PL-U4101C1 PcWaveForm
Archi_1 Script 言語仕様編

2016 年 4 月

Revision 1.07

お断り

記載されている会社名および製品名はその会社の所有する商標です。
記載された内容については事前のお断りなく変更させていただく場合がございます。
記載された内容は 2016 年 4 月現在のものです。
ご使用にあたっては、本取扱説明書の内容を十分お読みいただけますようお願い申し上げます。

本取扱説明書は、PDF 形式でプログラム CD の中に入っています。

株式会社 デイシー
〒198-0024
東京都青梅市新町 9-2190
電話: 0428-34-9860
メール: info@deicy.co.jp

※ Archi_1 Script とは PcWaveForm 上で記述し、任意の解析を行うことのできる機能を持った Script 言語体系の名称です。

改定履歴

発行日	Revision	内容
2010 年 3 月 20 日	1.00	初版 Script 文の増加に伴い PcWaveForm スクリプト機能編より Script 言語仕様部分を別冊としました。 同時に、新規 Script 文の追加および従来の機能強化を行いました。 <追加された構文> テキストファイル行 列数定義文 テキストファイル読み出し 文 テキストファイル格納文 テキストファ イル・セル書き込み文 参照チャネル local 属性定義文 ファイル存在確認 文 通知ダイアログ書き込み文 <機能強化した構文> ファイル番号定 義文、取扱属性の追加 <実行環境機能強化> 実行時エラーメッセージダイアログをノンモーダルからモーダル形 式に変更 <構文 bug 修正> ・マーク位置取得文でマーク番号に 0 を指定した場合の修正
2010 年 4 月 13 日	1.01	演算処理ブロック制御文 index 文、repeat_index 文、mark 文、repeat_mark 文の説明追加 “制御される演算処理ブロック内 に収録チャネルの記述が必要です”
2010 年 5 月 8 日	1.02	<修正された構文> 文字列検索文 検索方向の指定を可能としました。 文字列切り出し 文 切り出し方向指定を可能としました。 カレントフォルダ選択文 ダイ アログの表題表示を可能としました。 チェックボックス・ステータス取得文 ダイアログの表題表示を可能としました。 値入力 文 ダイアログの表題表示を可能としました。表示形式指定を可能としました。 演算処 理ブロック呼び出し文 引数に即値/文字列即値の使用を可能としました。 テキストファイル・セル読み出し文 同時に複数チャネルからの読み出しを可能としました。 結果シー ト・チャネル列書き込み文 チャネル列記述チャネルの Index 指定を可能としました。 メッセージ表 示文 数値と文字列を混在記述可能としました。 生成波形・Mark 情報定義文 引数の Index 指定を可能としました。 カレントフォルダ名取得文 フラグを追加し、カレントフォルダ内のフォルダ名取得を可能としました。 <追加された構文> 生成波形ファイル・チャネル番号定義文 波形ファイルのチャネル番号を定義可能としました。 ファ イル選択文 ファイル読み出しダイアログからファイル名の取得を可能としました。 テキスト表示選択 文 複数のテキストをダイアログで表示し、選択を可能としました。 ラジオボタンステータス取得文 ラ ジオボタンをダイアログで表示し、選択を可能としました。 文字列削除文 文字列から開始文字位置 および文字数を指定して削除可能としました。 文字列挿入文 文字列に文字列を挿入する事を可 能としました。 文字列置換文 文字列に含まれる文字列を置き換える事を可能としました。 Script 実行時環境宣言文 Script 実行メニューで設定する実行環境を Script 文での宣言を可能としました。 結果シート表題定義文 結果シートの表題を文字属性参照チャネルでの定義を可能としました。
2011 年 6 月 17 日	1.03	<追加された構文> グラフ番号定義文 グラフ表 示 X 軸定義文 グラフ表示 Y 軸定義文 グラフ表示チャネル番号定義文 グラフ 表示線種色定義文 グラフ描画文 グラフ格納文 グラフ枠内追加線定義 文 ファイル削除文 処理データ個数定 義文 バイナリファイル読み出し文 <機能追加された構文> テキストファイル・セル読み出し文 テキストフォームからの読み込み指定可能なフラグを追加しました。
2011 年 10 月 15 日	1.04	<変更された構文> バイナリファイル読み出し文 演算処理 ブロック呼び出し文 ボタンラベル宣言文
2012 年 7 月 27 日	1.05	<追加された構文> グラフ縦横比定義文 バイナリバッファ 定義文 バイナリファイル構造読み出し文 バイナリバッファ書き込み文 バイナリフ ァイル格納文 <機能追加された構文> 代入文 半角波括弧“{”,”}”で括った場合、改行を区切り文字として扱えるようにしました。 生成波形ファイル・データタイ プ定義文 int 形式指定時に SLOPE と OFFSET を指定可能としました。また、格納形式 指定方法を拡張しました。
2015 年 6 月 29 日	1.06	グラフ枠内文字列書き込み文
2016 年 4 月 21 日	1.07	ファイル番号定義文に grp 追記、グラフ格納文に説明追記

適用

本書(Archi_1 Script 言語仕様編)は、Archi_1 Script 言語仕様部分を別冊としたもので、各構文の仕様を説明します。
Archi_1 Script の記述方法については「Archi_1 Script 記述方法編」を、使用する演算関数については「Archi_1 Script 演算関数仕様編」をご参照下さい。

本書の記述内容は、下記のプログラムバージョンに対応します。

PcWaveForm　Ver.7.07 ～
PcWaveFormFANA Ver.5.37 ～
PcWaveFormWBV+ Ver.2.16 ～

PcWaveForm プログラムの取扱説明書構成は、下記の 5 部構成となっています。

- 基礎編
- 解析機能操作編
- Archi_1 Script 演算関数仕様編
- Arch_1 Script 言語仕様編(本書)
- Archi_1 Script 記述方法編

の5部構成となっています。

PcWaveForm プログラムの基本操作については基礎編を、解析機能の操作については解析機能操作編を参照下さい。

ご注意

- 本書は万全を記して作成しておりますが、万一、ご不明点や誤りなどお気づきのことがありましたらご連絡下さい。
- 本書の実行結果から生じるお客様の損害や不利益については、それが直接的、あるいは間接的を問わず一切責任を負いかねますのでご了承下さい。
- 本書は改良のため予告なしに変更する場合があります。
-
- 本書に記載された会社名や製品名は、各社の登録商標である場合がございます。

目次

1. Archi_1 Script 構文一覧表

1. 1. 演算処理記述関連文

種別	名称	構文概要
演算文	演算文	代入先 = 演算式
代入文	代入文	assign 代入先 =
定義文	参照チャネル信号名定義文	def ch_name 参照チャネル　信号名
定義文	参照チャネル単位定義文	def ch_unit 参照チャンル 単位
定義文	引継チャネル定義文	def inherit_ch チャネル列
定義文	ローカルチャネル定義文	def local_ch チャネル列
宣言文	Index 制御除外チャネル宣言文	dcl exempt_ch チャネル列
演算処理ブロック文	演算処理ブロック開始文	proc 演算処理ブロック名 [
演算処理ブロック文	演算ブロック終了文	] 演算処理ブロック名
演算処理ブロック呼出文	演算処理ブロック呼び出し文	call proc 演算処理ブロック名 引数チャネル列
演算処理ブロック制御文	データ番号実行制御文	index 開始 index データ個数
演算処理ブロック制御文	Mark 番号実行制御文	mark 開始マーク番号 終了マーク番号
演算処理ブロック制御文	条件実行制御文	case 比較値 >= 比較値
演算処理ブロック制御文	論理成立実行制御文	case_true 論理値
演算処理ブロック制御文	論理非成立実行制御文	case_false 論理値
演算処理ブロック制御文	繰り返しデータ番号実行制御文	repeat_index 開始データ番号 データ個数　終了データ番号
演算処理ブロック制御文	繰り返し Mark 番号実行制御文	repeat_mark 開始マーク番号 飛び越しマーク数　終了マーク番号
演算処理ブロック制御文	繰り返し条件実行制御文	repeat_case 比較値 = 比較値
演算処理ブロック制御文	繰り返し実行制御文	repeat 制御変数 開始値 増分値　終了値
その他	SUB(dll)実行文	call sub %ファイル番号 dll 関数名 引数列
定義文	サンプリング周期定義文	def sampl_period サンプリング周期 単位
定義文	処理データ個数定義文	def num_samps データ個数

1. 2. ファイル取扱共通文

種別	名称	構文概要
定義文	ファイル番号定義文	def file_id %ファイル番号 ファイル名 属性
定義文	カレントフォルダパス定義文	def folder_path フォルダパス格納先 フラグ
ユーザ I/F 文	ファイル選択文	get file_name ファイル名格納先　ファイル数格納先 拡張子
ユーザ I/F 文	カレントフォルダ選択文	get folder_select
入出力文	カレントフォルダパス取得文	read folder_path フォルダパス格納先 フラグ
入出力文	ファイルステータス取得文	read file_check %ファイル番号 確認コード
入出力文	フォルダ内ファイル情報取得文	read file_info &n &m &k $j wav
その他	ファイル削除文	delete file %ファイル番号

ファイル番号定義文はファイルを読み出す場合、生成格納する場合には記述必須です。

1. 3. 解析対象ファイル情報取得構文

種別	名称	構文概要
入出力文	収録ファイル読み出し文	read wave %ファイル番号 読み出し開始 index,データ個数
入出力文	チャネル数取得文	read num_ch チャネル数格納先
入出力文	チャネル番号取得文	read ch_series チャネル番号格納先
入出力文	収録チャネル信号名単位名取得文	read ch_name 信号名格納先 単位格納先　チャネル番号
入出力文	コメント取得文	read comment コメント番号 コメント格納先
入出力文	マーク個数取得文	read num_mark マーク個数格納先
入出力文	マーク位置取得文	read mark_pos マーク番号 Index 格納先
入出力文	マークメモ取得文	read mark_memo マーク番号 マークメモ格納先
入出力文	ヘッダファイル読み出し文	read header_info 文字列格納先 文字数格納先
ユーザ I/F 文	収録チャネル割り当て文	get replace_ch 割り当てチャネル列

収録ファイル読み出し文は、解析範囲指定して実行する場合は記述不要です。なお、収録ファイル読み出し文のファイル番号はファイル番号定義文の属性 wav で定義したファイル番号にのみ有効です。

解析対象ファイル情報取得構文と同様機能は演算関数にも用意されています。

1. 4. パラメータ選択表示関連文

種別	名称	構文概要
ユーザ I/F 文	諸元表ファイル参照文	get param %ファイル番号 諸元値格納先 単位格納先
ユーザ I/F 文	値入力文	get value 値格納先列 フラグ 表題
ユーザ I/F 文	メッセージ表示応答取得文	get reply 表題 ボタンラベル 1 ボタンラベル 2 応答格納先
ユーザ I/F 文	メッセージ表示文	disp message 表示文
ユーザ I/F 文	チェックボックス・ステータス取得文	get check_box_status 表示文字列 ステータス格納先 表題
ユーザ I/F 文	テキスト表示選択文	get text_page 表示テキスト列,選択ページ番号 表題
ユーザ I/F 文	ラジオボタンステータス取得文	get radio_button_status 選択項目名 選択項目番号 表題
ユーザ I/F 文	通知ダイアログ書き込み文	disp value 書き込みチャネル列 表示フラグ
入出力文	プログレスステータス書き込み文	write progress_status 書き込み文字列

1. 5. 波形ファイル生成関連文

種別	名称	構文概要
定義文	生成波形ファイル・サンプリング値定義文	def wav_sampling %ファイル番号 フラグ 周期 単位
定義文	生成波形ファイル・データタイプ定義文	def wav_type %ファイル番号 データタイプ
定義文	生成波形ファイル・コメント定義文	def wav_comment %ファイル番号 コメント番号 コメント
定義文	生成波形ファイル・マーク情報定義文	def wav_mark %ファイル番号 マーク index マークメモ
定義文	生成波形ファイル・チャネル番号定義文	def wav_ch_series %ファイル番号 チャネル番号
定義文	生成ファイル開始日付時刻定義文	def wav_datetime %ファイル番号 年月日 時刻 オフセット
入出力文	生成波形データファイル格納文	save wave %ファイル番号,格納チャネル列
入出力文	データ位置情報ファイル格納文	save pos_info %ファイル番号 $n $m $k $j \|$q|

演算結果波形ファイルの生成に記述必須構文はファイル番号定義文と生成波形データファイル格納文のみとなります。また、ファイル番号はファイル番号定義文の属性 wav で定義したファイル番号にのみ有効です。 データ位置情報ファイルは PcWaveForm で呼び出した時に、波形上に値を表示させるためのファイルです。

1. 6. 結果シート操作文

種別	名称	構文概要
宣言文	結果シート宣言ブロック開始文	dcl sheet ページ数 シート表題 {
宣言文	結果シート・ページ属性宣言文	page ページ番号: ページ表題
宣言文	結果シート・書き込みチャネル列宣言文	column 書き込みチャネル列
宣言文	結果シート・書き込み列フォーマット宣言文	format フォーマット列 フラグ
宣言文	結果シート宣言ブロック終了文	}sheet
入出力文	結果シート・チャネル列書き込み文	write ch_column ページ番号: 書き込みチャネル列
入出力文	結果シート・メモ列書き込み文	write memo_column ページ番号: メモ
入出力文	結果シート・書き込み行揃え文	write line_feed ページ番号: 改行数
定義文	結果シート表題定義文	def sheet_title ページ番号: 表題
入出力文	結果シート・ファイル格納文	save sheet %ファイル番号
入出力文	結果シート・ファイル読み出し文	read sheet %ファイル番号
その他	結果シートの初期化文	clr sheet ページ番号:

結果シートを格納しない場合は、上記すべての Script 文は使用しません。
既存の結果シートに追記しない場合は、結果シート読み出し文は記述不要です。 結果シートの格納/読み出しはファイル番号定義文の属性 sht で定義したファイル番号にのみ有効です。

1. 7. グラフ描画関連文

種別	名称	構文概要
定義文	グラフ ID 番号定義文	def graph_id @グラフ番号 表題 メモ 形式
定義文	グラフ表示 X 軸定義文	def graph_caxis @グラフ番号 枠情報 目盛情報 単位 軸属性
定義文	グラフ表示 Y 軸定義文	def graph_yaxis @グラフ番号 枠情報 目盛情報 凡例 軸属性
定義文	グラフ表示チャネル番号定義文	def graph_ch_series @グラフ番号 チャネル番号
定義文	グラフ描画線種色定義文	def graph_line @グラフ番号 表示線種 表示色
定義文	グラフ枠内追加線定義文	def graph_line_draw @グラフ番号 グラフ枠内追加線情報 線種 線色
定義文	グラフ描画縦横比定義文	def graph_aspect_ratio %グラフ ID 番号 グラフ縦横比
入出力文	グラフ描画文	polt x-y @グラフ番号 X 軸チャネル列 Y 軸チャネル列
入出力文	グラフ格納文	save plot %ファイル番号 @グラフ番号 X 軸チャネル列 Y 軸チャネル列

1. 8. テキストファイル取扱文

種別	名称	構文概要
定義文	テキストフォーム行列数定義文	def text_form %ファイル番号 行数,列数
入出力文	テキストフォーム読み出し文	read text_form %ファイル番号
入出力文	テキスト書式セル読み出し文	read cell %ファイル番号 行番号,列番号 方向 チャネル列 個数 フラグ
入出力文	テキストファイル行列数取得文	read num_cell %ファイル番号 列数
入出力文	テキストフォーム・セル書き込み文	write cell %ファイル番号 行番号,列番号 方向 チャネル列
入出力文	テキストフォーム格納文	save text_form %ファイル番号
入出力文	フォルダ内ファイル情報取得文	read file_info &n &m &k $j wav

演算結果数値を csv/txt 形式テキストファイルと生成する場合、あるいはテキスト形式収録データの解析、パラメータ取得に使用するテキストファイル操作関連構文です。

1. 9. 文字列取扱文

種別	名称	構文概要
文字列操作	文字属性配列要素数取得文	char num_element 対象文字列 個数格納先
文字列操作	文字列長さ取得文	char length 対象文字列 文字数格納先
文字列操作	文字列切り出し文	char extract 対象文字列 開始位置　文字数 格納先
文字列操作	文字列検索文	char find　対象文字列 検索文字列 $k $q
文字列操作	文字列配列の再構成文	char recomposition 対象文字列 再構成論理 フラグ　格納先
文字列操作	文字列削除文	char delete 対象文字列 開始位置　文字数 格納先
文字列操作	文字列挿入文	char insert 対象文字列 挿入文字列 開始位置　格納先
文字列操作	文字列置換文	char replace 対象文字列 置換対象 開始位置 置換文字列　格納先

文字列取扱 Script 文と同様機能は演算関数にも用意されています。

1. 10. バイナリファイル取扱関連文

種別	名称	構文概要
入出力文	バイナリファイル読み出し文	read bin %n オフセット byte 数 読み出し個数 格納先　読み出し属性
定義文	バイナリバッファ定義文	def bin_buf *バッファ番号 バッファサイズ
入出力文	バイナリファイルチャネル構造読み出し文	read bin_struct %ファイル番号 開始オフセット 読み出し数＜格納先チャネル:形式＞　読み出し後オフセット
入出力文	バイナリバッファ書き込み文	write bin_buf *バッファ番号　書き込み開始バイト位置 書き込み ch 列　書き込み後バイト位置 フラグ
入出力文	バイナリバッファ格納文	save bin_buf %ファイル番号 *バッファ番号列 フラグ

1. 11. その他

種別	名称	構文概要
宣言文	実行メニュー･ボタンラベル宣言文	dcl menu_label ボタンラベル名
宣言文	Script 実行時エラートラップ宣言文	dcl $err
宣言文	Script 実行時環境宣言文	dcl script_env フラグ
その他	実行時エラーフラグ初期化文	clr $err
その他	EXCEL 実行文	call excel エクセルマクロ名
その他	Script 実行終了文	end

2. Archi_1 Script 構文仕様

2. 1. 宣言文

2. 1. 1. 結果シート宣言ブロック開始文（結果シート取扱文）

機能:	結果シートの宣言ブロックの開始および結果シートのページ数を宣言します。
文法:	**dcl sheet** ページ枚数 シート表題 {
引数:	【ページ枚数】<必須><即値> 結果シート全体のページ数を即値で記述します。 【シート表題】<省略可><文字列即値> 結果シート全体 の表題を文字列即値で記述します。
記述例:	dcl sheet 8 “試験演算処理結果表” { dcl sheet 8 {
備考:	本文は Script 全体で 1 行記載でき、使用する結果シートブロックの開始行を意味します。

2. 1. 2. 結果シート・ページ属性宣言文（結果シート取扱文）

機能:	結果シートの使用するページ番号毎のページ属性を宣言します。
文法:	**page** ページ番号: ページ属性 ページ表題
引数:	【ページ番号】<必須><即値:> 即値で 1 から宣言されているページ枚数の範囲で記述し、末尾に半角コロン”:”を付けます。 【ページ属性】<省略可><keyword> キーワードで、cw または fw と記述します。記述省略した場合は cw と見なします。 cwと宣言したページは、後述する column 文および format 文以外は記述できません。 ページ属性 fw は予約キーワードで、現在未対応です。 【ページ表題】<省略可><文字列即値> 半角ダブルコーテーション”“で囲んだ文字列即値で記述します。 ページ表題は結果シートのペー ジ tab に表示されます。
記述例:	page 1: cw “演算結果 1” page 1: “演算結果 1” page 1: /*ページ属性、ページ表題を省略した場合*/
備考:	結果シート宣言文記述行から結果シート宣言終了文記述行の範囲に記述します。 本文は、複数行記述できますが、ページ番号は唯一無二である必要があります。 本文で宣言されていないページは、参照することはできません。

2. 1. 3. 結果シート・書き込みチャネル列宣言文（結果シート取扱文）

機能:	結果シート･ページ属性が cw の時のページごとに書き込むチャネル列を宣言します。
文法:	**column** チャネル列
引数:	【チャネル列】<必須><収録チャネル/数/値属性/文字属性参照チャネル><チャネル連続指定可> 書き込み列を構成するチャネル番号を半角カンマで、区切って記述します。記述可能なチャネルは、収録チャネル、参照チャネルの何れかとなります。（即値あるいは文字列即値は書き込めません。書き込む場合はいったん参照チャネルに代入しその参照チャネル番号を記述します。） ただし、同じチャネルを複数個記述できません。チャネル列記述のチャネル番号連続省略記述が使用できます。
記述例:	column $1,&2,#2,#5,$3,$4,$5 column $1,&2,#2,#5,$[3,3]
備考:	結果シート･ページ column 属性宣言文の直下行に記述し、同一ページに複数行記述できません。 dcl sheet 2 { page 1: column $[1,5]

2.1.4. 結果シート･書き込み列フォーマット宣言文 （結果シート取扱文）

機能:	結果シート･書き込み列宣言文に対応した書き込み形式を指定する場合に宣言します。 書き込み形式を省略値とする場合は、記述する必要はありません。
文法:	**format** フォーマット列　フラグ
引数:	【フォーマット列】<必須><Keyword+即値><繰り返し参照記述可> 半角カンマ","で区切ってフォーマットを記述します。 <フォーマット記述則> ① Exponential 形式の場合 ⇒ Em m は符号、小数点を含まず全桁設定します。 記述例:E3　12.345　→　1.23e1　　0.0000123　→　1.23e-5 ② Significant Format 形式の場合 ⇒ Sm m は符号、小数点を含まず有効桁数全桁を設定します。 記述例:S5　-12.34567→-12.345　123.4567→123.45 0.1234567→0.12345 ③ Fractal Format 形式の場合 ⇒ Fm m は小数点以下の桁数を設定します。 記述 例:　F3　-12.234567→-12.234　123.4567→123.456 0.1234567→0.123 ④ 文字属性チャネルの場合 ⇒ Am m は表示文字数を設定します。文字数省略ができます。 記述例:A3　"ABCdef" → "ABC" <繰り返し参照記述則> 文法:繰り 返し数(フォーマット列) 記述例: 4(A,F0) フォーマット列が設定した回数参照されます。 【フラグ】<省略可> 即値で記述します。フラグは、メモ列および名列の有 無を設定します。 0　Memo 列あり、名列あり 1　Memo 列なし、名列あり 2　Memo 列なし、名列なし 3　Memo 列あり、名列なし なお、省略時は 0 指定と同じです。
記述例:	format F6,S5,A,E7 2 format 2(F6,S5,A12,E7),S4,F4
備考:	結果シート･ページ内の書き込みチャネル列宣言文の直下行に記述します。なお、1 ページに複数行の記述はできません。 dcl sheet 2 { page 1: cw column $[1,5] format F3 2 page 2: column $4,$8,&2 format S5,F0,A 1 フォーマット列が column 文で宣言した列数に満たない場合は、循環して参照されます。例えば、宣言チャネル列数が 10 の場合で、記述したフォーマット列が F6,S5,A12,E7 であった場合、参照されるフォーマットは F6,S5,A12,E7,F6,S5,A12,E7,F6,S5 となります。 設定形式に変換できない、または、溢れてしまう場合、省略変換形式が使用されます。

2.1.5. 結果シート宣言ブロック終了文 （結果シート取扱文）

機能:	結果シート宣言ブロックの終結を意味します。
文法:	**}sheet**
引数:	なし
記述例:	}sheet
備考:	最後の列フォーマット宣言文の直下、または、列フォーマット宣言文が省略された場合は、列宣言文の直下に記述します。 dcl sheet 2 { page 1: column $[1,5] format F3 2 page 2: column $4,$8,&2 format S5,F0,A 1 }sheet

2. 1. 6. Index 制御禁止チャネル列宣言文 (演算処理記述関連文)

機能:	演算処理ブロック制御文で index 制御を受けないチャネルを宣言します。
文法:	**dcl exempt_ch** チャネル列
引数:	【チャネル列】<必須><収録チャネル/数値属性参照チャネル><チャネル連続指定可> 制御を除外するチャネルを半角カンマ","で区切って、記述します。記述可能なチャネルは、収録チャネル、参照チャネルのいずれ も記述できます。ただし、同じチャネルを複数個記述できません。
記述例:	dcl exempt_ch $5,$2,#5,$8 dcl exempt_ch $[1,3],#[2,4]
備考:	演算処理ブロック制御文には、演算対象範囲を index で制御する制御文があり、その制御文の直下に記述される演算処理ブロック内の複数の要素で構成されているチャネルの index は、制御文で設定した範囲に自動的に置き換わります。本文で設定されたチャネルは、制御文の直下に記述される演算処理ブロック内に在っても、再構成されません。また、本文で設定されたチャネルは、Script 中有効で途中解除することはできません。

2. 1. 7. 実行時エラートラップ宣言文 (その他文)

機能:	演算で発生する実行時 Error を検出します。
文法:	**dcl $err**
引数:	なし
記述例:	dcl $err
備考:	構文実行時に Error が発生すると、その時点で Script の実行を放棄しますが、本文にて宣言すると、Script 文の実行時に Error が発生しても Script の実行を放棄せず、$err に 1 を代入して次の行を実行します。Error が発生しない場合は、$err は 0 のままです。なお、演算式で実行時に発生する Error には対応していません。また、$err は、演算式中で扱えません。演算式で参照したい場合は、後述する代入 Script 文(assign 文)で、いったん、参照チャネルに代入してから使用します。

2. 1. 8. 実行時環境宣言文 (その他文)

機能:	実行メニューで設定する環境を宣言します。
文法:	**dcl script_env** 設定フラグ列
引数:	【設定フラグ列】<必須><即値/数値属性参照チャネル> 設定フラグ値を記述します。設定フラグは 4 種あり、設定フラグの合計も 4 個で無くてはなりません。 選定フラグの合計が 5 個以上有る場合は、余ったフラグ は無視されます。 <設定フラグ並び> 0: OFF 1: ON 設定フラグ1番目: Result Sheet Clear: 設定フラグ 2 番目: Result Sheet File Append Write: 設定フ ラグ 3 番目: Result Position_info File Append Write 設定フラ グ 4 番目: Wave File Auto Open
記述例:	dcl script_env 1,1,1,1 ← すべての機能を ONする場合 dcl script_env $1,$2 ← フラグ 1 番目と 2 番目のみ設定その他は初期値を使用する。
備考:	初期値は Script 実行メニューで設定されている内容となります。Script 実行時は本文で定義された内容にしたがい Script 実行終了後は、実行メニューで設定されている内容にしたがいます。

2. 1. 9. 実行メニュー・ボタンラベル宣言文 (その他文)

機能:	記述した Script を実行するメニューボタンのラベルを宣言します。
文法:	**dcl menu_label** メニュー・ボタンラベル名 フラグ
引数:	【メニュー・ボタンラベル名】<必須><文字列即値> Script 実行 Window の実行ボタンラベル名を文字列即値で記述します。 【フラグ】＜省略可＞＜即値＞ 実行終了通知ダイアログを表示する/しないを設定するフラグで、0 の時表示する。1 の時表示しないを設定できます。 なお、記述省略した場合は 0 と見なします。
記述例:	dcl menu_label "解析実行"
備考:	本文で宣言したラベル名が、Script 実行ボタンに表示されます。 Script 中で 1 行記述します。複数行記述された場合は、最新の menu_label 文で宣言された内容が表示されます。また、本文が存在しない場合は、メニュー・ボタンラベルは、当該 Script のファイル名が表示されます。

2. 2. 定義文

2. 2. 1. 結果シート表題定義文（結果シート取扱文）

機能:	結果シート表題を定義します。
文法:	**def sheet_title** ページ番号 表題
引数:	【ページ番号】<必須><即値:> ページ表題を設定するページ番号を記述します、記述は即値に続き半角コロン":"を付加します。 Sheet 全体の表題の場合は、0 と記述します。ただし、宣言されているページ数を越えて記述した場合は、実行時エラーとなります。 【表題】<必須><文字列即値/文字属性参照チャネル><Index 可><間接指定可> 書き込む表題を記述します。参照チャネルで記述した場合で、複数要素を持つ場合、記述したページ番号から要素ごとに参照され 書き込まれます。なお、要素数が宣言されているページ数を越えた場合、残り要素は無視されます。
記述例:	def sheet_title 1: &1
備考:	本文は、Script 中複数回定義された場合、最も後で定義された内容で表示されます。

2. 2. 2. 参照チャネル信号名定義文（演算処理記述関連文）

機能:	Script 中で記述する参照チャネルの信号名(単位)を定義します。
文法:	**def ch_name** 参照チャネル 信号名
引数:	【参照チャネル】<必須><数値属性/文字属性参照チャネル><Index 指定/間接指定可> 信号名を定義する参照チャネルを記述します。 【信号名】<必須><文字列即値/文字属性参照チャネル><index 指定可> 半角ダブルコーテーション""で囲んだ文字列即値または文字属性参照チャネルで記述します。単位も同時に設定する場合は半角コ ロン":"単位名を連結した文字列を記述します。文字属性参照チャネルで記述した場合、Index 指定が使用できます。
記述例:	def ch_name $1 "加速度:m/s^2" def ch_name $($2) &3
備考:	参照チャネル記述に間接指定を使用し、間接指定する数値属性参照チャネルが要素を持つ場合、信号名は文字属性参照チャネルで記述し、その要素数が等しくなくてはなりません。 本文で設定した参照チャネルは、演算式および代入 Script 文での信号名および単位記述を省略できます。なお、本文で設定すると、演算式あるいは、代入 Script 文で信号名および単位を記述しても参照されません。ただし、諸元表ファイル読み込み文で設定した参照チャネルは、代入されるチャネルあるいは項目に付けられている信号名および単位に付け替えられます。また、本文により信号名が設定された参照チャネル番号にすでに信号名が付いている場合、上書き更新されます。

2. 2. 3. 参照チャネル単位定義文（演算処理記述関連文）

機能:	Script 中で記述する参照チャネルの単位を定義します。
文法:	**def ch_unit** 参照チャネル 単位名
引数:	【参照チャネル】<必須><数値属性/文字属性参照チャネル><Index 指定/間接指定可> 単位名を定義する参照チャネルを記述します。 【単位名】<必須><文字列即値/文字属性参照チャネル> 半角ダブルコーテーションで""囲んだ文字列 即値または文字属性参照チャネルで記述します。文字属 性参照チャネルで記述した場合、Index 指 定が使用できます。
記述例:	def ch_unit $1 "m/s^2" def ch_unit $($2) &3
備考:	参照チャネル記述に間接指定を使用し、間接指定する数値属性参照チャネルが要素を持つ場合、信号名は文字属性参照チャネルで記述し、その要素数が等しくなくてはなりません。 本文で設定した参照チャネルは、演算式および代入 Script 文での単位記述を省略できます。なお、本文で設定すると、演算式あるいは、代入 Script 文で信号名および単位を記述しても参照されません。ただし、諸元表ファイル読み込み文で設定した参照チャネルは、代入されるチャネルあるいは項目に付けられている信号名および単位に付け替えられます。また、本文により信号名が設定された参照チャネル番号がすでに単位が付いている場合、上書き更新されます。

2. 2. 4. ファイル番号定義文 (ファイル取扱共通文)

機能: Script で使用する読み出しファイルまたは書き込みファイルのファイル番号を定義します。

文法: **def file_id** %ファイル番号 ファイル名 取扱属性

引数:

【ファイル番号】<必須><%即値>
ファイル番号を先頭文字"%"に続き即値で記述します。記述する範囲は 1～255 となります。

【ファイル名】<必須><文字列即値/文字属性参照チャネル><index 指定可><間接指定可>
対応するファイル名を記述します。なお、拡張子は原則付けないで記述しますが、後述する取扱属性とファイル拡張子の関連以外 のファイルで取り扱う場合のみ、拡張子付きファイル名で記述します。

【取扱属性】<必須><keyword>
定義するファイルの取扱属性を記述します。属性は英小文字 3 文字で対象ファイルを意味付けます。 記述した取扱属性により、読み出し/書き込み/格納 Scritp 文が異なります。

取扱属性	対象ファイル	属性
wav	PcWaveForm フォーマットの波形ファイル .hdr/.dat	テキスト/バイナリ
prm	諸元表ファイル .prm	テキスト
tbl	テーブルファイル .tbl	テキスト
pos	データ位置情報ファイル .drw	テキスト
sub	DLL ファイル.dll	テキスト
csv	CSV 形式ファイル(項目区切り文字半角カンマ) .csv	テキスト,csv
txt	テキスト形式ファイル (項目区切り文字 Tab) .tx	テキスト.txt
sht	結果シート･ファイル .sht	テキスト,csv
chr	テキスト形式(項目区切り文字無視)	テキスト
bin	バイナリファイル 拡張子問わない	バイナリ
grp	bmp ファイル保存	bmp

記述例:
def file_id %12 “ABC 諸元表” prm
def file_id %3 &2 wav
def file_id %4 “result_text.fad” txt
def file_id %5 “キロ程.kdt” csv

備考:
ファイルを扱う Script 文は、本文で宣言したファイル番号を参照してファイル名と関連付けられます。
本文は、複数行存在しても問題ありませんが、同じファイル番号で異なったファイルおよび属性を同時に扱うことはできません。同じファ イル番号で異なったファイル名または取扱属性が存在した場合、最新のファイル番号定義文が有効となります。

2. 2. 5. カレントフォルダパス定義文 (ファイル取扱共通文)

機能: カレントフォルダを定義します。

文法: **def folder_path** フォルダパス名

引数:
【フォルダパス名】<必須><文字列即値/文字属性参照チャネル>
フォルダパス名を記述します。

記述例:
def folder_path “c:¥試験データ¥2010 年 6 月”
sef floder_path &1

備考: 設定されたフォルダが存在していない場合、実行時 Error となります。

2.2.6. 生成波形ファイル・サンプリング値定義文（波形ファイル生成関連文）

機能:	File_id 文で取扱属性を wav で定義したファイルを格納する save wave 文が生成する Header File のサンプリング周波数を定義します。
文法:	**def wav_sampling** %ファイル番号 フラグ サンプリング値 単位 X オフセット値
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【フラグ】<必須><即値/数値属性参照チャネル> フラグを 0 また 1 で記述します。 フラグは、定義したサンプリング値が周波数なのか周期なのかを意味し、周波数の場合⇒0、周期の場合⇒1 となります。0 とした場合、ヘッダファイルには RATE 行が、1 とした場合、ヘッダファイルには INTERVAL 行が生成されます 【サンプリング値】<必須><即値/数値属性参照チャネル><Index 指定可> サンプリング値を記述します。 【単位】<必須><文字列即値/文字属性参照チャネル><Index 指定可> 単位を記述します。単位は、生成したデータ列方向の単位を定義するもので、フラグを周波数としても周期の単位で記述します。 時間サンプルの場合"sec"、距離サンプルの場合"m"を記述しますが、特に制限はありません。 【X オフセット値】<省略可><即値/数値属性参照チャネル> ヘッダファイルに記載する X 軸オフセット値を記述します。 X 軸オフセット値は、生成ファイル開始日付時刻定義文(def wav_datetime 文)でも設定可能です。ヘッダファイルに書き込まれる X 軸オフセット値は save wave 文直前に定義された内容が参照されます。
記述例:	def wav_sampling %2 0 1000 “sec” def wav_sampling %2 1 $2(2) &($1)
備考:	save wave 文にて、本文で記述したファイル番号に波形データを格納する時に save wave 文より前に記述された本文が参照され Header File の RATE 行または、INTERVAL 行を生成します。 なお、本文は、同じファイル番号で複数行の記述はできません。本文で明示的にサンプリングを定義しない場合、格納される波形ファイルのサンプリングは、解析時に設定されているサンプリング周波 数値がそのまま使用され、RATE 行で単位は sec となります。

2. 2. 7. 生成波形ファイル・データタイプ定義文 (波形ファイル生成関連文)

機能:
File_id 文で取扱属性を wav で定義したファイルを格納する save wave 文が生成するデータファイルのデータ形式属性、スロープ値、オフセット値を定義します。

文法:
def wav_type %ファイル番号 データ形式属性 スロープ値 オフセット値

引数:

【ファイル番号】<必須><%即値>
ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。
なお、記述するファイル番号は def file_id 文定義済みである必要があります。

【データ形式属性】<必須><Keyword>
格納するデータのバイナリ形式を定義します。データ形式により下表に示す Keyword を記述します。
なお、記述する Keyword は Keyword1 又は Keyword2 の何れで記述しても問題ありません。

格納データ形式	Keyword1	keyword2
符号付き 2byte 整数形式	integer	int16
4byte 浮動小数点形式	float	float32
8byte 浮動小数点形式	double	float64

【スロープ値】<省略可><即値/数値属性参照チャネル>

スロープ値を省略した場合:

データ形式属性に integer/int16 を指定した場合:
格納対象チャネル毎にデータの最大値を 25000 とした 2byte 整数形式に縮退変換され、save wave 文で指定する格納対象チャネルデータに対し下記の演算から各格納値を求めます。演算により求まったスロープ値、オフセット値をヘッダファイルの SLOPE 行と OFFSET 行に、縮退変換データをデータファイルに格納します。

格納対象チャネルデータ = 縮退変換データ*スロープ値+オフセット値

データ形式属性に float/float32/double/float64 を記述した場合:
格納対象チャネルデータはそのまま格納され、スロープ値 1、オフセット値 0 がヘッダファイルの SLOPE 行と OFFSET 行に格納されます。

スロープ値を記述した場合:

データ形式属性に Integer/int16 を指定した場合:
save wave 文で指定する格納対象チャネルデータと指定したスロープ値およびオフセット値から、下記演算式によって格納データを演算します。格納データをデータファイルに、指定したスロープ値、オフセット値をヘッダファイルの SLOPE 行と OFFSET 行に格納します。指定の際には、演算された格納データが int16 の範囲に収まるよう注意してください。収まらない場合は、警告メッセージを表示し、int16 の最大値/最小値に再設定されます。

格納データ =(格納対象チャネルデータ - オフセット値)/スロープ値

データ形式属性に float/float32/double/float64 を記述した場合:
格納対象チャネルデータはそのまま格納され、指定したスロープ値がヘッダファイルの slope 行に格納されます。

格納対象チャネルが複数チャネルの場合、スロープ値は要素毎に参照され、要素数が格納対象チャネル数に満たない場合は、実行時 Error となり、余る場合は無視されます。

【オフセット値】<省略可><即値/数値属性参照チャネル>
格納対象チャネルのオフセット値として生成するヘッダファイルに書き込まれます。但し、オフセット値を記述する場合、先行するスロープ値の記述省略はできません。スロープ値は記述されオフセット値が記述省略された場合は、オフセット値は 0 と見なします。
格納対象チャネルが複数チャネルの場合、スロープ値は要素毎に参照され、要素数が格納対象チャネル数に満たない場合は、実行時 Error となり、余る場合は無視されます。

記述例:

```
def wav_type %1 float
def wav_type %1 integer
def wav_type %1 double
def wav_type %1 int16 $1 $2
def wav_type %1 float32 $1
```

備考:	データ形式属性に integer/int16 を指定し、スロープ値を指定しなかった場合の例: 格納対象チャネルデータ$1 → 0,20000,40000 def wav_type %1 int16 save wave %1 $1 格納内容 データファイル 格納データ ← 0,12500,25000 ・・・・ MAX25000 に縮退される ヘッダファイル スロープ値 ← 1.6 オフセット値 ← 0 データ形式属性に integer/int16 を指定し、スロープ値を指定した場合の例: 格納対象チャネルデータ$1 → 0,2000,40000 スロープ値 → 1 オフセット値 → 0 def wav_type %1 int16 1 0 save wave %1 $1 格納内容 データファイル 格納データ ← 0,20000,32768 ・・・・ 2baye 範囲の最大値までしか記録できないため 2byte 範囲最大値に変換される ヘッダファイル スロープ値 ← 1 オフセット値 ← 0 本文を複数行記述した場合は save wave 文の直前に定義された内容が参照されます。 本文を記述せずデータ形式属性を定義しなかった場合は、符号付 2byte 整数形式、スロープ値省略で定義したものと見なします。

2. 2. 8. 生成波形ファイル・コメント定義文 (波形ファイル生成関連文)

機能:	File_id 文で取扱属性を wav で定義したファイルを格納する save wave 文が生成する Header File の comment 行を定義します。
文法:	**def wav_comment** %ファイル番号 コメント番号 コメント文
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【コメント番号】<必須><即値/数値属性参照チャネル><Index 指定可> 即値または数値属性参照チャネルで記述します。コメント番号はコメント行を意味し、0～3 の範囲で記述します。 0 を記述した場合、文字属性参照チャネルの 0～2 の要素が参照され、一度に 3 行を定義する意味となります。 【コメント文】<必須><文字列即値/文字属性参照チャネル><Index 指定可> コメント文を記述します。
記述例:	def wav_comment %2 1 &4 def wav_comment %2 $1 &4(3)
備考:	save wave 文にて、本文で記述したファイル番号に波形データを格納する時に save wave 文より前に記述された本文が参照され、Header File の comment のコメント番号行に本文で定義したコメントを格納します。 本文は、複数行記述しても問題ありませんが、同一ファイル番号の同一コメント番号で記述された場合は、最後に記述された行が有効 となります。

2. 2. 9. 生成波形ファイル・マーク情報定義文 (波形ファイル生成関連文)

機能:	File_id 文で取扱属性を wav で定義したファイルを格納する save wave 文が生成する Header File の Mark 行を定義します。
文法:	**def wav_mark** %ファイル番号 マーク位置 マークメモ
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【マーク位置】<必須><即値/数値属性参照チャネル><Index 指定可> マーク位置を記述します。マーク位置は、マークを付けるデータ番号を意味します。数値属性参照チャネルが複数の要素を持つ配 列型の場合、生成される Mark 行は複数行生成されます。 【マークメモ】<省略可><文字列即値/文字属性参照チャネル><Index 指定可> マークメモを記述します。マークメモが省略された場合は、マークメモは書き込まれません。 マーク位置とマークメモがいずれも参照チャネルで記述された場合、同じ Index が参照され、マークメモの要素数が少ない場合は、不 足したマーク位置のマークメモは何も書かれません。
記述例:	def wav_mark %1 12345 "開始" def wav_mark %1 $3 &4
備考:	本文は、複数行記述しても良く、同じマーク位置データ番号が存在しても問題ありません。 ただし同じマーク位置データ番号のマーク情報は、後で定義された内容に更新されます。 save wave 文の前に記述した同じファイル番号のすべての本文が参照され Mark 行を生成します。

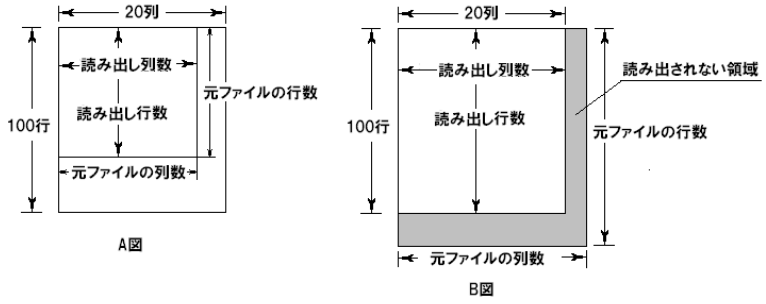
2. 2. 10. 生成ファイル開始日付時刻定義文 (波形ファイル生成関連文)

機能:	File_id 文で取扱属性を wav で定義したファイルを格納する save wave 文が生成する Header File の DATE 行 TIME 行および X_OFFSET 行を定義します。
文法:	**def wav_datetime** %ファイル番号 開始年月日 開始時刻 x_offset 値
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【開始年月日】<必須><文字列即値/文字属性参照チャネル><index 指定可> 収録開始年月日を記述します。記述形式は MM-DD-YYYY となります。 【開始時刻】<必須><文字列即値/文字属性参照チャネル><index 指定可> 収録開始時刻を記述します。記述形式は hh:mm:ss となります。 【x_offset 値】<省略可><即値/文字属性参照チャネル><index 指定可> x_offset 値を記述します。記述省略された場合は、0 が設定されます。
記述例:	def wav_datetime %1 &101 &102 -1.23
備考:	本文が定義されない場合、格納する波形ファイルの開始日付時刻は、ファイル格納時点での日付時刻が設定され、x_offset 値は、0 が設定されます。したがって、収録ファイルの読み出し後、演算処理を行って格納する場合に元の収録日付、時刻および x_offset 値を使用する場合など、DATE 行および TIME 行を自動生成しない場合などは、本文にて明示的に定義する必要があります。

2. 2. 11. 生成波形ファイル・チャネル番号定義文 (波形ファイル生成関連文)

機能:	格納する波形ファイルのチャネル並びを定義します。
文法:	**def wav_ch_series** %ファイル番号 チャネル並び
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【チャネル並び】<必須><即値/数値属性参照チャネル><間接指定可> 定義するチャネル番号並びを記述します。複数のチャネル番号を定義する場合、半角カンマ","で区切ってチャネル並びを記述する か、複数要素を持つ数値属性参照チャネルで記述します。 チャネル並びに、save wave 文で記述した書き込みチャネルが不足した場合は、余りは無視されます。逆に、書き込みチャネル数に定義チャネル番号が不足した場合は、定義したチャネル番号の最大番号から昇順に自動的に振られます。また、記述したチャネル並びに同じチャネル番号が存在した場合は、後出する同じチャネル番号は無視されます。
記述例:	def wav_ch_series %1 $1 def wav_ch_series %1 1,2,3,4,6,7,8,$1
備考:	本文を定義しない場合は、save wave 文で記述された書き込みチャネルの記述順に 1 から昇順に振られます。 なお、本文は、HeaderFile の CH_SERIES 行、および CH 行生成に反映されます。 CH_SERIES 行では、ここで定義された番号の先頭に"CH"が付加されます。

2. 2. 12. テキストフォーム行列数定義文 (テキストファイル取扱文)

機能:	def file_id 文で取扱属性を csv または txt で定義した仮想シートの行列数を定義します。
文法:	**def text_form** %ファイル番号 行列数
	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【行列数】<必須><即値/数値属性参照チャネル><index 指定可> 生成する行数および列数を半角カンマ","で区切って記述します。 複数要素を持つ配列型で記述しても Index0 のみ参照されますので、必ず行と列を記述します。。
記述例:	def text_form %1 300,500 /*行数=300、列数=500*/ def text_form %1 $1($2), $3
備考:	本文は同じファイル番号で複数行記述しても問題ありませんが、書き込みを行う前に最新の定義行が参照されます。ただし、すでに書き込みを行ったテキスト書式を未だファイル格納していない場合、本文を記述すると、書き込みした内容を失います。本文で定義した行列数が書き込み文で記述した開始行列番号を越える場合、書き込み文実行時に Error となります。 本文で定義した後、テキストファイルを読み出した場合、読み出すファイルの書き込み済行数、列数より定義した仮想シートの大きさが 小さい場合、はみ出る列数あるいは行数は仮想シート上に読み込まれません。 例えば、行数 100、列数 20 として定義した後、読み出した場合、下の A 図は定義した仮想シートが読み出すファイルより大きい場合、B 図は小さい場合を示します。  なお、格納されているテキストファイルを読み出した後、本文で行列数を定義すると、先に読み出したテキスト書ファイルが後で定義した行列数を越えていた場合は、本文で定義した範囲を越えた行列番号への書き込みはできません。ただし、書き込まれていた内容を失う事はありません。 このことは、テキストファイルを格納した時に異なり、読み出し後に行列数を定義した場合は、定義範囲を越えたセルへの書き込みはできませんが内容を失うことはありません。先に行列数を定義し、その後、テキストファイルを読み出した場合は、定義範囲を越えたセルの内容を失います。

2. 2. 13. サンプリング周期定義文 (演算処理記述関連文)

機能:	演算関数 PRD()で求める周期値を定義します。
文法:	**def sampl_period** 周期値 単位
引数:	【周期値】<必須><即値/数値属性参照チャネル> 周期値を記述します。 【単位】<省略可><文字列即値/文字属性参照チャネル> 設定した周期の単位を記述します。記述省略した場合、単位は sec と見なします。
記述例:	def sampl_period 1e-3 "sec"
備考:	を読み出した場合は、サンプリング周期はに書き込まれているサンプリング周期またはサンプリング周波数から自動的に参照されますので、定義する必要はありません。を読み出さずに、PRD()関数または関数内でサンプリング周期を参照している関数を使用すると、演算 式実行時 Error が発生します。 本文は、解析対象ヘッダファイルの RATE/INTERVAL 行に相当し、を読み出さない場合に疑似的に同じ役割を果たします。なお、本文で定義した後、を読み出すと、に記載されている値で更新されます。また、本文は Script 中に複数行存在しても良く、本文が実行されるごとに更新定義されます。同様に、を読み出した後に本文を実行すると、定義した値に更新されます。

2. 2. 14. 引継チャネル定義文（演算処理記述関連文）

機能:	call proc 文により呼び出される外部演算処理ブロックの開始文直下に記述し、call proc 文で記述された引数を引き継ぐチャネルを定義します。
文法:	**def inherit_ch** チャネル列
引数:	【チャネル列】<必須><収録チャネル/数値属性/文字属性参照チャネル/ファイル番号><連続指定可> 演算処理ブロックを外部サブルーチンとして使用する場合、引数に対応したチャネルを半角カンマ","で区切り記述します。記述できるチャネルは収録チャネル、数値属性参照チャネル、文字 属性参照チャネル、およびファイル番号のいずれも記述可能です。なお、ファイル番号は先頭文字半角"%"に続きファイル番号を記述します。
記述例:	proc Calc{ def inherit_ch #1,#2,$[1,5],&100,%2
備考:	記述行数は複数行存在しても構いませんが、本文は Scritp 行として連続した行に記述する必要があります。 proc 演算処理ブロック { def inherit_ch チャネル 列 def inherit_ch チャネル列 call proc 文で記述されるチャネル列とチャネル個数および型式が一致している必要があり、一致していない場合、および call proc してチャネル列が設定されている場合で呼び出し先演算処理ブロックに本文が存在していない場合、逆に call poc が存在せず、呼び出し先演算処理ブロックに本文が定義されている場合は、実行時 Errorとなります。 また、演算処理ブロック内で記述するチャネルでローカルチャネル定義を行わないチャネルは、本文で読み替えられたチャネルを含 め、全て Script 全体で共通な(Global.ch)チャネルとして扱われます。 また、サブルーチン内で本文で定義したチャネルに信号名、単位を演算処理ブロック内で定義した場合、引数で受け渡される元の参照 チャネルに信号名、単位を定義したことと等価となります。

2. 2. 15. ローカルチャネル定義文（演算処理記述関連文）

機能:	演算処理ブロックの開始文直下に記述し、演算処理ブロック内に記述されたチャネルの有効範囲を当該演算処理ブロック内に限定したローカルチャネルを定義します。
文法:	**def local_ch** チャネル列
引数:	【チャネル列】<必須><数値属性/文字属性参照チャネル><チャネル連続指定可> サブルーチン内でローカルに使用するチャネルを半角カンマ","で区切り記述します。
記述例:	proc Calc{ def local_ch $[1,5],&100
備考:	記述行数は複数行存在しても構いませんが、本文は連続した行に記述する必要があります。 proc 演算処理ブロック { def local_ch チャネル 列 def local_ch チャネル 列 同じ演算処理ブロックで引継チャネルの定義行が存在した場合、記述順は順不同です。 本文で定義された参照チャネルは定義されている演算処理ブロックおよびその演算処理ブロックにネストされている演算処理ブロック内 でローカルとして扱われ、同じチャネル番号が、当該演算処理ブロックから cal proc 文で呼び出される演算処理ブロックを含めた他の処 理で使用されていても、別のチャネルとして扱

2. 2. 16. グラフ番号定義文（グラフ描画関連文）

機能:	グラフを表示する場合は記述必須文です。本文ではグラフ番号、表題、グラフメモおよびグラフ形式を定義します。 グラフ番号は同じグラフ番号での他の定義文を一括管理します。
文法:	**def graph_id** %グラフ ID 番号 グラフ表題 グラフメモ グラフ属性
引数:	【グラフ ID 番号】<必須><先頭文字@に続き即値> グラフ ID 番号はファイル ID 番号記述と同じ形式で、先頭文字半角"%"に続きチャネル番号で記述します。 【グラフ表題】<省略可><文字列即値/文字属性参照チャネル><index 指定可> 表示するグラフ枠 外上部にグラフ表題として表示されます。 表示行数は一行で複数要素で記述しても先頭 index しか参照されません。 記述省略した場合は、何も表示しません。なお、グラフメモを表示する場合は、省略できません。 【グラフメモ】<省略可><文字列即値/文字属性参照チャネル> グラフ表題行直下行にグラフメモ行(1 行)として表示されます。 複数要素存在しても先頭 index しか参照されません。記述省略された場合は、何も表示しません。 【グラフ属性】<省略可><キーワード> グラフ形式を意味し、x-y と記述します。x-y は X 軸/Y 軸の 2 軸からなる 2 次元グラフを意味します。 記述省略した場合、x-y と見なします。
記述例:	def graph_id @2 "結果グラフ" x-y
備考:	グラフ番号は%に続き数値で表すファイル番号と共通記述します。従がって、ファイル番号ですでに定義さている番号を記述した場合、以降の同じ番号で定義されている Script 文のファイル番号は無効扱いとなります。

2. 2. 17. グラフ表示 X 軸定義文（グラフ描画関連文）

機能: グラフ属性 x-y で定義したグラフの X 軸表示方法を定義します。

文法: def graph_x_axis @グラフ番号 グラフ枠情報 目盛値情報 単位 軸属性

引数:

【グラフ番号】<必須><先頭文字%に続き即値>
グラフ番号の記述則は、先行文字半角"@"に続き番号記述と同じです。 記述するグラフ番号は、
先行して def graph_id 文により定義が必要です。

【グラフ枠情報】
グラフ左端値、グラフ右端値、グリッド線位置の 3 項目から構成され、半角カンマで区切って記述します。

【グラフ左端値】<必須><即値/数値属性参照チャネル><間接指定可>
表示するグラフ枠 X 軸の左端値を記述します。
軸属性 log の場合は、左端値含むディケード開始値に修飾されて参照されます。

【グラフ右端値】<必須><即値/数値属性参照チャネル><間接指定可>
表示するグラフ枠 X 軸の右端値を記述します。
軸属性 log の場合は、右端値を含むディケードの終端値に修飾されて参照されます。グラフ左端値と右端値がいずれも 0 の場合は、AutoScale を意味し、左端値と右端値が等しく 0 以外の場合は実行時 Error となります。
軸属性 log の場合は左端値>右端値の場合は実行時 Error となります。

【グリッド線位置】<省略可><即値./数値属性参照チャネル><間接指定可> 即値あるいは単一要素の参照チャネルで記述した場合はグリッド間隔を意味します。複数要素を持つ数値属性参照チャネルで 記述した場合はグリッド線位置となります。
なお、軸属性が log の場合はグリッド線位置あるいはグリッド線間隔の設定は無効で記述しても無視されます。

※ Autoscale とは、グラフ描画文(draw graph 文)で記述された X 軸チャネル列の内容に従うことを意味します。
X 軸チャネル列の中での最小値をグラフ X 左端値とし、同様に最大値をグラフ X 軸右端値とします。
ただし、グラフ描画文で X 軸チャネル列が記述省略されている場合は、グラフの X 軸は Index 並びとなり X 軸グラフ左端値 0、X 軸右端値はグラフ描画文(draw graph 文)で記述される Y 軸チャネル列の最も要素数の多い最大 Index 値となります。

【目盛値情報】目盛値情報は目盛値、目盛表示開始グリッド線番号、目盛表示グリッド線間隔を半角カンマで区切って記述します。

【目盛値】<省略可><文字属性参照チャネル/.Format キーワード>
目盛値は、例えば F2 の様に Format キーワードで記述した場合、目盛値は自動となります。Format は F 形式、S 形式、E 形式いずれも記述できます。文字属性参照チャネルで記述された場合は表示する目盛値を意味します。記述省略された場合は表示形式を意味し E 形式有効桁 4 桁と見なします。

【目盛表示開始グリッド線番号】<省略可><即値/数値属性参照チャネル><Index 指定可>
単一要素または即値で記述された場合は、表示開始グリッド番号と見なし、複数要素を持つ数列で記述された場合は、表示するグリッド線番号と見なします。グリッド線番号はグラフ左端を 0 として振られます。なお、軸属性 log の場合は、ディケード内グリッド線にはグリッド線番号は振られておらず、ディケードに区切りのみグリッド線番号が振られます。

【目盛表示グリッド線間隔】<省略可><即値/数値属性参照チャネル><Index 指定可> グラフ上に描画する目盛値を表示開始グリッド線番号から何本おきに表示するのかを記述します。
記述省略した場合は 0 と見なします。なお、目盛表示グリッド線番号が複数要素を持つ数列で記述された場合は、意味を持たないため記述されても無視されます。

【単位】<省略可><文字列即値/文字属性参照チャネル><index 指定可>
X 軸の表示単位を記述します。記述省略された場合、X 軸単位は表示しません。 単位を記述する場合は、目盛値情報の何れか 1 項目は必ず記述する必要があります。

【X 軸属性】<省略可><キーワード>
X 軸属性が linear 尺なのか、log 尺なのかを linear または log で記述します。
記述省略された場合は Linear と見なします。

記述例:
def graph_x_axis %2 0.0 F4 “sec” linear
def graph_x_axis %2 1,1000 F1 “Hz” log

備考:
本文が省略された場合、X 軸属性 linear、X 軸値は Index で AutoScale されます。
本文で AutoScale 指定された場合、参照されるグラフ描画文は最初に出会うグラフ描画文によって決定されます。
グラフの X 軸は同じグラフ ID 番号で、一種類しか定義できません。複数行存在した場合、draw 文の直前に記述されている行が有効となります。

2. 2. 18. グラフ表示 Y 軸定義文（グラフ描画関連文）

機能: グラフ属性 x-y で定義したグラフの Y 軸表示方法を定義します。

文法: def graph_y_axis @グラフ番号 グラフ枠情報 目盛情報 凡例情報 軸属性

引数:

【グラフ番号】<必須><先頭文字%に続き即値>
グラフ番号の記述則は、先行文字半角"@"に続き番号を記述します。
記述するグラフ番号は、先行して def graph_id 文により定義が必要です。

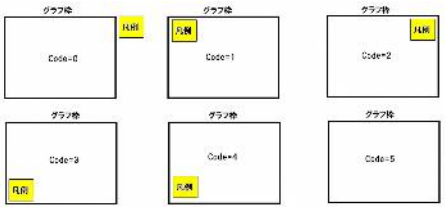
【グラフ枠情報】
グラフ枠下端値,グラフ枠上端値、グリッド線位置の 3 項目から構成され半角カンマで区切って記述します。
【グラフ下端値】<必須><即値/数値属性参照チャネル><間接指定可>
表示するグラフ枠 Y 軸の下端値を記述します。
軸属性 log の場合は、下端値含むディケード開始値に修飾されて参照されます。
【グラフ上端値】<必須><即値/数値属性参照チャネル><間接指定可>
表示するグラフ枠 Y 軸の上端値を記述します。軸属性 log の場合は、上端値を含むディケードの終端値に修飾されて参照されます。グラフ下端値と上端値がいずれも 0 の場合は、AutoScale を意味し、左端値と右端値が等しく 0　以外の場合は実行時 Error となります。軸属性 log の場合は左端値>右端値の場合は実行時 Error となります。
【グリッド線位置】<省略可><即値./数値属性参照チャネル><間接指定可>
即値あるいは単一要素の参照チャネルで記述した場合はグリッド間隔を意味します。複数要素を持つ数値属性参照チャネルで記述した場合はグリッド線位置となります。なお、軸属性が log の場合はグリッド線位置あるいはグリッド線間隔の設定は無効で記述しても無視されます。
※ Autoscale とは、グラフ描画文(draw graph 文)で記述された Y 軸チャネル列の内容に従うことを意味します。
Y 軸チャネル列の中での最小値をグラフ Y 軸下端値とし、同様に最大値をグラフ Y 軸上端値とします。

【目盛値情報】
目盛値情報は目盛値、目盛表示開始グリッド線番号、目盛表示グリッド線間隔を半角カンマで区切って記述します。
【目盛値】<省略可><文字属性参照チャネル/.Format キーワード>
目盛値は、例えば F2 の様に Format キーワードで記述した場合は目盛値は自動となります。Format は F 形式、S 形式、E 形式いずれも記述できます。文字属性参照チャネルで記述された場合は表示する目盛値を意味します。記述省略された場合は表示形式を意味し E 形式有効桁 4 桁と見なします。
【表示グリッド線番号】<省略可><即値/数値属性参照チャネル><Index 指定可>
単一要素または即値で記述された場合は、表示開始グリッド番号と見なし、複数要素を持つ数列で記述された場合は、表示するグリッド線番号と見なします。グリッド線番号はグラフ左端を 0 として振られます。なお、軸属性 log の場合は、ディケード内グリッド線にはグリッド線番号は振られておらず、ディケードに区切りのみグリッド線番号が振られます。
【目盛表示グリッド線間隔】<省略可><即値/数値属性参照チャネル><Index 指定可>
グラフ上に描画する目盛値を表示開始グリッド線番号から何本おきに表示するのかを記述します。
記述省略した場合は 0 と見なします。なお、目盛表示グリッド線番号が複数要素を持つ数列で記述された場合は意味を持ちませんので記述されても無視されます。

【凡例情報】<省略可><即値/数値属性参照チャネル>
凡例とは描画されるチャネル線色、チャネル番号、信号名、単位情報を囲んでグラフダイアログに表示するものです。凡例情報は凡例表示位置コード、表示基準軸コードを半角カンマで区切って記述します。

【凡例位置コード】<省略可><即値/数値属性参照チャネル>
凡例表示位置をコードで記述します。記述省略した場合は 0 と見なします。ただし、基準軸コードを記述する場合は記述省略できません。表示位置コード表を下表に示します。

位置コード	表示位置
0	グラフ枠外―右側
1	グラフ枠内―左上
2	グラフ枠内―右上
3	グラフ枠内―左下
4	グラフ枠内―右下
5	表示しない



【基準軸コード】<省略可><即値/数値属性参照チャネル>
凡例表示するチャネルをグラフ描画文の Y 軸チャネル列に記述されたチャネルを表示するのか X 軸チャネル列に記述されたチャネルを表示するのかの種別フラグで、
Y 軸チャネル列を表示する場合 ⇒ 0　　X 軸チャネル列を表示する場合 ⇒ 1
と記述します。ただし、グラフ描画文で X チャネル列が記述省略されている場合は基準軸コードに 1 は記述できません。また、凡例位置コードが記述省略されている場合は記述した基準コードは凡例位置コードを見なします。

【軸属性】<省略可><Keyword>
linear、log の何れかを記述します。記述省略した場合はlinearと見なします。

記述例:
```
def graph_y_axis @1 0,100
def graph_y_axis @1 0,100,20
def graph y_axis @1 0,100 F2
```

	def graph y_axis @1 0,100,20 F2
備考:	本文が記述省略された場合は、軸属性 linear で Grid 線 AutoScale 目盛値表示形式 E 形式となります。グラフの Y 軸は同じグラフ番号で、一種類しか定義できません。複数行存在した場合、グラフ描画文の直前に記述されている行が有効となります。

2. 2. 19. グラフ描画線種色定義文 (グラフ描画関連文)

機能: グラフ属性 x-y で定義したグラフの描画線種線色を定義します。

文法: def graph_line @グラフ番号 表示線種情報 表示色情報

引数:

【グラフ番号】<必須><先頭文字@に続き即値>
グラフ番号の記述則は、ファイル番号記述と同じです。記述するグラフ番号は、先行して def graph_id 文により定義が必要です。

【表示線種情報】
データ描画線種、グラフ枠線種、グリッド線種の 3 項目で構成され、半角カンマで区切って記述します。

【データ描画線種】<省略可><即値/数値属性参照チャネル><index 指定可>
描画するデータの線種をコードで記述します。グラフ描画文に記述した Y 軸チャネル数に要素数が満たない場合は循環して参照されます。記述省略した場合は実線細(Code=1)となります。ただし、続くグラフ枠線種を記述する場合は省略できません。

【グラフ枠線種】<省略可><即値/数値属性参照チャネル><Index 指定可>
描画するグラフ枠の線種をコードで記述します。記述省略した場合は実線太(Code=2)となります。ただし、続くグリッド線種を記述する場合は省略できません。

【グリッド線種】<省略可><即値/数値属性参照チャネル><Index 指定可>
グラフ枠内に描画するグリッド線の線種をコードで記述します。記述省略した場合は実線極細(Code=0)となります。
記述する線種コード表

Code	10 の桁:線の種類	1 の桁:線の太さ
0	実線	極細
1	点線	細
2	破線	太
3	1 点鎖線	極太
4	Bar	
5	Dot	

※ グラフ枠線種/グリッド線種の 10 桁は 0～3 までとなり Bar および Dot は使用できません。
※ 線の種類 Bar はグラフ描画文の Y 軸チャネル数が 1ch で、X 軸並びが Index 以外は使用できません。

【表示色情報】
データ描画線色、グラフ枠/グリッド線色、目盛値表示色を半角カンマで区切って記述します。

【データ描画線色】<省略可><文字列即値/文字属性参照チャネル><Index 指定可>
グラフに描画するチャネル線色を記述します。グラフ描画文に記述した Y 軸チャネル数に要素数が満たない場合は循環して参照されます。記述省略した場合はグラフ描画文の Y 軸チャネルの記述順に濃紺色、赤色、青色、葡萄色、紫色、紅紫と循環して参照されます。ただし、続くグラフ枠/グリッド線色を記述する場合は記述省略できません。

【グラフ枠/グリッド線色】<省略可><文字列即値/文字属性参照チャネル><Index 指定可>
グラフ枠、グリッド線色を記述します。グラフ枠とグリッド線を別の色で描画することはできません。記述省略した場合は黒色となります。ただし、続く目盛表示文字色を記述する場合は省略できません。

【目盛表示文字色】<省略可><文字列即値/文字属性参照チャネル><index 指定可>
目盛値表示文字色を記述します。記述省略した場合は黒色となります。
線色/文字色は文字列で先頭文字半角"#"に続き 16 進数文字 0～F の 6 桁で記述します。
代表的 16 色コードを示します。

表示色文字列	英名	漢字名
"#800000"	navy	濃紺色
"#0000FF"	red	赤色
"#FF0000"	blue	青色
"#000080"	maroon	葡萄色
"#800080	purple	紫色
"#FF00FF"	fuchsia	紅紫色
"#000000"	black	黒色
"#008000"	green	緑色
"#808000"	teal	緑青色
"#00FF00"	lime	黄緑色
"#008080"	olive	黄薄緑色
"#080808"	gray	灰色
"#C0C0C0"	silver	銀色
"#FFFF00"	aqua	薄青色
"#00FFFF"	ye low	黄色

※ 色指定は上記表以外の指定も可能です。

記述例: def graph_line @3 $1,3,2 &1,"#FF00FF","#000000"

備考:

2. 2. 20. グラフ表示チャネル番号定義文（グラフ描画関連文）

機能:	グラフ属性 x-y で定義したグラフに表示するチャネル番号を定義します。
文法:	**def graph_ch_series** @グラフ番号 チャネル番号
引数:	【グラフ番号】<必須><先頭文字%に続き即値> グラフ番号は先頭文字半角"@"に続き即値で記述します。 記述するグラフ番号は、先 行して def graph_id 文により定義が必要です。 【チャネル番号】<必須><即値/数値属性参照チャネル><index 指定可> 表示するチャネル列に振られるチャネル番号を意味します。 配列の要素が、表示するチャネル数に満たない場合は、以降のチャネ ル番号は、記述したチャネル番号の最も大きな値からインクリメントして振られます。
記述例:	def graph_ch_series @2 $10 def graph_ch_series @2 3
備考:	本文が省略された場合、チャネル番号は表示するチャネル列記述順に 1 から振られます。 本文で定義するチャネル番号は、グラフ表示文で記述するチャネル列のチャネル番号と関係なく、例えば、$1,$3,#2 と 3 チャネル表 示する場合の並び順に割り当てる表示上のチャネル番号を意味します。

2. 2. 21. グラフ枠内追加線定義文（グラフ描画関連文）

機能:	結果グラフ枠内に描画する追加線を定義します。
文法:	**def graph_line_draw** @グラフ番号 グラフ枠描画内追加線情報 線種　線色
引数:	【グラフ番号】<必須><@即値:> グラフ番号は先頭文字半角"@"に続き即値で記述します。 記述するグラフ番号は、先 行して def graph_id 文により定義が必要です。 【グラフ枠内追加線情報】 グラフ枠描画情報は X 軸座標,Y 軸座標、ペンアップダウンフラグの 3 項目から構成され半角カンマで区切って記述します。 【X 軸座標】<必須><数値属性参照チャネル> X 軸位置を記述します。記述範囲は X 軸定義文で設定されたグラフ左端値～右端値の範囲内で在る事が必要です。範囲外が指定された場合、強制的に左端値あるいは右端値となります。また、必ず複数の要素を持つ数列で定義します。 【Y 軸座標】<必須><数値属性参照チャネル> Y 軸位置を記述します。記述範囲は Y 軸定義文で設定されたグラフ下端値～上端値の範囲内で在る事が必要です。範囲外が指定された場合、強制的に下端値あるいは上端値となります。また、必ず複数の要素を持つ数列で定義します。 ※　X 軸位置、Y 軸位置の何れか要素数の少ない軸位置は循環して参照されます。 【ペンアップダウンフラグ】<省略可><数値属性参照チャネル> 現在の X 位置,Y 位置から次の X 位置,Y 位置に移動する時のペンアップダウンをフラグで記述します。 =0 はペンアップ、<>0 はペンダウンを意味します。記述した数列の要素数が X 位置,Y 位置のペアに満たない場合は、循環して参照されます。記述省略された場合は 1 と見なします。 【線種】<省略可><即値/数値属性参照チャネル> 線種コードを記述します。線種コードはグラフ線種線色定義文と同じです。記述省略された場合は、グラフ枠線種にしたがいます。複 数要素を持つ参照チャネルで記述された場合はペンダウンごとに循環して参照されます。 【線色】<省略可><文字列即値/文字属性参照チャネル> 線色を記述します。線色指定はグラフ線種線色定義文と同じです。記述省略された場合は、グラフ枠/グリッド色にしたがいます。複 数要素を持つ参照チャネルで記述された場合はペンダウンごとに循環して参照されます。
記述例:	def graph_line_drawr @1 $1,$2,$3　10 "#000000"
備考:	同一グラフ番号に対して複数行での定義はできません。複数行存在した場合は最新行の内容が参照されます。

2. 2. 22. グラフ描画縦横比定義文(グラフ描画関連文)

機能:	描画するグラフの縦横比を定義します。本文の記述省略された場合は接続されている DISPLAY の全画面表示となり、縦横比は DISPLAY の縦横比が採用されます。
文法:	**def graph_aspect_ratio** %グラフ ID 番号 グラフ縦横比
引数:	【グラフ ID 番号】<必須><先頭文字@に続き即値> グラフ ID 番号はファイル ID 番号記述と同じ形式で、先頭文字半角"%"に続きチャネル番号で記述します。 【グラフ縦横比】＜必須＞＜即値/数値属性参照チャネル＞ 縦横比とは横サイズ/縦サイズを意味します。例えば、2 とした場合は描画範囲の横幅が縦幅の 2 倍を意味します。本文を記述省略した場合で接続されている DISPLAY の縦横比で決定され例えば VGA の場合は 1.33 となります。
記述例:	def graph_aspect_ratio @2 2.4
備考:	本文で定義した縦横比で描画する場合、接続されている DISPLAY の縦横比を越える場合は、横幅が DISPLAY の横幅となり、逆に、小さい場合は横幅が DISPLAY の縦幅幅となります。

2. 2. 23. グラフ枠内文字列書き込み文(グラフ描画関連文)

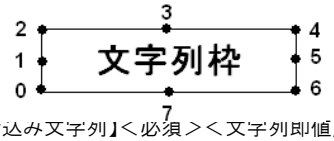
機能: グラフ枠内の指定した位置に文字列を書き込みます。

文法: **def graph_char_draw @グラフ番号 書き込み位置情報 書き込み文字列情報**

引数:

【グラフ番号】<必須><先頭文字%に続き即値>
グラフ番号は先頭文字半角"@"に続き即値で記述します。
記述するグラフ番号は、先行して def graph_id 文により定義が必要です。

【書き込み位置情報】<必須>
文字列を書き込む位置情報を記述します。位置情報は X 座標、Y 座標、書き込み原点コードの 3 種で構成されます。
【X 座標】＜必須＞＜即値/数値属性参照チャネル＞＜Index 指定＞＜間接指定可＞
文字列書き込み X 座標を記述します。
【Y 座標】＜必須＞＜即値/数値属性参照チャネル＞＜Index 指定＞＜間接指定可＞
文字列書き込み Y 座標を記述します。要素数は X 座標と等しい必要があります。
【書き込み原点コード】＜省略可＞＜即値/数値属性チャネル＞＜Index 指定＞＜間接指定可＞
X 座標、Y 座標が書き込み文字列枠のどの位置を指しているかを意味するコードで 0 から 7 で記述します。記述省略した場合は 0 と見なします。



【書き込み文字列】＜必須＞＜文字列即値/文字属性参照チャネル＞＜index 指定可＞＜間接指定可＞
書き込み文字列情報を記述します。文字列情報は書き込み文字列とサイズコードの 2 種から構成されます。
【書き込み文字列】＜必須＞＜即値/文字属性参照チャネル＞＜index 指定可＞＜間接指定可＞
書き込み文字列を記述します。書き込み位置情報の座標数に満たない場合は循環して参照され、余った場合は無視されます。
【文字サイズコード】＜省略可＞＜即値/数値属性参照チャネル＞＜index 指定可＞＜間接指定可＞

コード	太さ(10 の桁)	大きさ(1 の桁)
0	細	小
1	太	中
2		大

書き込み文字列の要素数に満たない場合は、循環して参照されます。
記述省略された場合は 11 と見なします。

記述例: def graph_char_draw @2 $3,$4,$5 &1,$6

備考:

2. 2. 24. 処理データ個数定義文 (演算処理記述関連文)

機能: 収録チャネルを参照しない場合に取り扱うデータ個数を定義してデータ番号実行制御文 (index)、繰り返しデータ番号制御文(repeat_index)を使用可能します。

文法: **def num_samps データ個数**

引数:
【データ個数】<必須><即値/数値属性参照チャネル:><index 指定可><間接指定可>
仮想のデータ個数を定義します。

記述例: def num_samps 10000

備考:
すでに本文記述以前に収録チャネルが記述されている場合は意味を持たず何もしません。
本文記述行以降で解析対象ファイルの収録チャネルを記述した場合は、その収録チャネルのデータ個数に自動更新されます。

2. 2. 25. バイナリバッファ定義文(バイナリファイル取扱関連文)

機能: バイナリ形式で格納する内容を一時的に書き込む内部バッファを定義します。

文法: **def bin_buf *バッファ番号 バッファサイズ**

引数:
【バッファ番号】＜必須＞＜*即値＞
バッファ番号を記述します。記述則は先頭文字半角"*"に続いて即値で記述します。

【バッファサイズ】＜必須＞＜即値/数値属性参照チャネル＞
正数(0<バッファサイズ)の場合:
記述したバッファサイズをバイト単位と見なしバッファを確保し、全て null(00h)に初期化します。
0 と記述した場合:
既に同じバッファ番号で定義されているバイナリバッファを削除(解放)します。 再び使用する場合は再定義が必要となります。なお、0 と記述した場合で、該当するバッファ番号が存在していない場合は無視されます。一旦、削除したバッファ番号に書き込むことはできません。使用する場合は再定義する必要があります
-1 と記述した場合:
既に同じバッファ番号で定義されているバイナリバッファの現在書き込まれている最終アドレス以降の null 部分を削除し、バッファサイズを再構成し、再構成されたバッファサイズを戻します。従って、-1 を指定する場合は、必ず数値属性参照チャネルで記述する必要があります。

記述例: def bin_buf *1 4096 /* 4096byte のバッファをバッファ番号1としてアロケーションする*/

備考: 既に定義済みのバッファ番号を再定義した場合、直前に定義されているバッファ内容を失います。

	一旦、定義した内容は、同じバッファ番号でサイズ 0 と再定義されるまで保持されます。

バイナリバッファが大きい場合、暗黙的に外部記憶装置上に採られる場合があります。

2. 3. 入出力文

2. 3. 1. 結果シート・チャネル列書き込み文（結果シート取扱文）

機能:	結果シートのチャネル列に書き込みます。
文法:	**write ch_column** ページ番号: チャネル列
引数:	【ページ番号】<必須><即値:> ページ番号を記述しします。記述則は末尾に半角コロン":"を付加します。ページ番号は、結果シートの書き込みページを意味し、page 文で宣言されている必要があります。 【チャネル列】<必須><収録チャネル/数値属性/文字属性参照チャネル><チャネル連続指定可> 書き込みをするチャネルを半角カンマ","で区切り記述します。 記述するチャネルは、column 文で宣言したチャネルに存在していなくてはなりません。なお、チャネル記述順は、シート上の列とは無関係で、シート上に書き込まれる列は、column 文で記述された順となります。
記述例:	write ch_column 1: $1,$2,$5 write ch_column 3: $[$2,$4],&[1,$5],#[1,2]
備考:	書き込むチャネルが単一要素からなる場合は、1 行、複数の要素からなる場合は、個数分の行に書き込まれます。また、同時に書き込むチャネルのデータ個数が揃っている必要はありません。

2. 3. 2. 結果シート・メモ列書き込み文（結果シート取扱文）

機能:	結果シートのメモ列にメモを書き込みます。ただし、format 文のフラグに、1 または、2 が宣言されている場合は、本文は無視され実行されません。
文法:	**write memo_column** ページ番号: メモ
引数:	【ページ番号】<必須><即値:> ページ番号を記述します。記述則は末尾に半角コロン":"を付加します。ページ番号は、結果シートのメモを書き込むページを意味 し、page 文で宣言されている必要があります。 【メモ】<必須><文字列即値/文字属性参照チャネル><Index 指定可> メモ行に書き込む文字列をします。
記述例:	write memo_column 1: “memo” write memo_column 1: &3($1)
備考:	メモが書き込まれる行は、現在までに各列に書き込まれた最も多い列の次の行のメモ列に書き込まれます。メモが書き込まれた行は改行され、次に write ch_column 文で書き込まれる行を揃えます。

2. 3. 3. 結果シート・書き込み行揃え文（結果シート取扱文）

機能:	結果シートの書き込み行を揃え、設定された改行数分改行します。
文法:	**write line_feed** ページ番号:　改行数
引数:	【ページ番号】<必須><即値:> ページ番号を記述します。記述則は末尾に半角コロン":"を付加します。ページ番号は、結果シートの行揃えを行うページを意味し、page 文で宣言されている必要があります。 【改行数】<必須><即値/数値属性参照チャネル> 改行行数を記述します。
記述例:	write line_feed 1: 0 write line_feed 2: $3
備考:	改行数 0 は、次に write ch_column 文を実行した場合に行が揃います。また、改行数が 1 以上の場合、設定された改行数分、空欄行を生成します。

2. 3. 4. 結果シート・ファイル読み出し文（結果シート取扱文）

機能:	File_id 文で取扱属性を sht で定義した格納されている結果シートを読み出します。
文法:	**read sheet** %ファイル番号
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。
記述例:	read sheet %4
備考:	すでに保存してある結果シートに追記したい場合に使用します。なお、読み出す結果シート・ファイルのページ数が結果シート宣言されているページ番号より多い場合は、余ったページは読み出されません。また、ページ内列数が、宣言されているページ内列数と一致しな いページも読み出されません。

2. 3. 5. 結果シート･ファイル格納文 （結果シート取扱文）

機能:	書き込んだ結果シート全体をファイルに格納します。
文法:	**save sheet** %ファイル番号
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。
記述例:	save sheet %6
備考:	ファイル番号で設定されたファイルがすでに存在している場合で Append(追記)設定されていない場合は上書きされます。存在していない場合は、新たにファイルが生成されます。また、Append 記録が設定されている場合で記録済みのファイルとページごとに書き込みチャネル列数がチェックされ、一致していないページは、記録済みファイルの当該ページは追記されず新たに書き込むチャネル列で更新されます。

2. 3. 6. フォルダ内ファイル情報取得文 （ファイル取扱共通文）

機能:	カレントフォルダに格納されているファイル情報を取得します。
文法:	**read file_info** ファイル名 日付 時刻 個数 拡張子
引数:	【ファイル名】<必須><文字属性参照チャネル> 取得したファイル名格納先を記述します。カレントフォルダに格納されている指定された属性のファイル名を取得して格納します。なお、引数で拡張子が記述されている場合は、格納されるファイル名に拡張子は付きません。 【日付】<必須><文字属性参照チャネル> 取得した日付格納先を」記述します。格納形式はファイルの作成年月日を YYYY/MM/DD の文字列で格納します。 【時刻】<必須><文字属性参照チャネル> 取得した時刻格納先を記述します。格納形式はファイルの生成時 刻を hh:mm:ss の文字列で格納します。 【個数】<必須><数値属性参照チャネル> 取 得したファイル個数格納先を記述します。 【拡張子】<省略可><keyword> ファイル情報を取得するファイルの拡張子を記述します。記述則はダブルコーテテーション"で囲んで拡張子を記述するか、ダブルコ ーテーションで囲まず、先頭半角ピリオド"."を含めた拡張子を記述します。なお、記述省略された場合は、フォルダ内に含むすべて のファイルとなり、格納されるファイル名は拡張子を付いたファイル名となります。
記述例:	read file_info &10 &20 &30 $40 .hdr
備考:	該当するファイルが複数存在した場合、ファイル名、日付、時刻の文字属性参照チャネルは、存在したファイル数分の要素からなる配列型となります。該当するファイルが存在しない場合は、ファイル名、日付、時刻は全て不定で、個数は 0 が戻ります。

2. 3. 7. カレントフォルダパス取得文 （ファイル取扱共通文）

機能:	現在設定されているカレントフォルダパス名を取得します。 ※ 演算関数：カレントフォルダパス名取得関数 CFPN(k)と同じ機能です。
文法:	**read folder_path** フォルダパス名 フラグ
引数:	【フォルダパス名】<必須><文字属性参照チャネル>または<数値属性参照チャネル> 取得したフォルダパス名の格納先を記述します。なお、格納先チャネルを文字属性参照チャネルで記述し場合は、取得したカレントフォルダパス名、またはフォルダ内のフォルダ名を格納し、数値属性参照チャネルで記述した場合は、カレントフォルダ内のフォルダ個数を格納します。ただし後述するフラグを記述省略するか、0 を記述した場合は必ず1となります。 【フラグ】<省略可><即値/数値属性参照チャネル> カレントフォルダ名を取得するか、カレントフォルダ内 のフォルダ名を取得するかのフラグを意味します。 フラグ=0 の時 カレントフォルダ名が戻ります。フラグが記述省略された場合は、フラグ=0 と同じ意味を持ちます。 フラグ=1 の時 カレントフォルダ内のフォルダ名、または個数が戻ります。
記述例:	read folder_path &100 0
備考:	フラグに 1 を記述し、フォルダパス名を文字属性参照チャネルで記述した場合で、カレントフォルダ内にフォルダが存在しない場合は、null で戻ります。戻りフォルダ個数のチェックは、文字属性参照チャネルをフォルダパス名格納先に記述し、実行後文字属性配列要素数取得文でも確認できます。

2. 3. 8. ファイルステータス取得文 （ファイル取扱共通文）

機能:	ファイル番号定義文で定義したファイルが OPEN 可能か否かのステータスを取得します。 ※ 演算関数:ファイルステータス取得関数 RFC(&m)と同じ機能です。
文法:	**read file_check** %ファイル番号 ステータス
	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【ステータス】<必須><数値属性参照チャネル> 取 得したファイルステータス格納先を記述します。 ファイルが Open 可能か否かのステータス格納先を数値属性参照チャネルで記述します。 ステー タス ⇒ 0 存在していて、Open 可能 ステータス ⇒ 1 存在しているが、現在使用中 Open 不可 ステータス ⇒ 2 存在していない
記述例:	read file_check %1 $1
備考:	

2. 3. 9. 収録ファイル読み出し文 （解析対象ファイル情報取得文）

機能:	File_id 文で取扱属性を wav で定義した収録ファイルを読み出します。
文法:	**read wave** %ファイル番号 開始 Index,データ個数
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【開始 Index,データ個数】<省略可><即値/数値属性参照チャネル><Index 指定/間接指定可> 読み出す開始データ番号、およびデータ個数を記述します。データ個数が 0 と記述された場合は、収録ファイルの header 部のみ読 み出します。また、省略された場合は、当該ファイルのすべてのデータが読 み出し対象となります。
記述例:	read wave %3 read wave %3 0,0
備考:	本文が実行されると、収録データに依存する内容が更新されます。なお、Script の実行を波形表示 Window で解析範囲を指定して行った場合は、収録ファイル読み出し文は必要ありません。

2. 3. 10. 解析対象ファイル名取得文 （解析対象ファイル情報取得文）

機能:	現在解析対象となっている収録ファイル名を取得します。 ※ 演算関数:名取得関数 CFNM()と同じ機能です。
文法:	**read file_name** ファイル名格納先チャネル
引数:	【ファイル名格納先チャネル】<必須><文字属性参照チャネル><index 指定可> 取得したファイル名格納先を記述します。
記述例:	read file_name &12 read file_name &12(0) read file_name &12($2)
備考:	が存在していない場合は、実行時 Error となります。※

2. 3. 11. チャネル数取得文 （解析対象ファイル情報取得文）

機能:	入力チャネルテーブルに書き込まれた収録チャネル数を取得します。 ※ 演算関数:収録チャネル数取得関数 NCH()と同じ機能です。
文法:	**read num_ch** チャネル数格納先チャネル
引数:	【チャネル数格納先チャネル】<必須><数値属性参照チャネル> 取得したチャネル数格納先を記述します。
記述例:	read num_ch $12
備考:	

2. 3. 12. チャネル番号取得文 （解析対象ファイル情報取得文）

機能:	解析対象ファイル(収録ファイル)の収録チャネル番号を取得します。 ※ 演算関数:収録チャネル番号取得関数 CHS()と同じ機能です。
文法:	**read ch_series** チャネル番号格納先チャネル
引数:	【チャネル番号格納先チャネル】<必須><数値属性参照チャネル> 取得したチャネル番号格納先を記述します。
記述例:	read ch_series $100
備考:	上記記述例で収録チャネルが#1,#3,#6,#7,#8 の 5ch 存在した場合、$100 の各要素には $100(0) ←1、$100(1) ←3、$100(2) ←6、$100(3) ←7、$100(4) ←8、と取得されます。※

2. 3. 13. 信号名単位名取得文 （解析対象ファイル情報取得文）

機能:	解析対象ファイル（収録ファイル）の信号名および単位名を取得します。 ※ 信号名の取得は 演算関数：収録チャネル信号名取得関数 CHNM()と同じ機能です。 ※ 単位名の取得は 演算関数：収録チャネル単位名取得関数 CUNT()と同じ機能です。
文法:	**read ch_name** 信号名格納先チャネル 単位名格納先チャネル 収録チャネル
引数:	【信号名格納先チャネル】<必須><文字属性参照チャネル> 文字属性参照チャネルで記述します。 【単位名格納先チャネル】<必須><文字属性参照チャネル> 文字属性参照チャネルで記述します。 【チャネル】<省略可><収録/参照チャネル><index 指定可><間接指定可> 信号名を読み出すチャネルを記述します。チャネルは先頭"#"文字の収録チャネル、先頭"$"文字の数値属性参照チャネルで記 述します。なお、チャネルは記述省略可能で、省略された場合はすべての収録チャネルが対象となり、収録チャネル昇順に信号名 格納参照チャネル、単位格納参照チャネルに格納されます。
記述例:	read ch_name &10 &11 read ch_name &10 &11 #2 /*収録チャネル 2ch の信号名、単位を呼び出す */ read ch_name &10 &11 $($1)
備考:	信号名が付けられていないチャネルの信号名は先頭文字"#"に続きチャネル番号として格納し、単位が付けられていないチャネルの単位は半角スペース 1 文字として格納します。

2. 3. 14. コメント取得文 （解析対象ファイル情報取得文）

機能:	解析対象ファイル（収録ファイル）からコメント行を取得します。 ※ 演算関数：解析対象ファルコメント取得関数と同じ機能です。
文法:	**read comment** コメント番号 コメント格納先チャネル
引数:	【コメント番号】<必須><即値/数値属性参照チャネル> 取得するコメント行番号を記述します。コメント行番号は、のヘッダに記載されているコメントの行番号を意味します。 【コメント格納先チャネル】<必須><文字属性参照チャネル> 取得したコメントの格納先を記述します。コメント番号に 0 を記述した場合、コメント行の全てを意味し、記述した文字属性参照チャネルの要素に行ごとに格納されます。
記述例:	read comment 1 &2
備考:	標準の収録ヘッダファイルは 1～3 行のコメント行が存在します。ヘッダファイルに指定したコメント行が存在しても何も書かれていない場合は、半角スペース 1 文字が戻ります。また、指定したコメント行が存在していない場合は、実行時 Error となります。

2. 3. 15. マーク個数取得文 （解析対象ファイル情報取得文）

機能:	現在解析対象に指定されている範囲に付けられているマーク個数を取得します。 ※ 演算関数：マーク個数取得関数 NMK()と同じ機能です。
文法:	**read num_mark** マーク個数格納先チャネル
引数:	【マーク個数格納先チャネル】<必須><数値属性参照チャネル> 取得したマーク個数格納先を記述します。
記述例:	read num_mark $10
備考:	読み出された結果が 0 の場合は、現在の解析範囲に Mark が含まれていないことを意味し、例えば、4 と代入された場合は、現在の解析範囲に Mark 番号1～4 の 4 個の Mark が存在していることを意味します。

2. 3. 16. マーク位置取得文 （解析対象ファイル情報取得文）

機能:	指定した Mark 番号位置のデータ番号を取得します。 ※ 演算関数：マーク番号 Index 取得関数 MRX(k)と同じ機能です。
文法:	**read mark_pos** Mark 番号 マーク位置格納先チャネル
引数:	【Mark 番号】<必須><即値/数値属性参照チャネル> Mark 番号は、Mark 番号位置のデータ番号を読み出したい Mark 番号を記述します。Mark 番号 0 は特別な意味を持ち、解析範囲に存在するすべての Mark 番号を意味し、Mark 位置格納先チャネルの Index0 から存在する Mark 個数分格納されます。 【マーク位置格納先チャネル】<必須><数値属性参照チャネル> 取得した Mark 位置(index)の格納先を記述します。マーク位置は解析範囲の先頭を 0 とした Index となります。
記述例:	read mark_pos 3 $21
備考:	現在設定されている解析範囲に存在しない Mark 番号を設定すると実行時 Error となります。※

2. 3. 17. マークメモ取得文 （解析対象ファイル情報取得文）

機能:	指定した Mark 番号に付けられているメモを取得します。 ※ 演算関数:マークメモ取得関数 CMRM(k)と同じ機能です。
文法:	**read mark_memo** Mark 番号 マークメモ格納先チャネル
引数:	【Mark 番号】<必須><即値/数値属性参照チャネル> Mark メモを取得する Mark 番号を記述します。Mark 番号 0 は特別な意味を持ち、解析範囲に存在するすべての Mark 番号を意味し、マーク位置格納先チャネルの Index0 から存在するマーク個数分格納されます。 【マークメモ格納先チャネル】<必須><数値属性/文字属性参照チャネル> 取得した Mark メモ格納先を記述します。数値属性参照チャネルで記述した場合、付けられている Mark メモに含まれる数字文字列を数値変換して格納します。マークメモに数字文字列を含まない場合は、数値変換できないため、数値 0 として格納されます。
記述例:	read mark_memo 0 $20 read mark_memo 3 &22
備考:	マークメモが存在していない Mark 番号を設定した場合、格納先参照チャネルの属性が文字属性の場合は半角スペース 1 文字、数値属性の場合は-1 が代入されます。 現在の解析範囲に存在しない Mark 番号を設定した場合、実行時 Error となります。

2. 3. 18. ヘッダファイル読み出し文 （解析対象ファイル情報取得文）

機能:	解析対象ファイル(収録ファイル)のヘッダ情報を取得します。
文法:	**read header_info** 格納先 取得結果行数 先頭文字列
引数:	【格納先】<必須><文字属性参照チャネル> 取得したヘッダ情報の格納先を記述します。 【取得結果行数】<必須><数値属性参照チャネル> 取得した行数の格納先を記述します。 【先頭文字列】<省略可><文字列即値/文字属性参照チャネル><index 指定可> ここで記述された文字列が行の先頭に存在している行のみ取得します。なお、取得した行の文字列は、ここで指定された文字列を先 頭から削除されます。
記述例:	read header_info &100 $1 read header_info &100 $1 “X_OFFSET “
備考:	のヘッダ情報が格納先で記述した文字属性参照チャネルに格納されます。 要素数は、ヘッダファイルの行数に一致し、ヘッダファイルの記載 1 行が1要素に対応します。

2. 3. 19. 生成波形データファイル格納文（波形ファイル生成関連文）

機能: File_id 文で取扱属性を wav で定義したファイルを、PcWaveForm 収録ファイルと同じ形式で、チャネル列で記述したチャネルの内容を格納します。

文法: **save wave** %ファイル番号 チャネル列

引数:
【ファイル番号】<必須><%即値>
ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。

【チャネル列】<必須><収録チャネル/数値属性/文字属性参照チャネル><チャネル連続指定可> ファイルに書き込み参照チャネル/収録チャネルを半角カンマ","で区切り記述します。 生成されるファイルのチャネル番号は、生成波形ファイル・チャネルう番号定義文にしたがって振られ、当該文が記述されていない 場合は、ここで記述されたチャネル順に 1 から振られます。

記述例:
save wave %8 $1,$3,#5,$[8,3]
save wave %8 $[$1,$3],#[1,3]

備考: 生成されるファイルは、ヘッダファイル".hdr"とデータファイル".dat"のペアで生成されます。すでにファイル番号で設定したファイルが存在している場合は、上書きされます。データファイルに書き込まれるデータ数は、チャネル列で記述されたチャネルの内、最もデータ数（要素数）の少ないチャネルに整合されます。また、生成されるファイルのチャネル順は、記述されたチャネル列にしたがい、チャネル番号は ch1 から新たに割り付けられます。したがって、チャンネル列で記述したチャネル番号との関連性はありません。 生成されるヘッダファイルの各行は以下の通りです。

行の意味	キーワード	初期値:設定可否
格納チャネル番号	SERIES	CH_1 から連続でチャネル数分自動生成 または def wav_ch_seris 文で定義可能
収録開始年月日	DATE	ファイル格納時年月日:def wav_datetime で定義可能
収録開始時刻	TIME	ファイル格納時時刻:def wav_datetime で定義可能
サンプリング周波数	RATE	現在のサンプリング:def wav_sampling で定義可能
チャネル物理量単位	VERT_UNITS	自動生成
サンプリング軸単位	HORZ_UNITS	Sec :def wav_sampl ing で定義可能
格納チャネル数	NUM_SERIES	自動生成
記録形式	STORAGE_MODE	INTERLACED （固定）
データ形式	FILE_TYPE	INTEGER:def wav_type で定義可能
Y 軸換算傾斜値	SLOPE	自動生成
Y 軸 OFFFSET 値	Y_OFFSET	自動生成
X 軸 OFFSET 値	X_OFFSET	0:def wav_sampling、def wav_datetime で定義可能
データ個数	NUM_SAMPS	自動生成
生成したデバイス	DEVICE	自動生成(Auto_Calc と記載される)
データファイル名	FILENAME	自動生成 (def file_id 文で定義した内容)
コメント1	COMMENT1	空欄:def wav_comment で定義可能
コメント 2	COMMENT2	空欄:def wav_oomment で定義可能
コメント 3	COMMENT3	空欄:def wav_oomment で定義可能
チャネル情報行	CHxx	NAME=信号名　チャネル数分、信号名自動生成
マーク行	MARK	初期値記載なし:def wav_mark で定義可能
終端行	END	自動生成

2. 3. 20. データ位置情報ファイル格納文（波形ファイル生成関連文）

機能:	File_id 文で取扱属性を pos で定義した PcWaveForm の Value_Draw Window の表示波形上に書き込む情報ファイルを生成します
文法:	**save pos_info** %ファイル番号 開始 Index 終了 Index 表示チャネル番号 表示属性 表示値
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【開始 Index】<必須><即値/数値属性参照チャネル><Index 指定可> 開始 index を記述します。開始 Index は波形グラフ上に描画する各値の範囲を指定する開始データ番号です。 【終了 Index】<必須><即値/数値属性参照チャネル><Index 指定可> 終了 index を記述します。終了 Index は波形グラフ上に描画する各値の範囲を指定する終了データ番号です。 【表示チャネル】<必須><即値/数値属性参照チャネル><Index 指定可> 値を表示するチャネル番号を記述します。波形グラフの描画対象チャネル番号を意味します。なお、チャネル番号は数値でそのま ま記述します。 【表示属性】<必須><即値/数値属性参照チャネル><Index 指定可> 同じ Index 指定範囲に表示する属性を即値または数値属性参照チャネルで記述します。 なお、同じ Index 範囲に複数の表示属性を指定する場合は、数値属性参照チャネルで記述し、その要素に属性を格納します。ただし、表示属性 0 は、単独でなければなりません。 1:= 開始 Index 終了 Index で設定された区間の最大値を書き込む 2:= 開始 Index 終了 Index で設定された区間の最小値を書き込む 3:= 開始 Index 終了 Index で設定された区間の最大値－最小値を書き込む 4:= 開始 Index 終了 Index で設定された区間の平均値を書き込む 5:= 開始 Index 終了 Index で設定された区間の時間を書き込む 6:= 開始 Index での値を書き込む 7:= 終了 Index での値を書き込む 8:= 開始 Index での値－終了 Index での値を書き込む 9:= 開始 Index での値と終了 Index での値の傾斜を書き込む 0:= 開始 Index 地点に設定した表示値を書き込む 表示属性 0 記述以外は、ここで生成されたファイルを使用して波形グラフに描画する時に、演算され、書き込む値が決定されます。 【表示値】<表示属性 0 の時必須><即値/数値属性参照チャネル><Index 指定可> 表示 属性に 0 を設定した時に必要な引数で、表示する値を記述します。
記述例:	save pos_info %7 $1 $2 $3 1
備考:	設定する表示対象チャネル番号は、当該ファイルを使用して波形上に書き込む収録ファイルに存在している必要があります。

2. 3. 21. テキストフォーム行列数取得文（テキストファイル取扱文）

機能:	def file_id 文で取扱属性が csv または txt で定義されたテキストファイルの行毎の列数を取得します。
文法:	**read num_cell** %ファイル番号 列数格納先
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【列数格納先】<必須><数値属性参照チャネル><間接指定可> 取得した列数の格納先を記述します。列数の格納先参照チャネルの要素数は、テキスト書式の行数に一致します。
記述例:	read num_cell %1 $1 $2
備考:	同じファイル番号でテキストファイル行列数定義文(def text_form 文)を記述する場合、テキストフォーム定義文より以前に記述される必要があります。テキストフォーム定義文以降に記述すると定義されたテキストフォームの大きさが戻る事になり意味を持ちません。 戻り列数格納先参照チャネルは、要素数が読み出した行数を意味し、各要素には当該行の列数が戻りますので、行数を求める場合は LEN(列数格納先)、最大列数を求める場合は、MAX(列数格納先)で行います。

2. 3. 22. テキストファイル読み出し文 (テキストファイル取扱文)

機能:	def file_id 文で取扱属性 csv または txt で定義されたテキストファイルをテキストフォームに読み出します。
文法:	**read text_form** %ファイル番号
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。
記述例:	read text_form %1
備考:	読み出されたテキストファイルは、テキストフォーム上の開始行列番号 1,1(左上)から書き込まれます。 なお、読み出す前に同じファイル番号で作成しているてテキストフォームが存在している場合は、作成中のテキスト書式の内容を失います。

2. 3. 23. テキストファイル・セル読み出し文 (テキストファイル取扱文)

機能:	テキストファイルから読み出し開始行列番号を指定して読み出し、参照チャネルに格納します。
文法:	**read cell** %ファイル番号 読み出し開始行列番号 方向フラグ 格納先チャネル列 セル数　読み出し先フラグ
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。属性はtxt、csv、chrの 3 種となります。 【読み出し開始行列番号】<必須><即値/数値属性参照チャネル> 読み出し開始 セルの行番号と列番号を半角カンマ","で区切って、記述します。 読み出し方向 0 の時は、行列番号で記述した行番号は読み出し行を意味し、列番号は読み出し開始列を意味します。 また、読み出し方向 1 の時は行列番号で記述した行番号は読み出し開始行を意味し、列番号は読み出し列を意味します。また、列は項目区切り文字で認識されます。読み出し先ファイルの取扱属性が csv の場合は半角カンマ","、txt の場合は tab 文字が列区切りとなり、chr の場合は区切り文字はありませんので列番号およびフラグは、必ず1で無ければなりません。1以外が記述されると実行時 Errorとなります。 【方向フラグ】<必須><即値/数値属性参照チャネル> 読み出し方向を記述します。　列方向(横方向)=0、行方向(縦方向)=1となります。 格納先チャネルが複数チャネル記述され、フラグ=0 の場合は開始行列位置から列方向に読み出し、次のチャネルは、開始行列位 置の行番号をインクリメントして、同じく列方向に読み出します。フラグ=1 の場合は、開始行列番号から行方向に読み出し、次のチャ ネルは開始行列位置の列番号をインクリメントして同じく行方向に読み出します。 【格納先チャネル列】<必須><数値属性/文字属性参照チャネル><間接指定可><連続指定可> 読み出し結果格納先チャネルを文字属性または数値属性参照チャネルで記述します。 複数 チャネルを格納する場合は半角カンマ","で区切って記述します。 なお、Cell を 1 個だけ読み出す場合は、$1(0)の様に指定チャネルに Index を付けて記述します。 【セル数】<省略可><即値/数値属性参照チャネル> チャネル当たりの読み出しセル数を記述します。0 と記述した場合は、当該列または行方向に存在しているセル数分読み出します。 記述省略した場合は 0 と見なします。ただし、読み出し先フラグを記述する場合は省略できません。 【読み出し先フラグ】<省略可><即値/数値属性参照チャネル> 読み出し先がファイル番号で定義されたファイルの場合は 0、ファイル番号で定義されたテキストフォームの場合は 1 と記述します。 記述省略された場合は 0 と見なします。 なお、1 として場合で、テキストフォームが定義されていない場合は実行時 ERROR となります。
記述例:	read cell %1 3,2 0 $1,$2,$3
備考:	存在していないテキストファイルに対して本文を実行すると実行時 Error となります。 格納先チャネルに、文字属性参照チャネルのみ記述した場合は、読み出すセルの値をすべて文字属性で読み出します。数値属性参照チャネルのみ記述した場合、読み出すセルに半角数字記載が存在すると数値変換し、存在しないと当該 index は null となります。指定された読み出し行、または列が存在しない場合は、存在したところまで読み出します。 なお、項目区切り文字無視(ファイル番号定義文での属性 chr)の場合、列は 1 個しか存在していません。

2. 3. 24. テキストフォーム・セル書き込み文 (テキストファイル取扱文)

機能: 定義されている仮想シート上のセルへの書き込みを行います。

文法: write cell % ファイル番号 書き込み開始行列番号 書き込み方向フラグ 書き込みデータ列

引数:

【ファイル番号】<必須><%即値>

ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。

【書き込み開始セル番号】<必須><即値/数値属性参照チャネル><index 指定可>

書き込み開始するセル番号の行番号と列番号を半角カンマ","で区切って記述します。

※ 行列番号は、左上セルの行列番号を 1,1 として定義されています。

【書き込み方向フラグ】<必須><即値/数値属性参照チャネル>

書き込み方向フラグを記述します。記述する書き込みフラグを下表に示します。。

値	機能
0	チャネルの要素を横方向(列方向)、チャネルを縦方向(行方向)にデータのみ書き込む
1	チャネルの要素を縦方向(行方向)、チャネルを横方向(列方向)にデータのみ書き込む
2	チャネルの信号名を横方向(列方向)に書き込む、信号名の無いデータ並びは空欄
3	チャネルの信号名を縦方向(行方向)に書き込む、信号名の無いデータ並びは空欄
4	チャネルの単位を横方向(列方向)に書き込む、単位の無いデータ並びは空欄
5	チャネルの単位を縦方向(行方向)に書き込む、単位の無いデータ並びは空欄
6	書き込み方向はフラグ 4と同じですが、信号名と単位を連結して書き込む
7	書き込み方向はフラグ 5と同じですが、信号名と単位を連結して書き込む
8	書き込み方向はフラグ 0と同じですが、文字列チャネルに含む区切り文字が無効
9	書き込み方向はフラグ 1と同じですが、文字列チャネルに含む区切り文字が無効

書き込み方向フラグ 2～5 は、書き込みデータ列をチャネル記述で行った時に有効で、即値記述は無視され、当該チャネルに付けられている信号名および単位を書き込みます。なお、記述されたチャネルが複数要素であっても書き込みセルは 1 個となります。

【書き込みデータ列】<必須><即値/数値属性チャネル/文字列即値/文字属性参照チャネル><index 指定可><間接指定可>

書き込むデータを即値、数値属性参照チャネル、文字列即値または文字属性参照チャネルで記述します。複数のチャネル、または即値を記述する場合は、半角カンマ","で区切って記述します。書き込みデータ列をチャネルで指定した場合 Format 指定が可能です。指定方法は、チャネルごとに、チャネル番号に続き半角括弧( )で Format 指定を記述します。

フォーマット記述則

① Exponential 形式の場合 ⇒ Em m は符号、小数点を含まず全桁設定します。
記述例:(E3) 12.345 → 1.23e1 0.0000123 → 1.23e-5
② Significant Format 形式の場合 ⇒ Sm m は符号、小数点を含まず有効桁数全桁を設定します。
記述例:(S5) -12.34567→-12.345 123.4567→123.45 0.1234567→0.12345
③ Fractal Format 形式の場合 ⇒ Fm m は小数点以下の桁数を設定します。
記述例: (F3) -12.234567→-12.234 123.4567→123.456 0.1234567→0.123
④ 文字属性チャネルの場合 ⇒ Am m は表示文字数を設定します。文字数省略ができます。
記述例:(A3) "ABCdef" → "ABC"
なお、Format 指定は即値、または書き込み方向 2～4 を記述した場合は指定できません。

記述例: write cell %1 $1,$2 0 $3 (F0),$4(F4),&3

備考:

書き込み開始行列番号から書き込み方向フラグにしたがって書き込みデータ列の内容を書き込みますが、現在設定されている仮想シートを越えた要素は書き込みません。書き込み方向フラグと書き込込まれた結果を示します。書き込むチャネルの内容を

&1(0)="aa",&1(1)="bb",&1(2)="cc",&1(3)="dd"
$2(0)=123,$2(1)=456,$2(2)=789
&3(0)="abc,def",&3(1)="ghi,jkl",&3(2)="mno" として、

```
write cell %1 1,1 0 "xyz",&1,$2,&3,1122  /* フラグ = 0 要素横、チャネル縦*/
write cell %1 1,1 1 "xyz",&1,$2,&3,1122  /* フラグ = 1 要素縦、チャネル横*/
write cell %1 1,1 8 "xyz",&1,$2,&3,1122  /* フラグ = 8 要素横、チャネル縦*/
write cell %1 1,1 9 "xyz",&1,$2,&3,1122  /* フラグ = 9 要素縦、チャネル横*/
```

実行した場合、仮想シート上には次の様に書き込まれます。ただし、フラグ 0,1 で書き込んだ場合で、要素に区切り文字を含んでいると、格納した仮想シートを表示した場合に区切り文字に反応して書き込んだ行列番号と表示された行列番号が異なることがあります。したがって、区切り文字そのものを含む文字列を書き込む場合は、フラグ 6,7 での書き込みを行います。

flag = 0

	C1	C2	C3	C4	C5	C6
R1	xyz					
R2	aa	bb	cc	dd		
R3	123	456	789			
R4	abc	def	ghi	jkl	mno	
R5	1122					
R6						

flag = 1

	C1	C2	C3	C4	C5	C6
R1	xyz	aa	123	abc	1122	
R2		bb	456	def		
R3		cc	789	ghi		
R4		dd		jkl		
R5				mno		
R6						

flag=8

	C1	C2	C3	C4	C5	C6
R1	xyz					
R2	aa	bb	cc	dd		
R3	123	456	789			
R4	abc,def	ghi,jkl	mno			
R5	1122					
R6						

flag=9

	C1	C2	C3	C4	C5	C6
R1	xyz	aa	123	abc.def	1122	
R2		bb	456	ghi,jkl		
R3		cc	789	mno		
R4		dd				
R5						
R6						

本文は、同じ仮想シートに対して複数行記述できます。ただし、すでに書き込まれているセルに書き込みを行った場合、当該セルの内容は上書きされます。

2. 3. 25. テキストファイル格納文（テキストファイル取扱文）

機能:	write cell 文で書き込んだ仮想シートを格納します。
文法:	**save text_form** %ファイル番号
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。
記述例:	def file_id %1 "result" csv save text_form %1
備考:	格納される行列数は、現在書き込まれている最大行および最大列となります。  書き込まれていない仮想シートを格納すると実行時 Error となります。 また、同じ名前のファイルがすでに存在している場合は、上書きされます。

2. 3. 26. プログレスステータス書き込み文（パラメータ選択表示関連文）

機能:	Script 実行時に表示されるプログレスステータス･ダイアログにメッセージを表示します。
文法:	**write progress_status** 表示文
引数:	【表示文】<必須><文字列即値/文字属性参照チャネル><index 指定可> プログレスステータス･ダイアログに表示する内容を記述します。複数要素を持つ文字属性参照チャネルで記述した場合、表示され る最大行数は、2 行までとな ります。
記述例:	write progress_status "No.1 file 処理中"
備考:	プログレスステータス･ダイアログに表示したメッセージは、新たに書き換えるか、Script の実行が終了するか、または、null 表示文""を書き込まない限り、表示されたままとなります。

2. 3. 27. グラフ描画文（グラフ描画関連文）

機能:	グラフ属性 x-y で定義したグラフにグラフにデータを描画します。
文法:	**plot** @グラフ番号 X 軸チャネル列 Y 軸チャネル列
引数:	【グラフ番号】<必須><先頭文字%に続き即値> グラフ番号は先頭文字半角"@"に続き即値で記述します。 記述するグラフ番号は、先行して def graph_id 文により定義が必要です。 【X 軸チャネル列】<X 軸 linear の時省略可><数値属性参照チャネル><間接指定可><連続指定可> グラフの X 軸データが格納された数値属性参照チャネルを意味します。 複数チャネルを記述する場合、半角カンマ","で区切って記述します。チャネル列(複数チャネル)で記述した場合、対応する Y 軸チャネル列の先頭から順次参照され、Y 軸のチャネル列個数に不足した場合、X 軸チャネル列は頭から循環して参照されます。なお、X 軸データの要素数は対応する Y 軸データの要素数に原則一致している必要があります。要素数が一致していない場合、X 軸 Y 軸何れか少ない要素数範囲で描画されます。 記述省略された場合は、X 軸は Index 並びと解釈されます。ただし X 軸属性が log の場合は、記述省略できません。 【Y 軸チャネル列】<必須><数値属性チャネル><間接指定><連続指定可> グラフの Y 軸データが格納された数値属性参照チャネルを記述します。 複数チャネルを記述する場合、半角カンマ","で区切って記述します。
記述例:	plot @2 $3,$4,$4 $1,$2,#3 %4 /*$3,$4,$4 が X 軸チャネル、$1,$2,#3 が Y 軸チャネルを意味します*/
備考:	同じグラフ番号で plot 文を繰り返すと、同じグラフ番号で直前に設定されている設定値が参照され新たにグラフを描画します。

2. 3. 28. グラフ格納文（グラフ描画関連文）

機能:	File_id 文で取扱属性を grp で定義したグラフ番号のグラフ仮想空間に描画し描画結果を指定されたファイル番号に bmp 形式で格納します。
文法:	**save plot** %ファイル番号 @グラフ番号 X 軸チャネル列 Y 軸チャネル列
引数:	【ファイル番号】<必須><%即値> ファイル番号を先頭文字"%"に続き即値で記述します。記述する範囲は 1～255 となります。 記述するファイル番号はファイル ID 定義文で定義済みの必要があります。 【グラフ番号】<必須><%即値> グラフ番号を先頭文字"@"に続き即値で記述します。 記述するグラフ番号はグラフ ID 定義文で定義済みの必要があります。 【X 軸チャネル列】<X 軸 linear の時省略可><数値属性参照チャネル><間接指定可><連続指定可> グラフの X 軸データが格納された数値属性参照チャネルを意味します。 複数チャネルを記述する場合、半角カンマ","で区切って記述します。チャネル列(複数チャネル)で記述した場合、対応する Y 軸チャネル列の先頭から順次参照され、Y 軸のチャネル列個数に不足した場合、X 軸チャネル列は頭から循環して参照されます。なお、X 軸データの要素数は対応する Y 軸データの要素数に原則一致している必要があります。要素数が一致していない場合、X 軸 Y 軸何れか少ない要素数範囲で描画されます。 記述省略された場合は、X 軸は Index 並びと解釈されます。ただし X 軸属性が log の場合は、記述省略できません。 【Y 軸チャネル列】<必須><数値属性チャネル><間接指定><連続指定可> グラフの Y 軸データが格納された数値属性参照チャネルを記述します。 複数チャネルを記述する場合、半角カンマ","で区切って記述します。
記述例:	save plot %1 @2 $1,$2 $3,$4
備考:	予めファイル番号定義文で取扱属性を grp としたファイル番号を指定する必要があります。

2.3.29. バイナリファイル読み出し文（バイナリファイル取扱関連文）

機能: バイナリファイルから指定したデータ個数分読み出します。

文法: read bin %n 読み出し開始バイト位置 読み出し数 格納先 読み出し後バイト位置　読み出し属性

引数:

【ファイル番号】<必須><先頭文字半角%に続く即値>
ファイル番号は予めファイル ID 定義文で取扱属性 bin として定義しておく必要があります。

【読み出し開始バイト位置】<必須><即値/数値属性参照チャネル><index 指定可>
読み出し開始位置をファイル先頭からのオフセット byte 数で記述します。

【読み出し数】<必須><即値/数値属性参照チャネル><index 可>
読み出すデータ個数を記述します。読み出し Byte 数は、読み出し属性に依存します。
読み出し個数が 0 の場合は、特別な意味を持ち、ファイルの EOF までを意味します。
形式キーワードは以下の通りです。

データ形式	属性	読み出し数指定	格納先属性
バイト単位	byte	バイト数単位	数値属性
符号付き整数形式(2byte)	int16	データ個数単位	数値属性
符号なし整数形式(2byte)	uint16	データ個数単位	数値属性
符号付き倍精度整数形式(4byte)	int32	データ個数単位	数値属性
符号無し倍精度整数形式(4byte)	uint32l	データ個数単位	数値属性
単精度浮動小数点形式(4byte)	float32	データ個数単位	数値属性
倍精度浮動小数点形式(8byte)	float64	データ個数単位	数値属性
文字列(改行迄)	char	行数単位	文字属性
符号付整数モトローラ(ビッグエンディアン)形式(2byte)	int16m	データ個数単位	数値属性

【格納先】<必須><数値属性/文字属性参照チャネル><間接指定可>
読み出したデータの格納先チャネルを記述します。数値属性参照チャネルを記述するか文字属性参照チャネルを記述するかは、後述する読み出し属性に依存し、読み出し属性が chre の場合は必ず文字属性参照チャネルを記述し、その他全ては数値属性参照チャネルを記述します。なお、読み出し属性と不整合な場合は、sntax.Error となります。

【読み出し後バイト位置】<必須><数値属性参照チャネル>
指定数読み出し後、次のバイト位置が戻ります。ただし、次のバイト位置が EOF の場合は-1 が戻ります。

【読み出し後バイト位置】<必須><数値属性参照チャネル>
戻りオフセットは読み出し終了後の次のバイト位置を示します。ただし、次のバイト位置が EOF の場合は-1 が戻り　ます。

【読み出し属性】<必須><keyword>
byte、int16、uint16、int32、uint32、float32、float64、int16m、chr から選択記述します。

記述例:

```
def file_id %1 “test.dat” bin
read bin %1 2048 12000 $1 int16
```

備考: 数値属性戻りの場合,Script 記述内では指定した読み出し属性に拘わりなく Double に変換されます。

2. 3. 30. バイナリファイルチャネル構造読み出し文(バイナリファイル取扱関連文)

機能: バイナリファイルから構造記述した参照チャネルにデータを読み出します。

文法: **read bin_struct** %ファイル番号 開始オフセット 読み出し数＜格納先チャネル:形式＞ 読み出し後オフセット

引数:

【ファイル番号】＜必須＞＜%即値＞
予めファイル番号定義文(def file_id 文)で属性 bin として定義したファイル番号を記述します。

【開始オフセット】＜必須＞＜即値/数値属性参照チャネル＞＜間接指定/index 指定可＞
読み出しを開始するアドレスをバイト単位で求めたファイルの先頭からのオフセットを記述します。
ファイルに存在しないオフセット値が記述された場合は実行時 Error となります。

【読み出し数】＜省略可＞＜即値/数値属性参照チャネル＞＜間接指定/Index 指定可＞
読み出し数に続き半角"<>"内に記述する格納先チャネル:形式列を繰り返して読み出す回数を記述します。
なお、記述省略した場合は 1 と見なします。また、0 と記載した場合は特別な意味をもち、ファイルの EOF まで
読み出すことを意味します。但し、ネスト記述される読み出し数に0は記載できません。
※ 読み出し数記述に続いて半角"<"が記述されなくてはなりません。

【格納先チャネル】＜必須＞＜数値属性/文字属性参照チャネル＞＜チャネル番号連続指定／間接指定可＞
格納先チャネルを半角コロンで接続して型式キーワードとペアで記述します。複数の格納先チャネルを記述する場合は半角カンマで区切って記述します。
形式キーワードは格納先チャネル毎に記述しますが、格納先チャネル記述をチャネル番号連続記述則で記述した場合は、全て同じ形式が適用されます。
100<$2:int16,$3:int16,$4:int16> は 100<$[2,3]:int16> と同じ意味を持ちます。

【形式】＜必須＞＜キーワード＞
型式キーワードは以下の通りです。

データ形式	キーワード	備考
バイト単位	byte	byte に続きバイト数記述 注1
符号付き整数形式(2byte)	int16	
符号なし整数形式(2byte)	uint16	
符号付き倍精度整数形式(4byte)	int32	
符号無し倍精度整数形式(4byte)	uint32	
単精度浮動小数点形式(4byte)	float32	
倍精度浮動小数点形式(8byte)	float64	
文字列(改行迄)	文字数	半角換算での文字数を記述
符号付整数モトローラ形式(2byte)	int16m	ビッグエンディアン

注1:バイト形式は、一回の読出しで記述されている格納先チャネルの要素を連続的に使用する個数をキーワード byte に続き即値で記述します。つまり、byte 形式は格納先チャネルの要素毎に 1byte 毎に格納されます。
例えば、読出し数 3 として格納先に$5:byte5 と記述された場合、一回目の読み出しで＄5(0)～$5(4)までの5要素に格納され、三回読出し後全体で 3*5＝15byte、要素$5(0)～$5(14)に格納されます。従って他の形式は一回の読み出し結果は1つの要素に格納されることと異なる点に留意ください。
なお、バイト数を記述省略された場合は 1 と見なします。

【読出し後オフセット】＜必須＞＜数値属性参照チャネル＞
読出し終了後、読出し最終バイト位置+1 の読出し終了後オフセット格納先を数値属性参照チャネルで記述します。

記述例:
```
read bin_struct %1 512 0<73<$[1,3]:int16,$4:byte>,$5:byte> $6
read bin_struct %1 0 1<&1:12> $6
```

備考:
格納先チャネル構造記述内に格納先チャネル構造記述を記述することができます。
例えば、読出し先ファイルが 512byte を 1 ブロックして記録されブロックの中が、符号付2バイト整数がch1～ch3 と 1byte のステータス情報の 7byte で 1 レコード構成し、73 レコードと終端コード 1byte で1ブロックとなっている場合に EOF を読み出す場合について例を示します。
1 レコードは <$1:int16,$2:int16,$3:int16,$4:byte> 又は <$[1,3]:int16,$4:byte>
1 ブロックは <73<$[1,3]:int16,$4:byte>,$5:byte>
従って、EOF まで一度に読出す場合は、読出し数に 0 として
```
read bin_struct %1 512 0<73<$[1,3]:int16,$4:byte>,$5:byte> $6
```
と記述することができます。

2.3.31. バイナリバッファ書き込み文(バイナリファイル取扱関連文)

機能: 定義されているバイナリバッファへ書き込みデータ形式を指定して書き込みます。

文法: **write bin_buf *バッファ番号 書き込み開始バイト位置 書き込み ch 列 書き込み後バイト位置 フラグ**

引数:

【バッファ番号】＜必須＞＜*即値＞
バッファ番号を記述します。記述則は先頭文字半角"*"に続いて即値で記述します。
記述するバッファ番号はバイナリバッファ確保定義文で定義されている必要があります。

【書き込みバイト位置】＜必須＞＜即値/数値属性参照チャネル＞
バイナリバッファは先頭 0 としてアドレスされます。書き込み開始するバイト位置を記述します。
記述するバイト位置は定義したバイナリバッファアドレスとして存在していない場合は実行時 ERROR となります。

【書き込み内容】＜必須＞＜数値属性参照チャネル/文字属性参照チャネル＞
書き込みチャネルを書き込み属性と一緒に記述します。複数チャネルを記述する場合は半角カンマで区切って記述します。なお、文字属性参照チャネルを記述した場合で、当該文字属性参照チャネルの書き込み属性が char 場合は、文字列の長さに書き込みバイトが決定されますので、後続する書き込み位置がずれる場合がありますので注意が必要です。

【書き込み属性】＜必須＞＜キーワード＞
参照チャネルに続き半角"‹"と"›"で囲んで書き込み属性キーワードを記述します。
形式キーワードは以下の通りです。

書き込み属性	内容	書き込み ch
byte	0～255 以内の数値	,数値属性
int16	符号付き 2byte 整数形式	数値属性
uint16	符号無し 2byte 整数形式	数値属性
int32	符号付き 4byte 整数形式	数値属性
uint32	符号無し 4byte 整数形式	数値属性
float32	単精度浮動小数点形式(4yte)	数値属性
float64	倍精度浮動小数点形式(8byte)	数値属性
数値※1	書き込み文字列の長さ指定	文字属性
int16m	符号付 2byte 整数モトローラ(ビッグエンディアン)形式	数値属性

※1 書き込み文字数(半角換算)を意味します。例えば、4 と記述し書き込み内容に文字属性参照チャネル&1 を、書き込み属性 char とした場合は、&1 の各要素は先頭から半角勘定で 4 文字分毎全ての要素が連続して書き込まれます。なお、要素の内容がここで記述された文字数より短い場合は余りバイトには null が書き込まれます。文字属性参照チャネル以外に指定すると文法 Error となります。

【書き込み後バイト位置】＜必須＞＜数値属性参照チャネル＞
書き込み終了後のバイト位置の格納先を数値属性参照チャネルで記述します。確保定義されているバイナリバッファサイズを越えて書き込みを行った場合、バッファフルとなり、-1 が戻ります。書き込み自体は指定されたバッファサイズの終端まで書き込まれます。

【フラグ】＜省略可＞＜即値/数値属性参照チャネル＞
書き込み方法を意味するフラグで0又は1を記述します。
0 を記述/記述省略した場合:
書き込みチャネル列に記述されたチャネル順に指定されたフォーマットで書き込まれます。
例えば、書き込みチャネル列に$1:int16,$2:float32 と記述された場合、$1 の全ての要素を 2byte 整数形式で書き込んだ後、$2 の全ての要素を 4byte 浮動小数点形式で書き込みます。
1 を記述した場合:
書き込みチャネル列に記述されたチャネルの index 順に指定されたフォーマットで書き込まれます。但し、記述したチャネル列の要素数はすべて同じでなければなりません。要素数の異なるチャネルを含んでいる場合は、実行時 Error となります。
例えば前述した例と同様に書き込みチャネル列に$1:int16,$2:float32 と記述された場合、
$1(0),$2(0),$1(1),$2(1),$1(2),$2(2),$1(3),$2(3)…の様に Index 順に$1 は 2byte 整数形式、$2 は 4byte 浮動小数点形式で書き込みます。

記述例: write bin_buf *1 $1:int16,$2:float32,&1:10 $4

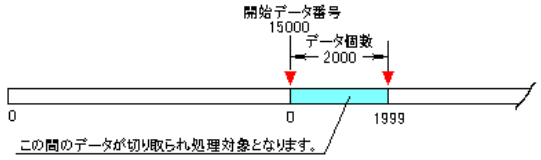
備考:
書き込み内容と書き込み属性による制限
属性 byte の場合は、0～255 範囲となります。
属性 int16、int16m の場合は-32768～+32767 範囲、小数点以下は切り捨てされます。
属性 uint16 の場合は 0～65535 範囲、小数点以下は切り捨て、但し負数の場合実行時 Error
属性 int32 の場合は、-2147483647～+2147483648 範囲、小数点以下は切り捨てされます。
属性 uint32 の場合は、0～4294967295 範囲、小数点以下は切り捨て、但し負数の場合は実行時 Error
なお、何れも文字属性参照チャネルに記述した場合或いは範囲を越えた場合は実行時 Error となります。

2. 3. 32. バイナリファイル格納文(バイナリファイル取扱関連文)

機能:	記述されたバッファ番号の内容をファイルに書き込みます。
文法:	**save bin_buf %ファイル番号 *バッファ番号列 フラグ**
引数:	【ファイル番号】＜必須＞＜%s 即値＞ 格納先ファイル番号を先頭文字半角"%"に続き即値で記述します。ファイル番号は、直前までに def file_id 文によって属性 bin で定義されている必要があります。 【バッファ番号】＜必須＞＜*即値＞ 格納するバッファ番号を先頭文字半角"*"に続き即値で記述します。同時に複数個のバイナリバッファを格納する場合は、半角カン マで区切って記述します。存在しないバッファ番号含んでいる場合、実行時 error となります。 【フラグ】＜省略可＞＜即値/数値属性参照チャネル＞ 追記格納するのか否かを示すフラグを意味します。0 と記述した場合は、上書き格納され、1 と記述された場合は追記処理されます。記述省略された場合は 0:上書き格納を意味します。
記述例:	save bin_buf %1 *1,*2,*3 1 /*バイナリバッファ *1,*2,*3 を%1 で定義したファイルに追記処理されます*/
備考:	

2. 4. 演算処理ブロック制御文

2. 4. 1. データ番号実行制御文 （演算処理記述関連文）

機能:	本文の直下に記述された演算処理ブロックで記述されるチャネルの演算対象囲を Index で指定します。 ※ 解析対象ファイル(収録ファイル)が存在していないと実行時 Error となります。
文法:	index 開始データ番号 データ個数
引数:	【開始データ番号】<必須><即値/数値属性参照チャネル><Index 指定可> 開始データ番号(index)を記述します。 【データ個数】<必須><即値/数値属性参照チャネル><Index 指定可> データ個数を記述 します。
記述例:	index 15000 2000
備考:	演算処理ブロック内に記述されるチャネルは、本文で設定された開始データ番号を 0 とし、データ個数分の長さに制御します。例えば、index 15000 2000 とした場合、演算処理ブロック内で記述されるチャネルの直前のデータ番号 15000 から 2000 個のデータが抽出され、データ番号 0～1999に振り替えられます。  $1 ← 0,10,20,30,40,50,60,70,80 index 2 2 proc test{ $2 = $1 $1 = $1*3 }test $1 → 0,10,60,90,40,50,60,70,80 proc test 内で切り取られた index2,3 の値のみ 2 倍される $3 → 20,30 proc test 内で切り取られた$1 の index2,3 の値が格納される なお、記述されているチャネルが、事前に dcl exempt_ch 文で宣言されていると、当該チャネルは制御されません(切り取られません)。開始データ番号、データ個数の双方あるいは何れかを、複数の要素から成る参照チャネルで設定した場合、その要素ごとに設定され実行を繰り返します。また、開始データ番号の参照チャネルとデータ個数の参照チャネルで要素数が異なる場合、何れか多い要素数で繰り返し、少ない要素数は循環して参照されます。 なお、制御される演算処理ブロック内に記述した dcl exempt_ch 宣言していないチャネルで要素数が複数のチャネルに設定した開始データ番号およびデータ個数が現在の解析範囲に無いときは、直下に記述される演算処理ブロックを実行しません。

2. 4. 2. マーク番号実行制御文 （演算処理記述関連文）

機能:	本文の直下に記述された演算処理ブロック内で参照するチャネルの演算対象範囲をマーク番号で指定します。 ※ 解析対象ファイル(収録ファイル)が存在していないと実行時 Error となります。
文法:	mark 開始マーク番号 終了マーク番号
引数:	【開始マーク番号】<必須><即値/数値属性参照チャネル><Index 指定可> 開始マーク番号を記述します。 【終了マーク番号】<必須><即値/数値属性参照チャネル><Index 指定可> 終了マーク番号を記述し ます。 開始マーク番号 < 終了マーク番号 で無ければなりません。
記述例:	mark 2 4
備考:	演算処理ブロック内で記述されるチャネルのデータ番号は、設定された開始マーク番号位置の index を 0 とし、終了マーク番号位置 index-1 までのデータ個数で再構成されます。 ただし、dcl exempt_ch 文で宣言されたチャネルは制御を受けません。 本文の直前までに設定された解析範囲に、本文で設定した開始マーク番号および終了マーク番号の双方が存在している必要がありま す。本条件を満たさない場合、あるいは、開始マーク番号>=終了マーク番号の場合、直下に記述される演算処理ブロックは実行されま せん。 なお、Mark 番号は、収録ファイルに付けられている Mark 番号と異なり、本文の直前までに設定された解析範囲に存在するマーク出現順に 1 から割り当てられた相対マーク番号を意味します。 開始マーク番号、終了マーク番号の双方を、複数の要素から成る参照チャネルで設定した場合、その要素が参照され、要素数分繰り返して実行されます。なお、要素数は同じでなくてはなりません。

2. 4. 3. 条件実行制御文 （演算処理記述関連文）

機能:	本文で設定した条件が成立した場合、本文の直下に記述された演算処理ブロックを実行します。
文法:	**case** 比較値 条件 比較値
引数:	【比較値】<必須><即値/文字列即値/数値属性/文字属性参照チャネル><Index 指定可> 比較値を記述します。複数要素を持つ数列で記述しても index0 のみ参照されます。 また、比較値同士が同じ属性の必要があります。 【条件】<必須><Keyword> Keyword です。比較条件を記述します。

KeyWord	内容
>	左側比較値が右側比較値より大きい
<	左側比較値が右側比較値より小さい
>=	左側比較値が右側比較値より大きいか等しい
<=	左側比較値が右側比較値より小さいか等しい
=	左側比較値と右側比較値が等しい
<>	左側比較値と右側比較値が等しくない

記述例:	case $1 > 100 case $1(1) = $2($5) case “test” = &3($2)
備考:	

2. 4. 4. 論理成立実行制御文 （演算処理記述関連文）

機能:	本文で設定した参照チャネルの論理が”1”の場合、本文の直下に記述された演算処理ブロックを実行します。
文法:	**case_true** 論理値
引数:	【論理値】<必須><数値属性参照チャネル><Index 指定可> 論理値を記述します。
記述例:	case_true $3
備考:	記述した数値属性参照チャネルの値が 0 または負数の場合は論理”0”、0 より大きい場合は、論理”1”として扱われます。

2. 4. 5. 論理非成立実行制御文 （演算処理記述関連文）

機能:	本文で設定した参照チャネルの論理が”0”の場合、本文の直下に記述された演算処理ブロックを実行します。
文法:	**case_false** 論理値
引数:	【論理値】<必須><数値属性参照チャネル><Index 指定可> 論理値を記述します。
記述例:	case_false $3
備考:	設定した参照チャネルの値が 0 または負数の場合は論理”0”、0 より大きい場合は論理”1”として扱われます。

2. 4. 6. 繰り返しデータ番号実行制御文 （演算処理記述関連文）

機能:	本文の直下に記述される演算処理ブロックの演算対象範囲 index で繰り返し指定しします。 ※ 解析対象ファイル(収録ファイル)が存在していないと実行時 Error となります。
文法:	**repeat_index** 開始データ番号 データ個数 終了データ番号
引数:	【開始データ番号】<必須><数値属性参照チャネル><Index 指定可> 開始データ番号(Index)を記述します。 【データ個数】<必須><即値/数値属性参照チャネル><Index 指定可> 開始データ番号からのデータ個数を記述します。 【終了データ番号】<必須><即値/数値属性参照チャネル><Index 指定可> 終了データ番号(index)を記述します。
記述例:	repeat_index $2 1000 10000 repeat_index $2 $3 $4
備考:	データ番号制御を受ける収録チャネル/参照チャネルは、Index 文と同じです。繰り返しは、開始データ番号に設定されたデータ個数分が加算されて自動更新され終了データ番号に等しいかまたは大きい場合に繰り返しを停止します。なお、制御されている演算処理ブロック内で、本文で設定している参照チャネルを明示的に変更しても問題ありません。なお、開始データ番号、データ個数、終了データ番号を複数の要素を持つ参照チャネルで設定しても、参照されるデータ番号は 0 のみとなります。 なお、開始データ番号またはデータ個数が現在の解析範囲に無いときは、直下に記述される演算処理ブロックを実行しません。

2. 4. 7. 繰り返しマーク番号実行制御文 （演算処理記述関連文）

機能:	本文の直下に記述される演算処理ブロック内の演算対象範囲をマーク番号で繰り返し制御します。 ※ 解析対象ファイル(収録ファイル)が存在していないと実行時 Error となります。
文法:	repeat_mark 開始マーク番号 飛び越し数 終了マーク番号
引数:	【開始マーク番号】<必須><数値属性参照チャネル><Index 指定可> 開始マーク番号を記述します。 【飛び越し数】<必須><即値/数値属性参照チャネル><Index 指定可> 飛び越すマーク個数を記述します。 【終了マーク番号】<必須><即値/数値属性参照チャネル><Index 指定可> 終了マーク番号を記述します。
記述例:	repeat_mark $3 2 10
備考:	データ番号制御を受けるチャネルは、mark 文と同じ振る舞いとなります。 繰り返しは、開始マーク番号に設定されたマーク番号に飛び越し数+1 が加算されて、終了マーク番号に等しいかまたは大きい場合に 繰り返しを停止します。 repeat_mark 2 1 5  なお、制御される演算処理ブロック内に開始マーク番号、および開始マーク番号+1 が解析範囲に存在しない場合は直下の演算処理 ブロックを実行しません。

2. 4. 8. 繰り返し条件実行制御文 （演算処理記述関連文）

機能:	本文で設定した条件が成立している間、本文の直下に記述された演算処理ブロックを繰り返し実行します。
文法:	repeat_case 比較値 1 条件 比較値 2
引数:	【比較値】<必須><即値/文字列即値/数値属性/文字属性参照チャネル><Index 指定可> 比較値を記述します。ただし何れか一方は参照チャネルで記述します。 本文にて制御される演算処理ブロック内で、参照チャネルで記述した比較値を更新し、条件不成立の状態を生成する必要がありま す。更新しないと無限 loop となります。 なお、比較子同士の属性は一致している必要があります。 【条件】<必須> Keyword で比較条件を記述します。 Keyword / 内容 > / 左側比較値が右側比較値より大きい < / 左側比較値が右側比較値より小さい >= / 左側比較値が右側比較値より大きいか等しい <= / 左側比較値が右側比較値より小さいか等しい = / 左側比較値と右側比較値が等しい <> / 左側比較値と右側比較値が等しくない
記述例:	repeat_case $1 > 100
備考:	演算処理ブロックを 0 から 9 までの 10 回 Loop させる例を記述します。 $1 = 0 repeat_case $1 < 10 proc test{ $1 = $1+1 }test 演算処理ブロック test は 10 回実行されます。

2.4.9. 繰り返し実行制御文（演算処理記述関連文）

機能:	本文の直下に記述された演算処理ブロックを本文で指定された開始値から増分値を加算して終了値に達するまで繰り返して実行します。
文法:	**repeat** 制御変数 開始値 増分　終了値
引数:	【制御変数】<必須><数値属性参照チャネル><Index 指定可> 制御変数を数値属性参照チャネルで記述します。実行時に値が決定されますので、特に初期値が格納されている必要はありません。 【開始値】<必須><即値/数値属性参照チャネル> 開始値を記述します。開始値は制御変数初期値を意味します。 【増分】<必須><即値/数値属性参照チャネル> 増分値を記述します。繰り返し時の制御変数加算値を意味します。 【終了値】<必須><即値/数値属性参照チャネル> 繰り返し終了値を記述します。。
記述例:	repeat $3 0 1 10
備考:	制御変数に開始値が代入され、実行するごとに設定されている増分が加算され、終了値を超えると繰り返しを停止します。なお、数値属性参照チャネルで記述した場合、制御される演算処理ブロック内で、明示的に各値を変更しても問題ありません。

2. 5. 演算処理ブロック定義文

2. 5. 1. 演算処理ブロック開始文（演算処理記述関連文）

機能:	直下に記述した一連の Script 文、または演算式が演算処理ブロックであることを示します。
文法:	**proc** 演算処理ブロック名{
引数:	【演算処理ブロック名】<必須><文字列> 半角ダブルコーテーション""で囲まず、文字列で記述します。演算処理ブロック名は全角半角を問いませんが先頭文字に"@"は使用できません。演算処理ブロック名は演算処理ブロック開始文と演算処理ブロック終了文は同じである必要があります。記述する演算処理ブロックが call proc で呼び出される Subroutine の場合、記述する演算処理ブロック名は Script 記述中で唯一無二である必要があります。
記述例:	proc 演算処理 {
備考:	演算処理ブロックの中に更に演算処理ブロックを構成することができます。

2. 5. 2. 演算ブロック終了文（演算処理記述関連文）

機能:	演算処理ブロック開始文で設定された演算処理ブロックの終結であることを示します。
文法:	}演算処理ブロック名
引数:	【演算処理ブロック】<必須><文字列> 対応する演算処理ブロック開始文に付けられた演算ブロック名と同じである必要があります。
記述例:	proc 演算処理 { 　| } 演算処理
備考:	

2. 6. 演算文･代入文

2. 6. 1. 代入文 (演算処理記述関連文)

機能:	代入先チャネルに代入値列で記述した内容を代入します。
文法:	**assign** 代入先チャネル “信号名:単位” = 代入値列 **assign** 代入先チャネル = 代入値列 **assign** 代入先チャネル “信号名:単位” ={代入値列} 代入値列は半角カンマ、改行で記述できます。 **assign** 代入先チャネル = {代入値列}
引数:	【代入先チャネル】<必須><数値属性/文字属性参照チャネル><Index 指定可><間接指定可> 数値属性参照チャネル、または文字属性参照チャネルで記述します。参照チャネル記述に Index 指定が使用できます。 【信号名:単位】<省略可><文字列即値> 信号名、単位とも文字列で記述し、信号名と単位を半角コロン”:”で区切り、全体を半角ダブルコーテーション””で囲みます。なお、信 号名または単位のみ記述する場合でも、区切り文字半角コロンが必要です。なお、信号名、単位が def ch_name 文(def ch_unit 文) で定義されている場合、定義文が優先され、ここで付けた信号名および単位名で更新されることはありません。 【代入値列】<必須><即値/文字列即値/収録/数値属性/文字属性参照チャネル><Index 指定可><チャネル連続指定可> 代入値列を半角”{“、”}”で括らない場合: 即値、または参照チャネルで半角カンマ”,”で区切って記述します。なお、Index 連続指定が使用できます。 文字属性参照チャネルに数値属性参照チャネル値を代入する場合、文字変換フォーマット設定を”( )”で囲んで記述します。変換フォーマット記述が無い場合、あるいは指定されたフォーマットに変換不可の場合は、省略形式変換で文字列に変換されます。また、文字列を即値で記述する場合は半角ダブルコーテーション” “で囲んで記述します。$err を代入する場合、代入先チャネルは数値属性で記述し代入値列は$err のみ記述 します。 代入値列を半角”{“、”}”で括る場合 記述する代入値は即値或いは文字列即値以外は記述できません。又、記述する代入値の属性は代入先属性と同じでなければ ならず、混在記述することはできません。なお、括弧内に記述する代入値は半角カンマ或いは改行で区切ります。
記述例:	assign $2 “油圧:Mpa” = $2,$4[0,3],$[5,2] assign &1 = $2(F6),”ABCD”,&3,$4[0,10](S5),&5[$6,$7] assign $2 “:m/s^2” = &1 assign &103 “ｼﾝｺﾞｳﾒｲ:ﾀﾝｲ” = &1|”--“|$2(S3),10<$2[0,2](S3)> assign &103($20) = &300 assign $3 “Error_Flag:” = $err assign “table:mm” $1 = { 123,1234,122,334,565,278 2212,1234,1223,3456,2345}
備考:	代入列の記述法補足(代入値列を“{}“で囲んだ場合は適用できません) ① 連結して代入する場合:半角”|”で結合して記述します。 記述 例: &1|”--“|$2|”⇒首都” &1 と$2 が複数の要素で構成される配列型の場合、結合する配列型参照チャネルの何れか個数の多い数となります。即値、または、単一要素の参照チャネルは、代入される個数分繰り返して参照されます。 &1 の要素 $2 の要素 代入される内容 &1(0)→”Tokyo” $2(0)→100 Tokyo--100⇒首都 &1(1)→”Seoul” $2(1)→200 Seoul--200⇒首都 &1(2)→”Beijing” $2(2)→300 Beijing--300⇒首都 $2(3)→400 --400⇒首都 $2(4)→500 --500⇒首都 ② 同じ値を繰り返して代入する場合: 文法: 繰り返し数<繰り返される代入列> ただし、繰り返される代入列をネスト構造にすることはできません。 記述例 1: 1000<200,50> 代入値 200 と 50 が 200,50,200,50…の様に 1000 回繰り返され代入されます。 記述例 2: $1<200> 繰り返し数は、$1 のデータ番号 0 の内容のみ参照され、代入値 200 が、$1 のデータ番号 0 の値分 200,200,200,200,….と代入されます。 記述例 3: $1<$2> 繰り返し数は、$1 のデータ番号 0 の内容のみ参照され、$2 が$1 のデータ番号 0 の値分代入されます。例えば$1 のデータ番号 0 の値が 10 で$2 が 3 個の要素を持つ配列型の場合、$2 の内容が 1,2,3 となっている場合、1,2,3,1,2,3,1,2,3,1,2,3…と 10 回繰り返され、合計 30 個の値が代入されます。 記述例 4: 2<$2,$3> $2 と$3 の内容が 2 回繰り返されることを意味し、$2 が要素数 5 で内容が 1,2,3,4,5、$3 が要素数 2 で内容が 10,20 となっている場合、1,2,3,4,5,10,20,1,2,3,4,5,10,20 が代入されます。

	③ 数値属性参照チャネルに文字属性参照チャネルを代入する場合 代入する文字属性参照チャネルに含まれる半角数字を数値に変換して代入します。数値変換できない場合は、0 が代入されます。 記述例: assign $2 = “123abc456” ⇒ 123 assign $2 = “abc12def34” ⇒ 12 assign $2 = “abc32” ⇒ 32 assign $2 = “abc-12df” ⇒ -12 assign $2 = “a1e+3b” ⇒ 1e+3 assign $2 = “a12.3e3” ⇒ 12.3e+3 assign $2 = “a.12.3” ⇒ 0.12

2.6.2. 演算文（演算処理記述関連文）

機能:	数値演算式を記述します。
文法:	格納先チャネル“信号名:単位”= 演算式
引数:	【格納先チャネル】<必須><数値属性参照チャネル><Index 指定可><間接指定可> 数値属性参照チャネルで記述します。参照チャネル記述にチャネル番号間接指定記述、および Index 指定記述が使用できます。 【信号名:単位】<省略可><文字列即値> 信号名、単位とも文字列で記述し、信号名と単位を半角コロン":"で区切り、全体を半角ダブルコーテーションで""で囲みます。なお、信号名または単位のみ記述する場合でも、区切り文字半角コロン":"が必要です。なお、すでに、信号名、単位が def ch_name 文 (def ch_unit 文)で定義されている場合、定義文が優先され、ここで付けた信号名および単位名で更新されることはありません。 【演算式】<必須><即値/収録/数値属性参照チャネル/関数><Index 指定可><間接指定可> PcWaveForm チャネル間演算機能と同じで、Operand と Operand の間に演算子を記述する Infix Notation 法を採用しています。演算式は全て半角英数字で記述します。オペランドは数値属性参照チャネル、直交座標系複素数属性参照チャネル、収録チャネル、即値、または関数が記述できます。演算順序は括弧を使用して行います。
記述例:	$2 “SPL:dB” = 20*LGT(RRT(1,WAC(#5))/20e-6) /* #5:=音圧信号(Pa) */ $3 “前後移動加速度実効値:m/s^2” = RRV(WBK(#1)) /* #1:=座位前後加速度(m/s^2) */ $4 “左右移動加速度実効値:m/s^2” = RRV(WBK(#2)) /* #2:=座位左右加速度(m/s^2) */ $5 “上下移動加速度実効値:m/s^2” = RRV(WBD(#3)) /* #3:=座位上下加速度(m/s^2) */ $6 “合成移動加速度実効値:m/s^2” = SQR((1.4*$3)^2+(1.4*$4)^2+$5^2) $7 “合成 MTVV:m/s^2” = MAX($6) /* 最大過渡振動値 */ $8 “MTVV 地点:sec” = MXP($6)*PRD() /* 最大過渡振動値発生地点 */
備考:	文法: 数値属性参照チャネル = 演算式 文字属性参照チャネル = 演算式 ただし演算式の演算結果属性が文字属性の場合 ※ 演算式の、“ = ”の前後は半角スペースが必要です。 ※ 信号名/単位定義文を使用せず演算式記述で信号名単位を付けることができます。 記述方法は、 参照チャネル“信号名:単位”= 演算式 "="の前にダブルコーテーションで囲んで信号名と単位を記述します。信号名と単位の区切り文字は半角コロン":"です。 なお、単位を記述しない場合でも区切り文字は記述必須となります。 演算子: オペランドが数値属性の場合: 加算“+” 減算 “－” 乗算 “*” 除算“/” べき乗算“^”で表します。 ただし、直交座標系複素数属性の参照チャネルはべき乗算はできません。また、オペランドに極座標系複素数属性の参照チャネル記述できません。極座標系複素数は使用する関数の引数として記述可能です。演算式の結果属性が左辺に記述したチャネルの属性として自動的に決定さます。したがって、演算式の結果属性が文字属性の場合は、左辺の参照チャネルは文字属性参照チャネルでなければなりません。 オペランドが文字属性の場合: 連結"|"のみ使用できます。 演算 順序: 演算式の左から右 べき乗→乗除算→加減算 の順となります。 演算順序制御: 括弧“(”、 “)”を使用します。 記述例:$1 = (#1+#3)*20 チャネルの指定方法: チャネルはそのまま記述できます。その場合、チャネルに格納されている内容すべてを一括して取り扱う ことを意味しています。記述するチャネルの index を指定して要素単位で扱う場合あるいは範囲指定する場合、およびチャネル番号を変数として取り扱う場合の記述方法を示します。 index 指定: 例:$1(2) $2($3) ()内に index を記述します。 index 指定に複数要素を持つ参照チャネルで、記述しても Index0 のみ参照されます。 左辺、右辺の何れにも記述できますが、指定された index がすでに存在している必要があります。 index 範囲指定: 例:$1[100,10] $2[$3,$4] []内に開始 index とデータ個数を記述します。 Index 指定およびデータ個数指定に複数要素を持つ参照チャネルで記述しても index0 のみ参照されます。 間接指定: 例: $($2) 属性を表す先頭文字に続く()内にチャンネル番号を記述します。 チャネル番号指定に複数要素を持つ参照チャネルで記述しても index0 のみ参照されます。 複数要素を持つ参照チャネルにチャネル番号列が格納されている場合は明示的に index 指定を併用して記述します。 下記に例を示します。 $2 = 1,3,6,7,8 $1 = $($2(2)) ← $1 に$6 の内容が代入されます。 演算 関数: 演算式に組み込み関数を使用できます。(演算関数については、別紙「Archi_1 演算関数 仕様書」をご参照下さい。) 関数名は、すべて半角大文字 3 文字で構成され、引数はカッコ内に半角カンマ","で区切って記述し、引数は、即値、収録チャネル、数値属性参照チャネル、または演算式で記述します。PcWaveForm のチャネル間演算機能で作成した演算式を使用する場合は、チャネル間演算 Window で作成した演算式ファイルを読み出し、そのまま貼り付けて使用することができます。

2. 7. ユーザ･インターフェース文

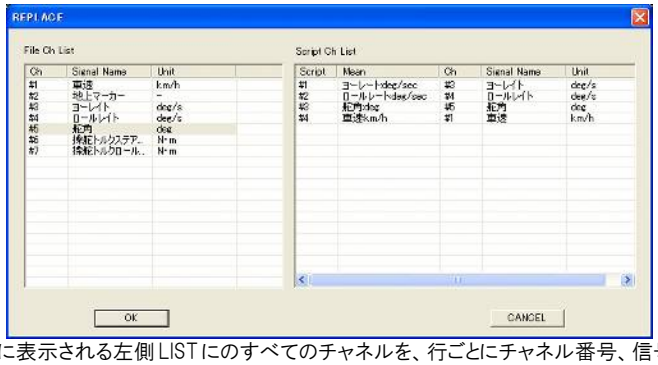
2. 7. 1. カレントフォルダ選択文 (ファイル取扱共通文)

機能:	参照フォルダダイアログを表示し、フォルダを指定します。
文法:	get folder_select 表題
引数:	【表題】<省略可><文字列即値/文字属性参照チャネル><index 指定可> ダイアログに表示する表題を記述します。複数要素を持つ参照チャネルで記述した場合、index0 と index1 の 2 行表示されます。
記述例:	get folder_select “収録ファイル格納先フォルダ選択”
備考:	実行されるとフォルダ参照ダイアログを表示し、カレントにしたいフォルダをクリックして選択します。ダイアログでキャンセルすると何もしません。  選択されたフォルダ名を取得する場合は、read folder_path 文、または、演算関数:カレントフォルダパス名取得関数 CFPN()を使用します。

2. 7. 2. ファイル選択文 (ファイル取扱共通文)

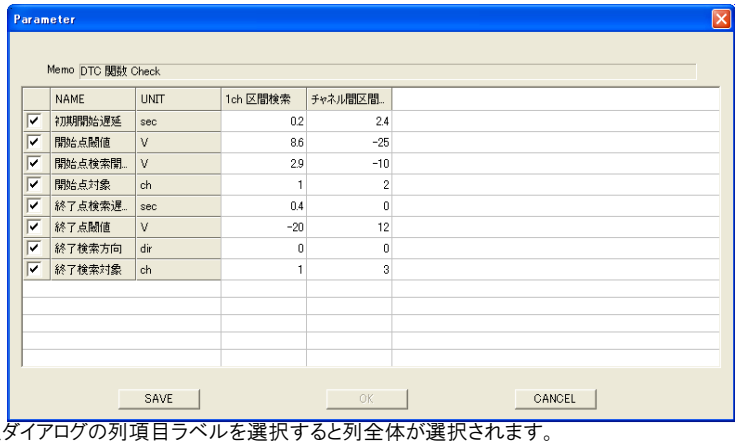
機能:	ファイル読み出しダイアログを表示させ、ダイアログからファイルを選択します。
文法:	get file_name ファイル名 ファイル個数 ファイル拡張子
引数:	【ファイル名】<必須><文字属性参照チャネル> 取得するファイル名格納先を記述します。複数ファイルが指定された場合は、戻り値は複数要素となります。 戻りファイル名の拡張子は後述する引数:ファイル拡張子設定で、複数個の拡張子を設定指定しないと付けられません。 【ファイル個数】<必須><数値属性参照チャネル> 選択されたファイル個数格納先を記述します。なお、ダイアログでキャンセルされた場合(選択されなかった場合)は 0 が戻ります。 【ファイル拡張子】<必須><文字列即値/文字属性参照チャネル> ダイアログに表示するファイルの種類(拡張子)を記述します。複数ファイル種類(拡張子)を記述する場合は複数要素で指定しま す。拡張子は半角ピリオド以降を指します。
記述例:	get file_name &1 $2 “kdt” get file_name &1 $2 “hdr”
備考:	get file_name 文が実行されるとカレントフォルダ内ファイル読み出しダイアログが表示されます。読み出したいファイルがカレントフォルダ以外の場合は、ダイアログを操作してフォルダを切り替えて選択できますが、予め def folder_path 文にてフォルダを定義しておくとダイアログ表示初期値として設定したフォルダの内容で設定したファイルの種類に一致した拡張子を持つファイルが表示されます。  フォルダが変更された場合は、暗黙的にカレントフォルダが変更されます。 なお、指定されたファイルが Open 可能か否かは、read file_check 文、または演算関数:ファイルステータス取得関数 RFC(&m)で確認する必要があります。

2. 7. 3. 収録チャネル割り当て文（解析対象ファイル情報取得文）

機能:	収録ファイルのチャネルと Script 中で記述する収録チャネル番号が異なっている可能性がある場合に、Script で記述した収録チャネル番号を解析対象収録ファイルのチャネル番号に一致させます。
文法:	get replace_ch 収録チャネル列
引数:	【収録チャネル列】<必須><収録チャネル><チャネル連続指定可> 割り当てる収録チャネル(Script で記述している収録チャネル)を半角カンマ","で区切って記述します。 【内容】<省略可><文字属性参照チャネル/戻り文字属性演算式> 収録チャネル列に記述したチャネルが Script 記述上で持つ意味を記述します。チャネル列に複数チャネル記述した場合は、文字属性配列で与え、各要素はチャネル列記述順に対応します。
記述例:	get replace_ch #3,#2,#5,#6 get replace_ch #[$1,$2]
備考:	収録チャネル列に記述するチャネルは、現在のに存在して居なくても良く、また、本文は、Script 中に複数回記述されても問題ありません。新たな get replace_ch 文に出会うまで、直前に設定された収録チャネル割り当てを維持します。本文が実行されるごとに収録チャネル割り当てダイアログが表示されますので、任意の収録ファイル上のチャネルを設定されている収録チャネルに割り当てることができます。 ダイアログの操作: 収録ファイル・チャネル割り当て文が実行されると、収録チャネル割り当て操作を行うダイアログが表示されます。  ダイアログ上に表示される左側 LIST にのすべてのチャネルを、行ごとにチャネル番号、信号名、単位が表示され、右側 LIST は、本文で設定された収録チャネルを行ごとに表示します。設定は、左側 LIST の行を選択し、ドラッグ&ドロップすることで行います。 割り当て初期値は、本文で設定したチャネル番号と同じチャネル番号がに存在する場合そのまま割り当てられ、存在してないチャネル の割り当てチャネル欄は空欄となります。割り当てされていないチャネルが存在した場合、ダイアログを閉じることができません。本文が 複数行記述された場合、設定した収録チャネルが直前割り当てられたチャネル番号と同じ場合、直前の割り当てチャネル番号を維持 し、収録ファイルに存在してないチャネルで新たに設定された収録チャネルは空欄となります。

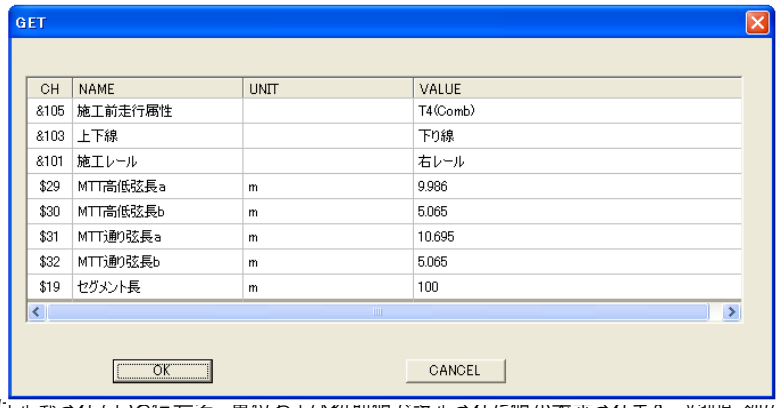
2. 7. 4. 諸元表ファイル参照文（パラメータ選択表示関連文）

機能:	予め作成されたテキスト形式の諸元表ファイルから、取得したい項目列および行を選択して諸元値を取得します。
文法:	get param %ファイル番号 諸元値格納先照チャネル列 諸元名単位格納先チャネル列
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【諸元値格納先チャネル】<必須><数値属性/文字属性参照チャネル> 諸元表から取得する諸元値の格納先を記述します。格納先参照チャネルは半角カンマ","で区切って記述します。ただし、取得する 諸元値が全て数値ではなく、文字列が含まれる場合は、数値属性チャネルと文字属性参照チャネルの双方を記述するか、文字属 性参照チャネルを記述します。 【諸元名単位格納先チャネル】<省略可><文字属性参照チャネル> 諸元値の信号名、単位の格納先を半角カンマで区切って記述します。記述省略された場合は、諸元名単位を取得しません。
記述例:	get param %2 $21 &1,&2 /* 数値属性で諸元値取得、同時に諸元名、単位を取得 */ get param %2 &22 &1,&2 /* 文字属性で諸元値取得、同時に諸元名、単位を取得 */ get param %2 $21 /* 数値属性で諸元値を取得、諸元名、単位が不要の場合 */ get param %2 &22 /* 文字属性で諸元値を取得、諸元名、単位が不要の場合 */
備考:	諸元表列選択ダイアログが表示され、諸元表から使用する列を選択し、ダイアログを閉じると、設定されている参照チャネルに、諸元表の行数分の項目値が代入されます。諸元値格納先チャネルに参照チャネルに数値属性と文字属性の 2 種を記述し、取得する諸元値に数値属性と文字属性が混在する場合、数値属性の項目は行順に数値属性参照チャネルの要素ごとに格納され、文字を含む諸元 値項目は、同じく文字属性項目の行順に文字属性参照チャネルの要素に格納されます。取得する項目が全て数値属性の場合、記述 された文字属性参照チャネルは使用しません。逆に、取得する項目が全て文字属性の場合、記述された数値属性参照チャネルは使 用しません。 記述した参照チャネルが文字属性参照チャネルのみの場合は、取得する項目は全て文字属性として扱われます。逆に、記述した参照 チャネルが数値属性のみで、取得する項目に数値変換できない文字列項目は無視されます。 諸元名単位格納先チャネルが設定されている場合、諸元項目毎の諸元名単位は設定されている文字属性参照チャネル要素に行順 に格納されますが、取得しなかった信号名単位の格納は行われません。 ダイアログの操作:

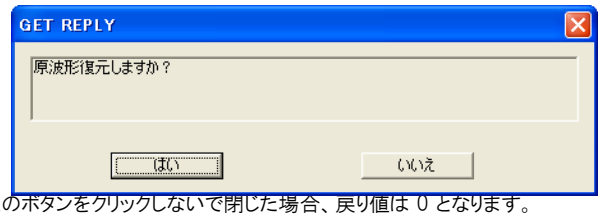


諸元表選択ダイアログの列項目ラベルを選択すると列全体が選択されます。
取得する諸元値行のチェックボックスを check します。初期値はすべての行が check されています。uncheck された行の諸元値は取得しません。このチェックボックスが uncheck された場合、使用しないことを意味し、当該行に対応した値は取得されません。 諸元表の編集機能: ダイアログに表示されている諸元表を編集することができます。編集したい項目をダブルクリックすると、入力欄に 変わります。キーボードから入力して Enter で終了すると入力確定します。また、選択された状態で、右クリックすると編集 Toolbox が表 示されます。編集 Toolbox のメニューは、Copy、Paste、InsertRow の 3 種となります。Copy-Paste する場合は、Copy 元のセルを選択 して編集ダイアログから Copy を選択します。次に Copy 先のセルを選択し、編集ダイアログから Paste を選択することで行うことができま す。また、新たに諸元列を追加したい場合は、編集 Toolbox から Insert Row を選択します。選択されると、諸元表の右側に新規列が生 成できます。
諸元表の更新格納: ダイアログで編集した結果を格納する場合、ダイアログ上の SAVE ボタンをクリックすると、上書き保存されます。 諸元表ファイルフォーマット: Script 中では、新規に諸元表ファイルを生成する機能はなく、予め EXCEL またはテキストエディタ等を使 用して作成して置く必要があります。
諸元表ファイルは拡張子".prm"のテキスト形式で項目区切りは半角カンマ","を使用します。
1 行目:メモ行です。
1 列目:"MEMO" 半角キーワードです。ダイアログのメモ欄に表示されます。
2 列目:メモ内容を任意文字列で記述します。ただし文字列に半角カンマを含んではいけません。
2 行目:列名行です。
1 列目;"NAME" 半角キーワードです。
2 列目:"UNIT" 半角キーワードです。
3 列目以降:列名を半角カンマ","区切りで任意文字列をで記述します。
2 行目以降:諸元値行です。
1 列目:信号名 任意文字列で記述します。ただし半角カンマ","を含んではいけません。
2 列目:単位名 半角英数字で記述します。ただし半角カンマ","を含んではいけません。
3 列目以降:諸元値を半角カンマ","で区切って記述します。諸元値が数値の場合、半角で記述します。

2. 7. 5. 値入力文 （パラメータ選択表示関連文）

機能:	Script 実行時に Keyboard から値を入力します。実行されると、値入力ダイアログが表示されます。
文法:	**get value** 入力値格納先チャネル列 フラグ 表題
引数:	【入力値格納先チャネル列】<必須><数値属性/文字属性参照チャネル><チャネル連続指定可> 入力値の格納先チャネル列を半角カンマ","で区切って記述します。 記述可能なチャネル数は最大 10 チャネルまでとなります。 また、ListBox の選択結果を取得する場合は、ListBox への表示内容と格納先を半角コロン":"で接続して記述します。 なお、記述する数値属性参照チャネルに表示形式を括弧内に F 形式、S 形式、E 形式の何れかで記述できます。 表示形式を記述省略した場合は E 形式(指数形式)で表示します。 【フラグ】<省略可><即値/数値属性参照チャネル> ダイアログへの表示内容を記述します。フラグ値は 0 または 1 の必要があります。フラグに 0 を記述した場合は、格納先チャネル番号、信号名、単位および値の 4 項目を表示し、フラグ 1 の場合は、信号名と値の 2 項目を表示します。なお、記述省略された場合は、フラグ 0 と同じ意味を持ちます。 【表題】<省略可><文字列即値/文字属性参照チャネル><index 指定可> ダイアログに表示する表題を意味します。複数要素を持つ参照チャネルで記述した場合、index0 と index1 の 2 行が表示されます。
記述例:	get value $1,$2,$3,&4,&5,$6 get value $[1,3],$5 /* $1,$2,$3,$5 と記述した場合と等価 */ get value $[1,3],&[4,2],$5 get value $6(F1):$7(F1),$8(F0),$5,&9 /* $6 は ListBox の内容で選択結果が$7 に戻る */ get value $6:$7[2,1],$8,$5[3,1],&9[3,10]:&10 /* 代入先データ番号を設定した場合 */
備考:	チャネルの開始データ番号、個数を設定する場合、いずれも即値で記述します。また、開始データ番号が当該チャネルのデータ番号として存在している必要があります。存在していない場合は、実行時 Error となります。また、選択的入力の参照側を除き、取得側チャネルに複数個の取得設定ができません。取得チャネル列に記載されるチャネル番号は、すでに定義されている必要があります。未定義の取得チャネルが設定された場合は実行時 Error となります。 ダイアログの操作:  ダイアログには定義されている信号名、単位および初期値で設定された値が表示されます。Value 列の変更したいセルをダブルクリックすると入力欄となります。当該セルが選択項目の場合、List Box となります。

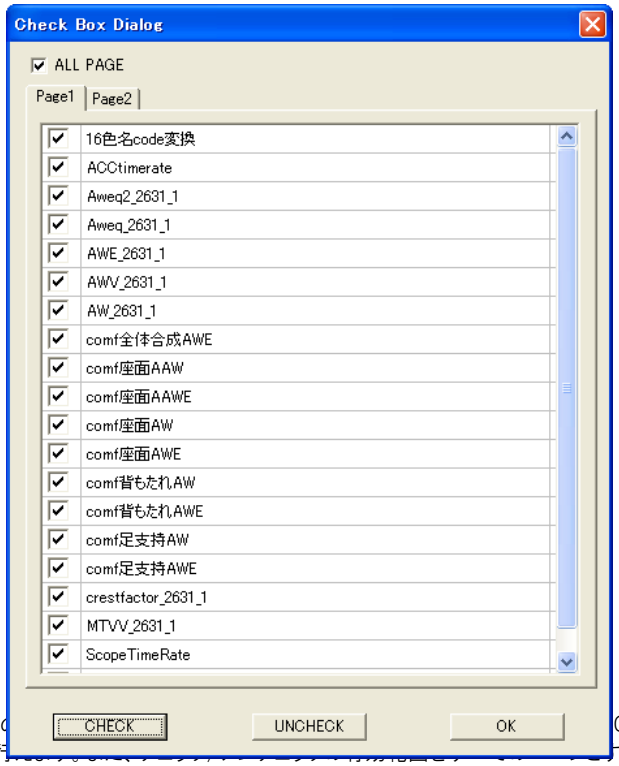
2. 7. 6. メッセージ表示応答取得文（パラメータ選択表示関連文）

機能:	ダイアログを表示させ応答を設定した参照チャネルに論理値で戻します。
文法:	**get reply** メッセージ文 ボタンラベル 1 ボタンラベル 2 応答格納先チャネル
引数:	【メッセージ文】<必須><文字列即値/文字属性参照チャネル><Index 指定可> 表示するメッセージを記述します。メッセージ文を複数行で表示する場合は、文字属性参照チャネルで記述し、Index0～2 に表示するメッセージを格納します。 【ボタンラベル 1】<必須><文字列即値/文字属性参照チャネル> 応答ボタンに表示するラベル名を記述します。ボタンラベル1はダイアログの左側ボタンラベルを意味します。 【ボタンラベル 2】<必須><文字列即値/文字属性参照チャネル> 応答ボタンに表示するラベル名を記述します。ボタンラベル1はダイアログの左側ボタンラベルを意味します。 【応答格納先チャネル】<必須><数値属性参照チャネル> 応答格納先を記述します。 左側ボタンが、クリックされた時に応答格納先チャネルに 1 が戻り、右側ボタンがクリックされた時に 0 が戻ります。
記述例:	get reply “原波形復元しますか?”“はい”“いいえ” $1 get reply &1 &2 &3 $4
備考:	ダイアログの操作:  ダイアログ上のボタンをクリックしないで閉じた場合、戻り値は 0 となります。

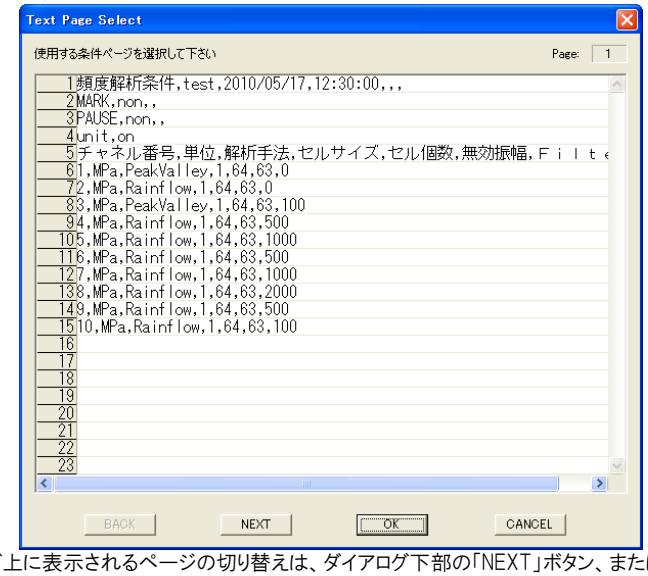
2. 7. 7. メッセージ表示文（パラメータ選択表示関連文）

機能:	メッセージを表示します。
文法:	**disp message** メッセージ文
引数:	【メッセージ文】<必須><文字列即値/文字属性参照チャネル/即値/数値属性参照チャネル><index 指定可> 表示するメッセージを記述します。記述されたメッセージ文はいったん文字属性に変換されます。メッセージ文が配列の場合、参照される Index は 0,1,2 の、3 行分が表示されます。メッセージ文は、文字列即値、文字属性参照チャネル、即値、数値属性参照チャネルを連結して記述できます。連結は、半角"|"で行います。また、数値属性参照チャネルの場合は、表示フォーマットを指定できます。フォーマット記述は、数値属性参照チャネルに続き半角括弧内()に記述します。記述可能なフォーマットは F 形式、E 形式、S 形式の 3 種類です。
記述例:	disp message “正常終了しました” disp message &1|”-“|$1(F0)
備考:	本文が実行されると、メッセージダイアログが表示されます。 

2. 7. 8. チェックボック･ステータス取得文（パラメータ選択表示関連文）

機能:	チェックボックスダイアログを表示し、チェックステータスを取得します。
文法:	get check_box_status 表示文字列 ステータス格納先チャネル 表題
引数:	【表示文字列】<必須><文字列即値/文字属性参照チャネル> チェックボックスの表示列を記述します。 【ステータス格納先チャネル】<必須><数値属性参照チャネル> チェックボックス･ステータス格納先を記述します。 表示するチェックボックスの表示初期値を設定する場合は、表示文字列に対応する要素に、checkとする場合 は 1 を、uncheckとする場合は 0 を代入して置きます。ダイアログに表示されるチェックボックスは与えられた表示文字列数に従属し表示されるチェックボックスはその 要素数分表示さ れ、戻りステータ格納先チャネルに格納される要素数は表示されるチェックボックスの数に一致 します。 【表題】<省略可><文字列即値/文字属性参照チャネル><index 指定可> ダイアログに表示する表題を記述します。複数要素を持つ参照チャネルで記述した場合、index0 と index1 の 2 行が表示されます。
記述例:	get check_box_status &2 $3 “演算項目を選択します”
備考:	チェックボックスを check した場合は参照チャネルの対応する要素は1となります。uncheck の場合は参照チャネルの対応する要素は 0 となります。また、チェックボックス表示初期値は、ステータス格納先チャネルで与えた内容にしたがいますが、要素数は一致している必要性はありません。例えば、文字列参照チャネルの要素数 < 数値属性参照チャネルの要素数の場合は、先頭 Index から文字列参照チャネル分初期値として参照されます。逆に、文字列参照チャネルの要素数 > 数値属性参照チャネルの要素数の場合は、不足分 は初期値 uncheck として使用されます。なお、いずれも戻り時点では、数値属性参照チャネルの要素数は文字列参照チャネルの要素 数に整合されます。また、初期値として参照される数値属性の値が 0 の場合は、uncheck となり、それ以外の値は、check として参照さ れます。 操作は、チェックボックスをクリックすることで行いますが、Shift＋クリックで直前に選択されたチェックボックスから現在クリック位置のチ ェックボックスまで一度に設定できます。  また、すべての CHECK」ボタン、または「UNCHECK」ボタンをクリックすることでも行 する場合は、ダイアログ上部のチェックボックス「ALL PAGE」をチェックします。「ALL PAGE」アンチェックすると、現在表示中のページが有効範囲となります。

2. 7. 9. テキスト表示選択文 （パラメータ選択表示関連文）

機能:	表示チャネル列で記述された複数個のチャネル内容をチャネルごと 1 ページに表示し、ページを選択します。
文法:	**get text_page** 表示チャネル列 選択ページ番号　表題
引数:	【表示チャネル列】<必須><文字属性参照チャネル><連続指定可> ダイアログに表示する内容を半角カンマ”,”で区切って記述します。 表示は記述したチャネルごとに1ページとなります。 【選択ページ番号】<必須><数値属性参照チャネル> 選択されたページ番号の格納先を記述します。ダイアログでチャンセルされた場合は 0 が戻ります。 【表題】<省略可><文字列即値/文字属性参照チャネル><index 指定可> ダイアログに表示する表題を記述します。複数要素を持つ参照チャネルで記述した場合、index0 と index1 の 2 行が表示されます。
記述例:	get text_page &[1,5] $1 “解析条件を選択します”
備考:	解析条件などを、複数のテキストファイルから選択する場合に、ファイルの内容をいったん、文字属性参照チャネルに読み出して本文で表示し、使用する解析条件を選択する場合などに使用します。 なお、その場合ファイル番号定義文の属性は chr（区切り文字無視）として定義した後、テキストセル読み出し文の行列番号は 1,1 として文字属性参照チャネルに格納します。 def file_id %1 “abcd.csv” chr read cell %1 1,1 0 &1  ダイアログ上に表示されるページの切り替えは、ダイアログ下部の「NEXT」ボタン、または「BACK」ボタンをクリックして切り替えます。 「OK」をクリックすると、表示されているページ番号が戻ります。

2. 7. 10. ラジオボタンステータス取得文 （パラメータ選択表示関連文）

機能:	選択したラジオボタンの結果を取得します。
文法:	get radio_button_status 表示選択項目 選択項目番号 表題
引数:	【表示選択項目】<必須><即値/文字属性参照チャネル><間接指定可> 選択させる項目名記述します。選択項目列は必ず複数で無ければ成らず、最大個数は。10 項目迄となり、11 項目以上記述されても残りは無視されます。 また、記述は半角カンマ”,”で区切って記述するか、複数要素を持つ文字列参照チャネルで記述します。あるいはその双方混在記 述します。 【選択項目番号】<必須><数値属性参照チャネル><index 指定可> 選択項目番号格納先を記述します。戻り、選択された項目番号は 0～9 までとなります。 【表題】<省略可><文字列即値/文字属性参照チャネル><Index 指定可> ダイアログに表示する表題を記述します。複数要素を持つ文字列即値で記述した場合、参照される Index は 0～2 迄となり、表題行 数は最大 3 行迄となります。
記述例:	assign &1 = ″高低左 10m 弦″,″高低右 10m 弦″,″通り左 10m 弦″,″通り右 10m 弦″,″水準″,″軌間″ $1 = 3 get radio_button_status &1 $1 “演算項目を選択して下さい”
備考:	 選択初期値は、選択項目番号で与えた index 値となります。

2. 8. 文字列操作文

2. 8. 1. 文字属性配列要素数取得文 (文字列取扱文)

機能:	文字属性参照チャネルの構成要素数を取得します。
文法:	**char num_element** 文字配列チャネル 要素数格納先チャネル
引数:	【文字配列チャネル】<必須><文字属性参照チャネル><チャネル番号間接指定可> 構成要素数を調べたい文字属性参照チャネルを記述します。参照チャネルの記述にチャネル番号間接指定が使用できます。 【要素数格納先チャネル】<必須><数値属性参照チャネル> 要素数格納先を記述します。
記述例:	char num_element &1 $2 char num_element &($3) $2
備考:	文字配列チャネルの構成要素が 0 の場合は戻り値は 0 となります。また、文字属性参照チャネルを間接指定し、複数の文字属性参照チャネルを対象とした場合、戻り数値属性参照チャネルは、複数の要素を持つ配列型となります。 数 値属性参照チャネルの構成要素数を取得する場合は、組み込み関数 LEN 関数を使用します。 ※ 演算関数⇒文字列配列要素数取得関数 ELM(&m)と同じ機能です。

2. 8. 2. 文字列長さ取得文 (文字列取扱文)

機能:	対象文字列の文字数を取得します。
文法:	**char length** 対象文字列 文字列長さ格納先チャネル
引数:	【対象文字列】<必須><文字属性参照チャネル><チャネル番号間接指定可> 文字列長さを調べたい文字列が格納された文字属性参照チャネルを記述します。 【文字列長さ格納先チャネル】<必須><数値属性参照チャネル> 取得した要素毎の 文字数格納先を記述します。戻り文字数は半角換算となります。
記述例:	char length &2 $1 char length &($2) $1
備考:	記述した文字属性参照チャネルが複数の要素を持つ配列型の場合、数値属性参照チャネルも配列型となります。文字属性参照チャネルを間接指定した場合、間接指定する数値属性チャネルが複数の要素を持つ配列型で記述しても、参照される Index は 0 のみとなります。 ※ 演算関数⇒文字列配列文字数取得関数 CLN(&m)と同じ機能です。

2. 8. 3. 文字列切り出し文 (文字列取扱文)

機能:	設定した文字列の開始位置から設定した文字数の文字列を切り出します。
文法:	**char extract** 対象文字列 開始位置 文字数 文字列格納先チャネル
引数:	【対象文字列】<必須><文字属性参照チャネル> 文字列即値、または切り出し対象文字列が格納された文字属性参照チャネルを記述します。 【開始位置】<必須><即値/数値属性参照チャネル> 文字列先頭を 1 として、切り出し開始文字位置を半角換算で即値、または数値属性参照チャネルで記述します。 開始文字位置が 0 の時は、切り出しを行わず、結果は半角スペース" "を戻します。 開始位置の符号が、正数の場合、先頭からの文字位置を意味し切り出し方向は終端方向に行い、負数の場合、終端からの文字位置を意味し、切り出し方向は先頭方向に行います。 なお、切り出された文字並びは、いずれも先頭方向からとなります。 【文字数】<必須><即値/数値属性参照チャネル> 切り出す文字数を半角換算で 即値または数値属性参照チャネルで記述します。 文字数が 0 の時は、開始文字位置から切り出し方向に、開始位置が正数の場合は終端文字まで、負数の場合は先頭文字まで切り出します。 【文字列格納先チャネル】<必須><文字属性参照チャネル> 切り出 した文字列格納先を文字属性参照チャネルで記述します。 開始文字位置、または切り取り文字数を参照チャネルで記述し、その参照チャネルが複数の要素を持つ配列型で記述された場合、受け取り参照チャネルは配列型となります。また、開始位置と切り出し文字数の要素数が異なる場合は、不足している要素は循環して参照されます。
記述例:	char extract &1 2 3 &2 /*&1 の 2 文字目から 3 文字切り出し &2 に格納 */
備考:	切り出し文字数に切り出し対象文字列が不足した場合は、対象文字列の終端までが取り出されます。また、開始文字位置が対象文字列の文字数より大きい場合は、文字列格納先チャネルは null となります。 ※ 演算関数⇒文字列切り出し関数 CEXT(k,n,&m と同じ機能です。

2. 8. 4. 文字列検索文 (文字列取扱文)

機能:	対象文字列に検査文字列が含まれているか否か検査し、含まれている場合は、その文字位置を戻します。
文法:	char find 対象文字列 検査文字列 検査開始位置　文字位置格納先チャネル
引数:	【対象文字列】<必須><文字列即値/文字属性参照チャネル> 検査対象文字列を文字列即値、または格納された文字属性参照チャネルで記述します。 【検査文字列】<必須><文字列即値/文字属性参照チャネル> 検査する文字列を文字列 即値、または格納された文字属性参照チャネルで記述します。 【開始文字位置】<必須><数値属性参照チャネル> 検査開始する対象文字列の文字位置を 即値、または数値属性参照チャネルで記述します。 文字位置は半角換算で指定します。 開始位置の符号が、正数の場合、先頭からの文字位置を意味し、負数の場合、終端からの文字位置を意味します。検索方向はい ずれも終端方向に行います。 【文字位置格納先チャネル】<必須><数値属性参照チャネル> 対象文字列の開始文字位置以降で検査文字列に一致した場合の文字位置格納先を記述します。戻り文字位置は半角換算となり ます。なお、格納される文字位置は、開始文字位置の符号に拘わりなく、先頭からの文字位置となります。 対象文字列が複数の要素を持つ場合、求めた結果の文字位置格納先チャネルも配列型となります。検査文字列を文字属性参照 チャネルで記述し、要素が複数の場合、対象文字列と要素ごとに検査されますが、検査文字列の要素数が検査対象文字列要素数 に不足している場合は、循環して参照されます。
記述例:	char find &1 “CAL_FILE:” $2
備考:	検査文字列が複数回含まれる場合、検査対象文字列の先頭から最初に一致した開始文字位置を戻します。一致しない場合は 0 が戻ります。 ※　演算関数⇒文字列検索関数 CFND(k,&n,&m)と同じ機能です。

2. 8. 5. 文字列配列の再構成文 (文字列取扱文)

機能:	配列型文字属性参照チャネルを再構成します。記述するフラグが 0 の時、再構成論理で記述された要素が 1 の要素を切り出し、フラグが 1 の時、再構成論理で記述された要素の値を index として並び換えます。
文法:	char recomposition 文字列配列　再構成論理 フラグ　結果格納先
引数:	【文字列配列】<必須><文字属性参照チャネル> 対象文字列配列を記述します。 【再構成論理】<必須><数値属性参照チャネル> フラグ記述が 0 または省略された時、再構成論理は 1 と 0 からなる数値属性参照チャネルで記述します。ただし、すべて 0 から成る数値属性参照チャネルを記述した場合は、実行時 errorとなります。 フラグ記述が 1 の時、再構成論理は並び換える index が格納された数値属性参照チャネルで記述します。 【フラグ】<省略可><即値/数値属性参照チャネル> フラグを記述します。フラグは 0 または 1 で、フラグ⇒0: 切り出し操作を行い、フラグ⇒1:並び変え操作を意味します。記述省略された場合は、フラグ⇒0 と等価となります。 【結果格納先】<必須><文字属性参照チャネル> 再構成した結果の格納先を記述します。
記述例:	char recomposition &100 $1 0 &101
備考:	<フラグ=0 の時> 再構成論理の要素値が 0 の場合、文字列配列の当該要素を削除して詰めます。記述した文字属性参照チャネルと数値属性参照チャネルの要素数が異なる場合で、数値属性参照チャネルの要素数が少ない場合は、数値属性参照チャネルの要素は循環して参照されます。また、数値属性参照チャネルの要素数が多い場合は、文字属性参照チャネルの要素数分のみ参照されます。 <フラグ=1 の時> 再構成論理の値をデータ番号と見なし、並び換え操作を行います。なお、再構成論理の要素数は、記述された文字列配列の要素数と等しくなければなりません。また、再構成論理に格納されているデータ番号は、唯一無二性である必要があります。 ※　演算関数⇒文字列配列再構成関数 CREC(k,n,&m)と同じ機能です。

2. 8. 6. 文字列削除文 (文字列取扱文)

機能:	文字属性参照チャネルの指定した文字位置から指定した文字数を削除します。
文法:	**char delete** 対象文字列 開始文字位置 文字数　結果格納先
引数:	【対象文字列】<必須><文字属性参照チャネル><index 指定可><間接指定可> 対象文字列を記述します。 【開始文字位置】<必須><即値/数値属性参照チャネル><index 指定可> 文字列削除 する開始文字位置を即値、または数値属性参照チャネルで記述します。 正数で設定した場合は先頭からの文字位置を示し、負数で設定した場合は終端からの文字位置を示します。開始文字位置が文字列長さを越えている場合、削除されません。複数要素を持つ参照チャネルで記述した場合は、対象文字列の要素数と開始文字位置の要素数が等しい必要があります。等しくない場合は、index0 のみ参照されます。 【文字数】<必須><即値/数値属性参照チャネル><index 指定可> 削除する文字数を記 述します。 開始文字位置＋文字数が文字列長さを越えた場合は、開始文字位置から以降全て削除されます。 複数要素を持つ参照チャネルで記述した場合は、対象文字列の要素数と文字数の要素数が等しい必要があります。等しくない場合 は index0 のみ参照されます。 【結果格納先】<必須可><文字属性参照チャネル><Index 指定可><間接指定可> 削除結果の文字 列の格納先を記述します。
記述例:	char dlete &1 3 10 &1" /*&1 の 3 文字目から 10 文字削除した結果を&1 に戻します*/
備考:	※ 演算関数⇒文字列削除関数 CDEL(k,n,&m)と同じ機能です。

2. 8. 7. 文字列挿入文 (文字列取扱文)

機能:	文字属性参照チャネルの指定した指定した文字列を挿入します。
文法:	**char insert** 対象文字列 挿入文字列 開始文字位置　結果格納先
引数:	【対象文字列】<必須><文字属性参照チャネル><index 指定可><間接指定可> 対象文字列を記述します。 【挿入文字列】<必須><文字列即値/文字属性参照チャネル><index 指定可> 挿入する文字列を記述します。複数要素を持つ配列で記述した場合、対象文字列の要素数と等しい必要があります。要素数が等し くない場合は、index0 のみ参照されます。 【開始文字位置】<必須><即値/数値属性参照チャネル><index 指定可> 文字列挿入開始文字位置を半角勘定で記述します。正数で設定した場合は先頭からの文字位置を示し、負数で設定した場合は終 端からの文字位置を示します。開始文字位置が文字列長さを越えている場合、挿入されません。複数要素を持つ参照チャネルで 記述した場合は、対象文字列の要素数と開始文字位置の要素数が等しい必要があります。等しくない場合は index0 のみ参照され ます。 【結果格納先】<必須可><文字属性参照チャネル><Index 指定可><間接指定可> 結果文字列格納 先を記述します。
記述例:	char insert &1 "abc" 3 &1" /*&1 の 3 文字目から abc を挿入した結果を&1 に戻します*/
備考:	※ 演算関数⇒文字列挿入関数 CINS(k,&n,&m)と同じ機能です。

2.8.8. 文字列置換文 (文字列取扱文)

機能:	文字属性参照チャネルと検査文字列が一致した時に記述した文字列に置換します。
文法:	char replace 対象文字列 置換対象文字列 検査開始文字位置 置換文字列　結果格納先
引数:	【対象文字列】<必須><文字属性参照チャネル><index 指定可><間接指定可> 文字列を置換する対象文字列を記述します。 【置換対象文字列】<必須><文字列即値/文字属性参照チャネル><index 指定可><間接指定可> 対象文字列に含まれる置換対象文字列文字列を記述します。複数要素を持つ参照チャネルで記述した場合は、対象文字列の要 素数と等しい必要があります。等しくない場合は index0 のみ参照されます。 【検査開始文字位置】<必須><即値/数値属性参照チャネル><index 指定可> 置換対象文字列の検索開始文字位置を記述します。正数で設定した場合は先頭からの文字位置を示し、負数で設定した場合は終端からの文字位置を示します。開始文字位置が文字列長さを越えている場合、置換されません。複数要素を持つ参照チャネルで記述した場合は、対象文字列の要素数と開始文字位置の要素数が等しい必要があります。等しくない場合は index0 のみ参照されます。 【置換文字列】<必須><文字列即値/文字属性参照チャネル><index 指定可><間接指定可> 検査文字列に一致した場合に置換する文字列を記述します。複数要素を持つ参照チャネルで記述した場合は、対象文字列の要素 数と等しい必要があります。等しくない場合は index0 のみ参照されます。 【結果格納先】<必須可><文字属性参照チャネル><Index 指定可><間接指定可> 置換結果の、文字列の格納先を文字属性参照チャネルで記述します。
記述例:	char replace &1 "abc" "def" 3 &1
備考:	※ 演算関数⇒文字列置換関数 CREP(k,&s,&n,&m)と同じ機能です。

2. 9. その他の構文

2. 9. 1. 結果シートの初期化文（結果シート取扱文）

機能:	結果シートの指定したページに書き込まれている内容を削除します。
文法:	**clr sheet** ページ番号:
引数:	【ページ番号】<必須><即値:> 初期化するページ番号を即値で記述し、末尾に半角コロン":"を付加します。 ページ番号 0 は、シート全体を初期化する場合に記述します。
記述例:	clr sheet 1: clr sheet 0:
備考:	本文は、結果書き込み内容を初期化するもので、宣言した構造を初期化することはできません。

2. 9. 2. SUB 実行文（演算処理記述関連文）

機能:	ファイル番号定義文で定義した dll ファイルの関数を実行します。
文法:	**call sub** %ファイル番号　関数名 引数チャネル列
引数:	【ファイル番号】<必須><%即値> ファイル番号を記述します。記述則は先頭文字"%"に続いて即値で記述します。なお、記述するファイル番号は def file_id 文定義済みである必要があります。 【関数名】<必須><文字列> 指定した dll ファイルから実行する関数名を記述します。 【引数チャネル列】<必須><数値属性属性参照チャネル><チャネル連続指定可> 実行する関数に受け渡す引数を半角カンマ","で区切って参照チャネルで記述します。参照チャネルの属性は、実行する関数により 文字属性、または数値属性で記述します。
記述例:	call sub %2　LightRectoMTT $1,$2,$3,$4,$5
備考:	実行可能な関数の引数が数値属性の場合、型式は Pointer か Double でなくてはなりません。 def file_id 文で定義した dll ファイルがカレントフォルダに存在しない場合は無視され実行されません。

2. 9. 3. EXCEL 実行文（演算処理記述関連文）

機能:	EXCEL を起動し設定したマクロファイルを実行します。
文法:	**call excel** "実行マクロファイル名"
引数:	【実行マクロ名】<必須><文字列即値> 実行マクロファイル名は半角ダブルコーテーション""で囲んだ文字列即値で記述します。拡張子は記述しません。
記述例:	call excel "報告書形式"
備考:	EXCEL が起動され、EXCEL を閉じると、本文の次の行から実行を再開します。なお、Script から直接データを渡すことはできませんので、マクロで読み出すデータファイルは予め格納して置く必要があります。

2. 9. 4. 実行時エラーフラグ初期化文（その他文）

機能:	dcl $err で演算 Error 検出を宣言した時に有効で、$err の内容を 0 に初期化します。
文法:	**clr $err**
引数:	なし
記述例:	clr $err
備考:	演算 Error 検出を宣言していない場合、本文は無視されます。

2. 9. 5. 実行終了文（その他文）

機能:	Script の実行を終了します
文法:	**end**
引数:	なし
記述例:	end
備考:	通常 call proc 文で参照される演算処理ブロックは、本文以降に記述します。end 以降に Script 文が存在していない場合、記述は不要です。また、演算処理ブロック内に記述した場合は、当該演算処理ブロックを終了 します。

2. 9. 6. 演算処理ブロック呼び出し文（演算処理記述関連文）

機能:	演算処理ブロックを Subroutine として呼び出し実行します。
文法:	**call proc** 演算処理ブロック名　引数チャネル列
引数:	【演算処理ブロック名】<必須><文字列> 　呼び出し先演算処理ブロック名を記述します。 【引数チャネル列】<省略可><即値/収録数値属性/文字列即値/文字属性参照チャネル> 呼び出し先演算処理ブロックの引数を半角カンマ","で区切って記述します。なお、引数を記述した場合、呼び出し先外部演算処理 ブロックに引き継ぎチャネル定義文(def inherit_ch)が存在し、その型式/並びが一致している必要があります。また、即値あるいは文 字列即値を引数として受け取る場合は、引き継ぎチャネル定義文(def Inherit_ch)の他に、ローカルチャネル定義文(deflocal_ch)で の定義も必要となります。なお、引数を Index 指定して受け渡す場合は、外部演算処理ブロックから戻るパラメータとしては記述でき ません。
記述例:	call proc 演算処理　　/* 受け渡すチャネルが存在しない場合 */ call proc 演算処理 $1,$2,$3,$4,#1,#4,&5,&6 call proc 演算処理 $[1,4],#1,#4,&[5,2] call proc 演算処理 $[$10,$11],#1,#4,&[$12,2]
備考:	呼び出される演算処理ブロックは end 文以降に記述します。呼び出される演算処理ブロック内に記述されているチャネルは、def local_ch 文で定義しない限り Script 文全体で同じチャネルとして扱われます。引数として記述したチャネル列は、呼び出す演算処理ブロックの開始文直下に記述された def inherit_ch 文により定義されたチャネルに置き換えて扱われます。

2. 9. 7. 通知ダイアログ書き込み文（パラメータ選択表示関連文）

機能:	通知ダイアログにチャネルの値を表示します。
文法:	**disp value** 表示チャネル列 表示フラグ
引数:	【表示チャネル列】<必須><収録/数値属性/文字属性チャネル><index 指定可><間接指定可> 　通知ダイアログに通知する内容を記述します。複数要素を持つチャネルを記述しても表示される要素は 0 のみとなります。 【表示フラグ】<省略可><即値/数値属性参照チャネル> 表示フラグ 　は、即値、または数値属性参照チャネルで記述します。 　表示フラグ 0:チャネル番号および Script 上の行番号を値と同時に表示します。 　表示フラグ 1:値のみ表示します。 　なお、省略時は 0 指定と同じです。
記述例:	disp value $1,$2(3),&3 0
備考:	

2. 9. 8. ファイル削除文（ファイル取扱共通文）

機能:	ファイル番号で指定されたファイルを削除します。
文法:	**delete file %ファイル番号**
引数:	【ファイル番号】<必須><%即値> 　ファイル番号は先頭文字半角"%"に続き即値で記述します。 記述するファイル番号は、 　先行して def file_id 文により定義が必要です。
記述例:	delete file %1
備考:	削除後、同じファイル番号でファイルを定義すると、当該ファイル番号で定義される内容で同じ属性の場合、定義内容を引き継ぎます。

索引

株式会社 デイシー
〒198-0024 東京都青梅市新町 9-2190
電話: 0428-34-9860
メール: info@deicy.co.jp