuibak
 - 05 2D parametric
- 7-8 Hierarchy
 - 00 Global top down
 - 01 Global defer
 - 02 Use hierarchy property

Entity	Status number
100 Circular arc	?????*
102 Composite curve	?????*
104 Conic arc	?????*
106 Copious data form=1-3, 11-13, 63	?????*
form=20-21, 31-38, 40	???01*
108 Plane form=0	?????*
110 Line	?????*
112 Parametric spline	?????*
116 Point	???????
124 Transformation matrix	*****
202 Angular dimension	???01??
206 Diameter dimension	???01??
208 Flag note	???01??

210 General label	????01??
212 General note	????01**
214 Leader (arrow)	????01**
216 Linear dimension	????01??
218 Ordinate dimension	????01??
222 Radius dimension	????01??
228 General symbol	????01??
230 Sectioned Area	????01??
314 Color definition	**0002**
404 Drawing	**0001**
406 Property	**??*??*
410 View	**??01**

6.2.3 読み込み／書き込み可能な IGES エンティティ

IGES Entity		Advance CAD アイテム
Point	↔	点
Line	↔	線分
Circular arc	↔	円／円弧
Conic arc	→	自由曲線
Parametric spline curve (ctype=1)	→	ストリングアイテム
Parametric spline curve (ctype=3)	↔	自由曲線
Parametric spline curve (ctype=6)	→	自由曲線
Copious data (FORM# 1, 2, 3, 11, 12, 13)	↔	ストリングアイテム
Composit curve	↔	ストリングアイテム
General note	↔	ジェネラルテキスト (ジェネラルノート)
General label	↔	ジェネラルテキスト (ジェネラルラベル)
General symbol	↔	ジェネラルテキスト (リファランスノート、ラベル)
Point dimension	↔	ジェネラルテキスト (リファランスノート)
Section (copious data FORM# 31)	←	ハッチング
Section (copious data FORM# 31 ~ 38)	→	複合アイテム
Composit cueve (copious data FORM#11)	←	塗潰しアイテム
Centerline(copious data FORM# 20, 21)	→	一点鎖線の (複数の) 線分
Simple closed area (copious data FORM# 63)	→	ストリング
Sectioned Area	→	ストリング (境界だけ)
(pattern code 0, 6 ~ 12, 17, 19)	→	ストリング (境界)、ハッチング
Linear dimension	↔	長さ寸法、累進寸法
Ordinate dimension	← →	オーディネイト寸法
Angular dimension	↔	角度寸法
Diameter dimension	← →	直径寸法
Radius dimension	↔	半径寸法

IGES Entity		Advance CAD アイテム
General label	←	点座標寸法
Singular Subfigure instance	→	シンボルインスタンス
Subfigure definition	→	シンボルファイル
Drawing entity	←→	ドローイングレイアウト
	←	複合アイテム
	←	幾何公差
	←	マークアイテム (溶接記号、面の肌記号、単独のマークアイテム)
分解	←	切断線 (テキストアイテム)
	←	弧長寸法
	←	APG アイテム
	←	シンボルインスタンス
	←	サブモデルインスタンス

6.2.4 日本語文字列の取り扱い

IGES Version 5 ではジェネラルノートのテキストフォント 2001 番として JIS 漢字表 (JIS C6226-1983) が追加され、日本語の文字列も組み込めるようになりました。これを漢字形式 0 とします。

IGES Version 4 以前では日本語の規定がないため、日本語を含む文字列の処理については各々の IGES プロセッサごとに独自の方式で対応していました。

Advance CAD でも以下の 4 つの方式で日本語文字列の処理を行なっていました。これらはそのまま残してあります。以下ではこのような非標準の処理について説明します。

いずれも IGES General note entity の Parameter Data を変更拡張しており、漢字を含む文字列を処理できない IGES プロセッサに対しても支障のない形で出力することを前提としています。

- IGES Entity type
General note entity (entity type #212)
- IGES Directory Entry
General note entity の規則どおりで変更はない。
- IGES Parameter Data for General note
日本語処理のため特別な規則を導入している。

● 漢字形式 1

(1) 漢字表現

漢字 1 文字は JIS 区番号と点番号を並べた 4 桁の整数で表現する。

漢字コード = 区番号 × 100 + 点番号

漢字コードの例

ひらがな 'あ' '0402' (4 区 2 点)

漢字 '幻' '2424' (2 4 区 2 4 点)

- (2) 漢字コードを並べるときは、' ' で区切り、漢字コードの並び全体を ' < ' と ' > ' でくくる。' < ' から ' > ' の間に現われる空白はすべて無視する。また漢字コードの先頭の ' 0 ' は無視する。
- (3) 漢字と ASCII 文字との混合も可能。ただし、ASCII の ' < ' は、漢字列の始まりの記号と区別するために、' << ' のように 2 文字にする。読み込み時は 1 文字の ' < ' と解釈する。

漢字を含む文字列の出力例

```
18H<02033, 2 790>IGES ' 漢字 IGES' (先頭の ' 0 ' やスペースはすべて無視)
10H<<KG/M**3> ' <KG/M**3>' (漢字は含まない)
```

● 漢字形式 2

- (1) Parameter data の Font code (FC)
文字フォント番号 90 のときは漢字文字列を表わす。漢字と ASCII 文字は混合できない。
- (2) Parameter data の Number of character (NC)
文字数は漢字の数で Parameter data の text string (TEXT) の実際の文字数とは異なる。これは IGES 仕様に反するので注意のこと。
- (3) 漢字表現
漢字 1 文字は JIS 区番号と点番号を並べた 4 桁の整数で表現する。
漢字コード = 区番号 × 100 + 点番号
漢字コードの例
ひらがな ' あ ' ' 0402 ' (4 区 2 点)
漢字 ' 幻 ' ' 2424 ' (2 4 区 2 4 点)
- (4) 漢字コードを並べるときは ' , ' で区切る。

漢字文字列の出力例

```
9H2033, 2790 (これは ' 漢字 ' を表現している)
```

● 漢字形式 3

- (1) 漢字表現
漢字 1 文字は ¥ (バックスラッシュ) とそれに続く 4 文字の合計 5 文字で表現する。
最初の ¥ (バックスラッシュ) は、次の 4 文字が漢字コードであることを示す文字で、次の 4 文字はシフト JIS コードを 16 進表現したものである。
漢字コードの例
かたかな ' シ ' ' ¥8356 ' (5 区 2 3 点)
漢字 ' 漢 ' ' ¥8ABF ' (2 0 区 3 3 点)
かたかな
- (2) 漢字と ASCII 文字との混合も可能。ただし漢字コードを示すバックスラッシュは ASCII 文字としては使えない。

漢字を含む文字列の出力例

```
28H¥8356¥8374¥8367JIS¥8ABF¥8E9A
```

これは、'シフト JIS 漢字' を表現している。

● 漢字形式 4

- (1) 漢字表現
シフト JIS 漢字コードを内部表現のまま出力し、漢字 1 文字は 2 バイトを占める。
この方法は IGES ファイルが ASCII テキストを前提にしていることを無視しており、IGES 仕様に反する。
パーソナルコンピュータで OS が MS-DOS である CAD システムが出力する IGES ファイルで多く使用されている。

● 参考：シフト JIS 漢字コードについて

JIS 日本字コードは第一水準、第二水準を含め、94 区 94 点に納められている。

シフト JIS 漢字コードではこれを、 $94 \times 94 = (94 \div 2) \times (94 + 94) = 47 \times 188$ と考える。

そして日本字の第 1 バイトの文字コードが n ($n=1 \sim 47$) を表わしていれば、これは JIS の第 $2n-1$ 区と第 $2n$ 区を意味する。第 2 バイトの文字コードは、JIS 第 $2n-1$ 区の第 1 点～第 94 点および JIS 第 $2n$ 区の第 1 点～第 94 点を表わす。

第1バイト用に47文字、第2バイト用に188文字必要である。
 第1バイトは81～9F、E0～FCの文字コードを、
 第2バイトは40～7E、81～FCの文字コードを、選んである。

6.2.5 アローヘッドの種類

アローヘッド種類	IGES Form #	Advance CAD マーク #
Wedge	1	1
Triangle	2	2
Triangle (filled in)	3	3
No arrow head	4	0(なし)
Circle	5	15
Circle (filled in)	6	4, 16
Rectangle	7	17
Rectangle (filled in)	8	18
Slash	9	5
Integral sign	10	6
Open Triangle	11	7

6.2.6 dimension entity

● 寸法テキストおよび公差テキスト

Advance CAD の IGES 出力はジェネラルノートのフォーム番号を使用しています。
 dimension entity では、1つの general note entity に寸法値・寸法許容差を含めるため、general note entity は複数の文字列を含むことになります。最初に寸法値のテキスト、続いて寸法許容差のテキスト、最後に付加テキストの順で出力します。
 ジェネラルノートが寸法許容差を含むときにはそれが区別できるように、ジェネラルノート
 のフォーム番号 (Form#) をつぎのように設定します。

上の寸法許容差だけ付く Form# 3 (superscript)

2番目のテキストが上の寸法許容差テキスト

下の寸法許容差だけ付く Form# 4 (Subscript)

2番目のテキストが下の寸法許容差テキスト

上の寸法許容差および下の寸法許容値の両方付く Form# 5 (Superscript/Subscript)

2番目のテキストが上の寸法許容差テキスト、

3番目のテキストが下の寸法許容差テキスト

読み込みのときも上記のフォーム番号を参考にします。しかしフォーム番号が上記以外のときは、つぎのように解釈します。

- ・ ジェネラルノート中の文字列のうち、+、-、±で始まる文字列は、寸法公差とみなします。
- ・ 文字高さが変化したところは公差文字列とみなします。一般に公差値テキストの文字高さは寸法値テキストの文字高さよりも小さいので、そのようにします。

- **長さ寸法**
寸法線 (leader or arrow) が抑止されているときは、その寸法線は blanked status01=(blanked) の leader (arrow) entity として出力します。
linear dimension の leader (arrow) line の blanked status が 01 (=blanked) のときは、その寸法線を表示しないよう制御します。
- **累進寸法**
IGES 出力時に、通常の linear dimension と区別するために以下の条件で出力します。
 - 1st leader は、blanked status 01
 - 1st witness line は、なし
 - 2nd leader, 2nd witness line があること。
 - 寸法テキストの general note のテキストの向きが witness line の向きと平行であること。
 IGES 入力時には上記条件を満たす linear dimension item は累進寸法として取り込みます。
- **寸法**
累進寸法の起点以外は、寸法値が 0 となるような寸法は入力できません。
- **オーディネイト寸法 (IGES Ordinate Dimension Entity)**
寸法補助線の折れ曲がり処理できます。
折れ曲がった寸法補助線は Witness line entity ではなく Leader entity で出力します。
累進寸法で寸法補助線を折り曲げた場合は、折れ曲がり処理できません。

6.2.7 general note entity

- **テキストフォント、スラント角などのテキストパラメータ**
IGES general note entity は複数のテキストを含み、それぞれにテキストフォント、スラント角などを個別に指定できます。しかし、Advance CAD のジェネラルテキストはテキストは 1 つだけしか持ちません (改行を入れて複数行にすることは可能)。
したがって、IGES general note entity 中の複数のテキストは Advance CAD では単一のテキストに結合します。そしてテキストフォント、スラント角などは最初のテキストのものが使用されます。
- **文字間隔・行間隔**
文字間隔は IGES には規定はないので text box width から割り出します。
Advance CAD では文字間隔と行間隔は文字高さの比 ($n/64$) で指定し、 $n=0 \sim 63$ です。 $n > 63$ のときは、すべて $n=63$ とします。
文字間隔は、box width と文字数から割り出します。
 - 1 文字の幅は文字高さの半分とします。
 - 文字高さ h 、文字数 m 、文字間隔 $n(0 \sim 63)$ とすると、文字間隔は $d=h*n/64.0$ となり、
 $\text{box/width}=0.5*(hm) + d*(m-1)$ となります。
 行間隔は第 1 行目と第 2 行目の text start point から割り出します。
文字高さ h 、行間隔 $n(0 \sim 63)$ とすると、行間隔は $d=h*n/64.0$ となります。
よって、 $h+d$ は第 1 行目と第 2 行目の text start point の距離 text baseline と直交方向に測った距離となります。
- **文字高さ 0 のジェネラルノート**
寸法アイテムで寸法テキストの文字高さを 0 にして記入し、別にジェネラルノートで寸法値を記入しているという例がありました。このような寸法アイテムを IGES 出力すると、寸法テキストはそのまま文字高さが 0 のジェネラルノートエンティティとして出力されます。

このような IGES ファイルを読み込むと、Advance CAD が停止してしまうことがあります。そのため、文字高さ 0 のときは文字高さを最小値 0.01 に変更して読み込むようにしました。このような文字高さ 0 は、使用しない方が安全です。

● 寸法付加テキスト

今までは風船の形状は円だけでしたが、正方形や円に水平な分割線のような風船も受け渡しできるようになりました。寸法公差のあとに付けた寸法付加テキストも渡すことができます。

IGES 出力

今までは風船の形状にかかわらず風船形状として円を出力していました。バージョン 5.0 からは正確に風船の形状が出力されます。

風船付きのラベルは IGES General Symbol entity(228) として出力します。

特定の形状の風船は General Symbol entity の FORM 番号を使用します。

Datam feature symbol FORM#1

Advance CAD マーク番号 51 (正方形)

Datam target symbol FORM#2

Advance CAD マーク番号 52 (円に水平な分割線)

上記以外の風船形状では FORM#0 です。

IGES 入力

General Symbol entity は Advance CAD のリファレンスラベルになります。このとき風船の形状記述図形から Advance CAD のマーク番号を決定することは困難です。以下の場合だけ簡単にマーク番号を決定できます。

FORM#1

FORM#2

FORM#0 で円が 1 つ

● IGES シンボルフォント

IGES ジェネラルノートのシンボルフォント #1, #2 およびドラフティングフォントの変換を改良し、より多くの記号をサポートできるようになりました。

変換対応表はつぎのとおりです。

Advance CAD IGES/WRITE では IGES ドラフティングフォント (#1003) を使用します。これは IGES version 4 からサポートされている新しいフォントです。

対応表

名称		IGES フォント			Advance CAD
		1001	1002	1003	
寸法補助記号					
Degree	°		\$	~	M-0
Diameter	φ	n		n	M-1
Plus/minus	±		#	`	M-2
Square	□			o	M-3
Arc length	⌒			^	M-4
幾何公差					
Straightness				u	マーク 111
Flatness		c		c	マーク 112

対応表

名称		IGES フォント			Advance CAD
		1001	1002	1003	
Circularity		e		e	マーク 113
Cylindricity		g		g	マーク 114
Profile of line		k		k	マーク 115
Profile of surface		d		d	マーク 116
Parallelism		f		f	マーク 117
Perpendicularity		l		b	マーク 118
Angularity		a		a	マーク 119
Position		j		j	マーク 120
Concentricity		r		r	マーク 121
Symmetry		i		i	マーク 122
Runout		h		h	マーク 123
Total runout				t	マーク 124
公差					
Least material condition				l	マーク 203
Maximum material condition		m		m	マーク 204
Projected tolerance zone		p		p	マーク 205
Regardless of feature side		s		s	マーク 206
算術					
Division symbol	÷		b		漢字 1-64
Less than or equal	≤		c		漢字 1-69
Greater than or equal	≥		d		漢字 1-70
Radical	√		f		漢字 2-69
Cross product	×		g		漢字 1-63
Congruence	≡		h		漢字 1-65
Not equal	≠		i		漢字 1-66
Integral	∫		j		漢字 2-73
Implication	⊃		k		漢字 2-31
Union	∨		l		漢字 2-43
Intersection	∧		m		漢字 2-42

対応表

名称		IGES フォント			Advance CAD
		1001	1002	1003	
Approximately equal			n		漢字 2-66
ギリシャ文字					
Sigma	Σ		o		漢字 6-18
Phi	φ		t		漢字 6-53
Theta	θ		u		漢字 6-40
Gamma	γ		v		漢字 6-35
Psi	ψ		w		漢字 6-55
Omega	ω		x		漢字 6-56
Lamda	λ		y		漢字 6-43
Alpha	α		z		漢字 6-33
Delta	δ		{		漢字 6-36
Mu	μ				漢字 6-44
Pi	π		}		漢字 6-48
マーカー／シンボル					
			a		不可
		b			不可
			e		不可
		t			不可
		u			不可
		v			不可
		w			不可
		x			不可
		y			不可
		z			不可
その他					
Up arrow	↑		p		漢字 2-12
Down arrow	↓		q		漢字 2-13
Right arrow	→		r		漢字 2-10
Left arrow	←		s		漢字 2-11

対応表

名称		IGES フォント			Advance CAD
		1001	1002	1003	
All around applicability	○	o			不可
Center line		q			マーク 201
Overscore	—		～		漢字 1-17
Counterbore				v	不可
Countersink	∇			w	不可
Depth				x	不可
Conical taper				y	不可
Slope				z	不可

6.2.8 IGES/WRITE で出力するグローバル部のパラメーター一覧

No. Type Description	Advance CAD の出力値	備考
1. 文字 Parameter delimiter character	1H,	
2. 文字 Record delimiter	1H;	
3. 文字 Product identification from sender	7HUnknown またはモデル名	
4. 文字 File name	IGES ファイルの名前	
5. 文字 System ID	45HC. ITOH Techno- Science Co., LTD Advance CAD X.X	X.X は Advance CAD のバージョン
6. 文字 processor version	3HX.X	X.X は Advance CAD のバージョン
7. 整数 Number of binary bits for integer representation	16	
8. 整数 Single precision magnitude	38	
9. 整数 Single precision significance	7	
10. 整数 Double precision magnitude	38	
11. 整数 Double precision significance	15	
12. 文字 Product identification for the receiver	7HUnknown またはモデル名	
13. 実数 Model space scale	REAL または 1.0	REAL=1.0/(ドロー イン縮尺値) ドローイング座標 に変換しない場合 ドローイング座標 に変換する場合
14. 整数 Unit flag	1 または 2	
15. 文字 Units	4HINCH または 2HMM	
16. 整数 Maximum number of line weight gradations	15	
17. 実数 Width of maximum line weight in units	REAL	REAL=2x15x (線幅のピッチ)

No. Type Description	Advance CAD の出力値	備考
18. 文字 Date & time of exchange file generation	13HYMMDD.HHN NSS	IGES/WRITE が実行された日時 (YY 年 西暦下 2 桁 MM 月 01-12 DD 日 01-31 HH 時 00-23 NN 分 00-59 SS 秒 00-59)
19. 実数 Minimum user-intended resolution	0.1D-9	
20. 実数 Approximate maximum coordinate value	REAL	REAL=(モデルが定義される領域の最大座標値)
21. 文字 Name of author	STRING	STRING= 作成者名 (t2)
22. 文字 Author's organization	STRING	STRING= 組織名 (t3)
23. 整数 IGES version number	8	
24. 整数 Drafting standard code	3 または 7	
25. 文字 Date and Time model was Create/Modified	13HYMMDD.HHN NSS	(No.18 と同値)

第 7 章 SXF

● 概要

国土交通省が（財）日本建設情報総合センター（JACIC）に委託して開発した SXF 形式 CAD データファイルの入出力を行う SXF 共通ライブラリ (Ver 3.2) を使用して SXF 形式 CAD データファイルの入出力を行います。

SXF 形式 CAD データファイルは、**sfc** と **p21** 2 種類の形式が可能です。

SXF (Scadec data eXchange Format)

SCADEC の開発成果である CAD データ交換標準（仕様、ツール等）全体の通称。

国際標準である ISO10303 STEP/AP202 規則に則った電子納品のための **P21** 形式及び関係者間での CAD データ交換のための簡易な形式である **sfc** 形式をサポートしています。

● 前準備

コンフィグレーションファイルに次のキーワードを追加してください。

P21 形式のファイルを格納するディレクトリとファイル拡張子を指定します。

```
#SXF:P21#        "path"        !.ext!
```

SFC 形式のファイルを格納するディレクトリとファイル拡張子を指定します。

```
#SXF:SFC#        "path"        !.ext!
```

変換パラメータファイルを格納するディレクトリとファイル拡張子を指定します。

```
#SXF:TABLE#      "path"        !.ext!
```

一時的に使用する作業ファイルを格納するディレクトリとファイル拡張子を指定します。

```
#SXF:WORK#       "path"        !.ext!
```

CAD 製図基準テンプレートファイルが格納されているディレクトリとファイル拡張子を指定します。このキーワードがない場合は **#SXF:TABLE#** を用います。

```
#SXF:TEMPLATE#    "path"        !.ext!
```

7.1 国土交通省 C A D 製図基準 (案) への対応および S X F 変換を円滑に行なうために

7.1.1 概要

C A D 製図基準 (案) での運用および S X F 変換を円滑に行なうためにサンプルファイル (製図基準テンプレートファイル、定数ファイル (線種、色割付)、変換テーブルファイル) を用意しています。C A D 製図基準 (案) での運用も C A D 製図基準 (案) を使用しない運用も同じ設定で行なえるようになっていきます。違いは C A D 製図基準 (案) で運用の場合は新規モデル作成と S X F 読み込みの時に責任主体と図面種別を設定する必要があります。

サンプルファイルには次のものがあります。

線種定義 (L I N E S T Y L E . T X T)	: 8.5.1 S X F 既定義線種を定義
色割付 (C O L O R A S S I G N . T X T)	: 色 (1-256) をレビジョン (1-256) に割付
色定義 (C O L O R T B L . T X T)	: 8.5.3 S X F 既定義色を定義
メニュー (U S E R O S M . M E N , U S E R Z O N . M E N)	
変換テーブルサンプル (S X F → A d v a n c e C A D)	: 8.3.3 参照
変換テーブルサンプル (A d v a n c e C A D → S X F)	: 8.3.5 参照
C A D 製図基準テンプレートサンプル	: 8.4 参照

C A D 製図基準 (案) では責任主体、工種、図面種類別に各要素のレイヤ名、線種、色が定められています。A d v a n c e C A D ではレイヤをクラス、色をレビジョンに割当ててすることで C A D 製図基準 (案) に従った運用が行えます。

C A D 製図基準 (案) での運用が必要なければ S X F / T E M P L A T E、D R W S T D、D R W T Y P コマンドおよび「製図基準テンプレートファイル」は無関係です。S X F / R E A D、S X F / W R I T E コマンドおよび「変換テーブル」だけで S X F 変換が可能です。

7.1.2 初期設定

ここでは、C A D 製図基準 (案) での運用と S X F 変換を円滑に行なうための設定を説明します。現在運用されている環境とは別の環境を作成して運用するように設定します。既存の環境を残す必要がない場合は既存の環境のまま設定してもさしつかえありません。いずれの場合も既存の環境を保存しておくことをお勧めします。

C A D 製図基準 (案) で運用するためのサンプルファイルを用意しています。これらを使用すると図面種類別の定数ファイルとアイテム属性定義ファイルが同名で作られます。そのために定数ファイルとアイテム属性定義ファイルの格納ディレクトリを分けておきます。

また既存の定数ファイル、カラーテーブルファイル、ユーザメニューファイルが書き換えられます。書き換えられるファイルの格納されているディレクトリを既存の運用と C A D 製図基準 (案) での運用のために区別しておきます。

ここでは、既存の運用で使用するディレクトリを X X X _ s t d、C A D 製図基準 (案) での運用と S X F 変換を円滑に行なうため使用するディレクトリを X X X _ c a d として記述します。

以下ではファイル名やディレクトリ名は標準設定での名前で記述します。標準設定のディレクトリ名／ファイル名は [] で囲んであります。実際にはコンフィグレーションファイルに記述されているディレクトリ／ファイル名になります。使用中のコンフィグレーションの内容を確認してください。

7.1.2.1 コンフィグレーションファイルとディレクトリの作成

参考：コンフィグレーションファイル名

標準設定では uenv ディレクトリの ACAD.SET になっています。
詳しくは「システム管理者の手引き：起動と環境設定」をご覧ください。

(1) 既存のコンフィグレーションファイル内の格納ディレクトリを変更します。

- アイテム属性定義ファイル：#ATTRTABLE#
格納ディレクトリ [maint] を atr_std に変更します。
- カラーテーブルファイル：#COLORTABLE#
格納ディレクトリ [maint] を clr_std に変更します。
- 定数ファイル：#CONSTANT#
格納ディレクトリ [maint] を cns_std に変更します。
- メニューファイル：#MENU:ASC#、#MENU:BMP#、#MENU:INP#
格納ディレクトリ [menu] を menu_std に変更します。
- モデルタイトルテンプレートファイル：#MODEL:TITLE#
格納ディレクトリ [maint] を title_std に変更します。

(2) ディレクトリを作成し、該当ファイルを新しいディレクトリに移動します。

- アイテム属性定義ファイル
atr_std ディレクトリを作成し、変更前のディレクトリ [maint] 内のアイテム属性定義ファイルを atr_std に移動します。
- カラーテーブルファイル
clr_std ディレクトリを作成し、変更前のディレクトリ [maint] 内のカラーテーブルファイル [COLORTBL.TXT] を clr_std に移動します。
- 定数ファイル
cns_std ディレクトリを作成し、変更前のディレクトリ [maint] 内の定数ファイル [CONSTANT.TXT] を cns_std に移動します。その他にも定数ファイルがあればそれも cns_std ディレクトリに移動します。
- メニューファイル
変更前のディレクトリ [menu] を menu_std という名前に変更します。
- モデルタイトルテンプレートファイル
title_std ディレクトリを作成し、変更前のディレクトリ [maint] 内のモデルタイトルテンプレートファイル [TITLE.TMP] を title_std に移動します。その他にもモデルタイトルテンプレートファイルがあればそれも title_std ディレクトリに移動します。

(3) C A D 製図基準 (案) での運用と SXF 変換を円滑に行なうためのコンフィグレーションファイルを作成し、格納ディレクトリを変更します。

[ACAD.SET] を例えば ACAD_CAD.SET という名前でコピーし ACAD_CAD.SET 内の格納ディレクトリを変更します。

- アイテム属性定義ファイル : #ATRTABLE#
格納ディレクトリ atr_std を atr_cad に変更します。
- カラーテーブルファイル : #COLORTABLE#
格納ディレクトリ clr_std を clr_cad に変更します。
- 定数ファイル : #CONSTANT#
格納ディレクトリ cns_std を cns_cad に変更します。
- メニューファイル : #MENU:ASC#、#MENU:BMP#、#MENU:INP#
格納ディレクトリ menu_std を menu_cad に変更します。
- モデルタイトルテンプレートファイル : #MODEL:TITLE#
格納ディレクトリ titlt_std を title_cad に変更します。

- (4) C A D 製図基準 (案) での運用と SXF 変換を円滑に行なうためのディレクトリを作成します。

以下のディレクトリをディレクトリ内のファイルも含めて C A D 製図基準 (案) での運用のためのディレクトリにコピーします。

既存の運用	C A D 製図基準 (案) での運用
atr_std	atr_cad
clr_std	clr_cad
cns_std	cns_cad
menu_std	menu_cad
title_std	title_cad

7.1.2.2 サンプルファイルのコピー

SXF 変換用サンプルディレクトリ sample/SXF 内の以下のファイルを該当ディレクトリにコピーします。

- 定数ファイル : #CONSTANT#
LINestyle.TXT を cns_cad ディレクトリにコピーします。
線種の既定値が記述されています。(8.5.1 SXF 既定義線種が定義されています)
- 定数ファイル : #CONSTANT#
CCOLORASSIGN.TXT を cns_cad ディレクトリにコピーします。
色割付の既定値が記述されています。(色 1-256 がレビジョン 1-256 に割付られています)
- カラーテーブルファイル : #COLORTABLE#
COLORTBL.TXT を clr_cad ディレクトリにコピーします。
色の既定値が記述されています。(8.5.3 SXF 既定義色が定義されています)

- ユーザメニューファイル : #MENU:ASC#
USERZON.MEN と USEROSM.MEN を menu_cad ディレクトリにコピーします。
すでに USERZON.MEN および USEROSM.MEN が存在していたならば、既存ファイル内の記述をコピー後の USERZON.MEN および USEROSM.MEN にも記述します。
製図基準 (DRWSTD)、図面種別 (DRWTYP)、クラス名を設定・表示および属性設定 (SXF/ATTR) メニューが記述されています。
- 製図基準テンプレートファイル : #SXF:TEMPLATE#
テンプレートファイル
H2005_維持管理_C A D 製図基準.TMP
H2005_施工_C A D 製図基準.TMP
H2005_設計_C A D 製図基準.TMP
H2005_測量_C A D 製図基準.TMP
必要なものを [maint] ディレクトリにコピーします。
C A D 製図基準 (案) H2005 が責任主体 (維持管理、施工、設計、測量) 別に記述されています。不要な責任主体のものはコピーする必要はありません。

7.1.2.3 標準定数ファイルに線種の既定値を登録

- (1) ACAD_CAD.SET を使用して AdvanceCAD を起動します。
- (2) 『その他設定』 → 『定数登録』 → 『定数呼出』 で LINSTYLE.TXT を呼び出します。
- (3) 『その他設定』 → 『定数登録』 → 『定数呼出』 で COLORASSIGN.TXT を呼び出します。
- (4) 『その他設定』 → 『定数登録』 → 『定数保存』 で 標準定数ファイル [CONSTANT.TXT] にすべての定数を保存します。
- (5) AdvanceCAD を終了します。

参考：コンフィグレーションファイルを指定して Advance CAD を起動する方法

- コマンドプロンプトから起動する場合
引数に -p でコンフィグレーションファイル名を指定します。
たとえば acad.exe -pACAD_CAD とします。
- アイコンから起動する場合
アイコンのプロパティ：リンク先に -p でコンフィグレーションファイル名を指定します。
例えば c:\acad\exe\acad.exe -pACAD_CAD とします。

7.1.2.4 各種設定ファイルの作成

CAD 製図基準 (案) で運用しない場合はここでの設定は必要ありません。

- (1) ACAD_CAD.SET を使用して AdvanceCAD を起動します。
- (2) 『データ交換_標準』 → S X F 『テンプレート』 で、以下の製図基準テンプレートファイルを 1 ファイルづつ指定して実行します。不要な責任主体のものは実行する必要はありません。
 - H2005_維持管理_C A D 製図基準.TMP
 - H2005_施工_C A D 製図基準.TMP

- H2005_設計_CAD 製図基準.TMP
- H2005_測量_CAD 製図基準.TMP

(3) Advance CAD を終了します。

『テンプレート』(SXF/TEMPLATE)は、製図基準テンプレートファイルを読み込んで各種の設定ファイルを作成します。たとえば、設計_CAD 製図基準_H2005.TBL を処理したときに作成されるファイルは以下のようになります。

- 製図基準ファイル (テキストライブラリファイル) : #TEXT#

ファイル名

製図基準テンプレート内の LS+FIL でファイル名が決められます。

例. H2005_設計_CAD 製図基準.TXT

内容

全工種の図面種別が記述されています。

例. H2005_設計_道路本体設計_位置図: LC
 H2005_設計_道路本体設計_平面図および交差点位置図: PL, IP
 H2005_設計_道路本体設計_縦断図: PF

- 定数ファイル : #CONSTANT#

ファイル名

製図基準テンプレート内の LS+T1+T2+T3 でファイル名が決められます。

例. H2005_設計_道路本体設計_位置図: LC.TXT
 H2005_設計_道路本体設計_平面図および交差点位置図: PL, IP.TXT
 H2005_設計_道路本体設計_縦断図: PF.TXT

内容

各図面種別に対応したクラス名 (レイヤ名) が記述されています。

例. 011-000-001 CLSNME001 T "D-TTL"
 011-000-002 CLSNME002 T "D-TTL-FRAM"
 011-000-003 CLSNME003 T "D-TTL-LINE"
 011-000-004 CLSNME004 T "D-TTL-TXT"

- アイテム属性定義ファイル : #ATRTABLE#

ファイル名

製図基準テンプレート内の LS+T1+T2+T3 でファイル名が決められます。

例. H2005_設計_道路本体設計_位置図: LC.TXT
 H2005_設計_道路本体設計_平面図および交差点位置図: PL, IP.TXT
 H2005_設計_道路本体設計_縦断図: PF.TXT

内容

各図面種別に対応した属性定義が記述されています。

例.	/ "要素"	線種	線幅	クラス	レビジョン
" 外枠 "	"	1	1	1	5
" タイトル枠、凡例図枠 "	"	1	1	2	5
" 区切り線、罫線 "	"	1	1	3	8
" 文字列 "	"	1	1	4	8

- SXF 変換テーブル : #SXF:TABLE#

ファイル名

製図基準テンプレート内の LS+FIL+"_ACADSXF", LS+FIL+"_SXFACAD" でファイル名が決められます。

例. H2005_設計_CAD 製図基準_ACADSXF.TBL : AdvanceCAD → SXF 変換テーブル
 H2005_設計_CAD 製図基準_SXFACAD.TBL : SXF → AdvanceCAD 変換テーブル

内容

SXF 変換時の変換パラメータが記述されています。

7.1.2.5 製図基準ファイルの絞り込みと自動問い合わせ

CAD 製図基準 (案) で運用しない場合はここでの設定は必要ありません。

製図基準は、コマンド『製図基準』(DRWSTD) および S X F 『読み込み』(SXF/READ) で設定することができ、#TEXT# で指定されたディレクトリ内のテキストライブラリファイルを選択します。選択されたテキストライブラリファイル名は製図基準としてモデルタイトル項目番号 208 の内容としてモデルに保存されます。

図面種別は、コマンド『図面種別』(DRWTYP) および S X F 『読み込み』(SXF/READ) で設定することができ、製図基準ファイル (テキストライブラリファイル) 内の該当図面種別を選択します。選択された図面種別はモデルタイトル項目番号 209 の内容としてモデルに保存されます。

モデルタイトルには初期値を設定したり自動問い合わせをする機能があり、製図基準と図面種別もこの機能を利用することができます。
必要ならばモデルタイトルテンプレートに項目番号 208 と 209 で製図基準と図面種別についての設定をします。

モデルタイトルについての詳細は「システム管理者の手引き：モデルタイトルの設定」をご覧ください。

- ・ 製図基準の初期値を " 設計 _ C A D 製図基準 " とする場合
(208) " 製図基準 " [0, 0, 0, 0, 0] !VAL=" 設計 _ C A D 製図基準 "!
- ・ 製図基準入力時にテキストライブラリファイルをワイルドカード指定で絞りこむ場合
(208) " 製図基準 " [0, 0, 0, 0, 0] !TLB="*C A D 製図基準"!

!TLB="*C A D 製図基準"!によりテキストライブラリファイルをワイルドカード指定によって絞り込みます。

- ・ 製図基準と図面種別を自動問い合わせにする場合
(208) " 製図基準 " [0, 0, 0, 1, 0] !TLB="*C A D 製図基準"!
(209) " 図面種別 " [0, 0, 0, 1, 0]

[0, 0, 0, 1, 0] の 5 番目の数値 1 によりこの項目は必須入力になり、Advance CAD の起動時や新規開始時に自動問い合わせ項目になります。

7.1.3 C A D 製図基準 (案) での運用

7.1.3.1 製図基準を設定する

製図基準を設定するには以下の方法があります。

- ・ モデルタイトルの製図基準 (208) を自動問い合わせ項目にしておく。
- ・ コマンド『製図基準』(DRWSTD) を選択する。
- ・ コマンド S X F 『読み込み』(SXF/READ) 時に指定する。

一覧表示された製図基準ファイルの内から該当する責任主体のものを選択します。

以下は "Hxxxx_設計_CAD 製図基準.TXT" を選択したものとして記述します。

このファイル名に "_ACADSXF" および "_SXFACAD" をつけた SXF 変換テーブルが必要です。

- S X F 『書込み』時は SXF 変換テーブル "Hxxxx_設計_CAD 製図基準_ACADSXF.TBL" が必要。
- S X F 『読み込み』時は SXF 変換テーブル "Hxxxx_設計_CAD 製図基準_SXFACAD.TBL" が必要。

製図基準ファイルはテキストライブラリファイルで、特に指定しないかぎり全てのテキストライブラリファイルが一覧表示されます。製図基準ファイル以外にもテキストライブラリファイルが存在する場合は、前述のモデルタイトルテンプレートを利用したファイル名の絞り込みをすると便利です。

7.1.3.2 図面種別を設定する

図面種別を設定するには以下の方法があります。

- モデルタイトルの図面種別 (209) を自動問い合わせ項目にしておく。
- コマンド『図面種別』(DRWTYP) を選択する。
- コマンド S X F 『読み込み』(SXF/READ) 時に指定する。

一覧表示された図面種別の中から該当する図面種別を選択します。

以下は "Hxxxx_設計_道路本体設計_位置図:LC" を選択したものとして記述します。

図面種別を選択すると以下の処理が行われます。

- 図面種別と一致するファイル名の定数ファイルを読み出す。
たとえばファイル名 "Hxxxx_設計_道路本体設計_位置図:LC.TXT" の定数ファイルを読み出します。この定数ファイルには「設計_道路本体設計_位置図」の場合のクラス名 (レイヤ名) が記述されています。
- 図面種別と一致するファイル名のアイテム属性定義ファイルを標準線属性ファイルとする。
たとえばファイル名 "Hxxxx_設計_道路本体設計_位置図:LC.TXT" を標準線属性ファイルとします。
このファイルはアイテム属性の設定/変更や表示マスクの設定コマンドなどで内容が一覧表示されます。一覧表示から該当するアイテム属性を選択することができます。

7.1.3.3 SXF ファイルを読み込む

S X F 『読み込み』(SXF/READ) で SXF ファイルを読み込む。このとき製図基準に該当する変換テーブル、たとえば "Hxxxx_設計_CAD 製図基準_SXFACAD.TBL" を使用し、レイヤがクラスに変換されます。

7.1.3.4 責任主体 (ライフサイクル) を変更する

たとえば「現況地物」を設計から施工に変更する場合は、該当要素のクラスを別のクラスに変更するのではなく、クラス名 "D-BGD" を "C-BGD" に変更します。

1 つのモデル内では同一のレイヤに対して 2 つ以上の責任主体を持つことはできません。たとえば「現況地物」として "D-BGD" と "C-BGD" の両方を混在させることはできません。

別のレイヤ間では複数の責任主体が混在してもかまいません。

7.1.3.5 S X F ファイルを出力する

S X F 『書込み』 (SXF/WRITE) で SXF ファイルを出力します。このとき製図基準に該当する変換テーブル、たとえば "Hxxxx_設計_C A D 製図基準_ACADSXF.TBL" を使用し、クラスがレイヤに変換されます。

7.1.3.6 その他の関連コマンド

- **カレントアイテム属性の設定**
『クラス (レイヤ)』 (CLS)、『レビジョン (色)』 (REV)、『線種』 (LFT)、『線幅』 (LWT) および『A T R』 (ATR) で『線属性』 (LAST) を指定することで標準線属性ファイル内の属性定義一覧が表示されます。該当行を選択することでカレント属性が設定できます。
- **特定レイヤの表示**
表示マスク『線属性から』 (ATR/DSP) で『標準線属性』 (LAST) を指定することで標準線属性ファイル内の属性定義一覧が表示されます。該当行を選択することでそのレイヤだけの表示／非表示が設定できます。
- **既存アイテムの属性変更**
属性の変更『一括変更』 (ATR/MOD) で『線属性一覧表』 (LAST) を指定することで標準線属性ファイル内の属性定義一覧が表示されます。該当行を選択することで変更後の属性が設定できます。

7.2 コマンド

● コマンド一覧

コマンド名	機能
SXF / TEMPLATE	製図基準テンプレートファイルを使用して環境を設定
DRWSTD	製図基準を設定
DRWTYP	図面種別を設定
SXF / READ	SXF ファイル読み込み
SXF / WRITE	SXF ファイル書き込み
SXF / MARKER	ポイントマーカー作成
SXF / HATCH _USR	ユーザ定義ハッチング作成
SXF / HATCH _PAT	パターンハッチング作成
SXF / HATCH _FIL	塗りハッチング作成
SXF / HATCH _DEF	既定義ハッチング (Area_control) 作成
SXF / USER _CLR	ユーザ定義色設定
SXF / USER _LFT	ユーザ定義線種設定
SXF / USER _LWT	ユーザ定義線幅設定
SXF / PIC _LIST	部分図の一覧表示
SXF / PIC _SCF	部分図倍率設定
SXF / LIST	各種名称の一覧を表示
SXF / SFIG _GEN	複合図形 (作図グループ・作図部品) 作成
SXF / SFIG _REF	複合図形 (作図グループ・作図部品) 配置
SXF / SFIG _EDT	複合図形 (作図部品) 変更
SXF / SFIG _DEL	複合図形 (作図グループ・作図部品) 削除
SXF / SFIG _BRK	作図グループ・作図部品を図形要素分解
SXF / MOVE	複合図形を含めたアイテムの移動・回転・縮小拡大
SXF / TTL _SET	表題欄設定
SXF / TTL _CRE	表題欄属性 (文字列に表題欄属性を付加)
SXF / SEQUENCE	表示順
SXF / PRINT	図面印刷 (図面範囲を印刷)
SXFCOPY / PRINT	画面印刷 (画面に表示されている部分を印刷)
SXF / PREVIEW	図面印刷プレビュー (図面印刷のプレビュー)
SXFCOPY / PREVIEW	画面印刷プレビュー (画面印刷のプレビュー)

コマンド名	機能
SXF / ATTRSET	属性付加機構（属性を付加、編集、削除）
SXF / CONFIRM	確認機能
SXF / OPTION	オプション設定

● 割込みコマンド一覧

割込みコマンド名	機能
SXF / ATTR	属性設定
SXF / LAY	レイヤ表示・非表示設定
SXF / CNT	等高線作成
SXF / VER	図形確認
SXF / ZOOMALL	図面枠全体表示
SXF / ZOOMALLVIE	データ全体表示
SXF / ZOOM	矩形領域拡大表示
SXF / ZOOMUP	拡大表示
SXF / ZOOMDOWN	縮小表示
SXF / PAN	表示位置移動
SXF / RPT	再表示

7.2.1 製図基準テンプレートファイルを使用して環境を設定

【機能】

テンプレートファイルを読み込み、製図基準ファイル・定数ファイル・属性定義ファイル・SXF 変換テーブルファイルを作成します。

【メニュー】

〔データ交換 1〕 → SXF 〔テンプレート〕

【構文】

```
SXF/TEMPLATE FNM filename <CE>
```

読み込むファイル名を指示します。

FNM filename : テンプレートファイル名を入力します。

各種ファイルを作成します。

<CE> : 処理を開始します。

【注意事項】

本コマンドで作成したテンプレート（各種定数ファイル）を使って SXF 変換を行う場合、既存の図面はテンプレートの設定とあわなくなり SXF 変換が正しく行われなくなる可能性があります。

【各種ファイルの内容】

製図基準ファイル（コマンド DRWSTD. DRWTYP で使用）
図面種別の一覧

定数ファイル（コマンド DRWTYP で使用）
クラス名、レビジョン名およびカラー割付けの定数

属性定義ファイル（コマンド ATR LAST で使用）
クラス、線種、線幅およびレビジョンの組み合わせ

SXF 変換テーブル（コマンド SXF/READ, SXF/WRITE で使用）
SXF 変換用テーブルファイル

7.2.2 製図基準を設定

【機能】

製図基準（設計、施工、測量又は維持管理）を設定します。

【構文】

DRWSTD filename

製図基準ファイル名を指定します。

filename : 製図基準ファイル名を一覧表示から選択します。

7.2.3 図面種別を設定

【機能】

図面種別（平面図、縦断図、... 等）を設定します。

【構文】

DRWTYP text

図面種別ファイル名を指定します。

text : 図面種別を一覧表示から選択します。

7.2.4 SXF ファイル読み込み

【機能】

SXF ファイルを読み込みます。

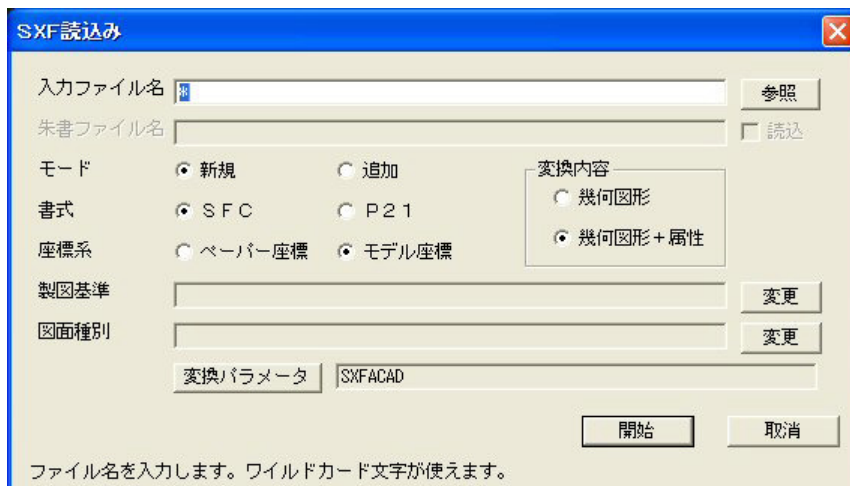
【メニュー】

〔データ交換 1〕 → SXF〔読み〕

【構文】

SXF/READ	NEW ADD	FNM	filename	ONN OFF	CADSTD filename PLANTYP text PARAM <CE>
----------	--	-----	----------	--	--

【SXF 読みダイアログ】



モデルの初期化を指示します。

NEW : モデルを初期化し読みます。

ADD : 既存のモデルを初期化しないで追加読みします。

読み込むファイルを指定します。

FNM filename : SXF ファイル名を入力します。

朱書きファイルの読みを指示します。

ONN : 朱書きファイルを読みます。

OFF : 朱書きファイルは読みません。

製図基準ファイルを指定します。

CADSTD filename : 製図基準ファイル名を入力します。

製図基準に対応した SXF 変換テーブルファイルが必要です。

図面種別を指定します。

PLANTYP text : 図面種別を入力します。

製図基準が設定されている必要があります。

図面種別に対応した定数ファイルが必要です。

製図基準・図面種別を指定すると製図基準に対応した SXF 変換テーブルを使って変換処理を行い、製図基準・図面種別に対応した定数ファイルを読み込みクラス名（レイヤ名）、カラー割付けを設定しま

す。また属性設定コマンド（ATR）を簡単に使用できるように製図基準・図面種別に対応した属性定義ファイル名を登録します。

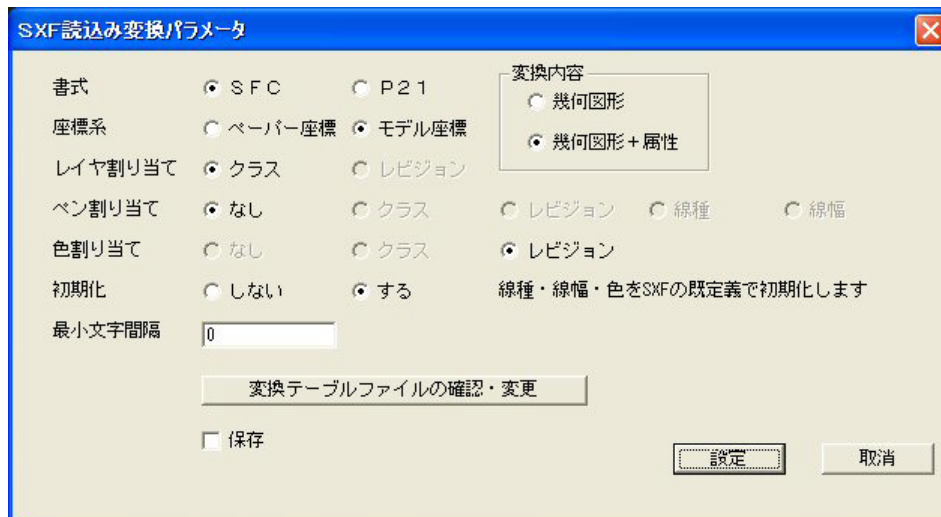
製図基準ファイル、変換テーブルファイル、属性定義ファイルおよび定数ファイルは SXF/TEMPLATE コマンドで作成します。

パラメータを指示します。

PARAM

: SXF から Advance CAD への変換パラメータを指示します。
画面に表示されてパラメータをクリックしてパラメータの変更および保存が行えます。（下記を参照）

【変換パラメータ】



書式を選択します。

SFC : SFC ファイルを読み込みます。

P21 : P21 ファイルを読み込みます。

座標系を選択します。

ペーパー座標 : 紙に作図した寸法で変換します。

モデル座標 : 実寸法で変換します。

レイヤ割当を選択します。

クラス : レイヤをクラスに割当てます。

レビジョン : レイヤをレビジョンに割当てます。

色割当を選択します。

なし : 色は変換しません。

クラス : 色をクラスに割当てます。

レビジョン : 色をレビジョンに割当てます。

最小文字間隔を入力します。

最小文字間隔 : 最小文字間隔を文字高さに対する比較で入力します。

変換内容を選択します。

幾何図形 : 幾何図形のみを読み込みます。

幾何図形+属性 : 幾何図形及び属性を読み込みます。

初期化を選択します。

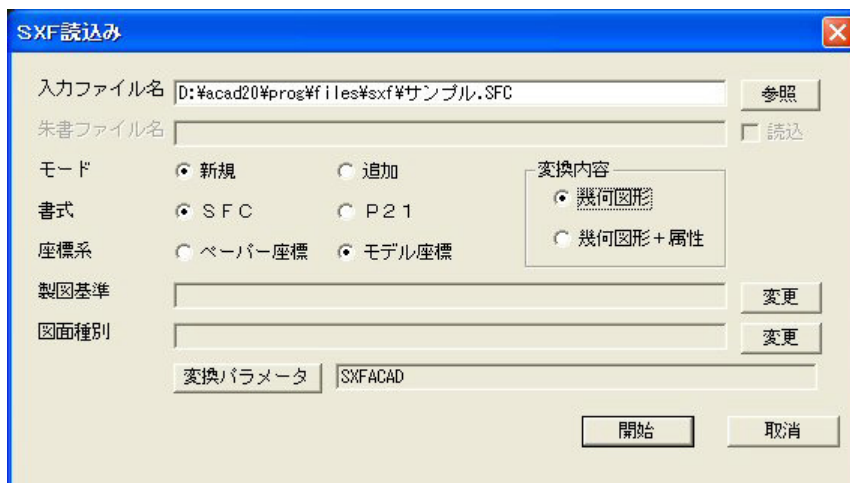
しない : モデルを初期化したときの線種、線幅、色を使用します。

する : 線種、線幅、色を SXF の既定義線種、線幅、色で初期化します。

設定内容を保存します。

保存 : 設定した内容をパラメータファイルに保存します。

【朱書きファイルがない場合】



【朱書きファイルがある場合】



図面ファイルに朱書きファイルがある場合は朱書きファイルを読み込むかどうかを選択します。
朱書きファイルは

図面配置ピクチャ
レイヤ (クラス) 名 " 朱書き "

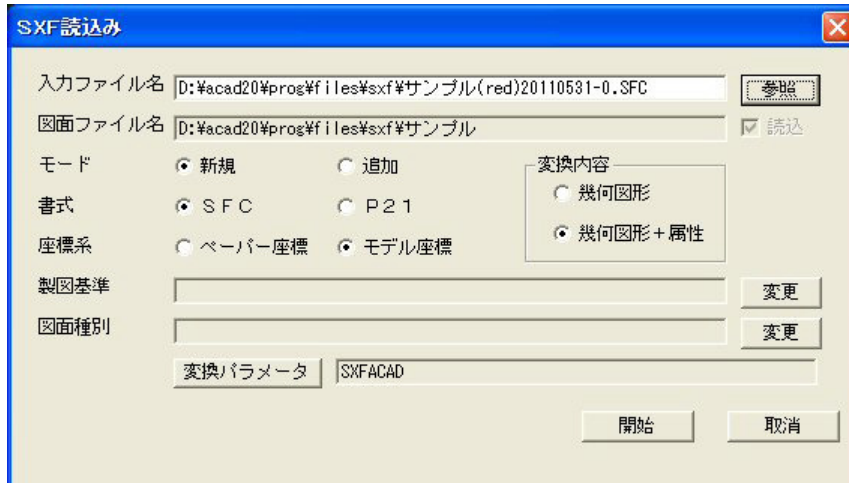
ペーパー座標

複合図形は各要素に展開

で読み込みます。

朱書きファイルを読み込んだ場合レイヤ (クラス) の選択マスクは、朱書きレイヤのみが選択
不可でそれ以外のレイヤは選択可になっています。

【朱書きファイルを入力ファイルに指定した場合】



朱書きファイルを入力ファイルに指定した場合、関連する図面ファイルを読込んだ後朱書きファイルを読み込みます。

朱書きファイルは

図面配置ピクチャ

レイヤ (クラス) 名 " 朱書き "

ペーパー座標

複合図形は各要素に展開

で読み込みます。

読み込み後レイヤ (クラス) の選択マスクは、朱書きレイヤのみが選択可でそれ以外のレイヤは選択不可になっています。

7.2.5 SXF ファイル書込み

【機能】

指示された図面配置ページのデータを SXF ファイルに書き込みます。

【メニュー】

〔データ交換 1〕 → SXF 〔書込み〕

【構文】

```
SXF/WRITE FNM filename PAGE s NME name ORG name PARAM <CE>
```

書込むファイルを指定します。

FNM filename : 出力ファイル名を入力します。

出力するページを指定します。

PAGE s : ページ番号を入力します。

作成者名を入力します。

NME name : 作成者名を入力します。

組織名を入力します。

ORG name : 組織名を入力します。

変換パラメータを指定します。

PARAM : 変換パラメータの確認・変更を行います。

【注意事項】

- ・図面名（シート名）、ピクチャ名（部分図名）、クラス名（レイヤ名）の設定は必須です。
- ・画像(.TIF、.JPG)は図面配置ピクチャに配置されている必要があります。

【SXF 書き込みダイアログ】

SXF書き込みダイアログボックスのスクリーンショット。このダイアログには、出力ファイル名、出力ページ、朱書きレイヤ、作成者、組織名、バージョン、書式、座標系、製図基準、図面種別、変換パラメータなどの設定項目があります。バージョンは3.1が選択されており、書式はSFCが選択されています。変換内容は幾何図形と幾何図形+属性の2つのオプションがあります。開始、朱書き出力、取消のボタンがあります。

出力ファイル名 参照

出力ページ

朱書きレイヤ

作成者

組織名

バージョン ☐ 2.0 ☐ 3.0 ☒ 3.1

書式 ☒ SFC ☐ P21

座標系 ☐ ペーパー座標 ☒ モデル座標

製図基準

図面種別

変換パラメータ

変換内容
☐ 幾何図形
☐ 幾何図形+属性

開始 朱書き出力 取消

組織名を入力します

【変換パラメータダイアログ】

SXF書き込み変換パラメータダイアログボックスのスクリーンショット。このダイアログには、バージョン、書式、座標系、レイヤ割り当て、ペン割り当て、非表示アイテム、作成者、組織名、変換プログラムなどの設定項目があります。バージョンは3.1が選択されており、書式はSFCが選択されています。変換内容は幾何図形と幾何図形+属性の2つのオプションがあります。設定、取消のボタンがあります。

バージョン ☐ 2.0 ☐ 3.0 ☒ 3.1

書式 ☒ SFC ☐ P21

座標系 ☐ ペーパー座標 ☒ モデル座標

レイヤ割り当て ☒ クラス ☐ レビジョン

ペン割り当て ☒ クラス ☐ レビジョン ☐ 線種 ☐ 線幅 ☐ 色

非表示アイテム ☒ 出力する ☐ 出力しない

作成者

組織名

変換プログラム

変換テーブルファイルの確認・変更

☐ 保存

設定 取消

バージョンを選択します。

Ver2.0

- ： 幾何図形のみ出力します。
イメージは最初に現れた TIF イメージを 1 つだけ出力します。
その他のイメージは出力されません。
円弧長寸法線は角度寸法線に変換して出力します。
クロソイド曲線は折れ線に変換して出力します。

表題欄フィーチャは出力されません。
背景色は出力されません。

Ver3.0 : 幾何図形および背景色、表題欄及び属性を出力します。

Ver2.0 からの変更点

- ・複数イメージ
- ・JPEG イメージ
- ・背景色
- ・図面表題欄フィーチャ
- ・既定義ハッチング Area_control
- ・等高線のデータ表現方法の変更
- ・イメージのデータ表現方法の変更

円弧長寸法線は角度寸法線に変換して出力します。
クロソイド曲線は折れ線に変換して出力します。

Ver3.1 : 幾何図形および背景色、表題欄及び属性を出力します。

Ver3.0 からの変更点

- ・弧長寸法フィーチャの追加
- ・クロソイドフィーチャの追加
(アイテムが編集された場合は折れ線に変換して出力します)
- ・ファイルのヘッダ情報に SXF ファイルのバージョン追加

書式を選択します。

SFC : SFC ファイルを出力します。

P21 : P21 ファイルを出力します。

座標系を選択します。

ペーパー座標 : 紙に作図した寸法で変換します。

モデル座標 : 実寸法で変換します。

レイヤ割当てを選択します。

クラス : クラスをレイヤに割当てます。

レビジョン : レビジョンをレイヤに割当てます。

作成者名を入力します。

作成者 : 作成者名の既定値を入力します。
変換時に変更することができます。

組織名を入力します。

組織名 : 組織名の既定値を入力します。
変換時に変更することができます。

変換プログラム名を入力します。

変換プログラム : 変換プログラム名。(変更できません)

変換内容を選択します。

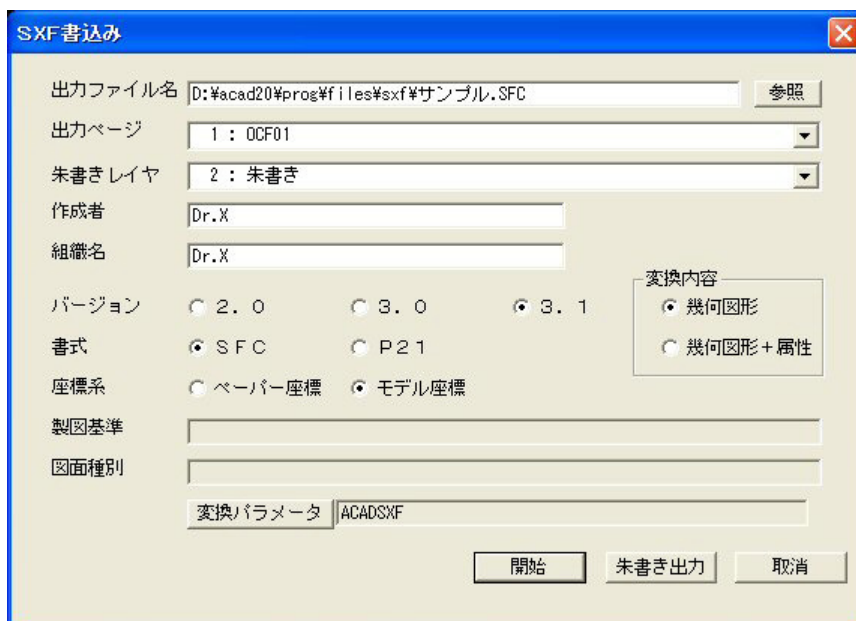
幾何図形 : 幾何図形のみを出力します。

幾何図形+属性 : 幾何図形及び属性を出力します。(Ver3.0、Ver3.1 のみ)

設定を保存します。

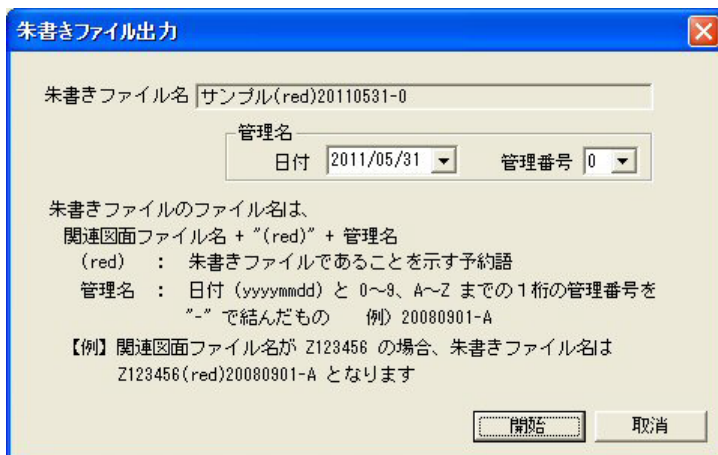
保存 : 設定した内容をパラメータファイルに保存します。

【朱書きファイルの書込み（ダイアログモードのみ）】



朱書きレイヤが設定されている場合、朱書きファイルの書込みができます。
 レイヤはクラスに割当てられているものとして処理しています。
 朱書きファイルの書込みを行う場合は関連図面ファイルが存在する必要があります。
 朱書きレイヤは『レイヤ設定』（SXF/LAY）の「朱書きレイヤ」で設定できます。
 朱書きの図形は図面配置ページに作成された図形を変換します。
 朱書き図形に複合図形（作図グループ、作図部品）は使用できません。

【朱書き出力】



管理名を入力します。

- 〔▼日付〕 : 日付けを選択します。
- 〔▼管理番号〕 : 管理番号を選択します。

出力します。

- 〔開始〕 : 朱書きファイルへ書込みます。

処理を取消します。

- 〔取消〕 : 処理を取消します。

備考) 朱書きファイル名の命名規則

関連図面ファイル名 + "(red)" + 管理名

(red) : 朱書きファイルであることを示す予約語

管理名 : 日付 (yyyymmdd) と 0 ~ 9、A ~ Z までの 1 桁の管理番号を
"-" で結んだもの。 例) 20110101-A

【例】関連図面ファイル名が Z123456 の場合、朱書きファイル名は
Z123456(red)20110101-A となります。

7.2.6 ポイントマーカーを作成

【機能】

ポイントマーカーを作成します。

【構文】

```
SXF/MARKER MRK no SCF scf ANG ang [P]+ <CE>
```

マーカーを選択します。

MRK no : マーカーコードを入力します。
1 : asterisk
2 : circle
3 : dot
4 : plus
5 : square
6 : triangle
7 : X

尺度を入力します。

SCF scf : 尺度を入力します。

角度を入力します。

ANG ang : 角度を入力します。

配置位置を入力します。

P : 配置位置を入力します。

入力を確定します。

<CE> : 入力を確定します。

7.2.7 ユーザ定義ハッチング作成

【機能】

ユーザ定義ハッチングを作成します。

【構文】

```
SXF/HATCH_USR [NEW_ONOFF IScrv] [PTN] [SHOW_ONOFF]  
[ORG org] [GAP gap] [ANG ang] [CUR] ITM [IScrv]+ <CE>
```

作成／修正を選択します。

NEW_ONNOFF : 作成または修正を選択します。(初期値：作成)
 修正するユーザ定義ハッチングを選択します。
IScrv : 修正の場合、ユーザ定義ハッチングを選択します。
 ハッチングのパターンを設定します。
PTN : 1つ次のパターンへ行く。(初期値：1、最高：4)
 現パターンの間隔が0の場合はパターン1へ行きます。
SHOW_ONOFF : 境界図形の表示・非表示を切り替えます。(初期値：表示)
ORG org : ハッチング線の基準点を入力します。(初期値：X=0,Y=0)
GAP gap : ハッチングの間隔を入力します。(初期値：2)
ANG ang : ハッチングの角度を入力します。(初期値：45)
CUR : ハッチングの色・線種・線幅を設定します。
 (初期値：青、実線、0.18)
 サブウィンドウモードの場合。
 現在設定されている色・線種・線幅を使用します。
 ダイアログモードの場合。
 ダイアログで色・線種・線幅を設定します。

境界図形を選択します。

ITM IScrv : 境界図形（閉じたストリングアイテム）を選択します。
 入力を確認します。
 <CE> : ユーザ定義ハッチングを作成又は修正します。

7.2.8 パターンハッチング作成

【機能】

パターンハッチングを作成します。

【構文】

```
SXF/HATCH_PAT [NEW_ONOFF IScrv] [PTN] [SHOW_ONOFF]

[ORG org] [GAP1 gap1] [GAP2 gap2] [SCL_X scl_x] [SCL_Y scl_y]

[ANG ang] [ANG1 ang1] [ANG2 ang2] ITM [IScrv]+ <CE>
```

作成／修正を選択します。

NEW_ONNOFF : 作成または修正の切替をします。(初期値：作成)
 修正するパターンハッチングを選択します。
IScrv : 修正の場合、パターンハッチングを選択します。
 パターンハッチングを設定します。
PTN : パターンを指定します。(初期値：足踏み)
 足踏み又は45度重ねあじろの切替を行います。
SHOW_ONOFF : 境界図形の表示・非表示を切り替えます。(初期値：表示)
ORG org : パターンの配置基準点を入力します。(初期値：X=0,Y=0)
GAP1 gap1 : パターンの配置間隔1を入力します。(初期値：200)
GAP2 gap2 : パターンの配置間隔2を入力します。(初期値：200)
ANG1 ang1 : パターンの配置角度1を入力します。(初期値：0)
ANG2 ang2 : パターンの配置角度2を入力します。(初期値：90)
SCL_X scl_x : パターンのX倍率を入力します。(初期値：1)
SCL_Y scl_y : パターンのY倍率を入力します。(初期値：1)
ANG ang : パターンの角度を入力します。(初期値：0)

境界図形を選択します。

ITM IScrv : 境界図形（閉じたストリングアイテム）を選択します。
 入力を確認します。
 <CE> : パターンハッチングを作成又は修正します。

7.2.9 塗りハッチング作成

【機能】

塗りハッチングを作成します。

【構文】

```
SXF/HATCH_FIL [NEW_ONOFF IScrv] [SHOW_ONOFF] ITM [IScrv]+ <CE>
```

作成／修正を選択します。

NEW_ONNOFF : 作成または修正を切替えます。（初期値：作成）

修正する塗りハッチングを選択します。

IScrv : 修正の場合、塗りハッチングを選択します。

塗りハッチングを設定します。

SHOW_ONOFF : 境界図形の表示・非表示を切り替えます。（初期値：表示）

境界図形を選択します。

ITM IScrv : 境界図形（閉じたストリングアイテム）を選択します。

入力を確認します。

<CE> : 塗りハッチングを作成又は修正します。

7.2.10 既定義ハッチング作成

【機能】

既定義ハッチング（Area_control）を作成します。

【構文】

```
SXF/HATCH_DEF [NEW_ONOFF IScrv] [SHOW_ONOFF] ITM [IScrv]+ <CE>
```

作成／修正を選択します。

NEW_ONNOFF : 作成または修正を切替えます。（初期値：作成）

修正する既定義ハッチングを選択します。

IScrv : 修正の場合、既定義ハッチングを選択します。

既定義ハッチングを設定します。

SHOW_ONOFF : 境界図形の表示・非表示を切り替えます。（初期値：表示）

境界図形を選択します。

ITM IScrv : 境界図形（閉じたストリングアイテム）を選択します。

入力を確認します。

<CE> : 既定義ハッチングを作成又は修正します。

7.2.11 ユーザ定義色設定（ダイアログモードでのみ動作）

【機能】

ユーザ定義色の設定・修正を行います。
 SXF の色 1 ～ 256 が ACAD 色（レビジョン） 1 ～ 256 に 1 対 1 で割当てられていて
 レビジョン 1 ～ 256 が色 1 ～ 256 に 1 対 1 で割当てられている場合のみ使用可能です。

【構文】

SXF/USER_CLR

【ユーザ定義色設定ダイアログ】



RGB 値を入力するまたはパレットのアイテムをクリックすると
 下のカラーパレットが表示されパレットでの色設定が行えます。

【カラーパレット】



7.2.12 ユーザ定義線種設定（ダイアログモードでのみ動作）

【機能】

ユーザ定義線種の設定・修正を行います。
 SXF 線種 1 ～ 32 が ACAD 線種 1 ～ 32 に 1 対 1 で割当てられている場合のみ使用可能です。

【構文】

SXF/USER_LFT

【ユーザ定義線種設定ダイアログ】

線種名は必須です。
描画 n と空白 n は必ず対で 1 組は設定してください。

線種表示
☐ 固定 固定ピッチで線種を表示します。
☐ 縮尺反映 設定されたピッチで線種を表示します。

7.2.13 ユーザ定義線幅設定（ダイアログモードでのみ動作）

【機能】
ユーザ定義線幅の設定・修正を行います。
SXF 線幅 1 ～ 16 が ACAD 線幅 1 ～ 16 に 1 対 1 で割当てられている場合のみ使用可能です。

【構文】
SXF/USER_LWT

【ユーザ定義線幅設定ダイアログ】

線幅表示
☐ 非表示 線幅は表示しません。
☐ 線幅番号 × ピッチで表示 線幅を線幅番号 × 線幅ピッチで表示します。
☐ 実 線幅で表示 設定された線幅で表示します。

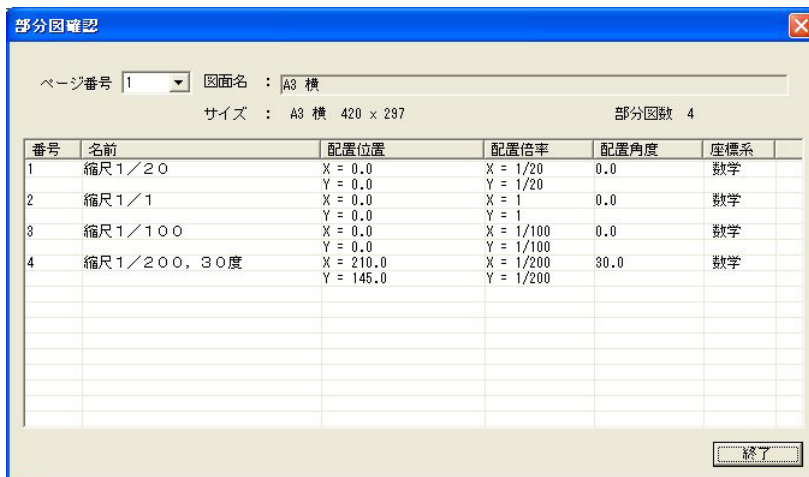
7.2.14 部分図一覧表示（ダイアログモードでのみ動作）

【機能】

部分図の一覧を表示します。

【構文】

SXF/PIC_LIST

【部分図一覧表示ダイアログ】


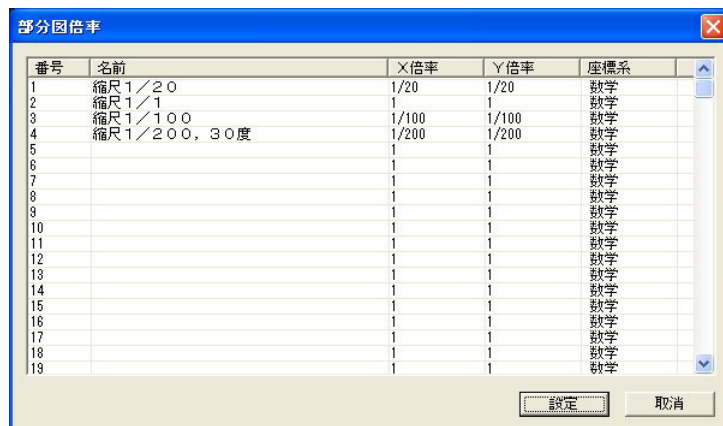
7.2.15 部分図の倍率設定（ダイアログモードでのみ動作）

【機能】

部分図の倍率、座標系を設定します。

【構文】

SXF/PIC_SCF

【部分図倍率設定ダイアログ】


【注意事項】

測地座標の設定はできますが測地座標での入力できません。

7.2.16 各名称の一覧を表示する（ダイアログモードでのみ動作）

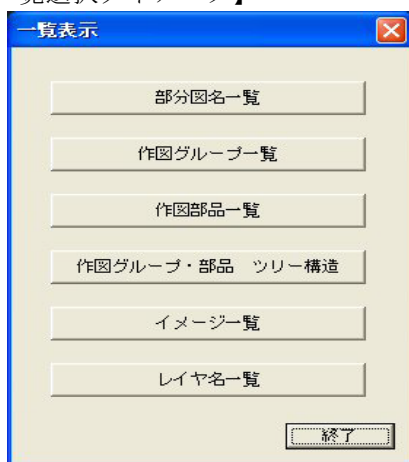
【機能】

部分図名、作図グループ名、作図部品、作図グループ・部品ツリー構造、イメージ及びレイヤ名の一覧を表示します。

【構文】

SXF/LIST

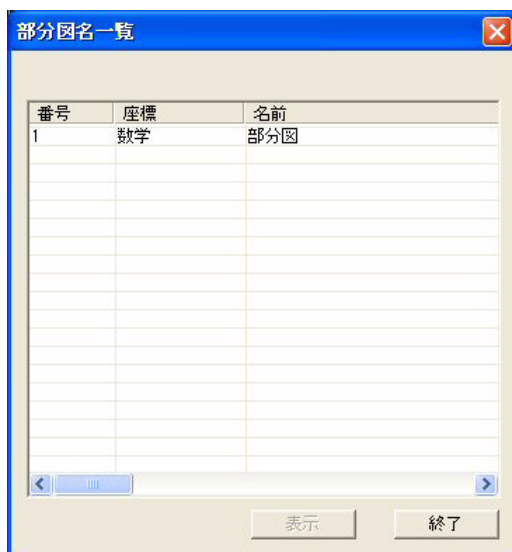
【一覧選択ダイアログ】



・部分図名一覧

部分図名 (ピクチャ名) の一覧を表示します。

【部分図名一覧ダイアログ】

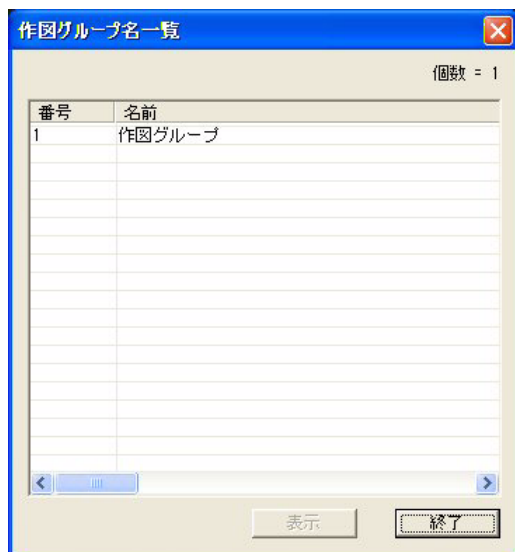


部分図名を選択して「表示」を選択すると部分図（ピクチャ）を点滅表示します。

・作図グループ一覧

作図グループ名の一覧を表示します。

【作図グループ名一覧ダイアログ】

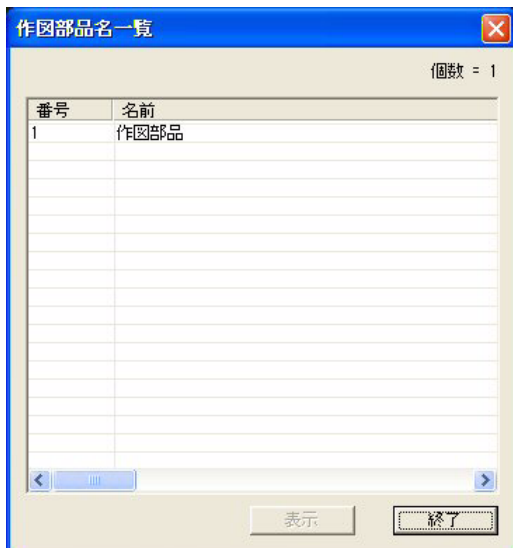


作図グループ名を選択して「表示」を選択すると選択された作図グループを点滅表示します。

・作図部品一覧

作図部品名の一覧を表示します。

【作図部品名一覧ダイアログ】



作図部品名を選択して「表示」を選択すると選択された作図部品を点滅表示します。

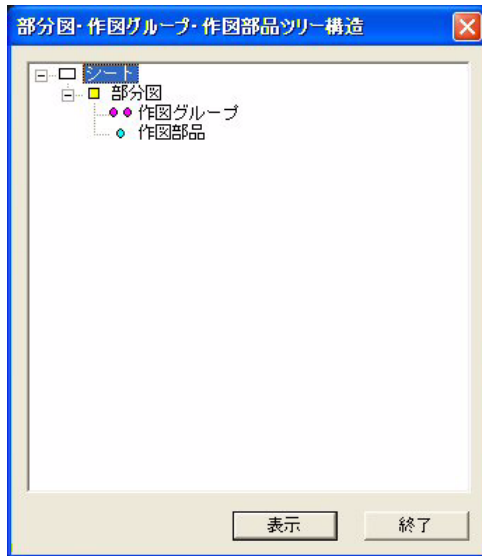
・作図グループ・部品ツリー構造

作図グループ・部品ツリー構造を表示します。

シート（図面配置）での部分図（ピクチャ）、作図グループ、作図部品のツリー構造を表示します。

図面配置でコマンドを実行してください。

【ツリー構造表示ダイアログ】

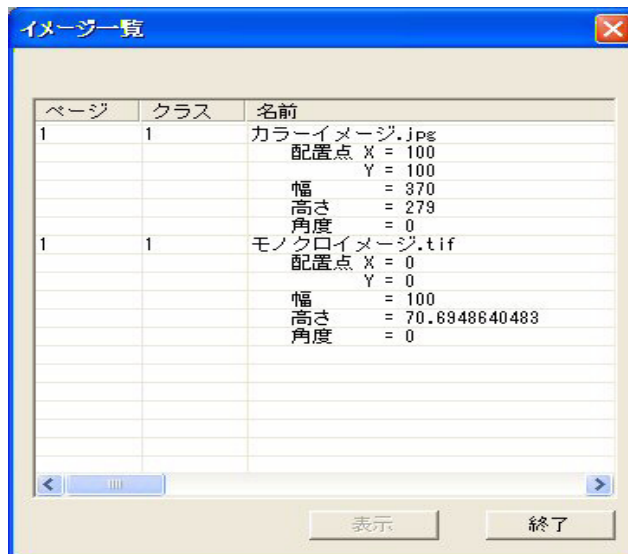


部分図（ピクチャ）、作図グループ、作図部品を選択して「表示」を選択すると選択された部分図（ピクチャ）、作図グループ、作図部品を点滅表示します。

・イメージ一覧

図面配置ピクチャに配置されているイメージ（TIFF、JPEG）の一覧を表示します。

【イメージ一覧表示ダイアログ】



イメージを選択して「表示」を選択すると選択されたイメージを点滅表示します。

・レイヤ名一覧

レイヤ名の一覧を表示します。

レイヤはクラスに割当てられているものとして処理しています。

【レイヤ名一覧ダイアログ】



7.2.17 複合図形（作図グループ・作図部品）配置（ダイアログモードのみ）

【機能】

複合図形（作図グループ・作図部品）を配置します。
レイヤはクラスに割当てられているものとして処理しています。

【構文】

SXF/SFIG_REF

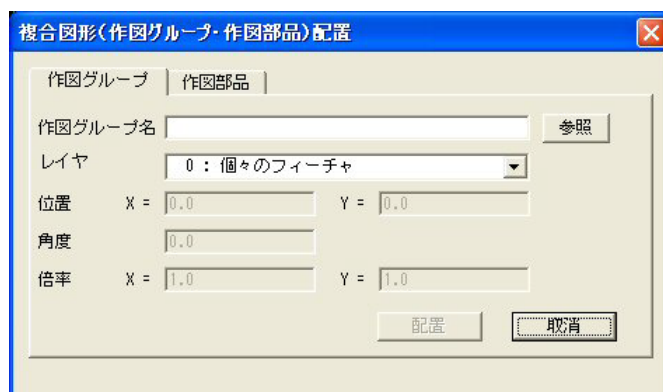
・ 作図グループ配置

作図グループを配置します。

作図グループは、配置位置 (0, 0)、倍率 (1, 1)、角度0度で配置されます。

作図グループは1度しか配置することができません。

【複合図形配置ダイアログ】



〔作図グループ〕を選択します。

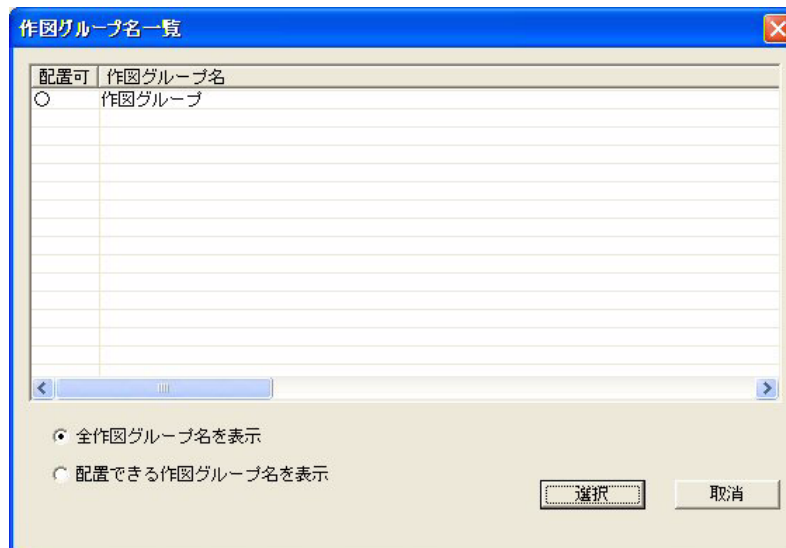
〔▼レイヤ〕を選択します。

レイヤ0は、作図グループ内の個々の図形で表示の制御を行います。

レイヤ n は、作図グループで表示の制御を行います。

〔作図グループ名〕入力又は〔『参照』〕で名前を入力します。
〔『配置』〕で作図グループを配置します。

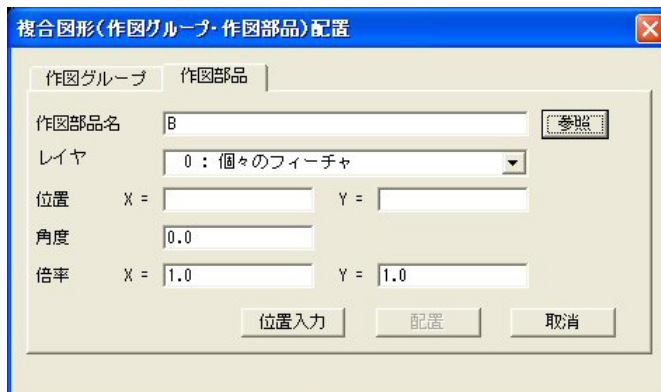
【作図グループの参照】



〔○全作図グループ名を表示〕又は〔○配置できる作図グループ名を表示〕で一覧を表示します。
○印の付いた〔作図グループ〕を選択します。既に配置されている作図グループは選択できません。
〔『選択』〕で作図グループが選択されます。

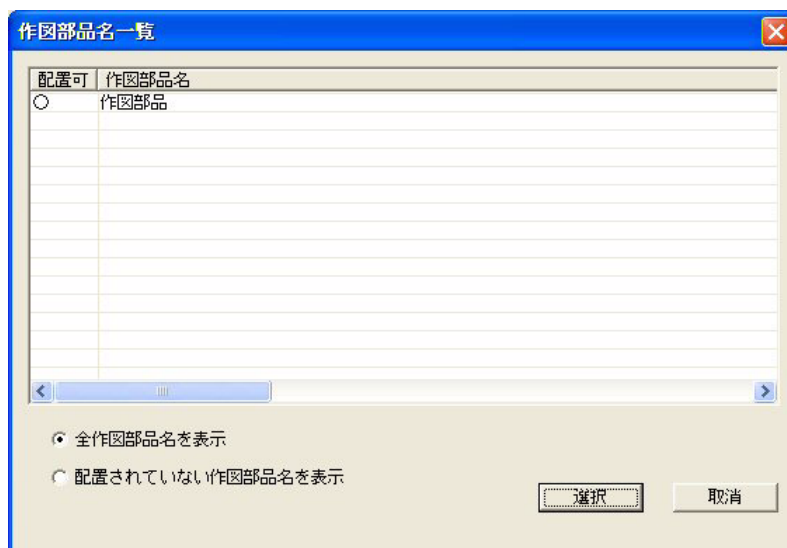
・作図部品配置

作図部品を配置します。
作図部品は、配置位置：任意、倍率：任意、角度：任意で配置できます。
【複合図形配置ダイアログ】



〔作図部品〕を選択します。
〔▼レイヤ〕を選択します。
レイヤ0は、作図部品内の個々の図形で表示の制御を行います。
レイヤnは、作図部品で表示の制御を行います。
〔作図部品名〕又は〔『参照』〕で名前を入力します。
〔位置〕、〔角度〕、〔倍率〕を入力します。
〔『配置』〕で作図部品を配置します。
〔位置〕入力のかわりに〔『位置入力』〕を選択するとデジタイズ入力で配置位置が入力できます。

【作図部品の参照ダイアログ】



〔○全作図部品を表示〕又は〔○配置されていない作図部品名を表示〕で一覧を表示します。

〔作図部品〕を選択します。すべての作図部品が配置可能です。

〔『選択』〕で作図部品を選択します。

7.2.18 複合図形（作図グループ・作図部品）作成・編集（ダイアログモードのみ）

【機能】

複合図形（作図グループ・作図部品）を作成、編集、削除します。

操作上の注意事項

- 1) 図面配置では動作しません。通常のピクチャで動作します。
- 2) ピクチャの状態で作成の内容が異なります。
 - ・ピクチャで作図グループ（作図部品）を編集中の場合
更新又はクリア処理になります。
 - ・ピクチャにデータがある場合
作成処理になります。
 - ・ピクチャにデータがない場合
編集又は削除処理になります。
 編集を行った場合ピクチャで作図グループ（作図部品）が編集集中であることを記憶します。

【構文】

SXF/SFIG_GEN

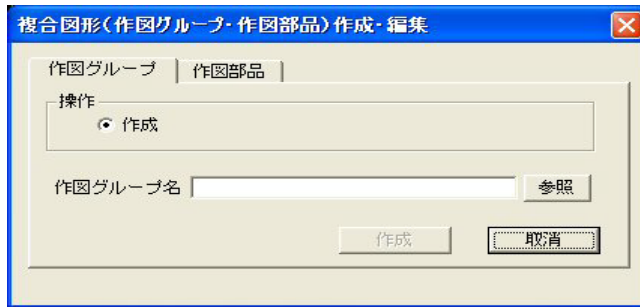
・作図グループ（作図部品）の作成

作図グループ（作図部品）を作成します。

現ピクチャが作図グループ（作図部品）の編集集中でなく

ピクチャに図形がある場合作成処理になります。

【複合図形作成・編集ダイアログ】



〔作図グループ〕（〔作図部品〕）を選択します。
 〔○作成〕を選択します。
 〔作図グループ名〕（〔作図部品名〕）入力又は〔『参照』〕で名前を入力します。
 〔『作成』〕で作図グループ（作図部品）を作成します。

【注意事項】

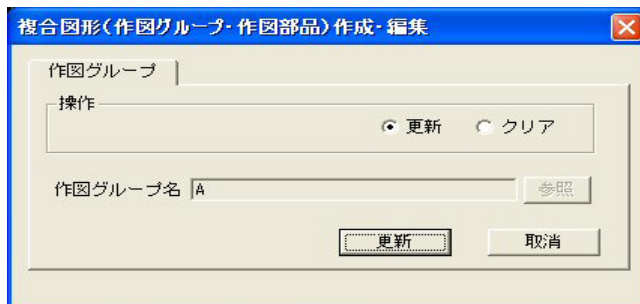
作成では次の点に注意してください。

1. 作図グループ・作図部品共配置するピクチャ倍率と同じピクチャ倍率で図形を作成してください。（文字、マーク等の大きさに影響します）
2. 作図グループ内に作図グループ・作図部品を配置することができます。
3. 作図部品内に作図部品は配置できますが作図グループは配置できません。

・ 作図グループ（作図部品）の更新

編集中の作図グループ（作図部品）を更新します。
 現ピクチャが作図グループ（作図部品）の編集の場合、更新（又はクリア）処理になります。

【複合図形作成・編集ダイアログ】

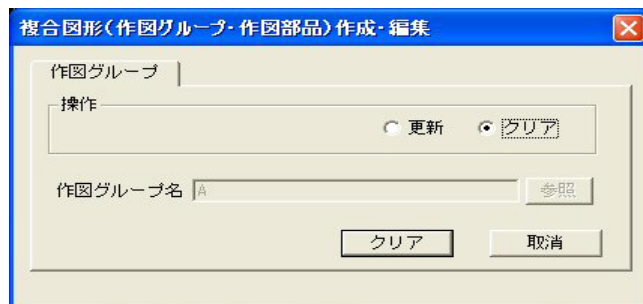


〔○更新〕を選択します。
 〔『更新』〕で作図グループ（作図部品）を更新します。

・ 作図グループ（作図部品）のクリア

編集中の作図グループ（作図部品）をクリア（現ピクチャの名前及び図形を削除）します。
 現ピクチャが作図グループ（作図部品）の編集の場合、クリア（又は更新）処理になります。

【複合図形作成・編集ダイアログ】



〔○クリア〕を選択します。

〔『クリア』〕で編集中の作図グループ（作図部品）をクリアします。

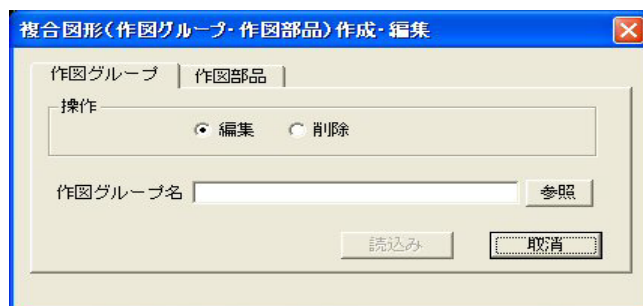
・作図グループ（作図部品）の編集

作図グループ（作図部品）を編集します。

ピクチャに図形がない場合、編集（又は削除）処理になります。

ピクチャ倍率を配置するピクチャの倍率と同じにしてから編集を行ってください。（文字、マーク等の大きさに影響します）

【複合図形作成・編集ダイアログ】



〔作図グループ〕（〔作図部品〕）を選択します。

〔○編集〕を選択します。

〔作図グループ名〕（〔作図部品名〕）入力又は〔『参照』〕で名前を入力します。

〔『読み込み』〕で作図グループ（作図部品）を現ピクチャに読み込みます。

ピクチャが作図グループ（作図部品）の編集中的であることを記憶します。

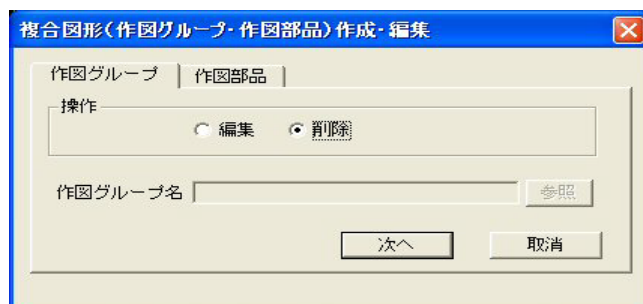
図形の編集を行った後、作図グループ（作図部品）の更新を行って修正を反映させます。

・作図グループ（作図部品）の削除

作図グループ（作図部品）を削除します。

ピクチャに図形がない場合、削除（又は編集）処理になります。

【複合図形作成・編集ダイアログ】

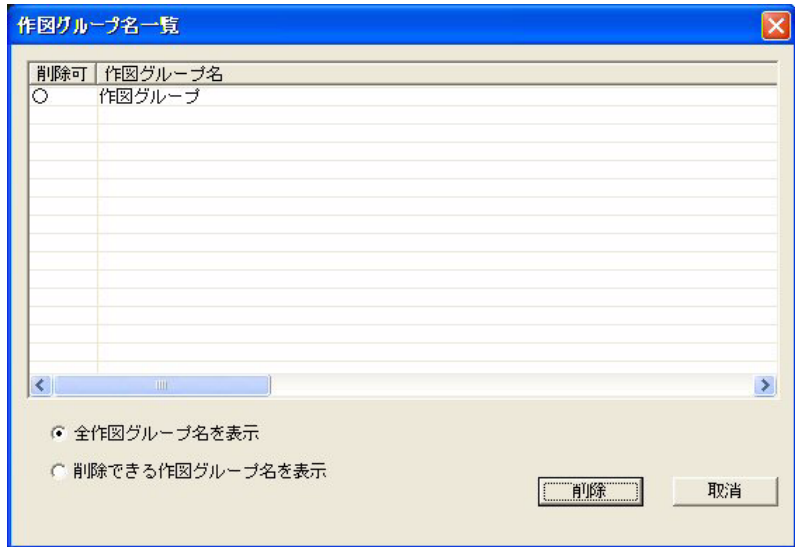


〔作図グループ〕（〔作図部品〕）を選択します。

〔○削除〕を選択します。

〔『次へ』〕で作図グループ（作図部品）名の一覧を表示します。

削除する作図グループ（作図部品）名の選択
【作図グループ名一覧ダイアログ】



〔○全作図グループ名を表示〕〔○全作図部品名を表示〕又は
〔○削除できる作図グループ名を表示〕〔○削除できる作図部品名を表示〕で
作図グループ（作図部品）名の一覧を表示します。
削除する〔作図グループ〕〔作図部品〕名を選択します。
〔削除〕で作図グループ（作図部品）を削除します。
配置されている作図グループ（作図部品）は削除できません。

7.2.19 複合図形（作図部品）配置情報修正（ダイアログモードのみ）

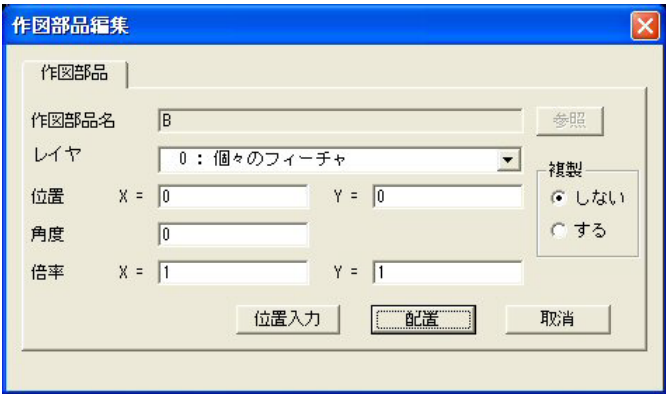
【機能】

配置されている作図部品の配置情報（位置、角度、倍率）を修正又は複製します。
レイヤはクラスに割当てられているものとして処理しています。

【構文】

SXF/SFIG_EDT	IS
--------------	----

IS : 修正する作図部品を選択します。
【作図部品編集ダイアログ】



複製 [○しない] [○する] を選択します。
 [▼レイヤ] を選択します。
 レイヤ 0 は、作図部品内の個々の図形で表示の制御を行います。
 レイヤ n は、作図部品で表示の制御を行います。
 [位置]、[角度]、[倍率] を入力します。
 [『配置』] で配置情報を修正 (複製) します。
 [位置] を入力するかわりに [『位置入力』] を選択すると
 デジタイズ入力での位置入力ができます。

7.2.20 複合図形 (作図グループ・作図部品) 削除

【機能】

配置されている作図グループ・作図部品を削除します。

【構文】

SXF/SFIG_DEL	IS	<CE>
--------------	----	------

IS : 削除する作図グループ・作図部品を選択します。

7.2.21 複合図形 (作図グループ・作図部品) 分解

【機能】

配置されている作図グループ、作図部品を図形要素に分解します。
 属性が付加されている作図グループ、作図部品は分解できません。
 分解する場合は属性を削除してください。

【構文】

SXF/SFIG_BRK	[IScrv]+	<CE>
--------------	----------	------

IScrv : 分解する作図グループ・作図部品を選択します。
 <CE> : 分解します。

7.2.22 図形を移動・回転・縮小拡大する

【機能】

複合図形 (作図グループ・作図部品) を含めたアイテムを移動・回転・縮小・拡大します。
 複製した作図グループは図形要素に展開されます。また異縮尺の場合作図部品は図形要素に
 展開されます。

備考 : 作図グループは原点 <0,0>、角度 0 度、倍率 <1,1> で 1 回だけ配置できます。
 作図部品は原点任意、角度任意、倍率任意で複数回配置できます。

【構文】

SXF/MOVE	$\left[\begin{array}{c} \text{OFF} \\ \text{DUP} \end{array} \right]$	[ORG org]	$\left[\begin{array}{c} \text{SCF xy} \\ \text{XSCF x} \\ \text{YSCF y} \end{array} \right]$
		[ANG ang]	$[\text{IS}] + \quad \langle \text{CE} \rangle 1 \quad \left\{ \begin{array}{c} \text{P} \\ \langle \text{CE} \rangle 2 \end{array} \right\}$

複製する／しないを選択します。

OFF : 複製しない。(既定値)

DUP : 複製する。

元図の基準点を指定します。

ORG org : 移動・縮小・拡大・回転の基準点を入力します。(既定値 : 0.0, 0.0)

移動（複製）後の倍率・角度・位置を入力します。

SCF xy : 縮小・拡大倍率を入力します。(X=Y) (既定値 : 1.0)

XSCF x : X 倍率を入力します。

YSCF y : Y 倍率を入力します。

ANG ang : 回転角度（度）を入力します。(既定値 : 0.0)

移動（複製）するアイテムを選択します。

[IS]+ : アイテムを選択します。

<CE>1 : アイテムを確定します。

選択したアイテムを囲う外形矩形をラバーバンドします。

移動（複製）する位置を入力します。

P : 移動後の位置を入力します。

<CE>2 : 移動しないで縮小・拡大、回転します。

7.2.23 表示順

【機能】

アイテムの表示順を変更します。

表示はアイテムの並び順に表示します。

表示順のとおりに表示させるためには SXF 表示コマンド

SXF/RPT、SXF/ZOOM、SXF/ZOOMALL、SXF/ZOOMALLVIE、SXF/ZOOMUP、SXF/ZOOMDOWN、SXF/PAN
を使用してください。

【注意事項】

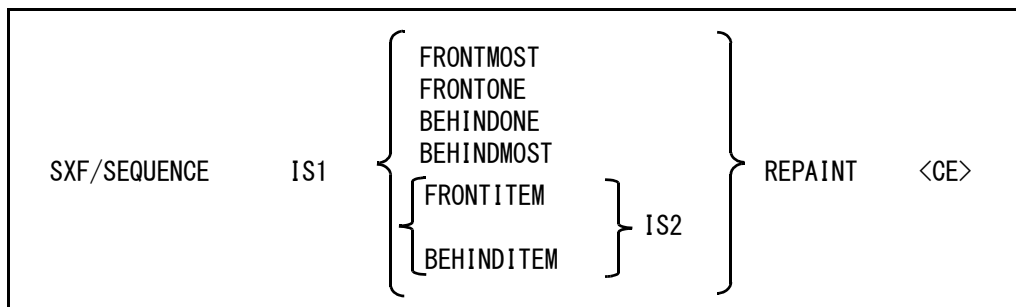
表示順の変更操作ではモデルにアイテム数と同じ空き容量が必要となります。

表示順の操作を行っていてアイテムの空き容量がアイテム数より少なくなると

表示順の操作はできなくなります。その場合、モデル圧縮またはモデル保存を

行って空き容量を増やしてください。

【構文】



表示順を変更するアイテムを選択します。

IS1 : 表示順を変更するアイテムを選択します。

表示位置を選択します。

FRONTMOST : 最前面に表示します。
 FRONTONE : 1つ前面に表示します。
 BEHINDMOST : 最後面に表示します。
 BEHINDONE : 1つ後面に表示します。
 FRONTITEM : 指定したアイテムの前面に表示します。
 BEHINDITEM : 指定したアイテムの後面に表示します。
 IS2 : 表示順の基準になるアイテムを選択します。

確認表示します。

REPAINT : 画面を再表示します。(確定の前に表示順を確認します。)

確定します。

<CE> : 表示順を確定します。
 確定しないでコマンドを終わらせると表示順は変更されません。

7.2.24 表題欄設定（ダイアログモードでのみ動作）

【機能】

表題欄フィーチャーの各項目の設定（変更・削除）を行います。

表題欄フィーチャーは SXF Ver3.0、Ver3.1 では出力できますが Ver2.0 では出力できません。

【構文】

SXF/TTL_SET

【表題欄ダイアログ】

項目	内容
事業名	
工事名	
契約区分	
図面名	
図面種別	
尺度	
受注会社名	
発注事業所名	

図面番号 図面総数
 図面作成日 0 年 1 月 1 日 (西暦)

〔事業名〕、〔工事名〕、〔契約区分〕、〔図面名〕、〔図面番号〕、〔図面総数〕、〔図面種別〕、〔尺度〕、〔図面作成日〕（西暦年・月・日）、〔受注会社名〕、〔受注事務所名〕を入力します。

〔事業名〕、〔工事名〕、〔契約区分〕、〔図面名〕、〔図面種別〕、〔尺度〕、〔受注会社名〕、〔受注事務所名〕の入力は次のダイアログが表示されます。

【文字列入力ダイアログ】

事業名を入力してください

〔事業名〕、〔工事名〕、〔契約区分〕、〔図面名〕、〔図面種別〕、〔尺度〕、〔受注会社名〕、〔受注事務所名〕は複数行入力できます。

【備考】

図面中の文字列に表題欄属性が付加されている場合、表題欄を修正（削除）した表題欄に対応する文字列も修正（削除）されます。

7.2.25 表題欄属性（ダイアログモードでのみ動作）

【機能】

文字列に表題欄属性の付加及び削除を行います。
表題欄属性が付加された文字列は表題欄フィーチャーに反映されます。

・ 表題欄属性付加

【機能】

文字列に表題欄属性を付加します。
表題欄属性が付加された文字列は表題欄フィーチャーに登録されます。

【構文】

SXF/TTL_CRE	[ATTR_CRE]	IS
-------------	------------	----

IS : 表題欄属性を付加する文字列を選択します。
この後表題欄属性一覧ダイアログが表示されます。

表題欄属性一覧から種別を選択します。

事業名
工事名
契約区分
図面番号
図面総数
図面種別
尺度
図面名
年月日
会社名
事務所名

の中から種別を選びます。

『選択』をクリックします。表題欄属性が付加されます。

・ 表題欄属性追加

【機能】

表題欄の1つの項目が複数行の場合、2行目以降に表題欄属性を付加します。
表題欄属性が付加された文字列は表題欄フィーチャーに追加されます。

【構文】

SXF/TTL_CRE	ATTR_ADD	IS1	IS2	<CE>
-------------	----------	-----	-----	------

IS1 : 表題欄属性に追加する文字列を選択します。

IS2 : 既に表題欄属性が付加されている文字列を選択します。

<CE> : 表題欄属性を追加します。

・ 表題欄属性削除

【機能】

表題欄の1つの項目が複数行の場合、2行目以降に表題欄属性を付加します。
表題欄属性が削除された文字列は表題欄フィーチャーから削除されます。

【構文】

SXF/TTL_CRE	ATTR_DEL	IS	<CE>
-------------	----------	----	------

IS : 既に表題欄属性が付加されている文字列を選択します。

<CE> : 表題欄属性を削除します。

7.2.26 属性設定（ダイアログモードでのみ動作）

【機能】

属性を設定します。
ピクチャ番号（部分図）、クラス番号（レイヤ）、レビジョン番号（色）、線種、線幅の設定を行います。

【構文】

SXF/ATTR

部分図（ピクチャ）、レイヤ（クラス）、色（レビジョン）、線種、線幅を設定します。

【属性設定ダイアログ】



7.2.27 等高線作成

【機能】

選択したアイテムに等高線属性を付加、編集または削除します。

【構文】

SXF/CNT	ALTITUDE	h	$\left\{ \begin{array}{l} \text{NUM} \\ \text{NME name} \end{array} \right\}$	ATTR_UNIT	u	[IS]+	<CE>
---------	----------	---	---	-----------	---	-------	------

ALTITUDE h : 高さを入力します。
NUM : 名称は 8 桁 (00000001 ～) の通し番号にします。(既定値)
NME name : 名称を入力します。
ATTR_UNIT u : 単位を入力します。(m、cm または mm 既定値 : m)
IScrv : 等高線にするアイテムを選択するします。
選択できるアイテムは、直線、円弧、自由曲線、ストリング、文字列。

7.2.28 レイヤの表示・選択マスク設定（ダイアログモードでのみ動作）

【機能】

レイヤの表示・選択マスクを設定します。

【構文】

SXF/LAY

・表示マスク

表示マスクを設定します。

『表示クラス』(CLS/DSP)『全でのピクチャへの設定』(ALLPIC) と同等の処理を行います。

【表示レイヤマスクダイアログ】



[[表示マスク]] を選択します。

〔レイヤ名〕を選択します。

[[設定]] を選択して表示レイヤを設定します。

・選択マスク

選択マスクを設定します。

『選択クラス』(CLS/SEL) と同等の処理を行います。

図面データを編集する場合に朱書きデータを保護し、朱書きデータを編集する場合に図面データを保護する意味合いがあります。

【選択レイヤマスクダイアログ】

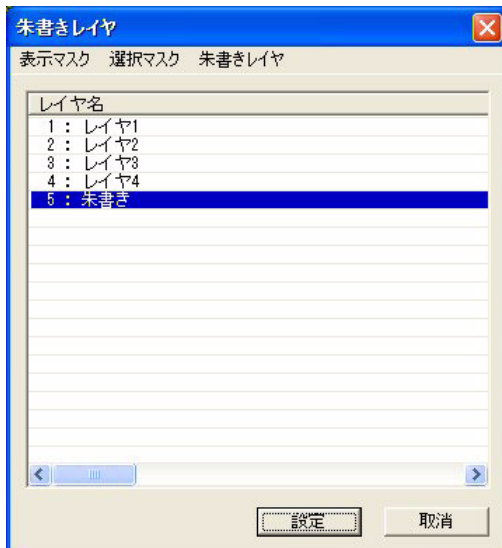


- [[選択マスク]] を選択します。
- [[レイヤ名]] を選択します。
- [[設定]] を選択して選択レイヤを設定します。

・朱書きレイヤ

- 朱書きレイヤを設定します。
- 朱書きレイヤの図形は図面ファイルには出力されません。
- 図面ファイルとは別の朱書きファイルに出力されます。

【朱書きレイヤ】



- [[朱書きレイヤ]] を選択します。
- [[レイヤ名]] を選択します。
- [[設定]] を選択して朱書きレイヤを設定します。
- 朱書きレイヤ名を変更した場合は再度朱書きレイヤの設定を行ってください。

7.2.29 図形の確認（ダイアログモードでのみ動作）

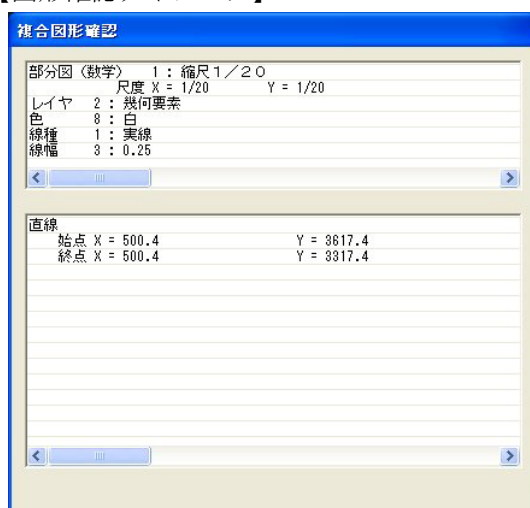
【機能】

図形の確認を行います。

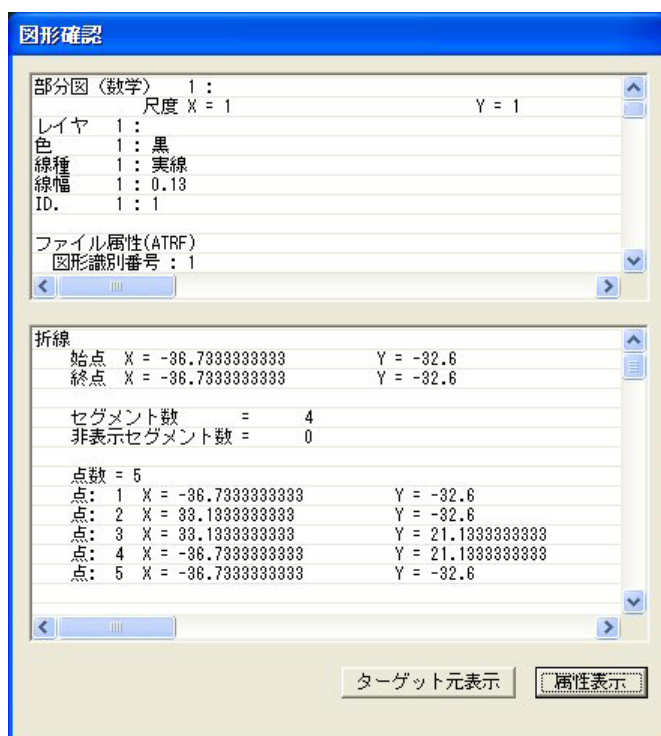
【構文】

SXF/VER	$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{ACTL_SHOW} \\ \text{ACTL_NOSHOW} \end{array} \right\} \\ \text{ATTR_LIST} \end{array} \right\} \left[\text{IS} \right]^+ \left\{ \begin{array}{l} \text{<CE>} \end{array} \right\}$
---------	--

- ACTL_SHOW : 非表示の既定義ハッチングを表示します。
- ACTL_NOSHOW : 非表示の既定義ハッチングは消去しません。
- ATTR_LIST : 属性一覧を表示します。
- IS : 確認するアイテムを選択します。
アイテムの情報がダイアログで表示されます。
- <CE> : 終了します。

【図形確認ダイアログ】


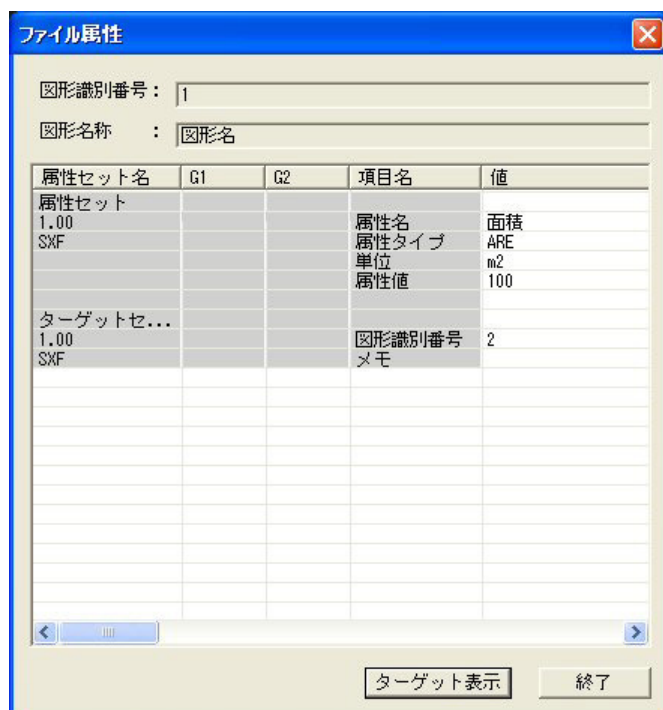
【図形確認ダイアログ（ファイル属性が付加されている場合）】



『ターゲット元表示』 : 選択したアイテムがターゲットになっている場合、
このアイテムをターゲットにしているアイテムを表示します。

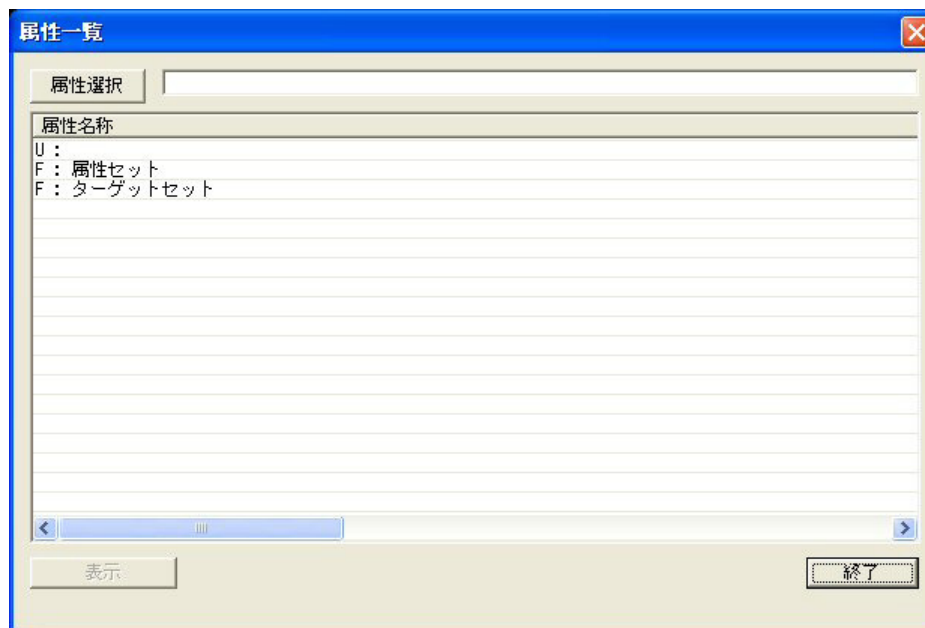
『属性表示』 : ファイル属性を表示します。

【ファイル属性表示ダイアログ】



『ターゲット表示』 : 選択したアイテムにターゲットが設定されている場合、ターゲットアイテムを表示します。

【属性一覧ダイアログ】

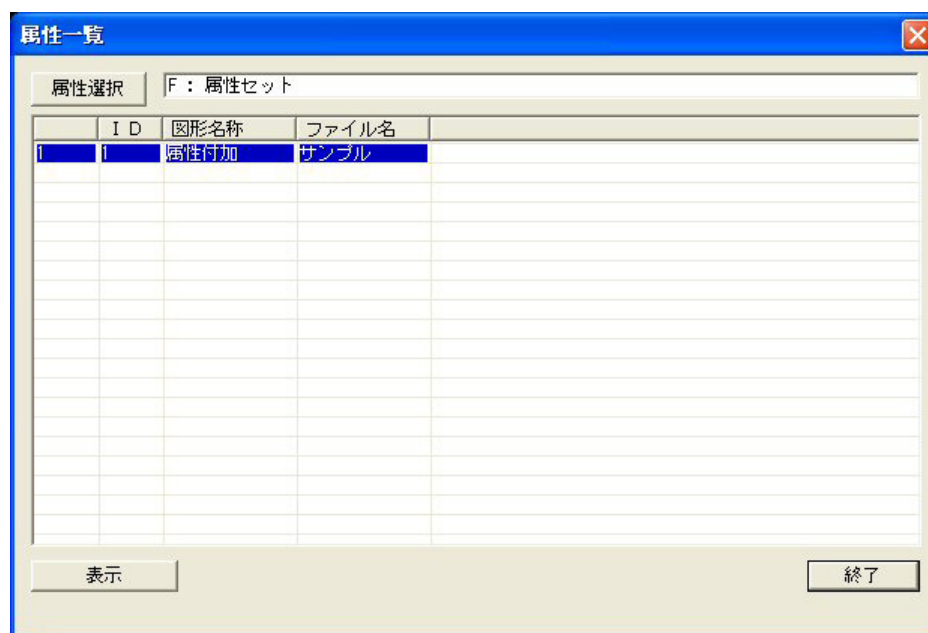


『属性選択』 : 属性名称の一覧を表示します。
 F : で始まる行はファイル属性です。
 属性セット名を表示します。
 U : で始まる行は単一属性です。
 属性名を表示します。
 S : で始まる行は文字属性です。
 属性名を表示します。

一覧から選択 : 一覧から属性を表示する属性名称を選択します。

『終了』 : 属性一覧処理を終了します。

【属性表示ダイアログ】



一覧から選択	:	一覧から属性を選択します。
『表示』	:	選択された属性が付加されているアイテムをハイライト表示します。
『終了』	:	属性一覧処理を終了します。

7.2.30 画面再表示

【機能】

画面を再表示します。

【構文】

SXF/RPT

7.2.31 矩形領域を拡大表示

【機能】

対角 2 点で囲われた矩形領域を拡大表示します。

【構文】

SXF/ZOOM [P1 P2]+ <CE>

P1 P2 : 矩形をの対角 2 点で指示します。
矩形の範囲がビューポートいっぱいに表示されます。

続けて何度でも範囲を指定できます。

7.2.32 拡大表示

【機能】

拡大表示します。

【構文】

SXF/ZOOMUP

表示する領域の面積が半分になるように拡大します。

倍率 = $\sqrt{2} = 1.414$

7.2.33 縮小表示

【機能】

縮小表示します。

【構文】

SXF/ZOOMDOWN

表示する領域の面積が倍になるように縮小します。

倍率 = $1/\sqrt{2} = 0.7071$

7.2.34 全体表示

【機能】

全体を表示します。

【構文】

SXF/ZOOMALL

アイテムの最大外形を表示範囲として画面に表示します。

図面配置のときは図面枠の最大外形を表示範囲として画面に表示します。

7.2.35 図形領域全体表示

【機能】

図形範囲全体を表示します。

【構文】

SXF/ZOOMALLVIE

アイテムの最大外形を表示範囲として画面に表示します。
 図面配置のときは図面枠と配置されているウィンドウのウィンドウ枠の最大外形を表示範囲として画面に表示します。

7.2.36 表示位置移動

【機能】

表示位置を移動します。

【構文】

SXF/PAN [P1 P2]+ <CE>

P1 : 移動する基準となる点を指示します。
 P2 : P1 の移動後の点を指示します。
 P1 の位置を P2 の位置へ移動します。続けて操作できます。

7.2.37 図面印刷（ダイアログモードでのみ動作）

【機能】

図面範囲を印刷します。

【構文】

SXF/PRINT

PLOT/PRINT とほぼ同じ機能ですが違いは、

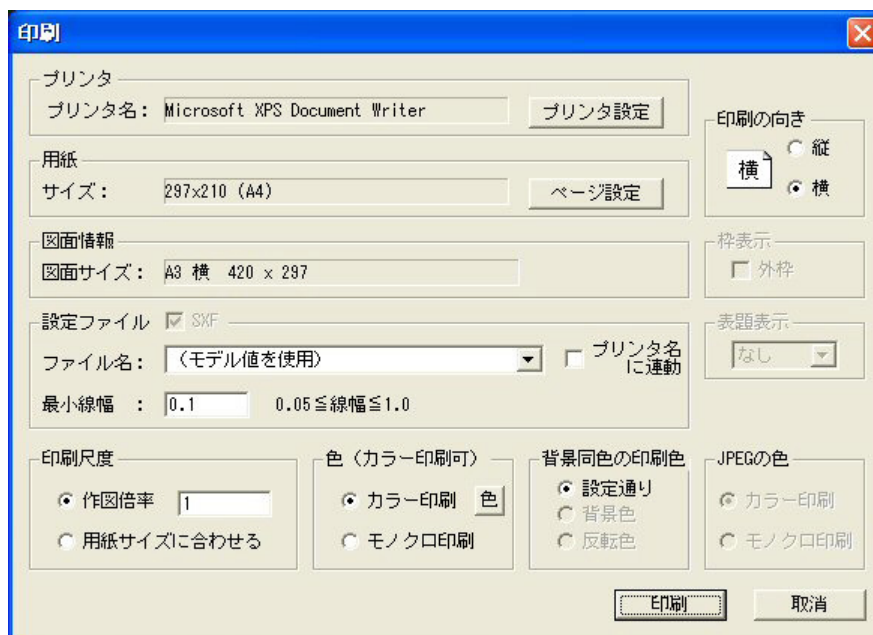
1. 現在の図面配置ページを印刷します。
2. アイテムの並び順に印刷します。
3. ペン割当はカラーに固定します。
4. ペンの数は 2 5 6 に固定します。

です。

設定ファイルのサンプル SXFPRINT.POP は、sample/PLOT にあります。

PLOT/PRINT、設定ファイルについては、プロッタ・ガイドを参照してください。

図面印刷ダイアログ



『プリンタ設定』

『ページ設定』

[印刷の向き]

[○縦]

[○横]

設定ファイル

[□ SXF]

[▼ファイル名]

[最小線幅]

[□ プリンタに連動]

印刷倍率

[○作図倍率]

[○用紙サイズに合わせる]

色

[○カラー印刷]

[『色』]

[○モノクロ印刷]

背景同色の印刷色

[○設定通り]

[○背景色]

[○反転色]

JPEG の色

[○カラー印刷]

[○モノクロ印刷]

『印刷』

『取消』

出力するプリンタを選択します。注1)

用紙サイズ、余白等を設定します。

印刷の向きを選択します。

用紙の縦方向に印刷します。

用紙の横方向に印刷します。

設定ファイルを設定します。

設定ファイルを SXF 用で使します。注2)

設定ファイルを選択します。注3)

設定ファイルで (モデル値を使用) を選択した場合

最小線幅を指定します。

設定ファイルをプリンタ名に連動させます。

印刷倍率を設定します。

倍率を指定して印刷します。

用紙サイズに合わせて印刷します。

印刷色を設定します。

カラーで印刷します。

色の RGB 値を変更します。

注) 変更した内容は印刷実行中のみ有効です。

モノクロで印刷します。

背景色が白/黒の場合の白/黒色の印刷指定を選択します。

注) 背景が黒の場合、白は黒で印刷します。

設定されたとおりに印刷します。

背景色と同じ色 (白/黒) で印刷します。

背景色の反転色 (黒/白) で印刷します。

モノクロ印刷の場合に JPEG の色を選択します。

カラーで印刷します。

モノクロで印刷します。

印刷を開始します。

印刷を取消します。

注1) 選択したプリンタが印刷可能でない場合は正しく印刷は行えません。

- 注2) チェックがオンの場合、設定ファイルを SXF 用として使用します。
 既定義色 (1 ~ 16) は、設定ファイルの設定を使用します。
 ユーザ定義色 (17 ~ 256) は、モデルに設定されている色を使用します。
 既定義線幅 (1 ~ 9) は、設定ファイルの設定を使用します。
 ユーザ定義線幅 (11 ~ 16) は、モデルに設定されている線幅を使用します。
 定義がない場合は線幅 1 を使用します。
 チェックがオフの場合、色・線幅は設定ファイルの設定を使用します。
 注3) 設定ファイルがない場合または『(モデル値を使用)』が選択された場合は
 モデルに設定されている色、線幅を使用します。

7.2.38 画面印刷（ダイアログモードでのみ動作）

【機能】

画面に表示されている範囲を印刷します。

【構文】

SXFCOPY/PRINT

HCOPY/PRINT とほぼ同じ機能ですが違いは、

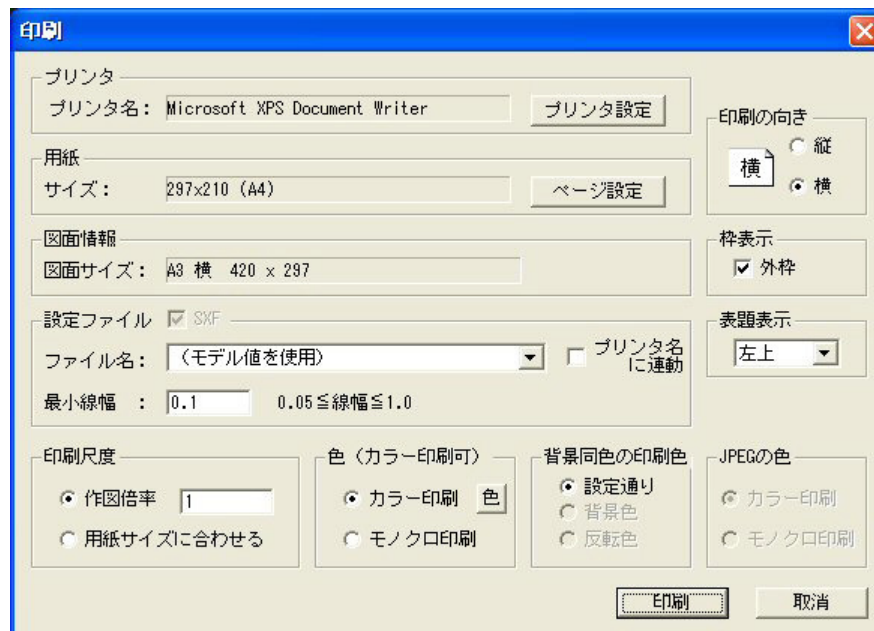
1. 現在の図面配置ページの画面に表示されている部分を印刷します。
1. アイテムの並び順に印刷します。
2. ペン割当はカラーに固定します。
3. ペンの数は 2 5 6 に固定します。

です。

設定ファイルのサンプル SXFPRINT.POP は、sample/PLOT にあります。

HCOPY/PRINT、設定ファイルについては、プロッタ・ガイドを参照してください。

画面印刷ダイアログ



『プリンタ設定』

出力するプリンタを選択します。注1)

〔『ページ設定』〕	用紙サイズ、余白等を設定します。
印刷の向き	印刷の向きを選択します。
〔○縦〕	用紙の縦方向に印刷します。
〔○横〕	用紙の横方向に印刷します。
枠表示	枠表示を設定します。
〔□外枠〕	枠を表示します。
表題表示	表題表示を設定します。
〔▼表示位置〕	表示位置を選択します。
設定ファイル	設定ファイルを設定します。
〔□ SXF〕	設定ファイルを SXF 用で使⽤します。注2)
〔▼ファイル名〕	設定ファイルを選択します。注3)
〔最小線幅〕	設定ファイルで（モデル値を使用）を選択した場合 最小線幅を指定します。
〔□プリンタに連動〕	設定ファイル名をプリンタ名に連動させます。
印刷倍率	印刷倍率を設定します。
〔○作図倍率〕	倍率を指定して印刷します。
〔○用紙サイズに合わせる〕	用紙サイズに合わせて印刷します。
色	印刷色を設定します。
〔○カラー印刷〕	カラーで印刷します。
〔『色』〕	色の RGB 値を変更します。
	注) 変更した内容は印刷実行中のみ有効です。
〔○モノクロ印刷〕	モノクロで印刷します。
背景同色の印刷色	背景色が白／黒の場合の白／黒色の印刷指定を選択します。
	注) 背景が黒の場合、白は黒で印刷します。
〔○設定通り〕	設定されたとおりに印刷します。
〔○背景色〕	背景色と同じ色（白／黒）で印刷します。
〔○反転色〕	背景色の反転色（黒／白）で印刷します。
JPEG の色	モノクロ印刷の場合に JPEG の色を選択します。
〔○カラー印刷〕	カラーで印刷します。
〔○モノクロ印刷〕	モノクロで印刷します。
〔『印刷』〕	印刷を開始します。
〔『取消』〕	印刷を取消します。

注 1) 選択したプリンタが印刷可能でない場合は正しく印刷は行えません。

注 2) チェックがオンの場合、設定ファイルを **SXF** 用として使⽤します。

既定義色（1 ～ 16）は、設定ファイルの設定を使⽤します。

ユーザ定義色（17 ～ 256）は、モデルに設定されている色を使⽤します。

既定義線幅（1 ～ 9）は、設定ファイルの設定を使⽤します。

ユーザ定義線幅（11 ～ 16）は、モデルに設定されている線幅を使⽤します。

定義がない場合は線幅 1 を使⽤します。

チェックがオフの場合、色・線幅は設定ファイルの設定を使⽤します。

注 3) 設定ファイルがない場合または『(モデル値を使用)』が選択された場合は
モデルに設定されている色、線幅を使⽤します。

7.2.39 図面印刷プレビュー（ダイアログモードでのみ動作）

【機能】

図面印刷をプレビューします。

SXF/PRINT と同じ機能ですが違いは、出力が画面になります。

【構文】

SXF/PREVIEW

【図面印刷プレビューダイアログ】



『プリンタ設定』

『ページ設定』

印刷の向き

〔○縦〕

〔○横〕

設定ファイル

〔□ SXF〕

〔▼ファイル名〕

〔最小線幅〕

〔□プリンタに連動〕

印刷倍率

〔○作図倍率〕

〔○用紙サイズに合わせる〕

色

〔○カラー印刷〕

〔『色』〕

〔○モノクロ印刷〕

背景同色の印刷色

〔○設定通り〕

〔○背景色〕

〔○反転色〕

JPEG の色

〔○カラー印刷〕

出力するプリンタを選択します。注1)

用紙サイズ、余白等を設定します。

印刷の向きを選択します。

用紙の縦方向に印刷します。

用紙の横方向に印刷します。

設定ファイルを設定します。

設定ファイルを SXF 用で使用します。注2)

設定ファイルを選択します。注3)

設定ファイルで (モデル値を使用) を選択した場合

最小線幅を指定します。

設定ファイル名をプリンタ名に連動させます。

印刷倍率を設定します。

倍率を指定して印刷します。

用紙サイズに合わせて印刷します。

印刷色を設定します。注4)

カラーで印刷します。

色の RGB 値を変更します。

注) 変更した内容はプレビュー実行中のみ有効です。

モノクロで印刷します。

背景色が白／黒の場合の白／黒色の印刷指定を選択します。

注) 背景が黒の場合、白は黒で印刷します。

設定されたとおりに印刷します。

背景色と同じ色 (白／黒) で印刷します。

背景色の反転色 (黒／白) で印刷します。

モノクロ印刷の場合に JPEG の色を選択します。

カラーで印刷します。

〔○モノクロ印刷〕	モノクロで印刷します。
〔『プレビュー』〕	プレビューを開始します。
〔『取消』〕	プレビューを取消します。

注 1) 選択したプリンタが印刷可能でない場合は正しくプレビューは行えません。

注 2) チェックがオンの場合、設定ファイルを **SXF** 用として使用します。

既定義色 (1 ~ 16) は、設定ファイルの設定を使用します。

ユーザ定義色 (17 ~ 256) は、モデルに設定されている色を使用します。

既定義線幅 (1 ~ 9) は、設定ファイルの設定を使用します。

ユーザ定義線幅 (11 ~ 16) は、モデルに設定されている線幅を使用します。

定義がない場合は線幅 1 を使用します。

チェックがオフの場合、色・線幅は設定ファイルの設定を使用します。

注 3) 設定ファイルがない場合または『(モデル値を使用)』が選択された場合はモデルに設定されている色、線幅を使用します。

注 4) プリンタがカラー印刷できない場合でもプレビューはカラーで表示されることがあります。

プレビュー画面の縮小・拡大はメッセージ領域に表示されているメニューで行ってください。

全体表示	拡大表示	対角 2 点
	縮小表示	位置移動
印刷	戻る	

全体表示

全体を表示します。

拡大表示

拡大表示します。

縮小表示

縮小表示します。

対角 2 点

対角 2 点で指定された範囲を表示します。

位置移動

表示位置を移動させます。

戻る

1 つ前の表示状態に戻します。

印刷

印刷処理を行います。

図面印刷ダイアログが表示されます。

プレビューで設定した内容が引き継がれます。

7.2.40 画面印刷プレビュー (ダイアログモードでのみ動作)

【機能】

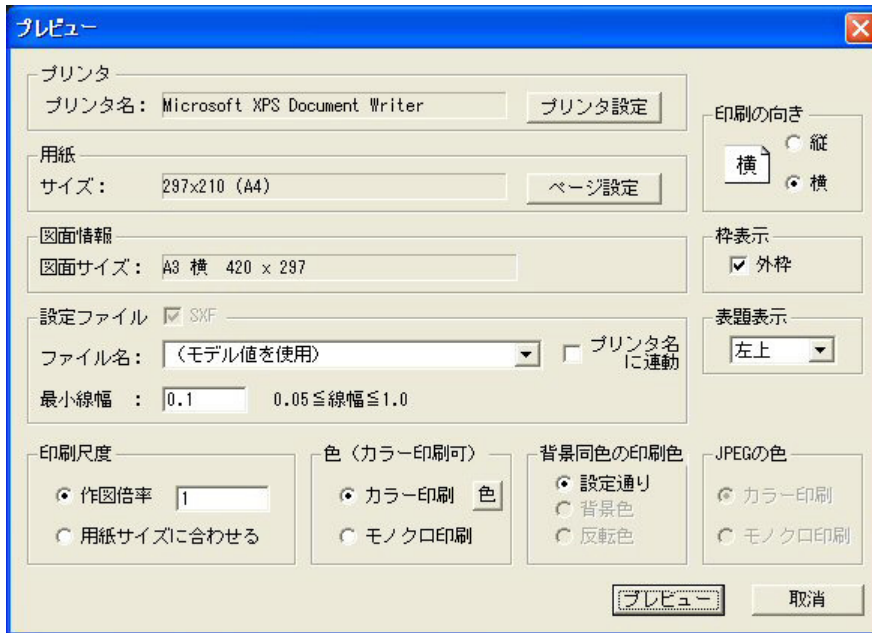
画面印刷をプレビューします。

SXFCOPY/PRINT と同じ機能ですが違いは、出力が画面になります。

【構文】

SXFCOPY/PREVIEW

【画面印刷プレビューダイアログ】



『プリンタ設定』

『ページ設定』

印刷の向き

〔○縦〕

〔○横〕

枠表示

〔□外枠〕

表題表示

〔▼表示位置〕

設定ファイル

〔□ SXF〕

〔▼ファイル名〕

〔最小線幅〕

〔□プリンタに連動〕

印刷倍率

〔○作図倍率〕

〔○用紙サイズに合わせる〕

色

〔○カラー印刷〕

〔『色』〕

〔○モノクロ印刷〕

背景同色の印刷色

〔○設定通り〕

〔○背景色〕

〔○反転色〕

JPEG の色

〔○カラー印刷〕

〔○モノクロ印刷〕

『プレビュー』

出力するプリンタを選択します。注1)

用紙サイズ、余白等を設定します。

印刷の向きを選択します。

用紙の縦方向に印刷します。

用紙の横方向に印刷します。

枠表示を設定します。

枠を表示します。

表題表示を設定します。

表示位置を選択します。

設定ファイルを設定します。

設定ファイルを SXF 用で使します。注2)

設定ファイルを選択します。注3)

設定ファイルで(モデル値を使用)を選択した場合

最小線幅を指定します。

設定ファイル名をプリンタ名に連動させます。

印刷倍率を設定します。

倍率を指定して印刷します。

用紙サイズに合わせて印刷します。

印刷色を設定します。注4)

カラーで印刷します。

色の RGB 値を変更します。

注) 変更した内容はプレビュー実行中のみ有効です。

モノクロで印刷します。

背景色が白／黒の場合の白／黒色の印刷指定を選択します。

注) 背景が黒の場合、白は黒で印刷します。

設定されたとおりに印刷します。

背景色と同じ色(白／黒)で印刷します。

背景色の反転色(黒／白)で印刷します。

モノクロ印刷の場合に JPEG の色を選択します。

カラーで印刷します。

モノクロで印刷します。

プレビューを開始します。

『取消』 プレビューを取消します。

注 1) 選択したプリンタが印刷可能でない場合は正しくプレビューは行えません。

注 2) チェックがオンの場合、設定ファイルを **SXF** 用として使用します。

既定義色 (1 ~ 16) は、設定ファイルの設定を使用します。

ユーザ定義色 (17 ~ 256) は、モデルに設定されている色を使用します。

既定義線幅 (1 ~ 9) は、設定ファイルの設定を使用します。

ユーザ定義線幅 (11 ~ 16) は、モデルに設定されている線幅を使用します。

定義がない場合は線幅 1 を使用します。

チェックがオフの場合、色・線幅は設定ファイルの設定を使用します。

注 3) 設定ファイルがない場合または『(モデル値を使用)』が選択された場合はモデルに設定されている色、線幅を使用します。

注 4) プリンタがカラー印刷できない場合でもプレビューはカラーで表示されることがあります。

プレビュー画面の縮小・拡大はメッセージ領域に表示されているメニューで行ってください。

全体表示	拡大表示	対角 2 点
	縮小表示	位置移動
印刷	戻る	

全体表示

全体を表示します。

拡大表示

拡大表示します。

縮小表示

縮小表示します。

対角 2 点

対角 2 点で指定された範囲を表示します。

位置移動

表示位置を移動させます。

戻る

1 つ前の表示状態に戻します。

印刷

印刷処理を行います。

画面印刷ダイアログが表示されます。

プレビューで設定した内容が引き継がれます。

7.2.41 属性付加機構 (ダイアログモードでのみ動作)

【機能】

アイテムに属性を付加 (編集・削除) します。

属性付加機構の詳細については国土交通省：SXF 表示機能及び確認機能要件書 (案)

<http://www.cals-ed.go.jp/> 電子納品に関する要領・基準

左側メニューの『電子納品に関する要領・基準』

ページ最後方の『SXF Ver3.1 仕様書、表示機能及び確認機能要件書 (案) へ』

ページ最後方の SXF Ver3.1 仕様書 (第 2 版) H21.6 『LZH ファイル：4837KB』内の

SXF Ver3.1 仕様書・同解説 付属書 属性付加機構編

SXF Ver3.1 仕様書・同解説 付属書 共通属性セット編

SXF Ver3.1 実装規約 内の SXF 仕様 実装規約 属性付加機構編

及びオープン CAD フォーマット評議会

<http://www.ocf.or.jp/> オープン CAD フォーマット評議会

左側メニューの『OCF 検定』

左側メニューの『OCF 検定合格基準、規約類』

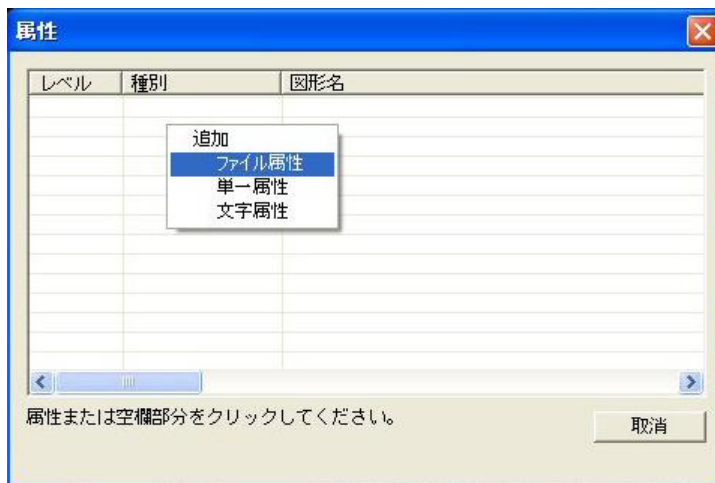
ページ前方の SXF 対応ソフトウェア検定 『SXF 実装規約 OCF 版（属性検定版）』を参照ください。

【構文】

SXF/ATTRSET	IS
-------------	----

IS : 属性を付加、修正及び削除するアイテムを選択します。
この後属性付加、編集、削除のダイアログが表示されます。

【属性一覧ダイアログ】



1. ダイアログの属性一覧の行をクリックします。
〔空白行〕をクリックすると属性の付加になります。
〔ファイル属性〕、〔単一属性〕、〔文字属性〕と表示されている行をクリックすると属性の挿入、修正、削除になります。
2. 属性付加の場合。
 - ・ファイル属性
 - ・単一属性
 - ・文字属性（選択した図形が文字列の場合のみ）
3. ファイル属性付加の場合。
 - ・メニューから〔ファイル属性〕を選択します。

【ファイル属性付加ダイアログ】

- ・「図形名称」を入力します。
- ・属性セットを追加します。
 - ・「<AttributeSet> or <TargetSet> 追加」行をクリックします。注1)
 - ・メニューから「[属性セット <AttributeSet> 追加]」を選択します。
 - ・属性セットの「名称」・「バージョン」・「設計者」を入力します。注2)
 - ・「『追加』」をクリックします。
 - ・属性グループを追加します。
 - ・「<AttrGroup> or <Attr> 追加」をクリックします。注1)
 - ・メニューから「[属性グループ <AttrGroup> 追加]」を選択します。
 - ・属性グループの「名前」を入力します。
 - ・「『追加』」をクリックします。
- 注) 属性グループがある場合のみ追加します。必須ではありません。
2階層まで付加することができます。
- ・属性を追加します。
 - ・「<AttrGroup> or <Attr> 追加」をクリックします。注1)
 - ・メニューから「[属性 <Attr> 追加]」を選択します。
 - ・属性の「属性名称」・「タイプ」・「単位」・「属性値」を入力します。注3)
 - ・「『追加』」をクリックします。
- ・ターゲットセットを追加します。
 - ・「<AttributeSet> or <TargetSet> 追加」をクリックします。注1)
 - ・メニューから「[ターゲットセット <TargetSet> 追加]」を選択します。
 - ・ターゲットセットの「名称」・「バージョン」・「設計者」を入力します。注2)
 - ・「『追加』」をクリックします。
 - ・ターゲットを追加します。
 - ・「<Target> 追加」をクリックします。注1)
 - ・メニューから「[ターゲット <Target> 追加]」を選択します。
 - ・ターゲットの「図形識別番号」・「メモ」を入力します。
 - 図形識別番号を入力する代わりに「『アイテム選択』」でターゲットにする図形を選択することができます。

注) SXF Ver3.1 に出力する場合、メモは失われます。

- ・『追加』をクリックします。
- ・『付加』をクリックします。

4. 単一属性付加の場合。

- ・メニューから [単一属性] を選択します。

【単一属性付加ダイアログ】

A dialog box titled "単一属性" (Single Attribute) with a blue header bar and a red close button. It contains several input fields: "図形識別番号" (Shape Identification Number) with the value "3", "図形名称" (Shape Name), "属性名" (Attribute Name), "属性タイプ" (Attribute Type), "単位" (Unit), and "属性値" (Attribute Value). The "属性名", "属性タイプ", and "単位" fields are grouped together. At the bottom right are "付加" (Add) and "取消" (Cancel) buttons. A message at the bottom left says "図形名称を入力してください" (Please enter the shape name).

- ・属性の [図形名称] ・ [属性名] ・ [属性タイプ] ・ [単位] ・ 属性値] を入力します。注 3)
- ・『付加』をクリックします。

5. 文字属性付加の場合。

- ・メニューから [文字属性] を選択します。

【文字属性付加ダイアログ】

A dialog box titled "文字属性" (Text Attribute) with a blue header bar and a red close button. It contains input fields: "図形識別番号" (Shape Identification Number) with the value "3", "属性名" (Attribute Name), "属性タイプ" (Attribute Type), and "単位" (Unit). The "属性名", "属性タイプ", and "単位" fields are grouped together. At the bottom right are "付加" (Add) and "取消" (Cancel) buttons. A message at the bottom left says "属性名を入力してください" (Please enter the attribute name).

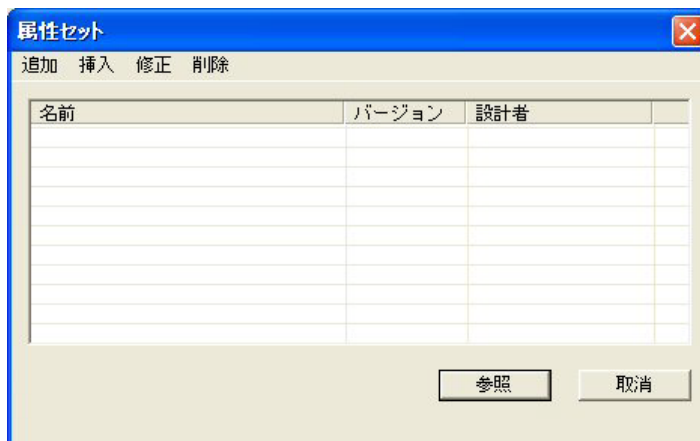
- ・属性の [属性名] ・ [属性タイプ] ・ [単位] を入力します。注 3)
- ・『付加』をクリックします。

注 1) 属性の入力規則

表示	クリックしたときの処理
<AttributeSet> or <TargetSet> 追加	属性セット又はターゲットセットの追加
<AttrGroup> or <Attr> 追加	属性グループ又は属性の追加
<Target> 追加	ターゲットの追加
<AttributeSet name=	属性セットセットの修正・削除
<TargetSet name=	ターゲットセットの修正・削除

表示	クリックしたときの処理
<AttrGroup name=	属性グループの修正・削除
<Attr name=	属性の修正・削除
<Target id=	ターゲットの修正・削除

注 2) ファイル属性の属性セット、ターゲットセットの参照



ファイル属性の属性セット、ターゲットセットのダイアログで『参照』をクリックすると属性セット、ターゲットセットを参照することができます。使用頻度の高い属性セット、ターゲットセットをあらかじめ登録しておくことで属性セット、ターゲットセットの入力を軽減することができます。

注 3) 属性名、属性タイプ、属性単位をコンボボックスから選択して入力することができます。

属性名：S、A、B、C、D、E、F、G、H、I、J、K、L の区分から選択します。

例えば区分 A を選択すると

- A-01 角度
- A-02 立体角
- A-03 長さ
- A-04 面積
- A-05 体積
- A-06 時間
- A-07 角速度
- A-08 速度
- A-09 加速度

の中から属性名を選択することができます。

選択した属性名の既定値の属性タイプ、単位が属性タイプ、単位に設定されます。属性タイプ、単位の横にあるコンボボックスから既定値以外の属性タイプ、単位を選択することもできます。

属性名、属性タイプ、単位に関しては

SXF Ver3.1 仕様書・同解説 付属書 属性付加機構編の

5 属性情報一覧

5-1 タイプと単位

5-2 既定義属性一覧

を参照してください。

7.2.42 確認機能（ダイアログモードでのみ動作）

【機能】

SXF 表示機能及び確認機能用件書（案）平成 21 年 3 月版に準拠した確認を行います。

【構文】

SXF/CONFIRM

【確認機能ダイアログ】

確認する項目にチェックをつけます。初期値は全てを確認するようになっています。

☐ レイヤ名	レイヤ名が命名規則に則って付けられているかを確認
☐ 用紙外図形	用紙外に作図されていないかを確認
☐ 重複図形	図形が重複していないかを確認
☐ ショートベクトル	短い線分が連続していないかを確認
☐ 図面の大きさ	図面の大きさが A サイズかを確認
☐ 面の正位	図面の方向が横で使用されているかを確認
☐ 輪郭線	輪郭線の線種・線幅が規則に則っているかを確認
☐ 余白	余白が規則に則っているかを確認
☐ 色	既定義色以外の色が使われていないかを確認
☐ 背景同色	背景色と同じ色が使われていないかを確認
☐ 線種	線種が規則に則っているかを確認

- [☐線幅] 線幅が規則に則っているかを確認
使用線幅 1:2:4 の組合せを指示することができます。
指示がない場合は自動で組合せを決めます。
- [☐文字の大きさ] 文字の大きさが規則に則っているかを確認
- [☐文字コード] 文字コードが規則に則っているかを確認
- [☐文字配置] 文字配置が規則に則っているかを確認
- [☐ SXF バージョン] SXF ファイルのバージョンを確認

各項目の詳細については国土交通省：SXF 表示機能及び確認機能要件書（案）

<http://www.cals-ed.go.jp/> 電子納品に関する要領・基準

左側メニューの『電子納品に関する要領・基準』

ページ最後方の『SXF Ver3.1 仕様書、表示機能及び確認機能要件書（案）へ』

ページ最後方の SXF 表示機能及び確認機能要件書（案）H21.3 『PDF ファイ：809KB』

及びオープン CAD フォーマット評議会

<http://www.ocf.or.jp/> オープン CAD フォーマット評議会

左側メニューの『OCF 検定』

左側メニューの『OCF 検定合格基準、規約類』

ページ中ほどの SXF 確認機能検定『同 実装規約』

を参照ください。

- [▼自動判別] 線幅の組合せを選択します。
自動判別 (既定値)
0.13, 0.25, 0.5
0.18, 0.35, 0.7
0.25, 0.5, 1.0
0.35, 0.7, 1.4
0.5, 1.0, 2.0
のいずれかを選択します。
- [『詳細』] 入力ファイルの情報を表示します。
- [『実行』] 確認処理を実行します。
確認結果ダイアログが表示されます。
- [『キャンセル』] 処理を取消します。

確認結果ダイアログ



確認処理後確認の結果を表示します。(＜CE＞で終了します)

ダイアログの一覧で背景が白色表示されている項目は目視確認が必要な項目です。一覧で背景が白色表示されている〔項目〕をマウスの左ボタンでクリックし〔『表示』〕で確認項目の図形をハイライト表示させることができます。複数項目選択できます。一覧で背景が白色表示されている〔項目〕をマウスの右ボタンでクリックすると確認した内容が表示されます。表示したダイアログはダイアログ内をマウスでクリックすると閉じます。

- 〔□マーク表示〕 図形の始点又は基準点にマークを表示します。
マークは表示倍率に関係なく一定の大きさで表示されます。
ショートベクトル等の確認に有効です。
- 〔□元図単色表示〕 元図を単色で表示します。
元図はグリッド色で表示します。
- 〔□ブリンク〕 確認図形を点滅表示します。
- 〔『表示色』〕 表示色の RGB 値を変更することができます。
変更は確認機能コマンドが実行されている間のみ有効です。
- 〔『設定保存』〕 表示設定の
マーク表示
元図単色表示
ブリンク
グリッド色、一時図形色
を保存します。
- 〔『表示』〕 一覧で選択された図形を表示します。
一時図形色で表示します。

『レイヤー一覧』 レイヤ名の一覧を別ダイアログで表示します。

【レイヤ名表示ダイアログ】



7.2.43 オプション設定（ダイアログモードでのみ動作）

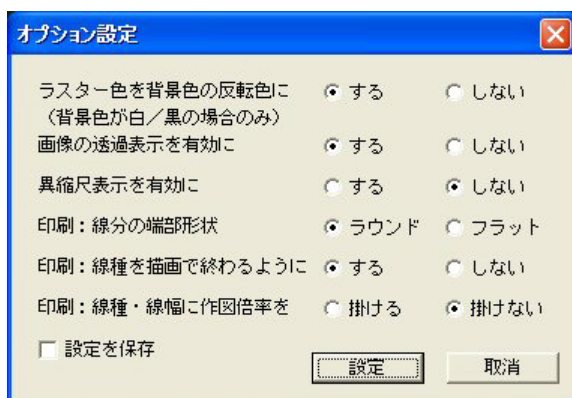
【機能】

各種設定を行います。

【構文】

SXF/OPTION

【オプション設定ダイアログ】



○ラスター色を背景色の反転色に

〔○する〕

： 背景色が白／黒の場合に限ってラスターの色を背景色の反転色で表示します。
グリッド色を背景色の反転色にします。

〔○しない〕

： グリッド色で表示します。

○画像の透過表示を有効に

- ☐[○する] : TIFF イメージの背景を透過モードで描画します。注1)
☐[○しない] : TIFF イメージの背景を画面の背景色で描画します。

○異縮尺表示を有効に

- ☐[○する] : XY 異縮尺の部分図（ピクチャ）を描画します。注2)
 描画のみでアイテムの作成・編集は行えません。
☐[○しない] : XY 異縮尺の部分図（ピクチャ）を小さい方の倍率で描画します。

○印刷：線分の端部形状

- ☐[○ラウンド] : ラウンドで描画します。
☐[○フラット] : フラットで描画します。

○印刷：線種を描画で終わるように

- ☐[○する] : 線分の端点が必ず描画で終わるようにします。
☐[○しない] : 線種をピッチどおりに描画します。

○印刷：線種・線幅に作図倍率を

- ☐[○掛ける] : 作図倍率を掛けて描画します。
☐[○掛けない] : 作図倍率は掛けずに描画します。

○設定を保存

- ☐[□設定を保存] : 設定内容を保存します。

注1) ラスタ（TIFF）背景透過表示に対応しているコマンドは、
 SXF/ZOOMALL、SXF/ZOOMALLVIE、SXF/ZOOM、SXF/ZOOMUP、
 SXF/ZOOMDOWN、SXF/PAN、SXF/RPT
 です。その他のコマンドは対応していません。

注2) XY 異縮尺に対応しているコマンドは、
 SXF/ZOOMALL、SXF/ZOOMALLVIE、SXF/ZOOM、SXF/ZOOMUP、
 SXF/ZOOMDOWN、SXF/PAN、SXF/RPT、SXF/VER
 です。その他のコマンドは対応していません。

7.3 変換テーブルファイル

7.3.1 書式

```

/
キーワード [ /注釈 ]
キーワード 値 [ /注釈 ]
キーワード 値 [、値] =値 [、値] [ /注釈 ]
キーワード 値 [-値] =値 [、値] [ /注釈 ]
  
```

のいずれかの記入方法になります。

キーワードは大文字小文字のいずれでも構いません。タブは空白文字と同等です。

/の後に注釈が記入できます。

1 カラム目が/の場合はその行全体が注釈となります。

/#は、変換プログラムがパラメータの保存で新しい注釈に置き換えます。

7.3.2 SXF → Advance CAD 変換パラメータ

ファイル名 : SXFACAD.TBL

- 入力する SXF ファイルの種類を指定します。

Format	For	
	Form = SFC	: 関係者間での CAD データ交換のための簡易な形式である .sfc 形式
	P21	: 電子納品のための .p21 形式

- 座標系を指定します。

ModelSpace	:	実寸法で変換します。
PaperSpace	:	紙に作図したときの寸法で変換します。

- レイヤの割当て方法を指定します。

LayerToClass	:	レイヤをクラスに割当てます。
LayerToRevision	:	レイヤをレビジョンに割当てます。

- 線種・線幅の割当て方法を指定します。

PenToNormal	:	SXF の線種・線幅を ACAD の線種・線幅に割当てます。
PenToClass	:	ペンをクラスに割当てます。
PenToRevision	:	ペンをレビジョンに割当てます。
PenToLineFont	:	ペンを線種に割当てます。
PenToLineWeight	:	ペンを線幅に割当てます。

注) PenToClass・PenToRevision・PenToLineFont・PenToLineWeight を選択した場合は SXF の線種・線幅をペンに割当てその割当てたペンをクラス・レビジョン・線種又は線幅に割当てます。

- 色の割当て方法を指定します。

ColorToClass	:	色をクラスに割当てます。
ColorToRevision	:	色をレビジョンに割当てます。

- SXF のレイヤを ACAD のクラスに割当てます。

Layer_Class	Layer_Name = Class_No	/Comment
	Layer_Name	: SXF のレイヤ名 OTHER は割当てられていないレイヤ全てを意味します。
	Class_No	: ACAD のクラス番号
	Comment	: 注釈

- SXF のレイヤを ACAD のレビジョンに割当てます。

Layer_Revision	Layer_Name = Revision_No	/Comment
	Layer_Name	: SXF のレイヤ名 OTHER は割当てられていないレイヤ全てを意味します。
	Revision_No	: ACAD のレビジョン番号
	Comment	: 注釈

- SXF の線種を ACAD の線種に割当てます。

Line_Type	Line_Type = Line_Font	/Comment
	Line_Type	: SXF の線種
	Line_Font	: ACAD の線種
	Comment	: 注釈

注) 割当てられなかった線種は、ACAD の線種 1 に割当てます。

- SXF の線幅を ACAD の線幅に割当てます。

Line_Width	Line_Width = Line_Weight	/Comment
	Line_Width	: SXF の線幅
	Line_Weight	: ACAD の線幅
	Comment	: 注釈

注) 割当てられなかった線幅は、ACAD の線幅 1 に割当てます。

- SXF の線種・線幅をペンに割当てます。

```
Pen          Line_Type, Line_Width = Pen_No      /Comment
              Line_Type       :   SXF の線種 (0 は全ての線種は意味します)
              Line_Width      :   SXF の線幅 (0 は全ての線幅を意味します)
              Pen_No          :   ペン番号
              Comment         :   注釈
```

- ペンを ACAD のクラスに割当てます。

```
Pen_Class     Pen_No = Class_No                  /Comment
              Pen_No          :   ペン番号
              Class_No        :   クラス番号
              Comment         :   注釈
```

- ペンを ACAD のレビジョンに割当てます。

```
Pen_Revision  Pen_No = Revision_No              /Comment
              Pen_No          :   ペン番号
              Revision_No     :   レビジョン番号
              Comment         :   注釈
```

- ペンを ACAD の線種に割当てます。

```
Pen_LineFont  Pen_No = Line_Font                /Comment
              Pen_No          :   ペン番号
              Line_Font       :   線種
              Comment         :   注釈
```

- ペンを ACAD の線幅に割当てます。

```
Pen_LineWeight Pen_No = Line_Weight            /Comment
              Pen_No          :   ペン番号
              Line_Weight     :   線幅
              Comment         :   注釈
```

- SXF の色を ACAD のクラスに割当てます。

```
Color_Class   Color_No = Class_No              /Comment
              Color_No        :   色番号
              Class_No        :   クラス番号
              Comment         :   注釈
```

- SXF の色を ACAD のレビジョンに割当てます。

```
Color_Revision Color_No = Revision_No          /Comment
              Color_No        :   色番号
              Revision_No     :   レビジョン番号
              Comment         :   注釈
```

- SXF の矢印を ACAD のマークに割当てます。

```
Arrow_Type    Arrow_Type, Size = Mark_No       /Comment
              Arrow_Type     :   矢印タイプ
              Size            :   大きさ
              Mark_No        :   マーク番号
              Comment         :   注釈
```

- SXF の点マーカーを ACAD のマークに割当てます。

```
Marker_Type   Marker_Type, Size = Mark_No      /Comment
              Marker_Type    :   マーカータイプ
              Size            :   大きさ
              Mark_No        :   マーク番号
              Comment         :   注釈
```

7.3.3 SXF → Advance CAD 変換パラメータ既定値

- レイヤ割当て：

レイヤをクラスに割当てて。
レイヤが現れた順番にクラス番号 1 から順に割当てて。

レイヤをレビジョンに割当てた場合は、
レイヤが現れた順番にレビジョン番号 1 から順に割当てて。

- 線種・線幅割当て：線種・線幅に割当てて

線種	説明	線種
1	continuous	1
2	dashed	2
3	dashed spaced	3
4	long dashed dotted	4
5	long dashed double-dotted	5
6	long dashed triplicate-dotted	6
7	dotted	7
8	chain	8
9	chain double dash	9
10	dashed dotted	10
11	double-dashed dotted	11
12	dashed double-dotted	12
13	double-dashed double-dotted	13
14	dashed triplicate-dotted	14
15	double-dashed triplicate-dotted	15
16	reserve	16
17	ユーザ定義線種 1	17
:		
32	ユーザ定義線種 1 6	32

線幅	説明	線幅
1	0.13	1
2	0.18	2
3	0.25	3
4	0.35	4

線幅	説明	線幅
5	0.5	5
6	0.7	6
7	1.0	7
8	1.4	8
9	2.0	9
10	reserve	10
11	ユーザ定義線幅 1	11
:		
16	ユーザ定義線幅 6	16

- 色割当て : 色をレビジョンに割当てる

色	説明	レビジョン
1	Black	1
2	Red	2
3	Green	3
4	Blue	4
5	Yellow	5
6	Magenta	6
7	Cyan	7
8	White	8
9	Deeppink	9
10	Brown	10
11	Orange	11
12	Light Green	12
13	Light Blue	13
14	Lavender	14
15	Light Gray	15
16	Dark Gray	16
17	ユーザ定義 1	17
:		
256	ユーザ定義 2 4 0	256

- 寸法矢印

矢印	説明	マーク番号
1	blanked arrow	271
2	blanked box	272
3	blanked dot	273
4	dimension origin	274
5	filled box	275
6	filled arrow	276
7	filled dot	277
8	integral symbol	278
9	open arrow	279
10	slash	280
11	unfilled arrow	281

7.3.4 Advance CAD → SXF 変換パラメータ

ファイル名 : ACADSF. TBL

- 出力する SXF ファイルのバージョンを指定します。

```

Version      Ver
              Ver = 2.0    :   バージョン 2.0
                              背景色出力なし
                              表題欄出力なし
                              画像は TIF を 1 つ出力
                              弧長寸法は角度寸法に変換
              3.0    :   バージョン 3.0
                              背景色出力あり
                              表題欄出力あり
                              画像は TIF・JPG 合わせて最大 30 出力
                              弧長寸法は角度寸法に変換
              3.1    :   バージョン 3.1
                              背景色出力あり
                              表題欄出力あり
                              画像は TIF・JPG 合わせて最大 30 出力
                              弧長寸法出力あり

```

- 出力する SXF ファイルの種類を指定します。

```

Format      Form
              Form = SFC    :   関係者間での CAD データ交換のための簡易な形式である . sfc 形式
              P21          :   電子納品のための . p21 形式

```

- 作成者名の既定値を指定します。

```

Author      Author
              Author        :   作成者名

```

- 組織名の既定値を指定します。

```

Organization Organization
Organization      : 組織名

```

- 変換する座標系を指定します。

```

ModelSpace      : 実寸法で変換します。
PaperSpace      : 紙に作図したときの寸法で変換します。

```

- レイヤ割当ての方法を指定します。

```

ClassToLayer     : クラスをレイヤに割当てます。
RevisionToLayer  : レビジョンをレイヤに割当てます。

```

- ペン割当ての方法を指定します。

```

ClassToPen       : クラスをペンに割当てます。
RevisionToPen    : レビジョンをペンに割当てます。
LineFontToPen    : 線種をペンに割当てます。
LineWeightToPen  : 線幅をペンに割当てます。
ColorToPen       : 色をペンに割当てます。

```

- 図枠クラスのレイヤ割当てを指定します。

```

Class_Layer      Class_No = Layer_Name      /Comment
Class_no         : クラス番号 = 257
Layer_Name       : レイヤ名
Comment          : 注釈

```

- 図枠レビジョンのレイヤ割当てを指定します。

```

Revision_Layer   Revision_No = Layer_Name    /Comment
Revision_no      : レビジョン番号 = 257
Layer_Name       : レイヤ名
Comment          : 注釈

```

- クラスのペン割当てを指定します。

```

Class_Pen        Class_No[-Class_No] = Pen_No /Comment
Class_Pen        Class_No[, Class_No...] = Pen_No /Comment
クラス番号をペン番号に割当てます。
Class_no         : クラス番号
Pen_No           : ペン番号
-               : 範囲指定時に使用
,               : 個別指定時に使用
Comment          : 注釈

```

- レビジョンのペン割当てを指定します。

```

Revision_Pen     Revision_No[-Revision_No] = Pen_No /Comment
Revision_Pen     Revision_No[, Revision_No...] = Pen_No /Comment
レビジョン番号をペン番号に割当てます。
Revision_no      : レビジョン番号
Pen_No           : ペン番号
-               : 範囲指定時に使用
,               : 個別指定時に使用
Comment          : 注釈

```

- 線種のペン割当てを指定します。

```

LineFont_Pen     Line_Font[-Line_Font] = Pen_No /Comment
LineFont_Pen     Line_Font[, Line_Font...] = Pen_No /Comment
線種をペン番号に割当てます。
Line_Font        : 線種
Pen_No           : ペン番号
-               : 範囲指定時に使用
,               : 個別指定時に使用
Comment          : 注釈

```

- 線幅のペン割当てを指定します。

```
LineWeight_Pen Line_Weight[-Line_Weight] = Pen_No /Comment
LineWeight_Pen Line_Weight[,Line_Weight...] = Pen_No /Comment
```

線幅をペン番号に割当てます。

```
Line_Weight : 線幅
Pen_No      : ペン番号
-          : 範囲指定時に使用
,          : 個別指定時に使用
Comment    : 注釈
```

- 色のペン割当てを指定します。

```
Color_Pen Color_No[-Color_No] = Pen_No /Comment
Color_Pen Color_No[,Color_No...] = Pen_No /Comment
```

色番号をペン番号に割当てます。

```
Color_No : 色番号
Pen_No   : ペン番号
-        : 範囲指定時に使用
,        : 個別指定時に使用
Comment  : 注釈
```

- ペンを S X F の線種・線幅に割当てます。

```
Pen Pen_No = Line_Type, Line_Width /Comment
Pen_No : ペン番号
Line_Type : 線種 (0の場合線種割当てを摘要します)
Line_Width : 線幅 (0の場合線幅割当てを摘要します)
Comment : 注釈
```

- ACAD の線種を S X F 線種に割当てます。

注) ペンの線種割当てで0が指定された場合この割当てを摘要します。

```
Line_Font Line_Font = Line_Type /Comment
Line_Font : 線種
Line_Type : 線種
Comment   : 注釈
```

- ACAD の線幅を S X F 線幅に割当てます。

注) ペンの線幅割当てで0が指定された場合この割当てを摘要します。

```
Line_Weight Line_Weight = Line_Width /Comment
Line_Weight : 線幅
Line_Width  : 線幅
Comment     : 注釈
```

- ACAD の色を S X F の色に割当てます。

```
Color Color_No = Color_No /Comment
Color_No : 色番号
Color_No : 色番号
Comment  : 注釈
```

- ACAD のマークを S X F の矢印に割当てます。

```
Arrow_Type Mark_No = Arrow_Type, Size /Comment
Mark_No : マーク番号
Arrow_Type : 矢印タイプ
Size : 大きさ
Comment : 注釈
```

- ACAD のマークを S X F の点マーカーに割当てます。

```
Marker_Type Mark_No = Marker_Type, Size /Comment
Mark_No : マーク番号
Marker_Type : マーカータイプ
Size : 大きさ
Comment : 注釈
```

7.3.5 Advance CAD → SXF 交換パラメータ既定値

- レイヤ割当て
クラスをレイヤに割当ててる。クラス名は必須。
- 線種・線幅：ペン割当てなし

線種	線種	説明
1	1	実線 : continuous
2	2	破線 : dashed
3	3	跳び破線 : dashed spaced)
4	4	一点長鎖線 : long dashed dotted
5	5	二点長鎖線 : long dashed double-dotted
6	6	三点長鎖線 : long dashed triplicate-dotted
7	7	点線 : dotted
8	8	一点鎖線 : chain
9	9	二点鎖線 : chain double dash
10	10	一点短鎖線 : dashed dotted
11	11	一点二短鎖線 : double-dashed dotted
12	12	二点短鎖線 : dashed double-dotted)
13	13	二点二短鎖線 : double-dashed double-dotted
14	14	三点短鎖線 : dashed triplicate-dotted
15	15	三点二短鎖線 : double-dashed triplicate-dotted
16	1	reserve
17	17	ユーザ定義線種 1
:		
32	32	ユーザ定義線種 1 6

線幅	線幅	説明
1	1	0.13
2	2	0.18
3	3	0.25
4	4	0.35

線幅	線幅	説明
5	5	0.5
6	6	0.7
7	7	1.0
8	8	1.4
9	9	2.0
10	1	reserve
11	9	ユーザ定義線幅 1
:		
16	9	ユーザ定義線幅 6

- 色割当て

色	色	説明
1	1	黒 : Black
2	2	赤 : Red
3	3	緑 : Green
4	4	青 : Blue
5	5	黄 : Yellow
6	6	桃 : Magenta
7	7	水 : Cyan
8	8	白 : White
9	9	牡丹 : Deeppink
10	10	茶 : Brown
11	11	橙 : Orange
12	12	薄緑 : Light Green
13	13	明青 : Light Blue
14	14	青紫 : Lavender
15	15	明灰 : Light Gray
16	16	暗灰 : Dark Gray
17	17	ユーザ定義色 1
:		
256	256	ユーザ定義色 2 4 0

• 寸法矢印

マーク番号	矢印	説明
271	1	blanked arrow
272	2	blanked box
236	3	blanked dot
274	4	dimension origin
275	5	filled box
276	6	filled arrow
277	7	filled dot
278	8	integral symbol
279	9	open arrow
280	10	slash
281	11	unfilled arrow
その他	9	open arrow

● 変換対応表

Advance CAD → SXF

点	点マーカー (dot)
直線	直線
円	円
円弧	円弧
橢円	橢円
橢円弧	橢円弧
ストリング	折線、円弧、スプライン
自由曲線	スプライン
クロソイド	折れ線 (修正されたまたは Ver2.0、3.0) クロソイド (Ver3.1)
直線寸法	直線寸法
半径寸法	半径寸法
直径寸法	直径寸法
角度寸法	角度寸法
座標寸法	複合図形
弧長寸法 (補助線平行) (補助線放射状)	複合図形 角度寸法 (Ver2.0、3.0) 弧長寸法 (Ver3.1)
注記	文字
マーク	複合図形 点マーカー (変換パラメータの設定が必要)
引出し注釈	引出し線
風船 (マーク番号 51) (51 以外)	風船 複合図形
幾何公差	複合図形
溶接記号	複合図形
データム	直線寸法
ハッチング	ハッチング (ユーザ定義) ハッチング (パターン) または要素展開
塗り潰し	ハッチング (塗り)
複合図形 (作図グループ) (作図部品)	複合図形 (作図グループ) (作図部品)
構成要素+等高線属性	等高線
複合アイテム	複合図形
シンボル	複合図形

Advance CAD → SXF

サブモデル	複合図形
メンバーアイテム	複合図形
ラスター配置	ラスター配置
図面配置	配置ピクチャを複合図形（部分図）に変換 シート
ピクチャ	部分図
クラス	レイヤ
レビジョン	色
色	色
線種	線種
線幅	線幅

SXF → Advance CAD

点マーカー（dot）	点
点マーカー	マーク 変換パラメータの設定が必要。
直線	直線
円	円
円弧	円弧
楕円	楕円
楕円弧	楕円弧
折線	ストリング
クロソイド	クロソイド（ストリング） 修正がなければクロソイドで保持
スプライン	自由曲線
直線寸法	直線寸法
半径寸法	半径寸法
直径寸法	直径寸法
角度寸法	角度寸法
弧長寸法	弧長寸法（補助線：放射状）
文字	注記
引出し線	引出し注釈
風船	風船
ハッチング（ユーザ定義）	ハッチング
ハッチング（塗り）	塗り潰し

SXF → Advance CAD

ハッチング（外部定義）	変換不可
ハッチング（パターン）	ハッチング（パターン）
複合図形（作図グループ） （作図部品）	複合図形（作図グループ） （作図部品）
等高線	構成要素＋等高線属性
ラスター配置	ラスター配置
シート	図面配置
レイヤ	クラス
色	レビジョン
線種	線種
線幅	線幅

7.4 製図基準テンプレートファイル

テンプレートファイルは、製図基準ファイル・定数ファイル・属性定義ファイル・SXF 変換テーブルファイルを一括作成するためファイルです。

本テンプレートを用いて変換をする場合、次の事項を前提としています。

7.4.1 前提条件

Advance CAD の色コード 1 ～ 16 は、既定義色として用います。
 Advance CAD の色コード 17 ～ 256 は、ユーザ定義色として用います。
 Advance CAD の線種コード 1 ～ 15 は、既定義線種として用います。
 Advance CAD の線種コード 17 ～ 32 は、ユーザ定義線種として用います。
 Advance CAD の線幅コード 1 ～ 9 は、既定義線幅として用います。
 Advance CAD の線幅コード 11 ～ 16 は、ユーザ定義線幅として用います。
 SXF のレイヤは、Advance CAD のクラスに割当てます。
 SXF の色は、Advance CAD のレビジョンに割当てます。

7.4.2 テンプレートファイルの場所

コンフィグレーションファイル (ACAD.SET) の #SXF:TEMPLATE# で指定したディレクトリ。
 #SXF:TEMPLATE# がない場合は #SXF:TABLE# で指定したディレクトリ。

7.4.3 テンプレートの書式

1 カラム目からキーワード

COLn	:	色の設定
REVn	:	色のレビジョン割当て
LFNTn	:	線種の設定
LWETn	:	線幅の設定
FIL	:	設計種別・図面種別一覧を出力するファイル名を設定
LS	:	責任主体の設定
CLS _n	:	レイヤ名のクラス割当て
T1	:	設計種別
T2	:	図面種別
T3	:	図面種別
L1	:	レイヤ名 (図面オブジェクト)
L2	:	レイヤ名 (作図要素)

で各種設定をします。

1 カラム目が / の行は注釈行です。

(1) 色を指定

指定する場合は、COL0 ～ COL16 すべてを指定してください。

指定が無い場合は既定値を用います。

COLn	!	色 !
n	:	Advance CAD の色番号 (0 ～ 16)
	:	0 はレイヤ指定で色指定がない場合の既定色になります。
色	:	色を指定します。
	:	色は次のいずれかを指定します。

黒、赤、緑、青、黄、桃、水、白、牡丹、茶、橙、薄緑、明青、青紫、明灰、暗灰

既定値

キーワード	色	キーワード	色
COL0	！ 白 ！	COL9	！ 牡丹 ！
COL1	！ 黒 ！	COL10	！ 茶 ！
COL2	！ 赤 ！	COL11	！ 橙 ！
COL3	！ 緑 ！	COL12	！ 薄緑 ！
COL4	！ 青 ！	COL13	！ 明青 ！
COL5	！ 黄 ！	COL14	！ 青紫 ！
COL6	！ 桃 ！	COL15	！ 明灰 ！
COL7	！ 水 ！	COL16	！ 暗灰 ！
COL8	！ 白 ！		

(2) 色のレビジョン割当てを指定

指定する場合は、すべての色を指定してください。

この指定がない場合は既定値を使用します。

REVn ！ 色 ！

n : Advance CAD のレビジョン番号 (1 ~ 256)

色 : 色を指定します。

色は次のいずれかを指定します。

黒、赤、緑、青、黄、桃、水、白、牡丹、茶、橙、薄緑、明青、青紫、明灰、暗灰

既定値

キーワード	色	キーワード	色
REV1	！ 黒 ！	REV9	！ 牡丹 ！
REV2	！ 赤 ！	REV10	！ 茶 ！
REV3	！ 緑 ！	REV11	！ 橙 ！
REV4	！ 青 ！	REV12	！ 薄緑 ！
REV5	！ 黄 ！	REV13	！ 明青 ！
REV6	！ 桃 ！	REV14	！ 青紫 ！
REV7	！ 水 ！	REV15	！ 明灰 ！
REV8	！ 白 ！	REV16	！ 暗灰 ！
		REV17 ~ REV256 は、色コード 17 ~ 256。	

(3) 線種を指定

指定する場合は、LFNT0 ~ LFNT15 すべてを指定してください。

指定が無い場合は既定値を使用します。

LFNTn [線種名]

n : Advance CAD の線種番号 (0 ~ 15)

0 は、レイヤ指定で線種指定がない場合の既定線種になります。

線種名： 線種名は次のいずれかを指定します。
 実線、破線、跳び破線、一点長鎖線、二点長鎖線、三点長鎖線、点線、
 一点鎖線、二点鎖線、一点短鎖線、一点二短鎖線、二点短鎖線、
 二点二短鎖線、三点短鎖線、三点二短鎖線

既定値

キーワード	線種名	キーワード	線種名
LFNT0	[実線]	LFNT9	[二点鎖線]
LFNT1	[実線]	LFNT10	[一点短鎖線]
LFNT2	[破線]	LFNT11	[一点二短鎖線]
LFNT3	[跳び破線]	LFNT12	[二点短鎖線]
LFNT4	[一点長鎖線]	LFNT13	[二点二短鎖線]
LFNT5	[二点長鎖線]	LFNT14	[三点短鎖線]
LFNT6	[三点長鎖線]	LFNT15	[三点二短鎖線]
LFNT7	[点線]		
LFNT8	[一点鎖線]		

- (4) 線幅を指定します。
 指定する場合は、LWET0 ～ LWET9 すべてを指定してください。

指定が無い場合は既定値を使用します。

LWETn (線幅名)

n : Advance CAD の線幅番号 (0 ～ 9)

0 は、レイヤ指定で線幅指定がない場合の既定線幅になります。

線幅名： 次の線幅名のいずれかを指定します。

0.13、0.18、0.25、0.35、0.5、0.7、1.0、1.4、2.0

既定値

キーワード	線幅	キーワード	線幅
LWET0	(0.13)	LWET5	(0.5)
LWET1	(0.13)	LWET6	(0.7)
LWET2	(0.18)	LWET7	(1.0)
LWET3	(0.25)	LWET8	(1.4)
LWET4	(0.35)	LWET9	(2.0)

- (5) 責任主体を指定します。

LS <種別> "注釈" # ファイル名 #

種別 : 次のいずれかを指定します。(半角英1文字)

S : 測量

D : 設計

C : 施工

M : 維持管理

注釈 : 説明

ファイル名 : ファイル名

指定が無い場合は 注釈_ を使用します。

- (6) 製図基準ファイル名(図面種別の一覧を保持)を指定します。
 指定がない場合は、"CAD 製図基準"をファイル名とします。
 この名前は、変換テーブルファイルにも用います。

例えば、責任主体が " 設計 " でファイル名が " C A D 製図基準 " の場合、変換テーブル
ファイル名は設計 _ C A D 製図基準 _ SXFACAD、設計 _ C A D 製図基準 _ ACADSXF にな
ります。

FIL " ファイル名 "
ファイル名 : ファイル名
ファイル名は、LS+FIL になります。

- (7) クラス番号とクラス名（レイヤ名）の対応を指定します。
Ln（レイヤ名指定）の設定で使用するクラス名（レイヤ名）すべてを指定してください。

CLSn < クラス名 > " 注釈 "
n : Advance CAD のクラス番号（0 ~ 256）
0 は、定義されていないクラスすべてを意味します。
クラス名 : クラス名（レイヤ名）
責任主体 + このクラス名 になります。
注釈 : クラス名（レイヤ名）の説明

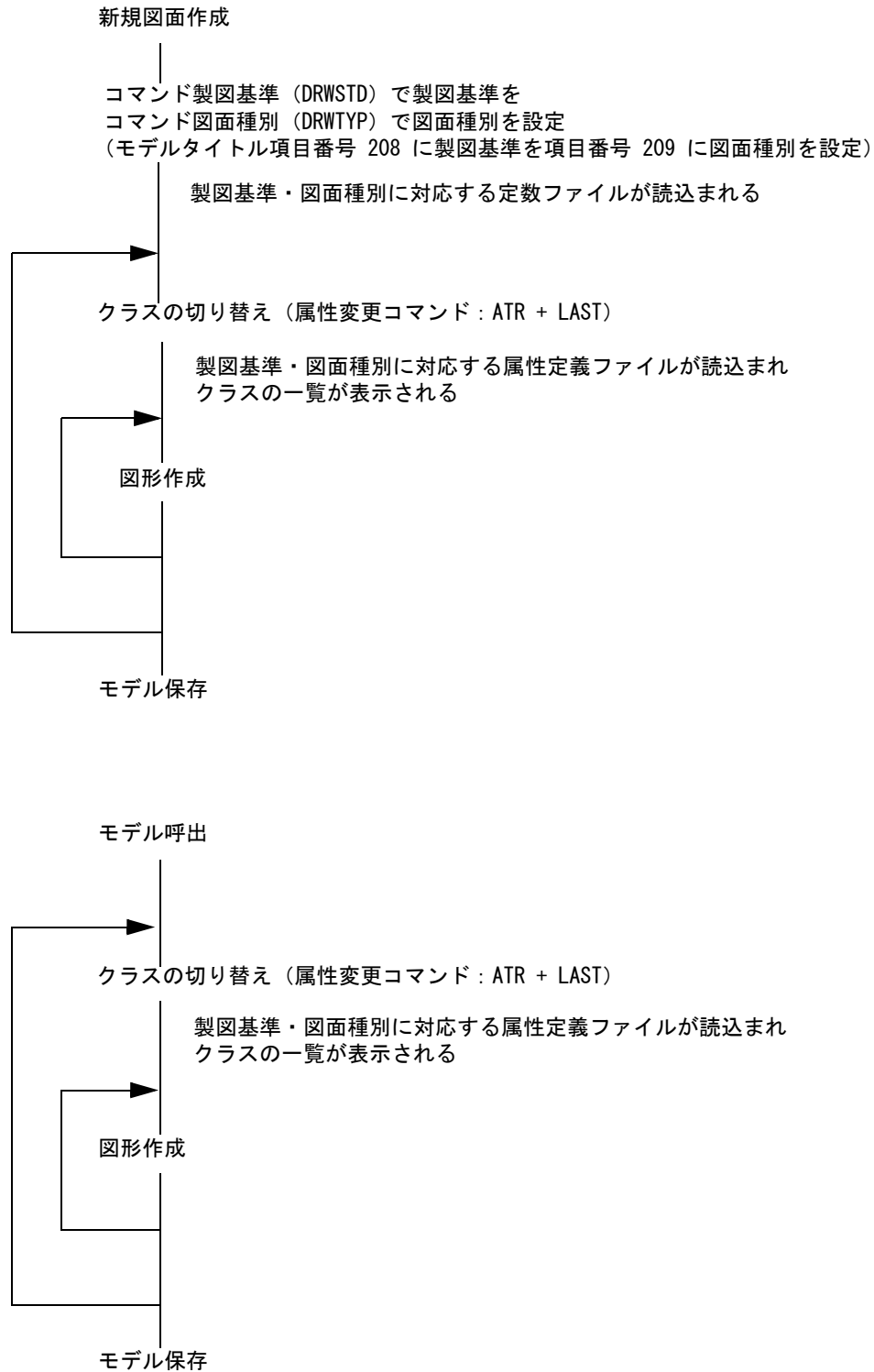
- (8) 図面種別を指定します。
定数ファイル・属性定義ファイルのファイル名になります。

Tn " 種別 " # ファイル名 #
n = 1 : 設計種別を指定します。
2 : 図面種別を指定します。
3 : 図面種別を指定します。
種別 : 種別名を指定します。
ファイル名 : ファイル名を指定します。
指定が無い場合は " 種別 _ " を使用します。
ファイル名は、定数ファイル・属性定義ファイルを作成する時に用います。
ファイル名は、LS+T1、LS+T1+T2 又は LS+T1+T2+T3 になります。

- (9) レイヤ名を指定します。

Ln < レイヤ名 > " 注釈 " ! 色 , ! [線種名 ,] (線幅名 ,)
n = 1 : 図面オブジェクト
2 : 作図要素
レイヤ名 : レイヤ名
レイヤ名は、責任主体 (LS)+L1 又は 責任主体 (LS)+L1+L2 になります。
注釈 : レイヤの説明
色 : レイヤで使用する色を指定します。(カンマで区切って複数指定可)
指定がない場合は、COL0 で設定された色となります。
色名の : 以降は注釈になります。
例. ! 白 : 任意 !
線種名 : レイヤで使用する線種を指定します。(カンマで区切って複数指定可)
指定がない場合は、LFNT0 で設定された線種となります。
線幅名 : レイヤで使用する線幅を指定します。(カンマで区切って複数指定可)
指定がない場合は、LWET0 で設定された線幅となります。

7.4.4 運用



7.5 SXF 既定義線種・線幅・色コード (参考資料)

7.5.1 既定義線種コード

線種名	コード	値 (name)	ピッチ
実線	1	continuous	
破線	2	dashed	6, 1.5
跳び破線	3	dashed spaced	6, 6
一点長鎖線	4	long dashed dotted	12, 1.5, 0.25, 1.5
二点長鎖線	5	long dashed double-dotted	12, 1.5, 0.25, 1.5, 0.25, 1.5
三点長鎖線	6	long dashed triplicate-dotted	12, 1.5, 0.25, 1.5, 0.25, 1.5, 0.25, 1.5
点線	7	dotted	0.25, 1.5
一点鎖線	8	chain	12, 1.5, 3.5, 1.5
二点鎖線	9	chain double dash	12, 1.5, 3.5, 1.5, 3.5, 1.5
一点短鎖線	10	dashed dotted	6, 1.5, 0.25, 1.5
一点二短鎖線	11	double-dashed dotted	6, 1.5, 6, 1.5, 0.25, 1.5
二点短鎖線	12	dashed double-dotted	6, 1.5, 0.25, 1.5, 0.25, 1.5
二点二短鎖線	13	double-dashed double-dotted	6, 1.5, 6, 1.5, 0.25, 1.5, 0.25, 1.5
三点短鎖線	14	dashed triplicate-dotted	6, 1.5, 0.25, 1.5, 0.25, 1.5, 0.25, 1.5
三点二短鎖線	15	double-dashed triplicate-dotted	6, 1.5, 6, 1.5, 0.25, 1.5, 0.25, 1.5, 0.25, 1.5

7.5.2 既定義線幅コード

線幅名	コード	値 (width)
0.13mm	1	0.13
0.18mm	2	0.18
0.25mm	3	0.25
0.35mm	4	0.35
0.5mm	5	0.5
0.7mm	6	0.7
1mm	7	1
1.4mm	8	1.4
2mm	9	2

7.5.3 既定義色コード

色名	コード	値 (name)	R	G	B
黒	1	black	0	0	0
赤	2	red	255	0	0
緑	3	green	0	255	0
青	4	blue	0	0	255
黄色	5	yellow	255	255	0
マゼンタ	6	magenta	255	0	255
シアン	7	cyan	0	255	255
白	8	white	255	255	255
牡丹	9	deeppink	192	0	128
茶	10	brown	192	128	64
橙	11	orange	255	128	0
薄緑	12	lightgreen	128	192	128
明青	13	lightblue	0	128	255
青紫	14	lavender	128	64	255
明灰	15	lightgray	192	192	192
暗灰	16	darkgray	128	128	128

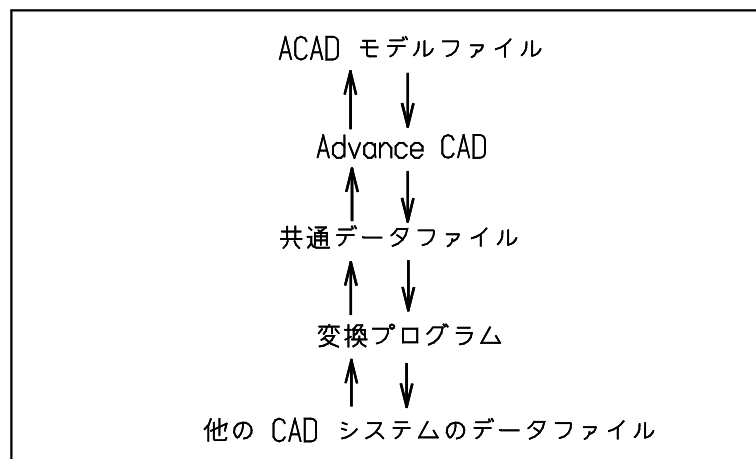
第 8 章 共通データファイル

● 概要

共通データファイルは Advance CAD と他の CAD システム間で図形データを交換するために用意した CAD データ交換ファイル書式です。Advance CAD は共通データファイルを読み込むコマンドと Advance CAD のモデルを共通データファイルに書き出すためのコマンドをもっています。したがって他の CAD システムの CAD データをこの共通データファイルに変換すれば、簡単に Advance CAD システムに取り込むことができます。また逆方向の転送のときは、共通データファイルを他の CAD システムが読み込めるように加工します。

共通データファイルは Advance CAD で決めた書式で、公式のものではありませんが、Advance CAD の図形データをテキストファイルとして出力する機能は重要です。テキストファイルは他の CAD システムや CAM システムの入力データとして加工するのが比較的簡単だからです。

相手側の CAD システムで共通データファイル进行处理するためには、相手側の CAD システムに変換機能を組み込むか、あるいは独立の変換プログラムを用意する必要があります。



なお、本章での“_”（アンダースコア）は半角のスペースを意味します。

8.1 共通データファイル バージョン 10.0

共通データファイル読み込みコマンドは、共通データファイルのバージョンレコードによってその共通データファイルを作成した Advance CAD のバージョンを知ります。

Advance CAD のバージョンが古い共通データファイルであればそれを読み込み、データ表現を現在のバージョンに適合するように変更します。したがって Advance CAD では古いバージョンの共通データファイルも読み込むことができます。つまり上位互換性は保持されています。しかし Advance CAD の新しいバージョンで作成した共通データファイルはそれより古いバージョンの Advance CAD では読み込めません。

バージョンレコードは以下のようになります。共通データファイルのバージョンとそれを作成した Advance CAD のバージョンがわかります。

version 4.3 以下	バージョンレコードはない。Advance CAD version 4.3 以下とみなされる。	
version 4.4	VERSION 2.0	COMMON DATA FILE
version 5.0 ~	VERSION 3.0	COMMON DATA FILE
version 8.0 ~	VERSION 4.0	COMMON DATA FILE
version 11.02 ~	VERSION 5.0	COMMON DATA FILE,
version 12.0 ~	VERSION 6.0	COMMON DATA FILE
version 13.0 ~	VERSION 7.0	COMMON DATA FILE
version 15.0 ~	VERSION 8.0	COMMON DATA FILE
version 18.0 ~	VERSION 9.0	COMMON DATA FILE
version 19.0 ~	VERSION 10.0	COMMON DATA FILE

本章では 共通データファイル バージョン 10.0 の構造および各レコードの書式について説明します。共通データファイルのバージョン 9.0 から 10.0 への変更点は、以下の 3 点です。

- バージョン番号レコード
- ドローイングヘッダーレコード
- サブレコードタイプの廃止と統合

8.1.1 共通データファイルの構造

バージョン番号 ブロック
 モデル データブロック
 ピクチャ データブロック
 ドローイング データブロック
 アイテム データブロック

8.1.2 バージョン番号ブロック

- (1) バージョン番号 ブロックの構造

バージョン番号

(2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	10.0
3 注釈	A40	"__COMMON_DATA_FILE, , _Advance_CAD_ver_19.0"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_19.0')

8.1.3 モデル データブロック

(1) モデル データブロックの構造

モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 256)
 ドローイング縮尺値

(2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE', 2I4/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I3	1 - 256
3 X 方向余弦	3G23.15	(L1, M1, N1)
4 Y 方向余弦	3G23.15	(L2, M2, N2)
5 Z 方向余弦	3G23.15	(L3, M3, N3)

レコードフォーマット

('MATRIX', I3/5X, 3G23.15/5X, 3G23.15/5X, 3G23.15)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G23. 15	

レコードフォーマット
 (' DRAWSCF____', G23. 15)

8.1.4 ピクチャデータブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1
 ピクチャ レコード #2
 ⋮
 ピクチャ レコード #N (N ≤ 256)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I3	1 - 256
3 ピクチャ縮尺値	G23. 15	
4 ピクチャローカル原点	2G23. 15	Xorg, Yorg
5 Z 値	G23. 15	
6 表示範囲	4G23. 15	Xmin, Ymin, Xmax, Ymax

レコードフォーマット
 (' PICTURE____', I3/5X, 3G23. 15/5X, 3G23. 15/5X, 2G23. 15)

8.1.5 ドローイング データブック

(1) ドローイング データブロックの構造

ドローイング ヘッダー	]	ドローイングページ #1
ドローイングタイトル #1		
ドローイングタイトル #2		
ドローイングタイトル #3		
配置ピクチャ レコード #1		
配置ピクチャ レコード #2	]	ドローイングページ #2
:		
配置ピクチャ レコード #i		
ドローイング ヘッダー		
配置ピクチャ レコード #1		
配置ピクチャ レコード #2	]	ドローイングページ #N (N ≤ 256)
:		
配置ピクチャ レコード #j		
:		
:		
ドローイング ヘッダー	]	ドローイングページ #N (N ≤ 256)
ドローイングタイトル #1		
配置ピクチャ レコード #1		
配置ピクチャ レコード #2		
:		
配置ピクチャ レコード #n		

(2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I3	1 - 256
3 テンプレート名	A32	"A0", "A1" . .
4 テンプレートサイズ	2G15.7	width, height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I4	0 - 1024

レコードフォーマット

('DRAWING__', I3, 1X, A32, 2G15.7, I3, I4)

(3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無

レコード項目	フォーマット	内容
6 変更された配置位置	2G15. 7	Xorg, Yorg
7 ドローイングタイトル	A60	文字列

レコードフォーマット

(10X, 2I4, I2, G15. 7, I2, 2G15. 7/(10X, A60))

(4) 配置ピクチャ レコード

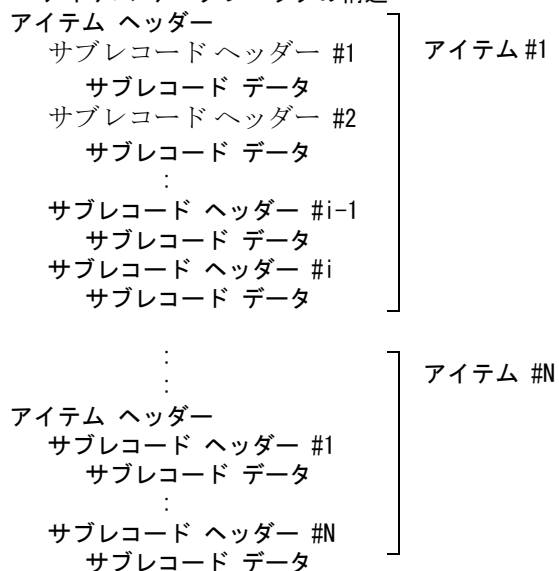
レコード項目	フォーマット	内容
1 配置ピクチャ番号	I3	1 - 256
2 配置ウインドウ番号	I3	0 - 255
3 配置位置	2G23. 15	Xorg, Yorg (ドローイング座標)
4 配置角度 (度)	G23. 15	
5 配置スケール	G23. 15	
6 ウインドウ領域左下	2G23. 15	Xpos, Ypos (モデル座標)
7 ウインドウサイズ	2G23. 15	width, height (モデル座標)
8 ウインド原点	2G23. 15	Xorg, Yorg (モデル座標)
9 ウインドウ角度 (度)	G23. 15	

レコードフォーマット

(10X, I3, I3, I3/5X, 3G23. 15/5X, 3G23. 15/5X, 3G23. 15/5X, 2G23. 15)

8.1.6 アイテム データブロック

(1) アイテム データブロックの構造



(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 63
2 クラス	I4	1 - 256
3 ラインウエイト	I3	1 - 16
4 ラインフォント	I3	1 - 63
5 レビジョン	I4	1 - 256
6 ピクチャ	I4	1 - 320
7 ブランクフラグ	I2	0 - 1
8 アイテム名	A12	文字列
9 注釈	A26	アイテムタイプの略称を注釈として記述

レコードフォーマット

(' ITEM', I3, I4, 2I3, 2I4, I2, 1X, A12, 14X, A26)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコード タイプ	I3	1 - 63
2 データモード	I2	0 - 4
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 64
5 パーシャルウエイト	I2	0 - 16
6 継続フラグ (後述)	I1	1: 継続 0: 最終
7 注釈	A26	サブレコードタイプの略称を注釈として記述

レコードフォーマット

(2X, 'SR', I3, I2, I5, 1X, I2, 1X, I2, 1X, I1, 32X, A26)

(4) サブレコード データ : サブレコードタイプ1 (分類サブレコード) の場合

レコード項目	フォーマット	内容
1 サブレコードデータ	I6	
2 注釈	A26	主分類コードと副分類コードを注釈として記述

レコードフォーマット

(5X, I6, 43X, A26)

(5) サブレコード データ : サブレコードタイプ1 (分類サブレコード) 以外の場合

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数 (short)
== 2	G15. 7	単精度浮動小数
== 3	G23. 15	倍精度浮動小数
== 4	I12	整数 (int)

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, I6)
== 2	(5X, G15. 7)
== 3	(5X, G23. 15)
== 4	(5X, I12)

● サブレコードヘッダーの継続フラグについて

注記要素の文字列を表現するために Advance CAD バージョン 17 まではサブレコードタイプ 11、12、13 (以下 SR11、SR12、SR13) の3つのサブレコードで表現していました。SR11 はデータモード1、SR12 はデータモード2、SR13 はデータモード0 の情報を保持しています。Advance CAD バージョン 18 以降の文字列はサブレコードタイプ 33 ひとつで、バージョン 17 での SR11、SR12、SR13 を表現します。ひとつのサブレコードで複数のデータモードを持つので複合型と呼びます。複合型サブレコードの時に継続フラグを使用します。

バージョン 17 以前

サブレコードタイプ	データモード	内容
11	1	テキストパラメータ
12	2	原点および枠
13	0	文字列

バージョン 18 以降

サブレコードタイプ	データモード	内容	継続フラグ
33	1	整数 (short) のパラメータ	1
33	3	倍精度実数のパラメータ	1
33	0	文字列	0

● サブレコード中の文字列の表現

文字列中の非印字文字や 8 単位文字 (全角文字やメタ文字) は次のように表現します。

- (1) 非印字文字 (16 進で 00 から 1f までと 7f) の表現
キーワード <hx> に続けて文字コードを 16 進で出力する。たとえば 16 進で 7f は <hx>7f と変換されて出力される。
ただし、16 進の 0d はキャリジリターンであり、<cr> と出力される。
- (2) 全角文字の表現
全角の部分をキーワード <ks> と <ke> でくくる。全角の部分は 8 単位を 7 単位に変換して (8 ビット目をオフにして) 出力する。たとえば、“漢字” という文字列は “<ks>4A:Z<ke>” と変換されて出力される。
- (3) メタ文字
メタ文字は、キーワード <mc> に続けてメタ文字 1 文字を出力する。たとえばメタ z は、<mc>z と変換されて出力される。

● サブレコードデータの変更

『プログラミングマニュアル』に記載されています。

Advance CAD バージョン 8 では、サブレコード 30 の内容に変更があります。

Advance CAD バージョン 9 では、ハッチングアイテムが大きく変更になり、サブレコード 19 が新たに追加されています。

Advance CAD バージョン 11 では、サブレコード 11 の内容に変更があります。

Advance CAD バージョン 12 では、倍精度浮動小数の出力桁数を 2 桁増やしました。また、データモードに int 型の整数を追加しました。

Advance CAD バージョン 18 では、SXF 仕様に対応するためにサブレコードタイプの見直しを行いました。その結果、いくつかのサブレコードタイプを廃止し、新しいサブレコードタイプを追加しました。またデータモードに複合型を追加しました。

Advance CAD バージョン 19 では、いくつかのサブレコードタイプを廃止し、新たなサブレコードタイプに統合しました。

8.2 参考：旧バージョンの共通データファイルの書式

8.2.1 参考：共通データファイル バージョン 1.0

● バージョン 1.0 の構造

モデルデータブロック
 ピクチャデータブロック
 ドローイングデータブロック
 アイテムデータブロック

● モデルデータブロック

(1) モデルデータブロックの構造

モデルタイトル
 マトリックス #1
 マトリックス #2
 ⋮
 マトリックス #N (N ≤ 16)
 ドローイング縮尺値

(2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル文字数	I4	1 -
3 モデルタイトル	A60	文字列

レコードフォーマット
 ('TITLE____', I4/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 16
3 X 方向余弦	3E16.6	(L1, M1, N1)
4 Y 方向余弦	3E16.6	(L2, M2, N2)
5 Z 方向余弦	3E16.6	(L3, M3, N3)

レコードフォーマット
 ('MATRIX____', I2, 3E16.6/12X, 3E16.6/12X, 3E16.6)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"

レコード項目	フォーマット	内容
2 ドローイング縮尺値	E16. 6	

レコードフォーマット
(' DRAWSCF____', E16. 6)

● ピクチャ データブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1
 ピクチャ レコード #2
 ⋮
 ピクチャ レコード #N (N ≤ 63)

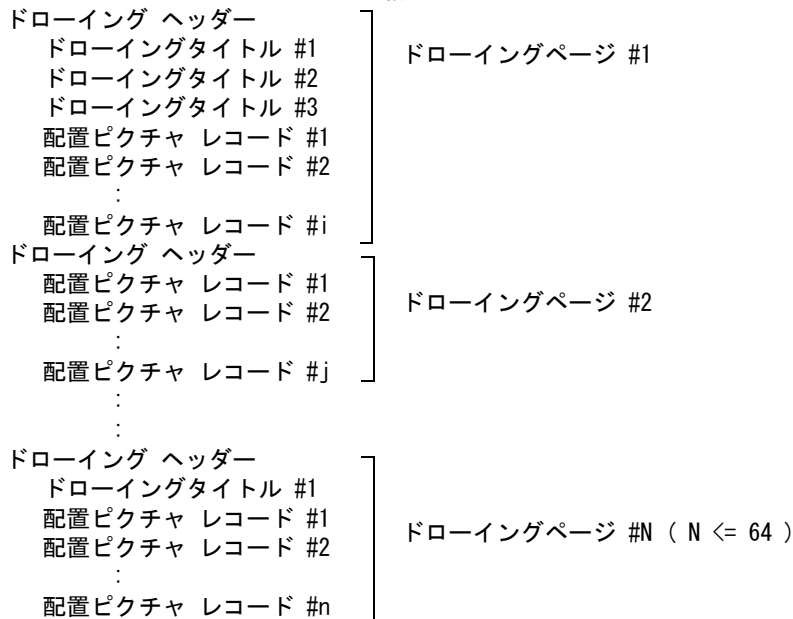
(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	E16. 6	
4 ゾーン原点	2E16. 6	Xorg, Yorg
5 マトリックス番号	I2	1 - 16
6 Z 値	E16. 6	
7 ピクチャゾーン	4E16. 6	Xmin, Ymin, Xmax, Ymax

レコードフォーマット
(' PICTURE____', I2, 3E16. 6, I2, E16. 6/10X, 4E16. 6)

● ドローイング データブロック

(1) ドローイング データブロックの構造



(1) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A6	"A0", "A1" . .
4 テンプレートサイズ	2E16. 6	width, height
5 ドローイングタイトル総数	I2	0 - 9
6 配置ピクチャ総数	I2	0 - 64

レコードフォーマット

('DRAWING__', I2, A6, 2E16. 6, 2I2)

(2) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル文字数	I4	-n - +n 0 : 無 正 : 表示 負 : 非表示
2 ドローイングタイトル	A60	文字列

レコードフォーマット

(10X, I4/(10X, A60))

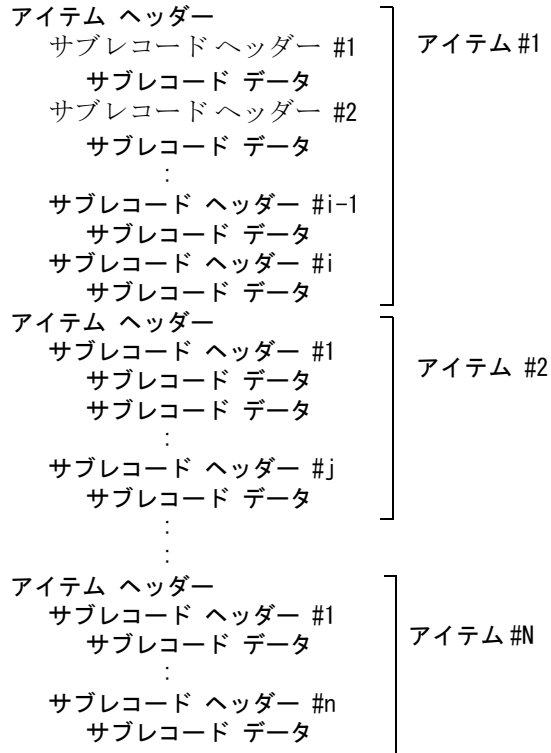
(3) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2E16.6	Xorg, Yorg

レコードフォーマット
(10X, I2, 2E16.6)

● アイテム データブロック

(1) アイテム データブロックの構造



(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I5	1 - 31
2 クラス	I5	1 - 255
3 ラインウエイト	I3	1 - 7
4 ラインフォント	I3	1 - 15
5 レビジョン	I3	1 - 127
6 ピクチャ	I3	1 - 63
7 ブランクフラグ	I2	0 - 1
8 特性フラグ	I2	0 - 1
9 アイテム名	A10	文字列

レコードフォーマット
(1X, 2I5, 4I3, 2I2, A10)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコード タイプ	I5	1 - 31
2 データ モード	I5	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I5	0 - 7
5 パーシャルウェイト	I5	0 - 3

レコードフォーマット
(5X, 5I5)

(4) サブレコード データ

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数
== 2	E16. 6	単精度浮動小数
== 3	E16. 10	倍精度浮動小数

レコードフォーマット	フォーマット
データモード == 0	(10X, A60)
== 1	(10X, 10I6)
== 2	(10X, 4E16. 6)
== 3	(10X, 4E16. 10)

8.2.2 参考：共通データファイルバージョン 2.0

● バージョン 2.0 の構造

バージョン番号 ブロック
 モデル データブロック
 ピクチャ データブロック
 ドローイング データブロック
 アイテム データブロック

● バージョン番号 ブロック

(1) バージョン番号 ブロックの構造

バージョン番号

(2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	2.0
3 注釈	A18	"__COMMON_DATA_FILE"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE')

● モデル データブロック

(1) モデル データブロックの構造

モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 63)
 ドローイング縮尺値

(2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE', 2I4/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 63
3 X 方向余弦	3G21. 13	(L1, M1, N1)
4 Y 方向余弦	3G21. 13	(L2, M2, N2)
5 Z 方向余弦	3G21. 13	(L3, M3, N3)

レコードフォーマット

(' MATRIX____', I2, 3G21. 13/12X, 3G21. 13/12X, 3G21. 13)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G15. 7	

レコードフォーマット

(' DRAWSCF____', G15. 7)

● ピクチャ データブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1

ピクチャ レコード #2

:

ピクチャ レコード #N (N <= 63)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	G15. 7	
4 ピクチャゾーン原点	2G15. 7	Xorg, Yorg
5 マトリックス番号	I2	1 - 63
6 Z 値	G21. 13	
7 ピクチャゾーン	4G15. 7	Xmin, Ymin, Xmax, Ymax

レコードフォーマット

(' PICTURE____', I2, 3G15. 7, 2X, G21. 13/12X, 4G15. 7)

● ドローイング データブロック

- (1) ドローイング データブロックの構造
- | | | |
|---|---|--------------------------|
| ドローイング ヘッダー
ドローイングタイトル #1
ドローイングタイトル #2
ドローイングタイトル #3
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #i | } | ドローイングページ #1 |
| ドローイング ヘッダー
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #j
:
:
ドローイング ヘッダー
ドローイングタイトル #1
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #n | } | ドローイングページ #2 |
| | } | ドローイングページ #N (N <= 64) |

(2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING____"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A6	"A0", "A1" . .
4 テンプレートサイズ	2G15.7	width,height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット

('DRAWING____', I2, 1X, A6, 2G15.7, 2I3)

(3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無
6 変更された配置位置	2G15.7	Xorg, Yorg

レコード項目	フォーマット	内容
7 ドローイングタイトル	A60	文字列

レコードフォーマット
(10X, 2I4, I2, G15. 7, I2, 2G15. 7/(10X, A60))

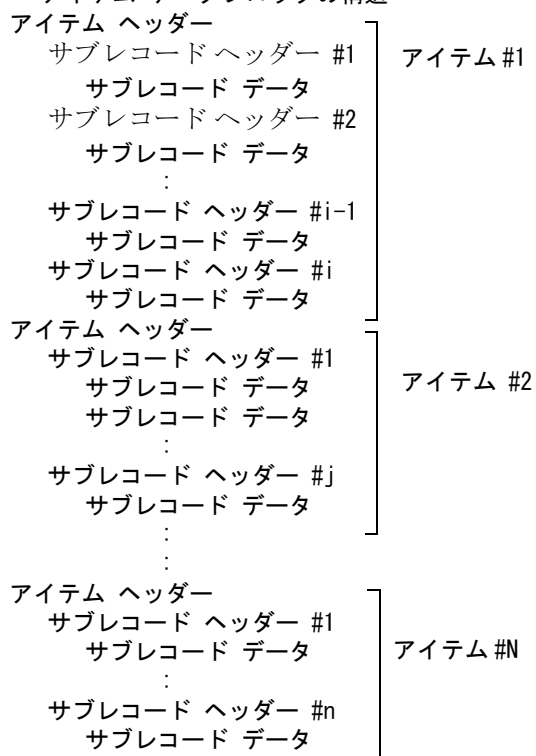
(4) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2G15. 7	Xorg, Yorg
3 ピクチャ回転角度 (度)	G15. 7	
4 配置ウインドウ番号	I3	0 - 255

レコードフォーマット
(10X, I2, 3G15. 7, 2X, I3)

● アイテム データブロック

(1) アイテム データブロックの構造



(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 31
2 クラス	I4	1 - 255
3 ラインウエイト	I2	1 - 7

レコード項目	フォーマット	内容
4 ラインフォント	I3	1 - 15
5 レビジョン	I4	1 - 127
6 ピクチャ	I3	1 - 63
7 ブランクフラグ	I2	0 - 1
8 特性フラグ	I2	0 - 1
9 アイテム名	A12	文字列

レコードフォーマット

(' ITEM', I3, I4, I2, I3, I4, I3, 2I2, 1X, A12)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコードタイプ	I3	1 - 31
2 データモード	I2	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 7
5 パーシャルウェイト	I2	0 - 3

レコードフォーマット

(2X, ' SR', I3, I2, I5, 2I2)

(4) サブレコード データ

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数
== 2	G15. 7	単精度浮動小数
== 3	G21. 13	倍精度浮動小数

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, 10I6)
== 2	(5X, 4G15. 7)
== 3	(5X, 2G21. 13)

8.2.3 参考：共通データファイル バージョン 3.0

● バージョン 3.0 の構造

バージョン番号 ブロック
 モデルデータブロック
 ピクチャデータブロック
 ドローイングデータブロック
 アイテムデータブロック

● バージョン番号 ブロック

(1) バージョン番号 ブロックの構造

バージョン番号

(2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	3.0
3 注釈	A39	"__COMMON_DATA_FILE, _Advance_CAD_ver_5.0"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_5.0')

● モデル データブロック

(1) モデル データブロックの構造

モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N <= 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N <= 63)
 ドローイング縮尺値

(2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE', 214/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 63
3 X 方向余弦	3G21. 13	(L1, M1, N1)
4 Y 方向余弦	3G21. 13	(L2, M2, N2)
5 Z 方向余弦	3G21. 13	(L3, M3, N3)

レコードフォーマット

('MATRIX____', I2, 3G21. 13/12X, 3G21. 13/12X, 3G21. 13)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G15. 7	

レコードフォーマット

('DRAWSCF____', G15. 7)

● ピクチャ データブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1

ピクチャ レコード #2

⋮

ピクチャ レコード #N (N ≤ 63)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	G15. 7	
4 ピクチャゾーン原点	2G15. 7	Xorg, Yorg
5 マトリックス番号	I2	1 - 63
6 Z 値	G21. 13	
7 ピクチャゾーン	4G15. 7	Xmin, Ymin, Xmax, Ymax

レコードフォーマット

('PICTURE____', I2, 3G15. 7, 2X, G21. 13/12X, 4G15. 7)

● ドローイング データブロック

(1) ドローイング データブロックの構造

ドローイング ヘッダー	}	ドローイングページ #1	
ドローイングタイトル #1			
ドローイングタイトル #2			
ドローイングタイトル #3			
配置ピクチャ レコード #1			
配置ピクチャ レコード #2	}	ドローイングページ #2	
⋮			
配置ピクチャ レコード #i			
ドローイング ヘッダー			
配置ピクチャ レコード #1			
配置ピクチャ レコード #2	}	ドローイングページ #N (N ≤ 64)	
⋮			
配置ピクチャ レコード #j			
⋮			
ドローイング ヘッダー			
ドローイングタイトル #1	}		
配置ピクチャ レコード #1			
配置ピクチャ レコード #2			
⋮			
配置ピクチャ レコード #n			

(2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A6	"A0", "A1" . .
4 テンプレートサイズ	2G15.7	width, height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット

('DRAWING__', I2, 1X, A6, 2G15.7, 2I3)

(3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無
6 変更された配置位置	2G15.7	Xorg, Yorg
7 ドローイングタイトル	A60	文字列

レコードフォーマット
(10X, 2I4, I2, G15. 7, I2, 2G15. 7/(10X, A60))

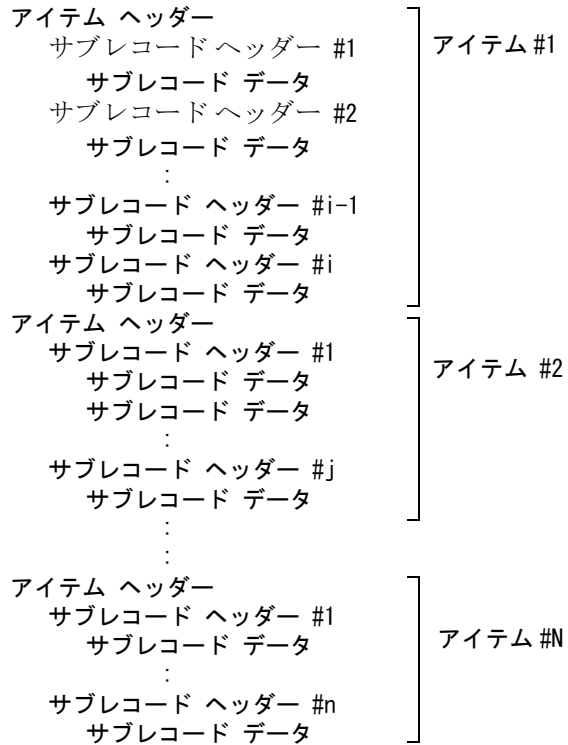
(4) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2G15. 7	Xorg, Yorg
3 ピクチャ回転角度 (度)	G15. 7	
4 配置ウインドウ番号	I3	0 - 255

レコードフォーマット
(10X, I2, 3G15. 7, 2X, I3)

● アイテム データブロック

(1) アイテム データブロックの構造



(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 31
2 クラス	I4	1 - 255
3 ラインウエイト	I2	1 - 7
4 ラインフォント	I3	1 - 15
5 レビジョン	I4	1 - 127
6 ピクチャ	I3	1 - 63

レコード項目	フォーマット	内容
7 ブランクフラグ	I2	0 - 1
8 特性フラグ	I2	0 - 1
9 アイテム名	A12	文字列

レコードフォーマット

('ITEM', I3, I4, I2, I3, I4, I3, 2I2, 1X, A12)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコードタイプ	I3	1 - 31
2 データモード	I2	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 7
5 パーシャルウェイト	I2	0 - 3

レコードフォーマット

(2X, 'SR', I3, I2, I5, 2I2)

(4) サブレコード データ

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数
== 2	G15. 7	単精度浮動小数
== 3	G21. 13	倍精度浮動小数

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, I6)
== 2	(5X, 4G15. 7)
== 3	(5X, 2G21. 13)

8.2.4 参考：共通データファイルバージョン 4.0

● バージョン 4.0 の構造

バージョン番号 ブロック
 モデル データブロック
 ピクチャ データブロック
 ドローイング データブロック
 アイテム データブロック

● バージョン番号 ブロック

(1) バージョン番号 ブロックの構造

バージョン番号

(2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	4.0
3 注釈	A40	"__COMMON_DATA_FILE, _Advance_CAD_ver_11.0"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_11.0')

● モデル データブロック

(1) モデル データブロックの構造

モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 63)
 ドローイング縮尺値

(2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE', 2I4/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 63
3 X 方向余弦	3G21. 13	(L1, M1, N1)
4 Y 方向余弦	3G21. 13	(L2, M2, N2)
5 Z 方向余弦	3G21. 13	(L3, M3, N3)

レコードフォーマット

(' MATRIX____', I2, 3G21. 13/12X, 3G21. 13/12X, 3G21. 13)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G15. 7	

レコードフォーマット

(' DRAWSCF____', G15. 7)

● ピクチャ データブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1

ピクチャ レコード #2

⋮

ピクチャ レコード #N (N ≤ 63)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	G15. 7	
4 ピクチャローカル原点	2G15. 7	Xorg, Yorg
5 Z 値	G21. 13	
6 表示範囲	4G15. 7	Xmin, Ymin, Xmax, Ymax

レコードフォーマット

(' PICTURE____', I2, 3G15. 7, 2X, G21. 13/12X, 4G15. 7)

● ドローイング データブロック

- (1) ドローイング データブロックの構造
- | | | |
|---|---|-------------------------|
| ドローイング ヘッダー
ドローイングタイトル #1
ドローイングタイトル #2
ドローイングタイトル #3
配置ピクチャ レコード #1
配置ピクチャ レコード #2
⋮
配置ピクチャ レコード #i | } | ドローイングページ #1 |
| ドローイング ヘッダー
配置ピクチャ レコード #1
配置ピクチャ レコード #2
⋮
配置ピクチャ レコード #j | } | ドローイングページ #2 |
| ⋮
ドローイング ヘッダー
ドローイングタイトル #1
配置ピクチャ レコード #1
配置ピクチャ レコード #2
⋮
配置ピクチャ レコード #n | } | ドローイングページ #N (N ≤ 64) |

(2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A6	"A0", "A1" . .
4 テンプレートサイズ	2G15.7	width,height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット
 ('DRAWING__', I2, 1X, A6, 2G15.7, 2I3)

(3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無
6 変更された配置位置	2G15.7	Xorg, Yorg
7 ドローイングタイトル	A60	文字列

レコードフォーマット
(10X, 2I4, I2, G15, 7, I2, 2G15, 7/(10X, A60))

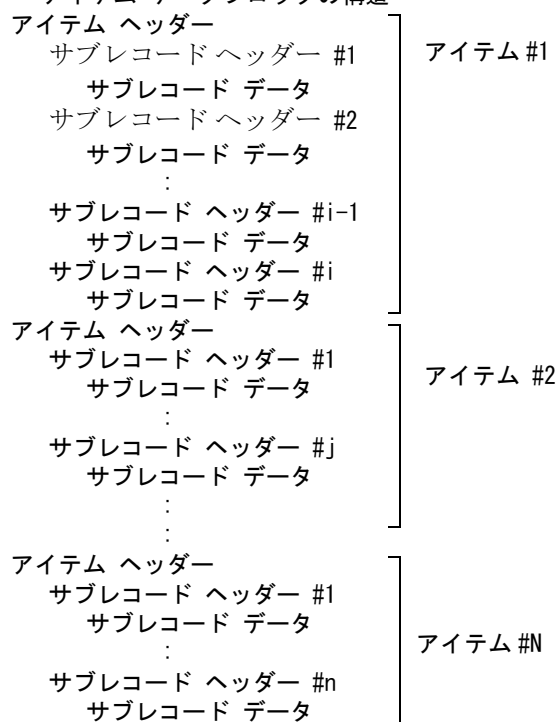
(4) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2G15.7	Xorg, Yorg
3 ピクチャ回転角度 (度)	G15.7	
4 配置ウインドウ番号	I3	0 - 255

レコードフォーマット
(10X, I2, 3G15.7, 2X, I3)

● アイテム データブロック

(1) アイテム データブロックの構造



(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 31
2 クラス	I4	1 - 255
3 ラインウエイト	I2	1 - 15
4 ラインフォント	I3	1 - 63
5 レビジョン	I4	1 - 127
6 ピクチャ	I3	1 - 63

レコード項目	フォーマット	内容
7 ブランクフラグ	I2	0 - 1
8 アイテム名	A12	文字列

レコードフォーマット
('ITEM' , I3, I4, 2I3, I4, I3, I2, 1X, A12)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコードタイプ	I3	1 - 31
2 データモード	I2	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 7
5 パーシャルウェイト	I2	0 - 3

レコードフォーマット
(2X, 'SR' , I3, I2, I5, 2I2)

(4) サブレコード データ

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数
== 2	G15. 7	単精度浮動小数
== 3	G21. 13	倍精度浮動小数

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, 10I6)
== 2	(5X, 4G15. 7)
== 3	(5X, 2G21. 13)

8.2.5 参考：共通データファイル バージョン 5.0

● バージョン 5.0 の構造

バージョン番号 ブロック
 モデルデータブロック
 ピクチャデータブロック
 ドローイングデータブロック
 アイテムデータブロック

● バージョン番号 ブロック

(1) バージョン番号 ブロックの構造

バージョン番号

(2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	5.0
3 注釈	A40	"__COMMON_DATA_FILE, _Advance_CAD_ver_11.02"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_11.02')

● モデルデータブロック

(1) モデルデータブロックの構造

モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 63)
 ドローイング縮尺値

(2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE', 214/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 63
3 X 方向余弦	3G21. 13	(L1, M1, N1)
4 Y 方向余弦	3G21. 13	(L2, M2, N2)
5 Z 方向余弦	3G21. 13	(L3, M3, N3)

レコードフォーマット

(' MATRIX____', I2, 3G21. 13/12X, 3G21. 13/12X, 3G21. 13)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G15. 7	

レコードフォーマット

(' DRAWSCF____', G15. 7)

● ピクチャ データブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1

ピクチャ レコード #2

⋮

ピクチャ レコード #N (N ≤ 63)

(2) ピクチャ レコード

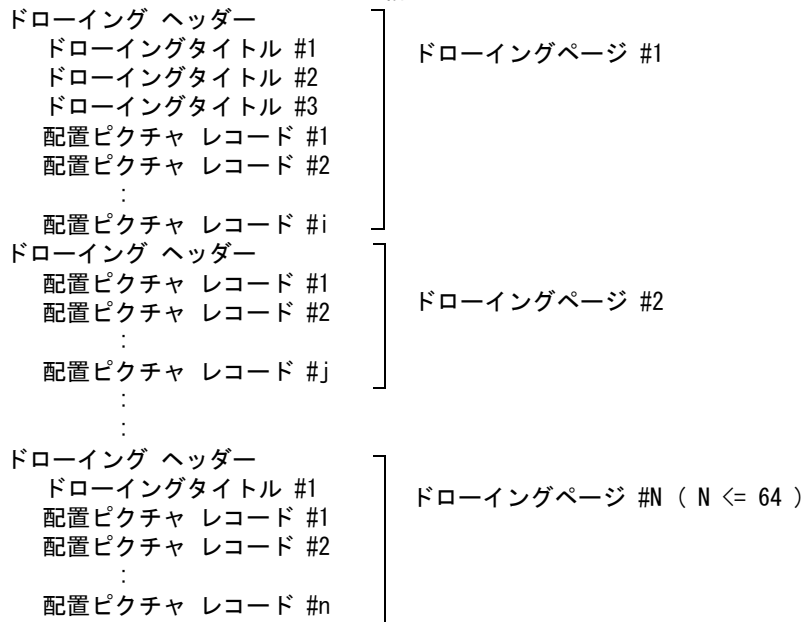
レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	G15. 7	
4 ピクチャローカル原点	2G15. 7	Xorg, Yorg
5 Z 値	G21. 13	
6 表示範囲	4G15. 7	Xmin, Ymin, Xmax, Ymax

レコードフォーマット

(' PICTURE____', I2, 3G15. 7, 2X, G21. 13/12X, 4G15. 7)

● ドローイング データブロック

(1) ドローイング データブロックの構造



(2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A6	"A0", "A1" . .
4 テンプレートサイズ	2G15.7	width, height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット

('DRAWING__', I2, 1X, A6, 2G15.7, 2I3)

(3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無
6 変更された配置位置	2G15.7	Xorg, Yorg
7 ドローイングタイトル	A60	文字列

レコードフォーマット
(10X, 2I4, I2, G15. 7, I2, 2G15. 7/(10X, A60))

(4) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2G15. 7	Xorg, Yorg (ドローイング座標)
3 ピクチャ回転角度 (度)	G15. 7	
4 配置ウィンドウ番号	I3	0 - 255
5 ウィンドウゾーン	4G21. 13	Xmin, Ymin, Xma, Yma (モデル座標)
6 ウィンドウ原点	2G21. 13	Xorg, Yorg (モデル座標)

レコードフォーマット
(10X, I2, 3G15. 7, 2X, I3/5X, 3G21. 13/5X, 3G21. 13)

● アイテム データブロック

(1) アイテム データブロックの構造

アイテム ヘッダー	
サブレコード ヘッダー #1	アイテム #1
サブレコード データ	
サブレコード ヘッダー #2	
サブレコード データ	
:	
サブレコード ヘッダー #i-1	アイテム #2
サブレコード データ	
サブレコード ヘッダー #i	
サブレコード データ	
:	
サブレコード ヘッダー #j	アイテム #N
サブレコード データ	
:	
:	
:	
アイテム ヘッダー	
サブレコード ヘッダー #1	アイテム #N
サブレコード データ	
:	
サブレコード ヘッダー #n	
サブレコード データ	

(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 31
2 クラス	I4	1 - 255
3 ラインウエイト	I3	1 - 15

レコード項目	フォーマット	内容
4 ラインフォント	I3	1 - 63
5 レビジョン	I4	1 - 127
6 ピクチャ	I3	1 - 63
7 ブランクフラグ	I2	0 - 1
8 アイテム名	A12	文字列

レコードフォーマット

('ITEM', I3, I4, 2I3, I4, I3, I2, 1X, A12)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコードタイプ	I3	1 - 31
2 データモード	I2	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 7
5 パーシャルウェイト	I2	0 - 3

レコードフォーマット

(2X, 'SR', I3, I2, I5, 2I2)

(4) サブレコード データ

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数
== 2	G15. 7	単精度浮動小数
== 3	G21. 13	倍精度浮動小数

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, 10I6)
== 2	(5X, 4G15. 7)
== 3	(5X, 2G21. 13)

8.2.6 参考：共通データファイルバージョン 6.0

● バージョン 6.0 の構造

バージョン番号 ブロック
 モデル データブロック
 ピクチャ データブロック
 ドローイング データブロック
 アイテム データブロック

● バージョン番号 ブロック

(1) バージョン番号 ブロックの構造

バージョン番号

(2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	6.0
3 注釈	A40	"__COMMON_DATA_FILE, _Advance_CAD_ver_12.0"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_12.0')

● モデル データブロック

(1) モデル データブロックの構造

モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 63)
 ドローイング縮尺値

(2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE', 2I4/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 63
3 X 方向余弦	3G23. 15	(L1, M1, N1)
4 Y 方向余弦	3G23. 15	(L2, M2, N2)
5 Z 方向余弦	3G23. 15	(L3, M3, N3)

レコードフォーマット

(' MATRIX____', I2/5X, 3G23. 15/5X, 3G23. 15/5X, 3G23. 15)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G15. 7	

レコードフォーマット

(' DRAWSCF____', G15. 7)

● ピクチャ データブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1

ピクチャ レコード #2

⋮

ピクチャ レコード #N (N ≤ 63)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	G15. 7	
4 ピクチャローカル原点	2G15. 7	Xorg, Yorg
5 Z 値	G23. 15	
6 表示範囲	4G15. 7	Xmin, Ymin, Xmax, Ymax

レコードフォーマット

(' PICTURE____', I2, 3G15. 7, G23. 15/12X, 4G15. 7)

● ドローイング データブロック

- (1) ドローイング データブロックの構造
- | | | |
|----------------|---|-------------------------|
| ドローイング ヘッダー | } | ドローイングページ #1 |
| ドローイングタイトル #1 | | |
| ドローイングタイトル #2 | | |
| ドローイングタイトル #3 | | |
| 配置ピクチャ レコード #1 | | |
| 配置ピクチャ レコード #2 | } | ドローイングページ #2 |
| 配置ピクチャ レコード #i | | |
| ドローイング ヘッダー | | |
| 配置ピクチャ レコード #1 | | |
| 配置ピクチャ レコード #2 | | |
| 配置ピクチャ レコード #j | } | ドローイングページ #N (N ≤ 64) |
| 配置ピクチャ レコード #n | | |
| ドローイング ヘッダー | | |
| ドローイングタイトル #1 | | |
| 配置ピクチャ レコード #1 | | |
| 配置ピクチャ レコード #2 | | |
| 配置ピクチャ レコード #n | | |

(2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A6	"A0", "A1" . .
4 テンプレートサイズ	2G15. 7	width, height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット
('DRAWING__', I2, I3, A6, 2G15. 7, 2I3)

(3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15. 7	
5 配置位置変更の有無	I2	1: 有 0: 無
6 変更された配置位置	2G15. 7	Xorg, Yorg

レコード項目	フォーマット	内容
7 ドローイングタイトル	A60	文字列

レコードフォーマット

(10X, 2I4, I2, G15. 7, I2, 2G15. 7/(10X, A60))

(4) 配置ピクチャ レコード

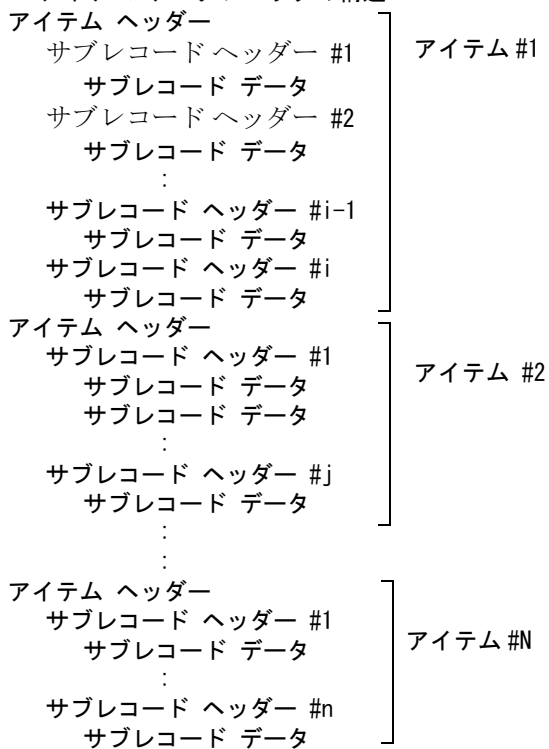
レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2G15. 7	Xorg, Yorg (ドローイング座標)
3 ピクチャ回転角度 (度)	G15. 7	
4 配置ウインドウ番号	I3	0 - 255
5 ウィンドウゾーン	4G21. 13	Xmin, Ymin, Xma, Yma (モデル座標)
6 ウィンドウ原点	2G21. 13	Xorg, Yorg (モデル座標)

レコードフォーマット

(10X, I2, 3G15. 7, 2X, I3/5X, 3G21. 13/5X, 3G21. 13)

● アイテム データブロック

(1) アイテム データブロックの構造



(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 31
2 クラス	I4	1 - 255
3 ラインウエイト	I3	1 - 15
4 ラインフォント	I3	1 - 63
5 レビジョン	I4	1 - 127
6 ピクチャ	I3	1 - 63
7 ブランクフラグ	I2	0 - 1
8 アイテム名	A12	文字列

レコードフォーマット

(' ITEM', I3, I4, 2I3, I4, I3, I2, 1X, A12)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコードタイプ	I3	1 - 31
2 データモード	I2	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 64
5 パーシャルウエイト	I2	0 - 15

レコードフォーマット

(2X, ' SR', I3, I2, I5, 1X, I2, 1X, I2)

(4) サブレコード データ

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数 (short)
== 2	G15. 7	単精度浮動小数
== 3	G23. 15	倍精度浮動小数
== 4	I12	整数 (int)

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, 10I6)

レコードフォーマット	フォーマット
== 2	(5X, 4G15. 7)
== 3	(5X, 2G23. 15)
== 4	(5X, I12)

8.2.7 参考：共通データファイルバージョン 7.0

● バージョン 7.0 の構造

バージョン番号 ブロック
 モデル データブロック
 ピクチャ データブロック
 ドローイング データブロック
 アイテム データブロック

● バージョン番号ブロック

- (1) バージョン番号 ブロックの構造
バージョン番号
- (2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	7.0
3 注釈	A40	"__COMMON_DATA_FILE, _Advance_CAD_ver_13.0"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_13.0')

● モデル データブロック

- (1) モデル データブロックの構造
 モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 63)
 ドローイング縮尺値
- (2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE ', 214/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 63
3 X 方向余弦	3G23. 15	(L1, M1, N1)
4 Y 方向余弦	3G23. 15	(L2, M2, N2)
5 Z 方向余弦	3G23. 15	(L3, M3, N3)

レコードフォーマット

(' MATRIX____', I2/5X, 3G23. 15/5X, 3G23. 15/5X, 3G23. 15)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G15. 7	

レコードフォーマット

(' DRAWSCF____', G15. 7)

● ピクチャデータブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1
 ピクチャ レコード #2
 ⋮
 ピクチャ レコード #N (N ≤ 63)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	G15.7	
4 ピクチャローカル原点	2G15.7	Xorg,Yorg
5 Z 値	G23.15	
6 表示範囲	4G15.7	Xmin,Ymin,Xmax,Ymax

レコードフォーマット

(' PICTURE____', I2, 3G15. 7, G23. 15/12X, 4G15. 7)

● ドローイング データブック

- (1) ドローイング データブロックの構造
- | | | |
|---|---|-------------------------|
| ドローイング ヘッダー
ドローイングタイトル #1
ドローイングタイトル #2
ドローイングタイトル #3
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #i | } | ドローイングページ #1 |
| ドローイング ヘッダー
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #j
:
:
ドローイング ヘッダー
ドローイングタイトル #1
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #n | } | ドローイングページ #2 |
| :
:
:
ドローイング ヘッダー
ドローイングタイトル #1
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #n | } | ドローイングページ #N (N ≤ 64) |

- (2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING____"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A6	"A0","A1" . .
4 テンプレートサイズ	2G15.7	width,height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット
 ('DRAWING____', I2, 1X, A6, 2G15.7, 2I3)

- (3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無

レコード項目	フォーマット	内容
6 変更された配置位置	2G15. 7	Xorg, Yorg
7 ドローイングタイトル	A60	文字列

レコードフォーマット

(10X, 2I4, I2, G15. 7, I2, 2G15. 7/(10X, A60))

(4) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2G15. 7	Xorg, Yorg (ドローイング座標)
3 ピクチャ回転角度 (度)	G15. 7	
4 配置ウインドウ番号	I3	0 - 255
5 ウインドウゾーン	4G23. 15	Xmin, Ymin, Xmax, Ymax (モデル座標)
6 ウインドウ原点	2G23. 15	Xorg, Yorg (モデル座標)

レコードフォーマット

(10X, I2, 3G15. 7, 2X, I3/5X, 3G23. 15/5X, 3G23. 15)

● アイテム データブロック

(1) アイテム データブロックの構造

アイテム ヘッダー サブレコード ヘッダー #1 サブレコード データ サブレコード ヘッダー #2 サブレコード データ ⋮ サブレコード ヘッダー #i-1 サブレコード データ サブレコード ヘッダー #i サブレコード データ	アイテム #1
アイテム ヘッダー サブレコード ヘッダー #1 サブレコード データ サブレコード データ ⋮ サブレコード ヘッダー #j サブレコード データ	アイテム #2
アイテム ヘッダー サブレコード ヘッダー #1 サブレコード データ ⋮ サブレコード ヘッダー #N サブレコード データ	アイテム #N

(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 31
2 クラス	I4	1 - 255
3 ラインウエイト	I3	1 - 15
4 ラインフォント	I3	1 - 63
5 レビジョン	I4	1 - 127
6 ピクチャ	I3	1 - 63
7 ブランクフラグ	I2	0 - 1
8 アイテム名	A12	文字列

レコードフォーマット

(' ITEM', I3, I4, 2I3, I4, I3, I2, 1X, A12)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコード タイプ	I3	1 - 31
2 データモード	I2	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 64
5 パーシャルウエイト	I2	0 - 15

レコードフォーマット

(2X, ' SR', I3, I2, I5, 1X, I2, 1X, I2)

(4) サブレコード データ : 塗り潰しパラメータ以外の時

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数 (short)
== 2	G15. 7	単精度浮動小数
== 3	G23. 15	倍精度浮動小数
== 4	I12	整数 (int)

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, I6)

レコードフォーマット	フォーマット
== 2	(5X, 4G15. 7)
== 3	(5X, 2G23. 15)
== 4	(5X, I12)

(5) サブレコード データ : 塗り潰しパラメータ (サブレコードタイプ 18) の時

レコード項目	フォーマット	内容
1 パターンの種類	I1	== 0 : ハードウェアパターン == 1 : マーク == 2 : 文字
2 テキストフォント番号	I3	
3 パターン番号	I6	ハードウェアパターンの時 : プロッタ/プリンタの塗り潰し パターン番号。(0 - 32767) マークの時 : マーク番号。(1 - 4095) 半角文字の時 : 十進表記の ASCII 文字コード * 256 + 32。 例えば "A" のときは 65 * 256 + 32 で 16672 になる。 全角文字の時 : 全角文字 2 バイトの EUC コードを符号付き整数 (short) で表す。 例えば "山" は EUC コードで BBB3。符号付き整数では -17485。
4 配置角度 (度)	G15. 7	
5 配置基準点 X	G15. 7	
6 配置基準点 Y	G15. 7	
7 大きさ X	G15. 7	
8 大きさ Y	G15. 7	
9 間隔 X	G15. 7	
10 間隔 Y	G15. 7	

レコードフォーマット

(5X, 3X, I1, 1X, I3, 1X, I6, 3G15. 7/5X, 4G15. 7)

8.2.8 参考：共通データファイルバージョン 8.0

● バージョン 8.0 の構造

バージョン番号 ブロック
 モデル データブロック
 ピクチャ データブロック
 ドローイング データブロック
 アイテム データブロック

● バージョン番号ブロック

- (1) バージョン番号 ブロックの構造
バージョン番号
- (2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	8.0
3 注釈	A40	"__COMMON_DATA_FILE, _Advance_CAD_ver_15.0"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_15.0')

● モデル データブロック

- (1) モデル データブロックの構造
 モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 63)
 ドローイング縮尺値
- (2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE ', 214/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I2	1 - 63
3 X 方向余弦	3G23. 15	(L1, M1, N1)
4 Y 方向余弦	3G23. 15	(L2, M2, N2)
5 Z 方向余弦	3G23. 15	(L3, M3, N3)

レコードフォーマット

(' MATRIX____', I2/5X, 3G23. 15/5X, 3G23. 15/5X, 3G23. 15)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G15. 7	

レコードフォーマット

(' DRAWSCF____', G15. 7)

● ピクチャデータブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1

ピクチャ レコード #2

⋮

ピクチャ レコード #N (N <= 63)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I2	1 - 63
3 ピクチャ縮尺値	G15. 7	
4 ピクチャローカル原点	2G15. 7	Xorg, Yorg
5 Z 値	G23. 15	
6 表示範囲	4G15. 7	Xmin, Ymin, Xmax, Ymax

レコードフォーマット

(' PICTURE____', I2, 3G15. 7, G23. 15/12X, 4G15. 7)

● ドローイング データブック

- (1) ドローイング データブロックの構造
- | | | |
|---|---|-------------------------|
| ドローイング ヘッダー
ドローイングタイトル #1
ドローイングタイトル #2
ドローイングタイトル #3
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #i | } | ドローイングページ #1 |
| ドローイング ヘッダー
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #j | } | ドローイングページ #2 |
| :
ドローイング ヘッダー
ドローイングタイトル #1
配置ピクチャ レコード #1
配置ピクチャ レコード #2
:
配置ピクチャ レコード #n | } | ドローイングページ #N (N ≤ 64) |

- (2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A32	"A0", "A1" . .
4 テンプレートサイズ	2G15.7	width,height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット
 ('DRAWING__', I2, 1X, A32, 2G15.7, 2I3)

- (3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無
6 変更された配置位置	2G15.7	Xorg, Yorg

レコード項目	フォーマット	内容
7 ドローイングタイトル	A60	文字列

レコードフォーマット
(10X, 2I4, I2, G15. 7, I2, 2G15. 7/(10X, A60))

(4) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I2	1 - 63
2 ピクチャ配置位置	2G15. 7	Xorg, Yorg (ドローイング座標)
3 ピクチャ回転角度 (度)	G15. 7	
4 配置ウインドウ番号	I3	0 - 255
5 ウインドウ領域左下	2G23. 15	Xpos, Ypos (モデル座標)
5 ウインドウサイズ	2G23. 15	width, height (モデル座標)
5 ウインドウ原点	2G23. 15	Xorg, Yorg (モデル座標)
6 ウインドウ角度 (度)	G23. 15	

レコードフォーマット
(10X, I2, 3G15. 7, 2X, I3/5X, 3G23. 15/5X, 3G23. 15)

● アイテム データブロック

(1) アイテム データブロックの構造

アイテム ヘッダー サブレコード ヘッダー #1 サブレコード データ サブレコード ヘッダー #2 サブレコード データ ⋮ サブレコード ヘッダー #i-1 サブレコード データ サブレコード ヘッダー #i サブレコード データ	アイテム #1
アイテム ヘッダー サブレコード ヘッダー #1 サブレコード データ サブレコード データ ⋮ サブレコード ヘッダー #j サブレコード データ	アイテム #2
アイテム ヘッダー サブレコード ヘッダー #1 サブレコード データ ⋮ サブレコード ヘッダー #N サブレコード データ	アイテム #N

(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 31
2 クラス	I4	1 - 255
3 ラインウエイト	I3	1 - 15
4 ラインフォント	I3	1 - 63
5 レビジョン	I4	1 - 127
6 ピクチャ	I3	1 - 63
7 ブランクフラグ	I2	0 - 1
8 アイテム名	A12	文字列

レコードフォーマット

(' ITEM', I3, I4, 2I3, I4, I3, I2, 1X, A12)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコード タイプ	I3	1 - 31
2 データモード	I2	0 - 3
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 64
5 パーシャルウエイト	I2	0 - 15

レコードフォーマット

(2X, ' SR', I3, I2, I5, 1X, I2, 1X, I2)

(4) サブレコード データ：塗り潰しパラメータ以外の時

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数 (short)
== 2	G15. 7	単精度浮動小数
== 3	G23. 15	倍精度浮動小数
== 4	I12	整数 (int)

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, I6)

レコードフォーマット	フォーマット
== 2	(5X, 4G15. 7)
== 3	(5X, 2G23. 15)
== 4	(5X, I12)

(5) サブレコード データ : 塗り潰しパラメータ (サブレコードタイプ 18) の時

レコード項目	フォーマット	内容
1 パターンの種類	I1	== 0 : ハードウェアパターン == 1 : マーク == 2 : 文字
2 テキストフォント番号	I3	
3 パターン番号	I6	ハードウェアパターンの時 : プロッタ/プリンタの塗り潰し パターン番号。(0 - 32767) マークの時 : マーク番号。(1 - 4095) 半角文字の時 : 十進表記の ASCII 文字コード * 256 + 32。 例えば "A" のときは 65 * 256 + 32 で 16672 になる。 全角文字の時 : 全角文字 2 バイトの EUC コードを符号付き整数 (short) で表す。 例えば "山" は EUC コードで BBB3。符号付き整数では -17485。
4 配置角度 (度)	G15. 7	
5 配置基準点 X	G15. 7	
6 配置基準点 Y	G15. 7	
7 大きさ X	G15. 7	
8 大きさ Y	G15. 7	
9 間隔 X	G15. 7	
10 間隔 Y	G15. 7	

レコードフォーマット

(5X, 3X, I1, 1X, I3, 1X, I6, 3G15. 7/5X, 4G15. 7)

8.2.9 参考：共通データファイルバージョン 9.0

● バージョン 9.0 の構造

バージョン番号 ブロック
 モデル データブロック
 ピクチャ データブロック
 ドローイング データブロック
 アイテム データブロック

● バージョン番号ブロック

- (1) バージョン番号 ブロックの構造
バージョン番号
- (2) バージョン番号

レコード項目	フォーマット	内容
1 識別文字	A8	"VERSION_"
2 バージョン番号	F3.1	9.0
3 注釈	A40	"__COMMON_DATA_FILE, _Advance_CAD_ver_18.0"

レコードフォーマット

('VERSION_', F3.1, '__COMMON_DATA_FILE, _Advance_CAD_ver_18.0')

● モデル データブロック

- (1) モデル データブロックの構造
 モデルタイトル #1
 モデルタイトル #2
 :
 モデルタイトル #N (N ≤ 255)
 マトリックス #1
 マトリックス #2
 :
 マトリックス #N (N ≤ 256)
 ドローイング縮尺値
- (2) モデルタイトル

レコード項目	フォーマット	内容
1 識別文字列	A10	"TITLE____"
2 タイトル番号	I4	1 - 255
3 タイトル文字数	I4	1 -
4 モデルタイトル	A60	文字列

レコードフォーマット

('TITLE ', 214/(10X, A60))

(3) マトリックス

レコード項目	フォーマット	内容
1 識別文字列	A10	"MATRIX____"
2 マトリックス番号	I3	1 - 256
3 X 方向余弦	3G23. 15	(L1, M1, N1)
4 Y 方向余弦	3G23. 15	(L2, M2, N2)
5 Z 方向余弦	3G23. 15	(L3, M3, N3)

レコードフォーマット

(' MATRIX____', I3/5X, 3G23. 15/5X, 3G23. 15/5X, 3G23. 15)

(4) ドローイング縮尺値

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWSCF____"
2 ドローイング縮尺値	G23. 15	

レコードフォーマット

(' DRAWSCF____', G23. 15)

● ピクチャデータブロック

(1) ピクチャ データブロックの構造

ピクチャ レコード #1
 ピクチャ レコード #2
 :
 ピクチャ レコード #N (N <= 256)

(2) ピクチャ レコード

レコード項目	フォーマット	内容
1 識別文字列	A10	"PICTURE____"
2 ピクチャ番号	I3	1 - 256
3 ピクチャ縮尺値	G23. 15	
4 ピクチャローカル原点	2G23. 15	Xorg, Yorg
5 Z 値	G23. 15	
6 表示範囲	4G23. 15	Xmin, Ymin, Xmax, Ymax

レコードフォーマット

(' PICTURE____', I3/5X, 3G23. 15/5X, 3G23. 15/5X, 2G23. 15)

● ドローイング データブック

(1) ドローイング データブロックの構造

ドローイング ヘッダー		
ドローイングタイトル #1	}	ドローイングページ #1
ドローイングタイトル #2		
ドローイングタイトル #3		
配置ピクチャ レコード #1		
配置ピクチャ レコード #2		
⋮		
配置ピクチャ レコード #i		
ドローイング ヘッダー		
配置ピクチャ レコード #1	}	ドローイングページ #2
配置ピクチャ レコード #2		
⋮		
配置ピクチャ レコード #j		
⋮		
⋮		
ドローイング ヘッダー		
ドローイングタイトル #1	}	ドローイングページ #N (N ≤ 64)
配置ピクチャ レコード #1		
配置ピクチャ レコード #2		
⋮		
配置ピクチャ レコード #n		

(2) ドローイング ヘッダー

レコード項目	フォーマット	内容
1 識別文字列	A10	"DRAWING__"
2 ドローイングページ番号	I2	1 - 64
3 テンプレート名	A32	"A0", "A1" . .
4 テンプレートサイズ	2G15.7	width,height
5 ドローイングタイトル総数	I3	0 - 255
6 配置ピクチャ総数	I3	0 - 128

レコードフォーマット

('DRAWING__', I2, 1X, A32, 2G15.7, 2I3)

(3) ドローイングタイトル

レコード項目	フォーマット	内容
1 タイトル番号	I4	1 - 255
2 タイトル文字数	I4	-n - +n 0: 無 正: 表示 負: 非表示
3 文字高さ変更の有無	I2	1: 有 0: 無
4 変更された文字高さ	G15.7	
5 配置位置変更の有無	I2	1: 有 0: 無
6 変更された配置位置	2G15.7	Xorg, Yorg
7 ドローイングタイトル	A60	文字列

レコードフォーマット
(10X, 214, 12, G15. 7, 12, 2G15. 7/(10X, A60))

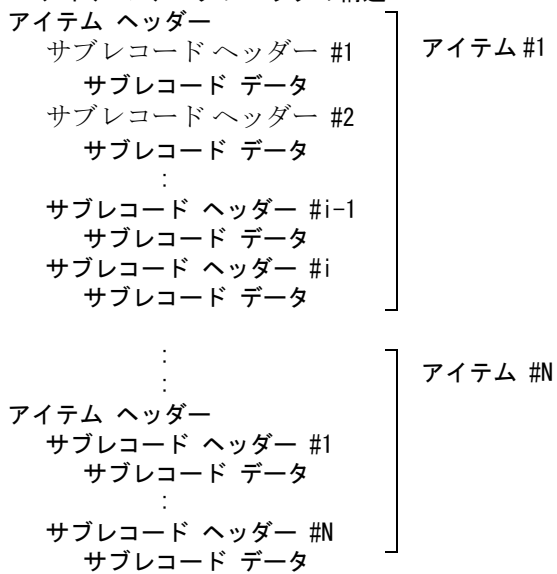
(4) 配置ピクチャ レコード

レコード項目	フォーマット	内容
1 配置ピクチャ番号	I3	1 - 256
2 配置ウインドウ番号	I3	0 - 255
3 配置位置	2G23. 15	Xorg, Yorg (ドローイング座標)
4 配置角度 (度)	G23. 15	
5 配置スケール	G23. 15	
6 ウインドウ領域左下	2G23. 15	Xpos, Ypos (モデル座標)
7 ウインドウサイズ	2G23. 15	width, height (モデル座標)
8 ウインド原点	2G23. 15	Xorg, Yorg (モデル座標)
9 ウインドウ角度 (度)	G23. 15	

レコードフォーマット
(10X, I3, 1X, I3/5X, 3G23. 15/5X, 3G23. 15/5X, 3G23. 15/5X, 2G23. 15)

● アイテム データブロック

(1) アイテム データブロックの構造



(2) アイテム ヘッダー

レコード項目	フォーマット	内容
1 アイテムタイプ	I3	1 - 63
2 クラス	I4	1 - 256
3 ラインウエイト	I3	1 - 16
4 ラインフォント	I3	1 - 63

レコード項目	フォーマット	内容
5 レビジョン	I4	1 - 256
6 ピクチャ	I4	1 - 320
7 ブランクフラグ	I2	0 - 1
8 アイテム名	A12	文字列
9 注釈	A26	アイテムタイプの略称を注釈として記述

レコードフォーマット
(' ITEM' , I3, I4, 2I3, 2I4, I2, 1X, A12, 14X, A26)

(3) サブレコード ヘッダー

レコード項目	フォーマット	内容
1 サブレコード タイプ	I3	1 - 63
2 データモード	I2	0 - 4
3 データ数	I5	0 -
4 パーシャルフォント	I2	0 - 64
5 パーシャルウェイト	I2	0 - 16
6 継続フラグ (後述)	I1	1: 継続 0: 最終
7 注釈	A26	サブレコードタイプの略称を注釈として記述

レコードフォーマット
(2X, ' SR' , I3, I2, I5, 1X, I2, 1X, I2, 1X, I1, 32X, A26)

(4) サブレコード データ : サブレコードタイプ1 (分類サブレコード) の場合

レコード項目	フォーマット	内容
1 サブレコードデータ	I6	
2 注釈	A26	主分類コードと副分類コードを注釈として記述

レコードフォーマット
(5X, I6, 43X, A26)

(5) サブレコード データ : サブレコードタイプ1 (分類サブレコード) 以外の場合

レコード項目	フォーマット	内容
1 サブレコードデータ		
データモード == 0	A60	文字列
== 1	I6	整数 (short)
== 2	G15. 7	単精度浮動小数
== 3	G23. 15	倍精度浮動小数

レコード項目	フォーマット	内容
== 4	I12	整数 (int)

レコードフォーマット	フォーマット
データモード == 0	(5X, A60)
== 1	(5X, 10I6)
== 2	(5X, 4G15. 7)
== 3	(5X, 2G23. 15)
== 4	(5X, 5I12)

・サブレコードヘッダーの継続フラグについて

注記要素の文字列を表現するために Advance CAD バージョン 17 まではサブレコードタイプ 11、12、13（以下 SR11、SR12、SR13）の 3 つのサブレコードで表現していました。SR11 はデータモード 1、SR12 はデータモード 2、SR13 はデータモード 0 の情報を保持しています。

Advance CAD バージョン 18 の文字列はサブレコードタイプ 33 ひとつで、バージョン 17 での SR11、SR12、SR13 を表現します。ひとつのサブレコードで複数のデータモードを持つので複合型と呼びます。複合型サブレコードの時に継続フラグを使用します。

バージョン 17 以前

サブレコードタイプ	データモード	内容
11	1	テキストパラメータ
12	2	原点および枠
13	0	文字列

バージョン 18

サブレコードタイプ	データモード	内容	継続フラグ
33	1	整数 (short) のパラメータ	1
33	3	倍精度実数のパラメータ	1
33	0	文字列	0

第 9 章 イメージファイル

● 概要

Advance CAD の画面をそのままラスターファイルに出力することができます。
ラスターファイルは技術文書を作るときの挿入図として使用できます。

9.1 イメージファイルの作成

● コマンド一覧

コマンド名	機能
DOC / RAS	ラスターファイルを作る
DOC / RAS 2	ラスターファイルを作る
DRAW / RAS	図面配置状態のラスターファイルを作る
DOC / EPS	Adobe Illustrator 用の EPS (Encapslated PostScript file) を作成する

9.1.1 ラスターファイルを作成する

【メニユ一】

〔ラスタ背景〕→〔ラスタ出力〕

【構文】

DOC/RAS	[FNM]	filename	$\left\{ \begin{array}{l} \text{TMPPLN} \\ \text{ACTPLN} \\ \text{TIFF} \\ \text{TIFFG4} \\ \text{BMP} \\ \text{JPEG} \end{array} \right\}$	$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{BOX p1 p2} \\ \text{CURPIC} \end{array} \right\} [\text{SZIE}] \text{ s1 } [\text{s2}] \\ \text{ALLPIC} \\ \text{ALL} \\ \text{INSIDE } [d]^+ \end{array} \right\}$	<CE>
---------	-------	----------	---	--	------

出力ファイルを指定する。

FNM filename : 出力するファイル名を入力する。出力ファイル名はコンフィグレーションファイル ACAD.SET ファイルの #DOC:RASTER# キーワードで指示されたディレクトリへ作成される。

出力ファイルのフォーマットを指定する。

TMPPLN	:	モノクローム (1 ビット) の SUN ラスターファイルで出力する。(省略時)
ACTPLN	:	カラー (8 ビット) の SUN ラスターファイルで出力する。
TIFF	:	カラー (8 ビット) の TIFF ファイル (uncompress 形式) で出力する。
TIFFG4	:	モノクローム (1 ビット) の TIFF G4 形式で出力する。
BMP	:	Microsoft Bitmap ファイルで出力する。Microsoft Bitmap フォーマットには、DIB (Device Independent Bitmap)、DDB の 2 種類があるが、ここで出力できるのは DIB のみである。
JPEG	:	カラー (24 ビット) の ISO JPEG 標準形式で出力する。

出力範囲を以下のいずれかで指定する。

BBOX p1 p2	: 対角の2点で指定した範囲を出力する。
CURPIC	: アクティブビューポートの表示範囲を出力する。(省略時)
ALLPIC	: 図形表示領域全体を出力する。
ALL	: Advance CAD の画面全体を出力する。

INSIDE [d]+ : 指定した図形表示領域とメニュー領域を出力する。指定したメニュー領域に出力したいメニューが表示されていない場合には、メニュー領域だけを指示しておく。
このあと出力したいメニューを画面に表示させ、割込みコマンド DOC/RAS2 を実行する。

ラスターファイルの大きさを指定する。出力範囲を BOX p1 p2 または CURPIC で指示した場合に指定できる。

SIZE s1 s2 : 出力するラスターファイルの幅と高さを指定する。単位はピクセル。大きさ指定に対して出力範囲の縦横比が異なる場合、どちらか一方が指定サイズいっぱいになるように調整される。s1 s2 どちらかの値をゼロと指定すると、その値を出力範囲の縦横比から計算する。大きさが指定されると図形領域は、指定された大きさに出力するために拡大／縮小される。

ファイルに出力する。

<CE> : ファイル出力を実行する。

9.1.2 ラスターファイルを作成する

【メニュー】

なし

【構文】

DOC/RAS2

コマンド DOC/RAS で設定した出力ファイル名、出力範囲を使ってラスターファイルを作成する。このコマンドはメニュー領域も含めて出力するとき、DOC/RAS を使うと、DOC/RAS コマンドのメニューが画面に現れてしまうのを避けるために用意したコマンドである。したがってこのコマンド DOC/RAS2 はオンスクリーンメニューに登録してしまうと無意味になる。

例

- (1) シングルビューポート時に現在の図形ゾーンを出力する。またはマルチビューポート時にアクティブビューポートを出力する。
DOC/RAS filename CURPIC <CE>
- (2) マルチビューポート時に図形ゾーン全体を出力する。
DOC/RAS filename ALLPIC <CE>
- (3) カレントピクチャの指定範囲を出力する。
DOC/RAS filename BOX d1 d2 <CE>
- (4) 指定したメニュー領域を出力する
DOC/RAS filename INSIDE [d]+ <CE>
- (5) Advance CAD の画面全体を出力する
DOC/RAS filename ALL <CE>
- (6) 直線を作成するメインメニューを出力する。
DOC/RAS filename INSIDE d
このとき直線を作成するメニューは表示されていない。直線を作成するメニューを表示させる。
LBP
つぎに DOC/RAS の <CE> に相当する割込みコマンドを入力する。
DOC/RAS2

9.1.3 図面配置状態のラスターファイルを作る

【メニュー】

〔図面配置〕 → 〔ラスター出力〕

【構文】

DRAW/RAS	$\left[\begin{array}{c} \text{NXT} \\ \text{PRV} \end{array} \right]$	[FNM] filename	$\left[\begin{array}{c} \text{TMPPLN} \\ \text{ACTPLN} \\ \text{TIFF} \\ \text{TIFFG4} \\ \text{BMP} \\ \text{JPEG} \end{array} \right]$	[BOX p1 p2]	$\left[\begin{array}{c} [\text{SZIE}] \text{ s1 } [\text{s2}] \end{array} \right]$	<CE>
----------	--	----------------	---	-------------	---	------

表示する頁を指示する

- NXT : 表示中の頁よりも後ろの有効頁（図面枠が設定されている頁のこと）を表示する。後ろに有効頁が存在しなければ、先頭の有効頁を表示する。
- PRV : 表示中の頁よりも前の有効頁（図面枠が設定されている頁のこと）を表示する。前に有効頁が存在しなければ、最終の有効頁を表示する。

出力ファイルを指定する。

FNM filename: 出力するファイル名を入力する。出力ファイル名はコンフィグレーションファイル ACAD.SET ファイルの #DOC:RASTER# キーワードで指示されたディレクトリへ作成される。

出力ファイルのフォーマットを指定する。

- TMPPLN : モノクローム (1 ビット) の SUN ラスターファイルで出力する。(省略時)
- ACTPLN : カラー (8 ビット) の SUN ラスターファイルで出力する。
- TIFF : カラー (8 ビット) の TIFF ファイル (uncompress 形式) で出力する。
- TIFFG4 : モノクローム (1 ビット) の TIFF G4 形式で出力する。
- BMP : Microsoft Bitmap ファイルで出力する。
Microsoft Bitmap フォーマットには、DIB (Device Independent Bitmap), DDB の 2 種類があるが、ここで出力できるのは DIB のみである。
- JPEG : カラー (24 ビット) の ISO JPEG 標準形式で出力する。

出力範囲を指定する。

BOX p1 p2: 対角の 2 点で指定した範囲を出力する。省略すると出力範囲は図面枠の最大外形になる。

ラスターファイルの大きさを指定する。

SIZE s1 s2 : 出力するラスターファイルの幅と高さを指定する。単位はピクセル。大きさ指定に対して出力範囲の縦横比が異なる場合、どちらか一方が指定サイズいっぱいになるように調整される。s1 s2 どちらかの値をゼロと指定すると、その値を出力範囲の縦横比から計算する。大きさが指定されると図形領域は、指定された大きさに出力するために拡大／縮小される。

ファイルに出力する。

<CE> : ファイル出力を実行する。

9.1.4 Adobe Illustrator で取り込める形式の EPS ファイルの出力を行う

【メニュー】

〔ラスター背景〕 → 〔パブリス出力〕

【構文】

DOC/EPS [FNM] filename	$\left[\begin{array}{l} \text{BOX p1 p2} \\ \text{CURPIC} \end{array} \right]$	<CE>
------------------------	---	------

出力するファイル名を指定する。

FNM filename : 出力するファイル名を入力する。出力ファイル名はコンフィグレーションファイル
ACAD.SET ファイルの #DOC:EPS# キーワードで修飾される。

出力する範囲をつぎのどちらかで指示する。

BOX p1 p2 : 対角の2点で指定した範囲を出力する。

CURPIC : アクティブビューポートの表示範囲を出力する。(省略時)

ファイル出力開始

<CE> : コマンドを実行する。

補足

このコマンドは、Adobe Illustrator 5.0 以降に対応した形式の EPS ファイルとして出力します。出力図形のサイズは、Advance CAD の原寸がそのまま適用されます。

他のドキュメント出力コマンドのような、SIZE 指定による強制的な縮尺変換はしていませんのでご注意ください。

なお、BOX もしくは CURPIC による出力範囲指定と Illustrator 側に取り込んだ後の用紙サイズは EPS ファイルの段階では対応が難しいので、用紙サイズと表示範囲の調整は Illustrator 側で取り込んだ後に調整ください。

第 10 章 マクロ

10.1 マクロとは何か

マクロは Advance CAD 利用者が Advance CAD をカスタマイズするための道具のひとつです。

オプションの多いコマンドをマクロを使って使いやすくする「入門レベル」から、設計作業手順をマクロを使って自動設計的な方式を目指すレベルまで、多様なカスタマイズができます。

マクロを作るには、実行したい Advance CAD のコマンド名とそのパラメータをその順に並べてテキストファイルに記述します。これはキーボードからコマンド名を入力してコマンドを実行する代わりに、それをテキストファイルに記述しておくわけです。

Advance CAD で幅 100, 高さ 50 の四角形を作るのは次のようなコマンドになります。

```
STR/CP @X0@Y0 @X100 @Y50 @X0 @Y0 <CE>
```

これをマクロにするには、上記の内容をそのままテキストファイルに記述すればよいのです。

マクロを実行するには、マクロを実行することを示すコマンド **MACRO** を入力し、続いて実行したいマクロのファイル名を入力します。ファイルの内容に誤りがなければ、ファイルに記述したコマンドが順次実行され四角形ができます。

上記のマクロの例では、いつも決まった大きさの四角形しかできません。マクロを実行するときに四角形の大きさを変えることができるようにすれば、このマクロの用途が広がります。そのためには四角形の大きさをマクロに指示する方法が必要です。これはマクロではつぎのように記述します。

```
echo " 四角形の幅を入力 ";  
w = getnum(1);  
echo " 四角形の高さを入力 ";  
h = getnum(1);  
STR/CP @X0@Y0 @X[w] @Y[h] @X0 @Y0 <CE>
```

詳細は後述しますが、四角形の幅を変数 *w*、高さを *h* で受け取り、それを四角形の大きさとして使用すればよいのです。

このコマンドを実行すると、マクロが四角形の幅と高さを問い合わせてきますから、適切な値を入力します。

```
" 四角形の幅を入力 "      50  
" 四角形の高さを入力 "    20
```

この結果、50×20 の四角形ができます。

次のマクロは大きさの異なる 4 つの四角形を作ります。

```

STR/CP
for (j=1; j<=4; j=j+1) {
    w = 20*j;
    h = 10*j;
    @X0@Y0 @X[w] @Y[h] @X0 @Y0 <CE>
}

```

for (j=1; j<=4; j=j+1) {} は、for ループといい、{} の中を 4 回繰り返すことを意味します。{} の中で毎回四角形の幅 (w) と高さ (h) を設定していますので、20×10, 40×20, 60×30, 80×40 という大きさが違う 4 つの四角形ができます。

このように計算式を使い for ループでマクロの流れを制御することで、単純に Advance CAD のコマンド名やそのパラメータを順次並べただけでは得られない、汎用性のある高機能なマクロを作ることができます。

マクロのうち、Advance CAD のコマンドやそのパラメータを順次並べた部分を「コマンドストリーム部」と呼び、大文字で記述します。そしてコマンドストリームの実行のながれを制御する部分や計算式の部分を「言語部」と呼び、小文字で記述します。このように大文字と小文字を使い分けることで、特別な区切り記号なしで自然にコマンドストリーム部と言語部を識別できるようになっています。

上記のマクロで、大文字のところだけを拾って読むと、コマンドストリーム部となります。Advance CAD のコマンド入力と同じです。

```
STR/CP @X0@Y0 @X[w] @Y[h] @X0 @Y0 <CE>
```

小文字の部分だけを拾って読むと、つぎのようになり、通常のプログラミング言語でおなじみの内容になります。

```

for (j=1; j<=4; j=j+1) {
    w = 20*j;
    h = 10*j;
}

```

マクロの言語部は、C 言語の構文をまねて作ってあります。C 言語の知識がある人ならばより簡単にマクロが作れるようになります。

10.2 マクロファイルの書式

マクロファイルはテキストファイルです。テキストエディタを使用して作成します。
マクロファイルは、コンフィグレーションファイル ACAD.SET のキーワード **#MACRO#** で定義されているディレクトリに作成します。ファイル拡張子は **". MAC"** で、固定です。ファイル名はたとえば、**"/home/acad/macro/TEST. MAC"** になります。

マクロは以下の規約に従って記述しなければなりません。

- 1 行 80 文字以内。
- コメント
"/*" から "*/" までをコメントとみなす。コメントを入れ子にしてもよい。
- トークン (言語部の識別子、Advance CAD のコマンドやその引数のこと) の間にスペースを 1 つ以上置くこと。
- 言語部とコマンドストリーム部の識別方法

言語部	小文字で記述
コマンドストリーム部	大文字で記述
- 日本語
文字列 (" から " の間) およびコメント (/* から */ の間) に日本語の文字を使用してもよいが、それ以外の場所に用いてはならない。
- タイトル
タイトルはファイル一覧表示のときにファイル名と同時に表示する、マクロの簡単な説明文のことである。
タイトルは、マクロの先頭に次の形式でコメントとして記述する。

```
/* #TITLE# 説明文 */
```

 #TITLE# はキーワード、説明分は 1 行とし、60 文字程度とする。
たとえば下記のように記述する。

```
/* #TITLE# 同心円を作る */
```

 または、

```
/*  
#TITLE# 同心円を作る  
*/
```

10.3 言語部

10.3.1 識別子 (名前)

識別子は、英字 (a ~ z), 数字 (0 ~ 9) およびアンダースコア (_) の列で記述します。

最初の文字は英字で始めます。 ([a-z][a-z0-9_]*)

最初の 15 文字までが有効です。識別子は予約語・変数名・ラベル名・組込み関数名などを表わします。

10.3.2 予約語

以下の識別子はマクロの予約語なので、変数名・ラベル名などには使用できません。

break, case, continue, default, do, echo, else, exit, extern, for, goto, if, return, switch, while

10.3.3 変数

10.3.3.1 組み込み関数名と変数名の区別

関数名と同じ変数名を使用できます。これは今後新しい組み込み関数が増えたとき、関数名と同じ変数名を使用しているマクロが実行時エラーになるのを避けるためです。

またコンパイル時に関数名の誤りを発見することができます。

名前のあとにカッコ "(" がくれば関数名、そうでなければ変数名とみなします。したがって引数なしの関数呼び出しでも必ず ")" を付けるようにしてください。

たとえば下記のように記述します。左辺の fname は変数名、右辺の fname() は関数呼び出しです。

```
clear();  
if(error()!=0) { return; }  
fname = fname();
```

10.3.3.2 内部変数と外部変数

内部変数とは、マクロ内での変数です。内部変数は、そのマクロ内だけで有効で、そのマクロを終了すると消去されます。たとえばマクロ A で変数 a, b を生成し、その中でマクロ B を呼出したとします。このマクロ B ではマクロ A の変数 a, b は参照できません。マクロ B で変数 a, b を生成した場合、これはマクロ A の変数とは別のものとなります。マクロ間で変数を共有したいときは外部変数を使います。

外部変数とはマクロの外部に保存されている変数で、マクロの実行が終了しても保持されます。Advance CAD を停止したときは外部変数テーブルは消滅します。マクロの中では extern 文で外部変数の宣言をすることにより、外部変数の参照と変更ができます。計算器 ([式] または { 式 }) や、APG パラメータは外部変数だけを参照あるいは変更でき、内部変数は参照できません。APG パラメータ名は 6 文字以内の制限があります。

10.3.3.3 外部変数の宣言

【書式】

```
extern 外部変数名, ..., 外部変数名 ;
```

説明

extern 文は外部変数を参照するために使います。指定した外部変数がまだ存在していなければ、自動的に追加します。

extern 文は外部変数を参照する前であればソースコード上のどこに記述してもかまいませんが、ソースコードの先頭に置くことを勧めます。組み込み定数は **extern** 宣言なしで直接参照します。

10.3.4 データの型および定数

マクロで扱うデータの型には以下の種類があります。

数値型
 文字列型
 座標型
 デジタイズ点型
 テンポラリポイント型
 スクリーン座標型
 カーソル位置型
 アイテム識別子型
 コマンドエンド型
 バックスペースキー型
 スペースキー型

10.3.4.1 数値型

数値定数は次の形式のいずれでもかまわない。

整数形式 2, -2 など
 実数形式 -2.718, .025 など
 指数形式 2.5E-5, -2E3 など

すべて倍精度実数として扱う。また論理式では数値 0 は論理値偽、それ以外の数値は論理値真として処理する。

10.3.4.2 文字列型

文字列定数はクォーテーション (") で囲んで記述する。

マクロソースコードは 80 文字以内なので、長い文字定数を記述する場合は継続行を使う。文字定数が次の行に続くことを示すには、行の終わりをバックスラッシュ \、続いてニューラインとする。次の行の最初のバックスラッシュまでは読み飛ばし、次の文字からが有効。これは 1 つの文字定数を継続行にしたとき、字下げ (インデント) をそろえることができるようにするため。

たとえば、変数 a に "C ITOH TECHNO-SOLUTIONS CO., LTD. Advance CAD" という文字列を設定するには、下記のようにする。

```
a = "C ITOH TECHNO-SOLUTIONS CO., LTD. ¥
¥Advance CAD";
```

- 文字列の長さ

文字列の長さは `length` 関数で調べることができる。

```
a = "String";
n = length(a);
```

文字列の最長は 2048 文字。長い文字列は後部が切断される。文字列を処理する演算子・関数は、すべてこの制限を守る。たとえば + 演算子で 2000 文字の文字列と 2000 文字の文字列を連結すれば、先頭 2048 文字だけが取られる。

- 部分文字列

```
t[l:h]
```

文字列 `t` の `l` 文字目から `h` 文字目までの文字列を意味する。`l` と `h` の範囲は $0 \leq l \leq h < \text{length}(t)$ である。

`l` と `h` が等しいときに `t[l]` と記述し、1 文字だけの参照となる。

```
a[0:2] → "Str"
a[0]   → "S"
a[5]   → "g"
```

- エスケープ文字

文字列にクォーテーション文字やキャリジリターンなどの特別な文字を含めるには、バックスラッシュを使った特別な表記を使用する。

¥" クォーテーション文字自身

¥¥ バックスラッシュ文字自身

¥r キャリジリターン（ジェネラルノート文字列内では改行の意味になる）

¥n ニューライン。ニューラインはファイル入出力関数で必要になる。UNIX ではテキストファイルの 1 行の終わりはニューライン（¥012）文字。

ファイルに文字列を書き込むとき、最後にニューラインを付けないと、すべての行が 1 つにくっついてしまう。¥r はキャリジリターン（復帰、¥015）文字を表す。これは Advance CAD のジェネラルノートではニューラインを意味するが、UNIX のテキストファイルでは通常の文字であり、ニューラインの意味にはならないことに注意すること。

¥a アラート。ベルを鳴らす文字（¥007）。エラーメッセージを出力する場合に使用する。

¥8 進数 非印字文字を文字列に挿入するために使用する。非印字文字はジェネラルノートで使用する文字に含めてはならない。またナル文字（¥000）は使用してはいけない。

¥M ジェネラルノートで使用するメタ文字を表す。¥M はキーボードのメタキー（ALT キー） + "z" を押すのと同じ役割を果たす。

ジェネラルノートで使用するメタ文字は以下のとおり。

メタ文字		¥M 表記
°	度	¥M0
φ	径	¥M1
±	プラス / マイナス	¥M2
□	角	¥M3
⌒	弧	¥M4
F	分数	¥MF
S	2 段文字	¥MS
M	マーク	¥MM
'	区切り	¥MY
;	終了	¥MZ

たとえば下記のように使用する。

```
/* Drafting symbols */
GNT "¥M0 Degree" <0, 90> <CE>
    "¥M1 Phai" <0, 80> <CE>
    "¥M2 Plus-Minus" <0, 70> <CE>
    "¥M3 Square" <0, 60> <CE>
    "¥M4 Arc Length" <0, 50> <CE>
/* Dual stack text */
GNT "ABC¥MSTOP¥MYBOTTOM¥MZXYZ" <0, 30> <CE>
GNT "ABCM¥MS2¥MZXYZ" <0, 20> <CE>
GNT "ABCH¥MS¥MY2¥MZ0XYZ" <0, 10> <CE>
/* Mark in text */
GNT "TOLERANCE¥MM111, 112, 113, 114, 115¥MZXYZ" <0, 0> <CE>
```

° Degree

φ Phai

± Plus-Minus

□ Square

⌒ Arc Length

ABC^{TOP}_{BOTTOM}XYZ

ABCM²XYZ

ABCH₂0XYZ

TOLERANCE—□○□⌒XYZ

10.3.4.3 座標型、デジタイズ点型、テンポラリポイント型

Advance CAD のモデル座標での点座標を持つ。デジタイズ点型はこの点が座標入力ではなく、ロケータデバイスによってデジタイズされた入力であることを意味する。テンポラリポイント型はテンポラリポイントとして `gettpt` または `tpt` 関数で作成されたことを意味する。

この型のデータは x 座標と y 座標の 2 つの要素を持つ。各々の要素は添字を使って参照する。

p[0] → モデル座標系 x 座標

p[1] → モデル座標系 y 座標

10.3.4.4 スクリーン座標型

グラフィックススクリーンのスクリーン座標を持つ。Advance CAD ウィンドウの左下隅が原点になる。

この型のデータは x 座標と y 座標の 2 つの要素を持つ。各々の要素は添字を使って参照する。

p[0] → スクリーン座標系 x 座標

p[1] → スクリーン座標系 y 座標

10.3.4.5 カーソル位置型

Advance CAD のウインドウの位置を持つ。主としてマクロのテンポラリウインドウで使用する。この型のデータはウインドウの列番号 (x)、行番号 (y) とウインドウの番号の 3 つの要素を持つ。各々の要素は添字 を使って参照する。

```
p[0] → 列番号 (x)
p[1] → 行番号 (y)
p[2] → ウインドウ番号 (w)
      w = 33 テンポラリウインドウ
      w = 6  メッセージウインドウ
```

10.3.4.6 アイテム識別子型

Advance CAD のアイテム識別子を持つ。アイテムの識別子は 1 ～ 262143 の整数である。

10.3.4.7 コマンドエンド型、バックスペースキー型、スペースキー型

キーボードからキャリジリターン、バックスペースキーまたはスペースキーだけを入力した時にできるデータを表す型。

数値型、文字型は標準的なデータ型だが、それ以外は Advance CAD 固有のデータ型である。Advance CAD 固有のデータは、以下の表にあるような生成関数、または会話型入力関数を使うと生成でき、変数に代入できる。

	要素数	生成関数	会話型入力関数
数値型	1	不要	getnum
文字列型	0 ～ 250	不要	gettxt
座標型	2	pce	getpnt
デジタイズ点型	2	pdg	
テンポラリポイント型	2	tpt	gettpt
スクリーン座標型	2	dig	getdig
カーソル位置型	3	pos	getpos
アイテム識別子型	1	idptr	なし
コマンドエンド型	1	なし	get*
バックスペースキー型	1	なし	get*
スペースキー型	1	なし	get*

マクロでは変数の型宣言はないので、変数のデータ型は最後に代入されたデータの型を持つことになる。

```
a = "String";    (変数 a は文字型)
b = sin(30);     (変数 b は数値型)
a = 100*b;       (変数 a は数値型に変わった)
```

ある変数がどの型のデータを持つかを調べるには、vtype 関数を使う。また、その型のデータの要素数を調べるには、length 関数を使う。

10.3.5 組み込み定数

有用な定数は名前を付けて登録してあります。これらの値は変更できません。

pi	円周率 π (3. 14159...)
exp	ネイピア定数 e (2. 71828...)
true	真 (1)
false	偽 (0)

10.3.6 配列

配列はプログラミング言語においてきわめて重要です。

配列がないと 10 個の数値を保持するのに 10 個の変数が必要になります。さらにそれらの数値を 10 倍するには 10 個の乗算式を書かなければなりません。配列があれば変数は 1 つで、添字を使ってループ処理が可能です。プログラムが簡素になり、変更も簡単です。

大きなサイズの配列を多く使うと、メモリを多く使うため、スワップファイルを使い果たしてしまうことがあります。このような場合はスワップファイルの大きさを増やさなければなりません。マクロ関数 `clear()` で不要な変数を消去すれば回復します。

尚、`array` 関数の要素数定義が上限を超えている場合は、その時点でエラーメッセージを表示してマクロの実行を終了します。

10.3.6.1 一次元配列

配列は `array` 関数で生成する。

```
a = array(size);
```

`size` は 1 ~ 163840。ただし長さ 1 の配列は効率が悪いので単純変数の方がよい。

`array` 関数で配列を生成したとき、配列の全要素は数値型で値は 0.0 に初期化される。

配列要素は添字を使って参照する。

```
a[j]
```

添字の範囲は $j=0 \sim \text{size}-1$ 。

配列の各要素には数値・文字列・座標・デジタイズ点など、配列以外の任意の型のデータを代入できる。配列要素を配列にすることはできない。

```
a[0]="Advance CAD";
```

```
a[1]=6.0;
```

```
a[2]="Dec. 1991"
```

配列要素 `a[0]` は文字列 "Advance CAD" を持っている。`a[0][0]` は文字 "A"、`a[0][1]` は文字 "d" を参照する。

配列の長さは `length` 関数で調べることができる。`length(a)` は配列 `a` の長さ 3 を返す。`length(a[0])` は配列の最初の要素の長さ、つまり文字列 "Advance CAD" の文字数 11 を返す。

配列の代入は、配列全体をコピーする。

```
b=a;
```

この時点では変数 `a` と `b` はまったく同じ内容を保持している。しかしデータを共有しているのではない。

```
b[1]=100.0;
```

`b[1]` は数値 100.0 だが、`a[1]` は 6.0 のままである。

部分配列
`a[l:h]`

配列 `a` の `l` 番目から `h` 番目までの要素からなる配列を意味する。

`l` と `h` の範囲は、 $0 \leq l \leq h < \text{length}(a)$ 。

```
a = array(6);  
b = a[2:4];
```

配列 `b` は長さ 3 となる。配列 `b` は配列 `a` の一部をコピーしており、共有してはいない。`b[0]=10.0`;
と変更しても、`a[2]` は 0.0 のままである。

10.3.6.2 二次元配列

```
a = array(n,m)
```

`n` および `m` の範囲は、 $1 \leq n, m, n*m \leq 163840$ 。

配列要素は 2 つの添字を使って参照する。添字の範囲は、 $i = 0 \sim n-1$,

```
j = 0 ~ m-1。  
a[i][j]
```

配列要素の順序

`a = array(4,3)` としたとき、配列 `a` の要素の順番はつぎのようになる。

```
a[0][0], a[0][1], a[0][2], a[1][0], a[1][1], a[1][2], a[2][0], a[2][1],  
a[2][2], a[3][0], a[3][1], a[3][2]
```

配列の長さは `length` 関数で調べることができる。

`a = array(4,3)` としたとき、`length(a)` は配列 `a` の長さ 4 を返す。配列 `a` は長さ 3 の一次元配列を 4 つ分持つという意味である。

`length(a[0])` は、配列 `a[0]` の長さ 3 を返す。`a` は二次元配列なので、`length(a[1])`,
`length(a[2])` も、同じく長さ 3 を返す。

部分配列

書式	範囲
<code>a[l:h]</code>	$0 \leq l \leq h < \text{length}(a)$
<code>a[i][l:h]</code>	$0 \leq i < \text{length}(a), 0 \leq l \leq h < \text{length}(a[i])$

`a = array(4,3)` のとき、つぎのように `b` を指定したとする。

`b = a[1:2];` とすると、`b` は (2,3) の二次元配列になる。このとき `length(b)` は 2、`length(b[0])` は 3 になる。

`b = a[0]` とすると、`b` は長さ 3 の一次元配列になる。

`b = a[3][0:1]` とすると、長さ 2 の一次元配列になる。

部分配列への代入の注意

左側の部分配列の長さと右側の配列の長さが異なる場合は、左辺と右辺の要素を 1 対 1 対応で代入していき、どちらかの配列の終了に達すると代入を終了する。

たとえば次のような 2 つの配列 `a, b` で考えてみる。

```
a = array(4, 3);
```

```
b = array(10);
```

右辺の配列の方が、左辺の配列よりも長い場合の例を考える。

```
a[0] = b;
```

左辺 a[0] は長さ 3、右辺 b は長さ 10 である。a[0][0] = b[0], a[0][1] = b[1], a[0][2] = b[2] の 3 要素の代入が行なわれる。

反対に、左辺の方が長い場合を考える。

```
a[0] = b[6:7];
```

左辺 a[0] は長さ 3、右辺 b[6:7] は長さ 2 である。a[0][0]=b[6], a[0][1]=b[7] の 2 要素の代入が行なわれる。a[0][2] は変更されない。

```
a[0] = 10;
```

a[0][0] = 10 の代入が行なわれる。a[0][1], a[0][2] は変更されない。

a = b は、b を a に代入することであり、a は作り直されるため b と同じ長さ 10 の配列となる。これは部分配列への代入ではない。

点列を配列とする場合は、二次元配列ではなく、一次元配列とし、その各要素を座標型とする方が効率的である。

```
points = array(4);
points[0] = pce(10, 10);
points[1] = pce(50, 10);
points[2] = pce(50, 40);
points[3] = pce(10, 40);
```

配列としては一次元だが、その要素の座標型自身が 2 つの要素 (x, y) を持つため、全体としては二次元配列と同じ効果がある。

このとき、points[1][0] は第 2 点の X 座標、points[1][1] は第 2 点の Y 座標を参照する。

10.3.7 演算子

演算子の優先度は以下の通りです。優先度が高いものから低いものの順で、1 行にあるものは同一の優先度です。

```
優先順位の指示  '(' および ')'
単項 '+' , 単項 '-' , 論理否定 (NOT) '!'
べき乗 '**'
乗算 '*', 除算 '/', 法 '%'
加算 '+', 減算 '-'
関係演算子 小さい '<', 小さいか等しい '<=', 大きい等しい '>=', 大きい '>'
等値演算子 等しい '==', 等しくない '!='
論理積 (AND) '&&'
論理和 (OR) '||'
条件演算子 '?' および ':'
代入演算子 '=', '+=', '-=', '*=', '/=', '%='
カンマ演算子 ','
```

同一優先度の演算子が続くときは、以下の順序になる。

- べき乗は右結合的演算子なので、べき乗が続くときは、最も右から順に計算する。
 $2^{**}2^{**}3$ は、 $((2^{**}2)^{**}3) = 4^{**}3 = 64$ ではなく、
 $(2^{**}(2^{**}3)) = 2^{**}8 = 256$ となる。
- それ以外の 2 項演算子はすべて左結合的演算子である。同じ優先度の演算子が続くときは、最も左から順に処理する。
 $2/4*2$ は、 $((2/4)*2) = 1$ であり、 $(2/(4*2)) = 0.25$ ではない。

10.3.7.1 単項演算子 '+' 絶対値

$a=-10$ のとき $+a$ は絶対値 10 となる。

10.3.7.2 単項演算子 '-' 符号反転

$a=-10$ のとき $-a$ は 10、 $a=10$ のとき $-a$ は -10 となる。

10.3.7.3 単項演算子が続くときはもっとも右から順に処理する。

$a=-10$ のとき $-+a = -(+a) = -10$

$a=10$ のとき $-+a = -(+a) = -10$

10.3.7.4 法演算子

$n\%m$ は、 n を m で割った残りを計算するのに使う。たとえば、 $m=2$ のとき、 $n\%m$ は 0 または 1 になる。

$n = 1 \quad 1\%2 \rightarrow 1$

$n = 2 \quad 2\%2 \rightarrow 0$

$n = 3 \quad 3\%2 \rightarrow 1$

$n = 4 \quad 4\%2 \rightarrow 0$

したがって、 $n\%2$ が 1 のときは奇数、0 のときは偶数と判定できる。

10.3.7.5 関係演算子、等値演算子

2 つの被演算子が文字列型のときは、文字列の比較を行う。

`"ab" < "ef"` (真)

`"ab" != "ab"` (偽)

`"AB" != "ab"` (真)

文字列と数値を混合することはできない。

`5 < "10"` (誤)

10.3.7.6 加算 '+' 演算子

+ 演算子の 2 つの被演算子が数値型のときは算術加算を行う。

+ 演算子の 2 つの被演算子が文字列型のときは文字列を連結する。

`a="AZ-"; b=a+"002"; c=a+"004";`

`b` は "AZ-002", `c` は "AZ-004" となる。

文字列と数値を混合することはできない。

10.3.7.7 条件演算子 $a ? b : c$

a が真のとき b , そうでないときは c を返す。

`n=a>b?a:b`

n は a が b より大きければ a , そうでなければ b となる。 a と b の大きい方を取り出す。

10.3.7.8 代入演算子 '+=' , '-=' , '*=' , '/=' , '%='

a op= b 形式の式の意味は、a=a op (b) である。ここで op は +, -, %, *, または / 演算子を表す。
 a+=10 は、a=a+10 となる。
 a/=b*c は、a=a/(b*c) となる。

10.3.7.9 複代入文

```
a=b=c=10;
これは以下と同等である。
c=10; b=c; a=b;
```

10.3.7.10 座標型・デジタイズ点型の演算子

点型のデータには、次のような算術演算子を使用できる。

```
p : 点型のデータ
s : 数値型データ

和      p=p+p
差      p=p-p
ベクトル内積 s=p*p
スカラ乗 p=s*p または p=p*s
スカラ除 p=p/s
ベクトル長 s=+p
逆向き  p=-p
```

10.3.7.11 カンマ演算子

カンマで区切られた式を左から右に演算する。
 a=10, b=20; は、a=10; b=20; と同じ結果になる。

カンマ演算子は for 文の初期設定式、再設定式で、多く使用される。

```
for (j=0, n=1; j<8; j+=1, n*=2) {
  echo "2**" j "=" n;
}
```

10.3.8 式と文

式とは、演算子を使ったつぎのようなものです。

```
n=10    ( 代入の式 )
n==10   ( 比較の式 )
10*n    ( 算術式 )
```

またつぎのような演算子のないものも式とみなします。

```
"abc"   ( 定数 )
n        ( 変数 )
sqrt(2)  ( 組み込み関数 )
```

式のうしろに文の終了を示すセミコロン (;) を置くと文になります。通常の文は代入文になります。

```
n=10;
x=cos(30);
y=sin(30);
```

代入文でない文としては組み込み関数を呼び出す場合があります。

`sqrt(2);` は 2 の平方根を計算しますが、結果は失われてしまうので意味がありません。
 しかし、`clear();` はマクロの変数テーブルを再初期化する関数で意味があります。
 文を 1 行に複数並べてもかまいません。
`n=0; m=20; s="String";`

後述のフロー制御文では `{ }` でくくった「ブロック」と呼ばれるものが出てきます。
`{ }` の中には 1 つ以上の文を置きます。ブロックは複数の文をひとまとめにするため、複文とも呼ばれます。`{ }` をあたかも 1 つの文として扱いますが、ブロックの終わりの `}` の次には文の終わりを示すセミコロンはつけません。

10.3.9 フロー制御文

10.3.9.1 while 文

```
while (条件式) {
    条件式が真のとき実行する文
}
```

まず条件式を計算する。結果が真であれば `{ }` 内に置かれた式を実行する。これを繰り返す。
 繰り返しは条件式の結果が偽になるまで続く。最初に条件式が偽であれば `{ }` 内は一度も実行されない。
 つぎのように条件式を常に真にすれば、無限ループとなる。無限ループを脱出するには `break` 文や `return` 文を使わなければならない。

```
while (true) {
    文
}
```

組み込み定数 `true` は真を表す数値 1 が設定されている。

10.3.9.2 do 文

```
do {
    文
} while (条件式);
```

まず `{ }` 内の文を実行する。次に条件式が真であれば、繰り返し実行する。条件式が偽になると終了する。`while` 文が前判定なのに対し `do` 文は後判定で、文は必ず 1 回は実行される。

10.3.9.3 for 文

```
for (初期設定式; 条件式; 再設定式) {
    条件式が真のとき実行する文
}
```

最初に初期設定式を行う。条件式が真であれば文を実行する。文を実行したのち再設定式を実行し、条件式が真であれば文を実行する。以下条件式が偽になるまで繰り返し実行する。

`for` 文はつぎの `while` 文に等しい。

```
初期設定式;
while (条件式) {
    条件が真のとき実行する文
    再設定式;
}
```

初期設定式、条件式、再設定式のいずれを省略してもよいが、区切りであるセミコロン ; は省略できない。条件式 2 を省略すると常に真となり、無限ループとなる。

```
for (;;) { /* 無限ループ */
    文
}
```

10.3.9.4 条件文

以下の 3 つの記述方法がある。

1. if (条件式) {
 条件式が真のとき実行する文
 }
2. if (条件式) {
 条件式が真のとき実行する文
 } else {
 条件式が偽のとき実行する文
 }
3. if (条件式 1) {
 条件式 1 が真のとき実行する文
 } else if (条件式 2) {
 条件式 1 が偽で
 条件式 2 が真のとき実行する文
 } else {
 条件式 1, 2 共に偽のとき実行する文
 }

10.3.9.5 switch 文

```
switch ( 式 ) {
case 定数 1:     文 1
                break;
case 定数 2:     文 2
                break;
default:        文 3
}
```

式の値が「定数 1」と等しいとき、「case 定数 1:」以下の「文 1」を実行する。「文 1」の終わりには break 文が必要である。break 文がないと、そのままつぎの「文 2」を実行する。break 文は switch 文から抜ける働きをする。

式の値が「定数 2」と等しいとき、「case 定数 2:」以下の「文 2」を実行する。

いずれでもないときは、default の「文 3」を実行する。「default:」は 1 つだけ記述できる。

式の値がどの case 定数とも一致せずしかも「default」がなければ、switch 中のどの文も実行されない。

定数は文字定数または数値定数でなければならない。

10.3.9.6 continue 文

```
continue;
```

while, do, for 文のブロック内で用い、そこからそのブロックの最後の文までを飛ばす (実行しない)。そして次の繰り返しを行う。ループを終了するのではない。

つぎのマクロは配列 a の中の正の要素のみを処理する。

```
for (i=0; i<n; i+=1) {
    if (a[i]<=0) { continue; }
    /* 正の値の要素の処理 */
}
```

10.3.9.7 break 文

break;

while, do, for, switch 文のブロック内で用い、その while, do, for, switch 文の終わりのつぎの文に移る。

break 文は continue 文とは違ってブロックから抜け出す。

次のマクロは配列 a の中の負の要素が現れたら for ループを終了する。

```
for ( i=0; i<n; i+=1) {
    if (a[i]<0) { break; }
}
```

10.3.9.8 goto 文、ラベル

goto 識別子 ;

識別子で示したラベルの次の文まで飛ぶ。

識別子 :

ラベルの設定。識別子のあとにコロン ':' を置く。

goto 文を多用するとマクロの流れを追いににくくなる。実際に goto を使わないでマクロを書く方がたいていの場合簡単である。goto の一般的な使い方は、入れ子になったループのいちばん内側からループ全体を抜け出す場合である。

```
for (...) {
    for (...) {
        if (...) { goto err; }
    }
}

err:
```

10.3.9.9 return 文

return;

現在実行中のマクロを終了し、そのマクロを呼出したマクロに戻る。

10.3.9.10 exit 文

exit;

マクロの実行を終了する。

10.3.9.11 echo 文

echo “文字列” または変数；

文字列はそのまま、変数はその値を表示する。いくつ並べてもかまわない。以下のデータ型は複数の要素を持つので、次のような形式で表示する。

座標型	(x, y)
デジタイズ点型	(x, y)
スクリーン座標型	(x, y)
カーソル位置型	(x, y, w)

たとえば "Point=(10,20)" と表示するには、下記のように記述する。

```
p=pce("@X", 10, "@Y", 20);
echo "Point=" p;
```

文字列中にベル文字を含めると、ベルが鳴る。ベル文字は "\a" と記述する。例えば下記のように記述する。

```
echo "%a"; /* ベルだけ鳴らす */
echo "Illegal input %a"; /* メッセージを表示してベルを鳴らす */
```

echo 文のメッセージはプロンプトウインドウの一番上段に表示する。echo 文のメッセージは入力待ちとなったときに一度だけ表示される。

```
for (;;) {
    echo "ベリファイするアイテムをピック" ;
    error();
    VER [getpnt(1)]
    if (error()==0) { break; }
    echo "%a ピックができません";
}
```

上記の例では、メッセージエリアに "ベリファイするアイテムをピック" と表示し、入力待ちになる。アイテムが正しくピックされ、VER コマンドが成功すれば、マクロを終了する。アイテムをピックできないとベルになり、プロンプトエリアに "ピックができません" と表示する。そしてもう一度同じことを繰り返す。

echo 文の引数並びに配列の要素や式を記述する場合はカッコで囲む。

```
a=array(3);
a[0]="abc";
a[1]=10;
a[2]=-100;
echo "a[0]=" (a[0]) ", 10*a[1]=" (10*a[1]) ", +a[2]=" (+a[2]);
```

10.3.10 組み込み関数

マクロで利用できる各種関数を説明します。ある関数は引数の数が可変です。たとえば max 関数では引数が 2 個以上必要です。このような場合、引数の上限は 32 個です。

10.3.10.1 算術関数

t = sin(s)	sine 関数。s は度。
t = cos(s)	cosine 関数。s は度。
t = tan(s)	tangent 関数。s は度。
s = asin(t)	arc sine 関数。 $ t \leq 1$ 。結果は度。
s = acos(t)	arc cosine 関数。 $ t \leq 1$ 。結果は度。
s = atan(t)	arc tangent 関数。結果は $-90^\circ < s < 90^\circ$
s = atan2(y, x)	arc tangent 関数。結果は $-180^\circ \leq s \leq 180^\circ$ y=0, x=0 は 0° を返す。

<code>s = sqrt(t)</code>	平方根。 $t \geq 0$
<code>s = int(t)</code>	整数化。(切り捨て)
<code>s = nint(t)</code>	整数化。(近い整数)
<code>s = log(t)</code>	自然対数。 $t > 0$
<code>s = log10(t)</code>	常用対数。 $t > 0$
<code>s = max(s1, ..., sn)</code>	最大値を得る。 $n \geq 1$ 。引数 <code>sn</code> が配列の場合、その全要素を対象とする。すべての配列要素のデータは数値型かアイテム識別子型でなければならない。
<code>s = min(s1, ..., sn)</code>	最小値を得る。 $n \geq 1$ 。引数 <code>sn</code> が配列の場合、その全要素を対象とする。すべての配列要素のデータは数値型かアイテム識別子型でなければならない。
<code>t = dms(d, m, s)</code>	度分秒を度に変換。
<code>s = inch(t)</code>	インチをミリメートルに変換。
<code>p = rot(p, angle)</code>	点を回転させる。 <code>p</code> は座標型またはデジタイズ点型。 <code>angle</code> は回転角度。たとえばつぎのようにする。 $p = \text{rot}(p / (+p), 90);$ $(+p)$ はベクトル <code>p</code> の長さ、つまり原点から点 <code>p</code> までの距離を計算する。 <code>()</code> は必要ないが、わかりやすいようにつけてある。 <code>p / (+p)</code> はベクトル <code>p</code> をその長さで割って長さ 1 のベクトル (正規化されたベクトル) を計算する。 <code>rot</code> 関数は正規化されたベクトルを 90° 回転する。これにより座標原点から点 <code>p</code> への直線と直交なベクトルが求まる。
<code>t = sinh(s)</code>	双曲線関数 <code>sine</code> , $ s < 709$
<code>t = cosh(s)</code>	双曲線関数 <code>cosine</code> , $ s < 709$
<code>t = tanh(s)</code>	双曲線関数 <code>tangent</code>
<code>t = asinh(s)</code>	双曲線関数 <code>arc sine</code>
<code>t = acosh(s)</code>	双曲線関数 <code>arc cosine</code> , $s \geq 1$
<code>t = atanh(s)</code>	双曲線関数 <code>arc tangent</code> , $ s < 1$

10.3.10.2 文字列

<code>t = upper(s)</code>	文字列 (<code>s</code>) の中のすべての小文字を大文字に変換する。英字 'a' ~ 'z' が 'A' ~ 'Z' に変わる。それ以外の文字には作用しない。
<code>t = lower(s)</code>	文字列 (<code>s</code>) の中のすべての大文字を小文字に変換する。英字 'A' ~ 'Z' が 'a' ~ 'z' に変わる。それ以外の文字には作用しない。
<code>n = strstr(s1, s2)</code>	文字列 <code>s1</code> の中で最初に現れた文字列 <code>s2</code> の位置を返す。 <code>n</code> は $0 \leq n < \text{length}(s1)$ 。文字列 <code>s2</code> が文字列 <code>s1</code> の中になくときは、-1 を返す。
<code>t = sprintf(format, arg1, ..., argn)</code>	<code>arg1, ..., argn</code> を <code>format</code> に従って文字列に変換する。 <code>format</code> は書式指定文字列で、下記の形式の変換指定を含めることができる。 マクロの数値は常に倍精度実数なので、 <code>long</code> 指定なし。
変換指定	<code>%[flags][size][.prec]type</code>
type	<code>d</code> 整数 <code>f</code> 実数 <code>[-]ddd.ddd</code> の形式 <code>e, E</code> 実数 <code>[-]d.dddde+dd</code> または <code>[-]d.dddE+dd</code> <code>g, G</code> 実数値が大きすぎも小さすぎもしないとき <code>f</code> 変換、そうでなければ <code>e</code> または <code>E</code> 変換が用いられる。どちらの形式が選ばれるかは、どちらが短くなるかによって決まる。 以上の <code>d, f, e, E, g, G</code> には数値型変数またはアイテム識別子型変数を当てる。 <code>s</code> 文字列文字列型変数を当てる。 <code>c</code> 1 文字 文字列型または数値型変数を当てる。 整数 (0 ~ 255) はそれをコードとする 1 文字 (バイト) に変換される。 <code>%c</code> 変換子は、文字として 0 ~ 255 の値を設定できるが、コード 0 ~ 31, 127, 128 ~ 159, 255 は ASCII

	文字、EUC 漢字割り当て外なので、むやみに使用しないこと。特に 0 は文字列の終わりを示す特別な文字なので、使用しないこと。
flags	% "%%" 文字を 1 文字出力する。 + 符号 (+ または -) を必ず付ける。 空白 負符号 (-) か空白を必ず付ける。 省略時は負数のみ符号が付く。 - 左ヅメで出力する。 省略時は右ヅメで出力する。
size	最小フィールド長を指定する。
.prec	実数の小数桁数 (f, e, E) 実数の出力桁数 (g, G)

この関数の出力は文字列。エラーのときはコマンドエンド型データを返す。
以下に式の例とその結果を示す。

式	結果
x=sprintf("%d", 10)	x="10"
x=sprintf("% d", 10)	x=" 10"
x=sprintf("%+d", 10)	x="+10"
x=sprintf("%10d", 10)	x=" 10"
x=sprintf("%-10d", 10)	x="10 "
x=sprintf("%+10d", 10)	x=" +10"
x=sprintf("%-+10d", 10)	x="+10 "
x=sprintf("%f", 10)	x="10.000000"
x=sprintf("%10.2f", 12.34567)	x=" 12.35"
x=sprintf("%10.2e", 12.34567)	x=" 1.23e+01"
x=sprintf("%10.3g", 12.34567)	x=" 12.346"
x=sprintf("%.3g", -12.34567)	x="-12.346"
x=sprintf("%%a%z")	x="%az"
x=sprintf("%s", "abc")	x="abc"
x=sprintf("%10s", "abc")	x=" abc"
x=sprintf("%-10s", "abc")	x="abc "
x=sprintf("%3s", "abcdefghijk")	x="abcde fghijk"

変数 x には日本語の " ア " を入れる式は下記のようなになる。
x = sprintf("%c%c", 165, 162);
" ア " は JIS 5 区 2 点で、EUC コードの内部表現としては第 1 バイトが 165, 第 2 バイトが 162 の値。

l = sscanf(t, format)
文字列 t を文字列 format に従って変換する。
format は制御文字列。下記の形式の変換指定を含めることができる。マクロの数値は常に倍精度実数なので、long 指定なし。
同様にマクロには文字型はなく、文字列型だけなので、変換子 c には文字列型データを対応させる。

変換指定 %[*][size]type

type
d 整数
e, f, g 実数
s 文字列
c 1 文字
% % 文字

* 変換するが、値は返さない。読み飛ばす時に使う。
size 最長フィールド長を指定する。

制御文字列中の変換指定以外の文字は、つぎのような処理になる。

- ・ 制御文字列中の空白文字は、複数空白文字を読み捨てる。
- ・ 空白と変換指定以外の文字は、入力文字列 t からの文字と一致していなければならない。

この関数の出力は変換指定子によるが、数値または文字列あるいはそれらを含む配列である。途中で変換が失敗したら変換が成功した分だけ返す。最初の変換子でいきなり失敗した場合はコマンドエンド型データを返す。以下に式の例とその結果を示す。

式	結果
x=sscan(" -1234 ", "%d")	x=-1234
x=sscan("123456789012", "%5d %5d")	x[0]=12345 x[1]=67890
x=sscan(" 123", "%f")	x=123
x=sscan("-123.45", "%g")	x=-123.45
x=sscan(" 123.5 1.6e-6 ", "%f%g")	x[0]=123.5 x[1]=1.6e-6
x=sscan(" 123.456789012345", "%6f %d")	x[0]=123.45 x[1]=6789012345
x=sscan("abc defg", "%s")	x="abc"
x=sscan(" abcd efg ", "%s")	x="abcd"
x=sscan(" abcd efg ", "%s%s")	x[0]="abcd" x[1]="efg"
x=sscan(" abcd=efg ", " abcd=%s")	x="efg"
x=sscan("abcdefghi jklm", "%10s %10s")	x[0]="abcdefghi j" x[1]="klm"
x=sscan("abc", "%c%c")	x[0]="a" x[1]="b"
x=sscan("abcdefg", "%*c%c%*c%c%*c%c")	x[0]="b" x[1]="d" x[2]="f"
x=sscan("123 235", "%*d%d")	x=235

t = encode(format, arg1, ..., argn)
文字列へ変換する。
sprintf ができる前からあったもので、マクロの互換性を保つために残してある。

format : 書式指定文字列。変換指定子は % で始まる。

%s : 文字列。
%f : 数値の変換。小数以下 6 桁で表示。
%g : 実数表示。または文字列。小数部のトレイリングゼロは除去。
%d : 整数表示。
% のつぎが上記以外するとき、% は捨てられる。したがって、%% は % 1 文字となる。
arg1, ... argn : 変換指定に従って変換される値。

以下は使用方法の例。t は "Angle = 45.000000" になる。
a = atan(1);
t = encode("Angle=%f", a);

この関数は文字列を連結するのにも使用できる。ファイル名の前にディレクトリ名、うしろに拡張子をつける場合はつぎのようになる。
s = "EXAMPLE";
t = encode ("/usr/files/%s.MDL", s);
t は "/usr/files/EXAMPLE.MDL" となる。

n = subs(s, rex [, rep] [, g])
文字列 s の中の rex に合致する部分文字列を、rep の文字列で置換 (substitute) する。

引数は以下のとおり。
s : 処理対象文字列を含む変数または配列要素。この関数は変数 s の内容を変更するので、文字定数や部分文字列ではない。
rex : 置換する部分文字列を指定する。255 文字が上限文字数。
rep : 合致した部分文字列を置換する文字列。省略すると合致した部分文字列を削除する。
g : 何番目の合致する部分文字列を処理するかを指示する。1 は最初に現れた合致文字列だけを置換し、その後に合致文字列が現れても処理しない。
同様に 2 は 2 番目の合致文字列だけを処理する。また -1 のような負の値は、最後の合致文字列、-2 は後ろから数えて 2 番目の合致文字列だけを処理する。0 または省略時は全ての合致文字列を置換する。

出力は以下のとおり。
n : この関数は置換した回数を返す。合致文字列が見つからなければ 0 を返す。

rex には UNIX ed コマンド と同じ正規表現を使用できる。以下の文字は正規表現に使用される文字で、特別な意味を持っている。

文字	機能
.	ドット。任意の一文字
[...]	集合中の任意の一文字
[^...]	集合に含まれない任意の一文字
*	0 回以上の繰り返し
^	行の先頭
\$	行の末尾
¥a	特殊文字 a を普通の文字にする
¥(...¥)	グループの指定。rep の ¥n で参照できる

rep にも特別な意味を持つ文字がある。

文字	機能
&	合致した部分文字列を表す。
¥n	n は 1 ～ 9 の 1 文字で、rex の中の n 番目のグループを表す。

以下は、変数 a の内容を "banana" とする例である。

式	結果
n = subs (a, "an", "--")	n=2, a="b----a"
n = subs (a, "an", "--", 2)	n=1, a="ban--a"
n = subs (a, "an", "--", -2)	n=1, a="b--ana"
n = subs (a, "an")	n=2, a="ba"

正規表現を使用した例として、変数 a の内容を "Advance CAD V15" としたときの式とその結果を示す。

式	結果
n = subs (a, "[^]*", "x")	n=3, a="x x x"
n = subs (a, "^[^]*", "x")	n=1, a="x CAD V15"
n = subs (a, "[^]*\$", "x")	n=1, a="Advance CAD x"
n = subs (a, "[^]*", "-&-")	n=3, a="-Advance- -CAD- -V15-"
n = subs (a, "V¥¥(.¥¥)", "version¥¥1")	n=1, a="Advance CAD version15"
n = subs (a, "[^]*", "&")	n=3, a の内容は変わらない。

ニューライン (¥n) の変換

fgets (後述) でファイルから文字列を読み込んだ場合の文字列の終端は、ほとんどがニューラインになります。文字列中のニューラインを削除する場合は次のようにします。

subs (a, "¥¥n")

キャリッジリターン (¥r) の変換

注記やモデルタイトルを複数行にした場合の改行文字はキャリッジリターンです。

キャリッジリターンを変換するには次のようにします。

キャリッジリターンを削除する : subs (a, "¥r")

キャリッジリターンを < CR > に変換する : subs (a, "¥r", "< CR >")

スラッシュ (/) とバックスラッシュ (¥) の変換

ディレクトリデリミタの Advance CAD での内部コードはスラッシュで表現していますが Window コマンドに渡す場合はバックスラッシュにする必要があります。スラッシュとバックスラッシュを変換するには次のようにします。

スラッシュをバックスラッシュに変換する : subs (a, "/", "¥¥")

バックスラッシュをスラッシュに変換する : subs (a, "¥¥¥¥", "/")

10.3.10.3 変数テーブル

`clear (v1, ..., vn)`

変数指定なしの場合は、全ての外部変数を消去する。マクロ内で `extern` 宣言で参照している外部変数も消去される。内部変数は消去しない。

変数を指定した場合は、指定変数だけを消去する。外部変数、内部変数どちらも指定できる。マクロ内で `extern` 宣言した外部変数も無効となり、それ以降は参照できなくなる。

`l = length(s)`

変数 `s` の要素数を調べる。

`length` を適用したとき、配列であれば配列の長さを、文字列型であれば文字数を、点座標型であれば `2(x,y)` を返す。つまりデータ型によって決まる。

変数が数値型のように純粋に単一なものに `length` 関数を適用したときは、長さは1。そして長さ1の場合は添字 `[0]` を付けて参照してもよい。

```
c=10;
c[0]=c[0]*2;
```

変数 `c` は、参照するとき `c` でも `c[0]` でもかまわない。これは配列ではない変数を配列と同様にみなして処理するときに都合がよいようにするための手段である。それ以外は `c` と記述すべきである。会話型入力関数などで、結果が配列であったり配列でなかったりするときに効果がある。

`s = vtype(v)`

変数 `v` の型を得る。

結果 0 = 未定義。 `vtype` は与えられた変数が未定義のときは0を返す。これは、ある変数が定義されていないければ、何かデフォルトの処理をする、といったことができるようにするため。

```
2 = 座標型 (@Xs@Ys など)
3 = 数値型
4 = 文字列型
5 = コマンドエンド型 <CE>
6 = デジタイズ座標型
7 = バックスペースキー型 <BS>
10 = スクリーン座標型
11 = スペースキー型 <SP>
12 = カーソル位置型
15 = アイテム識別子型
18 = テンポラリポイント型
30 = 配列
```

`n = vdump(filename, type, var, ...)`

指定した変数またはすべての外部変数をファイルに書き出す。

`filename` は出力ファイル名を表す文字列。

`type` は書き込み "w", 追加書き込み "a" を指定する。

`var` は出力する変数名の並び。変数名を指定しなければすべての外部変数を出力する。関数の値 `n` は出力した変数の数。

ファイルの内容はマクロと同じ形式である。`vdump` 関数で出力したファイルをマクロとして実行することができるので、変数の保存に使用できる。外部変数 `x, y, z` だけ残して他の外部変数を消去したいときは、以下のようにする。

※ 前提として `c:\ACAD\work` に `tmp` フォルダを作成する

```
vdump("/tmp/X. MAC", "w", x, y, z);
clear();
MACRO "/tmp/X. MAC"
```

またマクロのデバッグのとき、ある変数の内容がどのように変わっていくか、トレースするのにも使える。

```

vdump("/tmp/t.dat", "w", x);
for (j=0; j<10; j+=1) {
:
vdump("/tmp/t.dat", "a", x);
:
}

```

`l = array(n)`

一次元配列を生成する。`n` は配列の要素数で、 $1 \sim 16384$ 。

`l = array(n, m)`

二次元配列を生成する。 $2 \leq n, m$ かつ $n*m \leq 16384$

`s = error()`

Advance CAD のコマンドのエラーを調べる。

```

s = 0    : エラーなし
s ≠ 0    : エラーあり

```

Advance CAD のコマンドは入力エラー・実行エラーを発見すると、ブザーを鳴らし、エラーメッセージを表示する。そして最後に表示したエラーメッセージの番号を保持している。`error` 関数はこの最後のエラーメッセージ番号を取出す。

エラーメッセージは、ファイル `msg/ERRxx.TXT` を参照のこと。また負のエラーメッセージ番号はブザーを鳴らさない時で、正のエラーメッセージ番号で見る。

エラーメッセージ番号は次の 2 つの場合にエラーなし (=0) に再設定される。

- ・ マクロが呼び出されたとき
- ・ `error` 関数が使用されたとき

たとえば以下のように記述する。

```
if(error()) { return; }
```

10.3.10.4 会話型入力関数

マクロの中からオペレータの入力を得るのに使う。マクロは入力が完了するまで停止状態になる。入力が完了すると続行する。

以下の入力関数を使用できる。

```

getnum()
gettxt()
getpnt()
getpos()
getdig()
getany()
gettpt()

```

`n` は最大入力個数 ($1 \sim 64$) を指定する。

会話型入力関数は最大入力個数に達するかコマンドエンド<CE>を入力すると終了する。1 個目の入力で<CE>を入力すると<CE>を返す。複数入力時はバックスペースキーを入力すれば 1 つずつ入力を取り消す。1 つも入力がないときはバックスペースを入力するとバックスペースを返す。

`getnum` 関数は数値入力のための関数で、文字列や座標入力を行なっても無視する。`getnum` 関数は正しい数値入力があるまで終了しない。ただし、コマンドエンド<CE>と、バックスペース<BS>だけは、どの関数でも正しい入力として受付ける。

`l = getnum(n)`

最大 `n` 個の数値を入力する。

この関数の出力は数値型または配列。

l = gettxt(n)

最大 **n** 個の文字列を入力する。**n** は文字列の数であり、文字数ではない。

1 つの文字列は 250 文字以内。この関数の出力は文字列型または配列。

図面に記入した注記や寸法の文字列をピックして取り込むことができる。文字列の位置を指示するには、デジタイズするか座標を入力する。

s = gettxt(text);

文字列を編集する。引数で渡した文字列が入力ウインドウに表示され、修正できるようになる。

l = getpnt(n)

最大 **n** 個のモデル座標点を入力する。

点の入力方法は、座標を入力する方法とマウスなどでデジタイズする方法の 2 つがある。つぎのように座標入力したとき、出力は座標型。

<x, y>

@Xs@Ys, @DXs@DYs

デジタイズしたときは、出力はデジタイズ点型になる。

getpnt 関数を使用して入力するとき、テンポラリポイント、テンポラリマスクを切り替えることができる。

getpnt 関数は **n = 1** のときだけスペースキー入力を受け付ける。これはアイテム選択時の次候補アイテムの選択に使用する。

l = getdig(n)

最大 **n** 個のスクリーン座標を入力する。出力はスクリーン座標型または配列。

getdig 関数は **n=1** のときだけスペースキーの入力を受け付ける。マクロ内で MODEL/DSP, SYM/DSP コマンドなどで一覧表示をしたとき、次ページを表示させるためにスペースキー入力が必要なため。

l = getpos(n)

最大 **n** 個のカーソル位置を入力する。

メッセージウインドウ、グラフィックスウインドウ、またはテンポラリウインドウの位置を得る。出力はカーソル位置型または配列。

getpos 関数は **n=1** のときだけ、スペースキーの入力を受け付ける。この関数は画面に表示したファイル名などをピックして取り込むときに使用するため、次頁を表示するためのスペースキーの入力を受け付けると不便なため。

l = getany(n[, ignore]);

この関数は数値、文字列、モデル座標点、スクリーン座標、カーソル位置、コマンドエンド、スペースキー、バックスペースキーのいずれかを得るのに使う。

getnum は数値、gettext は文字列入力だけに用いるのに対して、getany は上記のように複数タイプの入力を受け付ける。

getany 関数は **n=1** のときだけスペースキー入力を受け付ける。

モデル座標点、スクリーン座標、カーソル位置は、ロケータデバイスを使用するが、いずれかひとつしか得ることはできない。

グラフィックウインドウでは、モデル座標点を得られ、スクリーン座標、カーソル位置は得ることができない。

メッセージウインドウ、テンポラリウインドウ（後述）では、カーソル位置だけを得ることができる。

コマンド入力するか、マクロテキスト割付ボタンを選択すると、マクロの実行を終了し、選択されたコマンドまたはマクロテキストを実行する。

これを避けるには、引数 **ignore** を 1 にする。**ignore** の値が 1 のときは、コマンド入力は受け付けなくなる。引数 **ignore** を省略した場合は **ignore=0** としたのと同じになる。

l = gettpt(n)

最大 n 個のテンポラリポイント座標点を入力する。

<x,y> のように座標入力した場合はその点をそのままテンポラリポイント座標とする。ディジタイズの場合はその時点でのテンポラリポイント指定（端点、交点など）に基づいてテンポラリポイントを作成する。テンポラリポイント指定が端点や自動点のときは一度のディジタイズで一つのテンポラリポイントが生成されるが、交点のときは一つのテンポラリポイントを作成するために二度のディジタイズが必要になる。

以前の版ではバックスペース <BS> はなかった。バックスペースを返すようにした理由は、次のようなときに使うと便利のため。

```
LCP
do {
    [p=getpnt(1)]
} while (vtype(p) != 5);
```

このプログラムセグメントは線分を作るためのものだが、getpnt 関数で <CE> が入力されるまで点入力を繰り返す。getpnt 関数でバックスペースを入力すると、それはそのまま LCP コマンドに送られる。

LCP コマンドは直前の点入力を取り消す。このように会話型入力関数にバックスペースを返すようにすると、便利である。

Version 15.02 以降、テンポラリポイント型の追加により以下の変更があります。

for 文、while 文、do-while 文、switch 文の条件式に会話型入力関数（getpnt(), getnum() など）を記述することはできなくなりました。これらの条件式に会話型入力関数が記述されているとコンパイルエラーになります。

if 文、else-if 文の条件式には従来通り記述できます。

if 文の条件式での記述例（正しく動作する）

```
if (vtype(p = getpnt(1)) == 6) {
    .
}
```

while 文の条件式の記述例（書き換えが必要）

```
STR/CP
while (vtype(p = getpnt(1)) != 5) { [p] }
GLO <CE>
```

これはコンパイルエラーになる。以下のように書き換える。

```
STR/CP
while (1) {   または   for (;;) {
    if (vtype(p = getpnt(1)) == 5) { break; }
    [p]
}
GLO <CE>
```

10.3.10.5 座標・デジタイズ点・スクリーン座標・カーソル位置型・テンポラリポイント型データの生成

マクロ内でこれらのデータ型を生成するには、以下の関数を使用する。

スクリーン座標型

```
p = dig(ix, iy)
```

カーソル位置型

```
p = pos(ix, iy, iw)
```

iw はウインドウの番号で、使用できるウインドウ番号は以下の通り。

33= テンポラリウインドウ

6 = メッセージウインドウ

デジタイズ点型

```
p = pdg(x, y)
```

座標型

形式 (1) $p = pce(x, y)$

形式 (2) $p = pce\left(\begin{Bmatrix} "@X" \\ "@DX" \end{Bmatrix}, x, \begin{Bmatrix} "@Y" \\ "@DY" \end{Bmatrix}, y\right)$

形式 (3) $p = pce\left(\begin{Bmatrix} "@R" \\ "@DR" \end{Bmatrix}, r, \begin{Bmatrix} "@A" \\ "@DA" \end{Bmatrix}, a\right)$

モデル座標のときは形式 (1)、補助座標系を生かすときは形式 (2),(3) を使う。

(2) は直交座標系、(3) は極座標系の入力である。(2), (3) の形式はその時点で評価し、直交座標系のモデル座標に変換する。

10.3.10.6 配列要素を並べ替える関数

```
err = sort(aname [, "option"]);
```

配列の要素を並べ替える。

aname : 並べ替えるデータを含む配列の変数名。部分配列は不可。配列の全要素のデータ型は文字列型のみ、数値型のみ、あるいは座標型、デジタイズ点型のみでなければならない。文字列型と数値型の混在や、他の型の要素が混じっていると処理できない。

option : 並べ替えの順序を指定する文字列を指定する。数値型、文字列型では降順（大きい方から小さい方）に並べたいとき "descend" と記述する。省略時は昇順。

座標型、デジタイズ点型では、省略時は X 座標で昇順、X 座標が等しければ Y 座標昇順で並べ替える。これ以外の方法で並べ替えたいときはつぎのようにしてソートキーを指定する。2つのキーの区切りは ; とする。

x : ascend (X 座標昇順)

x : descend (X 座標降順)

y : ascend (Y 座標昇順)

y : descend (Y 座標降順)

たとえばつぎのように記述する。

X 座標降順、Y 座標昇順のとき "x:descend;y:ascend"

Y 座標昇順、X 座標降順のとき "y:ascend;x:descend"

関数の値は、ソートした場合に 0、ソートできない場合に負の正数値 (-1) を返す。

入力引数 aname で指定した配列の要素が並べ替えられる。

10.3.10.7 アイテム識別子を返す関数

マクロ内でアイテム識別子型変数を生成する関数は以下のとおり。

関数の出力はアイテム識別子型。

```
id = idptr(s)
```

数値 s を Advance CAD のアイテム識別子とする。 $1 \leq s \leq 262143$

LBP コマンドで線分を 1 本引き、VER コマンドで調べるという場合は次のようになる。

```
LBP @X0@Y0 @X100 <CE>
VER [idptr(#CURMAXID)]
```

VER コマンドのシンタックスは「VER IS」である。IS は VER コマンドで調べたいアイテムをロケータデバイスで選択することを表わしている。

マクロの中では、そのアイテムの識別子がなんらかの方法で分かっているならば、ロケータデバイスで選択するかわりにその番号を使える。そのとき、コマンドに渡す数値がアイテム識別子であることを知らせるのが `idptr` 関数である。

LBP コマンドで新しく作った線分を指す識別子はシステムレジスタ `#CURMAXID` にある。これを VER コマンドに渡す。

`[idptr(#CURMAXID)]` は、レジスタ `#CURMAXID` の値をアイテム識別子に変換して `dvance CAD` に渡す。VER コマンドは正しく作動する。

これに対し、`[#CURMAXID]` とすると、レジスタ `#CURMAXID` の値がそのまま数値として `Advance CAD` に渡る。そして VER コマンドは不正入力とみなしエラー処理を行う。

```
ids = actlst()
```

アクティブリスト全体を得る。

```
ids = actlst(s1,s2)
```

アクティブリストの `s1` 番目から `s2` 番目までのアイテム識別子を得る。

$1 \leq s1 \leq s2 \leq \text{\#CURACTCNT}$

```
id = actlst(s)
```

アクティブリストの `s` 番目のアイテム識別子を得る。

$1 \leq s \leq \text{アクティブリストの中のアイテム数}$

アクティブリストが空のとき、`s` が正しい範囲にないときは、0 を返す。

`actlst` 関数はアクティブリスト中のすべてのアイテムに対して同じ処理をしたいときに使用できる。`actlst` 関数の出力は、アイテム識別子型である。たとえば以下のようにする。

```
n = #CURACTCNT;
for (i=1; i<=n; i+=1) {
    ITMPIC [actlst(i)]
    TEXT/BOX ONN <CE>
}
```

```
id = pick ("name")
```

指定の名前がついたアイテムの識別子を得る。

指定した名前がないときは、数値 0 が返る。

```
id = pick(p)
```

点 (座標型、デジタイズ点型) の近傍にあるアイテムの識別子を得る。

アイテムがピックできないときは数値 0 が返る。

10.3.10.8 ファイル入出力機能

UNIX ストリームファイル (テキストファイル) への出力、入力を行うことができる。

C 言語の関数を模して作成した。

<code>fopen</code>	ファイルをオープンする
<code>fclose</code>	ファイルをクローズする
<code>fputs</code>	ファイルに書き出す (出力)
<code>fgets</code>	ファイルを読む (入力)
<code>fseek</code>	ファイルのカレントポジションを移動する
<code>ftell</code>	ファイルのカレントポジションを得る
<code>fname</code>	ファイルのフルパス名を作成する

`fn = fopen(filename, type)`

ファイルをオープンする。ファイルがオープンできたらファイル記述子をあらわす 3～6 の正整数を返す。エラーのときは負の値を返す。ファイルは同時に 4 つまでオープンできる。

`type` はファイルへのアクセスの方法を指示する文字列で、以下の文字列を指定する。

- "r" : 既存のファイルの入力用オープン
- "w" : 新規ファイルの作成または既存ファイルの内容を切り捨てる出力用オープン
- "a" : 新規ファイルの作成または既存ファイルに追加する出力用オープン。
- "r+" : 既存ファイルの修正（入力・出力両用）のためのオープン。入出力はファイルの先頭から行われる。
- "w+" : 新規ファイルの作成または既存ファイルの内容を切り捨てる修正用オープン
- "a+" : 新規ファイルの作成または既存ファイルに追加する修正用オープン。

`err = fclose(fn)`

ファイル記述子番号 `fn` のファイルをクローズする。引数なしのときはオープンしているファイルをすべてクローズする。正常終了のとき 0 を返す。エラーのときは負の値を返す。

`n = fputs(s, fn)`

文字列 `s` をファイル記述子番号 `fn` のファイルへ出力する。

```
fputs("Hello world\n", fn);
```

ファイルへ "Hello world" の文字列とニューライン文字 (`\n`) を出力し、出力した文字数を返す。この場合は 12 文字。エラーのときは負の値を返す。

UNIX ではテキストファイルの 1 行の終わりはニューライン文字である。ファイルに出力する文字列の最後にニューライン文字がないと、すべての行がひとつにくっついてしまう。

特別なファイル記述子番号として 1 は標準出力、2 はエラー出力を表す。これらはオープンしないで使用する。マクロのデバッグに利用できるかもしれない。

`s = fgets(n, fn)`

ファイル記述子番号 `fn` のファイルから最高 `n` 文字読み込む。 `n` 文字読み込む前にニューライン (`\n`) が現れたら、そこまでを返す。1 文字も読み込めずにファイル終端に達したときは 0 を返す。エラーのときは負の値を返す。

$1 \leq n \leq 2048$ とする。

先ほどの `fputs` の結果を次のようにして読み込んだとする。

```
s=fgets(30, fn);
n=length(s);
```

読み込んだ文字列 `s` は "Hello world\n" となり、文字数 `n` は 12 となる。今度は最大入力文字数を 8 とする。

```
s=fgets(8, fn);
n=length(s);
```

`s` は "Hello wo" となり、文字数は 8 になる。1 行全部読み込むことができない。

続いて

```
s=fgets(8, fn);
n=length(s);
```

とすると、今度は `s` は残りの "rld\n", 文字数は 4 文字。したがって必要ならば読み込んだ文字列の最後がニューラインかどうかで 1 行の終わりを判定できる。

文字列中のニューラインを削除するには `subs()` を使用する。

```
s=fgets(250, fn);
subs(s, "¥¥n");
```

`err = fseek(fn, ofs, org);`

この関数はファイルのカレントポジションを移動する。

`fn` はファイル記述子番号。

`ofs` は移動する文字数。`org` で指定した場所に対する相対値。

`org` は `ofs` の基準位置。

- 0 = 最初から
- 1 = 現在の位置
- 2 = 終わり

正常終了のとき 0、エラーのときは負の値を返す。

ファイルの終わりへ移動する。

```
fseek (fn, 0, 2);
```

ファイルの先頭へ戻る。

```
fseek (fn, 0, 0);
```

```
ofs = ftell (fn);
```

ファイルのカレントポジションを得る。

fn はファイル記述子番号 (3 ～ 6)。関数の値はカレントポジションとしてファイルの先頭からのバイト数 (0 ～) を返す。エラーの場合は -1 を返す。

```
s = fname (type, name)
```

ファイル名を与えてディレクトリ・ファイルタイプを付けたフルパス名を作る。

type : ファイルの種類を表す文字列 ACAD.SET のキーワード “#MACRO#”, “#TEXT#” などを記述する。

name : ファイル名。ホームディレクトリを表す “～” 文字を含めてもよい。
“～” 文字はホームディレクトリ名に変換される。

エラーのとき s は長さ 0 の文字列となる。

引数なしで使えば、フルパスのテンポラリファイル名を得ることができる。

```
s=fname();
```

? テンポラリファイル

```
/acad/work/ACAD-20040204-090000-00105.TMP
```

ディレクトリ : 作業ディレクトリ

ファイル名 : “ACAD- ユーザ名 -YYYYMMDD-HHMMSS- プロセス ID”

デフォルトのセッションファイル名と同様。

詳しくは「システム管理者の手引き : AdvanceCAD の起動」を参照。

拡張子 : “.TMP”

10.3.10.9 ファイル属性を調べる

```
r = fstat (filename, keyword);
```

ファイルの属性 (rwx 権、サイズ、日付) を調べる。

```
fstat (filename, “exist”)
```

ファイルが存在するとき 1、ないとき 0 を返す。

```
fstat (filename, “read”)
```

ファイルが存在しリード権があるとき 1 を返す。それ以外は 0 を返す。

```
fstat (filename, “write”)
```

ファイルが存在しライト権があるとき 1 を返す。それ以外は 0 を返す。

```
fstat (filename, “execute”)
```

ファイルが存在し実行可能なとき 1 を返す。それ以外は 0 を返す。

```
fstat (filename, “size”)
```

ファイルサイズをバイト数で返す。ファイルが存在しないときは -1 を返す。

```
fstat (filename, “date”)
```

ファイルの更新日付を “YYYY/MM/DD HH:MM:SS” の 19 文字で返す。ファイルが存在しないときは -1 を返す。

10.3.10.10 テンポラリウィンドウ

グラフィックスクリーンの一部にテンポラリウインドウを開き、コマンド名やシンボルを表示させてメニューウインドウとして使用できる。

テンポラリウインドウを開いたとき、その下に隠れた部分は操作できない。テンポラリウインドウを閉じると元通りになる。

テンポラリウインドウの位置・大きさはメニューのゾーン定義ファイルで設定する。

```
+ (33) “Temporary Window” <1330, 240> <2250, 2000> [ 40 , 10 , 30 ]
```

(33) はテンポラリウインドウのウインドウ識別番号。

[40,10,30] は、このウインドウが 40 行 10 列の表になることを指示する。

これらは `mopen` 関数でテンポラリウインドウをオープンしたとき得ることができる。

```
p = mopen(m, uv1, uv2, ncol, nrow)
```

```
p = mopen(m)
```

Advance CAD 起動時は、ゾーン定義ファイルで設定した値が使用される。一度設定を変える
と次に変更するまで設定値を保持する。

`p = mopen(m)` は、設定を変更しないでテンポラリウインドウを開くときに使用する。

```
m          : ダミー引数。ウインドウ識別番号だが参照しない。33 と記す。
uv1, uv2    : テンポラリウインドウの対角点をスクリーン座標型で与える。この 2 点はメニュー
              ゾーン定義ファイル内の Graphic zone 行で定義されている領域の範囲内で指定す
              る。
ncol, nrow   : テンポラリウインドウの列数 (ncol), 行数 (nrow) を数値型 (整数) で与える。
p            : オープンが成功したときカーソル位置型データを返す。
              p[0] : xmax (列数)
              p[1] : ymax (行数)
              p[2] : ウインドウ番号
              エラーのときは数値 0 を返す。
```

```
mclose(p)
```

テンポラリウインドウをクローズする。

```
p          : クローズするウインドウをカーソル位置型データで与える。
              p[0], p[1] は参照しない。
              p[2] : クローズするウインドウ番号
```

```
mcolor(p, color, lf, lw)
```

テンポラリウインドウの表示色、罫線の線種、線太さを設定する。ウインドウオープン時に
初期値を設定する。(初期値は、色は 1、線種は 1、線幅は 1)。それ以外はつぎに変更するま
で設定値を保持する。

現在値を変更したくない表示属性に対しては、0 を渡す。

```
p          : 色を設定するウインドウをカーソル位置型データで与える。
              p[0], p[1] は参照しない
              p[2] はウインドウ番号 (=33)
color       : カラー番号。
lf          : 線種番号。罫線で使用する。0, 1 ~ 63。
lw          : 線太さ番号。罫線で使用する。0, 1 ~ 16。
```

```
mecho(p, arg1, ..., argn)
```

表示。グラフィックスゾーンへの文字列の表示はできない。

```
p          : 表示位置をカーソル位置型データで指定する。
              p[0] : x          1 ≤ x ≤ xmax
              p[1] : y          1 ≤ y ≤ ymax
              p[2] : ウインドウ番号
arg1, ..., argn : 表示する文字列、数値、定数または変数。以下のデータ型は複数の要素を持
                  つので、つぎのような形式で表示する。
                  スクリーン座標型 (x, y)
                  カーソル位置型 (x, y, w)
                  デジタイズ点型 (x, y)
                  座標型 (x, y)
```

```
merase(p)
```

ウインドウをイレーズする。

```
p          : イレーズする領域をカーソル位置型データで指定する。
              p[0] : x          0 ≤ x ≤ xmax
              p[1] : y          0 ≤ y ≤ ymax
              p[2] : ウインドウ番号
              x=0, y=0 のときは全体を消去する。
```

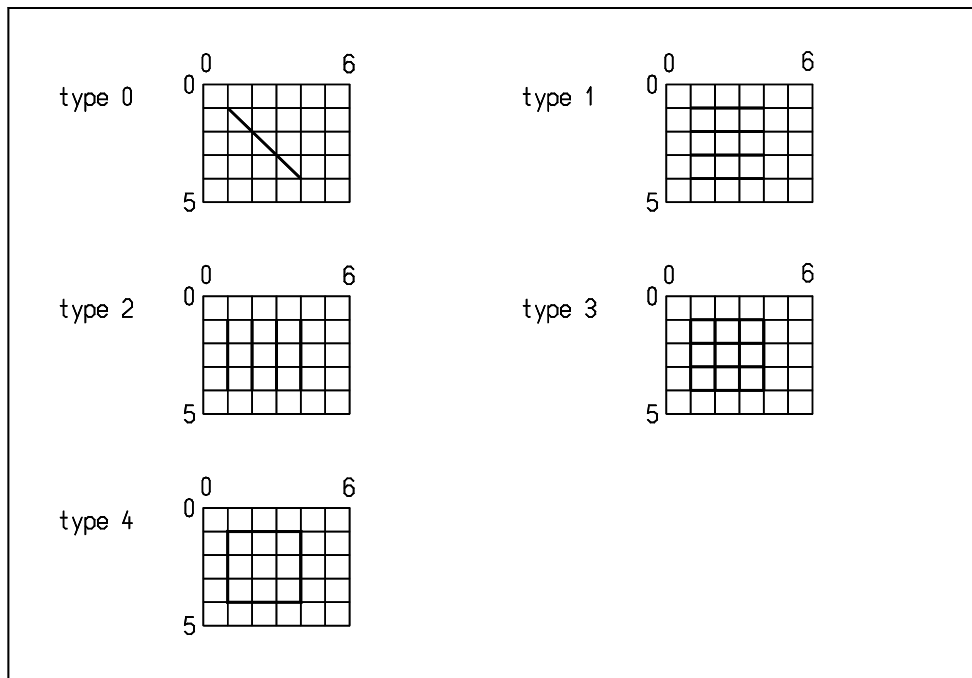
$x=0, y>0$ のときは横 1 行を消去する。
 $x>0, y=0$ のときは縦 1 列を消去する。
 $x>0, y > 0$ のときは y 行 x 列のみ消去する。

`mgrid(type, ps, pe)`

テンポラリウインドウに罫線を書く。

type : 罫線の種類を指定する。
 0 は線分。
 1 は複数の横罫を引く。
 2 は複数の縦罫を引く。
 3 は格子の罫線を引く。
 4 は 2 点を対角点にする矩形。
ps : 罫線の位置をカーソル位置型データで指定する。
 $ps[0]$ は x 。 $0 \leq x \leq xmax$
 $ps[1]$ は y 。 $0 \leq y \leq ymax$
 $ps[2]$ はウインドウ番号。(=33)
pe : 罫線の位置をカーソル位置型データで指定する。
 $ps[0]$ は x 。 $0 \leq x \leq xmax$
 $ps[1]$ は y 。 $0 \leq y \leq ymax$
 $ps[2]$ はウインドウ番号。(=33)

たとえば、テンポラリウインドウ 5 行 6 列の場合で、第 1 点が (1,1)、第 2 点が (4,4) の場合は、**type** の違いによって次の図のような結果になる。



`msym(name, p, flag)`

テンポラリウインドウにシンボルを表示する。

p はシンボル配置位置 (カーソル位置型)。

$p[0]: x \quad 1 \leq x \leq xmax$
 $p[1]: y \quad 1 \leq y \leq ymax$
 $p[2]:$ テンポラリウインドウ番号
flag: 省略可

テンポラリウインドウの指定の行 / 列に、シンボルを表示する。この場合は `msymup()` 関数は無効である。

バージョン7までは線種1、線幅1で表示していたが、バージョン8からは元の線種、線幅が反映される。シンボルの色は `mcolor` 関数で設定した色になる。

シンボル形状をウインドウの縦横比率の変化に合わせて伸縮させて表示するかどうかを指定する。

`flag == 0` : 伸縮しない。(省略時)

`flag == 1` : 伸縮する。

`msym(name, x, y, flag)` または

`msym(name, uv, flag)`

テンポラリウインドウにシンボルを表示する。`msymup()` 関数が有効。

`name` : シンボル名

`x, y` : シンボル配置位置 (数値2つ)。旧形式。

`uv` : シンボル配置位置 (スクリーン座標型)。バージョン9からの形式

`flag` : 省略可

`msymup()`

テンポラリウインドウのシンボルインスタンス中の寸法テキストを更新する。寸法テキストが変数名または計算式のときに、変数に値が設定されていれば変数名がその値に代わる。その変数が未定義ならば変数名のままとなる。

以下に例を示す。

```
/* Open Temporary window */
p = mopen(33);
ncol = p[0];
nrow = p[1];

/* display (x,y) */
for (y=1;y<=nrow;y+=1) {
    mcolor(p, y%15+1, 0, 0);
    p[1] = y;
    for (x=1;x<=ncol;x+=2) {
        p[0] = x;
        mecho(p, p[0], ",", p[1]);
    }
}

/* erase */
p[0] = 0;
for (y=1;y<=nrow;y+=3) {
    p[1] = y;
    merase(p);
}

/* close Temporary window */
mclose(p);
```

`mecho`, `merase` 関数は、テンポラリウインドウの他にメッセージウインドウにも使用できる。このウインドウは常に存在しているため、`mopen`, `mclose` は使えない。

表示する文字の色は変更できない。文字列の表示位置は `pos` 関数で生成するか `getpos` 関数で得られる。

10.3.10.11 アイテム数を得る関数

ピクチャ上のアイテム数を得る。

`s = getitm(catg, id)`

- | | | | | |
|------|---|---|---|--|
| catg | : | 0 | = | 指定されたピクチャのアイテム数。
id はピクチャ番号 (1-256) または 0 (全ピクチャの合計)。 |
| | | 1 | = | 現在のピクチャの指定されたアイテムタイプのアイテム数。
id はアイテムタイプ番号 (1-31)。 |
| | | 2 | = | 現在のピクチャの指定されたクラスのアイテム数。
id はクラス番号 (1-256)。 |
| | | 3 | = | 現在のピクチャの指定されたレビジンのアイテム数。
id はレビジョン番号 (1-256)。 |
| | | 4 | = | 現在のピクチャの指定された線種のアイテム数。
id は線種番号 (1-63)。 |
| | | 5 | = | 現在のピクチャの指定された線幅のアイテム数。
id は線幅番号 (1-16)。 |

アイテムタイプ番号

- 1 : 点
- 2 : 直線
- 3 : 円弧
- 4 : 自由曲線
- 5 : ストリング
- 9 : 複合アイテム
- 11 : 注記
- 12 : 記号
- 13 : 寸法
- 14 : 幾何公差
- 15 : ハッチング
- 16 : 塗潰し
- 27 : メンバー
- 28 : APG
- 30 : シンボル
- 31 : サブモデル
- 32 : イメージ

アイテム数 (0 ~) を返す。

10.3.10.12 モデル変更の有無を調べる関数

この関数でモデルが変更されているというのは以下のうちの一つ以上が該当することを言う。

- (1) アイテムが変更されている。ラスター背景の変更も含む。
 - (2) 図面配置情報が変更されている。
 - (3) モデルタイトルが変更されている。
 - (4) ドローイングタイトルが変更されている。
- この関数で「(1) アイテムが変更されている」状態を変更することはできない。

`s = dbchgd()`

モデル変更の有無を調べる。

モデルが変更されているときは 1、変更されていないときは 0 を返す。

`dbchgd(val)`

モデル変更の有無を設定する。

`val` はモデル変更の有無を示す整数を渡す。

0 : モデルは変更されていない。

1 : モデルは変更されている。

使用例

```
/* 現在のモデル変更の有無を保持しておく */
modified = dbchgd();
```

```

:
/* 図面配置情報を変更し、出図する */
:
/* モデル変更の有無を元に戻す */
dbchgd(modified);

```

10.3.10.13 マスクの状態を得る関数

選択マスク、表示マスクの状態を得る。

```

s = getmsk(catg, id)
catg : 1      = アイテムタイプ選択マスク。
          id はアイテムタイプ番号 (1-31)。
      2      = クラス選択マスク。
          id はクラス番号 (1-256)。
      3      = レビジョン選択マスク。
          id はレビジョン番号 (1-256)。
      4      = 線種選択マスク。
          id は線種番号 (1-63)。
      5      = 線幅選択マスク。
          id は線幅番号 (1-16)。
      6      = 現在のピクチャのアイテムタイプ表示マスク。
          id はアイテムタイプ番号 (1-31)。
      7      = 現在のピクチャのクラス表示マスク。
          id はクラス番号 (1-256)。
      8      = 現在のピクチャのレビジョン表示マスク。
          id はレビジョン番号 (1-256)。
      9      = 現在のピクチャの線種表示マスク。
          id は線種番号 (1-63)。
     10      = 現在のピクチャの線幅表示マスク。
          id は線幅番号 (1-16)。
     11      = 一時的なアイテムタイプ選択マスク。
          id はアイテムタイプ番号 (1-31)。
     12      = 一時的なクラス選択マスク。
          id はクラス番号 (1-256)。
     13      = 一時的なレビジョン選択マスク。
          id はレビジョン番号 (1-256)。
     14      = 一時的な線種選択マスク。
          id は線種番号 (1-63)。
     15      = 一時的な線幅選択マスク。
          id は線幅番号 (1-16)。

```

アイテムタイプ番号

```

1 : 点
2 : 直線
3 : 円弧
4 : 自由曲線
5 : ストリング
9 : 複合アイテム
11 : 注記
12 : 記号
13 : 寸法
14 : 幾何公差
15 : ハッチング
16 : 塗潰し
27 : メンバー
28 : APG
30 : シンボル
31 : サブモデル

```

32 : イメージ

選択／表示できるときは 1、できないときは 0 を返す。

※ 関連コマンド

MSAV
MRES

10.3.10.14 カラー割付情報を得る関数

カラー割付の状態を得る。

```
s = getclr(catg, id)
```

catg : 0 = カラーに関連する情報を得る。
id = 0 : 現在のカラー割付タイプを得る。
戻り値 = 1 : アイテムタイプによるカラー割付。
= 2 : クラスによるカラー割付。
= 3 : レビジョンによるカラー割付。
= 4 : 線種によるカラー割付。
= 5 : 線幅によるカラー割付。

1 = アイテムタイプによるカラー番号を得る。
id はアイテムタイプ番号 (1-31)。
指定されたアイテムタイプに割り付けられているカラー番号が返る。

2 = クラスによるカラー番号を得る。
id はクラス番号 (1-256)。
指定されたクラスに割り付けられているカラー番号が返る。

3 = レビジョンによるカラー番号を得る。
id はレビジョン番号 (1-256)。
指定されたレビジョンに割り付けられているカラー番号が返る。

4 = 線種によるカラー番号を得る。
id は線種番号 (1-63)。
指定された線種に割り付けられているカラー番号が返る。

5 = 線幅によるカラー番号を得る。
id は線幅番号 (1-16)。
指定された線幅に割り付けられているカラー番号が返る。

10.3.10.15 図面配置されているウィンドウ数を得る関数

図面配置されているウィンドウ数を得る。

```
s = getdrw(page)
```

page: 図面配置頁 (1-256)。

配置されているウィンドウ数 (0 ~) を返す。

10.3.10.16 ピクチャマトリックスを得る関数

ピクチャマトリックスを得る。

```
s = getmtx(pic, id)
```

pic : ピクチャ番号 (1-256)。
id : 1 = X 軸の X 方向成分。
2 = X 軸の Y 方向成分。
3 = X 軸の Z 方向成分。
4 = Y 軸の X 方向成分。

- 5 = Y 軸の Y 方向成分。
- 6 = Y 軸の Z 方向成分。
- 7 = Z 軸の X 方向成分。
- 8 = Z 軸の Y 方向成分。
- 9 = Z 軸の Z 方向成分。

指定されたピクチャのピクチャマトリックスの、指定された要素の値を返す。

10.3.10.17 製図用定義などの現在値を得る関数

マクロで製図用定数などを一時的に変更し、あとで変更前の値に戻すときなどに使用する。

```
s = getprm(catg, pid)

catg  :  0 = メニュー言語
        1 = 製図用定数
        2 = 演算定数
        3 = カーブ（円、自由曲線）表示用定数
        4 = モデル定数
        5 = レビジョン名
        6 = 線種線幅定数
        9 = ピクチャ名
       10 = スクリーンレイアウト名
       11 = クラス名
       12 = ドローイングレイアウト名
       15 = ピクチャスケールファクター
       16 = ピクチャ Z 値
       17 = 物差しカーソル定数
       18 = 同時設計定数
       19 = 線種パターン名
       20 = ピクチャタイトル
pid   :  定数の識別番号（正整数）。いくつかの定数の値を一度に取り出したいときは、識別番号
        の並びを配列型変数で渡す。
```

この関数で得た値を使って定数を再設定するには、対応する Advance CAD コマンドを使用する。

製図用定数の ASCII テキストフォント番号を一時的に変更し、終わったらもとに戻すには、つぎのようにする。

```
s = getprm(1,1); /* save */
TEXT/FONT 2      /* change */
:
TEXT/FONT [s]     /* restore */
```

【catg=0 メニュー言語

識別番号	設定コマンド
1	カレントピクチャのグリッドのオンオフを得る 戻り値：1 = オン 2 = オフ
2	メニュー言語を得る 戻り値：0 = 日本語メニュー 1 = 英語メニュー

【catg=1 製図用定数】

識別番号	設定コマンド	識別番号	設定コマンド
1	TEXT/FONT	2	TEXT/SIZE
4	DIM/TOLSIZE	5	TEXT/GAPHOR
6	TEXT/GAPVER	7	TEXT/ORGHOR
8	TEXT/ORGVER	9	TEXT/SLANT
10	TEXT/BOXMODE	11	DIM/RSL
12	DIM/ANGMODE	13	DIM/ANGRSL
14	DIM/ANGFRSL	15	DIM/TOLRSL
16	DIM/ORIENT	17	DIM/DIASYM
18	DIM/RADSYM	19	DIM/MARKTYPE
20	DIM/MARKSIZE	21	DIM/STUB
22	DIM/GAP	23	DIM/OVER
24	RFN/MARKTYPE	25	RFN/MARKSIZE
26	SEC/MARKTYPE	27	SEC/MARKSIZE
28	SEC/LINEMODE	29	DIM/MODE
30	DIM/UNIT	31	DIM/INTMODE
32	DIM/DPNT	33	LDR/RND
34	FCS/BOXSIZE	35	FCS/TEXTSIZE
36	FCS/MARKTYPE	37	FCS/MARKSIZE
38	BDIM/HABA	39	RDIM/OVR
40	RFN/MRKAWTYP	41	RFN/MRKAWSIZ
42	DIM/TOLUL	43	DIM/TOLLW
44	DIM/TOLUP	45	HYOU/WHIGHT
46	HYOU/THIGHT	47	HYOU/NKETA
48	HYOU/NCOL	49	HYOU/OFFSET
50	TEXT/DXBOX	51	TEXT/DYBOX
52	TEXT/YMIR	53	TEXT/XMIR
54	TEXT/DTATE	55	DIM/ZERO
56	DIM/ODHMARKTYPE	57	FCSDTM/BOX
58	FCSDTM/ARROW	59	LDR/ANG
60	SMARK/SIZE	61	WMARK/SIZE
62	DIM/UREP	63	DIM/FTYP

【catg=1 製図用定数】

識別番号	設定コマンド	識別番号	設定コマンド
64	DIM/FRES	65	TEXT/JSTHOR
66	LDR/MARKTYPE	67	LDR/MARKSIZE
68	DIM/SCALE	69	DIM/TOLSIZE2
70	TEXT/KFONT	71	DIM/TOLZERO
72	RFN/TEXTSIZE	73	SEC/TEXTSIZE
74	CLIN/OVTYPE	75	CLIN/OVSIZE
76	SMARK/TEXTSIZE	77	WMARK/TEXTSIZE
78	FCSDTM/TEXTSIZE	79	TEXT/ALIGN
80	TEXT/RATIO	81	SEC/LWT
82	WMARK/LFT	83	WMARK/DIST
84	DIM/TOLSTL	85	DIM/EXTSTL
87	TEXT/DENIALGAP	88	TEXT/DENIALOVS
89	TEXT/DOUBLEGAP	90	DIM/ACTVAL

【catg=2 演算定数】

識別番号	設定コマンド	識別番号	設定コマンド
1	FC/ITC	2	FC/TOL
3	FCL/ITC	4	FCL/TOL
5	MINDST		

【catg=3 カーブ表示用定数】

識別番号	設定コマンド	識別番号	設定コマンド
1	DSPATOL	2	DSPASEGMIN
3	DSPASEGMAX	11	PLTATOL
12	PLTASEGMIN	13	PLTASEGMAX
21	DSPFMOD	22	DSPFTOL
23	DSPFSEG	32	PLTFTOL
33	PLTFSEG		

【catg=4 モデル定数】

識別番号	設定コマンド	識別番号	設定コマンド
3	MDL/BOX (文字列型、例 “-150, -115, 150, 115”)		
5	KANA	6	MDL/LSTOUT
7	DLGMODE	8	IDT/BOXMODE
10	INCH	11	TRAPBOX
12	TRAPBOX/SCF	13	ZOOM/BLANK
14	ARC/RAD	15	FIG/COUNT
16	TED/CH	17	VIE/AUTO
18	PNUM/DSP		
27	MDL/INPCNF	28	MDL/OUTCNF
29	MDL/MRGCNF	30	MDL/INFOUT
31	PEN/MAX	32	DSP/TITLE1
33	DSP/START1	34	DSP/NBYTE1
35	DSP/TITLE2	36	DSP/START2
37	DSP/NBYTE2	38	DSP/TITLE3
39	DSP/START3	40	DSP/NBYTE3
41	DSP/TITLE4	42	DSP/START4
43	DSP/NBYTE4	44	DSP/TITLE5
45	DSP/START5	46	DSP/NBYTE5
51	ITMDUP/CURCLS	52	ITMDUP/CURREV
53	ITMDUP/CURLFT	54	ITMDUP/CURLWT
101	DSP/SUB	102	DSP/SYM
103	DSP/COMP	104	DSP/APG
105	DSP/MEMB	106	LWT/SUB
107	LWT/SYM	108	LWT/COMP
109	LWT/APG	110	LWT/MEMB
111	LFT/SUB	112	LFT/SYM
113	LFT/COMP	114	LFT/APG
115	LFT/MEMB	116	CLR/SUB
117	CLR/SYM	118	CLR/COMP
119	CLR/APG	120	CLR/MEMB

【catg=5 レビジョン名】

定数識別番号はレビジョン番号。戻り値はレビジョン名が設定されているときは文字型、レビジョン名が設定されていないときはコマンドエンド型になる。

【catg=6 線種線幅定数】

識別番号	設定コマンド	識別番号	設定コマンド
1	LFW	2	LWP
3	LWD	4	AFLD
11	LFT1	12	LFT2
13	LFT3	14	LFT4
15	LFT5	16	LFT6
17	LFT7	18	LFT8
19	LFT9	20	LFT10
21	LFT11	22	LFT12
23	LFT13		
31	LWT1	32	LWT2
33	LWT3	34	LWT4
35	LWT5	36	LWT6
37	LWT7	38	LWT8
39	LWT9	40	LWT10
41	LWT11	42	LWT12
43	LWT13		
51	CLS1	52	CLS2
53	CLS3	54	CLS4
55	CLS5	56	CLS6
57	CLS7	58	CLS8
59	CLS9	60	CLS10
61	CLS11	62	CLS12
63	CLS13		
71	REV1	72	REV2
73	REV3	74	REV4
75	REV5	76	REV6
77	REV7	78	REV8
79	REV9	80	REV10
81	REV11	82	REV12
83	REV13		

【catg=6 線種線幅定数】

識別番号	設定コマンド	識別番号	設定コマンド
1xx00 = Font pattern xx (xx=01-63) この場合は線種パターンを表す文字列が得られる。 たとえば "3,1,1,1"。			

【catg=9 ピクチャ名】

定数識別番号はピクチャ番号。戻り値はピクチャ名が設定されているときは文字型、ピクチャ名が設定されていないときはコマンドエンド型になる。

【catg=10 スクリーンレイアウト名】

定数識別番号はスクリーンレイアウト番号。戻り値はスクリーンレイアウト名が設定されているときは文字型、スクリーンレイアウト名が設定されていないときはコマンドエンド型になる。

【catg=11 クラス名】

定数識別番号はクラス番号。戻り値はクラス名が設定されているときは文字型、クラス名が設定されていないときはコマンドエンド型になる。

【catg=12 ドローイングレイアウト名】

定数識別番号はドローイングレイアウト番号。戻り値はドローイングレイアウト名が設定されているときは文字型、ドローイングレイアウト名が設定されていないときはコマンドエンド型になる。

【catg=15 ピクチャスケールファクター】

定数識別番号は、ピクチャ番号。

【catg=16 ピクチャ Z 値】

定数識別番号は、ピクチャ番号。

【catg=17 物差しカーソル】

識別番号	設定コマンド	識別番号	設定コマンド
1	CURSOR/TYPE	2	CURSOR/GSIZE
3	CURSOR/INTERVAL	4	CURSOR/XLEN
5	CURSOR/YLEN	6	CURSOR/XMIR
7	CURSOR/YMIR		

【catg=18 物差しカーソル】

識別番号	設定コマンド	識別番号	設定コマンド
1	CON_MULT_EXP	2	CON_DIR_WRT
3	CON_MULT_MEMB	4	CON_SEL_RPT

【catg=19 線種パターン名】

定数識別番号は線種番号。戻り値は線種パターン名が設定されているときは文字型、線種パターン名が設定されていないときはコマンドエンド型になる。

【catg=20 ピクチャタイトル】

定数識別番号はピクチャ番号。戻り値はピクチャタイトルが設定されているときは文字型、ピクチャタイトルが設定されていないときはコマンドエンド型になる。

10.3.10.18 環境変数の内容を得る関数

```
s = getenv(env)
env : 環境変数名
```

環境変数の内容を返す。
環境変数が設定されていない場合は長さ 0 の文字列を返す。

[特殊な環境変数名の処理]

以下の環境変数名は、指定された環境変数名が設定されていればその内容を返す。
設定されていなければ以下の内容を返す。

- HOSTNAME
システムのホスト名
- LOGNAME
 - UNIX : 環境変数 LOGNAME (ログインユーザ名) の内容
 - Windows : 環境変数 USERNAME (ログインユーザ名) の内容
- USERNAME
 - UNIX : 環境変数 LOGNAME (ログインユーザ名) の内容
 - Windows : 環境変数 USERNAME (ログインユーザ名) の内容
- HOME
 - UNIX : 環境変数 HOME の内容
 - Windows : 環境変数 HOMEDRIVE と環境変数 HOMEPATH の内容を連結したもの

10.3.10.19 menu 関数

マクロ実行後、メニューをマクロ実行直前の状態に戻す。

```
s = menu(iopt)
iopt ==1 現在のメニューの状態を保存させる。戻り値は 1。
      ==2 まだメニュー状態が保存されていなければ、メニューの状態を保存する。
           戻り値はメニュー状態を保存したときは 1、しなかったときは 0。
      ==3 保存されたメニュー状態を復元する。保存されていなければ何もしない。
           復元後はメニュー状態が「保存されていない」になる。戻り値は 0。
```

復元対象外のゾーン

以下のメニューゾーンは、メニュー表示用でなかったり、メニューページを変更しないゾーンである。または復元しては都合が悪いゾーンなので、対象外とする。
ゾーン番号 #1, 5, 6, 7, 8, 10, 11, 12, 13, 33

10.3.10.20 ロックファイル確認関数

モデルロックファイルの存在を確認する

```
res = getlock(type)
```

type == 0 ロックファイルのフルパスのファイル名を得る
 戻り値 文字列: フルパスロックファイル名
 <CE> : モデル名未定義時

res = getlck(type, mdlname)
 type == 0 ロックファイルのフルパスのファイル名を得る
 戻り値 文字列: フルパスロックファイル名
 <CE> : モデル名未定義時

type ==1 ロックファイルの状態を戻す
 戻り値 数値 : 0 ロックファイルなし
 数値 : 1 自身で作成したロックファイルがある
 数値 : 2 他のユーザーがモデルをロック中
 数値 : 3 ロックファイルを作成できない

mdlname ロック状態を調べたいモデル名
 ※ モデル名のみ場合は、コンフィグレーションファイル #MODEL# で
 定義されているフォルダを探す
 ※ 絶対パスの場合はそのまま使用

例

新規開始の場合

```
res = getlck(0)
res : なし
```

```
res = getlck(1)
res : 0
```

C:\acad\files\TEST.MDL に保存直後の場合

※ getlck(0) は getlck(0, "C:/acad/files/TEST.MDL") と同等

```
res = getlck(0)
res : C:/acad/files/TEST.MDL_LCK
```

```
res = getlck(1)
res : 1
```

読み取り専用フォルダのモデルを読み込んだ場合・・・Z:\files\TEST.MDL 場合

```
res = getlck(0)
res : Z:/files/TEST.MDL_LCK
```

```
res = getlck(1)
res : 3
```

特定のモデル名を指定する場合

```
res = getlck(0, "C:/acad/files/TEST.MDL")
res : C:/acad/files/TEST.MDL_LCK
```

```
res = getlck(1, "C:/acad/files/TEST.MDL")
res : 0 or 1 or 2 or 3
```

10.3.10.21 モデルロックファイル情報取得用関数

モデルロックファイル情報取得

```
res = getlst(type)
type == 1      ロックファイルの状態を戻す
      == 2      アカウンティング名
      == 3      PC 名
      == 4      使用開始日
```

```

== 5      使用開始時間
== 他      ロックファイルの内容全て出力

res = getlst(type, mdlname)
    type    == 1      ロックファイルの状態を戻す
            == 2      アカウンティング名
            == 3      PC 名
            == 4      使用開始日
            == 5      使用開始時間
            == 他      ロックファイルの内容全て出力

    mdlname  ロック状態を調べたいモデル名
            ※ モデル名のみ場合は、コンフィグレーションファイル #MODEL# で
              定義されているフォルダを探す
            ※ 絶対パスの場合はそのまま使用

    戻り値   <OE>   :   ロックファイルなし
            数値   :   情報なし
            文字列 :   type で指定した情報

例
新規開始の場合
    res = getlst(0, "E21053")
    res : vtype(res)==5( ロックファイルなし ) が戻る

C:\acad\files\E21053.MDL に保存直後の場合
    ※ getlst(0) は getlst(0, "C:/acad/files/E21053.MDL") と同等
    res = getlst(0)
    res : vtype(res)==4( 文字列 ) が戻る
    res の内容 → ユーザー名 [ アカウンティング名 ] PC 名 ( 使用開始日 使用開始時間 )

    res = getlst(1, "C:/acad/files/E21053.MDL")
    res : vtype(res) 4 が戻る
    res : ユーザー名

    ※ アカウンティング未使用の場合は取得できないので数値として 0 が戻る
    res = getlst(2, "C:/acad/files/E21053.MDL")
    res : vtype(res) 3 が戻る
    res : 0

```

10.3.10.22 モデル保存可能状態取得用関数

モデルが保存可能な状態か事前確認用

```

res = getmst()
    戻り値   0 :   保存可能
            1 :   保存不可能

```

保存不可には、読み取り専用フォルダに在るモデルファイル、ロックファイルで制限されている、Advance CAD が起動時読み込み専用オプション (-ro) で起動されている場合等がある。
保存可能の場合も、ディスク状態によるモデル書き込み時の破損、OS による書き込み制限等予期しないエラーも想定されるため、保存後のエラー処理は必要

10.3.10.23 コンフィグレーションファイル取得用関数

現在使用中のコンフィグレーションファイルのフルパス取得

```

s = getcnf()

```

戻り値 文字列: フルパスコンフィグレーションファイル名

10.3.10.24 ACAD 起動時指定オプション取得用関数

ACAD 起動時に指定したオプションを取得

```
s = getsta(catg)
```

catg	== 1	ACAD 起動時指定する、または起動時オプション -aUSER で指定されるアカウントユーザー名
	== 2	環境変数、または起動時オプション -pCONFFILE で指定されるコンフィグレーションファイル名 ※ getcnf() 関数と同じ
	== 3	-iICONFILE で指定されたアイコンファイル名
	== 4	-nPROPATY で指定された受信用プロパティ名
	== 5	sSESSIONNAME で指定されたセッションファイル名 ※ 起動オプションで指定されたセッションファイル名を取得するための関数 現在のセッションファイル名を取るためには getses() で取得する。

戻り値 文字列: typ で指定した情報
無指定の場合は長さ 0 の文字列が戻る

10.3.10.25 スタートアップマクロ名取得用関数

スタートアップマクロ名を取得

```
s = getsmn(catg)
```

catg	== 1	起動オプション -m1 で指定されたマクロ名を絶対パスで返す。無指定の場合はデフォルト値 (絶対パス ¥STARTUP01.MAC) を返す
	== 2	起動オプション -m2 で指定されたマクロ名を絶対パスで返す。無指定の場合はデフォルト値 (絶対パス ¥STARTUP02.MAC) を返す
	== 3	-起動オプション -m3 で指定されたマクロ名を絶対パスで返す。無指定の場合はデフォルト値 (絶対パス ¥STARTUP03.MAC) を返す
	== 4	起動オプション -m4 で指定されたマクロ名を絶対パスで返す。無指定の場合はデフォルト値 (絶対パス ¥STARTUP04.MAC) を返す
	== 5	起動オプション -m5 で指定されたマクロ名を絶対パスで返す。無指定の場合はデフォルト値 (絶対パス ¥STARTUP05.MAC) を返す
	== 6	起動オプション -m6 で指定されたマクロ名を絶対パスで返す。無指定の場合はデフォルト値 (絶対パス ¥STARTUP06.MAC) を返す

戻り値 文字列: マクロ名を絶対パスで返す。(存在確認はしないため、存在を確認したい場合は fstat 関数を使用する)

10.3.10.26 セッションファイル名取得用関数

セッションファイル名を取得

```
s = getses()
```

戻り値 文字列: セッションファイル名を絶対パスで返す。

10.3.10.27 作業中フォルダパス取得用関数

作業中のフォルダパスを取得

```
s = getcwd()
```

戻り値 文字列: 現在作業中のフォルダを戻す。
 ※getsta(9) 関数は、-w で指定されていない場合は値が取れないが、getcwd() の場合は、必ず値が戻る。

10.3.10.28 コンフィグレーションファイルで定義した最大値を取得する関数

コンフィグレーションファイルの MAXID、MAXSR、MAXSB で定義した最大値を取得

```
n = getdbmax(type)
type == 1    MAXID アイテム数の上限
      == 2    MAXSR 1 アイテムのサブレコード数の上限
      == 3    MAXSB 1 アイテム
```

10.3.11 制限

- ループ文、条件文の入れ子深さ
16 重以下
- ラベルの数
16 個以下
- 計算式の長さ
256 文字以内
- 変数の保持数制限
変数名の長さが 6 文字のとき、512 個保持できる。
変数名が短かければもっと保持できる。
- 変数の内容を保持するデータ領域
領域は動的に確保する。512 個の変数がすべて長さ 64 のリストで、その要素がすべて長さ 80 文字の文字列のとき、約 3 メガバイト使用する。
動的割付けなので、OS の swap が充分であれば無制限。関数 clear(); を実行すれば変数テーブルは再初期化され、小さくなる。
- extern 宣言
バージョン 9 からは、extern 宣言がない変数は内部変数となり、そのマクロの実行が終了すると消える。
メインのマクロとサブルーチンとして使用されるマクロ間で変数を共用するには、extern 宣言が必要になる。
マクロを調べてマクロ間で共有すべき変数、あるいはマクロの実行終了後も保持したい変数だけを extern 宣言する必要がある。面倒な場合は、マクロ内のすべての変数を extern 宣言するようにマクロを変更するプログラムを用意したのでそれを利用されたい。

```
prompt% mcnv9 マクロファイル名
マクロファイル名 : extern 宣言を追加したいマクロファイル名またはマクロライブラリファイル名 (*.MAC)
終了ステータス   : 正常終了なら 0、マクロに構文エラーなどがあると 1。
```

つぎのシェルスクリプトはディレクトリ内のすべてのマクロを一括して変換する。

```
#!/bin/sh
t="/tmp/mcnv9.$$~"
for i in `ls *.MAC`
```

```
do
  mcnv9 $i > $t
  if [ "$?" = "0" ]; then
    mv $i ${i}.OLD
    mv $t $i
    echo "MACRO $i modified"
  else
    echo "Error MACRO $i"
  fi
done
#
```

10.4 コマンドストリーム部

Advance CAD コマンドストリームは基本的にはキーボード入力と同じですが、注意すべき点は以下のとおりです。

- 全て大文字で書かなければならない。
- 文字列は必ずクォーテーションマーク (") で囲むこと。
たとえば下記のようにします。
GNT "text" @X0@Y0
- 座標
 - 1) '<x,y>' の形式
x および y は数値定数、変数、レジスタ参照、カリキュレータ呼出しのいずれでもかまわない。
 - 2) 'x,y' の形式
これは必ず座標イニシエータを用いること。
10,20 → @X10@Y20
- コマンドの終わりは '<CE>' または '<CR>' と記述する。
- カリキュレータ呼出し
マクロの中で変数の内容を取り出してコマンドストリームの一部としたいときには、次のように記述する。

[式]

まず式を計算し、その結果をコマンドストリームの一部としてそのままコマンド処理モジュールに渡す。式のところに変数名を書けば、その変数の内容をコマンドストリームの一部とすることができる。

コマンド入力時に計算式を使う場合に [式] という入力をするので、「カリキュレータ呼び出し」と呼ばれる。

例

```
:
r = 30;
:
RAD [r]
:
```

- マクロから 外部コマンドを実行する
マクロ中から外部コマンドを実行できる。
文字列の先頭の文字が ! であれば、それ以後の文字列は 外部コマンドとみなす。
外部コマンドには、dir 等のコマンドプロンプト上で動作する DOS コマンドやメモ帳 (notepad.exe) 等のアプリケーション、バッチ等が含まれる。
"! 外部コマンド "
"! 外部コマンド &"
"! 外部コマンド #"
 - 最後尾に & が記述されている場合は、バックグラウンドで起動する
コマンドプロンプトを起動せずに直接外部コマンドをバックグラウンドで起動する。アプリケーション起動完了後すぐに制御が戻る。それ以外の場合は、処理が終了し、外部コマンドが停止するまで制御は戻らない。

- 最後尾に # が記述されている場合は、コマンドプロンプトを起動せずに直接外部コマンドを起動する

コマンドプロンプト上に結果が表示される DOS コマンドの場合は、コマンドプロンプトが起動されていないと結果を確認することが出来ない。

逆にコマンドプロンプト上で入力を求める場合は、コマンドプロンプトが表示されないと、制御が戻らないため、ハングアップ状態になる。

```

"!DIR"      ... コマンドプロンプトに DIR の表示結果が表示される。
"!DIR &"    ... DIR の表示結果を表示すべき画面がないため、何も起こらず制御が戻る。
"!DIR #"    ... DIR の表示結果を表示すべき画面がないため、何も起こらず制御が戻る。
"!PAUSE"    ... コマンドプロンプトに 『続行するには何かキーを押してください . . .』の
               メッセージが表示され、キー入力で CAD に制御が戻る。
"!PAUSE &"  ... キー入力をすべき画面がないためプロセスは停止状態になるが、
               CAD に制御が戻る。
"!PAUSE #"  ... キー入力をすべき画面がないため CAD に制御が戻らず、
               コマンドプロンプトのプロセスを終了するまでハングアップ状態となる。

```

例えば、DIR /B B*.MDL > b_list.txt というコマンドを実行後、制御が戻ってから b_list.txt を処理する場合、& を使用してはいけな。これは、コマンドが完了する前に b_list.txt を参照してしまい、処理途中の未完成データを読み込む可能性があるため。

メモ帳を起動する場合

```

"!NOTEPAD " ... コマンドプロンプトが表示され、その中でメモ帳が起動される。
               CAD に制御が戻るのは、コマンドプロンプトが終了した時、
               コマンドプロンプトに制御が戻るのはメモ帳が終了した時
"!NOTEPAD &" ... メモ帳が起動され、即座に制御が戻る。
"!NOTEPAD #" ... メモ帳が起動され、CAD に制御が戻るのはメモ帳が終了した時。

```

注意

START コマンドの様にコマンドを起動させて自身は終了してしまうものもありますので、各コマンドの性質や用途を確認しながら使い分ける必要がある。

例

モデルファイルをクリップボードに登録する。
 "!DIR /B c:\acad\files*.MDL || findstr /E MDL | clip"

- マクロ内でのマクロの呼び出し

MACRO "name"

マクロからマクロを呼び出す。

マクロ呼出しの深さは 16 レベル以下。

再帰呼出しもできるが、無限呼出しにならないよう注意すること。

マクロの中でマクロ呼出しをする場合、呼出すマクロのコンパイルを失敗したらマクロの実行を停止する。呼出すマクロのコンパイルを失敗して実行を継続すると、意図しない結果になる可能性があるため。

- スイッチコマンド

GRID/SWITCH や RUBB/SWITCH などのステータス切り替えコマンドは現在値を反転する。現在値がオフならオン、オンならオフに切り替える。したがって、これらのコマンドの実行結果は現在値に依存する。現在値をオンにする、オフにするを、現在値に依存しないようにするには、たとえば GRID/SWITCH ならば GRID/ONN と GRID/OFF を使用する。

- パラメータ変更用のサブメニューを表示させたくないとき

いくつかのコマンドは、パラメータ変更のためのサブメニューを呼び出す。

マクロ内でこのコマンドを使用し、パラメータ変更用のサブメニューを表示させたくないときは、マクロ用のコマンド名をコマンド定義ファイルに登録しておくことで可能になる。コマンド定義ファイルにたとえば下記のように記述する。

```
[82, 5, 1]          !DRF/EDIT!
```

[82, 5, 1]

!DRFMODEMACRO!

- イベントデータのクリア

イベントデータとは、マウスのピック、タブレットのピックおよびキーボード入力データを指す。

モデル読み込みのように時間がかかるコマンドの次に、オペレータが入力操作をするコマンドが記述してあるマクロでは、モデル読み込み中に間違えてマウスを操作したりする (これがイベントデータ) と、それがつぎのコマンドの入力となってしまう、思わぬ結果になってしまう。

マクロ中に コマンド CLA/EVT を記述しておく、これを防ぐことができる。たとえば以下のようになる。

```
MODEL/READ "SAMPLE. MDL" <CE>
CLA/EVT
CCC [getpnt(2)] <CE>
```

- システムレジスタ

レジスタ参照 '#' に続いてレジスタ名。

レジスタ名は英大文字 (A ~ Z) で始まり英数字 (A ~ Z, 0 ~ 9) の並びとする。システムレジスタの一覧表を参照のこと。

システムレジスタは Advance CAD システムが保持している各種変数の現在値を参照するための機構である。たとえばクラス番号の現在値は #CURCLS というレジスタ名で参照する。レジスタ名は参照だけに用い、レジスタの値の更新はできない。
クラス番号を変更するには CLS コマンドを使用しなければならない。
レジスタ参照はコマンドストリーム部と言語部の両方に使用でき、どちらであるかは文脈による。

```
if (#CURCLS == 100) {
    CLS/MOD #CURCLS USEACT
}
c = #CURCLS;
```

このプログラムでは #CURCLS が 3 回現れる。最初は if 文の条件式の中なので、これは言語部として処理する。2 番目は CLS/MOD コマンドの引数なので、コマンドストリームの一部となる。3 番目は代入文で使用されているので言語部の一部である。

Advance CAD バージョン 4.4 から内容が「未定義」のシステムレジスタがある。

たとえばシステムレジスタ #MESX, #MESY は、MES/LOC コマンドなどにより値が設定されるまで内容は空で "未定義" となっている。内容が空のシステムレジスタをマクロ内で参照したときはマクロ変数タイプ 5 (<CE>) が得られる。

例

```
if (vtype(#MESX)==5) {
    echo "#MESX is empty" ; exit;
}
```

連番レジスタの添字

#MDLTTL001 ~ # MDLTTL207 などのような連番のレジスタは、添字を使ってアクセスできる。添字を使えるレジスタは、以下のとおり。

```
#CURAIM[i]      i=1 ~ 10
#SPCVER[i]       i=1 ~ 255
#MMGTTL[i]       i=1 ~ 202
#MMGMDL[i]       i=1 ~ 100
#MDLTTL[i]       i=1 ~ 207
#DRWTTL[i]       i=1 ~ 255
```

たとえばレジスタ #MDLTTL001 ~ #MDLTTL207 の内容を調べるとき、つぎのようにすればわかりやすく簡潔なマクロを書くことができる。

```
for (i=1; i<=207; i+=1) {
    v=vtype(#MDLTTL[i]);
}
```

(1) 現在値レジスタ

#CURX	点座標 X
#CURY	点座標 Y
#CURMAXID	データベースの最終アイテムのアイテム識別子
#CURIDPTR	選択されたアイテムのアイテム識別子
#CURACTCNT	アクティブリスト中のアイテム数
#CURRAD	半径値
#CURREV	レビジョン番号
#CURCLS	クラス番号
#CURLWT	線幅番号
#CURLFT	線種番号
#CURSLO	スクリーンレイアウト番号
#CURVP	ビューポート番号
#CURPIC	ピクチャ番号
#CURWIN	ウインド番号
#CURITMTYPE	アイテムタイプ番号
#CURFILNAME	モデルファイル名。入力された文字列のまま登録される。たとえば、下記のように test TEST.MDL products/test
#CURMDLNAME	モデル名。ディレクトリと拡張子を削除し、大文字にしたもの。たとえば下記のように TEST
#CURDRWSCF	ドローイング縮尺値
#CURPICSCF	現在のピクチャのピクチャ縮尺値
#CURAIMUSER	アカウントティングユーザ名
#CURAIMxx	アカウントティング情報。xx:#01 ~ #10
#CURYEAR	参照された年
#CURMONTH	参照された月
#CURDAY	参照された日
#CURHOUR	参照された時
#CURMINUTE	参照された分
#CURSECOND	参照された秒
#CURMINX	アクティブピクチャの表示範囲の左下座標 X
#CURMINY	アクティブピクチャの表示範囲の左下座標 Y
#CURMAXX	アクティブピクチャの表示範囲の右上座標 X
#CURMAXY	アクティブピクチャの表示範囲の右上座標 Y
#CURPNTMOD1	設定されているテンポラリポイントコマンドの番号 (6=TPON, 7=TPVR)
#CURPNTMOD2	設定されているテンポラリポイントコマンドの番号 (TPON, TPVR 以外)
#CURPNTANS	点が作成されたときのテンポラリポイントコマンドの番号。番号と内容は下記のとおり。 1 = TPND 2 = TPDG 3 = TPMD 4 = TPCR 5 = TPIN 6 = TPON 7 = TPVR 8 = TPAT 9 = TPSN 10 = TPDS 11 = TPOS
#CURPZNMIX	アクティブピクチャのピクチャゾーン左下 X

#CURPZNMINY アクティブピクチャのピクチャゾーン左下 Y
 #CURPZNMAXX アクティブピクチャのピクチャゾーン右上 X
 #CURPZNMXY アクティブピクチャのピクチャゾーン右上 Y
 #CURWCS アクティブピクチャの補助座標系が有効なとき "ON"、無効なとき "OFF"
 #CURWCSORGX 現在のアクティブピクチャの補助座標原点 X
 #CURWCSORGY 現在のアクティブピクチャの補助座標原点 Y
 #CURWCSVECC 現在のアクティブピクチャの補助座標 X 軸コサイン成分
 #CURWCSVECS 現在のアクティブピクチャの補助座標 X 軸サイン成分
 #CURLOGUSER ログインユーザ名
 #CURTXT 最後に入力された文字列。最後に指定された次の3つのうちのいずれかが格納される。

- キーボードから入力された文字列
- ファイル一覧によって選択されたファイル名
- TLB コマンドによって選択された文字列

たとえばマクロ内で TLB コマンドによって選択された文字列を取り出すときは、つぎのようにする。

```
GEOM /* TLB で選択された文字列を受けるための何もしないコマンド */
fname = "TLB-FILE-NAME";
TLB [fname]
for (;;) {
  p = getpos(1);
  [p]
  if (vtype(p) == 5) { break; }
}
if (#CURTXT != fname) {
  txt = #CURTXT; /* 選択された文字列 */
} else {
  /* 選択されなかった場合の処理 */
}
```

#CURIDTNPNT 「複数アイテムの自動選択」中の入力点数
 0 : アイテムが選択され自動選択機能を終えた。または、アイテムが
 選択できずに自動選択機能を終えた。
 1 : 1点目でアイテムが選択できず、2点目の入力待ち状態。
 2 以上: 1点目と2点目の矩形でアイテムが選択できず、多角形領域の入
 力待ち。
 #CURCCDNAME 同時設計。現在処理中のペアレント名。
 TON_START、TON_CALL、TON_MDL_READ で指定されたペアレント名を持つ。同時設計
 作業中でないときは未定義になる。
 #CURCCDMEMB 同時設計。現在実データ化されているメンバーの名前。
 実データ化されているメンバーがないときは未定義になる。
 #CURCMDNAME 実行中のコマンド名。
 たとえば2点間線ならば "LBP" となる。
 #CURMDLVER 処理中のモデルファイルのバージョン番号。
 このレジスタが参照されるたびディスク上のモデルファイルからバージョン番号を
 抽出する。モデルを呼び出したときのモデルファイルのバージョンをメモリ内に保
 持しているのではないことに注意。
 モデルファイルのバージョン番号と Advance CAD のバージョンの一覧

レジスタの値	Advance CAD のバージョン
1 ~ 8	各々バージョン 1 ~ 8
9	9 , 10
11	11
12	12 , 13
14	14

レジスタの値	Advance CAD のバージョン
15	15
16	16 , 17
18	18
19	19
20	20
0	モデル名が設定されていない。またはモデルファイルが存在しない。

(2) メジャーレジスタ

#MESDST	距離
#MESANG	角度
#MESLEN	長さ
#MESRAD	半径
#MESX	点座標 X
#MESY	点座標 Y
#MESAREA	面積
#MESIX	X 軸に関する断面二次モーメント
#MESIY	Y 軸に関する断面二次モーメント
#MESIXY	断面相乗モーメント
#MESVOL	体積

(3) 関係（アソシエイト）アイテムのレジスタ

#ASCVERNAME	アソシエイト名
#ASCVERFILE	サブモデル名
#ASCVERPIC	配置ピクチャ番号
#ASCVERX	配置位置 X
#ASCVERY	配置位置 Y
#ASCVERZ	配置位置 Z
#ASCVERANG	配置角度（度）

(4) ベリファイレジスタ

#VERIDPTR	アイテム識別子
#VERVP	ビューポート番号
#VERPIC	ピクチャ番号
#VERCLS	クラス番号
#VERREV	レビジョン番号
#VERLWT	線幅
#VERLFT	線種
#VERXS	始点 X
#VERYS	始点 Y
#VERXE	終点 X
#VERYE	終点 Y
#VERXC	円の中心点 X
#VERYC	円の中心点 Y
#VERRAD	円の半径
#VERLEN	長さ
#VERANG	角度（度）
#VERTXT	文字列
#VERTXTSIZ	文字の大きさ
#VERTXTANG	文字の配置角度（度）
#VERTXTAFT	文字の ASCII フォント番号
#VERTXTKFT	文字列の漢字テキストフォント番号
#VERTXTSLT	文字の傾斜角度（度）
#VERTXTRAT	文字縦横比率

#VERTXTALN	行幅整列処理係数
#VERTXTORI	文字列の向き（横書き、縦書き）
#VERTXTBOXW	文字列の枠（箱）の幅
#VERTXTBOXH	文字列の枠（箱）の高さ
#VERGAPHOR	水平方向の文字間隔
#VERGAPVRT	垂直方向の文字間隔
#VERORGHOR	水平方向の文字原点位置コード
#VERORGVRT	垂直方向の文字原点位置コード
#VERJSTHOR	文字列表示 水平基準
#VERBOXMOD	文字枠表示モード
#VERSPCHOR	文字枠の水平方向のゆとりサイズ
#VERSPCVRT	文字枠の垂直方向のゆとりサイズ
#VERMRKNUM	マーク番号
#VERMRKSIZ	マークの大きさ
#VERMRKANG	マークの配置角度
#VERFILNAME	サブモデル名／シンボル名／APG 名
#VERXSCF	縮尺 X
#VERYSCF	縮尺 Y
#VERXMIR	反転 X
#VERYMIR	反転 Y
#VERORGPIC	サブモデルの元のピクチャ番号
#VERITMTYPE	アイテムタイプ番号。ベリファイアイテム（VER）およびベリファイセグメント（VER/SEG）のときに登録される。

番号	アイテム
1	点 (Point)
2	直線 (Line)
3	円／円弧 (Circular Arc)
4	自由曲線 (Free curve)
5	ストリングアイテム (String item)
9	複合アイテム (Composit item)
11	ジェネラルテキスト (General note, General label, Reference note, Reference label, cutting plane line)
12	マーク (General mark)
13	寸法 (Dimension)
14	幾何公差 (Geometrical Tolerance)
15	ハッチングアイテム (hatching item)
16	塗り潰しアイテム (area fill item)
28	APG アイテム
30	シンボル (Symbol)
31	サブモデル (Submodel)

#VERSRTYPE サブレコードタイプ番号。ベリファイセグメント (VER/SEG) のときに登録される。

番号	サブレコード
2	点

番号	サブレコード
3	始点
4	線分
5	円、円弧
6	3 次 Bezier 曲線
33	文字列
34	マーク

#VERITMNAME	アイテム名
#VERDIMTYPE	寸法線のタイプ番号 寸法種類 * 10 + 副分類コード 寸法種類 1 = 長さ寸法 2 = 角度寸法 3 = 半径寸法 4 = 直径寸法 5 = 座標寸法 6 = 円弧長寸法 7 = 面取り寸法 副分類コード 0 = 単一寸法 1 = 直列寸法 2 = 並列寸法 3 = 累進寸法 (水平、垂直) 4 = 片側寸法 5 = 累進寸法 (累進水平、累進垂直、累進角度) 6 = 寸法補助線が放射状の円弧長寸法 実際に使用されている副分類コード 長さ寸法 0 1 2 3 4 5 角度寸法 0 1 2 5 半径寸法 0 直径寸法 0 座標寸法 0 円弧長寸法 0 6 面取り寸法 0
#VERMINX	アイテムを囲む最小矩形の左下座標 X
#VERMINY	アイテムを囲む最小矩形の左下座標 Y
#VERMAXX	アイテムを囲む最小矩形の右上座標 X
#VERMAXY	アイテムを囲む最小矩形の右上座標 Y
#VERAFLPAT	エリアフィルアイテムのパターン番号
#VERXHTGAP	ハッチングの間隔
#VERXHTMOD	平行線の場合は 0, 格子の場合は 1
#VERXHTPAT	ハッチングのパターン番号。パターンを使用しないときは未定義
#VERXHTLIN	表示ライン数。パターン使用時は登録ライン数。
#VERXHTOFF	非表示ライン数。ハッチングの通過点 X, Y およびハッチング角度は、#VERXC, #VERYC, #VERANG を使用する。
#VERDIMTOL	寸法の上下公差 (数値型 または 文字列型)
#VERDIMTOLU	寸法の上公差 (数値型 または 文字列型)
#VERDIMTOLL	寸法の下公差 (数値型 または 文字列型) 公差として数値以外が指定されていると文字列型になる。
#VERPLWT	部分線幅番号。部分線幅が指定されていないときはアイテムの線幅番号。ベリファイセグメント (VER/SEG) のときに登録される。
#VERPLFT	部分線種番号。部分線種が指定されていないときはアイテムの線種番号。ベリファイセグメント (VER/SEG) のときに登録される。

- | | |
|------------|--|
| #VERPICREF | ピクチャ参照サブモデルのときは 1、ピクチャ参照サブモデルでないときは 0。 |
| #VERDIMSCF | ベリファイ (VER) でサブモデルが選択されたときに登録される。 |
| #VERTXTLNG | 寸法値の倍率。ベリファイ (VER) で寸法が選択されたときに登録される。 |
| | 文字列の長さ。 |
- (5) 特性データベリファイのレジスタ
- | | |
|-------------|--|
| #SPCVERFILE | ファイル番号 |
| #SPCVERREC | カテゴリ番号。同一カテゴリの特性が複数入力されている場合は、その数を負の数でセットする。 |
| #SPCVERxxx | 特性データ。xxx: #001 ~ #255 |
- (6) モデル検索のレジスタ (MDLDSP)
- | | |
|-------------|-------------------------------------|
| #MMGMDLNAME | 選択されたモデルの名前 |
| #MMGMDLPATH | 選択されたモデルのフルパス名 |
| #MMGMDLMGNO | 選択されたモデルの管理番号 |
| #MMGREAD | 選択されたモデルの読み込み属性 (0: 可, 1: 不可) |
| #MMGWRITE | 選択されたモデルの書き込み属性 (0: 可, 1: 不可) |
| #MMGOWNER | 選択されたモデルの所有者名の有無 (0: 無, 1: 有) |
| #MMGDATE | 選択されたモデルの作成日 |
| #MMGUPDATE | 選択されたモデルの更新日 |
| #MMGOLDMDL | 選択されたモデルの .OLD ファイルの有無 (0: 無, 1: 有) |
| #MMGCASSET | 選択されたモデルのカセット名 |
| #MMGTTLxxx | 選択されたモデルのモデルタイトル。xxx : #001 ~ #202 |
| #MMGMDLCNT | 検索で得られたモデル名の数 |
| #MMGMDLxxx | 検索で得られたモデル名。xxx : #001 ~ #100 |
- (7) モデルタイトルのレジスタ (MODEL/TITLE)
- | | |
|------------|--------------------------|
| #MDLTTLxxx | モデルタイトル。xxx: #001 ~ #209 |
|------------|--------------------------|
- (8) パーツのレジスタ (PARTS, PVER)
- | | |
|-----------|--------------------------------|
| #PRTCTG | パーツの種別番号 (1= パーツ、2= パーツセット) |
| #PRTIDPRT | パーツで使用されたアソシエイトの識別子 |
| #PRTNMPRT | パーツ名 |
| #PRTSZPRT | パーツのサイズ名 |
| #PRTIDSET | パーツセットで使用されたアソシエイトの識別子 |
| #PRTNMSET | パーツセット名 |
| #PRTSZSET | パーツセットのサイズ名 |
| #PRTIDAPG | 最後に配置された APG ファイルに付くアソシエイトの識別子 |
- (9) 図面配置関係のレジスタ
- 最後に配置または表示された図面配置頁の内容を保持する。まだ図面配置されていなければ、値は入らない。
- | | |
|-------------|--|
| #DRWLASTPAG | 最後に配置または表示された図面配置頁 |
| #DRWTMPNAME | 図面枠シンボル名。サイズ指定の時はセットされない。 |
| #DRWTMPISZX | サイズ指定のときの横方向の大きさ。図面枠シンボル指定の時はセットされない。 |
| #DRWTMPISZY | サイズ指定のときの縦方向の大きさ。図面枠シンボル指定の時はセットされない。 |
| #DRWPICCNT | 配置されているウインドウ数。このレジスタを参照すると、レジスタ #DRWPICNUM と #DRWWINNUM は最初に配置されたウインドウのピクチャの番号とウインドウ番号を返す。 |
| #DRWPICNUM | 配置されているウインドウのピクチャ番号。 |
| #DRWWINNUM | 配置されているウインドウのウインドウ番号。ウインドウが複数配置されている場合は、このレジスタを参照するたびに #DRWPICNUM と #DRWWINNUM は次に配置されたウインドウのピクチャ番号とウインドウ番号を返すようになる。 |
| #DRWTTLxxx | 頁タイトル。xxx: #001 ~ #255 |
| #DRWPICPSCF | ピクチャ #DRWPICNUM のピクチャ縮尺値 |
| #DRWPICLSCF | ピクチャ #DRWPICNUM, ウインドウ #DRWWINNUM の配置時に指定された倍率 |
| #DRWPICANG | ピクチャ #DRWPICNUM, ウインドウ #DRWWINNUM の配置角度 |
| #DRWPICX | ピクチャ #DRWPICNUM, ウインドウ #DRWWINNUM の配置位置 X (図面配置座標系) |
| #DRWPICY | ピクチャ #DRWPICNUM, ウインドウ #DRWWINNUM の配置位置 Y (図面配置座標系) |

#DRWPICORGX	ピクチャ #DRWPICNUM, ウィンドウ #DRWWINNUM のウィンドウ原点 X (モデル座標系)
#DRWPICORGY	ピクチャ #DRWPICNUM, ウィンドウ #DRWWINNUM のウィンドウ原点 Y (モデル座標系)
#DRWPICMINX	ピクチャ #DRWPICNUM, ウィンドウ #DRWWINNUM のウィンドウ左下 X (モデル座標系)
#DRWPICMINY	ピクチャ #DRWPICNUM, ウィンドウ #DRWWINNUM のウィンドウ左下 Y (モデル座標系)
#DRWPICSIZEX	ピクチャ #DRWPICNUM, ウィンドウ #DRWWINNUM のウィンドウの大きさ X (幅) (モデル座標系)
#DRWPICSIZY	ピクチャ #DRWPICNUM, ウィンドウ #DRWWINNUM のウィンドウの大きさ Y (高さ) (モデル座標系)
#DRWPICANGW	ピクチャ #DRWPICNUM, ウィンドウ #DRWWINNUM の角度 (度)

(10) 配置したサブモデルのレジスタ

#SUBNAME	配置したサブモデル名
#SUBPICNUM	配置サブモデルのピクチャ番号
#SUBPICNAME	配置サブモデルのピクチャ名
#SUBANG	配置角度
#SUBXSCF	配置 X スケール
#SUBYSCF	配置 Y スケール
#SUBXMIR	配置 X ミラー
#SUBYMIR	配置 Y ミラー
#SUBX	配置 X 位置
#SUBY	配置 Y 位置

10.5 マクロの実行

10.5.1 実行コマンド

Advance CAD 上で次のようにマクロコマンドでマクロを起動し、実行させます。

- NEW** : マクロを再コンパイルしてから実行する。省略時すると、そのときの状態によってコンパイルするかしないかが自動的に選択される。つまり、実行するマクロのコンパイル結果が保存されていればコンパイルしないで実行し、それがなければコンパイルしてから実行する。NEW を指定すると、必ず再コンパイルしてから実行するので、常に最新のマクロソースファイルを実行できる。
- マクロ名** : マクロ名、またはマクロライブラリ中のマクロエレメント名。ディレクトリ名およびファイル拡張子は省略してよい。マクロ名にワイルドカードを使ってマクロファイル名の一覧表を表示し、ファイル名をピックアップして実行させることもできる。

Advance CAD はマクロをコンパイルしてから実行します。Advance CAD はしばらくの間コンパイル結果を保持しているので、次に同じマクロを実行するときはコンパイルなしで実行するため、効率的です。しかしマクロのデバッグなどでマクロソースファイルを修正している時は、マクロを再コンパイルさせるためにマクロのクリア (CLA/MAC) コマンドを実行しないとマクロソースファイルの修正が反映されません。モディファイヤ NEW を使用して MACRO NEW name とすれば必ず再コンパイルされますので、デバッグ時に便利です。

NEW のもうひとつの用途を紹介します。マクロの中からフォアグラウンドでユーザプログラムを実行し、そのプログラムが正常終了したかしないかを知りたいことがあります。たとえばつぎのマクロを考えてみましょう。

```
"!userprog"
MACRO NEW "USTAT"
if (ustat != 0) {
    echo "error in userprog, Err=" ustat;
    exit;
}
```

この例ではマクロからプログラム userprog を実行します。このプログラムはフォアグラウンドで実行していますので、この実行が完了するまで Advance CAD は停止しています。プログラム userprog の実行が終了するとマクロの残りが実行されます。このときプログラム userprog はプログラムが正常終了したかしないかをマクロで調べられるように、マクロファイル USTAT.MAC を生成して終了するようにします。

USTAT.MAC は次のような内容とします。

```
extern ustat=0;    あるいは
extern ustat=1;
```

このようにすると、マクロ USTAT が実行され、変数 ustat が設定されます。この変数はユーザプログラム userprog の実行結果を表わしています。

マクロ "UTEST" を実行するとき、モディファイヤ NEW を指示するのは、最新の UTEST.MAC が実行されることを保証するためです。

ひとつのマクロがきわめて長いとき、あるいは長いマクロをネストして呼出したときなど、どうしてもフリースペースが確保できない場合、マクロは強制終了させられます。また、マクロテーブルは空になります。

マクロの実行を強制終了したいときは ATTENTION キーを押します。

MACRO というコマンド名を入力しないで、マクロ名入力だけでマクロを起動することもできます。バージョン 7 までは、モデル定数のコマンド MACRO/DIRECT で 1 を指定することにより、マクロ名直接入力を可能にしていました。ただしこの場合、文字列がマクロファイル名と一致してしまうとマクロを実行してしまうことがありました。このため、バージョン 8 からはこれを廃止し、メニューファイルの USERCMD.MEN にマクロ名を登録しておき、登録されたマクロだけを直接入力可能にしました。USERCMD.MEN への登録はコマンド名定義で行います。ディスパッチャ番号 32、ドライバ番号 0、フォーム番号 +n とし、マクロ名をコマンド名フォーマットで記述します。以下は記述例です。

```
Command
+ [32, 0, 1] !USERMAC01!
+ [32, 0, 2] !USERMAC02!
+ [32, 0, 3] !USERMAC03!
```

10.5.2 ダイアログボックス

Windows 版ではモデル呼出しや定数設定などいくつかのコマンドでダイアログボックスが開きます。ダイアログボックスが開くとそれを閉じるまでコマンド入力などの他の操作はできません。マクロもダイアログボックスが閉じるまで先に進みません。

マクロでは目的によってダイアログボックスを開きたい場合と開きたくない場合があります。

ダイアログボックスを開きたい場合はつぎのように記述します。

```
DLGMODE 1 RVP/DRF
(ここで製図定数設定のダイアログボックスが開く。定数を設定してダイアログボックスを閉じると以下のマクロが実行される。)
```

```
GNT "Text" <0, 0> <CE>
```

ダイアログボックスを開かないで文字高さを 5mm に設定して注記を記入するマクロはつぎのように記述します。この場合はサブウィンドウが表示されますがダイアログボックスと違ってマクロの実行は停止しません。

```
DLGMODE 0 RVP/DRF TEXT/SIZE 5 <CE>
GNT "Text" <0, 0> <CE>
```

DLGMODE はダイアログボックスを使用するか使用しないかを設定するモデル定数です。(詳しくはコマンドリファレンスマニュアルの「モデル定数」DLGMODE をご覧下さい。)

10.5.3 図形表示を抑止する

マクロの処理時間を短縮するために、PIC コマンドでピクチャを切り替えたときや ZOOM/ALL で全体表示にしたときに図形表示を抑止することができます。

```
RPT/OFF      : ピクチャの図形要素や図面配置状態の表示をスキップする。
RPT/ONN     : 表示をスキップしない。(省略時)
```

マクロを終了したときには RPT/ONN の状態に戻る。

10.5.4 情報表示

マクロ内で注記の文字高さを設定するときやモデルタイトルを設定するときにサブウィンドウやダイアログボックスへの内容表示をしたくない場合があります。
このような表示を抑止するためのコマンドをマクロ内に記述することができます。

LIST/ONN : マクロ実行中でも情報を表示する（省略時）
LIST/OFF : マクロ実行中の情報表示をしない

マクロを終了したときには LIST/ONN の状態に戻る。

製図定数の RVP メニューまたはダイアログボックスを表示しないで文字高さを 5mm に設定して注記を記入するマクロはつぎのように記述します。

```
LIST/OFF RVP/DRF TEXT/SIZE 5 <CE>
GNT "Text" <0, 0> <CE>
```

LIST/OFF に対応しているコマンドは以下のコマンドです。

- 定数設定
- 製図一括修正
- 特性データ
- モデルタイトル
- モデル検索
- ペリファイ

10.5.5 メニュー／メッセージの表示

マクロ内でコマンドを切り替えるとそのコマンドに対応したメニュー／メッセージが表示されます。まとまった処理をマクロで実行する場合などはメニュー／メッセージが不要な場合があります。メニュー／メッセージの表示を抑止するためのコマンドをマクロ内に記述することができます。

MENU/ONN : マクロ実行中でもメニュー／メッセージを表示する（省略時）
MENU/OFF : マクロ実行中のメニュー／メッセージ表示をしない

マクロを終了したときには MENU/ONN の状態に戻る。

例. 現ピクチャのアイテムで始点が 0, 0 点のアイテムをピクチャ 2 にコピーする。

```
MENU/OFF
LIST/OFF
echo " マクロを実行中です ";
ZOOM/ALL
ACT/ALL <CE>
COPY OUTPIC 2 /* コピーコマンドを起動する */
error(); /* 以下でエラー判定をするので以前のエラー状態をクリアしておく */
n = #CURACTCNT;
for (i = 1; i <= n; i += 1) {
    VER [actlst(i)] <CE>
    if (error()) { continue; }
    x = #VERXS;
    y = #VERYS;
    if (vtype(x) != 3 || vtype(y) != 3) { continue; }
    if (x == 0.0 && y == 0.0) {
        [actlst(i)] /* コピーコマンドにアイテム識別子を渡す */
    }
}
```

```
<CE> /* コピーコマンドを実行する */
MAIN /* 何もしないコマンドを選択する */
```

10.5.6 ダイアログモードを切り替え

ダイアログモード設定コマンドでオン・オフを切り替えます。
このコマンドはマクロ内から呼び出します。

```
DLG/ONN      : マクロ実行中でダイアログモードをオンにする。
DLG/OFF      : マクロ実行中でダイアログモードをオフにする。
```

例. SYM/DSP の前後でダイアログモードのオン・オフを切り替えます。

```
T <4, 1> " 弁表示 "
!PID_CAT 2 PID_SYM DLG/OFF SYM/DSP DLG/ONN ACTPLN "V_*"!
```

```
ACADCMD.MEN
以下の行を追加
V [98, 6, 9] !DLG/ONN!
V [98, 6, 10] !DLG/OFF!
```

10.5.7 マクロの実行結果を 1 度のアンドゥで元に戻す

<CE> 入力時のアンドゥブロックカウン트의自動更新を中止します。
このコマンドはマクロ内から呼び出します。

```
UNDO/OFF      : このコマンドが呼び出された後は<CE>によるアンドゥブロックカウン트의
                  自動更新は行いません。
                  この設定は UNDO/ONN コマンドが呼び出された場合、またはマクロの実行
                  を終えた場合に解除します。
UNDO/ONN      : 現在の位置でアンドゥブロックカウン트를更新します。
                  このコマンドが呼び出された後は<CE>によるアンドゥブロックカウン트의
                  自動更新を行います。(通常の状態)
```

```
ACADCMD.MEN に、以下の行を追加
V [98, 6, 11] !UNDO/ONN!
V [98, 6, 12] !UNDO/OFF!
```

10.5.8 アテンションキー入力を無効にする

アテンションキー入力でマクロの実行を終了しないようにします。
このコマンドはマクロ内から呼び出します。

```
ATTN/OFF      : このコマンドが呼び出された後はアテンションキー入力を無効にします。
                  この設定は ATTN/ONN コマンドが呼び出された場合、またはマクロの実行
                  この設定は UNDO/ONN コマンドが呼び出された場合、またはマクロの実行
                  を終えた場合に解除します。
ATTN/ONN      : このコマンドが呼び出された後はアテンションキー入力を有効にします。
                  (通常の状態)
```

```
ACADCMD.MEN に、以下の行を追加
V [98, 6, 13] !ATTN/ONN!
```

10.5.9 クリップボード内文字列操作マクロ関数

クリップボード内の文字列を抽出するマクロ関数です。

● 使用方法

1) クリップボード内の内容を抽出する関数

```
txt = getclp()
txt: クリップボード内の文字列
      クリップボード内の文字列を返す。
```

1) クリップボードに文字列を設定する関数

```
s = setclp(txt)
txt: クリップボードに設定する文字列
s : 設定した文字列数
```

10.5.10 マクロのデバッグ

Advance CAD の外、つまり OS のコマンド受付けレベルで、Advance CAD のマクロのシンタックスチェックを行えます。これにはスタンドアロンのマクロチェッカープログラム (mcheck) を使用します。このプログラムは Advance CAD のマクロソースファイル (.MAC) を読み、コンパイルを行いますが、実行はしません。それ以外は Advance CAD のコマンド MACRO とまったく同じです。マクロのデバッグに便利です。

● 使用方法

(1) コマンド構文 【ユーティリティ名】

mcheck

【構文】

```
mcheck -pCONFIGFILE
```

-pCONFIGFILE : コンフィグレーションファイル名を指定する。省略時は ACAD.SET とみなされる。
Advance CAD コマンド名またはモディファイヤ名が正しいかどうかチェックするため、メニューファイルを参照する。そのため、メニューファイルのありかを記述したコンフィグレーションファイルが必要である。

(2) 実行

```
% mcheck <CR>
Enter macro source name :
```

マクロソースファイル名を入力する。ファイル拡張子をつける必要はない。

また、デフォルトディレクトリはコンフィグレーションファイルのキーワード #MACRO# で指定したディレクトリになる。

マクロライブラリの中のマクロにコンパイルできる。マクロライブラリ ACADUSR.MAC の中のマクロ U001 をコンパイルしたいときは、"USR\$U001" と入力する。

(3) マクロ コンパイルリスト

Advance CAD マクロコンパイラおよびマクロチェッカー (mcheck) のエラー、Advance CAD のマクロ実行時のエラーは、ファイル 'macro.lis' に出力する。

このファイルは Advance CAD, mcheck の起動ディレクトリに作成される。

Advance CAD 実行時に mcheck を使用するときは、Advance CAD の起動ディレクトリ以外で実行すること。たとえば /acad/macro ディレクトリなどで実行する。そうしないと macro.lis の内容がおかしくなるか、Advance CAD の実行がおかしくなる。

(4) エラーメッセージのリスト

マクロのエラーメッセージは、ファイル /acad/msg/ERR22.TXT に記述してある。

エラー番号とメッセージの対応は以下のとおり。

```

エラー番号 (メッセージ番号) "メッセージ"
/
/Macro compilation error
01      (2200001) "指定されたマクロソースが見つからない"
02      (2200002) "HELP モード。マクロの実行はできません"
03      (2200003) "マクロファイル読み込みエラー"
/
11      (2200011) "マクロエントリフル"
12      (2200012) "マクロ名テーブルフル"
13      (2200013) "マクロ格納領域が確保できませんでした"
14      (2200014) "実行中のマクロはコンパイルできません"
15      (2200015) "ループ、条件文の入れ子制限をこえました"
16      (2200016) "/* と */ が合ってません"
17      (2200017) "ラベルの数が多すぎます"
18      (2200018) "マクロのネスト制限を超えました"
/
21      (2200021) "不正な識別子"
22      (2200022) "不正な数値定数"
23      (2200023) "不正な文字定数"
24      (2200024) "不正な文字を使用またはトークンが不正な位置にある"
25      (2200025) "@ の後に不正な文字がある"
26      (2200026) "< の後に不正な文字がある"
/
      (2200027)
/
      (2200028)
29      (2200029) "不正な計算式"
30      (2200030) "コマンド名の誤り"
/
31      (2200031) "extern の外部変数並びの誤り"
32      (2200032) "if, while, for, switch の次に条件式がない"
33      (2200033) "if, while, do, for, switch の { がない"
34      (2200034) "if に対応しない else がある"
35      (2200035) "対応しない } がある"
36      (2200036) "else の次に if, { 以外がある"
37      (2200037) "break, continue 文がループの外"
38      (2200038) "goto の次にラベル名がない"
39      (2200039) "ラベル名の重複宣言"
40      (2200049) "for 文の条件式が誤り (init;cond;reinit)"
41      (2200041) "if, switch またはループが閉じていない"
42      (2200042) "未定義ラベル参照"
43      (2200043) "空マクロ"
44      (2200044) "switch 文の case, default が不正"

```

```

45      (2200045) " 文の最後に ; がない "
46      (2200046) "while, for, switch の次の条件式が不正 "
/
/ Macro run time error
101     (2200051) " 異常な命令検出 "
102     (2200052) " 空文字列 "
103     (2200053) " マクロ実行強制終了 "
104     (2200054) " 座標 <x, y> @Xs@Ys または、数値 @DSs が計算できなかった "
/
/ Calculator error
201     (2200061) " 計算式が誤り "
202     (2200062) " 使用できない文字がある "
203     (2200063) " 計算式のなかに予約語がある "
204     (2200064) " 外部変数名を二重宣言、または変数名が内部変数名である "
205     (2200065) " 未定義レジスタ参照 "
206     (2200066) " 変数格納領域が確保できませんでした "
207     (2200067) " 代入の左辺が変数でない "
208     (2200068) " 未定義変数参照 "
209     (2200069) " この関数は計算器では実行できません "
210     (2200070) " 変数、定数があるべきなのに無い "
211     (2200071) " 括弧が対応していない "
212     (2200072) " 未定義関数参照 "
213     (2200073) " 引数の数が不適当 "
214     (2200074) " 引数の値が不適当 "
215     (2200075) " ファイルがオープンされてない "
216     (2200076) " 条件演算子の ? と : が対応していない "
217     (2200077) " 同時にオープンできるファイル数の制限をこえた "
218     (2200078) " 未定義演算子 "
219     (2200079) " 変数、引数の型が不適当 "
      (2200080)
221     (2200081) " 演算子がない "
222     (2200082) " 添字の誤り "
223     (2200083) " 添字が範囲外 "

```

10.5.11 マクロライブラリ

現在、メニューソースファイル内に直接マクロテキストを記述できますが、マクロテキストの文字数は63文字以下で簡単なものしか記述できません。

```
T <1, 1> "LINE" !LBP [getpnt (2)]!
```

次のようにマクロを実行するようにすれば、文字長さの制限は回避できます。

```
T <1, 1> "LINE" !MACRO "MLINE"!
```

しかしこの方法では、メニューファイルとメニューで使用するいくつかのマクロファイルを管理しなければなりません。管理を簡単にするため、いくつかのマクロファイルをひとつのマクロファイルにまとめることができるようにしました。これをマクロライブラリファイルとよびます。

● マクロライブラリファイル名の規約

ACAD%%%.MAC %%% は任意の3文字 (英大文字)。

システムライブラリ ACADSYS.MAC

● マクロライブラリファイル

複数のマクロを記述する。

マクロの開始には次のコメント行をおくことによりマクロを区切る。

```
/* Macro name = name */
```

ここで = の次の name はマクロ名。大文字の英数字のみ。下線部は固定。

以下は例。

```
/* Macro name = M001 */
マクロ M001 の記述
/* Macro name = M002 */
マクロ M002 の記述
```

● マクロの実行

ライブラリのエレメントと単純マクロファイルの区別はつぎのようにする。
マクロ名はライブラリファイルの名前の一部を引用して '%%%' で始める。

```
MACRO "%%$name"      ACAD%%.MAC のエレメント name

MACRO "SYS$M001"      システムライブラリ ACADSYS.MAC のエレメント M001
MACRO "USR$U001"      ACADUSR.MAC のエレメント U001
```

それ以外のときは単純マクロファイルとみなす。

```
MACRO "EXAMPLE" EXAMPLE.MAC
```

40.5.12 マクロライブラリ作成ツール (参考:UNIX 版のみ)

Advance CAD バージョン 6 から、マクロファイルをひとつにまとめたマクロライブラリファイルをサポートしています。

マクロライブラリファイルはテキストエディタを使って作成できます。

テキストエディタを使用しないでマクロライブラリにマクロを追加したり、削除したりするシェルスクリプトをご紹介します。このシェルスクリプトを使用すれば、簡単でしかも編集の誤りを防ぐことができます。参考にしてください。まず具体的な使い方から説明します。

後述のリストのシェルスクリプトをファイル `mar` に、`awk` プログラムをファイル `mar.awk` に作ります。ファイル `mar` に実行可能属性 `x` を付けます。

```
etc %1 chmod +x mar
```

引数なしで実行すると、シェルスクリプトの使い方を表示して終了します。

```
etc %2 mar
Usage of Advance CAD macro archiver
List macro elements : mar t lib.MAC
Replace(Add) elements : mar r lib.MAC macro.MAC ...
Delete elements : mar d lib.MAC element_name ...
```

このシェルスクリプトは 3 つの機能を持っています。最初の引数で機能を指定します。

```
t      マクロライブラリ中のマクロをリストします。
r      マクロライブラリにマクロを追加します。すでに同じ名前のマクロがあれば更新します。
d      マクロライブラリからマクロエレメントを削除します。
```

2 番目の引数はマクロライブラリ名です。3 番目以降の引数は、`r` ではマクロライブラリに挿入するマクロファイル名、`d` では削除するマクロエレメント名です。

テキストライブラリ名は `ACADXXX.MAC` とし、マクロファイル `DEB1.MAC`, `DEB2.MAC`, ..., `DEB9.MAC` があるものとします。`ACADXXX.MAC` を除去し、何も無いところから開始しましょう。まずテキストライブラリ `ACADXXX.MAC` にマクロ `DEB1.MAC` ~ `DEB9.MAC` を追加します。ファイル `ACADXXX.MAC` は新しく作成されます。

```
etc %3 rm ACADXXX.MAC
etc %4 mar r ACADXXX.MAC DEB[1-9].MAC
Module DEB1 insert
Module DEB2 insert
Module DEB3 insert
```

```
Module-DEB4-insert
Module-DEB5-insert
Module-DEB6-insert
Module-DEB7-insert
Module-DEB8-insert
Module-DEB9-insert
```

マクロライブラリ内のマクロをリストするには、つぎのようにします。

```
etc %5 mar t ACADXXX.MAC
/* Macro name = DEB1 */
/* Macro name = DEB2 */
/* Macro name = DEB3 */
/* Macro name = DEB4 */
/* Macro name = DEB5 */
/* Macro name = DEB6 */
/* Macro name = DEB7 */
/* Macro name = DEB8 */
/* Macro name = DEB9 */
```

つぎにマクロライブラリから ~~DEB1, DEB3, ..., DEB9~~ の5つのマクロエレメントを削除してみます。マクロライブラリ内のマクロは拡張子なしです。コマンドラインに ~~DEB[13579]~~ のような正規表現は使用できないことに注意してください。

```
etc %6 mar d ACADXXX.MAC DEB1 DEB3 DEB5 DEB7 DEB9
Old module DEB1 removed
Old module DEB3 removed
Old module DEB5 removed
Old module DEB7 removed
Old module DEB9 removed
```

マクロライブラリをリストし、確認します。

```
etc %7 mar t ACADXXX.MAC
/* Macro name = DEB2 */
/* Macro name = DEB4 */
/* Macro name = DEB6 */
/* Macro name = DEB8 */
```

こんどはマクロ ~~DEB4.MAC~~ と ~~DEB5.MAC~~ を追加します。~~DEB4~~ はすでにマクロライブラリ内にあるので、新しい ~~DEB4~~ を挿入し、古い ~~DEB4~~ を削除します。

```
etc %8 mar r ACADXXX.MAC DEB[45].MAC
Module-DEB4-insert
Module-DEB5-insert
Old module-DEB4-removed
```

もう一度リストします。最後に挿入したマクロが先頭にきています。

```
etc %9 mar t ACADXXX.MAC
/* Macro name = DEB4 */
/* Macro name = DEB5 */
/* Macro name = DEB2 */
/* Macro name = DEB6 */
/* Macro name = DEB8 */
```

これでシェルスクリプト ~~mar~~ の3つの機能の使い方を説明しました。以下はシェルスクリプトと ~~awk~~ プログラムのリストです。

File: mar

```

#!/bin/sh
#
if [ $# -le 1 ]; then
    echo "Usage of Advance CAD macro archiver"
    echo "List macro elements      : $0 t lib.MAC"
    echo "Replace(Add) elements      : $0 r lib.MAC macro.MAC ..."
    echo "Delete elements            : $0 d lib.MAC element_name ..."
    exit 1
fi

swt="$1"
lib="$2"
out="tmp.$$"

if [ "$swt" = "t" ]; then
    egrep "/[*] Macro name =[ ¥t]*[A-Z][A-Z0-9]*[ ¥t]*¥*/" ${lib}

elif [ "$swt" = "r" ]; then
    if [ $# -le 2 ]; then
        echo "ERROR - No macro file name"
        exit 1
    fi

    shift 2
    for i in $*
    do
        if [ ! -f ${i} -o ${i} = ${lib} ]; then
            continue;
        fi
        name=`usr/bin/basename ${i} .MAC`
        namelist="${namelist}|${name}"
        echo "/[*] Macro name = ${name} */" >> ${out}
        cat ${i} >> ${out}
        echo "Module" ${name} "insert"
    done

    if [ ! -f ${lib} ]; then
        mv ${out} ${lib}
        exit 0
    fi

    nawk -f mar.awk -v type=${swt} -v namelist=${namelist} -v lib=${lib}
    -v out=${out} ${lib}
elif [ "$swt" = "d" ]; then
    if [ $# -le 2 ]; then
        echo "ERROR - No element name"
        exit 1
    fi

    if [ ! -f ${lib} ]; then
        echo "ERROR - " ${lib} " not found"
        exit 1
    fi

```

```

shift 2
for i in $*
do
    namelist="${namelist}|${i}"
done

nawk -f mar.awk -v type=${swt} -v namelist=${namelist} -v lib=${lib}
-v out=${out} ${lib}

else
    echo "ERROR - Unknown function letter ${swt}"
    exit 1
fi

```

File : mar.awk

```

BEGIN {
    if ((n = split(namelist, a, "|")) <= 0) {
        err = 1;
        exit;
    }
    for (i=1; i<=n; i++)
        b[a[i]];
}

/¥/¥* Macro name =[ ¥t]*[A-Z][A-Z0-9]*[ ¥t]*¥¥// {
    if ((ps = match($0, "/¥¥* Macro name =[ ¥t]*")) <= 0) {
        err = 2;
        exit;
    }
    ps += RLENGTH;
    if ((m = match(substr($0, ps), "[ ¥t]*¥¥*/")) <= 0) {
        err = 2;
        exit;
    }
    name = substr($0, ps, m-1);
    if (flag = (name in b)) {
        print "Old module " name " removed";
        nf++;
        next;
    }
}

flag == 0 { print $0 >> out }

END {
    if (err || (type=="d" && nf<=0)) {
        print "Error. No modification made";
        system("rm " out);
    } else {
        system("mv " lib " " lib ".OLD");
        system("mv " out " " lib);
    }
}

```

● シェルスクリプト mar の説明

左側の行番号は、説明のために付けたものです。

最初の行はこのシェルスクリプトが **bourne shell** を使用することを宣言しています。

```
#!/bin/sh
```

3～9 行はシェルスクリプトの使い方を表示します。

```
if [ $# -le 1 ]; then
  echo "Usage of Advance CAD macro archiver"
  echo "List macro elements : $0 t lib.MAC"
  echo "Replace (Add) elements : $0 r lib.MAC macro.MAC ..."
  echo "Delete elements : $0 d lib.MAC element_name ..."
  exit 1
fi
```

このシェルスクリプトは 2 つ以上の引数を必要とします。したがって実行時の引数が 1 つ以下ならば使い方を誤っているので、使い方を表示して終了させるようにします。

3 行目の if 文の \$# はシェルの組み込み変数でコマンドラインの引数の数を保持しています。5、6、7 行目の \$0 はシェルスクリプト名自身を保持しています。今回の場合は "mar" です。このシェルスクリプトを別の名前にリネームしたときも正しく新しい名前を表示するようにするには、その名前を直接記入せずに \$0 と記入します。

```
swt="$1"
lib="$2"
out="tmp.$$"
```

コマンドラインの 1 番目の引数 \$1 をシェル変数 swt, 2 番目の引数を lib に代入します。\$1, \$2 のままだまかまいませんが、シェルスクリプトが読みやすくなるように別の変数名を使います。もう 1 つの変数 out はテンポラリファイルの名前です。

tmp.\$\$ の \$\$ は組み込み変数で、このシェルスクリプトを実行しているプロセスの番号が入ります。このシェルスクリプトを同時に 2 つ以上実行してもファイル名が同じにならないようにするために使います。プロセス番号が 123 ならファイル名は "tmp.123" になります。シェル変数は代入文の右辺、あるいは参照の場合は変数名の前に \$ をつけなければなりません。このとき変数名を {} で囲んでもかまいません。

15～65 行目が主要部分で、1 番目の引数によって処理を分けています。

```
if [ "$swt" = "t" ]; then
  マクロのリストを行う部分
elif [ "$swt" = "r" ]; then
  マクロの挿入を行う部分
elif [ "$swt" = "d" ]; then
  マクロエレメントを削除する部分
else
  エラー
fi
```

マクロのリストを行う部分は 16 行目 1 行だけです。

```
egrep "/\* Macro name =[ Yt]*[A-Z][A-Z0-9]*[ Yt]*\*/" ${lib}
```

つぎはマクロの挿入を行う部分で、19～43 行目です。

マクロの挿入の場合は、3 番目の引数以後にマクロファイル名がなければならないので、引数が 2 つ以下だと誤りです。

```
if [ $# -le 2 ]; then
  echo "ERROR - No macro file name"
  exit 1
fi
```

24～35 行目はマクロファイルをヘッダー行をつけてテンポラリファイルにコピーします。それと同時に、挿入したマクロ名のリストを作ります。これは後でマクロライブラリから古いバージョンのマクロエレメントを除去するために使います。

```
shift 2
for i in $*
```

```

do
  if [ ! -f ${i} -o ${i} = ${lib} ]; then
    continue;
  fi
  name="/usr/bin/basename ${i} .MAC`
  namelist="${namelist}|${name}"
  echo "/* Macro name = ${name} */" >> ${out}
  cat ${i} >> ${out}
  echo "Module" ${name} "insert"
done

```

shift 2 は引数のうち最初の 2 つを取り除きます。残りの引数、つまりマクロファイル名ひとつひとつを for 文で処理します。

for i in \$* の \$* はシェルの組み込み変数で、すべての引数のならびを表します。

for i in \$* は引数を順次取り出し、それをシェル変数 i で参照します。そして do ~ done の間を繰り返します。for はすべての引数を処理すると終了します。

27~29 行の if 文は変数 i で参照するファイルが通常のファイルではない

(! -f \${i}) または (-o) 修正しようとしているマクロライブラリファイルである (\${i} = \${lib}) なら処理しないようにしています。

30 行目は変数 i のファイル名からディレクトリ名と拡張子 .MAC を取り除いた名前を変数 name に代入しています。ここでは /usr/bin/basename コマンドを使っています。ちなみにディレクトリ名だけを取り出すコマンドは /usr/bin/dirname です。

31 行目はマクロエレメント名のリストを作っています。変数 namelist の内容は最初は空です。順次バッチカルバー (|) とエレメント名を追加していきます。その結果、

"|DEB1|DEB2|DEB3" のような文字列になります。

最初のバッチカルバーは今回は必要です。これを取りたいときは、つぎのようにします。

```
namelist="${namelist:+${namelist}}|${name}"
```

この文中の + は namelist が空でないときそのつぎの "\${namelist}" が行われるように制御します。他にも :, =, ? などがあります。

32 行目はマクロエレメントヘッダをテンポラリファイルに書き込み、33 行目はマクロファイルをテンポラリファイルにコピーします。

37~40 行目は、指定したマクロライブラリが新しいファイルであればテンポラリファイルをマクロファイル名にリネームして終了します。

```

if [ ! -f ${lib} ]; then
  mv ${out} ${lib}
  exit 0
fi

```

すでにライブラリファイルがある場合、挿入したマクロファイルの古いバージョンをマクロライブラリから削除しなければなりません。これは挿入したマクロ以外のマクロだけをマクロライブラリファイルからテンポラリファイルにコピーすることでできます。

```
nawk -f mar.awk -v type=${swt} -v namelist=${namelist} -v lib=${lib} -v out=${out} ${lib}
```

この作業は nawk コマンドで行います。fmar.awk は nawk のプログラムを記述したファイル名を指定します。type=\${swt} はシェル変数 swt の内容を nawk の変数 type に渡すためのものです。つぎの namelist, lib も同様です。最後の引数 \${lib} は nawk が処理するファイル名です。ここではマクロライブラリファイル名です。

~~44～60 行目はマクロエレメントを削除する部分です。マクロの挿入とほとんど同じです。マクロライブラリからマクロエレメントを削除するのにやはり `nawk` を使用しています。~~

```
nawk -f mar.awk -v type=${swt} -v namelist=${namelist} -v lib=${lib} -v out=${out} ${lib}
```

● ~~awk プログラム mar.awk の説明~~

~~awk プログラムは 3 つの部分からなります。およそつぎのようなプログラムスタイルになります。~~

```
BEGIN { 前処理 }
入力ファイルの各行を処理する部分
END { 後処理 }
```

~~1～8 行目は前処理です。~~

```
BEGIN {
  if ((n = split(namelist, a, "|")) <= 0) {
    err = 1;
    exit;
  }
  for (i=1; i<=n; i++)
    b[a[i]];
}
```

~~変数 `namelist` は削除するマクロエレメント名のリストを保持しています。この変数は `nawk` の起動時に設定されています。`split(namelist, a, "|")` は `namelist` の内容をバーチカルバー (|) 文字で分割し、その結果を配列 `a` に作り出します。たとえば `"DEB1|DEB3|DEB5"` は `"DEB1"`, `"DEB3"`, `"DEB5"` の 3 つの文字列に分割されます。~~

~~分割に失敗したときは `(n<=0)`, `exit` します。ここでの `exit` は `END { 後処理 }` へ移ります。awk プログラムの終了ではありません。~~

~~後でマクロライブラリ中のエレメント名がこの削除エレメント名リストにあるかどうか比較します。このときつぎのような for ループで検索する方法があります。~~

```
for (i=1; i<=n; i++)
  if (a[i]==name)
    print "match";
```

~~ここでは、こうしないで `awk` の連想配列の機能を使います。連想配列は添字が文字列をとりま~~
~~す。6、7 行目が連想配列 `b` を作っている部分です。連想配列 `b` の要素を参照するにはつぎのよう~~
~~にします。~~

```
b["DEB1"], b["DEB3"], b["DEB5"]
```

~~あるエレメント名 (`name`) が連想配列 (`b`) の添字に使われているかどうかは、つぎの文でテストで~~
~~きます。~~

```
name in b
```

~~`b[name]` があれば真、なければ偽を返します。これは配列要素 `b[name]` の値には関係ありません。この方が for ループで検索するよりも簡単で、しかも処理が速くなります。これは 21 行目で使~~
~~用しています。~~

~~つぎは本体部分ですが、これは入力ファイルを 1 行ずつ読み込み、各行ごとに処理を繰り返しま~~
~~す。ファイルを 1 行ずつ読み込んだり、繰り返しの制御は `nawk` 自身が行うので、ユーザの `awk`~~
~~プログラム上には記述がありません。10～28 行目がそうです。~~

```
/Y/Y* Macro name = [ Yt]*[A-Z][A-Z0-9]*[ Yt]*Y*Y// {
  if ((ps = match($0, "Y* Macro name = [ Yt]*")) <= 0) {
```

```

err = 2;
exit;
}
ps += RLENGTH;
if ((m = match(substr($0, ps), "[ \t]*YY*/")) <= 0) {
err = 2;
exit;
}
name = substr($0, ps, m-1);
if (flag = (name in b)) {
print "Old module " name " removed";
nf++;
next;
}
}
flag = 0 { print $0 >> out}

```

~~10～26行目は、読み込んだ行が /* Macro name=NAME */ の文字列を含んでいるときに実行されます。nawk が読み込んだ行は awk 組み込み変数 \$0 で参照します。~~

~~11～20行目でマクロエレメント名を取り出し、変数 name に代入します。~~

~~21～25行は、そのエレメント名が削除エレメント名リスト (連想配列 b) にあるかどうかテストします。もしあれば変数 flag が 1 (真) になれば 0 (偽) になります。flag が真、つまり削除するエレメントであれば、つぎの入力行を処理するために next 文で終了します。10行目から繰り返しになります。そうでなければ28行目以下を処理します。~~

~~28行目は、flag が 0 であれば削除しないで変数 out で示すファイルにコピーします。そしてつぎの入力行を処理するために10行目へ戻ります。ここでは次の行は後処理の END なので、自動的に next が行われます。~~

最後は後処理です。

```

END {
if (err || (type=="d" && nf<=0)) {
print "Error. No modification made";
system("rm " out);
} else {
system("mv " lib " " lib ".OLD");
system("mv " out " " lib);
}
}

```

~~エラーが発生したり、エレメントの削除のときに、ひとつも合致するエレメントがなかったときは、何も変更しないことにします。作りかけのテンポラリファイルは削除します。system("rm " out); の文は nawk の中から rm コマンドを実行するためのものです。~~

~~成功したときは旧ライブラリファイルを別の名前で保存し、テンポラリファイルをマクロライブラリファイルにリネームします。~~

10.5.13 セッションファイルからマクロを作る (参考例)

「マクロ作成開始を指示し、マクロ作成終了指示までに実行したコマンドをマクロファイルに出力させておくと、自動的にマクロができる機能があるか」という質問があります。実際にこのような機能を持っている CAD システムもあります。

Advance CAD ではこの方法は採用しておりません。その理由はいくつかあります。

- ・ 誤操作をしたら正しいマクロにはならない

- ・ 変数にしたいところの指示が困難
- ・ 繰り返し、条件判定の組込みが困難

つまり自動作成したマクロを実用的なマクロに仕上げるためには、かなりの修正が必要だということです。しかし「たとえ手を加えなければならないとしても、元になるマクロができればよい」という意見もあります。そこで、これを実現する方法を考えました。

Advance CAD にはセッションファイルがあり、実行したコマンドがすべて保持されています。このファイルを使って操作を再実行させることができます。このセッションファイルからマクロファイルを作るのが妥当ではないかと考えました。まずセッションファイルから誤入力やエラー・バックスペースなどを除去します。念のためセッションファイルを使って再実行させ、正しく再現されることを確認します。そしてセッションファイルからマクロファイルを作るプログラムを実行します。最後にマクロを修正します。

macrogen はセッションファイルからマクロを作るユーティリティプログラムです。(ファイル macrogen は Version 13 から 実行形式ファイルになりました。)

使い方は次のとおりです。

```
prompt% macrogen セッションファイル名 <cr>
```

たとえばつぎのようにすると、セッションファイルから生成したマクロが画面に表示されます。

```
prompt% macrogen ACAD.LOG <cr>
```

これをファイルに保存したければ、つぎのようにします。ファイル TEST.MAC ができます。

```
prompt% macrogen ACAD.LOG > TEST.MAC <cr>
```

10.5.14 マクロの例

● 例 1 始点を指定して円周上に等分割点を発生させるマクロ

```
echo "Enter number of division";
n = getnum(1);
if (n < 1) {return;}
echo "Enter angle of first point";
a = getnum(1);
echo "Select circle";
ITM/SEL MARC <CE>
VER [getpnt(1)] <CE>
ITM/SEL MANY <CE>
if (#VERRAD < 0.1) {return;}
da = #VERANG/n;
PCE
for (j=0; j<=n; j+=1) {
    <[#VERXC+#VERRAD*cos(a)], [#VERYC+#VERRAD*sin(a)]>
    a += da;
}
<CE>
```

<各行の説明> 上記の行番号は説明上のものです。実際には記述しません。

行	内容
1 ~ 3	円弧の分割数を入力する。分割数が 1 よりも小さければ終了。
4 ~ 5	円周上の第 1 点の角度（単位は度）を入力する。

行	内容
6 ~ 10	円／円弧を入力する。 円弧以外をピックしないように、アイテム選択マスクを設定している。 VER コマンドで円の中心点 (#VERXC, #VERYC)、中心角 (#VERANG)、半径 (#VERRAD) をシステムレジスタに設定している。
11 ~ 17	円周に等間隔点を発生させる。 コマンドは PCE で、点の座標を for ループで計算する。

● 例2 クロソイド曲線

クロソイド曲線の始点と、始点における接線方向の点の2つを与え、パラメータ A と半径 R を入力する。クロソイド曲線を近似する自由曲線ができる。

```

/* Curve R*L = A^2 */
echo "Digitize origin";
if (vtype(p1=getpnt(1))!=6) { return ; }
echo "Digitize tangent";
if (vtype(p2=getpnt(1))!=6) { return ; }
echo "Enter parameter A";
if (vtype(a=getnum(1))!=3) { return ; } if (a <= 0) { return ; }
echo "Enter Radius R";
if (vtype(r=getnum(1))!=3) { return ; } if (r <= 0) { return ; }

wo = pce(0,0); wx = pce(1,0); /* Save WCS */
WCS/OFF WCS RTP [p1] [p2] <CE>
FREE @XO@YO
lng = a*a/r; dl = lng/10;
for (l=dl; l<=lng; l+=dl) {
  z = (l/a)*(l/a)/2;
  z2 = z*z; z4 = z2*z2; z6 = z4*z2;
  x = a*sqrt(2)*sqrt(z) * (1 - z2/10 + z4/216 - z6/9360);
  y = a*sqrt(2)*sqrt(z)*z/3 * (1 - z2/14 + z4/440 - z6/25200);
  @X[x]@Y[y]
}
<CE>
z = 180*(lng/2/r)/pi;
xm = x - r*sin(z);
ym = y + r*cos(z);
CRC RAD [r] @X[xm]@Y[ym] <CE>
WCS RTP [wo] [wx] <CE> /* Restore WCS */

echo "A=" a " R=" r " L=" lng " Xm=" xm " Ym=" ym:

```

● 例3 MES コマンドで測定した距離を平行線コマンドで使用する

```

/* AdvanceCAD Ver 6 Macro */
echo "Indicate TWO lines to measure distance";
if (vtype(p=getpnt(2)) != 30) { return; }
error();
MES/SEG [p] <CE>
if (error()) { return; }
LPL #MESDST
/* End */

```

● 例4 直列長さ寸法の寸法値を合計する

長尺もので一部の寸法値がオフスケールのとき、全体の寸法値を計算するときに使う。つぎのような直列長さ寸法の寸法値があり、(1000)の寸法値はオフスケールの寸法値とする。

50, 25, (1000), 20, 35

五つの寸法値をピックし <CE> と入力すると合計が求まる。

L=1130

```
/* AdvanceCAD Ver 6 Macro */
echo "Indicate dimension texts"; p=getpnt(8);
if (vtype(p)!=30) { return; }
if ((n=length(p))<2) { return; }

error();
dst = 0;
for (j=0; j<n; j+=1) {
  VER/SEG [p[j]]
  if (error()) { return; }
  s = #VERTXT;
  l = strstr(s, "¥MD");
  if (l >= 0) {
    l += 2;
  } else {
    l = strstr(s, "(") + 1;
  }
  m = length(s);
  d = sscanf(s[l:m], "%f");
  if (vtype(d)!=3) { return; }
  dst += d;
}

echo "L=" dst;
/* End */
```

● 例 5 テキストファイルの内容を表示する

```
/* AdvanceCAD ver 15 Macro */
w = mopen(33);
p = pos(1, 1, 33);
fd = fopen("/acad/macro/EXAMPLE5.MAC", "r");
while (1) {
  if (vtype(s=fgets(80, fd)) != 4)
    { break; }
  mecho(p, s);
  if (p[1] < w[1]) {
    p[1] += 1;
  } else {
    p = pos(0, 0, 33);
    merase(p);
    p = pos(1, 1, 33);
  }
}
fclose(fd);
mecho(p, "Hit any key to terminate");
getany(1);
mclose(w);
/* End */
```

● 例 6 円に沿った文字列を記入する

```
w0 = pce(0,0); wx = pce(1,0);
sa = 145; da = -5; r = 50;
TEXT/ORGHOR 1
TEXT/ORGVER 1
TEXT/FONT 102
TEXT/KFONT 2
WCS/OFF WCS RTP <0,0> [sa] <CE>
```

```
s="CITOH 伊藤忠 TECHNO-SCIENCE";
n = length(s);
a = 0;
for (j=0; j<n; j+=1) {
  if (s[j] >= " " && s[j] <= "~") { /* ASCII char */
    t = s[j];
  } else { /* Japanese char */
    t = s[j:j+1]; j += 1;
  }
  b = (sa-90) % 360;
  GNT [t] ANG [b] [pce("@R",r,"@A",a)] <CE>
  a += da; sa += da;
}
WCS RTP [wo] [wx] <CE>
```

第 11 章 同時設計（コンカレント）

11.1 概要

ひとつの製品をひとつのモデルで設計／製図すると、ひとりの設計者しか担当できず、設計期間が長くなることがあります。設計期間を短縮するには、製品で使用する部品単位での設計／製図を同時に何人かの設計者で分担し進め、最後にひとつの製品に組み立てる方法をとります。しかし、部品単位モデルを製品モデルに組み込んだとき、部品間の取り合いが悪く調整が必要となることがあります。そうした時はいくつかの部品モデルを修正しなければなりません。あらかじめ製品モデルに組み込んだ状態をプロットアウトするなどして、修正部分がわかるようにしてから作業しなければなりません。そして再度、製品モデルを呼び出して確認しなければなりません。

この問題は、製品モデル全体を見ながら、部品単位的设计ができるようになれば解決します。製品モデル全体を見ながら部分 design することができますから、錯覚が減り、他の部分との取り合いの矛盾を早めに見つけ、対処することができますようになります。

そして、各設計者の作業が済めば製品モデルが完成します。

Advance CAD ではこれを実現できる機能があり、それを**同時設計 (Concurrent Design)** 機能と呼びます。

基本的な考え方は次のようなものです。

製品モデルを作る

- 製品モデルはいくつかの部品モデルをサブモデルとして配置したものとする。このとき、部品のモデルは未完成であってもかまわない。

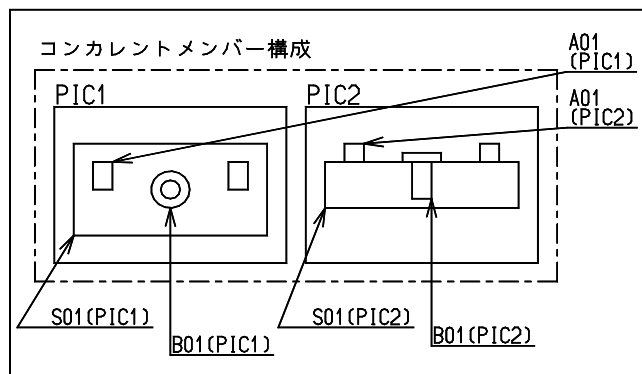
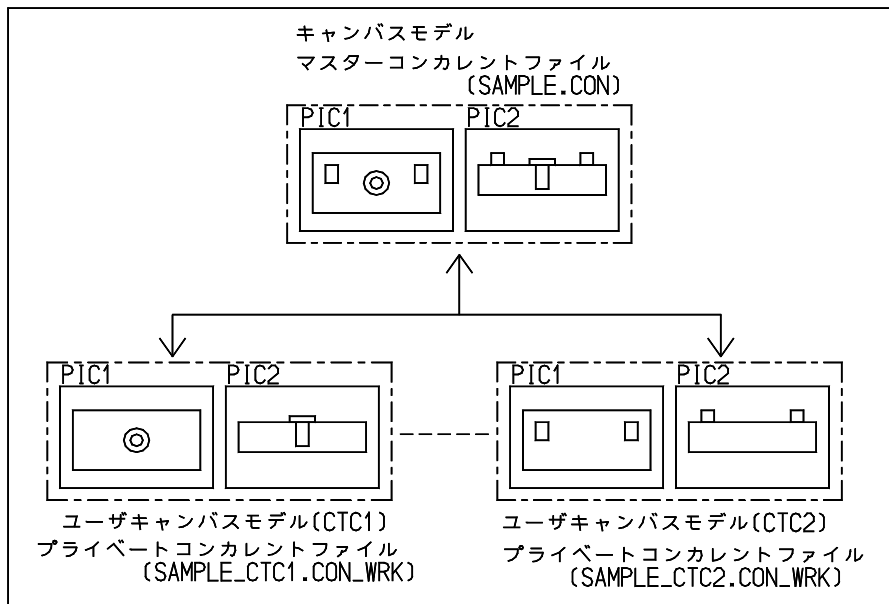
部品モデルを修正する

- 製品モデルを呼び出す。
- 修正したい部品のサブモデルを分解してキャンバスモデルのアイテムにする。
- 分解された部品を修正する。
- キャンバスモデルのアイテムを部品のモデルファイルに格納する。

このようにすれば、"同時設計" ができます。同時設計機能はこれを簡単に誤りなくできるようにしてくれます。

同時設計でのデータの関係は次の図のようになります。

(ペアレント名を "SAMPLE" とし、3 つの部品からできる簡単な製品を例として説明していきます。)



ペアレント名

コンカレント構成に付ける名称で、コンカレントの基本となります。

コンカレントマスタファイル (SAMPLE.CON)

コンカレントのキャンバスモデルを管理するファイルで、全メンバーが配置され、位置関係も正しいファイルです。

ペアレント名ごとにひとつです。次の情報を管理しています。

- ・ メンバー名
- ・ メンバーの使用状況
- ・ メンバーの配置条件
- ①ピクチャ
- ②レビジョン／クラス
- ③配置点, X 縮尺, Y 縮尺, 配置角度

ユーザコンカレントマスタファイル (SAMPLE_CTC.CON_WRK)

ユーザ毎の設計検討中に使用するコンカレントファイルで、ペアレント／ユーザ毎に作成されます。ユーザ毎の設計検討中に必要なメンバーのみを管理し、明示的にマスターコンカレントファイルへ更新する必要があります。

マスターコンカレントファイルと以下の点が違っていても可。

- メンバー構成（追加、削除）
- メンバーの配置位置
- メンバーの中身

キャンバスモデル

マスターコンカレントファイルの構成とメンバー配置をイメージするモデル。

ファイルとしては存在しないが、ユーザごとにユーザキャンバスモデルファイルとして存在します。

ユーザキャンバスモデル (SAMPLE_CTC.CON_MDL)

ユーザごとのキャンバスモデルファイルで、プライベートコンカレントファイルと対応しています。

キャンバスモデルとメンバー

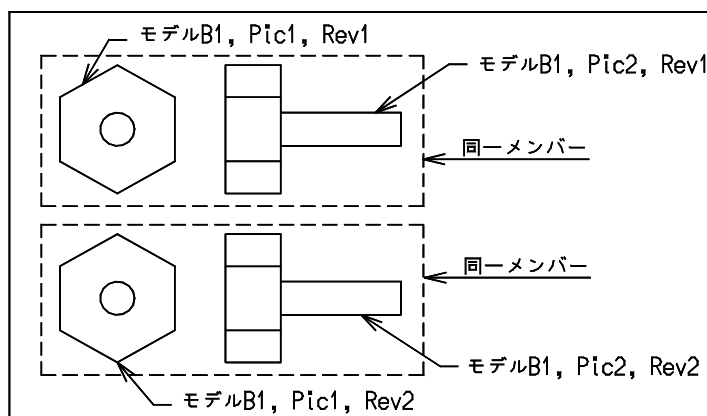
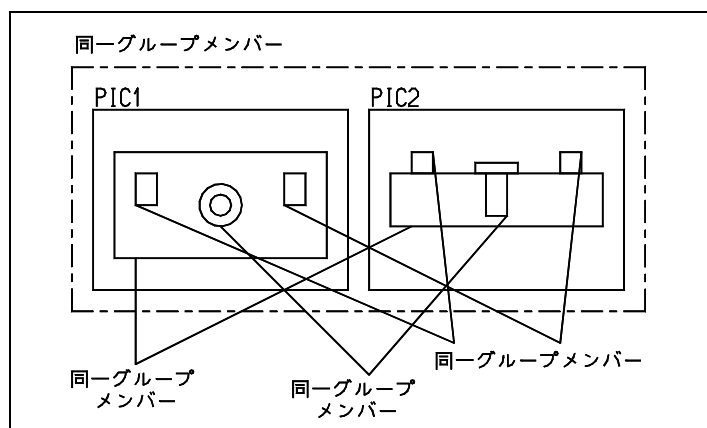
製品モデルを表示します。マスターコンカレントファイルにもとづいて構築されます。

この中には部品がメンバーアイテムとして配置されます。メンバーアイテムはサブモデルアイテムとほとんど同じものです。

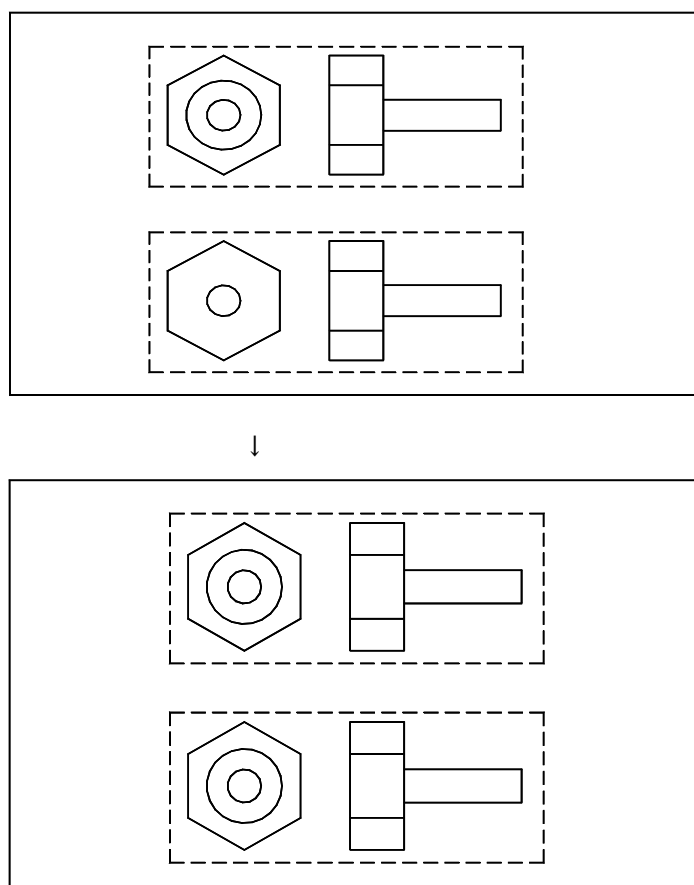
各々のメンバーアイテムは、部品モデルファイルのひとつのピクチャの内容をコピーして持っています。メンバーアイテムを修正し、更新を行うと対応するモデルファイルのピクチャが更新されます。

同一グループメンバー

メンバー名が同じで異なるピクチャ番号で、作成されたモデル。同一ピクチャに複数個同一モデル名・同一 PIC が配置された場合には異なるグループとみなし、その場合同一グループは同じレビジョン番号を持つものとする。



メンバー名が同じで、同じピクチャのメンバーが修正されメンバーに戻されると他のメンバー図形も修正されます。

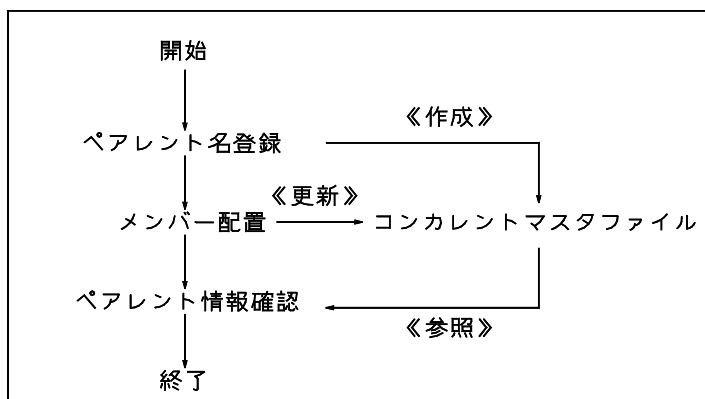


- 注 1) バージョン 1.2 よりコンカレント機能を見直し修正しました。バージョン 1.1 以前のコンカレントとはコマンド形態は大きく変更されておりませんが、CON __, QON __ コマンドを TON __ コマンドで統合し、スイッチにより処理方法を変えるようになりました。
- 注 2) モデルタイトルの 196-199 をコンカレントでリザーブしていましたが、バージョン 1.2 よりリザーブの必要が無くなりました。

11.2 機能説明

11.2.1 キャンバスモデル

同時設計機能を使用するには、ひとつの製品を表現するキャンバスモデルを作成します。下図の手続きを行います。



最初は、ペアレント名登録を行います。

TON __ START コマンドでペアレント名を指定しますと、新しく空のマスタコンカレントファイル (ペアレント名 .CON) とプライベートコンカレントファイル (ペアレント名 _ アカウント名 CON_WRK) が作成されます。

このファイルには、この製品に使用される部品であるメンバーの情報が記録されます。

このファイルは同時設計機能では、たいへん重要なファイルですから修正時には十分注意してください。

つづいて、このペアレントが参照するメンバーを配置します。つまりこの製品で使用される部品図を指定の位置に取り付けることです。部品図は完成品でも未完成でもかまいません。ペアレントの組み立てが完了したら、何人かで同時に部品図をこの製品に合わせて設計をすすめるのが目的ですから、部品の位置が決まればよいのです。

メンバーの配置には2通りの方法があります。

ひとつは **TON __ ADD** コマンドです。これは既存の部品モデルのひとつのピクチャを取り出し、メンバーとして配置します。部品モデル中のすべてのピクチャを一度に取り込むこともできます。

この場合は各ピクチャが別々のメンバーになり同一グループと見なされます。

メンバーが配置されるとき、コンカレントファイルにメンバー情報が追加されます。誤って配置したメンバーは、**TON __ DEL** コマンドで取り除いてください。

メンバーが削除されるとき、コンカレントファイルも更新されます。

もうひとつは **TON __ APD** コマンドで、製品図から部品図を切り出しメンバーに変換します。つまり、次のことを一度に行います。ひとつの部品図を表現する部分を取り出しモデルファイルを作ります。モデルファイルに格納した部分を削除します。そして先に作成したモデルのピクチャをメンバーとして配置します。

最後にペアレント情報を確認します。

TON __ DSP コマンドはマスタコンカレントファイルの情報を表示します。

キャンバスモデルができれば終了です。このモデルは保存する必要はありません。

キャンバスモデルを構築するのに必要な情報はコンカレントファイルに記録されています。

11.2.2 同時設計作業

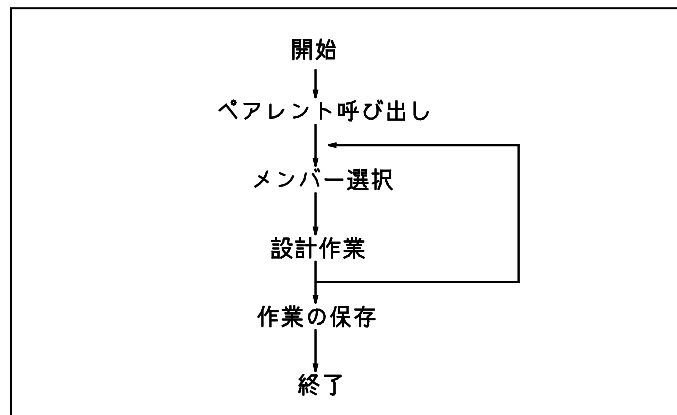
キャンバスモデルが完成したら、各部品の設計にはいります。何人かで同時に設計を進めますのであらかじめ、だれがどの部品、つまりどのメンバーを担当するかを決めます。

だれかが修正中のメンバーを他の人が修正しようすると警告がでます。

しかし、警告を無視して修正に入ることができます。修正を禁止しておりませんので、注意してください。

また、設計者はメンバーとして参照されているモデルを直接、修正してはいけません。

以下で説明するように、必ずペアレントを呼び出し、修正するメンバーを指定してから作業に入らなければなりません。そうしないと、だれがどのメンバーで作業しているかわからなくなります。



最初はペアレントを呼び出します。

TON __ CALL コマンドは、プライベートコンカレントファイルにもとづきキャンバスモデルを再構築します。プライベートコンカレントファイルが存在しない場合は、マスターコンカレントファイルから自動的に作成します。

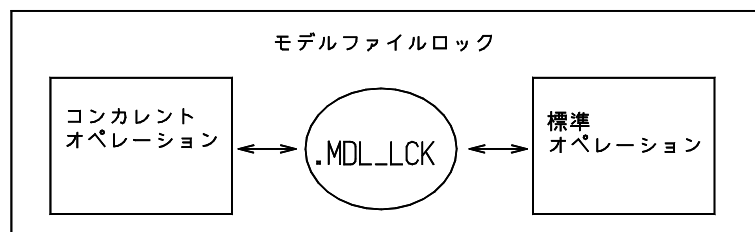
次に作業するメンバーを選択します。

TON __ SEL コマンドで作業するメンバーを選びます。このコマンドは、コンカレントファイルを調べ、そのメンバーが作業中であれば、警告を出します。キャンセルするか、警告を無視するか指示してください。

コンカレント機能では、メンバーの使用状況を独自に管理するために、マスターコンカレントファイルでメンバーの使用状況を管理しています。

このコマンドは、選択されたメンバーを分解します。その時点で member.MDL_LCK ファイルが作られます。このため標準コマンドでメンバーモデルを修正しようするとワーニングメッセージが表示されます。

このように、作業対象となっているメンバーをアクティブメンバーと言います。



このあとは、通常のコマンドで設計作業を進めます。

作業が一段落したら、その結果をメンバーのモデルファイルに反映させなければなりません。

- (1) TON __ FLE コマンドを実行すれば、メンバーのモデルファイルが更新されます。
具体的にいうと、キャンバスモデルのカレントピクチャ図形がメンバーのモデルファイルの該当ピク

チャに入り更新されます。そして格納した図形がメンバーに置き換えられます。最後に、MDL_LCK ファイルが解除されます。同一メンバー・ピクチャは自動的にすべて最新の形状に変更されます。

- (2) TON __ FLE コマンドを実行しないで、TON __ SEL コマンドで別のメンバーを選択してもかまいません。
自動保存に設定されている場合には、自動的に TON __ FLE と同じことを行ってから別のメンバーをアクティブにします。
- (3) もうひとつの方法は、キャンバスモデルを保存することです。
TON __ MDL __ WRITE コマンドを実行すると、やはり自動的に TON __ FLE と同じことを行ってから、ユーザキャンバスモデルファイルとして保存します。

以上の3つのどれも行わず Advance CAD を終了すると、作業は消滅してしまいます。また、アクティブメンバーが終了していないため、MDL_LCK ファイルが解除されませんので注意してください。

11.2.3 表示制御

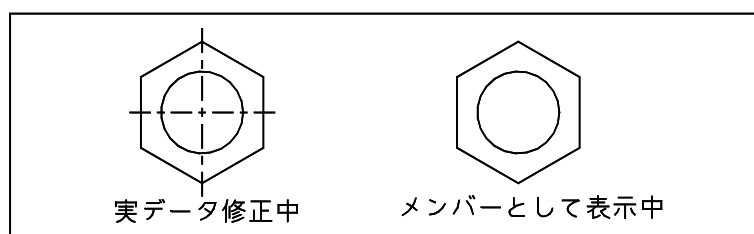
多くのメンバーを持つペアレントでは、図面が複雑になります。そこでとりあえず必要ないメンバーを非表示にすることで、図面を簡素化して作業します。

TON __ MOD コマンドはメンバーの表示 (show)、非表示 (no_show) を変更します。

同時設計では、何人かの設計者が別々のメンバーを修正しています。そこで、他の設計者が変更したメンバーをキャンバスモデルに反映させる必要があります。これを行うのが TON __ UPDATE コマンドです。

メンバー修正中には全データを必要としますが、メンバー表示時には補助線・寸法など非表示にしたい図形があります。それらの図形を非表示にするために TON __ DSP __ MOD コマンドがあります。クラス、アイテムの表示マスクをメンバー内のひとつひとつのアイテムに設定することができます。マスク情報はマスターコンカレントファイルには反映されません。

注) TON __ VER コマンドでも同様の設定ができます。



11.2.4 キャンバスモデルの保存／呼び出し

ペアレントの呼び出しは、TON __ CALL コマンドで行うと述べましたが、もうひとつ TON __ MDL __ READ コマンドがあります。

これは、TON __ MDL __ WRITE コマンドで保存したユーザキャンバスモデルファイルを読み出します。

ペアレント呼び出しとユーザキャンバスモデルファイルの呼び出しの違いは、次のようです。

ペアレント呼び出しは、コンカレントファイルにもとづいて、新規モデルにキャンバスモデルを再構築します。従って、スクリーンレイアウト・モデル定数・表示マスクなどがデフォルト値になっています。

これに対し、ユーザキャンバスモデルの呼び出しは、再構築は行いません。
保存を行った状態から作業をつづけることができます。

同時設計においてキャンバスモデルは、コンカレントファイルがもとになります。ですから、ユーザキャンバスモデルを保存することは、キャンバスモデルの構築という観点からは意味がありません。しかし、設計者が以前に作業していた状態から作業を開始できることに意味があります。

何人かの設計者が同時に同じペアレントを呼び出し作業していますから、各人が別々のキャンバスモデルファイルを持たなければなりません。そのために、デフォルトのキャンバスモデルファイル名は、" ペアレント名__ユーザ名. CON __ MDL " となります。ここでユーザ名というのは、Advance CAD のアカウント機能で設定するもので、設計者はそれぞれ異なるユーザ名を持つものとします。

11.2.5 コンカレントファイルとキャンバスモデルの整合

キャンバスモデル上で、直接メンバーの位置を移動したり、ピクチャ・クラス・レビジョンを変更した場合、コンカレントファイルと食い違いが生じます。
このようなとき、現状に合わせてコンカレントファイルを作り直す機能があります。
TON __ ADJ コマンドです。

11.2.6 ペアレント情報

メンバーごとに以下の内容を表示します。

状態記号

" * * " は変更されたメンバーであることを表します。
また、アクティブメンバーであれば番号の横に " + " が表示されます。

モデル名とピクチャ番号

メンバーに対応するモデルファイル名とピクチャ番号を表示します。

表示／非表示

メンバーが表示 (show) または非表示 (no_show) どちらであるかを表示します。

ピクチャ・クラス・レビジョン番号

メンバーアイテムのアイテム属性を表示します。

配置点

メンバーアイテムの配置位置 (X, Y) を表示します。

使用者名

このメンバーを修正してるユーザ名で " ユーザ名 @ ホスト名 " の形式で表示します。

例

```
Control file : SAMPLE Member Max : 3
SubmodelName OrgPic Pic Cls Rev Origin Host:user
< > 1 MEMBER-1 (1) show 1 101 2 0.00 0.00 cat:CTC
2 MEMBER-2 (2) no_show 1 102 3 100.00 100.00
< > 3 MEMBER-3 (1) show 1 103 4 0.00 0.00 dog:HAR
```

11.2.7 ペアレント情報の回復

何らかの原因でマスターコンカレントファイル（ペアレント名 .CON）がなくなった場合の復旧方法は、つぎのとおりです。

- (1) TON __ START コマンドで再作成を行う。
- (2) ユーザキャンバスモデルファイル（ペアレント名__ユーザ名 .CON_MDL）が存在する場合
 1. ユーザキャンバスモデルファイルを通常のモデルファイルに複写し、モデル呼び出し（MODEL / READ）コマンドで読み込む。
 2. ペアレント名を指示して TON __ ADJ コマンドを実行すると、ユーザキャンバスモデルファイルに残っていたメンバーにもとづき、コンカレントファイルを再作成する。

注）この時、非表示（no_show）メンバーはコンカレントモデルファイルに残りません。

11.3 準備

同時設計機能を使用するため以下の準備をします。

11.3.1 メニューファイルの修正

MEMB/SWITCH コマンド（14.4 章のメンバアイテムの選択を参照）をオンスクリーンメニューに表示させるために、メニューファイルの `#define CONCURRENT` を有効にします。
具体的には menu ディレクトリ内の ACADDEF.MEN を USERDEF.MEN というファイル名でコピーし、USERDEF.MEN の 5 行目の `#define CONCURRENT` の 1 カラム目の `"` を削除します。

<code>/#define CONCURRENT</code>	変更前
↓	
<code>#define CONCURRENT</code>	変更後

11.3.2 コンカレントファイルのディレクトリ

ACAD.SET にマスターコンカレントファイルのディレクトリとファイル拡張子を追加します。プライベートコンカレントファイルも同じディレクトリに作成されます。

```
#CONCURRENT#    "~/concur/"    !.CON!    Concurrent Master file
```

11.3.3 ユーザキャンバスモデルファイルのディレクトリ

ACAD.SET にユーザキャンバスモデルファイルのディレクトリとファイル拡張子を追加します。

```
#CON:MODEL#    "~/work/"    !.CON_MDL!    Concurrent User Canvas model file
```

11.3.4 ユーザ名の登録

ユーザ名の登録方法は、『システム管理者の手引き』の「セキュリティとアカウントिंग」の章をご覧ください。

11.3.5 モデル定数の設定

モデル定数 (RVP/MODEL) の、結合アイテムの [表示の制御およびカラー] 表示などを設定してください。

コンカレントの基本動作を決めるコンカレント定数 (RVP/CONCUR) の、以下の項目を設定してください。

- | | |
|--------------------------|-----------------|
| • 同一グループのメンバー全てをアクティブにする | CON_MULT_EXP! |
| • コンカレント保存時にモデルを更新する | !CON_DIR_WRT! |
| • 同一メンバーの複数配置を可能にする | !CON_MULT_MEMB! |
| • メンバー選択時に白色表示する | !CON_SEL_RPT! |

同一グループのメンバー全てをアクティブにする

異なるピクチャに同一グループのメンバーが配置されている場合にメンバー選択時に全てをアクティブにするかどうかを設定する。

オン (1) の場合は、選択されたメンバーの同一グループのメンバー全てをアクティブにする。

オフ (0) の場合には、選択されたメンバーのみをアクティブにする。

- 0 : 同一グループのメンバー全てをアクティブにしない
1 : 同一グループのメンバー全てをアクティブにする

コンカレント保存時にモデルを更新する

メンバー変更時に自動的にモデルを更新するかどうかを設定する。

オン (1) の場合は、自動的にモデルを更新する。

オフ (0) の場合には、自動的にモデルを更新しないので、更新指定時のみに更新されます。

- 0 : コンカレント保存時にモデルを更新しない
1 : コンカレント保存時にモデルを更新する

同一メンバーの複数配置を可能にする

同一メンバーを同一ピクチャに複数配置可能にするかどうかを設定する。

- 0 : 同一メンバーの複数配置を可能にしない
1 : 同一メンバーの複数配置を可能にする

※ TON __ SUB __ MEMB コマンドの場合には、自動的に複数配置可能になります。

メンバー選択時に白色表示する

コンカレントのメンバー選択時に白色表示を行うかどうかを設定する。

オン (1) の場合は、選択されたメンバーが白色表示され、決定するとメンバーが選択され実アイテムに変更されます。

オフ (0) の場合には、白色表示されません。

- 0 : メンバー選択時に白色表示しない
1 : メンバー選択時に白色表示する

11.3.6 制約

- (1) ひとつのキャンバスモデルは、メンバーを 2048 個まで持つことができます。
- (2) ひとつのメンバーの最大サイズは、設定可能ですがスピード等を考慮し標準のおよそ 1 M バイトまでとしてください。
- (3) ひとつのペアレントを構成するモデルファイルについて使用できるピクチャは #1 ~ #255 までです。ピクチャ #256 は使用しないでください。ドローイングスケール・ピクチャスケールは、全モデルを同じ設定にしてください。
- (4) UNDO 処理について
 - ・ UNDO の直後に UNUNDO をした場合は、何も影響はありません。
 - ・ TON __ FLE の直後は、UNDO はできません。
 - ・ コンカレント保存時にモデルを更新する設定になっている場合、TON __ SEL で直前にアクティブメンバーがあった場合、アクティブメンバーは、TON __ FLE 処理されるので UNDO はできません。TON __ SEL されたメンバーは、メンバーアイテムから実データベースに展開されています。そのため、UNDO した場合、展開された実データは直前のメンバーアイテムに戻ります。ただし、コンカレント上は "+" のアクティブメンバーが表示されます。
 - ・ TON __ DEL でメンバーを削除し、次に UNDO で復元した場合、このメンバーはアクティブモデル上では復元していますがコンカレントファイル上からは削除されたままです。このような UNDO は

コンカレントファイルとの不整合をひきおこすことになります。コンカレントファイルとの整合をとるには TON __ ADJ コマンドを実行してください。

- ・ UNDO コマンドで直前のメンバーアイテムに戻した場合、コンカレントはこれらのメンバーが選択されているものとして処理してしまいます。
TON __ SEL 実行中にアクティブメンバーの選択処理を中止し直前のメンバーアイテムに戻すには、TON __ CANCEL コマンドを使用します。
- (5) 作業途中で、メンバーにアイテムが存在しない状態でメンバー保存しようとした場合は、データエラーとみなし保存できません。メンバー中の全アイテムを削除するには、メンバー削除コマンドを使用してください。
- (6) 同一グループメンバーは 1 2 7 個まで定義できます。
- (7) コンカレントを使用する場合には、アソシエーション機能は使用できません。
パーツ機能、CUT_MOVE コマンドを使う場合に注意してください。
- (8) コンカレントの仕様上、アイテム名を使用して処理を行うこともできません。
- (9) コンカレントで使用するモデルファイルは、全て最新バージョンでお使いください。
バージョン 19 からは、旧バージョンのモデルファイルは自動的にアップグレードしますので、事前処理は不要です。

11.3.7 処理速度

同時設計では、複雑・大量のデータを処理するため処理速度が遅くなることがあります。処理速度をできるだけ速くするためには、次の方法があります。

- (1) 必要なメンバーのみをプライベートコンカレントファイルに取り込む。
メンバーが少なければ表示データが少ない分、処理速度は速くなる。
- (2) 必要なメンバーのみを表示 (show) にする。非表示 (no_show) のメンバーが多ければ表示データが少ない分、処理速度は速くなる。
- (3) TON __ MOD コマンドで全てのメンバーを表示にする指定 (SHOWALL) は、頻繁に使用しない。また、使用後はできるだけ必要なメンバーのみを表示状態にして作業を行うようにする。
- (4) TON __ MDL __ WRITE コマンドでは表示メンバーだけがユーザキャンバスモデルファイルに保存される。そのため、TON __ MDL __ READ コマンドでユーザキャンバスモデルファイルを読み込み後、表示メンバーを変更すると処理時間がかかる場合がある。

11.4 コマンド

● コマンド一覧

コマンド名	機能
TON_START	コンカレントファイルを新規作成する
TON_EXIT	コンカレントを終了する
TON_CALL	ペアレントを呼び出す
TON_MDL_READ	ユーザキャンバスモデルを読み込む
TON_MDL_WRITE	ユーザキャンバスモデルを保存する
TON_ADJ	コンカレントファイルを再作成する
TON_ADD	キャンバスモデルにメンバーを追加する
TON_APD	実データをメンバーに変換する
TON_DEL	メンバーを削除する
TON_UPDATE	メンバーを最新データに更新する
TON_SEL	アクティブにするメンバーを選択する
TON_FLE	アクティブメンバーを保存する
TON_FLE_CHK	保存するメンバーを指定する
TON_CANCEL	アクティブメンバーに加えた処理を全て放棄する
TON_DSP	ペアレント情報を表示する
TON_MOD	メンバーの表示／非表示を変更する
TON_MEMB_SUB	モデル中の全メンバーをサブモデルアイテムに変換する
TON_SUB_MEMB	モデル中の全サブモデルをメンバーアイテムに変換しコンカレントファイルを作成する
TON_SHOW_P	現在の「表示／非表示」の条件を保存する
TON_SHOW_L	「表示／非表示」の条件をロードする
TON_ADJ_MASTER	マスターコンカレントファイルを更新する
TON_ADJ_PRIVATE	プライベートコンカレントファイルを更新する
TON_DSP_MOD	メンバー表示のクラス／アイテム表示マスクを変更する
TON_VER	コンカレント情報をベリファイする

11.4.1 コンカレントファイルを新規作成する

【メニュー】

〔同時設計〕 → 〔マスタ作成〕

【構文】

TON_START name <CE>

新規作成するペアレント名を指定する。

name : ペアレント名を入力する。

新規作成を実行する。

<CE> : コマンドを実行する。

空のマスタコンカレントファイルができる。ファイルのディレクトリと拡張子は指定しなければ、キーワード #CONCURRENT# で指示した通りになる。

注) 全設計者に確認後、実行してください。

11.4.2 コンカレントを終了する

【メニュー】

〔同時設計〕 → 〔コンカレント終了〕

【構文】

TON_EXIT <CE>

コンカレントを終了する。

<CE> : コマンドを実行する。

11.4.3 アクティブメンバーの選択を中止する

【メニュー】

〔同時設計〕 → 〔編集取消〕

【構文】

TON_CANCEL <CE>

現在選択中のアクティブメンバーの処理を中止し、直前のメンバーアイテムの状態に戻す。

<CE> : コマンドを実行する。

11.4.4 ペアレントを呼び出す

【メニュー】

〔同時設計〕 → 〔呼出し〕

【構文】

TON_CALL [MASTER] name <CE>

マスターコンカレントファイルを指定する。

MASTER: プライベートコンカレントファイルではなく、マスターコンカレントファイルからペアレントを呼び出す。

呼び出すペアレント名を指定する。

name : 呼び出すペアレント名を入力する。ワイルドカード "*" を入力すると指定マスタファイルのリストが表示される。
リスト上をクリックするとそのペアレント名が使用される。

呼び出しを実行する。

<CE> : コマンドを実行する。
指定したコンカレントファイルに従って、新規モデルにメンバーを配置しキャンバスモデルを再構築する。
プライベートコンカレントファイルが存在しない場合は、自動的に作成される。

11.4.5 ユーザキャンバスモデルを読み込む

【メニュー】

〔同時設計〕 → 〔モデル読み込み〕

【構文】

TON_MDL_READ name <CE>

読み込むペアレント名を指定する。

name : 読み込むペアレント名を入力する。ワイルドカード "*" を入力するとプライベートコンカレントファイル名のリストが表示される。
リスト上をクリックするとそのペアレント名が使用される。
実際のモデルファイル名は " ペアレント名__ユーザ名.CON_MDL" となる。

読み込みを実行する。

<CE> : コマンドを実行する。
ユーザキャンバスモデルファイル " ペアレント名__ユーザ名.CON_MDL" がいない場合、以下を実行する。

- (1) プライベートコンカレントファイル " ペアレント名__ユーザ名.CON_WRK" が存在する時、プライベートコンカレントファイルに基づきコマンドを実行する。
- (2) プライベートコンカレントファイルが存在しない時、マスターコンカレントファイルに基づきコマンドを実行する。
このとき、全メンバーは表示 (show) 状態になる。

※ 読み込みが正常に終了した後にマクロ STARTUP05.MAC を実行します。該当するマクロがない場合はマクロを実行しません。

Advance CAD の起動オプション -m でマクロ名を指定することもできます。起動オプションについては「システム管理者の手引き」の「2.1.2 起動オプション」をご覧ください。

11.4.6 ユーザキャンバスモデルを保存する

【メニュー】

〔同時設計〕 → 〔モデル書込み〕

【構文】

TON_MDL_WRITE name <CE>

実際のファイル名は " ペアレント名__ユーザ名 .CON_MDL" となる。

書き込みを実行する。

<CE> : コマンドを実行する。

非表示 (No show) メンバーもユーザキャンバスモデルファイルに保存される。

※ 書き込みが正常に終了した後にマクロ STARTUP06.MAC を実行します。該当するマクロがない場合はマクロを実行しません。

Advance CAD の起動オプション -m でマクロ名を指定することもできます。

起動オプションについては「システム管理者の手引き」の「2.1.2 起動オプション」をご覧ください。

11.4.7 コンカレントファイルを再作成する

【メニュー】

〔同時設計〕 → 〔次頁〕 → 〔マスタ更新〕

【構文】

TON_ADJ name <CE>

メンバーの位置・ピクチャ・クラス・レビジョンの変更などをおこなった場合は、プライベートコンカレントファイルとくい違いが発生する。それを解消するために使う。

再作成するペアレント名を指定する。

name : ペアレント名を入力する。省略時は現在のペアレント名になる。

再作成を実行する。

<CE> : コマンドを実行する。

キャンバスモデルから、指定コンカレントファイルを作り直す。また、コンカレントファイル中のメンバーは、モデル名・ピクチャ名で昇順に並べかえられる。

11.4.8 キャンバスモデルにメンバーを追加する

【メニュー】

〔同時設計〕 → 〔メンバ追加〕

【構文】

TON_ADD	$\left[\begin{array}{c} \text{CHK s} \\ \text{MULT} \end{array} \right]$	name	$\left[\begin{array}{c} \text{s P} \\ \text{ALLPIC} \end{array} \right]$	<CE>
---------	---	------	---	------

同一メンバーの複数配置を可能にするとき

MULT : 同一メンバーの複数配置を可能にする。

配置するピクチャの最低アイテム数を指定する。

CHK s : 配置するピクチャの最低アイテム数を指定する。ピクチャのアイテム数がこの値より少ないと、そのピクチャは配置できない。

モデル名を指定する。

name : モデル名を入力する。

メンバーのピクチャ番号を指定する。

s : 読み込むピクチャ番号 (1 ~ 255) を入力する。
省略時はピクチャ番号 1 になる。

メンバーの配置点を指定する。

P : 配置点をテンポラリポイントで入力する。省略時は原点 (0, 0) となる。

モデル中の全てのピクチャを一括追加するとき

ALLPIC: モデルのそれぞれのピクチャをペアレントのそれぞれのピクチャの原点 (0, 0) に配置する。

メンバーの追加を実行する。

<CE> : コマンドを実行する。

11.4.9 実データをメンバーに変換する

【メニュー】

〔同時設計〕 → 〔メンバ変換〕

【構文】

TON_APD name [P] [s] $\left[\begin{array}{l} \text{ALLPIC} \\ \text{USEACT} \end{array} \right] \langle \text{CE} \rangle$

メンバーにするデータに付けるモデル名を指定する。

name : モデル名を入力する。

メンバーの配置点を指定する。

P : 配置点をテンポラリポイントで入力する。省略時は原点 (0, 0) となる。

メンバーのピクチャ番号を指定する。

s : 書き込みピクチャ番号 (1 ~ 255) を入力する。
省略時はピクチャ番号 1 になる。

メンバーにするデータを指定する。

ALLPIC: アクティブピクチャの全てのアイテムを選択する。

USEACT: アクティブリスト中のアイテムを使用する。省略時は、メンバー以外の実データを選択する。
ACT 矩形・ACT 選択では実データ以外選択できない。

コマンドを実行する。

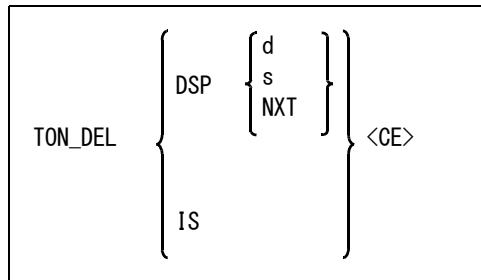
<CE> : メンバー変換を実行する。

11.4.10 メンバーを削除する

【メニュー】

〔同時設計〕 → 〔メンバ削除〕

【構文】



一覧表を表示する場合

DSP : メンバー一覧を表示する。
 メンバー数が多く一度に表示できないときは、次の修飾子を入力する。

UPP : 前頁を表示する。

DWN : 次頁を表示する。

削除したいメンバーを以下のどれかで指示する。選択された行の先頭に ">>" を表示する。

d : メンバー名をピックアップする。

s : 番号を入力する。

NXT : 選択メンバーの次のメンバーを選択する。

一覧表を表示しない場合

IS : メンバーをピックアップする。

メンバーの削除を実行する。

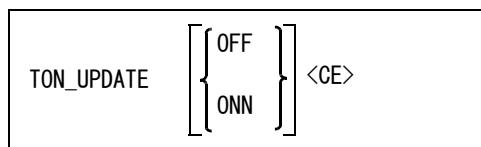
<CE> : コマンドを実行する。

11.4.11 全メンバーを更新する

【メニュー】

〔同時設計〕 → 〔メンバ更新〕

【構文】



アクティブモデル内の全メンバーが、最新のモデルを反映するようにする。

OFF : モデルとメンバーの最新更新日付を比較せずにメンバーを更新する。

ONN : モデルの更新日付がメンバーより新しい場合のみメンバーを更新する。

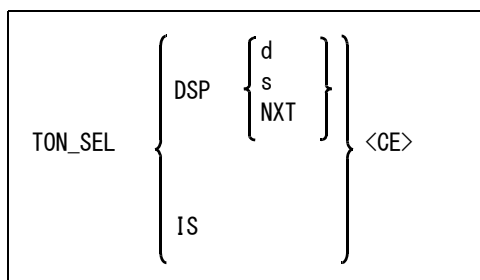
※ 他のユーザがメンバーを更新した後、自分のメンバーを最新の状態にするには、TON _
 UPDATE コマンドを使用します。

11.4.12 アクティブにするメンバーを選択する

【メニュー】

〔同時設計〕 → 〔メンバ選択〕

【構文】



同一グループのメンバー全てをアクティブにする

MULT : 同一グループのメンバー全てをアクティブにする

一覧表から選択する場合

DSP : メンバー一覧を表示する。

メンバー数が多く一度に表示できないときは、次の修飾子を入力する。

UPP : 前頁を表示する。

DWN : 次頁を表示する。

アクティブにするメンバーを以下のどれかで指示する。選択された行の先頭に ">>" を表示する。

* * の行: 他の設計者が修正中のメンバーを選ぶと次のメッセージが出る。

「メンバーは現在使用中ですので注意して下さい」

+ の行: アクティブメンバーを選ぶと次のメッセージが出る。

「選択されたメンバーは実データです」

d : メンバー名をピックする。

s : 番号を入力する。

NXT : 選択メンバーの次のメンバーを選択する。

一覧表を表示しない場合

IS : メンバーをピックする。

メンバーの選択を実行する。

<CE> : コマンドを実行する。

現在すでにアクティブなメンバーがあれば先にそれを保存する。

つづいて、選択されたメンバーを分解して、アクティブモデルのアイテムにする。選択されたメンバー名の前に "+" が表示される。

※ "コンカレント保存時にモデルを更新する" 設定になっている場合には自動的にメンバーの更新が行われますが、"更新しない" ではユーザがメンバーの更新を指定する必要があります。"更新しない" ではアクティブメンバーの変更時に現アクティブメンバーをモデルファイルに保存しません。

11.4.13 アクティブメンバーを保存する

【メニュー】

〔同時設計〕 → 〔メンバ保存〕

【構文】

TON_FLE <CE>

変更のあったメンバーを保存します。

TON_FLE_CHK コマンドで保存するメンバーを選択することが出来ます。TON_FLE コマンド実行後には、変更ステータスが全てクリアされますので、注意してください。

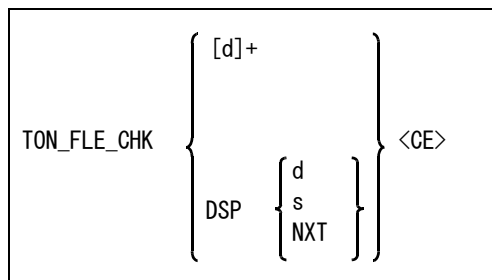
- ※ ”コンカレント保存時にモデルを更新する”設定になっている場合には自動的にメンバーの更新が行われますが、”更新しない”ではユーザがメンバーの更新を指定する必要があります。”更新しない”ではアクティブメンバーの変更時に現アクティブメンバーをモデルファイルに保存しません。
- ※ TON_FLE コマンドで変更メンバーをモデルファイルに保存して初めて、他のユーザは変更された最新のメンバーにアクセスできるようになります。

11.4.14 保存するメンバーを指定する

【メニュー】

〔同時設計〕 → 〔既存メンバ選択〕

【構文】



TON_FLE コマンドで保存するメンバーのリストを変更します。

一覧表を表示する場合

- DSP : 変更されたメンバーの一覧を表示する。
 メンバー数が多く一度に表示できないときは、次の修飾子を入力する。
- UPP : 前頁を表示する。
 DWN : 次頁を表示する。

メンバーを指定する。保存するメンバーの行頭に " ** " を表示している。選択すると " ** " が消える。もう一度選択すると " ** " が表示される。

- d : メンバー名をピックする。
 s : 番号を入力する。
 NXT : 選択メンバーの次のメンバーを選択する。

コマンドを実行する。

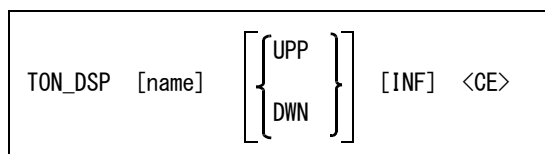
- <CE> : コマンドを終了する。

11.4.15 ペアレント情報を表示する

【メニュー】

〔同時設計〕 → 〔マスタ表示〕

【構文】



情報を表示したいペアレント名を指定する。

name : ペアレント名を入力する。省略時は現在のペアレントの情報を表示する。

メンバー数が多く一度に表示できないときは、次の修飾子を入力する。

UPP : 前頁を表示する。

DWN : 次頁を表示する。

情報をファイルに出力する。

INF : ファイル名はペアレント名と同じ。

ディレクトリは指定しなければ、#CONCURRENT# で指示したディレクトリ、拡張子は “.LST” となる。

ペアレント情報の表示を実行する。

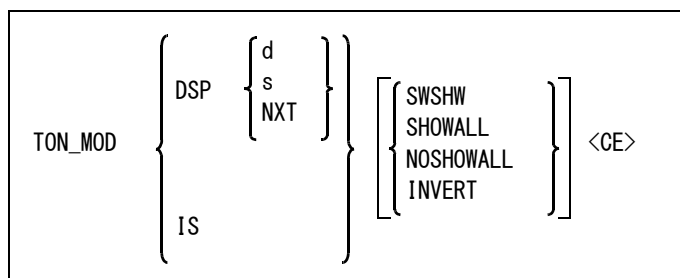
<CE> : コマンドを実行する。

11.4.16 メンバーの表示／非表示を変更する

【メニュー】

〔同時設計〕 → 〔表示変更〕

【構文】



一覧表を表示する場合

DSP : メンバー一覧を表示する。

メンバー数が多く一度に表示できないときは、次の修飾子を入力する。

UPP : 前頁を表示する。

DWN : 次頁を表示する。

アクティブにするメンバーを以下のどれかで指示する。選択された行の先頭に ">>" を表示する。

d : メンバー名をピックアップする。

s : 番号を入力する。

NXT : 選択メンバーの次のメンバーを選択する。

一覧表を表示しない場合

IS : メンバーをピックアップする。

メンバーの表示状態を指定する。

SWSHW : 指定されたメンバーを表示にする。もう一度入力すると非表示になる。

SHOWALL : 全てのメンバーを表示にする。

NOSHOWALL : 全てのメンバーを非表示にする。

INVERT : 表示を反転する。

表示／非表示の変更を実行する。

<CE> : コマンドを実行する。

11.4.17 メンバーアイテムをサブモデルに変換する

【メニュー】

〔同時設計〕 → 〔次頁〕 → 〔メンバーサブ〕

【構文】

TON_MEMB_SUB <CE>

モデル中の全メンバーアイテムを同じ名前のサブモデルに変換する。

注) メンバーが大きい場合、同一名称の複数のサブモデルに変換される場合がある。

11.4.18 サブモデルをメンバーに変換する

【メニュー】

〔同時設計〕 → 〔次頁〕 → 〔サブ→メンバ〕

【構文】

TON_SUB_MEMB name <CE>

モデル中の全サブモデルをメンバーに変換し、コンカレントファイルを作成する。

name : コンカレントファイル名を指定する。

注) 同一名称、同一ピクチャのサブモデルが複数配置されている場合、2 番目以降のサブモデルは変換対象にならない。

11.4.19 表示／非表示の条件を表示コントロールファイルに保存する

【メニュー】

〔同時設計〕 → 〔表示状態保存〕

【構文】

TON_SHOW_P n <CE>

メンバーの表示／非表示の条件を表示コントロールファイルに保存する。

n : 表示コントロールファイルのセット番号を入力する。(1 ～ 99) 表示コントロールファイル名は “コンカレントファイル名.Sn” となる。例えばコンカレントファイル名が CTC.CON でセット番号に 1 を指定すると、表示コントロールファイル名は “CTC.S01” となる。

11.4.20 表示コントロールファイルをロードする

【メニュー】

〔同時設計〕 → 〔表示状態呼出〕

【構文】

```
TON_SHOW_L n <CE>
```

コントロールファイルのセット番号を指定する。

n : セット番号を入力する。このセット番号に保存された表示／非表示の条件をロードする。

11.4.21 マスターコンカレントファイルを更新する

【メニュー】

〔同時設計〕 → 〔次頁〕 → 〔マスタファイル更新〕

【構文】

```
TON_ADJ_MASTER <CE>
```

マスターコンカレントファイルを更新する。

プライベートコンカレントファイルとマスターコンカレントファイルとの相違点を一覧表示する。一覧表の上の更新するメンバーを選択し、チェックを付ける。ここで選択したメンバーだけがマスターコンカレントファイルで更新される。

<CE> : コマンドを実行する。

● プライベートコンカレントファイルを更新する

【メニュー】

〔同時設計〕 → 〔次頁〕 → 〔プライベート更新〕

【構文】

```
TON_ADJ_PRIVATE <CE>
```

プライベートコンカレントファイルを更新する。

マスターコンカレントファイルとプライベートコンカレントファイルとの相違点を一覧表示する。一覧表の上の更新するメンバーを選択し、チェックを付ける。ここで選択したメンバーだけがプライベートコンカレントファイルで更新される。

<CE> : コマンドを実行する。

11.4.22 メンバー表示のクラス／アイテム表示マスクを変更

【メニュー】

〔同時設計〕 → 〔表示マスク変更〕

【構文】

```
TON_DSP_MOD INF maskdata [ IS ]+ <CE>
```

メンバーがメンバーアイテムとして表示されるマスクを設定する。
メンバーをアクティブにしている修正中は、このマスクは使用されません。

マスクを変更するメンバーを選択する。

IS : メンバーを選択する。

表示マスクを設定する。

INF : メンバーに設定する表示マスクを指定する。TON_VER コマンドの説明を参照のこと。

11.4.23 コンカレント情報のベリファイ

【メニュー】

〔同時設計〕 → 〔次頁〕 → 〔メンバーベリファイ〕

【構文】

TON_VER [IS]+ <CE>

コンカレント情報のベリファイ。
メンバーの表示マスクの確認および設定ができます。

IS : ベリファイするアイテムを指定する。

メンバーアイテムを選択すると、このメンバーアイテムに設定した表示マスクを表示します。アイテムタイプ表示マスクとアイテムクラス表示マスクを表示します。表示は 1、非表示は 0 で示します。

マスク設定を変更するには、この値をクリックします。値 1 が 0 に、0 が 1 に反転します。

「幾何図形 ON」、「幾何図形 OFF」、「全クラス ON」などをクリックし一括設定する方法もあります。

また、クラスの範囲を指定して設定するには「CLS ON From/to」、「CLS OFF From/to」をクリックします。次に開始と終了クラスの番号を入力します。クラス 1 0 1 から 1 2 0 を非表示にするなら次のようにします。

「CLS OFF From/to」をクリック

101

120

11.5 同時設計作業中のメンバーアイテムの選択

編集コマンドや削除コマンドなどほとんどのコマンドはメンバーアイテムは「選択不可」にしています。これらのコマンドでメンバーアイテムを選択する場合は、アイテムを選択する前にメンバーアイテムを「選択可」にしなければなりません。

メンバーアイテムの選択の許可は、許可されたコマンドの実行中のみ有効です。コマンドが切り換わると無効になります。

下記のコマンドはコマンド選択時に自動的にメンバーアイテムを「選択可」にしています。

TON_SEL、TON_APD、TON_DEL、TON_MOD、TON_DSP_MOD、TON_VER
VER、VER/SEG、VER/CNR
MES、MES/SEG、MES/LEN、MES/AREA、MES/ANG、MES/VOL、MES/CARD

上記以外のコマンドはコマンド選択時にメンバーアイテムを「選択不可」にしています。

メンバアイテムを選択したい場合は、メンバアイテムを「選択可」にしてからアイテムを選択します。MOVE コマンドでメンバーアイテムの位置を変更するならば、次のように明示的にメンバーアイテムを「選択可」にします。

```
MOVE @DX50 MEMB/SWITCH <dig> <CE>
```

このようにしてメンバーアイテムの位置を変更できますが、これはキャンパスモデル上だけであって、プライベートコンカレントファイルには反映しません。TON_ADJ コマンドを使ってプライベートコンカレントファイルを再作成できます。

11.5.1 メンバアイテムの選択の可否を切り換え

【メニュー】

〔メンバ〕

【構文】

MEMB/SWITCH

メンバアイテムが「選択不可」ならば「選択可」に、「選択可」ならば「選択不可」に切り換える。

※ つぎのコマンドはマクロで使用する则便利です。

【構文】

MEMB/ONN

メンバアイテムを選択可にする。

【構文】

MEMB/OFF

メンバアイテムを選択不可にする。

第 12 章 モデルダウンコンバータ

● 概要

バージョン 20 のモデルファイルは、バージョン 19 以下の Advance CAD では読込むことができません。

モデルダウンコンバータは、バージョン 20 のモデルファイルをバージョン 14 ～ 19 のモデルファイルに変換します。

バージョン 20 ～ 22 は同じモデル構造の為、ダウンコンバートは不要です。

12.1 ダウンコンバータ

12.1.1 使用方法

12.1.1.1 コマンド入力に変換する場合

コマンドプロンプトウインドウを開きます。
モデルがあるディレクトリに移動します。
次のプログラムを実行します。
実行できない場合は、プログラムをフルパスで入力してください。
例えば、C:\acad\exe\mdlconv と入力してください。

mdlconv [-バージョン番号] [モデルファイル名] <cr>

バージョン番号 : 変換後のモデルファイルのバージョン番号 (19,18,17,16,15,14)
省略するとバージョン 19 のモデルファイル構造に変換します。
Ver16 は Ver15 と同じモデル構造です。

モデルファイル名 : 変換するモデルファイル名 (ワイルドカード指定ができます)
Ver20 ~22 のモデルファイルのみ変換対象とします。
その他のバージョンは処理しません。

例)

Advance CAD バージョン 19 のモデルファイル構造に変換する場合は以下のようになります。
prompt% mdlconv -19 MODEL.MDL<cr>

注)

変換が正常終了するとモデルファイルは変換後のモデルとなります。
元のモデルファイルは、ファイル名の後に .OLD がついたファイル名になります。

12.1.1.2 ダイアログで変換する場合

コマンドプロンプトウインドウで次のプログラムを実行します。
実行できない場合は、プログラムをフルパスで入力してください。
例えば、C:\acad\exe\mdlconv_dlg と入力してください。
mdlconv_dlg

ダウンコンバータの変換パラメータ設定ダイアログが表示されます。

注) その他の起動方法

- 1) エクスプローラを開いてプログラムがあるホルダーに移動します。
プログラム mdlconv_dlg.exe をダブルクリックします。
- 2) モデルファイルをアイコンにドラッグ&ドロップします。

変換パラメータ設定ダイアログ



モデルファイルの場所、変換後モデルファイルの場所、変換バージョン等を指定します。

モデルファイルについて指定します。

〔場所〕

『参照』

☐ モデル一覧で Ver20 以外の
モデルは表示しない

変換するモデルがある場所を入力します。

変換するモデルがある場所を選択します。

モデル一覧で Ver20 以外のモデルは表示しません。

変換後モデルファイルについて指定します。

☐ モデルファイルと同じ場所

変換後モデルファイルをモデルファイルと同じ
場所に作成します。

変換後、モデルファイルの拡張子は .MDL. OLD に
なります。変換後モデルファイルが .MDL に
なります。

☐ 任意の場所

変換後モデルファイルを任意の場所に作成します。

〔場所〕

変換後のモデルの場所を入力します。

『参照』

変換後のモデルの場所を参照します。

注) モデルファイルと同じ場所は
指定できません。

バージョン

☐ 19

Ver19 のモデルに変換します。

☐ 18

Ver18 のモデルに変換します。

☐ 17

Ver17 のモデルに変換します。

☐ 16, 15

Ver15 のモデルに変換します。

☐ 14

Ver14 のモデルに変換します。

既にモデルがある場合

☐ 変換しない

変換しません。

☐ 上書きする

変換して置き換えます。

☐ .OLD にする

既にあるモデルを .OLD を付けた名前に変更してか
ら変換します。

『ログ設定』

変換ログファイルの場所を指定します。

ログは、ファイル名 mdlconv. log に出力します。

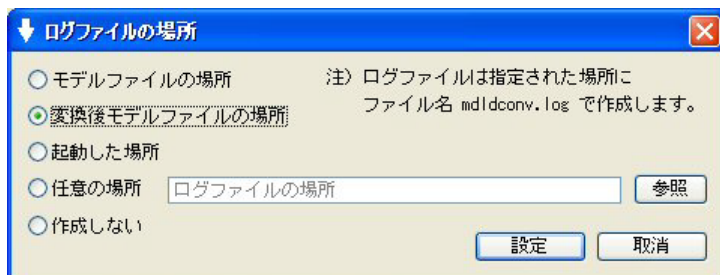
『次へ』

モデル一覧ダイアログを表示します。

『終了』

処理を終了します。

ログファイル設定ダイアログ

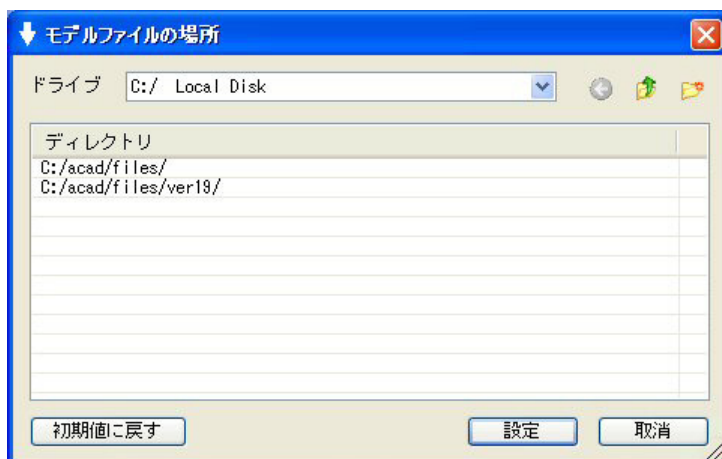


ログファイルを作成する場所を指定します。

- ☐ モデルファイルの場所
- ☐ 変換後モデルファイルの場所
- ☐ 起動した場所
- ☐ 任意の場所
- 『参照』
- ☐ 作成しない

モデルファイルがある場所。
 変換後モデルファイルの場所。
 本プログラムを起動した場所。
 任意の場所を入力します。
 任意の場所を選択します。
 ログファイルは作成しません。

参照ダイアログ



ディレクトリを選択します。

▼ドライブ

ドライブを選択します。
 ドライブのルートディレクトリが選択されます。

[ディレクトリ]

ディレクトリを選択します。
 一番上の行に表示されているディレクトリが現在
 選択されている場所です。
 一番上を選択すると1つ上のディレクトリになりま
 す。それ以外は選択したディレクトリになります。

『初期値に戻す』

初期値に戻します。

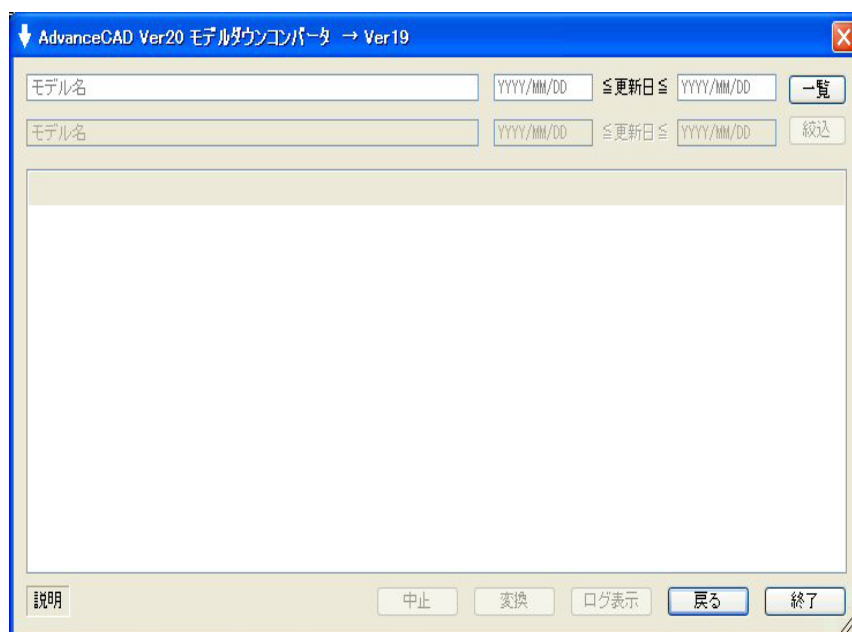
『設定』

選択したディレクトリを設定します。

『取消』

処理を取り消します。

モデル名一覧ダイアログ



変換するモデル名の一覧を表示します。

[モデル名]

一覧表示するモデル名を入力します。

[更新日 ≤]

一覧表示するモデルの更新日 ≤ を YY/MM/DD の形式で入力します。

更新日は、ファイルの更新日です。

[≤ 更新日]

一覧表示するモデルの ≤ 更新日を YY/MM/DD の形式で入力します。

『一覧』

[モデル名]、[更新日 ≤]、[≤ 更新日] の条件に合うモデルの一覧を表示します。

[モデル名]

絞り込むモデル名を入力します。

[更新日 ≤]

絞り込むモデルの更新日 ≤ を YY/MM/DD の形式で入力します。

[≤ 更新日]

絞り込むモデルの ≤ 更新日を YY/MM/DD の形式で入力します。

『絞り込』

現在表示されている一覧から [モデル名]、[更新日 ≤]、[≤ 更新日] の条件に合うモデルの一覧を表示します。

[説明]

モデル一覧の表示項目についての説明を表示します。

『中止』

変換処理を中止します。

『変換』

変換を開始します。

『ログ表示』

変換結果のログを表示します。

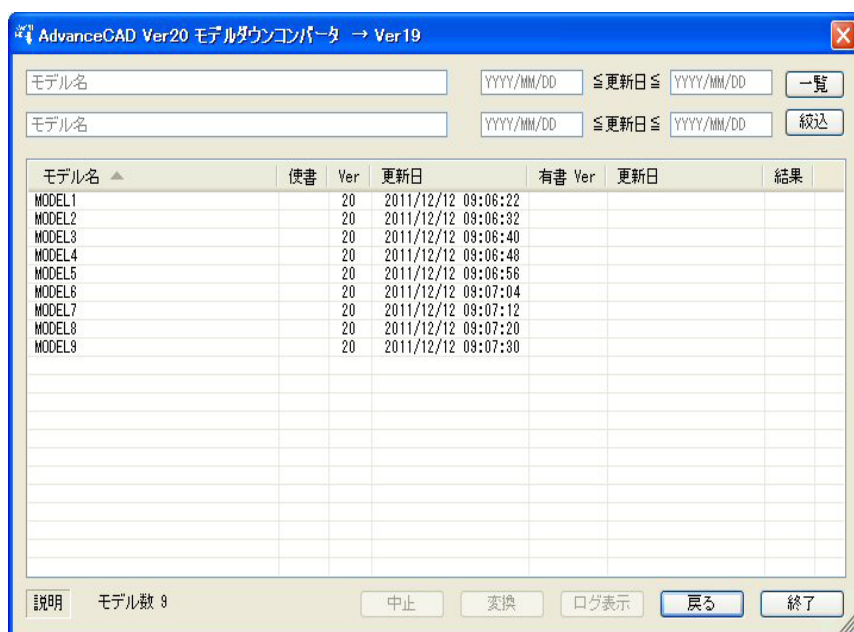
『戻る』

変換パラメータ設定ダイアログへ戻ります。

『終了』

処理を終了します。

モデル名一覧ダイアログ：モデルの一覧



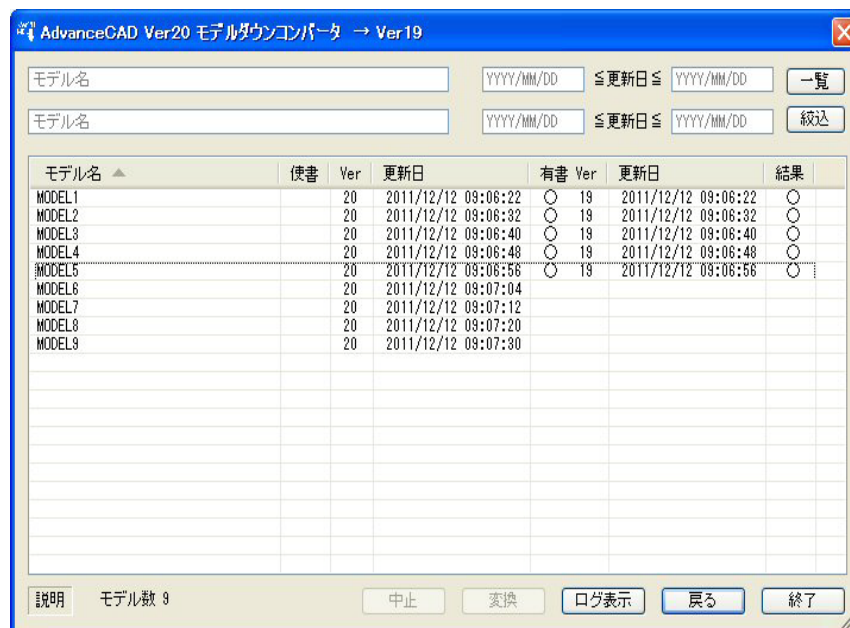
『一覧』でモデル一覧を表示した例です。

モデル名一覧ダイアログ：変換するモデルを選択



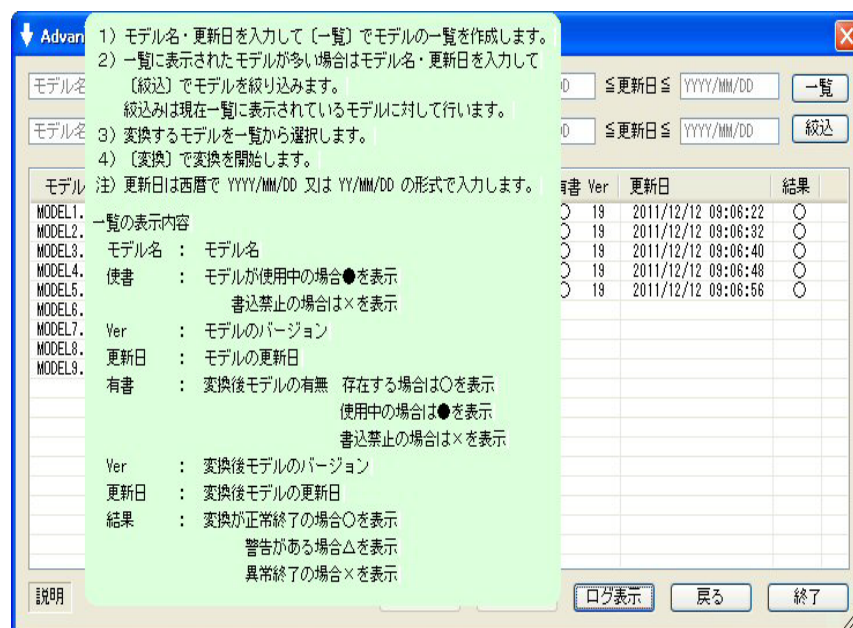
変換するでモデルを選択した例です。
注) 変換できないモデルは、選択できません。

モデル名一覧ダイアログ：変換処理終了



変換結果の例です。

モデル名一覧ダイアログ：一覧の項目の説明表示



モデル一覧の表示項目の説明を表示した例です。

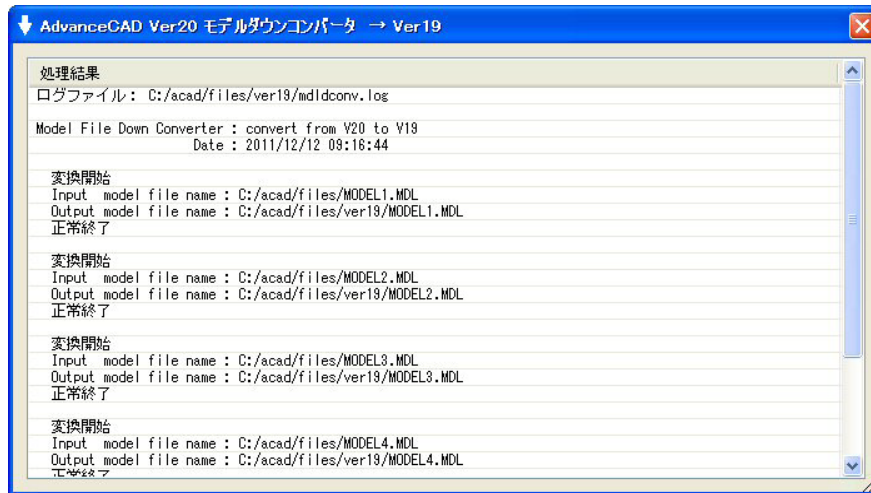
一覧表示項目

使書	使：モデルが使用中の場合は、●を表示。 書：モデルが書込み禁止の場合は、×を表示。
Ver	モデルのバージョン。
更新日	モデルの更新日時。
有書 Ver	有：変換後モデルが存在する場合は、○を表示。 使用中の場合は、●を表示。

更新日
結果

書：変換後モデルが書き込み禁止の場合は、×を表示。
Ver：変換後モデルのバージョン。
変換後モデルの更新日時。
変換が正常終了の場合は、○を表示。
警告がある場合は、△を表示。
異常終了の場合は、×を表示。

変換結果のログの表示



変換結果のログを表示した例です。

12.1.2 制限事項

[Ver19 に変換する場合]

製図図形関係

文字列の 2 重下線の間隔、打消し線の間隔及び端部延長量は失われます。

SXF 関係

属性及び表題欄属性は変換されません。(等高線属性は変換されます。)

図面配置関係

図面配置の背景色は変換されません。(ピクチャの背景色になります。)

[Ver18 に変換する場合]

図面配置関係

図面配置ページ番号 1 ～ 256 は 1 ～ 64 に変換します。

ページ総数が 64 を超える場合、変換は行いません。

配置されたウィンドウ総数が 256 を超える場合、変換は行いません。

配置された図面タイトル数が 3276 を超える場合及び図面タイトルの文字列の総数が 32760 を超える場合は、変換は行いません。

ウィンドウ関係

ウィンドウ総数が 512 を超える場合、変換は行いません。

イメージ

JPEG イメージは変換されません。

SXF 関係

表題欄は変換されません。

等高線は構成要素の図形に変換されます。

ペン割付け

ペンの数 256 は 255 に変換します。
 クラス、レビジョン、アイテム、線種および線幅のペン割付けで
 ペン番号 256 は 255 に変換します。

注) Ver19 の制限事項も含まれます。

[Ver17 に変換する場合]

図面名の文字数が 32 バイトを超えた場合は削除します。
 ピクチャ番号 1 ～ 320 は 1 ～ 63 に変換します。
 ピクチャの総数が 63 を超える場合、変換は行いません。
 クラス番号 256 は 255 に変換します。
 レビジョン番号 1 ～ 256 は 1 ～ 127 に変換します。レビジョン総数が 127 を超えた場合、
 127 を超えた分は 127 に変換します。
 ツールタイプ文字はストローク文字に変換します。
 半角カタカナは全角カタカナに変換します。
 ラスターイメージは 63 を超えた分は削除します。
 色コード 201 ～ 256 は 200 に変換します。
 変換後のサブモデル配置は、サブモデルの更新処理ができません。

SXF に関する以下の情報は失われます。

- SXF 線種名は削除
- SXF 線幅は削除
- SXF 文字フォント名は削除
- SXF ピクチャスケールは削除
- SXF ピクチャ座標系は削除
- SXF 複合図形（作図グループ・作図部品）は 複合アイテムに変換

注) Ver19、Ver18 の制限事項も含まれます。

[Ver15、Ver16 に変換する場合]

以下の項目は 名前 の長さが 32 バイト以下 の場合はそのままです。
 32 バイトを超えた場合は削除します。
 ピクチャ名、クラス名、レビジョン名

注) Ver19、Ver18、Ver17 の制限事項も含まれます。

[Ver14 に変換する場合]

以下の項目は 名前 の長さが 6 バイト以下 の場合はそのままです。
 6 バイトを超えた場合は削除します。
 ピクチャ名、クラス名、レビジョン名、スクリーンレイアウト名、ビューポート名、
 ウィンドウ名、ドローイングレイアウト名、図面枠シンボル名

モデルタイトルの総文字数が 8192 バイトを超えた場合は、超えた分のタイトルは削除します。

頁タイトルの総数が 1024 個を超えた場合、または頁タイトルの総文字数が 16384 を超えた場合は、超えた分のタイトルは削除します。

線グリッドのグリッドライン総数が 1024 本を超えている場合は、超えた分のグリッドラインは削除します。

角度をつけて傾けた矩形で設定されたウィンドウは、その矩形を包み込む水平な矩形に変換します。

注) Ver19、Ver18、Ver17、Ver15 の制限事項も含まれます。

#APG# 2
A
ACADOTHERMDM. SET 140
ACADSXF. TBL 307
acos 403
acosh 404
ACT/ASC 57
actlst 414
APG 1
APG/DSP 15
APG2DISP 181
APGADJUST 19
APGANAL 9
APGBATCH 42
apgconv 1
APGDELETE 173, 181
APGDISP 172
APGFILE. MDM 142
APGORG 20
APGPARAM 10
APG_PRECLR 33
APG_PREDEL 32
APG_PRESET 21, 22, 23, 24, 25, 26, 27, 28, 29, 30
APG_PREVER 32
APGREGEN 15
APGSET 13
APGTRACE 17
APX 2
array 410
ASC/LIST 57
ASC_ADD 54
ASC_BRK 55
ASC_CNG 56
ASC_DEL 56
ASC_MOD 61
ASC_MOVE 61
ASC_NEW 54
ASC_REL 55
ASC_SCALE 63
ASC_SET 58
ASC_STRETCH 62
ASC_SUB 60
ASC_VER 57
asin 403
asinh 404
associate item 51
atan 403
atan2 403
atanh 404

B

break 402

C

CASSETNAME. MDM 142

CHGNAME 202

CHGOWNER 165

clear 409

Concurrent Design 461

continue 401

cos 403

cosh 404

CREMDLINF 189

CSTDELETE 197

CSTLEN 197

D

DATA 144, 148, 151

dbchgd 420

DISCARD 176

DISCARDMODEL. MDM 142

DLGMODE 442

dms 404

do 400

DOC/EPS 385

DOC/RAS 382

DOC/RAS2 383

DRAW/RAS 384

DRWSTD 250

DRWTYP 250

E

echo 403

else 401

else if 401

encode 406

EPS 385

error 410

exit 402

extern 391, 430

F

fclose 415

fgets 415

FILDELETE 202

FILEMN 182

fname 416

fopen 415

for 388, 400

FORM 150

fputs 415

fseek 415

fstat 416

ftell 416

G

getany 411
getclr 422
getdig 411
getdrw 422
getenv 429
getitm 419
getmsk 421
getmtx 422
getnum 410
getpnt 411
getpos 411
getprm 423
gettpt 411
gettxt 411
goto 402
H
HOLE 125
I
idptr 413
if 401
IGES 211
IGES/READ 212
IGES/WRITE 213, 216
Illustrator 385
inch 404
INFDISP 162
INFUPDATE 165
int 404
L
length 392, 409
LIST/OFF 443
LIST/ONN 443
log 404
log10 404
lower 404
LPR 167
M
max 404
MAXCOL 143, 147, 150
MAXLIN 150
mcheck 445
mclose 417
mcolor 417
MDL2DISP 174
MDL3DISP 177
mdlconv 488
mdlconv_dlg 488
MDLDELETE 167, 175, 178
MDLDISP 159
MDLDRAW 163

MDLDUMP 191
MDLLIST 196
MDLLOAD 194
MDLPLOT 199
MDLREAD 164
mecho 417
MEMB/OFF 485
MEMB/ONN 485
MEMB/SWITCH 485
menu 429
MENU/OFF 443, 444
MENU/ONN 443, 444
merase 417
mgrid 418
min 404
MKHOLE 127
MODELDATE. LMG 142
MODELDATE. MDM 142
MODEL_DISPLAY_xxx. MDM 142
MODELFILE. LMG 142
MODELFILE. MDM 141
MODEL_LIST_TEMP. MDM 142
MODEL_SORT_KEY. MDM 142
MODELTITLE. LMG 142
MODELTITLE. MDM 142
mopen 417
msym 418
msympup 419
N
nint 404
O
OLDAPG. MDM 142
OLDMODEL. LMG 142
OLDMODEL. MDM 142
OLDSYMBOL. MDM 142
P
PAGEBIT. LMG 142
PAGEBIT. MDM 142
PAGETITLE. LMG 142
PAGETITLE. MDM 142
PARTINF. LMG 142
PARTINF. MDM 142
PARTS 115
partsgen 98
PASCCLR 123
PASCLIST 124
PBREAK 122
PDIM 122
pick 414
PLOTMN 201

PMOVE 121
PRTDEL 120
PRTGRP 119
PRTUPD 119
PSET 117
PSETBRK 123
psetgen 108
PSIZE 121
PTLDISP 162
PVER 120
R
RECOVER 175, 177
REFCREATE 188
REFINDEX. LMG 142
REFINDEX. MDM 142
REFLIST 186
REFMODEL. LMG 142
REFMODEL. MDM 142
return 402
rot 404
RPT/OFF 442
RPT/ONN 442
S
SCREEN 143, 147
SELDEV 201
sin 403
sinh 404
sort 413
SPC_CATVER 86
SPEC_ADD 83
SPEC_BIAS 86
speccheck 77
SPEC_CHK 95
SPEC_CLR 83
SPEC_COPY 85
SPEC_DEL 84
SPEC_DSP 83
SPEC_ERS 91
SPEC_FLN 81
SPEC_GET 83
SPEC_GNT 89
SPEC_INP 82
SPEC_MOD 84
SPEC_NODATA 88
SPEC_NUM 81
SPEC_OUT 93
SPEC_RFN 90
SPEC_SEL 86
SPEC_TAG 90
SPEC_UPD 85

SPEC_VER 85
SPEC_WRT 92
sprint 404
sqrt 404
sscan 405
strstr 404
SUBMDLINF.LMG 142
SUBMDLINF.MDM 142
subs 407
switch 401
SXF 239
SXF/ATTR 278
SXF/ATTRSET 294
SXF/CNT 278
SXF/CONFIRM 298
SXF/HATCH_FIL 260
SXF/HATCH_PAT 259
SXF/HATCH_USR 258
SXF/LAY 279
SXF/LIST 264
SXF/MARKER 258
SXF/MOVE 274
SXF/OPTION 301
SXF/PAN 286
SXF/PIC_LIST 263
SXF/PIC_SCF 263
SXF/PREVIEW 290
SXF/PRINT 286
SXF/READ 251
SXF/RPT 284
SXF/SEQUENCE 275
SXF/SFIG_BRK 273
SXF/SFIG_DEL 273
SXF/SFIG_EDT 272
SXF/SFIG_GEN 269
SXF/SFIG_REF 267
SXF/TEMPLATE 249
SXF/TTL_CRE 277
SXF/TTL_SET 275
SXF/USER_CLR 261
SXF/USER_LFT 261
SXF/USER_LWT 262
SXF/VER 281
SXF/WRITE 254
SXF/ZOOM 284
SXF/ZOOMALL 285
SXF/ZOOMALLVIE 286
SXF/ZOOMDOWN 285
SXF/ZOOMUP 285
SXFACAD.TBL 303

SXFCOPY/PREVIEW 291
SXFCOPY/PRINT 288
SYM2DISP 179
SYMBOLFILE.MDM 142
SYMBOLINF.LMG 142
SYMBOLINF.MDM 142
SYMDELETE 170, 179
SYMDISP 169

T

tan 403
tanh 404
TAR 204
TITLE 144, 147, 150
TITLEBIT.LMG 142
TITLEBIT.MDM 142
TON_ADD 476
TON_ADJ 476
TON_ADJ_MASTER 483
TON_ADJ_PRIVATE 483
TON_APD 477
TON_CALL 475
TON_CANCEL 474
TON_DEL 478
TON_DSP 481
TON_DSP_MOD 483
TON_EXIT 474
TON_FLE 479
TON_FLE_CHK 480
TON_MDL_READ 475
TON_MDL_WRITE 476
TON_MEMB_SUB 482
TON_MOD 481
TON_SEL 479
TON_SHOW_L 483
TON_SHOW_P 482
TON_START 474
TON_SUB_MEMB 482
TON_UPDATE 478
TON_VER 484
TRAIL 151

U

upper 404

V

vdump 409
vtype 409

W

while 400
WRITEONOFF 166, 202

い

イメージファイルの作成 382, 488
え

エスケープ文字 392

お

親子関係 130

か

カーソル位置型 413

き

共通データファイル 323

さ

座標型 413

し

条件文 401

所有者名 130

す

スクリーン座標型 412

て

デジタイズ点型 413

テンプレート 249

テンプレートファイル 249

と

同時設計 461

は

パーツ A P G 形状ファイル 98

パーツ情報ファイル 97

パーツコントロールファイル 98

パーツセット 108

パーツ定義ファイル 98

パーツマスターファイル 98

パラメトリック図形 97

ふ

部品名 130

部分文字列 392

ほ

ホスト名指定 136

も

文字列の長さ 392

モデル情報ファイル 138, 142

り

リスト出力 150

リスト出力テンプレート 150

わ

ワイルドカード 131