

iSeries



# VisualAge RPG パーツ解説書

*Windows* 版 バージョン 4



iSeries



# VisualAge RPG パーツ解説書

*Windows 版* バージョン 4

本書は、IBM WebSphere Development Studio Client for iSeries のバージョン 4、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

本書は、SD88-5040-03 の改訂版です。

本文および図表の変更や追加は、その変更や追加の箇所の左側に縦線で示してあります。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： SC09-2450-05  
iSeries™  
VisualAge® RPG Parts Reference  
Version 4 for Windows®

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2004.4

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、  
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1994, 2002. All rights reserved.

© Copyright IBM Japan 2004

# 目次

|                      |    |
|----------------------|----|
| 本書について               | ix |
| 本書で使用している強調表示の規則     | ix |
| 前提条件および関連情報          | ix |
| VisualAge RPG ライブラリー | ix |
| オンライン情報へのアクセス        | x  |
| オンライン・ブックの使用法        | x  |
| PDF 形式の資料            | x  |
| オンライン・ヘルプの使用法        | xi |

## このリリースの新機能 . . . . . xiii

## 第 1 章 パーツ . . . . . 1

|                    |    |
|--------------------|----|
| ActiveX            | 1  |
| アニメーション制御          | 3  |
| カレンダー              | 4  |
| キャンバス              | 5  |
| チェック・ボックス          | 7  |
| 組み合わせボックス          | 8  |
| *Component         | 10 |
| コンポーネント参照          | 11 |
| コンテナ               | 12 |
| DDE クライアント         | 14 |
| 入力フィールド            | 15 |
| グラフ                | 17 |
| グラフィック・プッシュボタン     | 18 |
| グループ・ボックス          | 20 |
| 水平方向のスクロール・バー      | 21 |
| イメージ               | 22 |
| Java Bean          | 24 |
| リスト・ボックス           | 25 |
| メディア               | 27 |
| メディア・パネル           | 28 |
| メニュー・バー            | 29 |
| メニュー項目             | 30 |
| メッセージ・サブファイル       | 31 |
| 複数行編集              | 32 |
| ノートブック             | 33 |
| ノートブック・ページ         | 34 |
| キャンバス付きノートブック・ページ  | 35 |
| ODBC/JDBC インターフェース | 36 |
| 外枠                 | 38 |
| ポップアップ・メニュー        | 39 |
| 進行状況バー             | 40 |
| プッシュボタン            | 41 |
| ラジオ・ボタン            | 42 |
| スライダー              | 43 |
| スピン・ボタン            | 45 |
| 静的テキスト             | 46 |
| 状況バー               | 47 |
| サブファイル             | 48 |
| サブメニュー             | 50 |

|              |    |
|--------------|----|
| タイマー         | 51 |
| 縦方向のスクロール・バー | 52 |
| ウィンドウ        | 53 |
| キャンバス付きウィンドウ | 55 |

## 第 2 章 パーツ属性 . . . . . 57

|            |     |
|------------|-----|
| Activate   | 58  |
| Active     | 59  |
| AddEvent   | 60  |
| AddItemEnd | 61  |
| AddLineEnd | 62  |
| AddLink    | 63  |
| AddMsgID   | 65  |
| AddMsgText | 66  |
| AddOffset  | 67  |
| AddRcd     | 68  |
| AddSrcEvt  | 70  |
| Alarm      | 71  |
| Alignment  | 72  |
| AllowChg   | 74  |
| AllowEdit  | 75  |
| AllowLink  | 77  |
| AppData    | 79  |
| AppName    | 80  |
| Arrange    | 81  |
| AttrValue  | 82  |
| AudioMode  | 83  |
| AutoInc    | 84  |
| AutoScroll | 85  |
| AutoSelect | 87  |
| BackColor  | 88  |
| BackMix    | 90  |
| BarLabel   | 91  |
| Bass       | 92  |
| BindPart   | 93  |
| BlankChar  | 94  |
| Border     | 95  |
| Bottom     | 97  |
| BufferDec  | 99  |
| BufferLen  | 100 |
| BufferPtr  | 101 |
| BufferType | 102 |
| Button     | 105 |
| ButtonIdx  | 106 |
| Buttons    | 107 |
| ButtonTip  | 108 |
| CanUndo    | 109 |
| CapsLock   | 110 |
| Case       | 111 |
| CellBGClr  | 112 |
| CellBGMix  | 114 |

|                      |     |                      |     |
|----------------------|-----|----------------------|-----|
| CellFGClr . . . . .  | 115 | DayNumRect . . . . . | 177 |
| CellFGMix . . . . .  | 117 | DayStart . . . . .   | 178 |
| Center . . . . .     | 118 | DColFRVCol . . . . . | 179 |
| CharData . . . . .   | 119 | DDEAddLink . . . . . | 180 |
| CharOffset . . . . . | 120 | DDEMode . . . . .    | 182 |
| Checked . . . . .    | 121 | DDERmvLink . . . . . | 183 |
| ChildCount . . . . . | 122 | DeActivate . . . . . | 184 |
| ChildList . . . . .  | 123 | Delete . . . . .     | 185 |
| ClearAll . . . . .   | 124 | DeleteRcd . . . . .  | 186 |
| ClearDate . . . . .  | 125 | DeleteRow . . . . .  | 187 |
| ClearMonth . . . . . | 126 | DelimChar . . . . .  | 188 |
| ClearYear . . . . .  | 127 | DeSelect . . . . .   | 189 |
| Clipboard . . . . .  | 128 | Dialog . . . . .     | 190 |
| ColBGClr . . . . .   | 129 | DIRName . . . . .    | 191 |
| ColBGMix . . . . .   | 131 | DlgOwner . . . . .   | 192 |
| ColFGClr . . . . .   | 132 | DoEvents . . . . .   | 193 |
| ColFGMix . . . . .   | 134 | DragEnable . . . . . | 194 |
| Collapsed . . . . .  | 135 | DropEnable . . . . . | 195 |
| BatchAdd . . . . .   | 136 | DropValue . . . . .  | 196 |
| NbrOfRec . . . . .   | 138 | DspHeight . . . . .  | 197 |
| EndAdding . . . . .  | 139 | DspWidth . . . . .   | 198 |
| ColNumber . . . . .  | 140 | EditColumn . . . . . | 199 |
| Color . . . . .      | 141 | EditIndex . . . . .  | 200 |
| ColorArea . . . . .  | 143 | EditItem . . . . .   | 201 |
| ColorMix . . . . .   | 145 | EditText . . . . .   | 202 |
| Column . . . . .     | 146 | EnableBtn . . . . .  | 203 |
| ColumnDec . . . . .  | 147 | Enabled . . . . .    | 204 |
| ColumnLen . . . . .  | 148 | Execute . . . . .    | 206 |
| ColumnName . . . . . | 149 | ExecutesQL . . . . . | 207 |
| Columns . . . . .    | 150 | ExtSelect . . . . .  | 208 |
| ColumnType . . . . . | 151 | Fetch . . . . .      | 209 |
| ColWidth . . . . .   | 152 | FetchNext . . . . .  | 210 |
| CompName . . . . .   | 153 | FetchPrior . . . . . | 211 |
| Connect . . . . .    | 154 | FieldExit . . . . .  | 212 |
| Connected . . . . .  | 155 | FileName . . . . .   | 213 |
| ConnectStr . . . . . | 156 | FillStyle . . . . .  | 215 |
| Copy . . . . .       | 157 | FirstSel . . . . .   | 217 |
| Count . . . . .      | 158 | Focus . . . . .      | 218 |
| CsrAtEnd . . . . .   | 159 | FontArea . . . . .   | 220 |
| CsrLine . . . . .    | 160 | FontBold . . . . .   | 222 |
| CsrPos . . . . .     | 161 | FontItalic . . . . . | 224 |
| CurrentDir . . . . . | 162 | FontName . . . . .   | 226 |
| CurrentRow . . . . . | 163 | FontSize . . . . .   | 227 |
| Cut . . . . .        | 164 | FontStrike . . . . . | 229 |
| DataGroup . . . . .  | 165 | FontUnder . . . . .  | 231 |
| DataPoint . . . . .  | 166 | ForeColor . . . . .  | 233 |
| DataType . . . . .   | 167 | ForeMix . . . . .    | 235 |
| DataValue . . . . .  | 168 | Format . . . . .     | 237 |
| Date . . . . .       | 169 | FrameRate . . . . .  | 238 |
| DateIdx . . . . .    | 170 | FrmtString . . . . . | 239 |
| DateText . . . . .   | 171 | GetItem . . . . .    | 240 |
| DateUnder . . . . .  | 172 | GetNewID . . . . .   | 241 |
| Day . . . . .        | 173 | GetRcdFld . . . . .  | 242 |
| DayIdx . . . . .     | 174 | GetRcdIcon . . . . . | 243 |
| DayLen . . . . .     | 175 | GetRcdText . . . . . | 244 |
| DayNumPos . . . . .  | 176 | GetTables . . . . .  | 245 |

|                        |     |                      |     |
|------------------------|-----|----------------------|-----|
| GetTables の例 . . . . . | 246 | Month . . . . .      | 312 |
| GnEqGrpCol . . . . .   | 247 | MonthArrow . . . . . | 313 |
| GnEqPntCol . . . . .   | 248 | MonthIdx . . . . .   | 314 |
| GraphType . . . . .    | 249 | MonthLen . . . . .   | 315 |
| GrphHiLite . . . . .   | 250 | MouseIcon . . . . .  | 316 |
| GroupLabel . . . . .   | 251 | MouseShape . . . . . | 317 |
| Handle . . . . .       | 252 | MsgData . . . . .    | 319 |
| HasPrpPage . . . . .   | 253 | MsgFile . . . . .    | 320 |
| HdgBGClr. . . . .      | 254 | MsgID . . . . .      | 321 |
| HdgBGMix . . . . .     | 256 | MsgSubText . . . . . | 322 |
| HdgFGClr. . . . .      | 257 | MsgText . . . . .    | 323 |
| HdgFGMix . . . . .     | 259 | Multiplier . . . . . | 324 |
| HdgIdx. . . . .        | 260 | MultSelect. . . . .  | 325 |
| HdgText . . . . .      | 261 | Name . . . . .       | 326 |
| Height . . . . .       | 262 | NbrOfImage . . . . . | 327 |
| HelpEnable . . . . .   | 264 | NbrOfLines . . . . . | 328 |
| HelpWindow . . . . .   | 266 | NbrOfSel . . . . .   | 329 |
| Hidden . . . . .       | 267 | NextLine . . . . .   | 330 |
| HighLight . . . . .    | 268 | NextPage . . . . .   | 331 |
| HitItem . . . . .      | 270 | NotSrcEvt . . . . .  | 332 |
| HlitPoints . . . . .   | 272 | NotSrcPart . . . . . | 333 |
| HostName . . . . .     | 273 | NotSrcWin . . . . .  | 334 |
| HRule . . . . .        | 274 | OCXProp . . . . .    | 335 |
| IconHandle . . . . .   | 275 | OCXPropIdx . . . . . | 336 |
| Index . . . . .        | 276 | Parameter . . . . .  | 337 |
| InPlace. . . . .       | 277 | ParamIndex . . . . . | 338 |
| InsertItem . . . . .   | 278 | OCXValue . . . . .   | 339 |
| InsertLine . . . . .   | 279 | OnTop . . . . .      | 340 |
| InsertMode . . . . .   | 280 | OpenEdit . . . . .   | 341 |
| InsertRow . . . . .    | 281 | FreezeCol . . . . .  | 342 |
| InsertText . . . . .   | 282 | OS . . . . .         | 343 |
| Interval . . . . .     | 283 | OutlineRcl. . . . .  | 344 |
| InUse . . . . .        | 284 | PageNumber . . . . . | 345 |
| InvName . . . . .      | 285 | PageSize . . . . .   | 346 |
| InvPName. . . . .      | 286 | Panel . . . . .      | 347 |
| IsData . . . . .       | 287 | PanelItem . . . . .  | 349 |
| Item . . . . .         | 288 | PanelMode . . . . .  | 351 |
| ItemCount . . . . .    | 290 | Parent . . . . .     | 352 |
| ItemKey . . . . .      | 291 | ParentId . . . . .   | 354 |
| Label . . . . .        | 292 | ParentList . . . . . | 355 |
| LabelPlace . . . . .   | 294 | ParentName . . . . . | 356 |
| Left. . . . .          | 295 | PartCount . . . . .  | 357 |
| LegendType . . . . .   | 297 | PartList . . . . .   | 358 |
| Length . . . . .       | 298 | PartName . . . . .   | 359 |
| LineNumber . . . . .   | 299 | PartType . . . . .   | 360 |
| LineText . . . . .     | 300 | Paste . . . . .      | 361 |
| LookNFeel . . . . .    | 301 | PBRange . . . . .    | 362 |
| Magnify . . . . .      | 302 | PBSetPos . . . . .   | 363 |
| MapViewCol . . . . .   | 303 | PBStep. . . . .      | 364 |
| Masked . . . . .       | 304 | PBStepSize . . . . . | 365 |
| Maximum . . . . .      | 305 | Platform . . . . .   | 366 |
| Method . . . . .       | 306 | PlugCmd . . . . .    | 367 |
| Method の例 . . . . .    | 308 | PlugDLL . . . . .    | 368 |
| MiniIcon . . . . .     | 309 | PlugID. . . . .      | 369 |
| Minimum . . . . .      | 310 | PlugRC . . . . .     | 370 |
| Mode . . . . .         | 311 | PlugResult. . . . .  | 371 |

|                         |     |                      |     |
|-------------------------|-----|----------------------|-----|
| Poke . . . . .          | 372 | ShowProp . . . . .   | 432 |
| Position . . . . .      | 373 | ShowRects . . . . .  | 433 |
| PrevLine . . . . .      | 374 | ShowTabs . . . . .   | 434 |
| PrevPage . . . . .      | 375 | ShowText . . . . .   | 435 |
| Print . . . . .         | 376 | ShowTips . . . . .   | 436 |
| PrintAsIs . . . . .     | 377 | SizeToFit . . . . .  | 437 |
| Printer . . . . .       | 378 | SortAsc . . . . .    | 438 |
| ProgresBar . . . . .    | 379 | SortDesc . . . . .   | 439 |
| Range . . . . .         | 380 | SQLError . . . . .   | 440 |
| ReadOnly . . . . .      | 381 | SQLMsgBox . . . . .  | 441 |
| RecordID . . . . .      | 383 | SQLQuery . . . . .   | 442 |
| RefAttr . . . . .       | 384 | StartAt . . . . .    | 443 |
| RefParent . . . . .     | 385 | StartAt の例 . . . . . | 444 |
| RefPart . . . . .       | 386 | StartNew . . . . .   | 445 |
| Refresh . . . . .       | 387 | StatusBar . . . . .  | 446 |
| RemoveItem . . . . .    | 388 | SwitchTo . . . . .   | 447 |
| RemoveLink . . . . .    | 389 | TabImage . . . . .   | 448 |
| RemoveMsg . . . . .     | 390 | TabLabel . . . . .   | 449 |
| RemoveRcd . . . . .     | 391 | Text . . . . .       | 450 |
| Request . . . . .       | 392 | TextEnd . . . . .    | 451 |
| ReturnVal . . . . .     | 393 | TextLength . . . . . | 452 |
| RmvSrcEvt . . . . .     | 394 | TextSelect . . . . . | 453 |
| RmvEvent . . . . .      | 395 | TextStart . . . . .  | 454 |
| RowBGClr . . . . .      | 396 | TextString . . . . . | 455 |
| RowBGMix . . . . .      | 398 | TickLabel . . . . .  | 456 |
| RowFGClr . . . . .      | 399 | TickNumber . . . . . | 457 |
| RowFGMix . . . . .      | 401 | Tile . . . . .       | 458 |
| Rows . . . . .          | 402 | Timeout . . . . .    | 459 |
| SBIndex . . . . .       | 403 | TimerMode . . . . .  | 460 |
| SBLLabel . . . . .      | 404 | TimerTicks . . . . . | 461 |
| SBPanels . . . . .      | 405 | TipText . . . . .    | 462 |
| SBPosition . . . . .    | 406 | Title . . . . .      | 463 |
| SBStyle . . . . .       | 407 | TitlePlace . . . . . | 464 |
| Scale . . . . .         | 408 | Top . . . . .        | 465 |
| Search . . . . .        | 409 | Topic . . . . .      | 467 |
| SearchType . . . . .    | 410 | TopLine . . . . .    | 468 |
| Selected . . . . .      | 411 | TopRecord . . . . .  | 469 |
| SelectIdx . . . . .     | 412 | Treble . . . . .     | 470 |
| SelectItem . . . . .    | 413 | UnBind . . . . .     | 471 |
| SelectList . . . . .    | 414 | UnderPoint . . . . . | 472 |
| SelectList の例 . . . . . | 415 | Undo . . . . .       | 473 |
| SelectRcd . . . . .     | 416 | UpdateRow . . . . .  | 474 |
| SelPrinter . . . . .    | 417 | UseData . . . . .    | 475 |
| Sequence . . . . .      | 418 | UseDelim . . . . .   | 476 |
| SetItem . . . . .       | 420 | UserData . . . . .   | 477 |
| SetRcdFld . . . . .     | 421 | Validate . . . . .   | 478 |
| SetRcdIcon . . . . .    | 422 | Value . . . . .      | 480 |
| SetRcdText . . . . .    | 423 | VColFRDCol . . . . . | 481 |
| SetTop . . . . .        | 424 | View . . . . .       | 482 |
| TopItem . . . . .       | 425 | ViewColumn . . . . . | 483 |
| SflNxtChg . . . . .     | 426 | Visible . . . . .    | 484 |
| ShData . . . . .        | 427 | VisTitle . . . . .   | 487 |
| ShDataLen . . . . .     | 428 | VisTitlSep . . . . . | 488 |
| ShDataName . . . . .    | 429 | Volume . . . . .     | 489 |
| ShDataPos . . . . .     | 430 | VRule . . . . .      | 490 |
| ShowMsgID . . . . .     | 431 | WeekDay . . . . .    | 491 |



|                      |     |
|----------------------|-----|
| WeekDayIdx . . . . . | 492 |
| Width . . . . .      | 493 |
| WindowMode . . . . . | 495 |
| WordWrap . . . . .   | 496 |
| WrkStnName . . . . . | 497 |
| X . . . . .          | 498 |
| XAxisLabel . . . . . | 499 |
| Y . . . . .          | 500 |
| YAxisLabel . . . . . | 501 |
| Year . . . . .       | 502 |
| YearIdx . . . . .    | 503 |
| YearLen . . . . .    | 504 |
| Yinc . . . . .       | 505 |

**第 3 章 パーツ・イベント . . . . . 507**

|                      |     |
|----------------------|-----|
| Activate . . . . .   | 507 |
| BeanEvent. . . . .   | 507 |
| Change. . . . .      | 508 |
| Click . . . . .      | 508 |
| Close . . . . .      | 509 |
| Collapsed . . . . .  | 509 |
| ColSelect . . . . .  | 509 |
| OpenEdit . . . . .   | 510 |
| Complete . . . . .   | 510 |
| Create . . . . .     | 510 |
| Data . . . . .       | 511 |
| DbClick . . . . .    | 511 |
| DeActivate . . . . . | 512 |
| Destroy . . . . .    | 512 |
| Drop . . . . .       | 513 |
| DropDown . . . . .   | 513 |
| Enter . . . . .      | 513 |
| ExecuteAck . . . . . | 514 |
| Expanded . . . . .   | 514 |
| FirstRec . . . . .   | 515 |
| GotFocus . . . . .   | 515 |
| KeyPress . . . . .   | 515 |
| LastRec . . . . .    | 516 |
| LClickTray . . . . . | 516 |
| Link . . . . .       | 516 |
| LostFocus . . . . .  | 517 |
| MenuSelect . . . . . | 517 |
| MouseDown . . . . .  | 517 |
| MouseEnter . . . . . | 518 |
| MouseExit. . . . .   | 518 |
| MouseMove . . . . .  | 518 |
| MouseUp . . . . .    | 519 |
| Moved . . . . .      | 519 |
| MthChange . . . . .  | 520 |
| NextRec . . . . .    | 520 |
| Notify . . . . .     | 520 |
| OCXEvent . . . . .   | 521 |
| PageDown. . . . .    | 521 |

|                      |     |
|----------------------|-----|
| PageEnd . . . . .    | 521 |
| PageSelect. . . . .  | 522 |
| PageTop . . . . .    | 522 |
| PageUp . . . . .     | 522 |
| PokeAck . . . . .    | 523 |
| Popup . . . . .      | 523 |
| Press . . . . .      | 524 |
| PrevRec . . . . .    | 524 |
| RClickTray . . . . . | 524 |
| ReSize . . . . .     | 524 |
| Scroll . . . . .     | 525 |
| Select . . . . .     | 525 |
| SelPending . . . . . | 526 |
| ShutDown. . . . .    | 526 |
| SpinDown. . . . .    | 526 |
| SpinEnd . . . . .    | 526 |
| SpinUp. . . . .      | 527 |
| Terminate . . . . .  | 527 |
| Tick . . . . .       | 527 |
| Timeout . . . . .    | 528 |
| VKeyPress . . . . .  | 528 |
| YearChange . . . . . | 529 |

**第 4 章 イベント属性 . . . . . 531**

|                        |     |
|------------------------|-----|
| %Alt . . . . .         | 532 |
| %Button . . . . .      | 532 |
| %Character . . . . .   | 532 |
| %ColNumber . . . . .   | 533 |
| %Complete . . . . .    | 533 |
| %Control . . . . .     | 533 |
| %Data . . . . .        | 534 |
| %EventName . . . . .   | 534 |
| %Index . . . . .       | 534 |
| %Item . . . . .        | 534 |
| %MouseX. . . . .       | 535 |
| %MouseY. . . . .       | 535 |
| %NewHeight . . . . .   | 535 |
| %NewMode . . . . .     | 535 |
| %NewWidth . . . . .    | 536 |
| %Page . . . . .        | 536 |
| %Part . . . . .        | 536 |
| %Position . . . . .    | 536 |
| %RealName . . . . .    | 537 |
| %Shift . . . . .       | 537 |
| %SrcCompName. . . . .  | 537 |
| %SrcEvtName . . . . .  | 537 |
| %SrcPartName . . . . . | 538 |
| %SrcWinName . . . . .  | 539 |
| %Window. . . . .       | 539 |

**用語集 . . . . . 541**

**参考文献. . . . . 555**



---

## 本書について

本書には、グラフィカル・ユーザー・インターフェース (GUI) アプリケーションの作成に使用する VisualAge® RPG パーツ、各パーツと関連した属性とイベント、および VARPG アプリケーションで照会できるイベント属性に関する参照情報が記載されています。本書は、VisualAge RPG を使用してアプリケーションのプログラミングを行う方々を対象としています。

本書を使用する前に、*WebSphere Development Studio Client for iSeries* ご使用に際してをお読みください。

---

## 本書で使用している強調表示の規則

本書の全体を通じて、属性、イベント、およびイベント属性の名前はすべて**太字体**で現れます。このことは、初出の用語、他の強調用語、および VisualAge RPG インターフェース上に現れるテキストの参照についても同様です。変数はイタリック体で現れます。VisualAge RPG の命令コードは「すべて大文字で」現れます。

---

## 前提条件および関連情報

iSeries および AS/400e の技術情報を調べるための開始点として iSeries Information Center を使用してください。Information Center をアクセスする方法には、以下の 2 つがあります。

- 次の Web サイトから:

<http://www.ibm.com/eserver/series/infocenter>

- OS/400 オーダーで配送された CD-ROM から:

*iSeries Information Center*, SK88-8055-03.

iSeries Information Center には、CL コマンド、システム・アプリケーション・プログラミング・インターフェース (API)、論理区画、クラスター、Java™、TCP/IP、Web サービス、およびセキュア・ネットワークなどのアドバイザーおよび重要なトピックが入っています。また、関連した IBM® Redbooks へのリンクおよび Technical Studio や IBM ホーム・ページなどのその他の Web サイトへのインターネット・リンクも含まれています。

---

## VisualAge RPG ライブラリー

VisualAge RPG ライブラリーには、以下の資料が含まれています。

*VisualAge for RPG プログラミング*

本書には、VisualAge RPG を使用してアプリケーションを作成する場合の具体的な情報が記載されています。これには、アプリケーション開発サイクルのすべての段階で実行する必要があるステップが、設計からパッケージおよび分配まで説明され

ています。VisualAge RPG アプリケーション開発の概念と工程を明らかにするために、プログラミング例が収録されています。

#### *ADTS/CS VisualAge for RPG* パーツ解説書

本書には、VisualAge RPG のパーツ、パーツ属性、パーツ・イベント、およびイベント属性に関する情報が記載されています。本書は、VisualAge RPG を使用してアプリケーションを開発しているすべての方々の参照資料です。

#### *VisualAge for RPG WINDOWS* 版 言語解説書

本書には、VisualAge RPG コンパイラーを使用して実装される RPG IV 言語に関する情報が記載されています。次の事項が含まれています。

- 文字セット、記号名および予約語、コンパイラー指示、ならびに標識などの言語基本エレメント
- データ・タイプおよびデータ形式
- エラーおよび例外処理
- 仕様
- 組み込み関数、式、および命令コード

製品全体の概要については、*WebSphere Development Studio Client for iSeries* ご使用に際してを参照してください。

関連資料のリストについては、本書の終わりにある参考文献を参照してください。

また以下のオンライン・ソースでも IBM WebSphere Development Studio Client for iSeries についての最新の情報を見つけることができます。

**「Development Studio Client」ホーム・ページ**  
[ibm.com/software/ad/wdsc/](http://ibm.com/software/ad/wdsc/)

---

## オンライン情報へのアクセス

VisualAge RPG には、種々のオンライン・ブックおよびオンライン・ヘルプが含まれています。プロダクトの使用中にヘルプにアクセスしたり、またプロダクトの使用上または別途に本書を表示することができます。

### オンライン・ブックの使用方法

オンライン・ブックを表示するためには、次のいずれかを実行してください。

- VisualAge RPG GUI Designer またはエディター・ウィンドウの「ヘルプ」プルダウン・メニューから本書の名前を選択してください。
- 「スタート」メニューから本書にアクセスしてください。「プログラム」→「IBM WebSphere Development Studio Client for iSeries」を選択します。次に「文書」を選択します。

### PDF 形式の資料

VisualAge RPG 資料は、URL <http://www.ibm.com/eserver/iseries/infocenter> の iSeries Information Center から PDF の形式で使用可能です。

**注:** ワークステーションで資料を PDF 形式で表示するには、Adobe Acrobat Reader (Windows 版 3.01 以降) が必要です。システム設置場所にこのプログラムがない場合には、Adobe Systems Web サイト (<http://www.adobe.com>) からコピーをダウンロードできます。

以下の VisualAge RPG 資料は、PDF 形式で使用可能です。

- *VisualAge for RPG* プログラミング
- *ADTS/CS VisualAge for RPG* パーツ解説書
- *VisualAge for RPG WINDOWS* 版 言語解説書

この製品については、*WebSphere Development Studio Client for iSeries* ご使用に際して、SD88-5054-03を参照してください。

## オンライン・ヘルプの使用方法

オンライン・ヘルプは VisualAge RPG のすべての領域から使用することができます。特定のウィンドウ、ダイアログ・ボックス、またはプロパティ・ノートブックのヘルプを表示するには、「ヘルプ」プッシュボタン (使用可能な場合) を選択します。

**注:** HTML 形式のヘルプを表示するには、ワークステーションには Netscape Navigator 4.04 以降、Microsoft® Internet Explorer 4.01 以降などのフレーム使用可能 Web ブラウザーが必要です。(推奨ブラウザは、Netscape Navigator 4.6 または Internet Explorer 5.0 です。)

### ヘルプ情報の使用方法

任意の時点でヘルプ情報を表示するためには、F1 キーを押してください。表示されるヘルプは、入力フォーカスを持つインターフェースの領域に固有のものです。入力フォーカスはメニュー項目、ウィンドウ、ダイアログ・ボックス、およびプロパティ・ノートブック上にあるか、あるいはこれらの特定の部分にあります。

ダイアログ・ボックスのヘルプ情報については、ウィンドウの右上隅の疑問符 (使用可能な場合) をクリックしてください。マウスの矢印の隣に疑問符 (?) が現れます。ワードまたはフィールドをクリックすると、その特定のフィールドについてのヘルプ情報が表示されます。

### 言語依存ヘルプの使用方法

言語依存ヘルプを表示するためには、編集ウィンドウで F1 キーを押してください。カーソルが命令コードに位置付けられている場合には、その命令コードのヘルプが表示されます。その他の場合は、現在の指定のヘルプが表示されます。



---

## このリリースの新機能

新規のパーツ属性およびイベントには以下のものがあります。

- **キャンバス・パーツ**には現在 **VKeyPress** イベントがあります。このイベントに  
応答できるすべてのパーツに 1 つのアクション・サブルーチンをコーディングする  
だけで済みます。
- **ActiveX: ReturnVal** は、メソッド呼び出しの値を戻します。**OCXPropIdx** は、  
配列プロパティの索引を設定または検索します。この配列は、ストリングまたは  
数値とすることができます。
- **コンテナ: DeleteRcd** は、1 つまたはすべてのレコードを削除するために設定  
できます。
- **\*component: HostName** は、ワークステーションのホスト名を `host_name`  
`IP_address` の形式で戻します。
- **メニュー項目: Visible** 属性が追加されました。さらにショート・キーが追加され  
ています。ショートカット・キーのテキストは、実行時にメニュー項目ラベルの  
横に自動的に表示されるようになります。

本書には、前のリリースの README からの情報および他の技術的訂正が含まれて  
います。前の README の内容には、以下の新規 VisualAge RPG パーツ属性の説  
明が含まれています。

- サブファイル・パーツ属性

**EditColumn**、**EditIndex**、および **EditText** は、列番号、列の行、または編集中  
のテキストを戻します。

**MapViewCol** および **ViewColumn** は、サブファイルのデータ列が表示される順  
序を変更するのに使用できます。

**DColFRVCol** および **VColFRDCol** は、それぞれデータ列と表示列を戻します。

**SortAsc** および **SortDesc** は、列のデータをソートするのに使用できます。

- ウィンドウ・パーツおよびイメージ・パーツ用の **PrintAsIs** は、イメージを印刷  
し、その縦横比を維持します。
- **DropValue** は、ドロップ操作の後でラベルを変更または保存するために使用され  
ます。
- **OnTop** は、指定されたノートブック・ページを現行ページに変更します。
- **グラフ・パーツ**には現在 **Popup** イベントがあります。





---

## 第 1 章 パーツ

パーツとは、グラフィカル・ユーザー・インターフェース (GUI) アプリケーションを作成する時に使用するビルド・ブロックです。これらは、ビルド時にテンプレートとして働くので、レイアウトを迅速かつ容易に作成することができます。各パーツは関連した属性およびイベントを有しています。

この章では、VisualAge® RPG パーツについて説明します。それぞれの説明には以下が含まれています。

- パーツの概説
- パーツに適用される属性のリスト
- パーツに適用されるイベントのリスト
- 適用される制約事項

---

### ActiveX



**\* 制約事項:** このパーツは Java® アプリケーションではサポートされていません。

ActiveX™ パーツを使用して、ActiveX コントロール・オブジェクトをプロジェクトに追加してください。そうすると、アプリケーションは、その属性にアクセスでき、イベントをモニターすることができます。(ActiveX 制御は、第三者ベンダーによって開発および提供されます。)

追加しようとしている ActiveX コントロールに精通する必要があります。VARPG GUI Designer は、これらのパーツによって提供される機能を制御できません。

**注:** VARPG は、C++ で書かれたインターフェースを持つ ActiveX 制御でしか機能しません。使用したい ActiveX 制御で VARPG が機能するかどうかを ActiveX 制御のプロバイダーに確認してください。

## パーツ属性

|            |            |            |
|------------|------------|------------|
| Activate   | AddEvent   | Bottom     |
| DeActivate | HasPrpPage | Height     |
| Left       | Method     | OCXProp    |
| OCXPropIdx | OCXValue   | ParamIndex |
| Parameter  | ParentName | PartName   |
| PartType   | Refresh    | ReturnVal  |
| RmvEvent   | ShowProp   | Top        |
| UserData   | Visible    | Width      |

## 適用可能なイベント

|        |         |          |
|--------|---------|----------|
| Create | Destroy | OCXEvent |
|--------|---------|----------|

## アニメーション制御



Windows® アプリケーションでは、アニメーション・コントロール・パーツが AVI 拡張子のビデオ・ファイルを再生します。このパーツは、ビデオが実際にはプログラム・ロジックとは独立に再生されるという点で、メディア・パーツとは異なります。このパーツの典型的な使用法の 1 つは、ファイルがあるフォルダーから別のフォルダーに移動中であるといった何らかの進行を示す AVI ファイルを表示することです。

アニメーション制御パーツは、音声を伴わずにビデオ・ファイルを再生します。AVI ファイルは、RLE (ラン・レングス・エンコード) 方式で圧縮されていない限り、圧縮形式にすることはできません。

Java アプリケーションでは、アニメーション・コントロール・パーツは、NbrOfImage 属性を使用して、アニメーション GIF ファイル・シーケンスを再生するために使用されます。

### パーツ属性

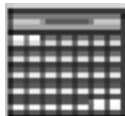
|            |           |            |
|------------|-----------|------------|
| FileName   | FrameRate | Handle*    |
| Left       | Mode      | NbrOfImage |
| ParentName | PartName  | PartType   |
| Top        | UserData  | Visible    |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

## カレンダー



カレンダー・パーツは、月間カレンダーを表します。ユーザーは、月矢印の 1 つをクリックし、次の月または直前の月に行くことによって、カレンダーをナビゲートすることができます。

また、特定の日付に行ったり、ユーザーが選択した日付を決定したり、短いテキスト・コメントをカレンダーの個々の日に追加したりするなど、プログラムによってカレンダーを完全に制御することもできます。

### パーツ属性

|            |             |            |
|------------|-------------|------------|
| Border     | Bottom      | ClearAll   |
| ClearDate  | ClearMonth  | ClearYear  |
| Color      | ColorArea   | ColorMix   |
| Date       | DateIdx     | DateText   |
| DateUnder  | Day         | DayIdx     |
| DayLen     | DayNumPos   | DayNumRect |
| DayStart   | Enabled     | FontArea   |
| FontBold   | FontItalic  | FontName   |
| FontSize   | FontStrike* | FontUnder* |
| FrmtString | Handle*     | Height     |
| HRule      | Left        | Month      |
| MonthArrow | MonthIdx    | MonthLen   |
| OutlineRcl | ParentName  | PartName   |
| PartType   | Refresh     | ShowRects  |
| ShowText   | TipText     | Top        |
| UserData   | Visible     | VRule      |
| WeekDay    | WeekDayIdx  | Width      |
| Year       | YearIdx     | YearLen    |

\* 注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|           |            |            |
|-----------|------------|------------|
| Click     | Create     | Destroy    |
| DbClick   | MouseDown  | MouseEnter |
| MouseExit | MouseMove  | MouseUp    |
| MthChange | YearChange |            |

## キャンバス



ウィンドウまたはノートブック・ページ上に複数のパーツを入れたい場合には、ウィンドウまたはノートブック・ページでキャンバス・パーツを使用してください。キャンバス上の各種のパーツを指示してクリックし、それらを配置してから編成して、グラフィカル・ユーザー・インターフェースを作成することができます。

キャンバス・パーツは、ウィンドウまたはノートブック・ページのどちらかのクライアント域を占めます。ウィンドウまたはノートブック・ページにキャンバスがない場合には、パーツがメニュー・バーやメッセージ・サブファイルなどのウィンドウの拡張機能でない限り、クライアント域には 1 つのパーツしか入れることができません。

たいていの場合、複数のパーツを持つウィンドウおよびノートブック・ページを作成します。その場合、ユーザーはキャンバス・パーツを持つノートブック・ページおよびキャンバス・パーツを持つウィンドウを使用する必要があります。これらはすでにキャンバス・パーツを含んでいるため、ユーザーのステップを省くことができます。

ビルド時に、**FileName** 属性を指定して、ビットマップ・イメージをキャンバスの背景として組み込むこともできます。このイメージは、**Tile** 属性を設定することによってタイル表示することができます。Java アプリケーションの場合には、GIF または JPG イメージを背景として組み込むことができます。

### 注:

1. キャンバス・パーツ、ウィンドウ (キャンバスなし) パーツ、およびノートブック・ページ (キャンバスなし) パーツは、パーツ・パレットではなくパーツ・カタログに入っています。頻繁に使用したい場合には、それらをパーツ・パレットに追加することができます。
2. キャンバス・パーツ上のパーツにデフォルトのフォントの設定 (デフォルトのシステム・フォント) がある場合には、それらはキャンバス・パーツに指定されたフォントの定義を継承します。キャンバス上の大多数のパーツにフォントを適用するには、個々のパーツごとのフォントではなく、キャンバス・パーツのフォントを指定してください。

関連情報については、以下を参照してください。

- 53 ページの『ウィンドウ』
- 55 ページの『キャンバス付きウィンドウ』
- 34 ページの『ノートブック・ページ』
- 35 ページの『キャンバス付きノートブック・ページ』.

## パーツ属性

|             |            |            |
|-------------|------------|------------|
| BackColor   | BackMix    | Bottom*    |
| Enabled     | FileName   | FontBold   |
| FontItalic  | FontName   | FontSize   |
| FontStrike* | FontUnder* | Handle*    |
| Height      | Left       | ParentName |
| PartName    | PartType   | ReadOnly   |
| Refresh     | Tile       | Top*       |
| UserData    | Width      |            |

\*注: 制約事項については、属性の説明を参照してください。

## 適用可能なイベント

|         |           |           |
|---------|-----------|-----------|
| Click   | Create    | DbClick   |
| Destroy | MouseDown | MouseMove |
| MouseUp | Popup     | VKeyPress |

## チェック・ボックス



たとえばオンとオフのように、明確に区別できる 2 つの状態の間でユーザーに選択させたい場合には、チェック・ボックス・パーツを使用してください。

チェック・ボックスに関連したラベルには、選択された時の設定が記述されます。

一般的に言えば、オプションのリストを提供するため、チェック・ボックスのグループを使用することができます。ユーザーは、1 つまたは複数のチェック・ボックスを選択することもできるし、あるいは何も選択しないことも可能です。オプションは互いに排他的でなく、したがって、1 つのチェック・ボックスの選択が、ウィンドウ上のその他のチェック・ボックスに影響しません。ユーザーが 2 つ以上のオプションから 1 つだけ選択できるようにしたい場合には、チェック・ボックスの代わりにラジオ・ボタンを使用してください。詳細については、42 ページの『ラジオ・ボタン』を参照してください。

チェック・ボックスの状態を設定するには、ユーザーはマウスでチェック・ボックスをクリックするか、チェック・ボックスがフォーカスされている時にキーボードのスペース・キーを押すか、または (割り当てられている場合には) 簡略記号キーを押すことができます。簡略記号キーの割り当てについては、*VisualAge for RPG* プログラミング, SC09-2449-05 を参照してください。

### パーツ属性

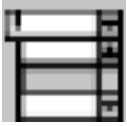
|             |             |           |
|-------------|-------------|-----------|
| AddLink*    | AllowLink*  | BackColor |
| BackMix     | Bottom      | Checked   |
| Enabled     | Focus       | FontBold  |
| FontItalic  | FontName    | FontSize  |
| FontStrike* | FontUnder*  | ForeColor |
| ForeMix     | Handle*     | Height    |
| HighLight*  | Label       | Left      |
| ParentName  | PartName    | PartType  |
| Refresh     | RemoveLink* | ShowTips  |
| TipText     | Top         | UserData  |
| Visible     | Width       |           |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|           |           |            |
|-----------|-----------|------------|
| Create    | Destroy   | MouseEnter |
| MouseExit | MouseMove | Popup      |
| Select    |           |            |

## 組み合わせボックス



組み合わせボックスは、特定の情報を入力するオプションまたは通常使用される選択項目のリストから選択するオプションをユーザーに提供します。

これは、ユーザーがスクロールできる値のリストを示すリスト・ボックスのついた入力フィールドで構成されます。ユーザーがこれらの値の 1 つを選択すると、その値が入力フィールドに表示され、既存のテキストと置き換えられます。これに代わる方法として、ユーザーは入力フィールドに値を直接入力することができますが、この値はリストのどの値とも一致している必要はありません。

組み合わせボックスには、いろいろなスタイルがあります。パーツのプロパティ・ノートブックから必要なスタイルを選択することができます。

関連情報については、以下を参照してください。

- 25 ページの『リスト・ボックス』
- 15 ページの『入力フィールド』

### パーツ属性

|             |             |            |
|-------------|-------------|------------|
| AddItemEnd  | AutoScroll* | BackColor  |
| BackMix     | Bottom      | Case*      |
| Count       | DelimChar   | DeSelect*  |
| DragEnable* | DropEnable* | Enabled    |
| FieldExit   | FirstSel    | Focus      |
| FontBold    | FontItalic  | FontName   |
| FontSize    | FontStrike* | FontUnder* |
| ForeColor   | ForeMix     | GetItem    |
| Handle*     | Height      | Index      |
| InsertItem* | ItemKey     | Left       |
| ParentName  | PartName    | PartType   |
| ReadOnly    | Refresh     | RemoveItem |
| Search*     | SearchType* | Selected   |
| SelectItem  | Sequence*   | SetItem    |
| SetTop*     | ShowTips    | SizeToFit* |
| Text        | TipText     | Top        |
| UseDelim    | UserData    | Visible    |
| Width       |             |            |

\*注: 制約事項については、属性の説明を参照してください。



## 適用可能なイベント

|            |           |           |
|------------|-----------|-----------|
| Change     | Create    | Destroy   |
| Drop*      | DropDown* | Enter     |
| GotFocus   | KeyPress  | LostFocus |
| MouseEnter | MouseExit | MouseMove |
| Popup      | Select    | VKeyPress |

\*注: 制約事項については、イベント記述を参照してください。

## \*Component

\*Component パーツによって、プログラマーはコンポーネントおよびシステム・ワイドの属性にアクセスし使用することができます。

\*Component パーツはコンポーネントの「パーツ表示」です。それぞれのコンポーネントに 1 つの \*Component パーツが自動的に作成されます。これは非表示でパレット上にはありません。

### パーツ属性

|                            |                      |                       |
|----------------------------|----------------------|-----------------------|
| Active*                    | Alarm                | AppData               |
| Button                     | ClipBoard            | CurrentDir            |
| Dialog                     | DIRName*             | DlgOwner              |
| DoEvents*                  | DspHeight            | DspWidth              |
| FileName                   | HelpWindow           | HostName*             |
| LookNFeel*                 | MsgData              | MsgFile*              |
| MsgID                      | MsgText              | Name                  |
| OS                         | Parent               | PartCount             |
| PartList                   | Platform             | PlugCmd*              |
| PlugDLL*                   | PlugID*              | PlugRC*               |
| PlugResult*                | Printer*             | SelPrinter*           |
| 427 ページの 『ShData』          | 428 ページの 『ShDataLen』 | 429 ページの 『ShDataName』 |
| 430 ページの 『ShDataPos』       | 431 ページの 『ShowMsgID』 | 447 ページの 『SwitchTo』 * |
| 497 ページの<br>『WrkStnName』 * |                      |                       |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

このパーツに関連したイベントはありません。

## コンポーネント参照



コンポーネント参照パーツは、1 つの VARPG コンポーネントを他のコンポーネントと通信できるようにします。コンポーネント参照パーツを使用して、他のコンポーネント上のパーツに作用します。参照されるコンポーネントは、コンポーネント参照パーツとして同じアプリケーションで実行中でなければなりません。

コンポーネント参照パーツは、また他のコンポーネントの指定されたイベントをモニターします。モニター・イベントが現れると、コンポーネント参照パーツによって **Notify** イベントが通知されます。

### パーツ属性

|            |           |            |
|------------|-----------|------------|
| AddSrcEvt  | AttrValue | Bottom     |
| CompName   | Left      | NotSrcEvt  |
| NotSrcPart | NotSrcWin | ParentName |
| PartName   | PartType  | RefAttr    |
| RefParent  | RefPart   | RmvSrcEvt  |
| UserData   | Visible   |            |

### 適用可能なイベント

|        |         |        |
|--------|---------|--------|
| Create | Destroy | Notify |
|--------|---------|--------|

## コンテナー



コンテナー・パーツを使用して、関連したレコードを保管します。レコードは、アイコン・ビュー、ツリー・ビュー、テキスト・ツリー・ビュー、または詳細ビューに表示することができます。

### パーツ属性

|             |            |            |
|-------------|------------|------------|
| AddRcd      | Arrange    | BackColor  |
| BackMix     | BatchAdd*  | BlankChar  |
| Bottom      | ChildCount | ChildList  |
| Collapsed   | ColNumber  | Count      |
| DeleteRcd   | DeSelect   | EditItem   |
| Enabled     | EndAdding* | ExtSelect* |
| FirstSel    | Focus      | FontBold   |
| FontName    | FontItalic | FontSize   |
| FontStrike* | FontUnder* | Focus      |
| ForeMix     | GetNewID   | GetRcdFld  |
| GetRcdIcon  | GetRcdText | Handle*    |
| InUse*      | Height     | Label      |
| Left        | MiniIcon   | NbrOfRec*  |
| ParentId    | ParentList | ParentName |
| PartName    | PartType   | ReadOnly   |
| RecordID    | Refresh    | RemoveRcd  |
| Selected    | SelectRcd  | SetRcdFld  |
| SetRcdIcon  | SetRcdText | SetTop*    |
| ShowTips    | SortAsc    | SortDesc   |
| TipText     | Top        | UserData   |
| View*       | Visible    | VisTitle   |
| VisTitlSep  | Width      |            |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|            |           |           |
|------------|-----------|-----------|
| Click      | Collapsed | ColSelect |
| Create     | DbClick   | Destroy   |
| Enter      | Expanded  | GotFocus  |
| KeyPress   | LostFocus | MouseDown |
| MouseEnter | MouseExit | MouseMove |

|           |       |        |
|-----------|-------|--------|
| MouseUp   | Popup | Select |
| VKeyPress |       |        |

## DDE クライアント



\* **制約事項:** このパーツは Java アプリケーションではサポートされていません。

DDE クライアントは、動的データ交換 (DDE) プロトコルをサポートする他のアプリケーション (スプレッドシート・アプリケーションなど) とデータを交換するために使用します。

交換は **DDE 会話**と呼ばれます。会話を開始するアプリケーションが **クライアント**で、これに応答するアプリケーションが**サーバー**です。あるアプリケーションが DDE をサポートしているかどうかを調べるには、そのアプリケーションに付属の資料を参照してください。

DDE クライアント・パーツは**コールド・リンク**と**ホット・リンク会話**の両方をサポートします。コールド・リンク会話は、サーバー・プログラムに対する明示的要求を作成するクライアント・プログラムからなっています。ホット・リンク会話は、データが変更される時に自動的にクライアント・プログラムを更新するサーバー・プログラムからなっています。

コールド・リンクまたはホット・リンク会話は、DDE クライアント・パーツのプロパティ・ノートブックからユーザーのプログラム・ロジックで構成することができます。

### パーツ属性

|            |            |            |
|------------|------------|------------|
| AppName    | Bottom     | DDEAddLink |
| DDEMode    | DDERmvLink | Execute    |
| Format     | Item       | Left       |
| ParentName | PartName   | PartType   |
| Poke       | Request    | TimeOut    |
| Top        | Topic      | UserData   |
| Visible    |            |            |

### 適用可能なイベント

|            |         |           |
|------------|---------|-----------|
| Create     | Data    | Destroy   |
| ExecuteAck | PokeAck | Terminate |
| Timeout    |         |           |

## 入力フィールド



予測できないなにかの値をユーザーに入力させたい場合には、入力フィールド・パーツを使用します。入力フィールドとは、ユーザーがテキストを入力するか、あるいは入れることができる領域のことです。通常、その境界は指示されます。現在表示されているよりも多くの情報がある場合には、ユーザーは入力フィールドのテキストをスクロールすることができます。

ユーザーは、文字、数字、あるいは 2 バイト文字セット (DBCS) データを受け入れるように、入力フィールド・パーツを構成することができます。

この入力フィールドを読み取り専用にして、ユーザーによって直接変更できない情報を含めることができます。

パーツ・パレットの入力フィールド・パーツを指示してクリックしてから、サブファイル・パーツ上のそれをクリックして、サブファイル入力フィールドを作成することができます。

### パーツ属性

|             |            |             |
|-------------|------------|-------------|
| AddLink*    | Alignment  | AllowLink*  |
| AutoScroll* | AutoSelect | BackColor   |
| BackMix     | Bottom     | CapsLock    |
| Copy        | CsrAtEnd   | Cut         |
| DataType    | Delete     | DragEnable* |
| DropEnable* | Enabled    | FieldExit   |
| Focus       | FontBold   | FontItalic  |
| FontName    | FontSize   | FontStrike* |
| FontUnder*  | ForeColor  | ForeMix     |
| Handle*     | Height     | InsertMode* |
| Left        | Masked     | ParentName  |
| PartName    | PartType   | Paste       |
| ReadOnly    | Refresh    | RemoveLink* |
| ShowTips    | Text       | TextEnd     |
| TextLength  | TextSelect | TextStart   |
| TipText     | Top        | UserData    |
| Visible     | Width      |             |

\*注: 制約事項については、属性の説明を参照してください。

## 適用可能なイベント

|           |           |            |
|-----------|-----------|------------|
| Change    | Click     | Create     |
| DbClick   | Destroy   | Drop       |
| GotFocus  | KeyPress  | Link*      |
| LostFocus | MouseDown | MouseEnter |
| MouseExit | MouseMove | MouseUp    |
| Popup     | VKeyPress |            |

\*注: 制約事項については、イベント記述を参照してください。



## グラフ



グラフ・パーツにより、プロジェクト内でグラフを作成および設計することができます。実行時に、データをグラフに送り、グラフ属性およびグラフ・タイプを変更することができます。グラフ・パーツは、「円」、「線」、「棒」、および「線と棒」グラフ・タイプをサポートします。

棒グラフおよび折れ線グラフ・タイプは、**ToolTip** テキスト制御をサポートします。使用可能な場合には、データ・ポイント上にマウスを移動すると、**ToolTip** テキスト制御が表示されます。プログラム・ロジック内でこのサポートを使用するには、そのポイントの値を **TipText** 属性に設定して、そのグラフ・パーツが入っているウィンドウに **ShowTips** 属性を設定してください。

### パーツ属性

|             |             |             |
|-------------|-------------|-------------|
| AutoInc     | BarLabel    | Bottom      |
| Color       | ColorArea   | ColorMix    |
| DataGroup   | DataPoint   | DataValue   |
| Enabled     | FillStyle*  | FontArea    |
| FontBold    | FontItalic  | FontName    |
| FontSize    | FontStrike* | FontUnder*  |
| GnEqGrpCol  | GnEqPntCol  | GraphType   |
| GroupLabel  | GrphHiLite  | Handle*     |
| Height      | HitItem*    | HlitPoints* |
| LabelPlace  | Left        | LegendType  |
| ParentName  | PartName    | PartType    |
| Refresh     | StartNew    | TipText     |
| Title       | TitlePlace  | Top         |
| UnderPoint* | UseData     | UserData    |
| Visible     | Width       | XAxisLabel  |
| YAxisLabel  | Yinc        |             |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|           |           |            |
|-----------|-----------|------------|
| Click     | Create    | DblClick   |
| Destroy   | MouseDown | MouseEnter |
| MouseExit | MouseMove | MouseUp    |
| Popup     |           |            |

---

## グラフィック・プッシュボタン



頻繁に使用するアクションへの便利なアクセスを提供するためには、グラフィック・プッシュボタンを使用してください。

グラフィック・プッシュボタンはプッシュボタンと同じ機能を提供します。これは、ユーザーが選択すると開始されるアクションを示しますが、その機能を記述するラベルを表示する代わりにイメージを表示します。FileName 属性は、使用するイメージの名前を指定します。

有効な **Windows** イメージ形式には、次が含まれます。

- Windows および OS/2<sup>®</sup> ビットマップ (BMP、VGA、BGA、RLE、DIB、RL4、RL8)
- アイコン (ICO)
- Microsoft/Aldus タグ・イメージ・ファイル・フォーマット (TIF、TIFF)
- CompuServe グラフィック交換形式 (GIF)
- ZSoft PC Paintbrush イメージ・ファイル形式 (PCX)
- Truevision Targa/Vista ビットマップ (TGA、VST、AFI)
- Amiga インターリーブ・ビットマップ形式 (IFF、ILBM)
- X Windows ビットマップ (XBM)
- IBM プリンター・ページ・セグメント (PSE、PSEG、PSEG38PP、PSEG3820)
- Joint Photographic Experts Group 形式 (JPG、JPEG)

注: この製品の JPG/JPEG 形式に対するサポートは、部分的に Independent JPEG Group の働きに基づいています。

有効な **Java** イメージ形式には、次が含まれます。

- CompuServe グラフィック交換形式 (GIF)
- Joint Photographic Experts Group 形式 (JPG、JPEG)

関連情報については、41 ページの『プッシュボタン』を参照してください。

## パーツ属性

|            |           |          |
|------------|-----------|----------|
| Bottom     | Enabled   | FileName |
| Focus      | Handle*   | Height   |
| HelpEnable | HighLight | Left     |
| ParentName | PartName  | PartType |
| Refresh    | ShowTips  | TipText  |
| Top        | UserData  | Validate |
| Visible    | Width     |          |

\* 注: 制約事項については、属性の説明を参照してください。

## 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Create    | Destroy    | GotFocus  |
| LostFocus | MouseEnter | MouseExit |
| MouseMove | Popup      | Press     |

## グループ・ボックス



グループ・ボックスは、ウィンドウ内のパーツのグループを視覚的に区別するために使用します。

グループ・ボックスは、パーツが関連していることを示すために、パーツのグループの周りに描かれた四角のボックスです。通常グループ・ボックスにラベルを付ける場合に有用です。ラベルが必要でない場合には、外枠を使用することができます。

### パーツ属性

|             |            |            |
|-------------|------------|------------|
| Bottom      | Enabled    | FontBold   |
| FontItalic  | FontName   | FontSize   |
| FontStrike* | FontUnder* | ForeColor  |
| ForeMix     | Handle*    | Height     |
| Label       | Left       | ParentName |
| PartName    | PartType   | Refresh    |
| Top         | UserData   | Visible    |
| Width       |            |            |

注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

## 水平方向のスクロール・バー



ユーザーが情報のペインを左から右または右から左にスクロールできるようにするには、水平方向のスクロール・バー・パーツを使用します。この情報は、ファイルのリストであったり、データベース中のレコードであったり、また文書中の列などである場合があります。Range 属性を使用してスクロールするオブジェクトの合計数を表し、 PageSize 属性を使用して 1 ページに表示できるオブジェクトの数を決定することができます。

### パーツ属性

|            |          |          |
|------------|----------|----------|
| Bottom     | Enabled  | Focus    |
| Handle*    | Height   | Left     |
| NextLine   | NextPage | PageSize |
| ParentName | PartName | PartType |
| Position   | PrevLine | PrevPage |
| Range      | Top      | UserData |
| Visible    | Width    |          |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |        |
|--------|---------|--------|
| Create | Destroy | Scroll |
|--------|---------|--------|

## イメージ



ピクチャーを表示するには、イメージ・パーツを使用します。FileName 属性は、使用するイメージの名前を指定します。

有効な **Windows** イメージ形式には、次が含まれます。

- Windows および OS/2 ビットマップ (BMP、VGA、BGA、RLE、DIB、RL4、RL8)
- アイコン (ICO)
- Microsoft/Aldus タグ・イメージ・ファイル・フォーマット (TIF、TIFF)
- CompuServe グラフィック交換形式 (GIF)
- ZSoft PC Paintbrush イメージ・ファイル形式 (PCX)
- Truevision Targa/Vista ビットマップ (TGA、VST、AFI)
- Amiga インターリーブ・ビットマップ形式 (IFF、ILBM)
- X Windows ビットマップ (XBM)
- IBM プリンター・ページ・セグメント (PSE、PSEG、PSEG38PP、PSEG3820)
- Joint Photographic Experts Group 形式 (JPG、JPEG)

**注:** この製品の JPG/JPEG 形式に対するサポートは、部分的に Independent JPEG Group の働きに基づいています。

有効な **Java** イメージ形式には、次が含まれます。

- CompuServe グラフィック交換形式 (GIF)
- Joint Photographic Experts Group 形式 (JPG、JPEG)

これらのファイルは、ホスト上ではなく、プログラマブル・ワークステーション (PWS) 上にあります。システム特有のビットマップおよびアイコン・ファイルは、適切な実行時ディレクトリー (RT\_JAVA または RT\_WIN32) に保管して、アプリケーションをパッケージ化する時にパッケージ・ユーティリティーがそれらを組み込むようにしてください。

**注:** イメージ・パーツは、キャンバス付きのノートブック・ページかまたはキャンバス付きウィンドウにしかドロップできません。

## パーツ属性

|          |             |           |
|----------|-------------|-----------|
| AddLink* | AllowLink*  | BackColor |
| BackMix  | Border      | Bottom    |
| Enabled  | FileName    | Handle*   |
| Height   | Left        | Magnify   |
| Panel    | ParentName  | PartName  |
| PartType | Print       | PrintAsIs |
| Refresh  | RemoveLink* | ShowTips  |
| TipText  | Top         | UserData  |
| Visible  | Width       |           |

\* 注: 制約事項については、属性の説明を参照してください。

## 適用可能なイベント

|           |           |            |
|-----------|-----------|------------|
| Click     | Create    | DbClick    |
| Destroy   | Link*     | MouseEnter |
| MouseExit | MouseMove |            |

\* 注: 制約事項については、イベント記述を参照してください。

## Java Bean



\* **制約事項:** このパーツは、Windows アプリケーションではサポートされません。

プロジェクトに **JavaBeans®** を追加するには、**Java Bean** パーツを使用します。Java メソッドを呼び出すことによって、直接 **JavaBeans** を使用できます。Java メソッドの呼び出しの詳細については、*VisualAge for RPG プログラミング*, SC09-2449-05 を参照してください。

**Java Bean** パーツを使用するアプリケーションを開発するには、**Sun Microsystems** の **Java 2 ソフトウェア開発キット (J2SDK)**、標準版、バージョン 1.2 以上がワークステーションにインストールされている必要があります。J2SDK がいない場合には、**Sun Microsystems** から次の URL でダウンロードすることができます。

<http://java.sun.com/products/>

J2SDK をインストールしたら、**PATH** 環境変数を **Java** コンパイラと **jvm.dll**(J2SDK および **JRE** (Java 実行時環境) のパーツ) の両方が入っている場所を示すように設定してください。たとえば、J2SDK のホーム・ディレクトリーが **x:\jdk1.2** の場合には、次の **path** ステートメントを追加してください。

```
x:\jdk1.2\bin
```

```
x:\jdk1.2\jre\bin\classic
```

### パーツ属性

|          |          |            |
|----------|----------|------------|
| AddEvent | Bottom   | Enabled    |
| Height   | Left     | ParentName |
| PartName | PartType | RmvEvent   |
| ShowProp | Top      | UserData   |
| Visible  | Width    |            |

### 適用可能なイベント

|        |         |           |
|--------|---------|-----------|
| Create | Destroy | BeanEvent |
|--------|---------|-----------|



## リスト・ボックス



リスト・ボックス・パーツを使用して、1 つまたは複数の項目を選択できる項目のリストをユーザーに提供します。リスト・ボックスは読み取り専用の項目から構成されています。リスト・ボックス中の項目は文字のストリングです。

ユーザーは水平方向および縦方向のスクロール・バーを使って、現在表示されていないリストのセクションを見ることができます。ユーザーは、1 項目だけを選択するように、あるいは複数項目を選択できるようにリスト・ボックスを構成することができます。Search、SearchType、および Case 属性を使用して、リストの特定の項目を容易に検索することができます。

### パーツ属性

|            |             |             |
|------------|-------------|-------------|
| AddItemEnd | AddLink*    | AllowLink*  |
| BackColor  | BackMix     | Bottom      |
| Case*      | Count       | DelimChar   |
| DeSelect   | DragEnable* | DropEnable* |
| Enabled    | ExtSelect   | FirstSel    |
| Focus      | FontBold    | FontItalic  |
| FontName   | FontSize    | FontStrike* |
| FontUnder* | ForeColor   | ForeMix     |
| GetItem    | Handle*     | Height      |
| Index      | InsertItem* | ItemKey     |
| Left       | MultSelect  | NbrOfSel    |
| ParentName | PartName    | PartType    |
| Refresh    | RemoveItem  | RemoveLink* |
| Search*    | SearchType* | Selected    |
| SelectItem | SelectList  | Sequence*   |
| SetItem    | SetTop      | ShowTips    |
| SizeToFit  | TipText     | Top         |
| TopItem    | UseDelim    | UserData    |
| Visible    | Width       |             |

\* 注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Create    | Destroy    | Drop*     |
| Enter*    | GotFocus   | KeyPress  |
| LostFocus | MouseEnter | MouseExit |

|           |       |        |
|-----------|-------|--------|
| MouseMove | Popup | Select |
| VKeyPress |       |        |

\* 注: 制約事項については、イベント記述を参照してください。

## メディア



メディア・パーツは、オーディオ情報を再生または記録し、ビデオ・ファイルを再生するために使用します。

メディア・パーツによって、ユーザー・プログラムは wave (.WAV)、MIDI (.MID)、および QuickTime ムービー (.MOV) ファイルを処理することができます。オーディオ・ファイルを使用するには、これらのファイルの処理に使用可能なサウンド・カードがコンピューターに装備されていなければなりません。サウンド・ファイルを記録するためには、マイクロホンまたはサウンド・カード用にサポートされているその他の入力装置が必要となります。メディア・パーツでは、MIDI ファイルは記録することはできません。

Java アプリケーションには、Java メディア・フレームワーク (JMF) API が必要です。メディア・パーツは、Java 環境でのオーディオおよびビデオ・ファイルの再生だけをサポートしています。

処理できるビデオ・ファイルの形式は次の通りです。 MPEG (\*.mpg) ファイル、QUICKTIME ムービー (\*.mov) ファイル、\*.dat ファイル、Microsoft® Video for Windows \*.avi ファイルが Windows 用にサポートされています。これらのビデオ・ファイルを再生するためには、コンピューターに適切なドライバーがなければなりません。

### パーツ属性

|             |            |            |
|-------------|------------|------------|
| AddLink*    | AllowLink* | AudioMode  |
| Bass*       | Bottom     | FileName   |
| Handle*     | InPlace    | Left       |
| Length      | Panel      | ParentName |
| PartName    | PartType   | Position   |
| RemoveLink* | Top        | Treble*    |
| UserData    | Visible    | Volume     |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|          |        |         |
|----------|--------|---------|
| Complete | Create | Destroy |
| Link*    |        |         |

\*注: 制約事項については、イベント記述を参照してください。

## メディア・パネル



**注:** このパーツは Java アプリケーションではサポートされていません。

メディア・パネル・パーツを使用して、頻繁に使用するアクションを容易にアクセスできるようにします。

また、メディア・パネル・パーツを使用して、プログラム・ロジックを作成することなしに、他のパーツをユーザー制御することもできます。たとえば、それを使用してプッシュボタンを作成したり、メディア・パーツのボリュームまたはモードを制御するスライダー・コントロールを作成することができます。

メディア・パネル・パーツのプロパティ・ノートブックでは、次のことを決定することができます。

- 定義された一連のボタンからどのボタンをメディア・パネルに入れるか
- 位置およびボリュームのスライダー・コントロールを表示するかどうか

**注:** メディア・パネル・パーツはキャンバス付きのノートブックまたはキャンバス付きウィンドウにだけ入れることができます。

### パーツ属性

|            |           |            |
|------------|-----------|------------|
| AddLink    | AllowLink | BackColor  |
| BackMix    | Bottom    | Enabled    |
| Handle     | Height    | Left       |
| PanelItem  | PanelMode | ParentName |
| PartName   | PartType  | Position   |
| RemoveLink | Top       | UserData   |
| Visible    | Volume    | Width      |

### 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Change    | Create     | Destroy   |
| Link      | MouseEnter | MouseExit |
| MouseMove | Popup      | Press     |

## メニュー・バー



メニュー・バー・パーツは、ユーザーがプルダウン・メニューにアクセスするために使用します。メニュー・バーには、サブメニュー・パーツおよびメニュー項目パーツを追加することができます。

メニュー・バーは、ウィンドウ・フレームの上の方、タイトル・バーのすぐ下に表示されます。ユーザーがメニュー・バーからメニュー項目を選択すると、プルダウン・メニューが現れ、メニュー上に項目を表示します。メニュー項目を選択すると、記述されたアクションが即時に開始されます。

**注:** このパーツのプロパティ、イベントなどを操作できるのは、プロジェクト・ツリー・ビューのそのポップアップ・メニューからだけです。

関連情報については、以下を参照してください。

- 30 ページの『メニュー項目』
- 50 ページの『サブメニュー』
- 39 ページの『ポップアップ・メニュー』

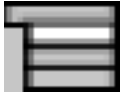
### パーツ属性

| PartType | PartName | ParentName |
|----------|----------|------------|
| UserData |          |            |

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

## メニュー項目



プルダウンあるいはポップアップ・メニューを構成するには、メニュー項目を使用します。

メニュー項目は、ユーザーがその項目を選択した時に開始されるアクションを記述します。

メニューを構成するには、次のようにしてください。

1. サブメニュー・パーツをメニュー・バーまたはポップアップ・メニューにドロップする。
2. メニュー項目をサブメニューにドロップする。

**注:** このパーツのプロパティ、イベントなどを操作できるのは、プロジェクト・ツリー・ビューのそのポップアップ・メニューからだけです。

関連情報については、以下を参照してください。

- 29 ページの『メニュー・バー』
- 39 ページの『ポップアップ・メニュー』
- 50 ページの『サブメニュー』

### パーツ属性

|          |            |          |
|----------|------------|----------|
| Checked  | Enabled    | FileName |
| Label    | ParentName | PartName |
| PartType | UserData   | Visible  |

### 適用可能なイベント

|        |         |            |
|--------|---------|------------|
| Create | Destroy | MenuSelect |
|--------|---------|------------|

## メッセージ・サブファイル



定義済みメッセージを表示するか、エラーや状況情報など、プログラム・ロジックで提供したテキストを表示するには、メッセージ・サブファイル・パーツを使用してください。

このパーツは常にウィンドウ・フレームの最下部に位置付けされ、ウィンドウの幅を実行します。ウィンドウの幅のサイズ変更を行うことはできませんが、さらにメッセージを表示するようにその高さのサイズ変更を行うことはできます。実行時点で、ユーザーはすべてのメッセージを見るために、スクロール・バーを使用することができます。

### パーツ属性

|             |             |             |
|-------------|-------------|-------------|
| AddMsgID    | AddMsgText  | Count       |
| DragEnable* | DropEnable* | Enabled     |
| FirstSel    | FontBold    | FontItalic  |
| FontName    | FontSize    | FontStrike* |
| FontUnder*  | ForeColor   | ForeMix     |
| GetItem     | Handle*     | Height      |
| Index       | MsgSubText  | NbrOfSel    |
| ParentName  | PartName    | PartType    |
| RemoveMsg   | Selected    | ShowTips    |
| TipText     | UserData    | Visible     |

\* 注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Create    | Destroy    | Drop      |
| Enter     | MouseEnter | MouseExit |
| MouseMove | Popup      | Select    |

## 複数行編集



ユーザーが複数行のテキストを入力できるようにしたい場合には、複数行編集パーツを使用します。

複数行編集パーツは境界を定義しています。テキストのすべてが表示されているとは限りません。ユーザーは、上下左右にスクロールして、現在表示されていないテキストを表示することができます。

### パーツ属性

|             |             |             |
|-------------|-------------|-------------|
| AddLineEnd  | AddOffset   | BackColor   |
| BackMix     | Bottom      | CanUndo     |
| CharOffset  | Copy        | CsrLine     |
| CsrPos      | Cut         | Delete      |
| DragEnable* | DropEnable* | Enabled     |
| Focus       | FontBold    | FontItalic  |
| FontName    | FontSize    | FontStrike* |
| FontUnder*  | ForeColor   | ForeMix     |
| Handle*     | Height      | InsertLine  |
| InsertText  | Left        | LineNumber  |
| LineText    | NbrOfLines  | ParentName  |
| PartName    | PartType    | Paste       |
| ReadOnly    | Refresh     | ShowTips    |
| Text        | TextEnd     | TextLength  |
| TextSelect  | TextStart   | TextString  |
| TipText     | Top         | TopLine     |
| Undo        | UserData    | Visible     |
| Width       | WordWrap    |             |

\* 注: 制約事項については、その属性の説明を参照してください。

### 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Change    | Click      | Create    |
| DbiClick  | Destroy    | Drop      |
| GotFocus  | KeyPress   | LostFocus |
| MouseDown | MouseEnter | MouseExit |
| MouseMove | MouseUp    | Popup     |
| VKeyPress |            |           |



## ノートブック



名前、出荷先アドレス、注文、信用などのカテゴリーにソートされたカスタマー情報といったトピックで論理的に類別できるデータを提示するには、ノートブック・パーツを使用してください。

ノートブック・パーツは、バインド・ノートブックのグラフィカル表現です。(Windows アプリケーションでは、これはタブ制御として知られています。)ユーザーはノートブックにページを追加することができ、またそれらのページを、タブ付きの区切り記号によって分離されたセクションにグループ化することができます。ノートブック・ページにキャンバスがある場合には、ページに複数のパーツを追加することができます。ノートブック・ページにキャンバスがない場合には、ページに 1 つのパーツしか追加することができません。

ユーザーは、ノートブックのページをめくって、あるページから次のページに移動することもできるし、区切り記号タブをクリックして、あるセクションに直接進むこともできます。

次の方法でノートブック・ページを追加することができます。

- ノートブック・パーツ用のプロパティ・ノートブックを使用する。
- プロパティ・タブ またはキャンバス付きノートブック・ページを、ノートブック・パーツで指示してクリック (ドラッグ・アンド・ドロップ) する。

関連情報については、以下を参照してください。

- 34 ページの『ノートブック・ページ』
- 35 ページの『キャンバス付きノートブック・ページ』

### パーツ属性

|            |             |            |
|------------|-------------|------------|
| BackColor  | BackMix     | Bottom     |
| Count      | Enabled     | Focus      |
| FontBold   | FontItalic  | FontName   |
| FontSize   | FontStrike* | FontUnder* |
| ForeColor  | ForeMix     | Handle*    |
| Height     | Left        | PageNumber |
| ParentName | PartName    | PartType   |
| Refresh    | ShowTabs*   | Top        |
| UserData   | Visible     | Width      |

\*注: 制約事項については、属性の説明を参照してください。

## 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

---

## ノートブック・ページ



ノートブックにページを追加するためにはノートブック・ページ・パーツを使用します。

ユーザーはノートブック・ページに 1 つのパーツしか追加できません。そのパーツは、ページ全体に適合するように自動的にサイズ設定されます。ページに複数のパーツを追加したい場合には、ノートブック・ページでキャンバス・パーツを指示してクリックしなければなりません。代替方法として、ユーザーはキャンバス・パーツ付きノートブック・ページを使用して、ステップを保管することができます。

**注:** このパーツのプロパティ、イベントなどを操作できるのは、プロジェクト・ツリー・ビューのそのポップアップ・メニューからだけです。

ユーザーは左および右矢印キーを押して 1 つのページから次のページに移動することができます。

関連情報については、次を参照してください。

- 33 ページの『ノートブック』
- 35 ページの『キャンバス付きノートブック・ページ』

### パーツ属性

|          |          |            |
|----------|----------|------------|
| Enabled  | OnTop    | ParentName |
| PartName | PartType | Refresh    |
| TabImage | TabLabel | UserData   |
| Visible  |          |            |

## 適用可能なイベント

|             |         |            |
|-------------|---------|------------|
| Create      | Destroy | PageSelect |
| SelPending* |         |            |

\* **注:** 制約事項については、イベント記述を参照してください。

## キャンバス付きノートブック・ページ



ノートブック・パーツにページを追加するためにはキャンバス付きのノートブックを使用します。

キャンバス・パーツは、ノートブック・ページ・パーツのクライアント域を占めます。キャンバス・パーツにパーツを追加することによって、ユーザーはグラフィカル・ユーザー・インターフェースを作成することができます。

ページに 1 つのパーツのみを追加したい場合には、キャンバス・パーツ付きのノートブック・ページの代わりに、ノートブック・ページ・パーツを使用することができます。ノートブック・ページ・パーツは、その中にキャンバスを持っていないので、追加したパーツは自動的にサイズが決まります。

関連情報については、以下を参照してください。

- 33 ページの『ノートブック』
- 34 ページの『ノートブック・ページ』

### パーツ属性

|          |          |            |
|----------|----------|------------|
| Enabled  | OnTop    | ParentName |
| PartName | PartType | Refresh    |
| TabImage | TabLabel | UserData   |
| Visible  |          |            |

### 適用可能なイベント

|            |         |            |
|------------|---------|------------|
| Create     | Destroy | PageSelect |
| SelPending |         |            |

## ODBC/JDBC インターフェース



ODBC/JDBC インターフェース・パーツは、Windows ODBC API または Sun Microsystems の JDBC API をサポートするデータベース・ファイルを処理できる機能を提供します。これらのデータベース・ファイル・タイプの例としては、Foxpro、Access、および Paradox があります。

ODBC/JDBC インターフェース・パーツを使用できるアプリケーションを開発するには、SQL に精通している必要があり、Windows ODBC SDK または Sun Microsystems の Java 2 ソフトウェア開発キット (J2SDK) 標準版のいずれかがワークステーションにインストールされていなければなりません。

ODBC SDK がない場合には、次の URL の Microsoft からダウンロードすることができます。

<http://www.microsoft.com/odbc/download.htm>

JDBC サポートは、Java™ 2 ソフトウェア開発キット (J2SDK) バージョン 1.2 (Windows 用) のパーツです。J2SDK がない場合には、Sun Microsystems から次の URL でダウンロードすることができます。

<http://java.sun.com/products/>

JDBC データベースのデータにアクセスして操作するアプリケーションには、適切な JDBC 2.0 承諾ドライバーが必要です。JDBC ドライバーおよびその他の情報は次の URL で検索することができます。

<http://java.sun.com/products/jdbc/>

**注:** JDBC はアプレットではサポートされません。

ODBC または JDBC データベースは 1 つまたは複数のテーブルから構成されています。データは一連の行としてテーブルに保管されます。各行またはレコードには、データの入った多数の列が含まれています。行を操作するために、あるいはプログラム・フィールドとテーブルの列の間でデータを移動するために、ユーザーのプログラムは ODBC/JDBC インターフェース・パーツ属性とともに SQL ステートメントを実行依頼することができます。

既存のデータベースを処理するためには、その前にユーザーの VARPG プログラムをデータベースに接続し、どのテーブルを参照するかを指示しなければなりません。テーブル中の行を操作するためには、ユーザーのプログラムで、ODBC/JDBC インターフェース・パーツによって戻され保守されるレコードを識別するレコード・セットを作成しなければなりません。行の中のデータをアクセスするためには、テーブル行で使用される各列を、プログラム中のプログラム・フィールドとバインドしなければなりません。Java アプリケーションでは、ポインターはサポートされません。列はパーツにバインドされます。バインドに使用できるパーツは、静的テキストおよび入力フィールドのパーツだけです。

ODBC/JDBC インターフェース・パーツを使用する Java アプリケーションを作成する場合には、そのアプリケーションを実行するエンド・ユーザーはワークステーションに *varpgjdb.jar* ファイルをインストールし、そのクラスパス・ステートメントにその場所を追加しなければなりません。パッケージ・ユーティリティーにはこの JAR ファイルが含まれています。この JAR ファイルは *WDSCjava* サブディレクトリーにあります。

### パーツ属性

|             |              |            |
|-------------|--------------|------------|
| AllowChg*   | BindPart     | Bottom     |
| BufferDec*  | BufferLen*   | BufferPtr* |
| BufferType* | CharData     | Column     |
| ColumnDec   | ColumnLen    | ColumnName |
| Columns     | ColumnType   | Connect    |
| Connected   | ConnectStr   | CurrentRow |
| DeleteRow   | ExecuteSQL   | Fetch      |
| FetchNext   | FetchPrior   | GetTables  |
| Handle*     | Height       | InsertRow  |
| IsData      | Left         | ParentName |
| PartName    | PartType     | Refresh    |
| Rows*       | SQLException | SQLMsgBox  |
| SQLQuery    | Top          | UnBind     |
| UpdateRow   | UserData     | Visible    |
| Width       |              |            |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

## 外枠



パーツのグループが関連していることを指示するためには、その回りに外枠を使用してください。

外枠とは、長方形の、ラベルのないボックスです。ボックスにラベルが必要な場合には、代わりにグループ・ボックス・パーツを使用してください。

関連情報については、 20 ページの『グループ・ボックス』を参照してください

### パーツ属性

|          |            |          |
|----------|------------|----------|
| Bottom   | Handle*    | Height   |
| Left     | ParentName | PartName |
| PartType | Refresh    | Top      |
| UserData | Visible    | Width    |

注：制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

## ポップアップ・メニュー



ユーザー・インターフェース上の特定のパーツに関連する多くの選択項目を表示するためには、ポップアップ・メニュー・パーツを使用します。ユーザーは、ポップアップ・メニュー・パーツにメニュー項目パーツおよびサブメニュー・パーツを追加することができます。

このメニューは、ユーザーが該当するキーまたはマウス・ボタンを押した時に現れるので、“ポップアップ”と呼ばれています。

**注:** このパーツのプロパティ、イベントなどを操作できるのは、プロジェクト・ツリー・ビューのそのポップアップ・メニューからだけです。

関連情報については、以下を参照してください。

- 29 ページの『メニュー・バー』
- 30 ページの『メニュー項目』
- 50 ページの『サブメニュー』

### パーツ属性

|            |          |          |
|------------|----------|----------|
| Handle*    | InvName  | InvPName |
| ParentName | PartName | PartType |
| UserData   | Visible* | X        |
| Y          |          |          |

**\*注:** 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

このパーツにはイベントはありません。

## 進行状況バー



ファイルのコピー、データベースのロード、などの処理の進行をグラフィカルに示すためには、進行状況バー・パーツを使用してください。

たとえば、100 個のファイルのコピーの進行状況を示すために **PBRange** 属性を 100 に設定し、**PBStepSize** 属性を 10 に設定することができます。そうすると、ユーザーのコードは、コピー・ファイル処理をモニターし、10 個のファイルがコピーされるごとに、進行状況バーの標識を前方に移動することができます。

Java アプリケーションでは、進行状況バーの幅がその高さよりも小さい場合には、進行状況バーは縦方向となります。

### パーツ属性

|          |            |          |
|----------|------------|----------|
| Bottom   | Handle*    | Height   |
| Left     | ParentName | PartName |
| PartType | PBRange    | PBSetPos |
| PBStep   | PBStepSize | Top      |
| UserData | Visible    | Width    |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|



## プッシュボタン



頻繁に使用するアクションへの便利なアクセスを提供するためには、プッシュボタンを使用してください。

各プッシュボタン・パーツは特定のアクションをコントロールします。ユーザーがプッシュボタンをクリックすると、ただちにアクションが開始されます。プッシュボタン上のテキスト・ラベルは、そのアクションを説明するものです。

18 ページの『グラフィック・プッシュボタン』と比較してください

### パーツ属性

|           |             |            |
|-----------|-------------|------------|
| BackColor | BackMix     | Border*    |
| Bottom    | Enabled     | Focus      |
| FontBold  | FontItalic  | FontName   |
| FontSize  | FontStrike* | FontUnder* |
| ForeColor | ForeMix     | Handle*    |
| Height    | HelpEnable  | HighLight  |
| Label     | Left        | ParentName |
| PartName  | PartType    | Refresh    |
| ShowTips  | TipText     | Top        |
| UserData  | Validate    | Visible    |
| Width     |             |            |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Create    | Destroy    | GotFocus  |
| LostFocus | MouseEnter | MouseExit |
| MouseMove | Popup      | Press     |

## ラジオ・ボタン



関連してはいるが、互いに排他的な選択項目のグループの中から 1 つだけユーザーが選択できるようにしたい場合には、ラジオ・ボタンを使用します。ユーザーが選択を行った時に、グループ内の前に選択した選択項目は選択解除されます。

ラジオ・ボタンは、そばにテキストのあるラベル付きの立体的な円形のボタンとして現れます。選択すると、円形のボタンはドットとして表示されます。

ユーザーが一度に複数の選択項目を選択できるようにしたい場合には、ラジオ・ボタンを使用しないでください。その場合には、7 ページの『チェック・ボックス』を参照してください。

### パーツ属性

|           |             |            |
|-----------|-------------|------------|
| BackColor | BackMix     | Bottom     |
| Checked   | Enabled     | Focus      |
| FontBold  | FontItalic  | FontName   |
| FontSize  | FontStrike* | FontUnder* |
| ForeColor | ForeMix     | Handle*    |
| Height    | HighLight*  | Label      |
| Left      | ParentName  | PartName   |
| PartType  | Refresh     | SelectIdx  |
| ShowTips  | TipText     | Top        |
| UserData  | Visible     | Width      |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|            |           |           |
|------------|-----------|-----------|
| Create     | Destroy   | Enter     |
| MouseEnter | MouseExit | MouseMove |
| Popup      | Select    |           |

## スライダー



スライダー軸に沿ったスライダー標識を動かすことによって、ユーザーが値を表示、設定、あるいは変更できるようにしたい場合には、スライダー・パーツを使用します。

スライダーは通常、秒数または度数などの度を超えた増分を持つ値に使用されるか、あるいは完了したタスクのパーセントを表示するために使用されます。

デフォルトによって、スライダーは左サイドにスライダー・シャフトを持って、ボックスの中央に水平方向に置かれます。スケールは、シャフトについての計測単位を表示するために表示されます。

以下を行うためには、スライダー・パーツ用のプロパティ・ノートブックを使用してください:

- スライダーが戻ることのできる値の範囲を設定
- ウィンドウ内でスライダーを縦方向または水平方向位置に設定
- スライダーが表す計測単位を示す目盛りを提供

### パーツ属性

|             |            |             |
|-------------|------------|-------------|
| AddLink*    | AllowLink* | BackColor   |
| BackMix     | Bottom     | Enabled     |
| Focus       | FontBold   | FontItalic  |
| FontName    | FontSize   | FontStrike* |
| FontUnder*  | ForeColor  | ForeMix     |
| Handle*     | Height     | Left        |
| Maximum     | Minimum    | ParentName  |
| PartName    | PartType   | Refresh     |
| RemoveLink* | ShowTips   | TickLabel   |
| TickNumber  | TipText    | Top         |
| UserData    | Value      | Visible     |
| Width       |            |             |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|            |           |           |
|------------|-----------|-----------|
| Change     | Create    | Destroy   |
| GotFocus   | Link*     | LostFocus |
| MouseEnter | MouseExit | MouseMove |
| Popup      |           |           |

**\*注:** 制約事項については、イベント記述を参照してください。

## スピン・ボタン



論理的な連続順序を持つ、関連しているが互いに排他的な選択項目のグループ (たとえば年月など) を順序通りに表示するために、スピン・ボタンを使用します。選択項目は、リング状に配置されているように表示されます。ユーザーは、上矢印キーを押して次に高い値に移ったり、下矢印キーを押して次に低い値に移ったりすることによって、選択項目全体で移動 (すなわち“スピン”) することができます。あるいは、選択項目の 1 つをスピン・ボタンの入力フィールドに直接入力することもできます。

### パーツ属性

|            |            |             |
|------------|------------|-------------|
| AddItemEnd | Alignment* | BackColor   |
| BackMix    | Bottom     | Enabled     |
| Focus      | FontBold   | FontItalic  |
| FontName   | FontSize   | FontStrike* |
| FontUnder* | ForeColor  | ForeMix     |
| Handle*    | Height     | Left        |
| Maximum    | Minimum    | ParentName  |
| PartName   | PartType   | ReadOnly    |
| Refresh    | RemoveItem | ShowTips    |
| Text       | TipText    | Top         |
| UserData   | Value      | Visible     |
| Width      |            |             |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|            |           |           |
|------------|-----------|-----------|
| Change     | Create    | Destroy   |
| GotFocus   | Link*     | LostFocus |
| MouseEnter | MouseExit | MouseMove |
| Popup      | SpinDown  | SpinEnd   |
| SpinUp     |           |           |

\*注: 制約事項については、イベント記述を参照してください。

## 静的テキスト



入力フィールド・パーツのプロンプトなどのその他のパーツに対しては、ラベルとして静的テキスト・パーツを使用します。静的テキスト・パーツは、エンド・ユーザー入力を受け入れません。Java アプリケーションでは、静的テキストを表示できるのは単一の行上だけです。

### パーツ属性

|             |             |             |
|-------------|-------------|-------------|
| Alignment   | BackColor   | BackMix     |
| Bottom      | DataType    | DragEnable* |
| DropEnable* | DropValue*  | Enabled     |
| FontBold    | FontItalic  | FontName    |
| FontSize    | FontStrike* | FontUnder*  |
| ForeColor   | ForeMix     | Handle*     |
| Height      | Label       | Left        |
| ParentName  | PartName    | PartType    |
| Refresh     | ShowTips    | TipText     |
| Top         | UserData    | Visible     |
| Width       |             |             |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Click     | Create     | DblClick  |
| Destroy   | Drop       | Link*     |
| MouseDown | MouseEnter | MouseExit |
| MouseMove | MouseUp    | Popup     |

\*注: 制約事項については、イベント記述を参照してください。

## 状況バー



ウィンドウのプロセスまたはアクションについての追加情報を指定するためには、状況バー・パーツを使用してください。状況バーに対して最大 5 つのペインを作成することができます。状況バー・パーツには、ウィンドウ・パーツの `StatusBar` 属性よりも融通性があります。

デフォルトでは、状況バーはウィンドウの最下部に作成されます。ただし、プロパティ・ノートブックを使用して、その位置を最上部に変更することができます。また、枠のスタイル、ペインの数、およびテキストの配置を設定することもできます。

### パーツ属性

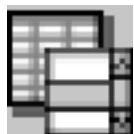
|          |            |          |
|----------|------------|----------|
| Handle*  | ParentName | PartName |
| PartType | SBIndex    | SBLabel  |
| SBPanes  | UserData   | Visible  |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

## サブファイル



それぞれが 1 つまたは複数のフィールドからなるレコードのリストを表示するには、サブファイル・パーツを使用してください。

サブファイル・パーツは iSeries™ サブファイルと類似した機能を持っています。ユーザーは、サブファイルのスクロール・バーを使用して、リストを水平方向または垂直方向にスクロールできます。

サブファイル入力フィールドを作成するには、「参照フィールドの定義」ウィンドウまたはパーツ・パレットからフィールドを指示してクリックして、サブファイル・パーツの上にそれをクリックしてください。プロパティ・ノートブックを使用してフィールドを追加することもできます。

**注:** サブファイル・パーツはキャンバス付きのノートブックまたはキャンバス付きウィンドウ上にだけ指示してクリックすることができます。

### パーツ属性

|             |             |            |
|-------------|-------------|------------|
| AddItemEnd  | AllowEdit   | AutoSelect |
| BackColor   | BackMix     | Bottom     |
| ButtonIdx   | Buttons     | ButtonTip  |
| CapsLock    | CellBGClr   | CellBGMix  |
| CellFGClr   | CellFGMix   | ColBGClr   |
| ColBGMix    | ColFGClr    | ColFGMix   |
| ColNumber   | ColWidth    | Count      |
| DColFRVCol  | DeSelect    | EditColumn |
| EditIndex   | EditText    | EnableBtn  |
| Enabled     | ExtSelect*  | FirstSel   |
| Focus       | FontArea    | FontBold   |
| FontItalic  | FontName    | FontSize   |
| FontStrike* | FontUnder*  | ForeColor  |
| ForeMix     | FreezeCol*  | Handle*    |
| HdgBGClr    | HdgBGMix    | HdgFGClr   |
| HdgFGMix    | HdgIdx      | HdgText    |
| Height      | Hidden      | HRule      |
| Index       | ItemCount   | Left       |
| MapViewCol  | MultiSelect | NbrOfSel   |
| OpenEdit    | PageSize    | ParentName |
| PartName    | PartType    | RemoveItem |
| RowBGClr    | RowBGMix    | RowFGClr   |



|            |            |            |
|------------|------------|------------|
| RowFGMix   | Scale      | Selected   |
| SelectItem | SelectList | SetTop     |
| SflNxtChg  | ShowTips   | SortAsc    |
| SortDesc   | SizeToFit  | StartAt    |
| TipText    | Top        | TopRecord  |
| UserData   | VColFRDCol | ViewColumn |
| Visible    | VRule      | Width      |

\* 注: 制約事項については、属性の説明を参照してください。

#### 適用可能なイベント

|           |            |           |
|-----------|------------|-----------|
| Change    | ColSelect  | Create    |
| Destroy   | Enter      | FirstRec  |
| GotFocus  | KeyPress   | LastRec   |
| LostFocus | MouseEnter | MouseExit |
| MouseMove | NextRec    | OpenEdit  |
| PageDown  | PageEnd    | PageTop   |
| PageUp    | Popup      | PrevRec   |
| Select    | VKeyPress  |           |

## サブメニュー



サブメニューは次のために使用します:

- 既存のメニューにあるメニュー項目から新しいカスケード・メニューを開始する。
- メニュー・バー上のメニュー項目からプルダウン・メニューを開始する。

サブメニューを作成した後に、ツリー・ビューでのみ、メニュー項目パーツをサブメニュー・パーツ上で指示してクリック (またはドラッグ・アンド・ドロップ) することによってサブメニューにメニュー項目を追加することができます。

**注:** このパーツのプロパティ、イベントなどを操作できるのは、プロジェクト・ツリー・ビューのそのポップアップ・メニューからだけです。

関連情報については、30 ページの『メニュー項目』を参照してください。

### パーツ属性

| ParentName | PartName | PartType |
|------------|----------|----------|
| UserData   |          |          |

### 適用可能なイベント

|        |         |  |
|--------|---------|--|
| Create | Destroy |  |
|--------|---------|--|

## タイマー



ユーザー・プログラムが事前設定時間間隔で操作を必ず実行しなければならない場合には、タイマー・パーツを使用します。たとえば、ウィンドウをクローズするか、または非活動化されている一定期間の後にアプリケーションを終了するためなどに、タイマー・パーツを使用することができます。

タイマー・パーツは時間の単位をカウントし、2つのイベントの間の事前設定時間間隔を計測して、インターバルが経過すると2番目のイベントに切り替えます。

GUIビルダーにタイマー・パーツを作成すると、そのパーツがアイコンとして設計ウィンドウ上に表示されます。しかし、タイマー・パーツのためのプロパティ・ノートブックでは、プログラムの実行中はアイコンを表示したくない旨を指定することができます。

**注:** 精密なタイミングが必要な場合には、タイマー・パーツは使用しないようにしてください。システムでは他のプログラムが実行されているために、**Tick** イベントは必ずしも指定された正確な間隔で起こらないことがあります。

### パーツ属性

|             |            |            |
|-------------|------------|------------|
| AddLink*    | AllowLink* | Bottom     |
| Interval    | Left       | Multiplier |
| ParentName  | PartName   | PartType   |
| RemoveLink* | TimerMode  | TimerTicks |
| Top         | UserData   | Visible    |

\* 注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |       |
|--------|---------|-------|
| Create | Destroy | Link* |
| Tick   |         |       |

\* 注: 制約事項については、イベント記述を参照してください。

## 縦方向のスクロール・バー



情報ペインを縦方向にスクロールできるようにするためには、縦方向のスクロール・バー・パーツを使用してください。この情報は、ファイルのリストであったり、データベース中のレコードであったり、また文書中の列などである場合があります。 `Range` 属性を使用してスクロールするオブジェクトの合計数を表し、`PageSize` 属性を使用して 1 ページに表示できるオブジェクトの数を判別することができます。

### パーツ属性

|            |          |          |
|------------|----------|----------|
| Bottom     | Enabled  | Focus    |
| Handle*    | Height   | Left     |
| NextLine   | NextPage | PageSize |
| ParentName | PartName | PartType |
| Position   | PrevLine | PrevPage |
| Range      | Top      | UserData |
| Visible    | Width    |          |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|        |         |        |
|--------|---------|--------|
| Create | Destroy | Scroll |
|--------|---------|--------|

## ウィンドウ



ウィンドウは、プログラムと相互に作用するユーザーの基本的な手段です。ユーザーのアプリケーションには、少なくとも 1 つのウィンドウが含まれていなければなりません。

たとえば、メニュー・バー、ポップアップ・メニュー、およびメッセージ・サブファイルのようなウィンドウ・フレームについて拡張するパーツを除いて、ウィンドウのクライアント域にはただ 1 つのパーツしか追加できません。追加できるパーツは自動的にクライアント域のサイズに合わせられます。

ウィンドウに複数のパーツを入れたい場合には、キャンバス・パーツを追加しなければなりません。あるいは、キャンバス・パーツ付きのウィンドウを使用して、ステップを節約します。

**注:** ウィンドウ・パーツは、パーツ・パレットでなく、パーツ・カタログのフレーム・セクションにあります。

関連情報については、以下を参照してください。

- 5 ページの『キャンバス』
- 55 ページの『キャンバス付きウィンドウ』

### パーツ属性

|             |             |             |
|-------------|-------------|-------------|
| Bottom      | Center      | Enabled     |
| FileName*   | Focus*      | FontBold*   |
| FontItalic* | FontName*   | FontSize*   |
| FontStrike* | FontUnder*  | Handle*     |
| Height      | IconHandle* | Label       |
| Left        | MouseIcon*  | MouseShape* |
| ParentName  | PartName    | PartType    |
| PBRange     | PBSetPos    | PBStep      |
| PBStepSize  | Print       | PrintAsIs   |
| ProgresBar  | Refresh     | SBLLabel    |
| SBPosition  | SBStyle     | ShowTips    |
| StatusBar   | Top         | UserData    |
| Visible     | Width       | WindowMode* |

\*注: 制約事項については、属性の説明を参照してください。

## 適用可能なイベント

|            |            |            |
|------------|------------|------------|
| Activate   | Close      | Create     |
| DeActivate | Destroy    | LClickTray |
| Moved      | RClickTray | ReSize     |
| ShutDown   |            |            |

## キャンバス付きウィンドウ



ウィンドウは、エンド・ユーザーがプログラムと対話を交わすのに用いる最も重要な手段です。キャンバス付きウィンドウ・パーツのキャンバスによって、ウィンドウに多くのパーツを追加することができます。

キャンバス部分の各種のパーツを指示してクリックし、それらを配置してから編成して、グラフィカル・ユーザー・インターフェースを作成することができます。メニュー・バー、ポップアップ・メニュー、およびメッセージ・サブファイルなどの、ウィンドウ・フレームの拡張となるパーツを追加することもできます。

ウィンドウのクライアント域に 1 つのパーツだけを入れる必要がある場合には、キャンバス・パーツ付きのウィンドウは必要ありません。代わりにウィンドウ・パーツ (パーツ・カタログの**フレーム**・セクションにある) を使用してください。キャンバスがなければ、追加したパーツは、クライアント域に合わせて自動的にサイズ変更されます。

関連情報については、以下を参照してください。

- 5 ページの『キャンバス』
- 53 ページの『ウィンドウ』

### パーツ属性

|             |             |             |
|-------------|-------------|-------------|
| Bottom      | Center      | Enabled     |
| FileName*   | Focus*      | FontBold*   |
| FontItalic* | FontName*   | FontSize*   |
| FontStrike* | FontUnder*  | Handle*     |
| Height      | IconHandle* | Label       |
| Left        | MouseIcon*  | MouseShape* |
| ParentName  | PartName    | PartType    |
| PBRange     | PBSetPos    | PBStep      |
| PBStepSize  | Print       | PrintAsIs   |
| ProgresBar  | Refresh     | SBLLabel    |
| SBPosition  | SBStyle     | ShowTips    |
| StatusBar   | Top         | UserData    |
| Visible     | Width       | WindowMode* |

\*注: 制約事項については、属性の説明を参照してください。

### 適用可能なイベント

|          |       |        |
|----------|-------|--------|
| Activate | Close | Create |
|----------|-------|--------|

|            |            |            |
|------------|------------|------------|
| DeActivate | Destroy    | LClickTray |
| Moved      | RClickTray | ReSize     |
| ShutDown   |            |            |



---

## 第 2 章 パーツ属性

VisualAge RPG を使用してグラフィカル・ユーザー・インターフェース (GUI) を作成する時には、そのパーツの **プロパティ・ノートブック** をオープンして、パーツのほとんどの属性を設定または変更することができます。

また、実行時に組み込み関数または命令コードを使用して、VisualAge RPG アプリケーション・プログラムのほとんどのパーツ属性を設定することもできます。これは、**%setatr** 組み込み関数をフリー・フォームの EVAL 命令コードでコーディングするか、または SETATR 命令コードを使用して実行することができます。パーツ属性の値を検索するには、**%getatr** 組み込み関数をフリー・フォームの EVAL 命令コードでコーディングするか、または GETATR 命令コードを使用します。

この項ではパーツの属性について説明します。それぞれの説明には以下が含まれています。

- 属性についての一般的な説明。
- 属性のデータ・タイプ。
- 属性が適用されるパーツのリスト。
- 組み込み関数または命令コードを使用して、属性を設定または取得できるかどうかを示すテーブル。このテーブルは、パーツの **プロパティ・ノートブック** を使用して属性を設定できるかどうかを示しています。
- 属性に許可された値。
- フリー・フォームおよび命令コードの構文の例。構文の詳細については、*VisualAge for RPG WINDOWS 版 言語解説書* を参照してください。

**注:** この章で使用している例のいくつかでは大文字のテキストだけを使用しています。しかし、ロジックは大文字でも小文字でも入力することができます。

- 適用可能な制約事項。

## Activate

ActiveX オブジェクトを活動化するには、この属性を 1 に設定します。

タイプ 数値

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'AX1': 'Activate') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AX1'      SETATR  1          'ACTIVATE'
*
```

## Active

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

コンポーネントの別のインスタンスが実行中であるかどうかを決定します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

使用可能な値

アクティブ値は次の 1 つとすることができます。

- 0** 現在のインスタンスがコンポーネントの唯一のインスタンスです
- 1** コンポーネントの別のインスタンスが実行中です

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C                               EVAL      Act=%getatr('*Component': '*Component': 'Active')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C      '*COMPONENT'  GETATR      'Active'  Act
*
```

## AddEvent

イベント・フィルター・リストにイベント名を追加します。特殊値 \*ALL を使用して、ActiveX または Java Bean パーツに、リストに保管された使用可能なすべてのイベントを VARPG プログラムに送らせることができます。デフォルトでは、ActiveX または Java Bean パーツは、イベントを経路指定しません。

AddEvent(\*ALL) を出した後で新規の部分的イベント・リストをビルドするには、最初に RmvEvent 属性でリストを空にします。

**タイプ** スtring

**適用可能なパーツ**

|         |           |  |
|---------|-----------|--|
| ActiveX | Java Bean |  |
|---------|-----------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'AX1':'AddEvent')='EVT2'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'AX1'      SETATR  'EVT2'      'ADDEVENT'
```

## AddItemEnd

新規項目をリストの終わりに追加します。

サブファイル・パーツの場合には、この属性は、その妥当性検査タイプとして定義された値を持つサブファイル入力フィールドに適用されます。この属性を使用して、サブファイル入力フィールドの妥当性検査リストに値を追加します。最初に、**ColNumber** 属性を設定して、処理する列を識別します。

**タイプ** スtring

**適用可能なパーツ**

|           |          |         |
|-----------|----------|---------|
| 組み合わせボックス | リスト・ボックス | スピン・ボタン |
| サブファイル    |          |         |

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

リストに追加する項目、またはサブファイル入力フィールドの妥当性検査値

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'LB1':'AddItemEnd')=newitem
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      SETATR  'NEWITEM'  'ADDITEMEND'
```

## AddLineEnd

複数行編集の最終行の後に新しい行を追加します。

タイプ スtring

適用先

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

テキストの改行

フリー・フォームの例

入力フィールドからテキスト値を取得し、それを複数行編集の終わりに追加します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'MLE1':'AddLineEnd')=newtext
*
```

固定形式の例

複数行編集の終わりに行を追加します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C  'MLE1'      SETATR  'NEWTEXT'  'ADDLINEEND'
*
```

## AddLink

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

ターゲット・パーツをこのソース・パーツにリンクします。また、**AllowLink** 属性も設定された場合には、このパーツを変更した時にターゲット・パーツは **Link** イベントを生成します。

リンクできるパーツ・タイプのリストについては、次のテーブルを参照してください。

| 次の場合        | ソース・パーツ   | リンク・ターゲット・パーツの応答  |
|-------------|-----------|---|
| モードまたは位置の変更 | メディア      | メディア・パネルがメディア・パーツのモード変更に応答します。  |
| 選択          | チェック・ボックス | タイマー・パーツの使用可能／使用不能  |
| 変更された値      | 入力フィールド   | メディア: ファイル名が設定されます。<br>入力フィールド: テキスト値が更新されます。<br>イメージ: ファイル名が設定されます。<br>スライダー: 数値が更新されます。<br>スピン・ボタン: 数値が更新されます。<br>静的テキスト: テキスト値が更新されます。 |
| ファイル名の変更    | イメージ      | 静的テキスト: テキスト値が更新されます。   |
| 選択          | リスト・ボックス  | メディア: ファイル名が設定されます。<br>入力フィールド: テキスト値が更新されます。<br>イメージ: ファイル名が設定されます。<br>静的テキスト: テキスト値が更新されます。   |
| 制御の変更       | メディア・パネル  | メディア・パーツがメディア・パネル・パーツのモード変更に応答します。  |
| アームの変更      | スライダー     | 入力フィールド: テキスト値が更新されます。<br>スライダー: 数値が更新されます。<br>静的テキスト: テキスト値が更新されます。  |
| 目盛り         | タイマー      | 静的テキスト: テキスト値が更新されます。<br>タイマー・カウントがゼロに設定されます。   |

## タイプ スtring

### 適用可能なパーツ

|           |         |          |
|-----------|---------|----------|
| チェック・ボックス | 入力フィールド | イメージ     |
| リスト・ボックス  | メディア    | メディア・パネル |
| スライダー     | タイマー    |          |

### 命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | いいえ      |

**\*注:** GUI Designer プロパティ・ノートブックを使用して設定することができます。これは、「リンク」タブを使用し、「ターゲット・パーツ名」フィールドと「ターゲット親パーツ名」フィールドに埋め込みを行って実行してください。

### 使用可能な値

リンク先のパーツの名前。

パーツ名は次の形式でなければなりません。

```
'window|part'
```

この *window* はそのパーツが入っているウィンドウであり、*part* は実際のパーツ名です。

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CB1': 'AddLink')='WIN1|TIMER1'
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CB1'          SETATR      'WIN1|TIMER1' 'ADDLINK'
*
```



## AddMsgID

メッセージをメッセージ・サブファイル・パーツに追加します。

メッセージはメッセージ・サブファイル・パーツの既存メッセージの後に追加されます。

**タイプ** 数値

**適用可能なパーツ**

|              |  |  |
|--------------|--|--|
| メッセージ・サブファイル |  |  |
|--------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

メッセージ ID (msgid) であり、これは最大 4 桁の数値です。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MSG1': 'AddMsgID') = 1234
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MSG1'      SETATR  1234          'ADDMSGID'
```

## AddMsgText

メッセージをメッセージ・サブファイル・パーツに追加します。メッセージはストリングの形式で追加されます。

メッセージはメッセージ・サブファイル・パーツの既存メッセージの後に追加されます。

**タイプ** ストリング

**適用可能なパーツ**

|              |  |  |
|--------------|--|--|
| メッセージ・サブファイル |  |  |
|--------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

メッセージ・テキスト・ストリング

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'MSG1':'AddMsgText')='Help!'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'MSG1'      SETATR      'Help!'      'ADDMSGTEXT'
*
```

## AddOffset

指定されたテキストを、**CharOffset** 属性によって指定された文字オフセットの複数行の編集に挿入します。

タイプ スtring

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

複数行編集に挿入するString

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C                               EVAL    %setatr('win01': 'MLE1': 'CharOffset') = 10
C                               EVAL    %setatr('win01': 'MLE1': 'AddOffset') = 'XXX'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C   'MLE1'      SETATR   10      'CHAROFFSET'
C   'MLE1'      SETATR   'XXX'   'ADDOFFSET'
*
```

## AddRcd

新しいレコードをコンテナに追加します。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

この属性に渡される値は、それぞれがスペースで区切られた、次のパラメーターから構成されるStringでなければなりません。

**レコード構造**

**ID** レコードを識別するために使用される数値。コンテナに入れたい最大のレコード数を指定するためにこの値を宣言します。

**注:** VisualAge RPG でユーザー固有な ID を生成させるためには、**GetNewID** 属性を使用することができます。

**Text** コンテナがアイコン・ビューまたはツリー・ビューにある時に、レコード・アイコンの横に表示されるテキスト。

**FileName**

アイコンのファイル名。

**ParentID**

このレコードに対する親レコードの ID。このレコードに親レコードがない場合には、このパラメーターは 0 に設定されます。

**field\_data**

この値は任意指定です。指定された場合には、レコード内の対応するフィールドにデータが入れます。レコード内の各フィールドにデータを指定しなければなりません。GU Designer にあるコンテナ・パーツの各レコード用のフィールドの数値が、すでに定義されています。したがって、その数だけを指定します。

**注:** フィールドのデータがまだ使用可能でない場合には、フィールドのプレースホルダーとして下線 ( \_ ) 文字を使用してください。フィールドに組み込みブランクがある場合には、レコードを追加する前に、下線文字と置き換える必要があります。組み込みブランクは、追加済みレコードのフィールド・データ中に現れることとなります。

フィールド・データは、**SetRcdFld** 属性を使用してレコードを追加した後で更新することができます。

### フリー・フォームの例

```

*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D NewRecord
D RecordID          1      6
D FileName          8      64
D Parent            1      6   INZ(0)
D Text              8      20   INZ('Next record')
*
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          Eval      RecordID=%getatr('win01': 'CN1': 'GetNewID')
C          Eval      FileName = 'D:\VRPG\NEW.ICO'
C          Eval      %setatr('win01': 'CN1': 'AddRcd') = NewRecord
*

```

### 固定形式の例

```

*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CN1'          SETATR      NewRecord      'ADDRCD'
*

```

## AddSrcEvt

新規ソース・イベントを、コンポーネント参照パーツによってモニターされるイベントの現行リストに追加します。この属性によって、1つのコンポーネント参照パーツで複数のイベントをモニターできます。リストからソース・イベントを除去するには、**RmvSrcEvt** 属性を使用します。

追加するソース・イベントを識別するには、コンポーネント、ウィンドウ、パーツ、およびイベント名を全部一緒に指定してください。このstringの形式は次の通りです。

```
'COMPONENTNAME | WINDOWNAME | PARTNAME | EVENTNAME'
```

コンパイラーは妥当性検査を行わないので、それぞれのソース・イベントは正確に識別する必要があります。

**タイプ** string

**適用可能なパーツ**

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D Event          S          100
*
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                  EVAL      Event='compb|win02|ef02|change'
C                  EVAL      %setatr('win01': 'CRP1':
C                  'AddSrcEvt')=Event
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D Event          S          100
*
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                  EVAL      Event='compb|win02|ef02|change'
C 'CRP1'           SETATR    Event      'ADDSRCEVT'
*
```

## Alarm

スピーカーからビープ音を鳴らします。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

この属性の値は、1 に設定しなければなりません。

フリー・フォームの例

次のように、スピーカーをビープ音に設定します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('*Component':'*Component':'Alarm')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C    '*COMPONENT' SETATR  1          'ALARM'
*
```

## Alignment

この属性は、テキストおよびラベルがパーツ内にどのように配置されるかを決定します。

**タイプ** 数値

**適用可能なパーツ**

|          |          |        |
|----------|----------|--------|
| 入力フィールド* | スピン・ボタン* | 静的テキスト |
|----------|----------|--------|

\* **制約事項:** この属性は、いったん設定されると、実行時に変更できません。

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\***注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

**Alignment** は、次のいずれかとすることができます。

- 1 テキストが左寄せされます。
- 2 テキストが右寄せされます。
- 3 テキストが中央にそろえられます。



## フリー・フォームの例

パーツのテキストが左寄せされる場合には、その右寄せも行なわれます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C                               EVAL      align = %getatr('win01': 'ST1': 'Alignment')
*
C      align                    IFEQ      1
C                               EVAL      %setatr('win01': 'ST1': 'Alignment') = 2
*
C                               ELSE
C                               EVAL      %setatr('win01': 'ST1': 'Alignment') = 1
C                               ENDF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C      'ST1'                    GETATR    'ALIGNMENT'  ALIGN      1 0
C      ALIGN                    IFEQ      1
C      'ST1'                    SETATR    2          'ALIGNMENT'
*
C                               ELSE
C      'ST1'                    SETATR    1          'ALIGNMENT'
*
*
C                               ENDF
*
```

## AllowChg

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

列データをサポートされない VARPG データ・タイプとして識別します。この列データに更新は実行されません。

**FetchNext、FetchPrior、InsertRow、または UpdateRow** を実行すると、データはプログラム・フィールド・バッファに、あるいはプログラム・フィールド・バッファから対応するテーブル列に移動されます。列がサポートされない VARPG データ・タイプを持っている場合には、**AllowChg** 属性を 0 に設定してください。デフォルト値は 1 です。(サポートされるデータ・タイプのリストについては、102 ページの『BufferType』を参照してください。)

**AllowChg** 値を設定する前に、Column 属性を必要な列に設定してください。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**AllowChg** は、次のいずれかとすることができます。

**0** サポートされない VARPG データ・タイプ。

**1** サポートされる VARPG データ・タイプ。これがデフォルトです。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C                               EVAL      %setatr('win01': 'ODBC1': 'AllowChg') = 0
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'ODBC1'      SETATR  0          'AllowChg'
*
```

## AllowEdit

サブファイル入力フィールドを編集できるかどうかを決定します。編集状況は、サブファイル、レコード、列、またはセルのレベルで指定することができます。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**AllowEdit** は、次のいずれかとすることができます。

- 1 サブファイル全体の編集を防止します。
- 2 **Index** 属性によって指定されたレコードの編集を防止します。
- 3 **ColNumber** 属性によって指定された列の編集を防止します。
- 4 **ColNumber** および **Index** 属性によって指定されたセルの編集を防止します。
- 11 サブファイル全体の編集を許可します。
- 12 **Index** 属性によって指定されたレコードの編集を許可します。
- 13 **ColNumber** 属性によって指定された列の編集を許可します。
- 14 **ColNumber** および **Index** 属性によって指定されたセルの編集を許可します。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'SFL1': 'Index') = 5
C          EVAL      %setatr('win01': 'SFL1': 'AllowEdit') = 12
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SFL1'      SETATR  5          'INDEX'
C  'SFL1'      SETATR  12         'ALLOWEDIT'
*
```

## AllowLink

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

ソース・パーツの変更時に **AddLink** または **DDEAddLink**属性によって指定されたターゲット・パーツが **Link** イベントを生成するかどうかは、この属性によって判別されます。

**タイプ** 数値

**適用可能なパーツ**

|           |         |          |
|-----------|---------|----------|
| チェック・ボックス | 入力フィールド | イメージ     |
| リスト・ボックス  | メディア    | メディア・パネル |
| スライダー     | タイマー    |          |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。リンク・タブを使用し、「**ターゲット通知可能**」を選択することによって、これを行なってください。

**使用可能な値**

**AllowLink** は、次のいずれかとすることができます。

- 0** **Link** イベントが通知されません
- 1** **Link** イベントが通知されます

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      allow = %getatr('win01': 'EF1': 'AllowLink')
*
C  allow      IFEQ      1
C          EVAL      %setatr('win01': 'EF1': 'AllowLink') = 0
*
C          ELSE
C          EVAL      %setatr('win01': 'EF1': 'AllowLink') = 1
C          ENDIF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'EF01'    GETATR    'ALLOWLINK'  ALLOW      1 0
*
C  ALLOW      IFEQ      1
C  'EF01'    SETATR    0          'ALLOWLINK'
*
C          ELSE
C  'EF01'    SETATR    1          'ALLOWLINK'
*
C          ENDIF
*
```

## AppData

他のコンポーネントが参照できるグローバル・データを含みます。この属性によって、これを使用しない場合に宣言する必要があるパラメーターを渡すための代替方法が提供されます。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('*Component':'*Component':'AppData')
C          = NData
C          EVAL    TmpFld=%getatr('*Component':'*Component':
C          'AppData')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C '*COMPONENT' SETATR  NData    'APPDATA'
C '*COMPONENT' GETATR  'APPDATA'  TmpFld
*
```

## AppName

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

これは、DDE 会話で DDE クライアント・パーツが接続する先のサーバー・アプリケーションの名前です。サーバー名はサーバー・アプリケーションによって決まります。そのアプリケーションで使用するための **AppName** については、そのアプリケーションの資料を参照してください。

**AppName** の設定では、DDE 会話は開始されません。DDE 会話の開始については、182 ページの『DDEMode』を参照してください。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

サーバー・アプリケーションの名前

サーバー・プログラムが VisualAge RPG アプリケーションである場合には、この属性はその拡張子が組み込まれたアプリケーションの名前となります。たとえば、VisualAge RPGアプリケーションがサーバーと呼ばれた場合には、**AppName** 属性は次のとおり設定されます。

```
EVAL %setatr('WIN':'DDE1':'AppName') = 'SERVER.EXE'
```

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL          %setatr('win01': 'DDE1': 'AppName') = 'APP01'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'DDE1'          SETATR 'APP01'          'APPNAME'
*
```



## Arrange

アイコン・ビューにおけるコンテナ内の項目を、各アイコンが行ごとに並ぶよう配置します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

この属性の値は 1 でなければなりません。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C          Eval      %setatr('window': 'CN1': 'Arrange') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C  'CN1'      SETATR  1          'ARRANGE'
```

## AttrValue

コンポーネント参照パーツによって参照される属性の値。最初に、コンポーネント参照パーツを設定して、コンポーネント、ウィンドウ、および属性のパーツ名を識別します。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'COMP1': 'AttrValue')='Test'
*
```

固定形式の例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'COMP1'    SETATR  'Test'      'ATTRVALUE'
```

## AudioMode

パーツについての操作モードを設定します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| メディア |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**AudioMode** は、次のいずれかとすることができます。

- 1 休止
- 2 再生
- 3 レコード
- 4 停止

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'AUDIO01': 'AudioMode') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AUDIO01'  SETATR  1          'AUDIOMODE'
```

## AutoInc

**DataValue** 属性が設定されるたびに、この属性によって現在の **DataPoint** 値が増分されます。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**AutoInc** では、次の 1 つが戻されます。

**0** **DataPoint** は自動的に増やされません

**1** **DataPoint** は自動的に増やされます

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      auto = %getatr('win01': 'GP1': 'AutoInc')
*
C      auto      IFEQ      1
C          EVAL      %setatr('win01': 'GP1': 'AutoInc') = 0
*
C          ELSE
C          EVAL      %setatr('win01': 'GP1': 'AutoInc') = 1
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'GP01'      GETATR      'AUTOINC'      Auto      1 0
*
C      AUTO      IFEQ      1
C      'GP01'      SETATR      0      'AUTOINC'
*
C          ELSE
C      'GP01'      SETATR      1      'AUTOINC'
*
C          ENDIF
*
```

---

## AutoScroll

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

さらに多くのテキストを入力する時、あるいは入力フィールドに表示できる情報より多くの情報が使用可能な時に、ユーザーは入力フィールド中のデータをスクロールすることができます。

**タイプ** 数値

**適用可能なパーツ**

|           |         |  |
|-----------|---------|--|
| 組み合わせボックス | 入力フィールド |  |
|-----------|---------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**AutoScroll** は、次のいずれかとすることができます。

- 0** 自動スクロールは使用できません。
- 1** 自動スクロールを使用することができます。

## フリー・フォームの例

入力フィールドの **AutoScroll** 属性を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      rc = %getatr('win01': 'EF1': 'AutoScroll')
*
C   rc      IFEQ      1
C          EVAL      %setatr('win01': 'EF1': 'AutoScroll') = 0
*
C          ELSE
C          EVAL      %setatr('win01': 'EF1': 'AutoScroll') = 1
C          ENDIF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'EF1'   GETATR   'AUTOSCROLL' rc          1 0
*
C   rc      IFEQ      1
C   'EF1'   SETATR   0          'AUTOSCROLL'
*
C          ELSE
C   'EF1'   SETATR   1          'AUTOSCROLL'
C          ENDIF
*
```

## AutoSelect

この属性は、入力フィールドがフォーカスを得たとき、またはサブファイル・フィールドが編集用にオープンされたときに、フィールド内のテキストが強調表示されるかどうかを示します。サブファイル・パーツの場合には、最初に **ColNumber** 属性を設定して、処理する列を識別します。

タイプ 数値

適用可能なパーツ

|         |        |  |
|---------|--------|--|
| 入力フィールド | サブファイル |  |
|---------|--------|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**AutoSelect** は次のいずれかの値とすることができます。

- 0 入力フィールドの値は強調表示されません
- 1 入力フィールドの値は強調表示されます

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      highlight=%getatr('win01':'EF1':'AutoSelect')
*
C  highlight  IFEQ      0
C          EVAL      %setatr('win01':'EF1':'AutoSelect')=1
C          ELSE
C          EVAL      %setatr('win01':'EF1':'AutoSelect')=0
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'EF1'      GETATR   'AUTOSELECT'  highlight      1 0
*
C  'EF1'      SETATR   1              'AUTOSELECT'
```

## BackColor

パーツの背景カラーを決定する数値。

**BackMix** 属性を使用して背景カラーの属性を設定した場合には、その背景カラーの混合値に最も近いカラー・インデックスが戻されます。

**タイプ** 数値

**適用可能なパーツ**

|          |           |           |
|----------|-----------|-----------|
| キャンバス    | チェック・ボックス | 組み合わせボックス |
| コンテナ     | 入力フィールド   | イメージ      |
| リスト・ボックス | メディア・パネル  | 複数行編集     |
| ノートブック   | プッシュボタン   | ラジオ・ボタン   |
| スライダー    | スピン・ボタン   | 静的テキスト    |
| サブファイル   |           |           |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

背景カラーは次の 1 つでなければなりません。

- 1 デフォルト(Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色



- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

注: デフォルト・カラー (-1) を SETATR に使用することはできません。

#### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      color = %getatr('win01': 'EF1': 'BackColor')
*
C  color    IFEQ      *Red
C          EVAL      %setatr('win01': 'EF1': 'BackColor') = *Blue
C          ENDIF
*
```

#### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'EF1'    SETATR   *BLUE      'BACKCOLOR'
```

## BackMix

背景カラーの混合を照会または設定します。

**タイプ** ストリング

**適用可能なパーツ**

|          |           |           |
|----------|-----------|-----------|
| キャンバス    | チェック・ボックス | 組み合わせボックス |
| コンテナ     | 入力フィールド   | イメージ      |
| リスト・ボックス | メディア・パネル  | 複数行編集     |
| ノートブック   | プッシュボタン   | ラジオ・ボタン   |
| スライダー    | スピン・ボタン   | 静的テキスト    |
| サブファイル   |           |           |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

背景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL    color=%getatr('win01':'EF1':'BackMix')
*
C  color    IFNE    '10:23:200'
C          EVAL    %setatr('win01':'EF1':'BackMix')='10:23:200'
C          ENDIF
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'EF1'    SETATR  '10:23:200'  'BACKMIX'
*
```

## BarLabel

現在のデータ項目のラベル (**DataPoint** および **DataGroup** によって指定された) を示します。

タイプ スtring

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'GPH1': 'BarLabel') = 'BAR01'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GPH1'      SETATR  'BAR01'      'BARLABEL'
```

## Bass

注: この属性は Windows にのみ適用されます。

メディア・パーツのバス・レベルを設定します

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| メディア |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

バスは 0 - 100 の値とすることができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'AUD1': 'Bass') = 37
*
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'AUD1'      SETATR  37          'BASS'
```

## BindPart

**制約事項:** この属性は、Java アプリケーションにのみ適用されます。

データ列にバインドされるかまたはバインドされているパーツを識別します。Java 環境では、入力フィールドおよび静的テキスト・パーツの場合にのみ、バインドがサポートされます。

**タイプ** スtring

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

バインドされる親/パーツ名の形式は次の通りです。

```
'FRAMEWINDOW|PARTNAME'
```

ここで FrameWindow は、PartName を所有するウィンドウの名前です。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'WIN1': 'BindPart')
C          = 'WIN1|ENT2'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'WIN1'      SETATR  'WIN1|ENT2'  'BINDPART'
*
```

## BlankChar

文字列中の空白文字を表すことになる文字を指定します。

コンテナー・パーツは、レコードの要素の区切り文字として空白を使用します。追加する要素 (アイコンのテキストなど) に空白が含まれている場合には、**BlankChar** 属性を空白以外の文字に設定してください。

デフォルトの空白文字は下線 ( \_ ) です。

**タイプ** 文字列

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| コンテナー |  |  |
|-------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'CTR1':'BlankChar')='&'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C      'CTR1'      SETATR  '&'      'BLANKCHAR'
```

## Border

パーツが枠を持つかどうかを決定します。

タイプ 数値

適用可能なパーツ

|       |      |          |
|-------|------|----------|
| カレンダー | イメージ | プッシュボタン* |
|-------|------|----------|

\* 制約事項: Java アプリケーションでは、この属性を実行時に設定および取り出すことができるのは、プッシュボタン・パーツの場合だけです。

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**Border** は、次のいずれかとすることができます。

- 0 パーツには枠がありません
- 1 パーツに枠があります

## フリー・フォームの例

次のように、パーツの **Border** 属性を切り替えます。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      brdval = %getatr('win01': 'IMG1': 'Border')
C   brdval  IFEQ      0
C          EVAL      %setatr('win01': 'IMG1': 'Border') = 1
*
C          ELSE
C          EVAL      %setatr('win01': 'IMG1': 'Border') = 0
C          ENDIF
*
```

## 固定形式の例

次のように、プッシュボタンの現行枠の設定を切り替えます。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'IMG1'  GETATR   'BORDER'   brdval      1 0
C   brdval  IFEQ     1
C   'IMG1'  SETATR   0           'BORDER'
*
C          ELSE
C   'IMG1'  SETATR   1           'BORDER'
C          ENDIF
```



## Bottom

パーツまたはオブジェクトの画面の下の座標をピクセル数で指定します。

**注:** 原点は左上隅にあり、ゼロで始まります。

**タイプ** 数値

**適用可能なパーツ**

|               |                |                    |
|---------------|----------------|--------------------|
| ActiveX       | カレンダー          | キャンバス*             |
| チェック・ボックス     | 組み合わせボックス      | コンポーネント参照          |
| コンテナ          | DDE クライアント     | 入力フィールド            |
| グラフ           | グラフィック・プッシュボタン | グループ・ボックス          |
| 水平方向のスクロール・バー | イメージ           | Java Bean          |
| リスト・ボックス      | メディア           | メディア・パネル           |
| 複数行編集         | ノートブック         | ODBC/JDBC インターフェース |
| 外枠            | 進行状況バー         | プッシュボタン            |
| ラジオ・ボタン       | スライダー          | スピン・ボタン            |
| 静的テキスト        | サブファイル         | タイマー               |
| 縦方向のスクロール・バー  | ウィンドウ          |                    |

\* **制約事項:** キャンバス・パーツの場合には、この属性は実行時にのみ取り出すことができます。

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

ピクセルの数

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      bot = %getatr('win01': 'Image1': 'Bottom')
C  bot          SUB      100          newbot
C          EVAL      %setatr('win01': 'Image1': 'Bottom') = newbot
*
```

## 固定形式の例

そのウィンドウ内のイメージ 200 ピクセルを移動します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'IMAGE1'  GETATR  'BOTTOM'  OldBot      4 0
C  OldBot    ADD      200        NewBot      4 0
C  'IMAGE'   SETATR  NewBot      'BOTTOM'
*
```

## BufferDec

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

バッファ一列の小数点以下の桁数を指定します。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'BufferDec')=4
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'ODBC1'   SETATR   4          'BufferDec'
*
```

## BufferLen

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

列データに使用可能なプログラム・フィールドの長さを指定します。ODBC/JDBC インターフェース・パーツは、データを列から (または列に) 移動する時に、設定してプログラム・フィールドに入れるデータの長さを知る必要があります。

この属性を使用する前に、**Column** 属性を必要な列に設定してください。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'BufferLen')=20
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'ODBC1'   SETATR   20          'BufferLen'
```

## BufferPtr

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

プログラム・フィールドを指すポインター値を指定します。ODBC/JDBC インターフェイス・パーツは、この値を使って、データをデータベースからプログラム・フィールドに移動します。

この属性を使用する前に、**Column** 属性を必要な列に設定してください。

プログラム・フィールドのアドレスを取得するには、%ADDR 組み込みを次のように使用します。

```
D FieldPtr          S          * INZ(%Addr(Field))
```

**注:** この属性は Java 環境では適用できません。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェイス |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**注:** SETATR 演算命令の場合には、BufferPtr 属性は **固定形式構文でのみ** コーディングすることができます。

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1++++++0opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C 'ODBC1' SETATR FieldPtr 'BufferPtr'
*
```

## BufferType

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

**BufferPtr** 属性によって参照される VisualAge RPG プログラム・フィールドのデータ・タイプを示します。 ODBC/JDBC インターフェース・パーツは、プログラム・フィールドとテーブル列の間でデータを移動するときに、**BufferType** 属性を使用して、正しいデータ変換を実行します。フィールド・タイプが適切かどうかの検査は行われないので、この属性を正しく設定することが重要です。

**Column** 属性は、**BufferType**属性を使用する前に設定してください。プログラム・フィールドがインターフェース上のパーツと関連づけられている場合には、**DataType** 属性を使用してバッファ・タイプを取得することができます。

列に ODBC/JDBC インターフェース・パーツでサポートされないデータ・タイプが入っている場合には、その列の **AllowChg** 属性を 0 に設定してください。ODBC/JDBC インターフェース・パーツは、プログラム・フィールドとその列の間のデータ移動を行なわなくなります。データはもとのままです。

対応する VARPG データ・タイプをサポートされている SQL データ・タイプに設定するためには、次の図表を使用してください。図表にリストされている

**BufferLen** および **BufferDec** 属性のみを指定してください。

| SQL データ・タイプ | VARPG データ・タイプ | プログラム・フィールドの長さを指定 (BufferLen を使用) します | バッファ桁の小数点以下の桁数を指定 (BufferDec を使用) します |
|-------------|---------------|---------------------------------------|---------------------------------------|
| 文字          | CHAR          | X                                     |                                       |
| 小数部分        | Zoned         | X                                     |                                       |
| 整数          | Zoned         | X                                     |                                       |
| 短整数         | Zoned         | X                                     |                                       |
| 倍精度         | 8F            |                                       |                                       |
| 倍精度         | Zoned         | X                                     | X                                     |
| 浮動          | 4F            |                                       |                                       |
| 浮動          | Zoned         | X                                     | X                                     |
| 実数          | 4F            |                                       |                                       |
| 実数          | Zoned         | X                                     | X                                     |

文字、10 進数、整数、または短整数データ・タイプの場合には、**BufferLen** 属性のみを指定します。

倍精度、浮動、および実データ・タイプは、VARPG では浮動 (F) または Zoned として定義できることに注意してください。これらを Zoned として定義した場合には、VARPG ランタイムは、列間でデータを移動するときに、**BufferDec** 属性で指定された小数点以下の桁数のみを使用します。この方式では、データ・ソースが **BufferDec** 属性で指定された小数点以下の桁数より大きい場合に、精度が失われることがあります。これらのフィールドを浮動 (F) として定義する場合には、**BufferLen** または **BufferDec** 属性を指定しないようにしてください。

**タイプ** 数値

## 適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

## 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

## 使用可能な値

**BufferType** は、次のいずれかとすることができます。

**0** 数値データを示します。

サポートされる SQL データ・タイプには次のものがあります。

- 小数部分
- 整数
- 短整数
- 浮動
- 実数
- 倍精度

**1** 文字データを示します。

サポートされる SQL データ・タイプには次のものがあります。

- 文字
- 可変長文字
- 日付
- 時刻
- タイム・スタンプ

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'BufferType')=1
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C      'ODBC1'      SETATR      1          'BUFFERTYPE'
*
```



## Button

「OK」または「キャンセル」ボタンが押されたかどうか、あるいはファイル・ダイアログがクローズされたかどうかを照会します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

使用可能な値

**Button** 状況は、次のいずれかとすることができます。

**0** ファイル・ダイアログはクローズされていません

**1** 「OK」ボタンが押されました

**2** 「キャンセル」ボタンが押されました

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      But=%getatr('*Component':'*Component':'Button')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      '*COMPONENT' GETATR      'Button'      But
*
```

## ButtonIdx

ナビゲーション・ボタンの索引を設定して取得します。**EnableBtn** 属性を活動化する前に、索引を設定する必要があります。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

索引は次のものに設定することができます。

- 1 pagetop ボタン
- 2 pageup ボタン
- 3 prevrec ボタン
- 4 nextrec ボタン
- 5 pagedown ボタン
- 6 pageend ボタン

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+t+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'SFI1': 'ButtonIdx') = 4
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+t+++++Len++D+HiLoEq----
*
C 'SFI1'      SETATR  4          'BUTTONIDX'
*
```

# Buttons

ナビゲーション・バーを拡大または縮小します。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**Buttons** は、次のいずれかとすることができます。

- 0** ボタンを表示しません
- 1** ボタンを表示します

フリー・フォームの例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      btnval = %getatr('win01': 'SF1': 'Buttons')
C  btnval  IFEQ      0
C          EVAL      %setatr('win01': 'SF1': 'Buttons') = 1
*
C          ELSE
C          EVAL      %setatr('win01': 'SF1': 'Buttons') = 0
C          ENDIF
*
```

固定形式の例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SF1'    GETATR   'BUTTONS'  btnval      1 0
C  btnval  IFEQ     1
C  'SF1'    SETATR   0          'BUTTONS'
*
C          ELSE
C  'SF1'    SETATR   1          'BUTTONS'
C          ENDIF
```

## ButtonTip

**ButtonIdx** によって設定された対応するナビゲーション・ボタンのツール・ヒント・テキストを設定します。

テキストは、ストリングまたはメッセージ (\*MSGnnnn) として設定することができます。

タイプ ストリング

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

ツール・ヒント・ストリング

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('win01': 'SF11': 'ButtonTip') = string
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C    'SF11'    SETATR    STRING    'BUTTONTIP'
```

## CanUndo

最後のアクションのやり直しが可能かどうかを示します。1 が戻された場合には、最後のアクションはやり直しが可能です。0 は、不可能であることを示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      Chng=%getatr('win01':'ML1':'CanUndo')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'ML1'      GETATR  'CANUNDO'    CHNG
*
```

## CapsLock

入力フィールドまたはサブファイル入力フィールドに入力されているテキストを大文字に変換するかどうかを決定します。

**注:** サブファイル入力フィールドの **CapsLock** を取り出しまたは設定するためには、**ColNumber** 属性が正しい列番号に設定されていなければなりません。

タイプ 数値

適用可能なパーツ

|         |                 |  |
|---------|-----------------|--|
| 入力フィールド | 48 ページの『サブファイル』 |  |
|---------|-----------------|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**CapsLock** は、次のいずれかとすることができます。

- 0** テキストは大文字に変換されません。
- 1** テキストが大文字に変換されます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'EF1': 'CapsLock') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'EF1'      SETATR  1          'CAPSLOCK'
```

## Case

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

検索比較が大文字・小文字を区別するかどうかを決定します。

**タイプ** 数値

**適用可能なパーツ**

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**Case** は、次のいずれかとすることができます。

**0** 比較は大文字小文字の区別がありません。

**1** 比較は大文字小文字の区別があります。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C                               EVAL      %setatr('win01': 'CB1': 'Case') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C 'CB1'          SETATR  1          'CASE'
```

## CellBGClr

サブファイル・セルの背景カラーを決定する数値。そのカラーを設定するセルを識別するためには、 **Index** および **ColNumber** 属性を使用しなければなりません。

**CellBGMix** 属性を使用して背景カラーを設定した場合には、戻される背景カラーが混合カラーに最も近いカラーです。

**タイプ** 数値

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

このカラーは次の 1 つでなければなりません。

- 1 デフォルト(Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色



- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      CBcolor = %getatr('win01': 'SFL1': 'CellBGClr')
*
C          EVAL      %setatr('win01': 'SFL1': 'ColNumber') = 1
C          EVAL      %setatr('win01': 'SFL1': 'Index') = 4
C          EVAL      %setatr('win01': 'SFL1': 'CellBGClr') = 5
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SFL1'    GETATR   'CELLBGCLR'  'CBcolor'      2 0
*
C  'SFL1'    SETATR   1              'COLNUMBER'
C  'SFL1'    SETATR   4              'INDEX'
C  'SFL1'    SETATR   5              'CELLBGCLR'
*
```

## CellBGMix

サブファイル・セルの混合背景カラーを照会または設定します。そのカラー混合を照会または設定するセルを識別するためには、**Index** および **ColNumber** 属性を使用しなければなりません。

タイプ スtring

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

背景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      CBc1rMx = %getatr('win01': 'SFL1': 'CellBGMix')
*
C  CBc1rMx  IFNE      '10:23:200'
C          EVAL      %setatr('win01': 'SFL1': 'CellBGMix') = '10:23:200'
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SFL1'   SETATR   '10:23:200'  'CELLBG MIX'
```

## CellFGClr

サブファイル・セルの前景カラーを決定する数値。そのカラーを設定するセルを識別するためには、 **Index** および **ColNumber** 属性を使用しなければなりません。

**CellFGMix** 属性を使用して前景カラーを設定した場合には、戻される前景カラーが混合カラーに最も近いカラーです。

**タイプ** 数値

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

カラーは次の 1 つでなければなりません。

- 1 デフォルト(Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      CFcolor = %getatr('win01': 'SFL1': 'CellFGClr')
*
C          EVAL      %setatr('win01': 'SFL1': 'ColNumber') = 1
C          EVAL      %setatr('win01': 'SFL1': 'Index') = 4
C          EVAL      %setatr('win01': 'SFL1': 'CellFGClr') = 5
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SFL1'      GETATR  'CELLFGCLR'  'CFcolor'      2 0
*
C  'SFL1'      SETATR  1            'COLNUMBER'
C  'SFL1'      SETATR  4            'INDEX'
C  'SFL1'      SETATR  5            'CELLFGCLR'
*
```

## CellFGMix

サブファイル・セルの前景カラーを照会または設定します。そのカラー混合を照会または設定するセルを識別するためには、**Index** および **ColNumber** 属性を使用しなければなりません。

タイプ スtring

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

前景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      CFc1rMx=%getatr('win01':'SFL1':'CellFGMix')
*
C   CFc1rMx   IFNE    '10:23:200'
C          EVAL      %setatr('win01':'SFL1':'CellFGMix')='10:23:200'
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'SFL1'   SETATR  '10:23:200'  'CELLFGMIX'
*
```

## Center

これが 1 に設定されると、ウィンドウは、初めて表示されて時に、中央に配置されます。必要によって、ユーザーは後からこれを移動することができます。

ウィンドウを中央そろえた後で、左上隅の x または y 座標が負の場合には、ウィンドウのタイトル・バーが常に表示されるように、その x y 座標は (0,0) に設定されます。

**タイプ** 数値

**適用可能なパーツ**

|       |              |  |
|-------|--------------|--|
| ウィンドウ | キャンバス付きウィンドウ |  |
|-------|--------------|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | いいえ      |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

この属性は 1 にしか設定できません。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('FW1': 'FW1': 'Center') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'FW1'      SETATR  1          'CENTER'
```

## CharData

特定の列の文字テキストを戻りレコード・セットで戻します。この属性を使用する前に **Column** 属性を設定してください。

タイプ スtring

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      ColData=%getatr('win01':'ODBC1':'CharData')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'ODBC1'      GETATR  'CHARDATA'  COLDATA
*
```

## CharOffset

複数行編集の文字オフセット値を判別し、ここに **AddOffset** 属性を設定した時にテキストが挿入されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

文字オフセット

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'CharOffset') = 10
*
C          EVAL      value = %getatr('win01': 'MLE1': 'CharOffset')
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MLE1'      SETATR  10          'CHAROFFSET'
C  'MLE1'      SETATR  'text'     'ADDOFFSET'
```



## Checked

パーツを検査するかどうかを決定します。

タイプ 数値

適用可能なパーツ

|           |        |         |
|-----------|--------|---------|
| チェック・ボックス | メニュー項目 | ラジオ・ボタン |
|-----------|--------|---------|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。チェック・ボックスおよびラジオ・ボタンのパーツでは、「**Selected**」スタイルを選択してこれを実行してください。メニュー項目のパーツでは、「**Checked**」スタイルを選択してこれを実行してください。

使用可能な値

パーツの **Checked** 状態は、次のいずれかとすることができます。

**0** パーツは検査されません。

**1** パーツは検査されます。

フリー・フォームの例

チェック・ボックスのチェック済み状態をラジオ・ボタンと同じに設定します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      ischecked = %getatr('win01': 'RB1': 'Checked')
*
C          EVAL      %setatr('win01': 'CB1': 'Checked') = ischecked
*
```

固定形式の例

チェック・ボックスのチェック済み状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CB1'      GETATR  'CHECKED'  ischecked
*
C  ischecked  IFEQ    1
C  'CB1'      SETATR  0          'CHECKED'
*
C          ELSE
C  'CB1'      SETATR  1          'CHECKED'
C          ENDIF
*
```

## ChildCount

**RecordId** によって指定される親の直接の子の数を戻します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| コンテナー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      children = %getatr('win01': 'CTR1': 'ChildCount')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CTR1'      GETATR  'CHILDCOUNT' children
*
```

## ChildList

コンテナ・パーツに入っているレコードについて、その直系の子のブランク区切りリストを戻します。**RecordId** 属性を使って、どのレコードの子リストが望みであるかを指定してください。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      list = %getatr('win01': 'CTR1': 'ChildList')
*
```

固定形式の例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'CTR1'      GETATR  'CHILDLIST' list
*
```

## ClearAll

**ClearAll** が設定されると、カレンダーからすべてのユーザー・テキストが除去されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**ClearAll** の有効な値は 1 だけです

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'ClearAll') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'CAL1'          SETATR      1          'CLEARALL'
*
```

## ClearDate

**ClearDate** は、設定されると、**DayIdx/MonthIdx/YearIdx** によって指示された日付に設定されたすべてのユーザー・テキストを除去します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**ClearDate** の有効な値は 1 だけです

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'ClearDate') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C 'CAL1'          SETATR      1          'CLEARDATE'
```

## ClearMonth

**ClearMonth** は、設定されると、**MonthIdx/YearIdx** によって指示された月に設定されたすべてのユーザー・テキストを除去します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**ClearMonth** の有効な値は 1 だけです

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'ClearMonth') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'CAL1'          SETATR      1          'CLEARMONTH'
```

## ClearYear

**ClearYear** は、設定されると、**YearIdx** によって指示された年に設定されたすべてのユーザー・テキストを除去します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**ClearYear** の有効な値は 1 だけです

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'ClearYear') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C 'CAL1'      SETATR  1          'CLEARYEAR'
```

## Clipboard

クリップボード間でテキストをコピーします。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

クリップボードの目次を表示します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('*Component':'*Component':'Clipboard')=text
*
C          EVAL      text=%getatr('*Component':'*Component':'Clipboard')
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      '*COMPONENT' SETATR  text      'CLIPBOARD'
*
C      '*COMPONENT' GETATR  'CLIPBOARD' text
```



## ColBGClr

サブファイル列の背景カラーを決定する数値。**ColNumber** 属性を使って、カラーが設定される列を識別する必要があります。

**ColBGMix** 属性を使用して背景カラーを設定した場合には、戻される背景カラーが混合カラーに最も近いカラーです。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

このカラーは次の 1 つでなければなりません。

- 1 デフォルト (Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      CBcolor = %getatr('win01': 'SFL1': 'COLBGCLR')
*
C          EVAL      %setatr('win01': 'SFL1': 'COLNUMBER') = 1
C          EVAL      %setatr('win01': 'SFL1': 'COLBGCLR') = 5
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'SFL1'      GETATR  'COLBGCLR'  'CBcolor'      2 0
*
C 'SFL1'      SETATR  1          'COLNUMBER'
C 'SFL1'      SETATR  5          'COLBGCLR'
*
```

## ColBGMix

サブファイル列の背景カラーの混合を照会または設定します。**ColNumber** 属性を使って、カラー混合が照会または設定される列を識別する必要があります。

**タイプ** スtring

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

背景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      CBclrMx = %getatr('win01': 'SFL1': 'ColBGMix')
*
C  CBclrMx  IFNE     '10:23:200'
C          EVAL      %setatr('win01': 'SFL1': 'ColBGMix') = '10:23:200'
C          ENDIF
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'SFL1'   SETATR   '10:23:200'  'COLBGMIX'
*
```

## CoIFGClr

サブファイル列の前景カラーを決定する数値。**ColNumber** 属性を使って、カラーが設定される列を識別する必要があります。

**CoIFGMix** 属性を使用して前景カラーを設定した場合には、戻される前景カラーが混合カラーに最も近いカラーです。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

このカラーは次の 1 つでなければなりません。

- 1 デフォルト (Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      CFcolor = %getatr('win01': 'SFL1': 'COLFGCLR')
*
C          EVAL      %setatr('win01': 'SFL1': 'COLNUMBER') = 1
C          EVAL      %setatr('win01': 'SFL1': 'COLFGCLR') = 5
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'SFL1'    GETATR   'COLFGCLR'  'CFcolor'      2 0
*
C 'SFL1'    SETATR   1           'COLNUMBER'
C 'SFL1'    SETATR   5           'COLFGCLR'
*
```

## ColFGMix

サブファイル列の前景カラーの混合を照会または設定します。**ColNumber** 属性を使って、カラー混合が照会または設定される列を識別する必要があります。

**タイプ** スtring

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

前景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      CFclrMx = %getatr('win01': 'SFL1': 'ColFGMix')
*
C  CFclrMx  IFNE     '10:23:200'
C          EVAL      %setatr('win01': 'SFL1': 'ColFGMix') = '10:23:200'
C          ENDIF
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'SFL1'   SETATR   '10:23:200'  'COLFGMIX'
*
```

## Collapsed

ツリー・ビューのコンテナに適用されます。**RecordID** 属性によって指定されるレコードの項目を拡大または縮小するためには、この属性を使用することができます。

0 のレコード ID を指定すると、ツリー・ビュー中のすべてのレコードの状態が反転されます。

Java アプリケーションで特定のレコードを拡大した場合には、そのレコードの子および親レコードが可視になります。レコードを縮小した場合には、子レコードだけが隠され、親レコードは可視のまま残されます。

**タイプ** 数値

**適用可能なパーツ**

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**Collapsed** は、次のいずれかとすることができます。

- 1**      レコードを縮小
- 0**      レコードを拡張

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
* レコードを縮小
C                    Eval        %setatr('window': 'CN1': 'Collapsed') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C    'CN1'            SETATR    1            'COLLAPSED'
*
```

## BatchAdd

多くのオブジェクトを追加するとき、PM 互換スタイルを使用することでパフォーマンスを向上させることができます。

PM 互換コンテナは、まずメモリーにレコードを収集し、一度に 1 つずつではなく、1 つの追加操作で、そのレコードのすべてをコンテナに追加します。

PM 互換コンテナを使用するには、コンテナ・パーツの プロパティ・ノートブックの「スタイル」タブにある「PM 互換」チェック・ボックスをチェックします。また、ソース・コードを変更する必要もあります。

注:

1. コンテナが PM 互換の場合、項目の通常の追加は引き続き実行できますが、パフォーマンスは向上されません。
2. このスタイルは、Pentium II またはローエンド Pentium III など、速度の遅い CPU のみのものであります。ハイエンド Pentium III または速度の速い CPU の場合、このスタイルを使用する必要はありません。
3. PM 互換スタイル・コンテナでの追加操作を完了するには、**NbrOfRec** および **EndAdding** 属性も使用する必要があります。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**BatchAdd** は、次のいずれかとすることができます。

- 1 バッチ追加モードをオンにします。
- 0 バッチ追加モードをオフにします。

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
```

\*

```
*Set 'BATCHADD' attribute to indicate the beginning of  
*batch adding. Also need to use 'NBROFREC' to indicate  
*how many records are going to be added; this value will  
*be used to allocate memory, so it must be equal to or  
*greater than the number of records to be added.
```

```
C 'CT1' Setatr 1 'BATCHADD'
```



```

C      'CT1'          Setatr   200          'NBROFREC'
* Now add records to the container in a normal manner.
* NOTE: these records are actually in memory only, not
* in the container part yet.
* ... .. (codes that add records into container)
*
* All the records have been added. Now use 'ENDADDING'
* attribute to ask the runtime to really add all the records
* into the container part.
C      'CT1'          Setatr   1           'ENDADDING'
*NOTE: When a record is still in memory, it can't be used as the
*parent when try to add another record.
*

```

### 固定形式の例

```

*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CN1'          SETATR   1           'BATCHADD'
*

```

## NbrOfRec

バッチ追加メソッドを使用して追加するレコードの数をランタイムに通知します。それに応じて、ランタイムはメモリーを割り振ります。この値は、少なくとも、追加されるレコードの実際の数と同じにする必要があります。

**BatchAdd** および **EndAdding** 属性も参照してください。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| コンテナー |  |  |
|-------|--|--|

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

追加されるレコードの数。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'CT1'          Setatr    200          'NBROFREC'
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'CN1'          SETATR    200          'NBROFREC'
```

## EndAdding

バッチ追加が終了したことをランタイムに通知します。次に、ランタイムは、メモリー内のすべてのレコードをコンテナ・パーツに追加します。

**BatchAdd** および **NumOfRec** 属性も参照してください。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

1 バッチ追加の終了

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CT1'          Setatr  1          'ENDADDING'
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CT1'          Setatr  1          'ENDADDING'
```

## ColNumber

その他のサブファイル属性を設定する時に使用するフィールドの列番号を設定します。参照されているサブファイル・レコードのフィールドを指示するためには、この属性が **Index** 属性と一緒に使用されます。サブファイルのデータ列が表示される順序を変更するために、これは **ViewColumn** 属性および **MapViewCol** 属性とも一緒に使用されます。

**タイプ** 数値

**適用可能なパーツ**

|      |        |  |
|------|--------|--|
| コンテナ | サブファイル |  |
|------|--------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

フィールドの列番号

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'SF1': 'Index') = 5
C          EVAL      %setatr('win01': 'SF1': 'ColNumber') = 2
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'SF1'      SETATR  5          'INDEX'
C   'SF1'      SETATR  2          'COLNUMBER'
*
```

## Color

**ColorArea** 属性の値によって示されるエリアのカラーを設定して取得します。

タイプ 数値

適用可能なパーツ

|       |     |  |
|-------|-----|--|
| カレンダー | グラフ |  |
|-------|-----|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

このカラーは次の 1 つでなければなりません。

- 1 デフォルト(Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

注: デフォルト・カラー (-1) を SETATR に使用することはできません。

#### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      color = %getatr('win01': 'GP1': 'Color')
*
C  color    IFEQ      *Red
C          EVAL      %setatr('win01': 'GP1': 'Color') = *Blue
C          ENDIF
*
```

#### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GP1'    SETATR   *BLUE      'COLOR'
```

## ColorArea

表示または設定時に、**Color** および **ColorMix** 属性を適用する必要がある対象の領域を指示します。

タイプ 数値

適用可能なパーツ

|       |     |  |
|-------|-----|--|
| カレンダー | グラフ |  |
|-------|-----|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

グラフ・パーツの場合には、**ColorArea** は次のいずれかとすることができます。

- 1 タイトル・カラー
- 2 背景カラー
- 3 軸ラベル・カラー
- 4 軸行カラー
- 5 データ・ポイント・ラベル・カラー
- 6 強調表示カラー
- 7 グラフ背景カラー
- 8 枠カラー
- 9 バー/スライス/行カラー

カレンダー・パーツの場合には、**ColorArea** は次のいずれかとすることができます。

- 1 背景
- 2 グリッド
- 3 カレンダー内部
- 4 月矢印
- 5 月矢印のアウトライン
- 6 月/年のアウトライン
- 7 月/年の背景
- 8 月/年の前景
- 9 曜日名の背景

- 10 曜日名の前景
- 11 選択日付のアウトライン
- 12 当日のアウトライン
- 13 ユーザー・アウトラインのデフォルト・カラー
- 14 ユーザー・テキストのデフォルト・カラー
- 15 枠
- 16 日付数字
- 17 日付数字アウトライン
- 18 ユーザー・アウトライン (実行時のみ)
- 19 ユーザー・テキスト (実行時のみ)

#### フリー・フォームの例

```

*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      colarea = %getatr('win01': 'GP1': 'ColorArea')
*
C  colarea  IFEQ      3
C          EVAL      %setatr('win01': 'GP1': 'ColorArea') = 4
C          ENDIF
*

```

#### 固定形式の例

```

*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GP1'    SETATR   4          'COLORAREA'
*

```



## ColorMix

**ColorArea** 属性の値によって示されるカラー混合を設定して取得します。

タイプ スtring

適用可能なパーツ

|       |     |  |
|-------|-----|--|
| カレンダー | グラフ |  |
|-------|-----|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

カラー混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq----
*
C          EVAL      color=%getatr('win01':'GP1':'ColorMix')
*
C  color      IFNE      '10:23:200'
C          EVAL      %setatr('win01':'GP1':'ColorMix')='0:0:255'
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq----
*
C  'GP1'      SETATR   '10:23:200'  'COLORMIX'
```

## Column

テーブル内のどの列が他の属性 (**BufferLen**、**BufferPtr**、**BufferType**) などで参照されているかを明確にします。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

1 からテーブル内の列の数まで。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'Column')=12
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'ODBC1'      SETATR  12          'COLUMN'
```

## ColumnDec

列に定義されている小数点以下の桁数を戻します。この属性を使用する前に **Column** 属性を設定してください。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Dec=%getatr('win01':'ODBC1':'ColumnDec')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ODBC1'      GETATR      'COLUMNDEC'      DEC
*
```

## ColumnLen

テーブル内の特定の列の定義されている長さを戻します。この属性を使用する前に **Column** 属性を設定してください。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Len=%getatr('win01':'ODBC1':'ColumnLen')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ODBC1'      GETATR      'COLUMNLEN'      LEN
*
```

## ColumnName

レコード・セット内の特定の列の名前を戻します。この属性を使用する前に **Column** 属性を設定してください。 **SQLQuery** および **ExecuteSQL** 属性を使用した SQL 照会が正常に実行されると、その列名が使用可能になります。

タイプ スtring

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      CName=%getatr('win01!':'ODBC1':'ColumnName')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'ODBC1'      GETATR      'COLUMNNAME'  CNAME
*
```

## Columns

結果セット内の列の数を戻します。**SQLQuery** および **ExecuteSQL** 属性を使用した SQL 照会が正常に実行されると、その列の数が使用可能になります。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Cols=%getatr('win01': 'ODBC1': 'Columns')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ODBC1'      GETATR      'COLUMNS'      COLS
*
```

## ColumnType

列のデータ・タイプを戻します。この属性を使用する前に、**Column** および **ExecuteSQL** 属性を設定してください。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

戻される値

戻される値は、次の 1 つとすることができます。

- 1 可変長文字
- 2 長い可変長文字
- 3 文字
- 4 数値
- 5 小数部分
- 6 整数
- 7 短整数
- 8 タイム・スタンプ
- 0 次の **JDBC データ・タイプ**の場合: 日付、時刻、大整数、ビット、2 倍長、浮動、実、小整数、2 進数、長い可変長 2 進数、可変長 2 進数、その他、NULL

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq---
*
C          EVAL      CType=%getatr('win01': 'ODBC1': 'ColumnType')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq---
*
C 'ODBC1'      GETATR  'COLUMNTYPE'  CType
*
```

## ColWidth

サブファイル列の幅をピクセル数で決定します。

影響を受ける列を示すように、 **ColNumber** 属性を設定する必要があります。

**タイプ** 数値

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('win01': 'SFI1': 'ColWidth') = 50
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SFI1'      SETATR  50          'COLWIDTH'
*
```



## CompName

参照されるコンポーネントの名前。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          Eval      %setatr('win1': 'comp1': 'CompName')='Palette'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'COMP1'      SETATR  'Palette'      'COMPNAME'
*
```

## Connect

ODBC/JDBC インターフェース・パーツをデータベースと接続するには、この属性を 1 に設定します。この属性を使用する前に、**ConnectStr** 属性を使用して、接続先のデータ・ソースを決定してください。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'ODBC1':'Connect')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'ODBC1'      SETATR      1          'CONNECT'
*
```

## Connected

テーブルへの接続の試みが成功したかどうかを決定します。この属性が 1 を戻すと、接続は成功しました。その他の値は、接続が失敗したことを示します。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Status=%getatr('win01':'ODBC1':'Connected')
*
```

固定形式の例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ODBC1'      GETATR      'CONNECTED'      STATUS
*
```

## ConnectStr

データ・ソースと接続するために ODBC マネージャーに渡す接続ストリングを提供します。この属性は、Connect 属性を使用する前に設定してください。接続ストリングを \*BLANKS に設定すると、「データ・ソースの選択」ダイアログが現れます。このダイアログから、接続先のテーブルを選択することができます。

接続ストリングの構文については、ODBC API 資料を参照してください。

### タイプ ストリング

#### 適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

#### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

#### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'ConnectStr')=Str
*
```

#### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'ODBC1'      SETATR  STR          'CONNECTSTR'
```

## Copy

選択されたテキストをクリップボードにコピーします。選択されたテキストは削除されません。

タイプ 数値

適用可能なパーツ

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

使用可能な値は 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('win01': 'MLE1': 'Copy') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C 'MLE1'      SETATR 1          'copy'
```

## Count

パーツのリスト部分の項目数を戻します。

リストに項目を追加または除去する時には、**Count** 属性が更新されます。

**タイプ** 数値

**適用可能なパーツ**

|              |        |          |
|--------------|--------|----------|
| 組み合わせボックス    | コンテナ   | リスト・ボックス |
| メッセージ・サブファイル | ノートブック | サブファイル   |

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      rc = %getatr('win01': 'LB1': 'Count')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'LB1'      GETATR  'COUNT'      count          2 0
*
```

## CsrAtEnd

入力フィールドにフォーカスが当てられた時に、カーソルを入力フィールド内の最後の文字の後に位置づけるには、この属性を 1 に設定します。

タイプ 数値

適用可能なパーツ

|         |  |  |
|---------|--|--|
| 入力フィールド |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'EN1': 'CsrAtEnd') = 1
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C  'EN1'      SETATR  1          'CSRATEND'
*
```

## CsrLine

カーソルが位置づけられている行番号を示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

行番号

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'MLE1': 'CsrLine') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'MLE1'      SETATR  2          'CSRLINE'
*
```



## CsrPos

複数行編集フィールドの先頭文字から見たカーソルの相対的な位置を示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

カーソル位置

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'CsrPos') = 4
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'MLE1'          SETATR  4          'CSRPOS'
*
```

## CurrentDir

現行作業ディレクトリーを戻します。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C           EVAL      Dir = %getatr('*Component': '*component':
C           'CurrentDir)
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C '*component' GETATR 'CurrentDir' Dir
*
```

## CurrentRow

結果セットの現在の行番号を戻します。

**注:** Java アプリケーションでは、この属性に適切な JDBC 2.0 承諾ドライバーが必要です。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      CRow=%getatr('win01':'ODBC1':'CurrentRow')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'ODBC1'      GETATR  'CURRENTROW'  CROW
*
```

## Cut

選択されたテキストを削除し、それをクリップボードにコピーします。

タイプ 数値

適用可能なパーツ

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

使用可能な値は 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'Cut') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'MLE1'      SETATR      1          'cut'
*
```

## DataGroup

特定のデータ項目を示すために **DataPoint** と組み合わせて使用されるか、または点の集まりを全体として参照する時に単独で使用されます。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

この値は 0 より大きくなければなりません。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C          EVAL      %setatr('win01': 'GR1': 'DataGroup') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C 'GR1'      SETATR  1          'DATAGROUP'
```

## DataPoint

特定のデータ項目を示すために **DataGroup** と組み合わせて使用されます。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

この値は 0 より大きくなければなりません。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'GR1': 'DataPoint') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'GR1'          SETATR      1          'DATAPOINT'
*
```

## Data Type

このフィールドで使用できるデータのタイプを決定します。

タイプ 数値

適用可能なパーツ

|         |        |  |
|---------|--------|--|
| 入力フィールド | 静的テキスト |  |
|---------|--------|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | いいえ      |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

Data Type は、次のいずれかとすることができます。

- 0 ゾーン数値 - このパーツは数字だけをサポートします。また、ハイフン (-) を負の数を示すための先行文字としてサポートします。
- 1 文字 - このパーツは英数字だけをサポートします。
- 2 DBCS 専用 - このパーツは DBCS 文字だけをサポートします。
- 3 DBCS 択一 - このパーツは、最初に入力された文字に応じて、SBCS 文字または DBCS 文字のいずれか一方（両方は不可）をサポートします。
- 4 DBCS 混用 - このパーツは SBCS 文字と DBCS 文字の両方をサポートします。

フリー・フォームの例

入力フィールドの文字タイプを表示します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      type = %getatr('win01': 'EF1': 'DataType')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'EF1'      GETATR  'DATATYPE'  type          1 0
*
```

## DataValue

特定のデータ項目に割り当てた値を指定します。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'GR1': 'DataValue') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GR1'          SETATR  1          'DATAVALUE'
*
```



## Date

現在選択されている日付を示します。日付は常に形式 YYYYMMDD のストリングで、**Year/Month/Day** 属性と同期化されます。

タイプ ストリング

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

形式 YYYYMMDD のデータ

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'Date') = '19980221'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CAL1'          SETATR      '19980221'          'DATE'
*
```

## Dateldx

現在のカレンダー・ページに表示されない日付に関するデータをユーザーが設定できるようにします。日付は常に形式 YYYYMMDD のstringで、**YearIdx/MonthIdx/DayIdx** と一致します。

タイプ string

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

索引値

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'DateIdx') =
C          = '19980221'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CAL1'          SETATR '19980221'          'DATEIDX'
*
```

## DateText

**DayIdx/MonthIdx/YearIdx** によって示される日付の日付区画の中に表示されるテキスト。

タイプ スtring

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'DateText') = 'text'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CAL1'      SETATR  'text'      'DATETEXT'
```

## DateUnder

日付をstring YYYYYMMDD としてマウス・カーソルの下に戻します。マウスが日付の上になければ、**DateUnder** は空stringを戻します。

タイプ string

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      yyyyymmdd = %getatr('win01': 'CAL1':
C          'DateUnder')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CAL1'          GETATR  'DATEUNDER'      yyyyymmdd
*
```

## Day

現在選択されている日を示します。これが設定されると、カレンダー・パーツ内の選択された日付が、変更を反映するように更新されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

月のうちの日

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'Day') = 21
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CAL1'          SETATR      21          'DAY'
*
```

## DayIdx

現在のカレンダー・ページに表示されない日付に関するデータをユーザーが設定できるようにします。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

索引値

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'DayIdx') = 21
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'CAL1'          SETATR  21          'DAYIDX'
*
```

## DayLen

曜日名を表示するための形式を示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**DayLen** は、次のいずれかとすることができます。

- 1 フルネーム
- 2 省略された 3 文字の名前
- 3 省略された 2 文字のプレフィックス (たとえば、SU、MO、TU...)
- 4 省略された 1 文字のプレフィックス (たとえば、S、M、T...)

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'DayLen') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CAL1'      GETATR  'DAYLEN'    daylen
*
```

## DayNumPos

月のうちの日の番号の位置を示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**DayNumPos** は、次のいずれかとすることができます。

- 1 左上
- 2 右上
- 3 左下
- 4 右下

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'DayNumPos') = 3
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CAL1'      GETATR  'DAYNUMPOS'  daynumpos
*
```



## DayNumRect

月のうちの日にアウトラインが付けられるかどうかを示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**DayNumRect** は、次のいずれかとすることができます。

**0** 番号にアウトラインは付けられません

**1** 番号にアウトラインが付けられます

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      isout = %getatr('win01': 'CAL1': 'DayNumRect')
*
C          EVAL      %setatr('win01': 'CAL1': 'DayNumRect') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'CAL1'    GETATR   'DAYNUMRECT' isout
*
```

## DayStart

週の最初の日が何曜日であるかを決定します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**DayStart** は、次のいずれかとすることができます。

- 0** Sunday
- 1** Monday
- 2** Tuesday
- 3** Wednesday
- 4** Thursday
- 5** Friday
- 6** Saturday

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'DayStart') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CAL1'      SETATR  2          'DAYSTART'
```

## DCoIFRVCoI

**ViewColumn** 属性によって指定された列に対応するデータ列を戻します。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'SF1':'ViewColumn') = 4
C          EVAL      DataCo14 = %getatr('win01':'SF1':'DCoIFRVCoI')
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'SF1'      SETATR   4          'VIEWCOLUMN'
C 'SF1'      GETATR   'DCoIFRVCoI' DATACo14
*
```

サブファイルに 1 列目に入れられたフィールド **CustNo** があるが、ここで **CustNo** が 4 列目に現れるようにビューを変更すると、GETATR 演算命令では、実際にデータ列がサブファイルのどこにあるかを示す **1** が戻されます。

## DDEAddLink

この属性を設定することによって、**AppName** および **Topic** 属性によって定義されたサーバー・アプリケーションと、パーツ・パラメーターによって指示されたパーツとの間で DDE ホット・リンク会話を確立します。

DDE ホット・リンク会話が確立された時には、DDE クライアント・パーツの **Item** 属性によって指定された項目が変更されると、サーバー・アプリケーションは自動的に **Data** イベントを DDE クライアント・パーツに通知します。**AllowLink** 属性が設定された場合には、DDE クライアント・パーツはリンクされたパーツを新規データによって更新します。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | いいえ      |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。これは、「追加」プッシュボタンを選択し、「DDE リンクの追加」ダイアログ・ボックスに値を入力して実行してください。

**使用可能な値**

パーツ・パラメーターは次の通りコピーしなければなりません。

'window|part'

## フリー・フォームの例

入力フィールドとスプレッドシートのセルの間にホット・リンク会話をセットアップします。セルが変更された時に、入力フィールドの内容は変更されます。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'DDE1':'AppName')='APP01'
C          EVAL      %setatr('win01':'DDE1':'Topic')='APP01.WG2'
C          EVAL      %setatr('win01':'DDE1':'Item')='A1'
C          EVAL      %setatr('win01':'DDE1':'DDEAddLink')='win1|ef1'
*
```

## 固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          'DDE1'      SETATR   'win1|ef1'   'DDEADDLINK'
*
```

## DDEMode

DDE クライアント・パーツの現行モードを判別します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

次の 1 つの値によって表されるパーツ・モード。

- 1 開始 - DDE 会話を開始します。会話を開始する前に、**AppName** および **Topic** 属性を設定しておかなければなりません。

DDE 会話が正常に開始された場合には、サーバー・アプリケーションと DDE クライアント・パーツの間でコールド・リンクが確立されます。コールド・リンク会話では、サーバー・アプリケーションからデータを取り出すためには、DDE クライアント・パーツが **Request** 属性を設定し、サーバー・アプリケーションにデータを送信するためには、**Poke** 属性を設定しなければならないことを表します。

- 2 終了 - DDE 会話を終了します。この属性が設定されている時には、打ち切り要求がサーバー・アプリケーションに送信されます。サーバー・アプリケーションが終了要求を処理すると、ユーザー・プログラムに **Terminate** イベントを通知することによって応答します。このイベントを処理するためには、ユーザー・プログラムの中に打ち切りアクション・サブルーチンをコーディングする必要があります。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1++++++OpCode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C                               EVAL      %setatr('win01': 'DDE1': 'DDEMode') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1++++++OpCode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C   'DDE1'          SETATR    1          'DDEMODE'
*
```

## DDERmvLink

この DDE クライアント・パーツからリンクを除去します。**Item** 属性は、除去するリンクを判別するために使用されます。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

パーツ・パラメーターは次の通りコーディングしなければなりません。

'window|part'

**フリー・フォームの例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'DDE1':'DDERmvLink')='win1|ef1'
*
```

**固定形式の例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'DDE1'      SETATR  'win1|ef1'  'DDERMVLINK'
*
```

## DeActivate

ActiveX オブジェクトを非活動化するには、この属性を 1 に設定します。

タイプ 数値

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'AX1': 'DeActivate') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C  'AX1'      SETATR  1          'DEACTIVATE'
*
```



## Delete

選択されたテキストを削除します。

タイプ 数値

適用可能なパーツ

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

使用可能な値は 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'Delete') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'MLE1'      SETATR      1          'delete'
*
```

## DeleteRcd

コンテナ・パーツの特定レコードまたはすべてのレコードを削除します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**DeleteRcd** は、次のいずれかとすることができます。

**0** すべてのレコードを削除します。

**n** **RecordID** 属性によって指定されたレコードを削除します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CT1': 'DeleteRcd') = 0
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CT1'      SETATR  0          'DELETERCD'
```

## DeleteRow

結果セット内の削除する行番号を決定します。先に行を取り出して削除する必要はありません。この属性によって、いずれかの行が削除されます。

取り出された行を削除するには、**CurrentRow** 属性を使用してその行番号を取得してください。

**注:** Java アプリケーションでは、この属性に適切な JDBC 2.0 承諾ドライバーが必要です。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'DeleteRow')=4
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'ODBC1'      SETATR  4          'DELETEROW'
```

## DelimChar

データ・ストリング中の複数の項目を分離する区切り文字を指定します。  
**DelimChar** 属性と **UseDelim** 属性は一緒に使用します。

たとえば、セミコロン (;) を使用して、データ・ストリング中の項目を次のように区切ることができます: abcd;efgh;ijkl。 **AddItemEnd** 属性を使用すると、このストリングから縦の項目リストが作成されます。これによって、単一の操作で複数の項目を追加することができます。

**タイプ** ストリング

**適用可能なパーツ**

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'CB1':'DelimChar') = ';'
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CB1'      SETATR      ';'      'DELIMCHAR'
```

## DeSelect

パーツのリスト部分の項目を選択解除します。

Windows アプリケーションでは、**DeSelect** は次の場合には無効です。

- コンテナが拡張または複数選択に設定され、ツリー・ビューにある場合。
- サブファイルが単一選択に設定されている場合。

これらの場合には、項目が選択されると項目は選択されたままとなります。

Java アプリケーションでは、コンテナをツリー・ビュー内の拡張または複数選択として設定することができます。

**タイプ** 数値

**適用可能なパーツ**

| 組み合わせボックス* | コンテナ | リスト・ボックス |
|------------|------|----------|
| サブファイル     |      |          |

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

選択解除する項目の番号を指定してください。リストのすべての項目を選択解除するためには、この値をゼロに指定してください。

**フリー・フォームの例**

リスト内の 5 番目の項目を選択解除します。

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'LB1': 'DeSelect') = 5
*
```

**固定形式の例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      SETATR  5          'DESELECT'
```

## Dialog

「オープン」または「別名保管」ダイアログが選択可能です。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**Dialog** は、次のいずれかとすることができます。

- 1 オープン
- 2 別名保管

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      rc = %getatr('*component': '*component':
C          'Dialog')
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C '*component' GETATR 'DIALOG'      rc
```

## DIRName

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

コンポーネントが常駐するディレクトリーの名前を戻します。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

使用可能な値

テキスト・String

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      dir = %getatr('*component': '*component':
C          'DIRName')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C '*component' GETATR 'DIRNAME' dir
*
```

## DlgOwner

ダイアログを所有するウィンドウを示します。指定しないと、所有権のデフォルトはデスクトップになります。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

この値の形式は次のとおりでなければなりません。

WindowName WindowName

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('*component': '*component': 'DlgOwner')=
C          'win1 win1'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C '*component' SETATR 'win1 win1' 'DLGOWNER'
*
```



## DoEvents

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

他のイベントが関連アクション・サブルーチンの実行をトリガーできるように、現在実行中のアクション・サブルーチンから制御を解放するには、この属性を 1 に設定します。これは、アクション・サブルーチンが長時間実行ループの中にあっただけの場合にとくに有用です。

**タイプ** 数値

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('*component':'*component':'DoEvents')=1
*
```

**固定形式の例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      '*COMPONENT' SETATR      1          'DOEVENTS'
*
```

## DragEnable

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

この属性が使用可能な場合には、そのパーツにポインターを位置づけて、右マウス・ボタンを押したままにして、そのパーツのテキストまたはラベルを移動することができます。

**タイプ** 数値

**適用可能なパーツ**

|              |         |          |
|--------------|---------|----------|
| 組み合わせボックス    | 入力フィールド | リスト・ボックス |
| メッセージ・サブファイル | 複数行編集   | 静的テキスト   |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

次の 1 つです。

**0** パーツのテキストまたはラベルをドラッグすることはできません。

**1** パーツのテキストまたはラベルをドラッグすることができます。

**注:** この属性は 1 にしか設定することができません。

**フリー・フォームの例**

次のように、パーツの **DragEnable** を使用可能にします。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'EF1': 'DragEnable') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'EF1'      SETATR  1          'DRAGENABLE'
*
```

## DropEnable

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

パーツが **Drop** イベントに応答するかどうかを判別します。

**タイプ** 数値

**適用可能なパーツ**

|              |         |          |
|--------------|---------|----------|
| 組み合わせボックス    | 入力フィールド | リスト・ボックス |
| メッセージ・サブファイル | 複数行編集   | 静的テキスト   |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

次の 1 つです。

- 0** パーツは **Drop** イベントに応答しません
- 1** パーツは **Drop** イベントに応じます

**注:** この属性のデフォルト設定は 0 (すなわちオフ) です。ビルダーでいったん 1 に設定すると、これをリセットすることはできません。

**フリー・フォームの例**

次のように、パーツをドロップ使用可能にします。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'EF1': 'DropEnable') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C  'EF1'          SETATR  1          'DROPENABLE'
*
```

## DropValue

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

ドロップ操作でテキストまたはラベルを変更できるかどうかを決定します。パーツが **Drop** イベントに応答できるように、**DropEnable** 属性を設定します。

たとえば、ユーザーが「メール送信」ダイアログを開始するために e メール・アドレスにドラッグ・アンド・ドロップできるように、「Send Mail」ラベルをコーディングできます。ドロップ操作後に「Send Mail」ラベルを未変更のままにしておくには、DropValue 属性を 0 に設定します。

**タイプ 数値**

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| 静的テキスト |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

次の 1 つです。

**0** パーツのテキストまたはラベルは、変更できません。

**1** パーツのテキストまたはラベルは変更できます。

**注:** Drop イベントを使用できるように、**DropEnable** 属性を 1 に設定してください。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'ST1': 'DropValue') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'ST1'      SETATR      1          'DROPVALUE'
*
```

## DspHeight

表示の高さをピクセル単位で戻します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

次のように、表示幅を取り出します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      DH=%getatr('*Component':'*Component':'DspHeight')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      '*COMPONENT' GETATR  'DSPHEIGHT'  DH
*
```

## DspWidth

表示幅をピクセル単位で戻します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

次のように、表示幅を取り出します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      Dwd=%getatr('*Component': '*Component':
C          'DspWidth')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C '*COMPONENT' GETATR 'DSPWIDTH' Dwd
*
```

## EditColumn

編集用にオープンされているセルの列番号を戻します。編集のためにオープンされているセルがない場合、0 が戻されます。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      edcol = %getatr('win01': 'SF1': 'EditColumn')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'SF1'          GETATR      'EDITCOLUMN'  EDCOL
*
```

## EditIndex

編集用にオープンされているセルの行番号を戻します。編集のためにオープンされているセルがない場合、0 が戻されます。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      edrow = %getatr('win01': 'SF1': 'EditIndex')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'SF1'          GETATR      'EDITINDEX'  EDROW
*
```



## EditItem

Windows アプリケーションでは、この属性を 1 に設定すると、アイコンまたはツリー・ビューでコンテナのレコード・テキストの編集フィールドがオープンします。Java アプリケーションでは、この属性が適用されるのはツリー・ビューだけです。この属性を使用する前に、**ColNumber** 属性を設定しなければなりません。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CN1': 'EditItem')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CN1'      SETATR  1          'EDITITEM'
```

## EditText

現在編集集中のセルの編集済みテキストを戻します。編集用にオープンされているセルがない場合には、空テキストを戻します。

**タイプ** スtring

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      celtext = %getatr('win01': 'SF1': 'EditText')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SF1'      GETATR  'EDITTEXT'  celtext      100
*
```

## EnableBtn

ナビゲーション・ボタンを使用可能または使用不能にします。**EnableBtn** 属性を使用する前に、**ButtonIdx** を設定する必要があります。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

値を次のものに設定することができます。

- 0 ボタンを使用不能にする
- 1 ボタンを使用可能にする

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'SF11': 'EnableBtn') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SF11'      SETATR  1          'ENABLEBTN'
```

## Enabled

パーツがユーザー・インターフェース・イベントに応答するかどうかを判別します。

**注:** キャンバス上のパーツにこの属性を指定すると、そのキャンバス・パーツに設定された属性値が優先権を持ちます。

**タイプ** 数値

**適用可能なパーツ**

|               |                   |           |
|---------------|-------------------|-----------|
| カレンダー         | キャンバス             | チェック・ボックス |
| 組み合わせボックス     | コンテナ              | 入力フィールド   |
| グラフ           | グラフィック・プッシュボタン    | グループ・ボックス |
| 水平方向のスクロール・バー | イメージ              | Java Bean |
| リスト・ボックス      | メディア・パネル          | メニュー項目    |
| メッセージ・サブファイル  | 複数行編集             | ノートブック    |
| ノートブック・ページ    | キャンバス付きノートブック・ページ | プッシュボタン   |
| ラジオ・ボタン       | スライダー             | スピン・ボタン   |
| サブファイル        | 縦方向のスクロール・バー      | ウィンドウ     |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

次の 1 つです。

- 0** パーツはイベントに応答することはできません。
- 1** パーツはイベントに応答することができます。

## フリー・フォームの例

次のように、パーツの使用可能状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      rc = %getatr('win01': 'EF1': 'Enabled')
*
C   rc                          IFEQ      0
C                               EVAL      %setatr('win01': 'EF1': 'Enabled') = 1
*
C                               ELSE
C                               EVAL      %setatr('win01': 'EF1': 'Enabled') = 0
C                               ENDIF
*
```

## 固定形式の例

次のように入力フィールドの使用可能状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'EF1'                       GETATR   'ENABLED'   rc           1 0
C   rc                           IFEQ     1
C   'EF1'                       SETATR   0           'ENABLED'
*
C                               ELSE
C   'EF1'                       SETATR   1           'ENABLED'
C                               ENDIF
*
```

## Execute

この属性を設定すると、DDE 会話のサーバー・アプリケーションに 実行 コマンドが通知されます。コマンド・ストリングの構造はサーバー・アプリケーションによって決まります。サポートされるコマンドおよびコマンドの構造についてはサーバー・アプリケーションの資料を参照してください。

この属性を設定する前に、DDE 会話が正常に開始されていなければなりません。

サーバー・アプリケーションが 実行 コマンドを実行すると、**ExecuteAck** イベントが生成されます。正常に実行されたかどうかを判別するためには、ユーザーの ExecuteAck イベント・アクション・サブルーチンの **%DDEAck** イベント属性をチェックしてください。

**タイプ** ストリング

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'DDE1': 'Execute') = 'FILEOPEN'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'DDE1'      SETATR  'FILEOPEN'  'EXECUTE'
*
```

## ExecuteSQL

この属性を 1 に設定すると、最後の **SQLQuery** 属性によって実行依頼された SQL ステートメントが実行されます。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'ExecuteSQL')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ODBC1'      SETATR      1          'EXECUTESQL'
```

## ExtSelect

パーツが拡張選択を使用できるかどうかを決定します。拡張選択によって、ユーザーは項目を選択してから、追加の項目を選択することができます。複数の項目にマウス・カーソルをドラッグすることによって、これらにマークを付けることができます。

**MultiSelect** 属性は **ExtSelect** 属性上では無効です。 **ExtSelect** 属性は、**MultiSelect** または単一選択のいずれかの設定を指定変更します。

タイプ 数値

適用可能なパーツ

|          |              |            |
|----------|--------------|------------|
| コンテナ(1.) | リスト・ボックス(2.) | サブファイル(2.) |
|----------|--------------|------------|

### 制約事項:

1. Windows アプリケーションでは、コンテナ・パーツの場合にのみ実行時にこの属性を設定することができます。
2. Java アプリケーションでは、リスト・ボックスおよびサブファイル・パーツの場合にのみ実行時にこの属性を取り出すことができます。

### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

### 使用可能な値

次の 1 つです。

- 0 拡張選択を使用することはできません。
- 1 拡張選択を使用することができます。

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      rc = %getatr('win01': 'LB1': 'ExtSelect')
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      GETATR  'EXTSELECT'  rc          1 0
*
```



## Fetch

この属性を N に設定すると、接続されたデータ・ソースから N 番目の行が戻されます。(照会の結果セットに NUMBER のレコードが含まれていると仮定すると、値 N は 1 から NUMBER までの任意の数字にできます。) 戻されるデータは、最後に出された取り出し要求のタイプによって決まります。**GetTables** 属性要求の場合には、**Fetch** はデータベースのテーブルのリストを戻します。**ExecuteSQL** 要求の場合には、**Fetch** はテーブルから 1 行分のデータを戻します。

**注:** Java アプリケーションでは、この属性に適切な JDBC 2.0 承諾ドライバーが必要です。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'Fetch')=1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'ODBC1'      SETATR      1          'FETCH'
*
```

## FetchNext

この属性を 1 に設定すると、結果セットからデータの次の行が取り出されます。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'FetchNext')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'ODBC1'      SETATR      1          'FETCHNEXT'
*
```

## FetchPrior

この属性を 1 に設定すると、結果セットからデータの直前の行が取り出されます。

**注:** Java アプリケーションでは、この属性に適切な JDBC 2.0 承諾ドライバーが必要です。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'FetchPrior')=1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'ODBC1'      SETATR  1          'FETCHPRIOR'
*
```

## FieldExit

Enter キーが押された時、次のタブ移動可能入力フィールドにカーソルがタブ移動するようにします。

**タイプ** 数値

**適用可能なパーツ**

|           |         |  |
|-----------|---------|--|
| 組み合わせボックス | 入力フィールド |  |
|-----------|---------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

次の 1 つです。

- 0** カーソルは次の適用可能入力フィールドにタブ移動しません
- 1** カーソルは次の適用可能入力フィールドにタブ移動します

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'EF1': 'FieldExit') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'EF1'      SETATR  1          'FIELDEXIT'
*
```

---

## FileName

パーツと関連づけるファイルを指定します。

メディア・パーツの場合には、これを有効なウェーブ (.wav) または MIDI (.mid) ファイル、あるいは .mov または .mpg 拡張子を持つ有効なビデオ・ファイルとすることができます。

アニメーション制御パーツを使用している Windows アプリケーションでは、このファイルは .avi の拡張子を持つビデオ・ファイルとすることができます。

**NbrOfImage** 属性のアニメーション・コントロール・パーツを使用する Java アプリケーションでは、ファイルは GIF イメージとすることができます。

キャンバス、グラフィック・プッシュボタン、イメージ、またはメニュー項目の場合には、このファイルは有効なイメージ・ファイルでなければなりません。有効なイメージ形式のリストについては、22 ページの『イメージ』を参照してください。

メニュー項目パーツの場合には、**FileName** がビットマップを参照していると、そのメニュー項目はビットマップとして表示されます。**FileName** がアイコン・ファイルを参照していると、メニュー項目テキストの横にそのアイコンが表示されます。Java アプリケーションでは、**FileName** が GIF ファイルを参照していると、メニュー項目テキストの横にその GIF イメージが表示されます。

Java アプリケーションでは、**FileName** は **Window** パーツのアイコンとして表示される GIF ファイルを参照します。

メディア・パーツの場合に **FileName** 属性を変更すると、そのファイルは処理されません。ファイルを処理するためには、**AudioMode** 属性を設定しなければなりません。

キャンバス、グラフィック・プッシュボタン、またはイメージ・パーツの **FileName** 属性を変更すると、そのパーツ上のピクチャーが即時に変更されることとなります。

\*Component の場合に **FileName** 属性を設定すると、**Dialog** 属性で選ばれたダイアログが表示されます。

完全なパス情報を持つファイル名またはファイル名だけのいずれかを使用してファイル名を指定することができます。ファイル名だけが指定された場合には、そのファイルを実行時サブディレクトリー (Java アプリケーションの場合には RT\_JAVA サブディレクトリー、Windows アプリケーションの場合には RT\_WIN32 サブディレクトリー) に入れる必要があります。ファイル名だけが指定されると、RT\_JAVA または RT\_WIN32 サブディレクトリーが現行ディレクトリーの場合にしかファイルは見つかりません。

ファイル名に関する考慮事項の詳細については、*VisualAge for RPG プログラミング* を参照してください。

## タイプ ストリング

### 適用可能なパーツ

|                |       |              |
|----------------|-------|--------------|
| アニメーション制御      | キャンバス | *Component   |
| グラフィック・プッシュボタン | イメージ  | メディア         |
| メニュー項目         | ウィンドウ | キャンバス付きウィンドウ |

### 命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**\*制約事項:** Java アプリケーションでは、この属性だけが設定可能です。

### 使用可能な値

このパーツと関連付けるワークステーション・ファイルの名前

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'audio1':'FileName')='BARK.WAV'
*
```

### 固定形式の例

次の通り、イメージ・パーツの現行の filename を保管して、それを新規の値に設定します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'IMAGE01'  GETATR  'FILENAME'  oldfile      64
C      'IMAGE01'  SETATR  newfile    'FILENAME'
*
```

## FillStyle

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

領域の埋め込み方法を説明します。

**タイプ** 数値

**適用可能なパーツ**

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**FillStyle** は次の 1 つとすることができます。

- 1 リボンとじ
- 2 中空き
- 3 低密度の埋め込み
- 4-9 埋め込み密度の度合いの増加
- 10 高密度の埋め込み
- 11 縦方向ストライプ
- 12 水平方向ストライプ
- 13 正方向斜めストライプ
- 14 超正斜線ストライプ
- 15 逆斜線ストライプ
- 16 超逆斜線ストライプ
- 17 ハーフトーン埋め込み
- 18 ハッチング
- 19 斜線ハッチング

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      style = %getatr('win01': 'GP1': 'FillStyle')
*
C  style    IFEQ      3
C          EVAL      %setatr('win01': 'GP1': 'FillStyle') = 12
C          ENDIF
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GP1'    SETATR   12      'FILLSTYLE'
```



## FirstSel

パーツのリスト部分で最初に選択した項目の索引値を戻します。索引値は 1 から始まります。リスト項目が選択されないと、ゼロの値が戻されます。

**タイプ** 数値

**適用可能なパーツ**

|              |        |          |
|--------------|--------|----------|
| 組み合わせボックス    | コンテナ   | リスト・ボックス |
| メッセージ・サブファイル | サブファイル |          |

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq----
*
C          EVAL      ix = %getatr('win01': 'LB1': 'FirstSel')
*
```

**固定形式の例**

リスト・ボックスで項目が選択されたかどうかをチェックして、選択しなかった場合には、メッセージが表示されます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq----
*
C  'LB1'      GETATR  'FIRSTSEL'  IX          3 0
C  IX        IFEQ    *ZERO
C  'Select error'DSPLY  STYLE      rc
C          ENDIF
*
```

## Focus

パーツに現在フォーカスが当てられているかどうかを判別します。パーツにフォーカスがある場合には、入力カーソルはそこに移動されて、ユーザーの入力がそこに送られます。たとえば、入力フィールドに入力するためには、前もってその入力フィールドにフォーカスがなければなりません。

パーツがフォーカスされると、**GotFocus** イベントを受信します。

**タイプ** 数値

**適用可能なパーツ**

|           |                |                |
|-----------|----------------|----------------|
| チェック・ボックス | 組み合わせボックス      | 12 ページの『コンテナー』 |
| 入力フィールド   | グラフィック・プッシュボタン | 水平方向のスクロール・バー  |
| リスト・ボックス  | 複数行編集          | ノートブック         |
| プッシュボタン   | ラジオ・ボタン        | スライダー          |
| スピン・ボタン   | サブファイル         | 縦方向のスクロール・バー   |
| ウィンドウ*    |                |                |

\* **制約事項:** Java アプリケーションでは、ウィンドウ・パーツの場合にのみ実行時にこの属性を設定することができます。

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

次の 1 つです。

- 0** パーツに入力フォーカスは当てられていません。
- 1** パーツに入力フォーカスが当てられています。

**注:** この属性は 1 にしか設定することができません。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'EF1': 'Focus') = 1
*
C          EVAL      rc = %getatr('win01': 'EF1': 'Focus')
```

## 固定形式の例

次の通り、パーツのフォーカスを切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'EF1'      GETATR  'FOCUS'      HasFocus      1 0
*
C 'EF1'      SETATR  1              'FOCUS'
```

## FontArea

他のフォント関連属性が適用される、カレンダー、グラフ、またはサブファイル・パーツ内の領域を示します。

**タイプ** 数値

**適用可能なパーツ**

|       |     |        |
|-------|-----|--------|
| カレンダー | グラフ | サブファイル |
|-------|-----|--------|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

カレンダー・パーツの場合には、値は次の 1 つとすることができます。

- 1 月/年
- 2 曜日名
- 3 ユーザー・テキストのデフォルト
- 4 日番号
- 5 ユーザー・テキスト

グラフ・パーツの場合には、値は次のいずれかとすることができます。

- 1 タイトル・フォント
- 2 軸ラベル・フォント
- 3 データ・ポイント・ラベル・フォント
- 4 凡例フォント

サブファイル・パーツの場合には、値は次のいずれかとすることができます。

- 1 見出し
- 2 すべてのレコード
- 3 1 つのレコード (INDEX によって指示される)

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      Farea = %getatr('win01': 'GRA1': 'FontArea')
*
C          EVAL      %setatr('win01': 'GRA1': 'FontArea') = 5
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'GRA1'   GETATR   'FONTAREA'   Farea           1 0
*
```

## FontBold

ラベルまたはテキストを **太字体** フォントで表示するかどうかを判別します。

**タイプ** 数値

**適用可能なパーツ**

|              |           |           |
|--------------|-----------|-----------|
| カレンダー        | キャンバス     | チェック・ボックス |
| 組み合わせボックス    | コンテナ      | 入力フィールド   |
| グラフ          | グループ・ボックス | リスト・ボックス  |
| メッセージ・サブファイル | 複数行編集     | ノートブック    |
| プッシュボタン      | ラジオ・ボタン   | スライダー     |
| スピン・ボタン      | 静的テキスト    | サブファイル    |
| ウィンドウ*       |           |           |

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

**注:** カレンダーおよびグラフ・パーツの場合には、実際に影響を受けるフォント域は、**FontArea** 属性の値によって決まります。

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。これは、「フォント」タブを使用し、「フォントの変更」プッシュボタンを選択することによって実行してください。

## 使用可能な値

**FontBold** は、次のいずれかとすることができます。

**0**      フォントを太字体で表示しません。

**1**      フォントを太字体で表示します。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C                               EVAL    bold = %getatr('win01': 'ST1': 'FontBold')
*
C    bold                       IFEQ     0
C                               EVAL     %setatr('win01': 'ST1': 'FontBold') = 1
C                               ELSE
C                               EVAL     %setatr('win01': 'ST1': 'FontBold') = 0
C                               ENDF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C    'ST1'                       GETATR  'FONTBOLD'  bold          1 0
*
C    'ST1'                       SETATR  1           'FONTBOLD'
```

## FontItalic

ラベルまたはテキストを イタリック体 フォントで表示するかどうかを判別します。

タイプ 数値

適用可能なパーツ

|              |           |           |
|--------------|-----------|-----------|
| カレンダー        | キャンバス     | チェック・ボックス |
| 組み合わせボックス    | コンテナー     | 入力フィールド   |
| グラフ          | グループ・ボックス | リスト・ボックス  |
| メッセージ・サブファイル | 複数行編集     | ノートブック    |
| プッシュボタン      | ラジオ・ボタン   | スライダー     |
| スピン・ボタン      | 静的テキスト    | サブファイル    |
| ウィンドウ*       |           |           |

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

**注:** カレンダーおよびグラフ・パーツの場合には、実際に影響を受けるフォント域は、**FontArea** 属性の値によって決まります。

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。これは、「フォント」タブを使用し、「フォントの変更」プッシュボタンを選択することによって実行してください。



## 使用可能な値

**FontItalic** は、次のいずれかとすることができます。

**0** フォントをイタリック体で表示しません。

**1** フォントをイタリック体で表示します。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      italic = %getatr('win01': 'ST1': 'FontItalic')
*
C  italic      IFEQ      0
C          EVAL      %setatr('win01': 'ST1': 'FontItalic') = 1
C          ELSE
C          EVAL      %setatr('win01': 'ST1': 'FontItalic') = 0
C          ENDIF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'ST1'      GETATR  'FONTITALIC' italic      1 0
*
C  'ST1'      SETATR  1          'FONTITALIC'
```

## FontName

このパーツが適用されるテキストまたはラベルに使用するフォントを指定します。

タイプ スtring

適用可能なパーツ

|              |           |           |
|--------------|-----------|-----------|
| カレンダー        | キャンバス     | チェック・ボックス |
| 組み合わせボックス    | コンテナ      | 入力フィールド   |
| グラフ          | グループ・ボックス | リスト・ボックス  |
| メッセージ・サブファイル | 複数行編集     | ノートブック    |
| プッシュボタン      | ラジオ・ボタン   | スライダー     |
| スピン・ボタン      | 静的テキスト    | サブファイル    |
| ウィンドウ*       |           |           |

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

**注:** カレンダーおよびグラフ・パーツの場合には、実際に影響を受けるフォント域は、**FontArea** 属性の値によって決まります。

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。これは、「フォント」タブを使用し、「フォントの変更」プッシュボタンを選択することによって実行してください。

使用可能な値

フォント名

フリー・フォームの例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      oldfont = %getatr('win01': 'ST1': 'FontName')
*
C          EVAL      %setatr('win01': 'ST1': 'FontName') = 'COURIER'
*
```

固定形式の例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'ST1'      GETATR  'FONTNAME'  oldfont      64
*
C  'ST1'      SETATR  'COURIER'   'FONTNAME'
```

## FontSize

このパーツで使用するフォントのサイズを指定します。**FontSize** 属性はポイント数で表現されます。

タイプ 数値

適用可能なパーツ

|              |           |           |
|--------------|-----------|-----------|
| カレンダー        | キャンバス     | チェック・ボックス |
| 組み合わせボックス    | コンテナ      | 入力フィールド   |
| グラフ          | グループ・ボックス | リスト・ボックス  |
| メッセージ・サブファイル | 複数行編集     | ノートブック    |
| プッシュボタン      | ラジオ・ボタン   | スライダー     |
| スピン・ボタン      | 静的テキスト    | サブファイル    |
| ウィンドウ*       |           |           |

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

**注:** カレンダーおよびグラフ・パーツの場合には、実際に影響を受けるフォント域は、**FontArea** 属性の値によって決まります。

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。これは、「フォント」タブを使用し、「フォントの変更」プッシュボタンを選択することによって実行してください。

使用可能な値

ポイントで指定されたフォント・サイズ。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len+D+HiLoEq----
*
C                               EVAL      oldsize = %getatr('win01': 'ST1': 'FontSize')
*
C                               EVAL      %setatr('win01': 'ST1': 'FontSize') = 10
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'ST1'          GETATR   'FONTSIZE'   oldsize      2 0
*
C   'ST1'          SETATR   10           'FONTSIZE'
```

## FontStrike

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

テキストまたはラベルを取り消し線フォントで表示するかどうかを判別します。

**タイプ** 数値

**適用可能なパーツ**

|              |           |           |
|--------------|-----------|-----------|
| カレンダー        | キャンバス     | チェック・ボックス |
| 組み合わせボックス    | コンテナー     | 入力フィールド   |
| グラフ          | グループ・ボックス | リスト・ボックス  |
| メッセージ・サブファイル | 複数行編集     | ノートブック    |
| プッシュボタン      | ラジオ・ボタン   | スライダー     |
| スピン・ボタン      | 静的テキスト    | サブファイル    |
| ウィンドウ        |           |           |

**注:** カレンダーおよびグラフ・パーツの場合には、実際に影響を受けるフォント域は、**FontArea** 属性の値によって決まります。

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。これは、「フォント」タブを使用し、「フォントの変更」プッシュボタンを選択することによって実行してください。

## 使用可能な値

**FontStrike** は、次のいずれかとすることができます。

**0** フォントを取り消し線で表示しません。

**1** フォントを取り消し線で表示します。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      strikeouts=%getatr('win01':'ST1':'FontStrike')
*
C      strikeouts                IFEQ        0
C                               EVAL      %setatr('win01':'ST1':'FontStrike')=1
C                               ELSE
C                               EVAL      %setatr('win01':'ST1':'FontStrike')=0
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ST1'                    GETATR    'FONTSTRIKE' strikeouts          1 0
*
C      'ST1'                    SETATR    1          'FONTSTRIKE'
*
```

## FontUnder

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

テキストまたはラベルを 下線付きにするかどうかを判別します。

**タイプ** 数値

**適用可能なパーツ**

|              |           |           |
|--------------|-----------|-----------|
| カレンダー        | キャンバス     | チェック・ボックス |
| 組み合わせボックス    | コンテナー     | 入力フィールド   |
| グラフ          | グループ・ボックス | リスト・ボックス  |
| メッセージ・サブファイル | 複数行編集     | ノートブック    |
| プッシュボタン      | ラジオ・ボタン   | スライダー     |
| スピン・ボタン      | 静的テキスト    | サブファイル    |
| ウィンドウ        |           |           |

**注:** カレンダーおよびグラフ・パーツの場合には、実際に影響を受けるフォント域は、**FontArea** 属性の値によって決まります。

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。これは、「フォント」タブを使用し、「フォントの変更」プッシュボタンを選択することによって実行してください。

**使用可能な値**

**FontUnder** は、次のいずれかとすることができます。

- 0**      テキストに下線を付けません。
- 1**      テキストに下線を付けます。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      underline=%getatr('win01':'ST1':'FontUnder')
*
C  underline  IFEQ      0
C          EVAL      %setatr('win01':'ST1':'FontUnder')=1
C          ELSE
C          EVAL      %setatr('win01':'ST1':'FontUnder')=0
C          ENDF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'ST1'      GETATR   'FONTUNDER'  underline      1 0
*
C  'ST1'      SETATR   1              'FONTUNDER'
```



## ForeColor

パーツの前景カラーを決定する数値。

**ForeMix** 属性を使用して前景カラーを設定した場合には、戻される前景カラーが混合カラーに最も近いカラーです。

**タイプ** 数値

**適用可能なパーツ**

|              |           |          |
|--------------|-----------|----------|
| チェック・ボックス    | 組み合わせボックス | コンテナー    |
| 入力フィールド      | グループ・ボックス | リスト・ボックス |
| メッセージ・サブファイル | 複数行編集     | ノートブック   |
| プッシュボタン      | ラジオ・ボタン   | スライダー    |
| スピン・ボタン      | 静的テキスト    | サブファイル   |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

このカラーは次の 1 つでなければなりません。

- 1 デフォルト(Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色

- 10 暗赤色
- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

注: デフォルト・カラー (-1) を SETATR に使用することはできません。

### フリー・フォームの例

次の通りフィールドを反転表示します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      fcolor = %getatr('win01': 'EF1': 'ForeColor')
C          EVAL      bcolor = %getatr('win01': 'EF1': 'BackColor')
*
C          EVAL      %setatr('win01': 'EF1': 'ForeColor') = 2
C          EVAL      %setatr('win01': 'EF1': 'BackColor') = 15
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'EF1'      GETATR  'FORECOLOR'  fcolor          2 0
*
C 'EF1'      SETATR  2            'FORECOLOR'
```

## ForeMix

前景の混合カラーを照会または設定します。

タイプ ストリング

適用可能なパーツ

|              |           |          |
|--------------|-----------|----------|
| チェック・ボックス    | 組み合わせボックス | コンテナ     |
| 入力フィールド      | グループ・ボックス | リスト・ボックス |
| メッセージ・サブファイル | 複数行編集     | ノートブック   |
| プッシュボタン      | ラジオ・ボタン   | スライダー    |
| スピン・ボタン      | 静的テキスト    | サブファイル   |

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

## 使用可能な値

前景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      colormix = %getatr('win01': 'EF1': 'ForeMix')
*
C                               EVAL      %setatr('win01': 'EF1': 'ForeMix') = '10:23:200'
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'EF1'          SETATR      '10:23:200'  'FOREMIX'
*
```

## Format

**Item** 属性によって参照され、DDE 会話にサーバー・アプリケーションに送信されるデータの形式を判別します。この属性の値はサーバー・アプリケーションによって判別されます。有効なフォーマット・タイプについては、サーバー・アプリケーションの資料を参照してください。デフォルト値は `cf_text` です。この属性は **Poke** 属性を使用する前に設定しなければなりません。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

参照され、サーバー・アプリケーションに送信されたデータの形式

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'DDE1': 'Format') = 'bitmap'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'DDE1'      SETATR  'bitmap'      'Format'
```

## FrameRate

アニメーション GIF シーケンスにおいて、フレーム間で一時停止する時間 (ミリ秒) を決定します。

タイプ 数値

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| アニメーション制御 |  |  |
|-----------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

イメージ間で一時停止するミリ秒数。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'WIN1':'FrameRate')=1000
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'ANI'      SETATR  1000      'FRAMERATE'
```

## FrmtString

月と年がどのような組み合わせで表示されるかを示します。

MMONTH または YYEAR が書式ストリングの中に見つかり、それぞれが現在の月または年で置き換えられます。たとえば、July 1997 に対する書式ストリング "MMONTH - YYEAR" は "July - 1997" となります。

タイプ ストリング

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('win01': 'CAL1': 'FrmtString') =
C          'MMONTH/YEAR'
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C  'CAL1'    SETATR  'MMONTH/YEAR' 'FRMTSTRING'
```

## GetItem

リストから項目を戻します。戻される項目は、最初に **Index** 索引を設定することによって判別されます。

**タイプ** スtring

**適用可能なパーツ**

|           |          |              |
|-----------|----------|--------------|
| 組み合わせボックス | リスト・ボックス | メッセージ・サブファイル |
|-----------|----------|--------------|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      x = %getatr('win01': 'LB1': 'GetItem')
*
```

**固定形式の例**

次の通り、リスト・ボックスに項目を保管した後で、それを変更します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      SETATR  3          'INDEX'
C  'LB1'      GETATR  'GETITEM'  saveitem      64
C  'LB1'      SETATR  'New string' 'SetItem'
*
```



## GetNewID

コンテナ・レコードの新しい固有の ID を戻します。この属性を使用して、各コンテナ・レコード ID が固有であることを保証します。戻されるレコード ID は数値です。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      id = %getatr('win01': 'CN1': 'GetNewID')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'CN1'      GETATR  'GetNewID'   id
*
```

## GetRcdFld

コンテナ・レコードのフィールドの内容を戻します。コンテナの **RecordID** 属性は、どのコンテナ・レコードを使用するかを指示するために設定しなければなりません。また、**ColNumber** 属性は、データを検索するフィールドを指示するために設定しなければなりません。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL    %setatr('win01': 'CN1': 'RecordID')=ID
C          EVAL    %setatr('win01': 'CN1': 'ColNumber')=1
C          EVAL    data = %getatr('win01': 'CN1': 'GetRcdFld')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'CN1'      SETATR  ID          'RECORDID'
C 'CN1'      SETATR  1          'COLNUMBER'
C 'CN1'      GETATR  'GETRCDFLD' data
*
```

## GetRcdIcon

ツリー・ビューおよびアイコン・ビュー内のレコードを表すために使用されるアイコンのファイル名を戻します。**RecordID** 属性は、どのレコードを参照しているかを指示するために設定しなければなりません。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      file = %getatr('win01': 'CN1': 'GetRcdIcon')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'CN1'      GETATR  'GETRCDICON' file
*
```

## GetRcdText

コンテナ中のレコードに対してアイコン・テキストを戻します。**RecordID** 属性は、どのレコードを参照しているかを指示するために設定しなければなりません。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      text = %getatr('win01': 'CN1': 'GetRcdText')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CN1'      GETATR  'GETRCDTEXT' text
*
```

## GetTables

この属性を 1 に設定すると、データ・ソースに格納されているテーブルのリストが検索されます。ODBC/JDBC インターフェース・パーツは、このリストをデータ・セットとして戻します。この属性は SQL Tables ODBC API と等価です。

**FetchNext** 属性を使用して、戻されたデータ・セットからテーブル・レコードを取り出すことができます。

ODBC の場合には、リスト内の各レコードは 1 つのテーブルに対応し、次の情報を含んでいます。

| 列 | 列の名前     | データ・タイプ | コメント             |
|---|----------|---------|------------------|
| 1 | テーブル修飾子  | 文字      |                  |
| 2 | テーブル所有者  | 文字      | テーブル所有者          |
| 3 | テーブル名    | 文字      | テーブルの名前          |
| 4 | テーブル・タイプ | 文字      | テーブル・タイプを表すストリング |
| 5 | 注釈       | 文字      | テーブルの説明          |

列の説明の詳細については、ODBC API 資料を参照してください。

JDBC の場合には、次の情報が戻されます。

| 列 | 列の名前        | データ・タイプ | コメント  |
|---|-------------|---------|---|
| 1 | TABLE_CAT   | 文字      | テーブル・カタログ (NULL のこともあります)                   |
| 2 | TABLE_SCHEM | 文字      | テーブル・スキーマ (NULL のこともあります)                   |
| 3 | TABLE_NAME  | 文字      | テーブルの名前                                     |
| 4 | TABLE_TYPE  | 文字      | テーブル・タイプを表すストリング。代表的な値として TABLE、VIEW、SYSTEM |
| 5 | REMARKS     | 文字      | TABLE、ALIAS などがあります。テーブルの説明                 |

列の説明の詳細については、JDBC API に関する文書を参照してください。

## タイプ 数値

### 適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'GetTables')=1
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ODBC1'      SETATR      1          'GETTABLES'
*
```

## GetTables の例

この例は、接続されたデータ・ソースからテーブルのリストを取得して、それらのテーブルを TABLES というリスト・ボックスにリストする方法を示しています。

```
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
C      'ODBC1'      Setatr      1          'GetTables'
C      'ODBC1'      Setatr      1          'Fetch'
C      'ODBC1'      Getatr      'IsData'    Temp          1 0
*
C
C      'Tables'     Setatr      0          'RemoveItem'
*
C
C      'ODBC1'      DoW        Temp = 1
C      'ODBC1'      Setatr      3          'Column'
C      'ODBC1'      Getatr      'CharData'  TableName      20
C      'Tables'     Setatr      TableName  'AddItemEnd'
C      'ODBC1'      Setatr      1          'FetchNext'
C      'ODBC1'      Getatr      'IsData'    Temp
C
C      EndDo *
C      EndIf
```

## GnEqGrpCol

同じグループ内のすべてのデータ・ポイントが同じカラーになるように、カラーのランダム・セットを生成します。

この属性は、棒グラフを表示する時に便利です。

**タイプ** 数値

**適用可能なパーツ**

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

値は 1 にしか設定できません。

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'GR1':'GnEqGrpCol')=1
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'GR1'      SETATR  1          'GNEQGRPCOL'
```

## GnEqPntCol

指定されたデータ・ポイントのすべての値が同じカラーになるように、カラーのランダム・セットを生成します。

この属性は、円グラフを表示する時に便利です。

**タイプ** 数値

**適用可能なパーツ**

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

値は 1 にしか設定できません。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'GR1':'GnEqPntCol')=1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'GR1'      SETATR  1          'GNEQPNTCOL'
```



## GraphType

どのタイプのグラフを使用するかを指示します。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**GraphType** は、次の 1 つとすることができます。

- 1 折れ線グラフ
- 2 棒グラフ
- 3 棒グラフと折れ線グラフ
- 4 円グラフ

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      GRtype = %getatr('win01': 'GRT1': 'GraphType')
*
C          EVAL      %setatr('win01': 'GRT1': 'GraphType') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GRT1'      GETATR  'GRAPHTYPE'  'GRtype'      1 0
*
C  'GRT1'      SETATR  2          'GRAPHTYPE'
```

## GrphHiLite

項目を強調表示するかどうかを判別します。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**GrphHiLite** は、次のいずれかとすることができます。

**0** 項目は強調表示されません。

**1** 項目が強調表示されます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      lite = %getatr('win01': 'GR1': 'GrphHiLite')
*
C  lite      IFEQ      0
C          EVAL      %setatr('win01': 'GR1': 'GrphHiLite') = 1
C          ELSE
C          EVAL      %setatr('win01': 'GR1': 'GrphHiLite') = 0
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GR1'      GETATR  'GRPHHILITE'  lite          1 0
*
C  'GR1'      SETATR  1          'GRPHHILITE'
```

## GroupLabel

現行グループのラベルを示します。グループ・ラベルは、グラフが円グラフ・モードの場合には各点群の見出しとして使用され、グループ・ラベルを表示するように凡例が設定された場合には凡例の中で使用されます。

タイプ スtring

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'graph1':'GroupLabel')='July'
*
```

固定形式の例

グラフ・パーツの現在のラベルを保管し、それを新しい値に設定します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'GRAPH01'      GETATR      'GROUPLABEL'      oldlabel
C      'GRAPH01'      SETATR      'July'            'GROUPLABEL'
*
```

## Handle

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

パーツのウィンドウのハンドルを戻します。Windows API に精通している場合には、このハンドルを使用して API 機能呼び出すことができます。ODBC 処理の場合には、環境ハンドルを戻します。

たとえば、ODBC/JDBC インターフェース・パーツがデータベースに接続すると、環境ハンドルが作成され、パーツに戻されます。パーツはこのハンドルを使用して ODBC API と通信します。**Handle** 属性を使用して、プログラム内のハンドル変数を検索し、ODBC API 機能呼び出すことができます。ハンドル変数は 32 ビットの値です。この変数は、ハンドルを、長さが 10 で小数点以下の桁数が 0 である数字データ・タイプとして受け取るようにコーディングする必要があります。

```
DName+++++++ETDsFrom+++To/L+++IDc.Keywords+++++++
D hWnd          S          10I 0
```

**タイプ** 数値

**適用可能なパーツ**

次のものを除くすべてのパーツ: ActiveX、\*Component、コンポーネント参照、メニュー・バー、メニュー項目、ノートブック・ページ、キャンバス付きノートブック・ページ、サブメニュー、タイマー。

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**戻される値**

パーツのハンドルを戻します。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      hWnd = %getatr('win01': 'FW1': 'Handle')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'FW1'      GETATR  'HANDLE'      hWnd
```

## HasPrpPage

ActiveX オブジェクトがそれ自身のプロパティ・ページ・ダイアログを持っている場合には、1 の値を返します。持っていない場合には、0 を返します。

タイプ 数値

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      PPG=%getatr('win01':'AX1':'HasPrpPage')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'AX1'      GETATR      'HASPRPPAGE' PPG
*
```

## HdgBGClr

サブファイル見出しの背景カラーを決定する数値。

**HdgBGMix** 属性を使用して背景カラーを設定した場合には、戻される背景カラーが混合カラーに最も近いカラーです。

**タイプ** 数値

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

カラーは次の 1 つでなければなりません。

- 1 デフォルト
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      HBcolor = %getatr('win01': 'SFL1': 'HdgBGClr')
*
C          EVAL      %setatr('win01': 'SFL1': 'HdgBGClr') = 5
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'SFL1'    GETATR   'HDGBGCLR'   HBcolor      2 0
*
C  'SFL1'    SETATR   5             'HDGBGCLR'
```

## HdgBGMix

サブファイル見出しの背景カラーの混合を照会または設定します。

タイプ スtring

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

背景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      HBclrMx=%getatr('win01':'SFL1':'HdgBGMix')
*
C  HBclrMx  IFNE      '10:23:200'
C          EVAL      %setatr('win01':'SFL1':'HdgBGMix')='10:23:200'
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SFL1'   SETATR    '10:23:200'  'HDGBGMIX'
```



## HdgFGClr

サブファイル見出しの前景カラーを決定する数値。

**HdgFGMix** 属性を使用して前景カラーを設定した場合には、戻される前景カラーが混合カラーに最も近いカラーです。

**タイプ** 数値

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

カラーは次の 1 つでなければなりません。

- 1 デフォルト
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      HFcolor = %getatr('win01': 'SFL1': 'HdgFGCLr')
*
C          EVAL      %setatr('win01': 'SFL1': 'HdgFGCLr') = 5
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'SFL1'      GETATR  'HDGFGCLR'   HFcolor      2 0
*
C 'SFL1'      SETATR  5           'HDGFGCLR'
```

## HdgFGMix

サブファイル見出しの前景カラーの混合を照会または設定します。

タイプ スtring

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

前景カラーの混合は次の形式でなければなりません。

'red:green:blue'

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      HFc1rMx = %getatr('win01': 'SFL1': 'HdgFGMix')
*
C   HFc1rMx      IFNE      '10:23:200'
C          EVAL      %setatr('win01': 'SFL1': 'HdgFGMix')
C          = '10:23:200'
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C   'SFL1'      SETATR      '10:23:200'   'HDGFGMIX'
*
```

## HdgIdx

複数行列見出しの行を識別します。たとえば、値 1 は 1 行目を示します。この属性を **ColNumber** 属性と一緒に使用して、**HdgText** 属性で指定されたテキストを検索または変更することができます。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**HdgIdx** は 1 - 3 にすることができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'SF1':'HdgIdx')=3
C          EVAL      %setatr('win01':'SF1':'ColNumber') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          'SF1'      SETATR  3          'HDGIDX'
C          'SF1'      SETATR  2          'COLNUMBER'
*
```

## HdgText

列見出しのテキスト。このテキスト・ストリングは最大 3 行のテキストを含むことができます。これは、メッセージ・ファイルからのメッセージ ID または置き換えラベルとすることができます。このテキストを実行時に変更するには、まず **ColNumber** および **HdgIdx** 属性を設定してください。

タイプ ストリング

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'SF1':'HdgText')='HEAD1'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SF1'      SETATR  'HEAD1'      'HDGTEXT'
*
```

## Height

パーツの高さをピクセル数で指定します。

**Height** 属性は、使用されるディスプレイ解像度によって決まる最大高さを超えることができません。たとえば、480 ピクセルの高さ表示で作業中の時には、600 ピクセルのパーツの高さ表示を作成することができません。

**タイプ** 数値

**適用可能なパーツ**

|                    |               |                |
|--------------------|---------------|----------------|
| ActiveX            | カレンダー         | キャンバス          |
| チェック・ボックス          | 組み合わせボックス     | コンテナ           |
| 入力フィールド            | グラフ           | グラフィック・プッシュボタン |
| グループ・ボックス          | 水平方向のスクロール・バー | イメージ           |
| Java Bean          | リスト・ボックス      | メディア・パネル       |
| メッセージ・サブファイル       | 複数行編集         | ノートブック         |
| ODBC/JDBC インターフェース | 外枠            | 進行状況バー         |
| プッシュボタン            | ラジオ・ボタン       | スライダー          |
| スピン・ボタン            | 静的テキスト        | サブファイル         |
| 縦方向のスクロール・バー       | ウィンドウ         |                |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

パーツの高さ (ピクセル数)

**フリー・フォームの例**

次の通り、パーツの高さを変更します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1++++++0pcode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C                               EVAL    howtall = %getatr('win01': 'PB1': 'Height')
C   howtall                      add      howtall
C                               EVAL    %setatr('win01': 'PB1': 'Height') = howtall
*
```

**固定形式の例**

次の通り、プッシュボタンの 100 ピクセル数を短くします。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1++++++0pcode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
```

```
C    'PB1'      GETATR  'HEIGHT'  OldHeight  2 0
C    OldHeight  SUB     100      NewHeight  2 0
C    'PB1'      SETATR  NewHeight 'HEIGHT'
```

\*

## HelpEnable

ユーザーがプッシュボタンまたはグラフィック・プッシュボタンを押した時に、ヘルプ情報が表示されます。これが最も有用なのはウィンドウ用の一般ヘルプを表示したい場合です。通常、パーツのヘルプを表示するためには、そのパーツがフォーカスされている時に F1 キーを押してください。

**注:** ユーザーがプッシュボタンまたはグラフィック・プッシュボタン・パーツにフォーカスを合わせなければ、ヘルプを表示できません。

**タイプ** 数値

**適用可能なパーツ**

|                |         |  |
|----------------|---------|--|
| グラフィック・プッシュボタン | プッシュボタン |  |
|----------------|---------|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。ヘルプの表示スタイルを選択することによって、これを行ってください。

**使用可能な値**

**HelpEnable** は、次のいずれかとすることができます。

**0** このパーツにヘルプは使用可能ではありません

**1** このパーツにヘルプが使用可能です



## フリー・フォームの例

次のように、パーツの「ヘルプ使用可能」状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      rc = %getatr('win01': 'PB1': 'HelpEnable')
*
C      rc                       IFEQ       0
C                               EVAL      %setatr('win01': 'PB1': 'HelpEnable') = 1
*
C                               ELSE
C                               EVAL      %setatr('win01': 'PB1': 'HelpEnable') = 0
C                               ENDIF
*
```

## 固定形式の例

入力フィールドの「ヘルプ使用可能」状況を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'PB1'                     GETATR   'HELPEENABLE'  HasHelp      1 0
C      HasHelp                   IFEQ       1
C      'PB1'                     SETATR   0              'HELPEENABLE'
*
C                               ELSE
C      'PB1'                     SETATR   1              'HELPEENABLE'
C                               ENDIF
*
```

## HelpWindow

特定のヘルプ・パネル・ウィンドウが表示されます。helpid の値は、ヘルプ・ファイルに指定されたヘルプ ID です。0 のヘルプ ID 値が使用された場合には、Windows スタイルの "ヘルプ・トピック" ダイアログが表示されます。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

helpid

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('*Component': '*Component': 'HelpWindow')=
C          12
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C '*Component' SETATR 12          'HELPWINDOW'
*
```

## Hidden

サブファイル内の列が可視であるかどうかを決定します。影響を受ける列を示すように、**ColNumber** 属性を設定する必要があります。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**Hidden** は、次のいずれかとすることができます。

**0** 列は隠されません

**1** 列は隠されます

フリー・フォームの例

列の Hidden 状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      rc = %getatr('win01': 'SF1': 'Hidden')
*
C   rc      IFEQ      0
C          EVAL      %setatr('win01': 'SF1': 'Hidden') = 1
*
C          ELSE
C          EVAL      %setatr('win01': 'SF1': 'Hidden') = 0
C          ENDIF
*
```

固定形式の例

属性の Hidden 状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'SF1'    GETATR   'HIDDEN'  IsHidden  1 0
C   IsHidden IFEQ     1
C   'SF1'    SETATR   0          'HIDDEN'
*
C          ELSE
C   'SF1'    SETATR   1          'HIDDEN'
C          ENDIF
*
```

## HighLight

指定されたパーツを強調表示するかどうかを決定します。強調表示パーツは、マウス・ポインターがボタン制御を超えている間に左マウス・ボタンが押されたボタンと同じ表現です。

タイプ 数値

適用可能なパーツ

|            |                |         |
|------------|----------------|---------|
| チェック・ボックス* | グラフィック・プッシュボタン | プッシュボタン |
| ラジオ・ボタン*   |                |         |

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

命令コード

|        |          |
|--------|----------|
|        | 実行時に操作可能 |
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**HighLight** は、次のいずれかとすることができます。

- 0** パーツは強調表示されません。
- 1** パーツが強調表示されます。

## フリー・フォームの例

次のように、パーツの HighLight 状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      rc = %getatr('win01': 'PB1': 'HighLight')
*
C      rc                       IFEQ      0
C                               EVAL      %setatr('win01': 'PB1': 'HighLight') = 1
*
C                               ELSE
C                               EVAL      %setatr('win01': 'PB1': 'HighLight') = 0
C                               ENDIF
*
```

## 固定形式の例

次のように、パーツの HighLight 状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'PB1'                     GETATR   'HIGHLIGHT'  IsHighlighted  1 0
C      IsHighlighted             IFEQ      1 0
C      'PB1'                     SETATR   0              'HIGHLIGHT'
*
C                               ELSE
C      'PB1'                     SETATR   1 0              'HIGHLIGHT'
C                               ENDIF
*
```

## HitItem

**UnderPoint** を設定した後、**HitItem** は指定されたポイントの下にあったグラフの外観を指示します。

バーまたはスライスがヒットした場合には、**DataGroup** および **DataPoint** が設定されます。

グラフが円グラフ・モードで、グループ・ラベルがヒットした場合には、**DataGroup** が設定されます。

**注:** この属性は、Java アプリケーションの場合には適用できません。

**タイプ** 数値

**適用可能なパーツ**

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**戻される値**

次の 1 つです。

- 0** ヒットしていません。
- 1** バーがヒットしました。
- 2** スライスがヒットしました。
- 3** 凡例がヒットしました。
- 4** タイトルがヒットしました。
- 5** X 軸ラベルがヒットしました。
- 6** Y 軸ラベルがヒットしました。
- 7** グループ・ラベルがヒットしました。

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      itemhit = %getatr('win01': 'GRA1': 'HitItem')
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'GRA1'      GETATR   'HITITEM'      itemhit      1 0
*
```

## HlitPoints

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

**HlitPoints** が設定されると、**DataGroup** によって指示されたグループ内のすべてのデータ項目が強調表示されます。これは、グラフが棒グラフ・モードの場合にだけ当てはまります。

**タイプ** 数値

**適用可能なパーツ**

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

使用可能な値は次の通りです。

**0** 現行データ・グループは強調表示されません

**0 以外の任意の値**

現行データ・グループが強調表示されます

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'GP1': 'HlitPoints') = 20
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'GP1'      SETATR  20          'HLITPOINTS'
```



## HostName

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

VisualAge RPG アプリケーションを実行中のワークステーションのホスト名を *host\_name ip\_address* の形式で戻します。たとえば、次のようになります。

is267sys 9.21.230.171

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      HostIs=%getatr('*component':'*component':'HostName')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      '*component' GETATR  'HOSTNAME'  HOSTIS
*
```

## HRule

水平方向の罫線を表示します。

タイプ 数値

適用可能なパーツ

|       |        |  |
|-------|--------|--|
| カレンダー | サブファイル |  |
|-------|--------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**HRule** は、次のいずれかとすることができます。

- 0** 罫線を消します
- 1** 罫線を表示します

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Hzrule = %getatr('win01': 'SF1': 'HRule')
*
C  Hzrule  IFEQ      0
C          EVAL      %setatr('win01': 'SF1': 'HRule') = 1
C          ELSE
C          EVAL      %setatr('win01': 'SF1': 'HRule') = 0
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SF1'   GETATR   'HRULE'   Hzrule       1 0
*
C  'SF1'   SETATR   1         'HRULE'
```

## IconHandle

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

「ウィンドウのハンドル」アイコンを戻します。

**タイプ** 数値

**適用可能なパーツ**

|       |              |  |
|-------|--------------|--|
| ウィンドウ | キャンバス付きウィンドウ |  |
|-------|--------------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      ichW=%getatr('win01':'FW1':'IconHandle')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'FW1'          GETATR  'ICONHANDLE' ICHW
*
```

## Index

パーツのリスト部分の索引値を確立します。この値は特定の属性を使用する時に影響を受ける項目を決定します。たとえば、**GetItem** 属性を使用して項目を検索する前に、その項目の **Index** 属性を設定しなければなりません。

タイプ 数値

適用可能なパーツ

|           |          |              |
|-----------|----------|--------------|
| 組み合わせボックス | リスト・ボックス | メッセージ・サブファイル |
| サブファイル    |          |              |

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

索引値

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'LB1': 'Index') = 4
*
C          EVAL      index = %getatr('win01': 'LB1': 'Index')
```

固定形式の例

次の通り、リスト・ボックスの現行の索引値を保管した後で、それを変更します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          'LB1'      GETATR  'INDEX'      OldIndex      2 0
C          'LB1'      SETATR  NewIndex     'INDEX'      2 0
*
```

## InPlace

注: この属性は Windows にのみ適用されます。

**InPlace** は、ビデオ・ファイルの再生位置を指定するために使用します。 **InPlace** を 1 に設定すると、ビデオはメディア・パーツの位置で再生されます。0 に設定すると、ファイルは別のウィンドウで再生されます。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| メディア |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**InPlace** は、次のいずれかとすることができます。

**0** このファイルは別のウィンドウでオープンされます

**1** このファイルはメディア・パーツの位置でオープンされます

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'AUD1': 'InPlace') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AUD1'      SETATR  1          'INPLACE'
*
```

## InsertItem

\* **制約事項:** この属性は Java アプリケーションではサポートされていません。

項目をパーツのリスト部分に挿入します。パーツの **Sequence** 属性の現行設定は、項目をリストに挿入する位置を判別します。

索引方式を使用して項目をリストに挿入する場合には、その索引値にある項目とその項目の後にあるすべての項目は、そのリストで 1 つ下の位置に移動します。

**タイプ** スtring

**適用可能なパーツ**

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

リストに挿入する項目

**フリー・フォームの例**

次の通り、リスト・ボックスに項目を挿入します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'LB1': 'Index') = 3
C          EVAL      %setatr('win01': 'LB1': 'InsertItem') = 'new'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      SETATR  3          'INDEX'
C  'LB1'      SETATR  'new'      'InsertItem'
*
```

## InsertLine

指定されたストリングを、複数行の編集で現行の **LineNumber** 属性によって指定された位置に挿入します。挿入された行の後のすべての行は 1 行下にシフトされます。

タイプ ストリング

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

複数行編集に追加するストリング

フリー・フォームの例

次の通り、複数行編集でオフセット 10 の位置にテキストを挿入します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'LineNumber') = 10
C          EVAL      %setatr('win01': 'MLE1': 'InsertLine') = 'text'
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MLE1'      SETATR  10          'LINENUMBER'
C  'MLE1'      SETATR  'text'     'INSERTLINE'
*
```

## InsertMode

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

パーツの入力フィールド部分が挿入モードになっているかどうかを判別します。挿入モードがオンになっている時には、フィールドに文字を入力するとカーソルの右側の文字がシフトされます。オフになっている時には、文字を入力すると既存の文字がオーバーライドされます。

**タイプ** 数値

**適用可能なパーツ**

|         |  |  |
|---------|--|--|
| 入力フィールド |  |  |
|---------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**使用可能な値**

**InsertMode** は、次のいずれかとすることができます。

- 0** 入力フィールドは挿入モードになっていません。
- 1** 入力フィールドは挿入モードになっています。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      mode = %getatr('win01': 'EF1': 'InsertMode')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'EF1'      GETATR  'INSERTMODE' InsertMode      1 0
*
```



## InsertRow

この属性を 1 に設定すると、新しい行がテーブルに追加されます。その新しい行が追加される前に、まずデータが、テーブル列にバインドされているプログラム・フィールドから移動されます。

**注:** Java アプリケーションでは、この属性に適切な JDBC 2.0 承諾ドライバーが必要です。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'InsertRow')=1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'ODBC1'   SETATR   1          'INSERTROW'
```

## InsertText

指定されたテキストを複数行編集の現在のカーソル位置に挿入します。

タイプ スtring

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

挿入されるテキスト・String

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'MLE1': 'InsertText') = 'New'
*
```

固定形式の例

次の通り、複数行の編集で現行カーソル位置にテキストを追加します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'MLE1'      SETATR  'New'      'InsertText'
*
```

## Interval

タイマー・パーツのインターバル値を決定します。この属性を **Multiplier** 属性と一緒に使用して、**Tick** イベントを生成する前に時間が経過するミリ秒数を判別します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| タイマー |  |  |
|------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

インターバル値 (ミリ秒数)

インターバルの最大値は 65535 です。

インターバルの最小値は 100 です。

フリー・フォームの例

次の通り、タイマー・パーツの間隔値を 1 秒 (1000 ミリ秒) に変更します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      oldint=%getatr('win01':'TIMER01':'Interval')
*
C          EVAL      %setatr('win01':'TIMER01':'Interval')=1000
*
```

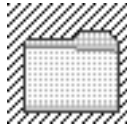
固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'TIMER01'  GETATR  'INTERVAL'  OldInt          6 0
*
C  'TIMER01'  SETATR  1000        'INTERVAL'
```

## InUse

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

コンテナのオブジェクトが、現在使用中として強調されているかどうかを照会します。使用中強調のオブジェクトはオープン設計ウィンドウ内にあり、次の通り、そのアイコンの後の陰影枠で表示されます。



**RecordID** 属性は、どのレコードを参照しているかを指示するために設定しなければなりません。

**タイプ** 数値

**適用可能なパーツ**

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**InUse** は、次のいずれかとすることができます。

**0**      使用中ではありません

**1**      使用中

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C                               EVAL    %setatr('win01': 'CN1': 'InUse') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'CN1'           SETATR   1           'INUSE'
```

## InvName

**\*制約事項:** Java アプリケーションにのみ適用されます。

ポップアップ・メニューを呼び出すかまたは固定するパーツの名前を指定します。  
この属性は、**InvPname**、**X**、および **Y** 属性と一緒に使用されます。

**タイプ** スtring

**適用可能なパーツ**

|             |  |  |
|-------------|--|--|
| ポップアップ・メニュー |  |  |
|-------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'POP1':'InvName')='INVN1'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C      'POP1'      SETATR      'INVN1'      'INVNAME'
*
```

## InvPName

**\*制約事項:** Java アプリケーションにのみ適用されます。

ポップアップ・メニューを呼び出すかまたは固定するパーツの親名を指定します。  
この属性は、**InvName**、**X**、および **Y** 属性と一緒に使用されます。

**タイプ** スtring

**適用可能なパーツ**

|             |  |  |
|-------------|--|--|
| ポップアップ・メニュー |  |  |
|-------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'POP1':'InvPName')='INVPN1'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C      'POP1'      SETATR      'INVPN1'      'INVPNAME'
*
```

## IsData

**Fetch**、**FetchNext**、または **FetchPrior** 演算命令でデータが戻されたかどうかを示します。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

戻される値

**0** データが戻されなかったことを示します。

**1** データが戻されたことを示します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Flag=%getatr('win01':'ODBC1':'IsData')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'ODBC1'      GETATR  'ISDATA'      FLAG
```

## Item

DDE 項目は、DDE 会話の中で参照される特定の単位のデータです。DDE 項目の例はスプレッドシート・アプリケーションのセルです。DDE 項目の形式はサーバー・アプリケーションによって決定されます。この形式については、サーバー・アプリケーションの資料を参照してください。スプレッドシートの例では、これは「A1」などとなります。

次のテーブルには、DDE 会話の 1 つの項目とできるパーツ、および **Poke** または **Request** イベント中に参照される属性が要約されます。他のパーツ・タイプが参照されている場合には、操作は無視されます。

| パーツ・タイプ   | Poke     | Request       |
|-----------|----------|---------------|
| メディア      | Filename | Filename      |
| チェック・ボックス | Label    | Label         |
| 入力フィールド   | Text     | Text          |
| イメージ      | Filename | Filename      |
| リスト・ボックス  | なし       | Selected item |
| スライダー     | Value    | Value         |
| 静的テキスト    | Label    | Label         |
| タイマー      | Count    | Count         |

### タイプ ストリング

#### 適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

### 使用可能な値

DDE 会話で参照されているデータの単位。

サーバー・プログラムが VisualAge RPG コンポーネントである場合には、**Item** は参照されるパーツの名前です。これは、次の通り指定されます。

'window|part'



### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'DDE1': 'Item') = 'A1'
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'DDE1'      SETATR      'A1'      'ITEM'
*
```

## ItemCount

サブファイル列の値のリスト中の項目の数を返します。最初に、処理するランを識別する **ColNumber** 属性を設定します。サブファイル列には、妥当性検査としての値がなければなりません。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      ItCnt = %getatr('win01': 'SF1': 'ItemCount')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'SF1'      GETATR  'ITEMCOUNT'  ITCNT
*
```

## ItemKey

リスト内の項目のキー値を戻します。どの項目が参照されているかを示すために、まず **Index** 属性を設定する必要があります。

**注:** キーが定義されていないと、**ItemKey** はヌル・ストリングを戻します。

**タイプ** ストリング

**適用可能なパーツ**

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**注:** 組み合わせボックスに対する **ItemKey** は、プロパティ・ノートブックを使用して設定することができます。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      keyitem = %getatr('win01': 'LB1': 'ItemKey')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      GETATR  'ITEMKEY'  keyitem
*
```

## Label

プッシュボタンに表示されるテキストなど、パーツと関連したテキスト・ストリングです。

**label** 属性は、次の形式の 1 つとすることができます。

- 入力された通りのストリング
- 置換ラベル
- メッセージ・ファイルからのメッセージ ID

コンテナー列のタイトルがブランクの場合には、コンテナーは正しく表示されません。これを訂正するためには、タイトル中のすべてのブランクを下線文字 ( \_ ) で置き換えてください。

Windows では、プロパティ・ノートブック でタイトル・バーが使用不可になっていても、タイトル・バーが表示されます。タイトル・バーの設定を選択しない時には、タイトル・バー区域でマウスを使用してウィンドウを移動することはできませんが、ウィンドウのクローズにはキーボードを使用しなければなりません。

**タイプ** ストリング

**適用可能なパーツ**

|           |         |           |
|-----------|---------|-----------|
| チェック・ボックス | コンテナー   | グループ・ボックス |
| メニュー項目    | プッシュボタン | ラジオ・ボタン   |
| 静的テキスト    | ウィンドウ   |           |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。「一般」タブを使用し、「ラベル」または「タイトル」フィールドに値を指定することによって、これを行ってください。

**使用可能な値**

パーツ・ラベル

## フリー・フォームの例

次の通り、プッシュボタンのラベルを変更します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      checked = %getatr('win01': 'CB1': 'Checked')
*
C      checked                 IFEQ        0
C                               EVAL      %setatr('win01': 'PB1': 'Label') = 'No'
*
C                               ELSE
C                               EVAL      %setatr('win01': 'PB1': 'Label') = 'Yes'
C                               ENDF
*
```

## 固定形式の例

次の通り、プッシュボタンのテキストを変更します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'PB1'                   GETATR   'LABEL'      01dText      20
C      'PB1'                   SETATR   'Cancel'    'LABEL'
```

## LabelPlace

指示されたデータ項目のラベルを表示するグラフ上の位置を指定します。

グラフが円グラフ・モードの時には、この属性は適用されません。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**LabelPlace** は、次のいずれかとすることができます。

- 0 データ項目はラベルを持つことはできません
- 1 ラベルは X 軸に表示しなければなりません
- 2 このラベルは、項目の値が正の場合にはバーの上に、項目の値が負の場合にはバーの下に表示しなければなりません。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('Win1': 'Gra1': 'LabelPlace') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'Gra1'          SETATR  2          'LABELPLACE'
*
```

## Left

このパーツが入っているオブジェクトまたは画面の座標を左端からのピクセル数で指定します。

**タイプ** 数値

**適用可能なパーツ**

|                    |               |                |
|--------------------|---------------|----------------|
| ActiveX            | アニメーション制御     | カレンダー          |
| キャンバス              | チェック・ボックス     | 組み合わせボックス      |
| コンポーネント参照          | コンテナー         | DDE クライアント     |
| 入力フィールド            | グラフ           | グラフィック・プッシュボタン |
| グループ・ボックス          | 水平方向のスクロール・バー | イメージ           |
| Java Bean          | リスト・ボックス      | メディア           |
| メディア・パネル           | 複数行編集         | ノートブック         |
| ODBC/JDBC インターフェース | 外枠            | 進行状況バー         |
| プッシュボタン            | ラジオ・ボタン       | スライダー          |
| スピン・ボタン            | 静的テキスト        | サブファイル         |
| タイマー               | 縦方向のスクロール・バー  | ウィンドウ          |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

ピクセルの数

**フリー・フォームの例**

次の通り、そのウィンドウでパーツを左に移動します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL    leftside = %getatr('win01': 'image1': 'Left')
C          EVAL    %setatr('window': 'part': 'Left') =
                  leftside - 120
*
```

## 固定形式の例

次の通り、プッシュボタン 200 ピクセルを右に移動します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'PB01'      GETATR  'LEFT'      OldLeft      2 0
C   OldLeft     ADD      200        NewLeft      2 0
C   'PB01'      SETATR  NewLeft     'LEFT'
```



## LegendType

グラフ・パーツで使用される凡例のタイプを示します。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

この値は次の 1 つでなければなりません。

- 0 凡例なし
- 1 ポイントのみ
- 2 グループのみ
- 3 すべての項目

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('Win1': 'Gra': 'LegendType') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'Gra1'      SETATR  2          'LegendType'
```

# Length

注: この属性は Windows にのみ適用されます。

ファイルの長さをミリ秒で指定します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| メディア |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+Hi LoEq----
*
C          EVAL      Len = %getatr('win01': 'AUD01': 'Length')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+Hi LoEq----
*
C 'AUD01'   GETATR  'LENGTH'   Len          6 0
*
```

## LineNumber

**InsertLine** 属性によって複数行編集で新規行を挿入する行を判別します。この値は、ゼロより大きく、現行の **NbrOfLines** 属性と等しいかこれより小さくなくてはなりません。これが現行の **NbrOfLines** 属性より大きい場合には、新規の行は最後の行の後に挿入されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

行番号

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      oldline = %getatr('win01': 'MLE01': 'LineNumber')
C          EVAL      %setatr('win01': 'MLE01': 'LineNumber') = 5
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MLE01'  GETATR  'LINENUMBER'  OldLine      2 0
C  'MLE01'  SETATR  5              'LINENUMBER'
*
```

## LineText

**LineNumber** で指定された複数行編集行のテキストの行を検索します。

タイプ スtring

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

使用可能な値

行のテキスト

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      lines = %getatr('win01': 'MLE1': 'NbrOfLines')
C          Do        lines      ix
C          EVAL      %setatr('win01': 'MLE1': 'LineNumber') = ix
C          EVAL      nextline= %getatr('win01': 'MLE1': 'LineText')
C          ...
C          EndDo *
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'MLE1'      GETATR  'LINETEXT'  line
*
```

## LookNFeel

\*制約事項: Java 専用の属性。

Java アプリケーションでスイングの見栄え (PLAF) 機構を使用して、GUI がどのような "見栄え" になるかを制御できるようにします。

タイプ ストリング

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**LookNFeel** は、次のいずれかとすることができます。

**Look** ストリング値

**Metal** javax.swing.plaf.metal.MetalLookAndFeel

**Motif** com.sun.java.swing.plaf.motif.MotifLookAndFeel

**Windows**

com.sun.java.swing.plaf.windows.WindowsLookAndFeel

フリー・フォームの例

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          eval      %setatr('*Component': '*Component':
C          'LookAndFeel')=
C          'com.sun.java.swing.plaf.motif.' +
C          'MotifLookAndFeel'
*
```

## Magnify

表示イメージのスケールを制御するイメージ・パーツ上に、拡大スライダーの値をパーセントで指定します。ゼロの値はイメージ・スケールをイメージ・パーツに合わせます。

**タイプ** 数値

**適用可能なパーツ**

|      |  |  |
|------|--|--|
| イメージ |  |  |
|------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

拡大スライダーの値は、次の 1 つにすることができます。

**0** スケール合わせ

**25 - 200**

パーセントでの拡大値

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win1': 'Img1': 'Magnify') = 50
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'IMG1'      SETATR  50          'MAGNIFY'
*
```

## MapViewCol

サブファイル・データ列 (**ColNumber** 属性で指定) を **ViewColumn** 属性で指示した順序で表示するには、この属性を 1 に設定します。デフォルトでは、データ列が「サブファイル」プロパティ・ノートブックの「フィールド・リスト」ページにリストされたのと同じ順序で表示されます。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

- 1 サブファイル・データ列を ViewColumn 属性によって示された順序で表示します。

フリー・フォームの例

データ列 2 (ColNumber) を表示可能列 4 (ViewColumn) として表示します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'SF1': 'ColNumber') = 2
C          EVAL      %setatr('win01': 'SF1': 'ViewColumn') = 4
C          EVAL      %setatr('win01': 'SF1': 'MapViewCol') = 1
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq----
*
C  'SF1'      SETATR  2      'COLNUMBER'
C  'SF1'      SETATR  4      'VIEWCOLUMN'
C  'SF1'      SETATR  1      'MAPVIEWCOL'
*
```

## Masked

属性を設定した時には、入力フィールドに入力すると、入力した文字ではなくて、アスタリスク (\*) 文字が表示されます。この属性が最も頻繁に使用されるのはユーザーがパスワードまたはアカウント番号のような機密データを入力する時です。

タイプ 数値

適用可能なパーツ

|         |  |  |
|---------|--|--|
| 入力フィールド |  |  |
|---------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | いいえ      |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

フィールドのパスワード保護状態は、次の 1 つにすることができます。

**0** フィールドはマスクされません。

**1** フィールドはマスクされます。

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      isMasked = %getatr('win01': 'EF1': 'Masked')
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'EF1'      GETATR  'MASKED'    isMasked      1 0
*
```



## Maximum

スライダーまたはスピン・ボタンから戻ることができる最大値を決定します。

この属性を使用できるのは、数字スピン・ボタンの場合だけです。

最大値は 65535 です。

**タイプ** 数値

**適用可能なパーツ**

|       |         |  |
|-------|---------|--|
| スライダー | スピン・ボタン |  |
|-------|---------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

パーツの最大値

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      oldmax=%getatr('window':'SLIDER01':'Maximum')
*
C          EVAL      %setatr('window':'SLIDER01':'Maximum')=100
*
```

**固定形式の例**

次の通り、スライダーの最大値を変更します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          'SLIDER01'  GETATR  'MAXIMUM'  OldMax      3 0
C          'SLIDER01'  SETATR  100        'MAXIMUM'
*
```

## Method

組み込む ActiveX メソッドを、そのメソッドに渡される値とともに指定します。プロパティおよびメソッドを設定することによって、ActiveX パーツを制御することができます。

Netscape 4.x などのある種の ActiveX コントロールが活動化されていると、アプリケーション・モーダル・ダイアログが表示され、GUI Designer または VARPG アプリケーションの下では隠される場合があります。アプリケーションまたは GUI Designer がロック状態の症状を見せるため、このことに注意してください。

ストリング値の構文は次の通りです。

```
method_name, value1, value2, ...
```

### 制約事項

メソッドおよびプロパティ・パラメーターは、数値 (すなわち、整数および浮動)、ストリング (char\*)、およびポインター (PTR) データ・タイプに制限されています。フォント、カラー、ユーザー定義などのデータ・タイプはサポートされていません。

WDSC V5.X 以前のリリースでは、ActiveX メソッドを呼び出すときに、ポインター・タイプ・パラメーターを渡すことができません。

メソッドへポインター・タイプ・パラメーターを渡すためのサポートが、WDSC V5.1 で追加されました。これは、次のように行います。

VARIANT 構造へのポインターをコントロールに渡すと想定します。また、ActiveX コントロール DieRoll (ダイス) があると仮定します。これは、メソッドを持ちます。

```
NewRoll(PTR to VARIANT number)
```

ダイスが呼び出されると、ダイスは「number」ポインターによって示された、構造内の値を確認し、その数の点を表示します。

まず、VARIANT 構造を調べます。(注: 詳しくは、Microsoft の ActiveX コントロールの資料を参照してください。)

```
struct VARIANT {
    VARTYPE vt; //Indicate data type of this structure. VT_I2(i.e., 2) is an integer.
    unsigned short wReserved1;
    unsigned short wReserved2;
    unsigned short wReserved3;
    union {
        unsigned char bVal; // VT_UI1
        short iVal; // VT_I2
        long lVal; // VT_I4
        floatfltVal; // VT_R4
        double dblVal; // VT_R8
        VARIANT_BOOL xbool; // VT_BOOL
        DATE date; // VT_DATE
        BSTR bstrVal; // VT_BSTR
        short FAR* piVal; // VT_BYREF|VT_I2
        long FAR* plVal; // VT_BYREF|VT_I4
        ... ..
    }
}
```

この場合、「vt」メンバーを 2 に設定する必要があり (定数 VT\_I2)、union メンバー iVal を 1 から 6 までの値 (ダイスの点の数) に設定する必要があります。

```

d pview      s      10u 0 based(p)
d ptr        s      *
d str1       s      10
d
d VARIANT    DS
d type       1      2b 0 inz(x'02')          2=VT_I2
d reserved1  3      4b 0 inz(x'00')
d reserved2  6      6b 0 inz(x'00')
d reserved3  7      8b 0 inz(x'00')
d iVal       9      10b 0 inz(x'06')          6 dots
d padding1   11     12b 0 inz(x'00')
d padding2   13     16b 0 inz(x'0000')

```

```

C*Rool dice to '6':
C          EVAL      PTR= %ADDR(VARIANT)
C*Convert the pointer to a string, and pass it to the method call.
C          eval      p=%addr(ptr)
C          eval      str1=%char(pview)
C          eval      method='NewRoll,'+str1
C          eval      %setatr(%window:'ocx2':'method')=
C          method

```

### タイプ スtring

#### 適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

#### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

#### フリー・フォームの例

```

*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'AX1':'Method')='PN1'
*

```

#### 固定形式の例

```

*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'AX1'      SETATR 'PN1'      'METHOD'
*

```

## Method の例

次の例では、ActiveX パーツの DisplayText メソッドが呼び出され、単一値を渡されます。メソッドの数字の値が必要な場合には、同等の文字を渡してください。

```
D dspText          S          Inz('DisplayText,New Text')
*
C   'MSCAL' Setatr dspText  'Method'
```

## Minilcon

この属性は、コンテナ・パーツのアイコンの表示を判別します。この属性が設定されると、アイコンはコンテナ・パーツでミニアイコンとして表示されます。この属性が設定されないと、アイコンは通常のアイコンとして表示されます。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**Minilcon** は、次のいずれかとすることができます。

**0** アイコンは通常のサイズで表示されます。

**1** アイコンはミニアイコンとして表示されます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'Cont1': 'Miniicon') = 0
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C 'Cont1'      SETATR  0          'MINIICON'
*
```

## Minimum

スライダーまたはスピン・ボタンから戻すことができる最小値を決定します。

この属性を使用できるのは、数字スピン・ボタンの場合だけです。

最小値は -32767 です。

**タイプ** 数値

**適用可能なパーツ**

|       |         |  |
|-------|---------|--|
| スライダー | スピン・ボタン |  |
|-------|---------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

パーツの最小値

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      oldmin = %getatr('window': 'SLIDER01': 'Minimum')
*
C          EVAL      %setatr('window': 'part': 'Minimum') = 0
*
```

**固定形式の例**

次の通り、スライダーの最小値を変更します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          'SLIDER01'  GETATR  'MINIMUM'  OldMin      3 0
C          'SLIDER01'  SETATR   0          'MINIMUM'
*
```

## Mode

ビデオ・ファイルまたは GIF ファイルを再生するかどうかを決定します。

タイプ 数値

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| アニメーション制御 |  |  |
|-----------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**Mode** は、次のいずれかとすることができます。

- 1 ファイルの再生を開始します
- 2 ファイルの再生を停止します

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'WIN1': 'Mode') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'WIN1'      SETATR  1          'MODE'
```

## Month

現在選択されている月を示します。これが設定されると、カレンダー・パーツ内の選択された日付が、変更を反映するように更新されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

月

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'Month') = 02
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'CAL1'      SETATR      02      'Month'
*
```



## MonthArrow

ユーザーが月または年全体を逆方向または正方向にスクロールできるように、月 (< または >) あるいは年 (<< または >>) 矢印が使用可能であるかどうかを指示します。矢印がなければ、ユーザーには現在表示されている月または年を変更する手段がなくなります。

**タイプ** 数値

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**MonthArrow** は、次のいずれかとすることができます。

**0** 矢印は表示されません

**1** 矢印は表示されます

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      isshown = %getatr('win01': 'CAL1': 'MonthArrow')
*
C          EVAL      %setatr('win01': 'CAL1': 'MonthArrow') = 1
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'CAL1'    GETATR   'MONTHARROW'  isshown
*
```

## MonthIdx

現在のカレンダー・ページに表示されない日付に関するデータをユーザーが設定できるようにします。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

索引値

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'MonthIdx') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'CAL1'      SETATR      2          'MonthIdx'
*
```

## MonthLen

月の名前を表示するための形式を示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**MonthLen** は、次のいずれかとすることができます。

- 1 フルネーム
- 2 省略された 3 文字の名前
- 3 2 個の数字
- 4 1 個の数字しかない月の場合にゼロが埋め込まれる 2 個の数字

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'CAL1': 'MonthLen') = 3
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'CAL1'      GETATR  'MONTHLEN'   MonthLen           1 0
*
```

## Mouselcon

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

マウス・ポインターの形状のために使用するアイコン・ファイルの名前を指定します。この属性を使用する前に、**MouseShape** 値を 99 に設定してください。

ファイルはカーソル (.CUR) ファイルまたはアイコン (.ICO あるいは .ANI) ファイルとすることができます。

**タイプ** スtring

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'WIN1':'MouseIcon')=Myicon
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'WIN1'      SETATR  MYICON      'MOUSEICON'
*
```

## MouseShape

親またはその子のウィンドウ・フレームにポインターが位置付けられた時に、そのマウス・ポインターの形状を変更します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**MouseShape** は、次のいずれかとすることができます。

- 0 デフォルト
- 1 矢印アイコン
- 2 砂時計アイコン
- 3 停止または正規外アイコン
- 4 砂時計を持つ矢印 (アプリケーション開始中)
- 5 十字線
- 6 I 字形
- 7 上矢印
- 8 北東/南西矢印
- 9 北/南矢印
- 10 北西/南東矢印
- 11 東/西矢印
- 12 全面矢印 (北/南/東/西)
- 13 ポインティング・ハンド・カーソル
- 99 **Mouselcon** 属性によって指定された通りのユーザー定義アイコン。(この値は、Windows アプリケーションにのみ適用されます。)

**MouseShape** 属性は永続的なものではなく、ウィンドウ・フレームに適用されるだけです。マウス・ポインターの形状をデフォルトから変更した場合には、この属性を **0** にリセットしなければ、デフォルトのポインター形状を使用することはできません。

#### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('Win1': 'Win1': 'MouseShape') = 2
*
```

#### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'Win1'      SETATR      2          'MOUSESHAPE'
*
```

## MsgData

メッセージ・テキストに現れる変数に使用する置換テキストを指定します。複数の変数があるメッセージの場合には、それぞれの置換テキスト・ストリングはスペースで区切ります。置換テキストの一部である空白を組み込むためには、下線 ( \_ ) 文字を使用してください。たとえば、メッセージ変数 %1 および %2 のテキスト・ストリング "A B C" および "CustNo" を置き換えるためには、次の通りメッセージ・データを指定してください。

```
'A_B_C CustNo'
```

タイプ ストリング

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

メッセージ変数の置換ストリング

フリー・フォームの例

次の通り、メッセージ 0001 のメッセージ・テキストを表示して、置換データの CustNo を使用してください。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('*Component': '*Component': 'MsgID')=0001
C          EVAL    %setatr('*Component': '*Component': 'MsgData')=CustNo
C          EVAL    text = %getatr('*Component': '*Component': 'MsgText')
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  '*COMPONENT' SETATR  0001      'MSGID'
C  '*COMPONENT' SETATR  CustNo    'MSGDATA'
C  '*COMPONENT' GETATR  'MSGTEXT' text
*
```

## MsgFile

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

コンポーネントのメッセージ・ファイルにメッセージ ID が見つからない時に使用するデフォルトのメッセージ・ファイルを指定します。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

デフォルトのメッセージ・ファイルの完全名の *filename.xml* です。ファイルの検索には現在のパスが使用されます。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('*Component': '*Component': 'MsgFile')
C          = Fn
C          EVAL    Mf = %getatr('*Component': '*Component':
C          'MsgFile')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C  '*COMPONENT' SETATR  Fn          'MSGFILE'
*
C  '*COMPONENT' GETATR  'MSGFILE'  Mf
*
```



# MsgID

コンポーネント・メッセージ・ファイルのメッセージ番号を指定します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

メッセージ番号

フリー・フォームの例

次の通り、メッセージ 0001 のメッセージ・テキストを表示して、置換データの CusNo を使用してください。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('*Component':'*Component':'MsgID')=0001
C          EVAL    %setatr('*Component':'*Component':'MsgData')=CusNo
C          EVAL    text=%getatr('*Component':'*Component':'MsgText')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  '*COMPONENT' SETATR 0001 'MSGID'
C  '*COMPONENT' SETATR CusNo 'MSGDATA'
C  '*COMPONENT' GETATR 'MSGTEXT' text
*
```

## MsgSubText

メッセージ・サブファイル・パーツのメッセージ置換テキストを定義します。  
**AddMsgID** 属性または **AddMsgText** 属性を使用して、メッセージ・サブファイル・パーツにメッセージを追加する前に、この属性を設定しなければなりません。メッセージがメッセージ・テキスト中に置換変数を持っている時に、メッセージ置換テキストが参照されます。メッセージ置換変数は、`%` 記号の後に数値を続けることによって識別されます。各メッセージ置換変数は **MsgSubText** 属性の対応する空白区切り語によって置き換えられ、その後でそのメッセージがメッセージ・サブファイル・パーツに追加されます。別の **MsgSubText** 属性が設定されるまで、このメッセージ置換ストリングが有効となります。

タイプ スtring

適用可能なパーツ

|              |  |  |
|--------------|--|--|
| メッセージ・サブファイル |  |  |
|--------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

メッセージ置換テキスト

フリー・フォームの例

メッセージ 1000 が次で構成されている場合には、  
続行するためには %1 を押し、取り消すためには %2 を押してください。

メッセージは次の通り表示されます。

続行するためには YES を押し、取り消すためには NO を押してください。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'MSG1':'MsgSubText')='Yes No'
C          EVAL      %setatr('win01':'MSG1':'AddMsgID')=1000
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'MSG1'      SETATR  'Yes No'      'MSGSUBTEXT'
C 'MSG1'      SETATR  1000          'ADDMSGID'
*
```

## MsgText

コンポーネント・メッセージ・ファイルから指定されたメッセージ ID のメッセージ・テキストを戻します。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

次の通り、メッセージ 0001 のメッセージ・テキストを表示します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('*Component': '*Component': 'MsgID')=0001
C          EVAL      text=%getatr('*Component': '*Component': 'MsgText')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          '*COMPONENT' SETATR  0001      'MSGID'
C          '*COMPONENT' GETATR  'MSGTEXT' text
*
```

## Multiplier

タイマー・インターバルの値を判別し、この時間が経過した後で、タイマーで **Tick** イベントが起こります。

最大の乗数値は 65535 です。

最小の乗数値は 1 です。

**タイプ** 数値

**適用可能なパーツ**

|      |  |  |
|------|--|--|
| タイマー |  |  |
|------|--|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | いいえ      |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

乗数値

**フリー・フォームの例**

**Interval** 属性が 1 000 ミリ秒に設定された場合には、タイマーは 3 秒ごとに **Tick** イベントを生成します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'TIMER01': 'Multiplier') = 3
*
```

**固定形式の例**

タイマーの **Multiplier** 属性を変更します:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'TIMER01'   SETATR   3          'MULTIPLIER'
```

## MultSelect

パーツのリスト部分で複数の項目を選択できるかどうかを決定します。この属性が設定されない場合には、項目を選択すると、現在選択されている項目が選択解除されます。

**ExtSelect** 属性によって提供される時に拡張選択を使用可能にするためには、リスト・ボックスでこの属性を設定しなければなりません。

**MultSelect** 属性は **ExtSelect** 属性上では無効です。 **ExtSelect** 属性は、**MultSelect** または単一選択のいずれかの設定を指定変更します。

タイプ 数値

適用可能なパーツ

|          |        |  |
|----------|--------|--|
| リスト・ボックス | サブファイル |  |
|----------|--------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

使用可能な値

複数選択状況は次の値の 1 つとすることができます。

**0** 項目を 1 つだけ選択することができます。

**1** 複数の項目を選択することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C                               EVAL      rc = %getatr('win01': 'LB1': 'MultSelect')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C      'LB1'          GETATR      'MULTSELECT' rc              1 0
*
```

## Name

コンポーネントの名前を戻します。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

次の通り、コンポーネントの名前を表示します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      appname=%getatr('*Component':'*Component':'Name')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C '*COMPONENT' GETATR 'NAME'      appname
*
```

## NbrOfImage

\*制約事項: Java 専用の属性。

GIF アニメーション順序で再生するイメージの数を設定します。この属性は、**FileName** および **Mode** 属性と一緒に使用します。

たとえば、基本ファイル *File.GIF* を 10 個のイメージの連続として再生するには、**FileName** を *File.GIF* に、**NbrOfImage** を 10 に、**Mode** を 1 に設定します。また、*File1.GIF*、*File2.GIF* ... *File10.gif* などのいくつかの異なる GIF イメージを再生することもできます。

タイプ 数値

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| アニメーション制御 |  |  |
|-----------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

再生するイメージの数。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'WIN1':'NbrOfImage')=10
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'WIN1'      SETATR  1          'NBROFIMAGE'
```

## NbrOfLines

複数行編集の行数を戻します。複数行編集が空の場合には、値ゼロが戻されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      NumL = %getatr('win01': 'MLE1': 'NbrOfLines')
*
```

固定形式の例

次の通り、複数行編集の行数を表示します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MLE1'      GETATR  'NBROFLINES' NumL          2 0
*
```



## NbrOfSel

パーツのリスト部分で選択された項目数を戻します。

現在選択されている項目がない場合には、値ゼロが戻されます。

**タイプ** 数値

**適用可能なパーツ**

|          |              |        |
|----------|--------------|--------|
| リスト・ボックス | メッセージ・サブファイル | サブファイル |
|----------|--------------|--------|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

次の通り、リスト・ボックスで選択された項目数を表示します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      num=%getatr('win01':'LB1':'NbrOfSel')
*
C      num          IFEQ      *ZERO
C          EVAL      %setatr('win01':'MSG1':'AddMsgText')='None'
C          ENDIF
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'LB1'        GETATR    'NBROFSEL'    NbrSelected    2 0
*
```

## NextLine

この属性を 1 に設定すると、スクロール・バー・スライダーは、項目のリストで次の行を表す位置に移動します。スライダーは、スクロール・バー・パーツに対して設定された **Range** に対応して移動します。

たとえば、サブファイル・パーツがあり、コマンド・キーまたはカーソル移動キーが押された時に、レコードのセットにおける位置を示したい場合があります。サブファイルに割り当てられたスクロール・バー・パーツは、サブファイル中のレコードの現在数に基づいてユーザーの位置を示すだけです。**VKeyPress** イベントを取り込むことによって、該当するスライダー属性を設定することができます。スライダーが新しい位置に移動すると、**Scroll** イベントが通知されて、**%Position** イベントが更新されます。

**PageSize** 属性は、表示されている項目についての可視ページ・サイズを判別します。スクロールされる項目の数の上限は、**Range** 属性で決定されます。

**タイプ** 数値

適用可能なパーツ

|               |              |  |
|---------------|--------------|--|
| 水平方向のスクロール・バー | 縦方向のスクロール・バー |  |
|---------------|--------------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'VSC1':'NextLine')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'VSC1'      SETATR      1          'NEXTLINE'
*
```

## NextPage

この属性を 1 に設定すると、スクロール・バー・スライダーは、項目のリストで次のページを表す位置に移動します。スライダーは、スクロール・バー・パーツに対して設定された **Range** に対応して移動します。

たとえば、サブファイル・パーツがあり、コマンド・キーまたはカーソル移動キーが押された時に、レコードのセットにおける位置を示したい場合があります。サブファイルに割り当てられたスクロール・バー・パーツは、サブファイル中のレコードの現在数に基づいてユーザーの位置を示すだけです。**VKeyPress** イベントを取り込むことによって、該当するスライダー属性を設定することができます。スライダーが新しい位置に移動すると、**Scroll** イベントが通知されて、**%Position** イベントが更新されます。

**PageSize** 属性は、表示されている項目についての可視ページ・サイズを判別します。スクロールされる項目の数の上限は、**Range** 属性で決定されます。

**タイプ** 数値

適用可能なパーツ

|               |              |  |
|---------------|--------------|--|
| 水平方向のスクロール・バー | 縦方向のスクロール・バー |  |
|---------------|--------------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'VSC1':'NextPage')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'VSC1'      SETATR      1          'NEXTPAGE'
```

## NotSrcEvt

コンポーネント参照パーツに、別のコンポーネントの特定ウィンドウの特定パーツでの特定イベントのオカレンスをモニターさせるためには、この属性を **NotSrcPart** および **NotSrcWin** の属性と組み合わせて使用してください。この属性は所要のイベントを識別するために使用されます。そのイベントのオカレンスについてモニターしたい特定パーツおよびウィンドウを指定するためには、**NotSrcPart** および **NotSrcWin** を使用してください。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      event = %getatr('win01': 'CR1': 'NotSrcEvt')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CR1'      GETATR  'NOTSRCEVT'  event
*
```

## NotSrcPart

コンポーネント参照パーツに、別のコンポーネントの特定ウィンドウの特定パーツでの特定イベントのオカレンスをモニターさせるためには、この属性を **NotSrcEvt** および **NotSrcWin** の属性と組み合わせて使用してください。この属性は所要のパーツを識別するために使用されます。そのパーツおよびそのパーツが入っている特定のウィンドウでモニターしたい特定のイベントを指定するためには、**NotSrcEvt** および **NotSrcWin** を使用してください。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      PartN = %getatr('win1': 'CR1': 'NotSrcPart')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'CR1'      GETATR  'NOTSRCPART' PartN
*
```

## NotSrcWin

コンポーネント参照パーツに、別のコンポーネントの特定ウィンドウの特定パーツでの特定イベントのオカレンスをモニターさせるためには、この属性を **NotSrcEvt** および **NotSrcPart** の属性と組み合わせて使用してください。この属性は所要のウィンドウを識別するために使用されます。そのウィンドウでモニターしたい特定パーツおよびイベントを指定するためには、**NotSrcPart** および **NotSrcEvt** を使用してください。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      WinNam = %getatr('win1': 'CR1': 'NotSrcWin')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CR1'      GETATR  'NotSrcWin' WinNam
*
```

## OCXProp

プロパティの名前を決定します。この名前は、**OCXValue** 属性のプロパティ値を決定するために使用されます。

タイプ スtring

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'AX1':'OCXProp')='PN1'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'AX1'      SETATR      'PN1'      'OCXPROP'
*
```

## OCXPropIdx

ActiveX 配列プロパティの索引を設定または検索します。この属性を **OCXProp** (プロパティ名) および **OCXValue** (プロパティ値) 属性と一緒に使用して、配列プロパティを取り扱うことができます。

多次元配列プロパティのエレメントを設定するには、各エレメントの索引値を 'n1 n2 n3' (n1 は次元 1 の索引など) のようなストリングに入れて渡します。

**タイプ** ストリング

**適用可能なパーツ**

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'AX1':'OCXPropIdx')='0'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AX1'      SETATR  '0'          'OCXPROPIDX'
```



## Parameter

ActiveX イベントの 1 つのパラメーターを検索します。最初に、**Parameter** 属性を設定する必要があります。

タイプ スtring

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      param=%getatr('win01':'AX1':'Parameter')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AX1'      GETATR  'Paramameter' param
*
```

## ParamIndex

ActiveX イベント・パラメーターの索引を設定します。この属性を **Parameter** 属性とともに使用して、イベント・パラメーターを取り扱うことができます。

タイプ 数値

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

1 -- N (N は、イベント内のパラメーター数)。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'AX1':'ParamIndex')=2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AX1'      SETATR  2          'ParamIndex'
*
```

## OCXValue

**OCXProp** 属性によって提供されたプロパティの値を決定します。

タイプ スtring

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'AX1':'OCXValue')='VL1'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AX1'      SETATR  'VL1'      'OCXVALUE'
*
```

## OnTop

指定したノートブック・ページを現行ページとなるように設定するか、あるいはノートブック・ページが現行ページであるかどうかを判断します。

タイプ 数値

適用可能なパーツ

|            |                   |  |
|------------|-------------------|--|
| ノートブック・ページ | キャンバス付きノートブック・ページ |  |
|------------|-------------------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

- 1 指定したノートブック・ページを現行ページとなるように設定します。

戻される値

- 1 このノートブック・ページは現行ページです。
- 0 ノートブック・ページは現行ページではありません。

フリー・フォームの例

ノートブック・ページ **NP4** を現行ページとなるように設定します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'NP4': 'OnTop') = 1
C          EVAL      Curr = %getatr('win01': 'NP4': 'OnTop')
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'NP4'      SETATR   1      'ONTOP'
C 'NP4'      GETATR   'ONTOP'  CURR
```

## OpenEdit

サブファイル・レコード中の入力フィールドを編集可能にするかどうかを決定します。この属性を設定する前に、**ColNumber** および **Index** の属性を設定しなければなりません。

**OpenEdit** 属性によって戻される値は、サブファイルのフィールドが編集のために現在オープンされているかどうかを指示します。したがって、この属性の表示の前に、**ColNumber** および **Index** の属性を設定する必要はありません。

**OpenEdit** 属性がゼロに設定された場合には、編集のために現在オープンされているフィールドはクローズされます。**ColNumber** および **Index** 属性は無視されません。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**OpenEdit** は、次のいずれかとすることができます。

- 0 フィールドは編集可能ではありません。
- 1 フィールドは編集可能です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      isopen = %getatr('win01': 'SF1': 'OpenEdit')
*
C          EVAL      %setatr('win01': 'SF1': 'OpenEdit') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'SF1'      GETATR  'OPENEDIT'  isopen
*
```

## FreezeCol

**FreezeCol** 属性を N (N>0) に設定すると、サブファイルをスクロールするための横方向スクロール・バーを使用しても、field1 から fieldN はスクロールされません。通常の操作に戻すには、この属性の値を 0 に設定してください。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

0 以上で、フィールド数より少ない整数。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+Hi LoEq----
*
C          EVAL      frzCol = %getatr('win01': 'SF1': 'FreezeCol')
*
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+Hi LoEq----
*
C      'SF1'      SETATR      2          'FreezeCol'
*
```

## OS

オペレーティング・システムの名前を返します。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

戻される値

WINDOWS: Windows オペレーティング・システム

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      ostring=%getatr('*Component':'*Component':'Os')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      '*COMPONENT' GETATR  'OS'          ostring
*
```

## OutlineRcl

**YearIdx/MonthIdx/DayIdx** 日付を囲む長方形があるかどうかを示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**OutlineRcl** は、次のいずれかとすることができます。

- 0** 長方形がありません
- 1** 長方形があります

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'OutlineRcl') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'CAL1'      SETATR  1          'OUTLINERCL'
```



## PageNumber

現在どのページ番号が選択されているかを判別します。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| ノートブック |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

現在選択されているページ番号

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'NB1': 'PageNumber') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'NB1'      SETATR      1          'PAGENUMBER'
*
```

## PageSize

サブファイル・パーツの場合には、可視であるデータの行数を示します。これは、一度に 1 ページずつ処理したい場合に役立ちます。

横または縦方向のスクロール・バー・パーツの場合には、可視ページのサイズを決定します。**PageSize** 値は、**Range** 値を超えることができません。

タイプ 数値

適用可能なパーツ

|               |              |         |
|---------------|--------------|---------|
| 水平方向のスクロール・バー | 縦方向のスクロール・バー | サブファイル* |
|---------------|--------------|---------|

\* **制約事項:** サブファイル・パーツの場合には、この属性は実行時にしか取得できません。

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

横または縦方向のスクロール・バー・パーツの場合には、**PageSize** 値は 1 から 32,767 までの値とすることができます。これは、**Range** 値を超えることができません。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      PS = %getatr('win01': 'SFL1': 'PageSize')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'SFL1'      GETATR  'PAGESIZE'  PS
*
```

## Panel

**注:** この属性は Windows にのみ適用されます。

イメージ・パーツの場合には、**panel** によって、イメージ・パーツが接続された拡大パネルを持つかが決まります。拡大パネルにはイメージの操作を可能にする次の 2 つのコントロールが含まれます。

### スケールをチェック・ボックスに合わせる

このチェック・ボックスがチェックされていない時は、イメージのスケールがイメージ・パーツに合わせてられます。このチェック・ボックスがチェックされている時は、拡大スライダーが使用可能になります。

### 拡大スライダー

使用可能な時は、このスライダーが、イメージの拡大をそのオリジナル・サイズの 25% - 200% の間で調整します。

メディア・パーツの場合には、**panel** は、ファイル再生用のメディア・パネルを表示するために使用します。

**タイプ** 数値

### 適用可能なパーツ

|      |      |  |
|------|------|--|
| イメージ | メディア |  |
|------|------|--|

### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

## 使用可能な値

次の 1 つです。

**0** 拡大パネルが接続されないか、あるいは再生パネルが隠されます。

**1** 拡大パネルが接続されるか、あるいは再生パネルが表示されます。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      oldval = %getatr('window': 'MMP1': 'Panel')
*
C          EVAL      %setatr('window': 'MMP1': 'Panel') = 1
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'MMP1'   GETATR   'PANEL'   OldVal      1 0
C 'MMP1'   SETATR   1         'PANEL'
*
```

## PanelItem

メディア・パネル上で最後に変更されたボタンを示す値を戻します。ボタンの変更は、ユーザーがマウスでそれをクリックするか、あるいは **PanelMode** 属性を設定することによって変更することができます。ユーザーがメディア・パネルのボタンのいずれか 1 つをクリックした後で、この属性をチェックすることができます。これによって、**Press** イベントが起こることになります。その後で、ユーザーの `press` アクション・サブルーチンで **PanelItem** 属性を使用して、実行する機能を判別することができます。

**タイプ** 数値

**適用可能なパーツ**

|          |  |  |
|----------|--|--|
| メディア・パネル |  |  |
|----------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**戻される値**

押される最後のメディア・パネル・ボタンの状況:

- 0** ボタンは押されていません
- 1** 休止
- 2** 再生
- 3** レコード
- 4** 停止
- 5** 次のトラック
- 6** 前のトラック
- 7** 次の索引
- 8** 前の索引
- 9** ボリューム
- 10** 位置

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      button = %getatr('win01': 'MMP1': 'PanelItem')
*
C  button      CASEQ  2          PLAY
C  button      CASEQ  3          RECORD
C          ...
C          ENDCS
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'MMP1'      GETATR  'PANELITEM'  button      2 0
*
C  button      CASEQ  2          PLAY
C  button      CASEQ  3          RECORD
C          ...
C          ENDCS
*
```

## PanelMode

メディア・パネル・パーツの操作モードを設定します。

タイプ 数値

適用可能なパーツ

|          |  |  |
|----------|--|--|
| メディア・パネル |  |  |
|----------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

メディア・パネル・パーツの操作モードは次とすることができます。

- 1 休止
- 2 再生
- 3 レコード
- 4 停止

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MMP1': 'PanelMode') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MMP1'      SETATR  1          'PANELMODE'
```

## Parent

**PartList** 属性で子のリストが検索される親パーツを設定します。直接の子のリストが戻されます。コンポーネントのフレーム・ウィンドウのリストを表示するには、**Parent** 属性を予約名 **\*component** に設定します。

注: この属性を設定しなければ、**PartCount** または **PartList** を使用できません。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

親パーツ名の形式は次の通りです。

PARENTNAME|PARTNAME

**ParentName** は、**PartName** を所有するウィンドウ・パーツの名前です。ウィンドウ内のパーツのリストを取り出したい場合には、キャンバス名をパーツ名として使用してください。



## フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('*Component': '*Component': 'Parent')
C          = 'WIN1|CANVS1'
*
```

## 固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      '*component' SETATR      'WIN1|CANVS1' 'PARENT'
*
```

親ウィンドウ・パーツからキャンバス名を取得したい場合には、ウィンドウ名を 2 回指定する必要があります。次のコードの例は、この実行方法を示すものです。(WinName 変数には、親ウィンドウのウィンドウ名が入っています。)

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
* Window を親として設定します。ウィンドウ名を 2 回入力してください。
C          EVAL      %setatr('*Component': '*Component': 'Parent')
C          = WinName + '|' + WinName
*
* キャンバス名を持つ PartList を取得して、それを Canvas 変数に割り当てます。
C          EVAL      Canvas=%getatr('*component': '*component': 'PartList')
```

この属性が戻す値については、**PartList** の説明を参照してください。

## ParentId

コンテナ中にあるレコードの親レコードのレコード ID を戻します。この属性を使用する前に、**RecordID** 属性は、親の ID を決定したいレコードの ID に設定します。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

次の例では、レコードが選択されていることを前提としています。選択されたレコードの親レコードのレコード ID が決定されます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
CSRNO1Factor1+++++Opcode(E)+Extended-factor2+++++-----
* Set the ID of the selected record
C          EVAL      %setatr('Main':'CT1':'RecordID')=
C          %getatr('Main':'CT1':'FirstSel')
* Get the ID of the parent record of the selected record
C          EVAL      Parent=%getatr('Main':'CT1':'ParentId')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CT1'      GETATR  'FIRSTSEL'  ID          4 0
C 'CT1'      SETATR  ID          'RECORDID'
C 'CT1'      GETATR  'PARENTID'  PARENT      4 0
*
```

## ParentList

現行レコードの親 id のブランク区切りのリストを戻します。この属性を使用する前に、**RecordID** 属性を現行レコードの id に設定してください。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

次の例では、レコードが選択されていることを前提としています。親レコードのレコード ID が決定されます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
CSRNO1Factor1+++++Opcod(E)+Extended-factor2+++++-----
* Set the ID of the selected record
C          EVAL      %setatr('Main':'CT1':'RecordID')=
C          %getatr('Main':'CT1':'FirstSel')
* Get the list of parent IDs for the selected record
C          EVAL      ParentL=%getatr('Main':'CT1':'ParentList')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C   'CT1'      GETATR   'FIRSTSEL'   ID           4 0
C   'CT1'      SETATR   ID           'RECORDID'
C   'CT1'      GETATR   'PARENTLIST' PARENTL     4 0
*
```

## ParentName

パーツ・パラメーターによって参照されるパーツが入っているパーツの名前を戻します。ウィンドウ上にあるパーツの場合には、ウィンドウの名前が戻されます。ウィンドウ・パーツにこの属性を使用する時には、ウィンドウ・パーツ名が戻されません。

**タイプ** スtring

**適用可能なパーツ**

\*Component パーツを除くすべてのパーツ

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      PName = %getatr('win01': 'EF1': 'ParentName')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'EF1'      GETATR  'PARENTNAME' PName          1 0
*
```

## PartCount

属性 **Parent** によって設定された親パーツの子の数を戻します。

タイプ 数値

適用可能なパーツ

|            |           |  |
|------------|-----------|--|
| *Component | Java Bean |  |
|------------|-----------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

使用可能な値

指定された親パーツの子の数。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Numchild=%getatr('*Component':'*Component':
C          'PartCount')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          '*Component' GETATR      'PARTCOUNT'  Numchild
*
```

## PartList

**Parent** 属性によって設定された親パーツ・セットの子パーツのリストを戻します。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**戻される値**

Stringの形式は、いくつかのパーツ名からなる連続Stringです。各パーツの名前は、10文字の固定String長を占めます。パーツ名部分が10文字より小さい場合には、名前の右側にBlankが埋め込まれます。

FRAMEbbbbCANVASbbbbENTRYFIELD

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      list = %getatr('*Component': '*Component':
C          'PartList')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C '*Component' GETATR 'PARTLIST' list
*
```

## PartName

パーツの名前を戻します。これは、ノートブック設定を使用してパーツを最初に作成した時に、そのパーツに割り当てられた名前です。

名前の値の長さは 10 文字です。この値は、結果フィールドに左寄せして戻されます。結果フィールドが 10 桁より小さい場合には、名前の値が切り捨てられます。結果フィールドが 10 文字より大きい場合には、フィールドにはブランクが埋め込まれます。

**タイプ** スtring

**適用可能なパーツ**

\*Component パーツを除くすべてのパーツ

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      ctlname = %getatr('win01': 'MLE1': 'PartName')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'MLE1'      GETATR  'PARTNAME'  ctlname      10
*
```

## PartType

パーツのタイプを戻します。VARPG パーツで戻される値は、FVDES の文字の接頭部が付けられます。たとえば、入力フィールド・パーツで戻される値は *FVDESEntryField* です。

**注:** コンポーネント参照パーツでは、戻される値は FVDESV の文字の接頭部が付けられます。

**タイプ** スtring

**適用可能なパーツ**

\*Component パーツを除くすべてのパーツ

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      Type = %getatr('win01': 'MLE1': 'PartType')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'MLE1'      GETATR  'PARTTYPE'   Type          50
*
```



## Paste

クリップボードからのテキストをカーソル位置にペーストします。テキストが選択された場合には、クリップボードからのテキストが選択されたテキストに取って代わります。

タイプ 数値

適用可能なパーツ

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

使用可能な値は 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'Paste') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MLE1'      SETATR  1          'paste'
*
```

## PBRange

進行状況バーの上界を決定します。デフォルト値は 100 です。この値は、プログラムがモニターしている処理の上限を表すことができます。たとえば、プログラムが 1,000 個のファイルをアップロードしている場合には、この処理の上限を 1,000 に設定したいことがあります。

タイプ 数値

適用可能なパーツ

|        |       |  |
|--------|-------|--|
| 進行状況バー | ウィンドウ |  |
|--------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
* Set the range for a progress bar attached to a window
C          EVAL      %setatr('win01':'WIN1':'PBRange')=1000
* Set the range for a progress bar part
C          EVAL      %setatr('win01':'PBar1':'PBRange')=1000
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
* Set the range for a progress bar attached to a window
C  'WIN1'      SETATR   1000      'PBRANGE'
* Set the range for a progress bar part
C  'PBAR1'     SETATR   1000      'PBRANGE'
```

## PBSetPos

進行状況バー標識を指定された絶対位置に移動します。この値は、**PBRange** 属性がどのように設定されたかによって異なります。

タイプ 数値

適用可能なパーツ

|        |       |  |
|--------|-------|--|
| 進行状況バー | ウィンドウ |  |
|--------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'WIN1':'PBSetPos')=50
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'WIN1'      SETATR  50          'PBSETPOS'
```

## PBStep

進行状況バー標識を **PBStepSize** 値で定義されたステップ単位の数だけ移動するには、この属性を 1 に設定します。

タイプ 数値

適用可能なパーツ

|        |       |  |
|--------|-------|--|
| 進行状況バー | ウィンドウ |  |
|--------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'WIN1':'PBStep')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'WIN1'      SETATR      1      'PBSTEP'
```

## PBStepSize

**PBStep** 属性がオンに設定された時に進行状況バー標識がどれだけ移動するかを決定します。デフォルト値は 10 です。

タイプ 数値

適用可能なパーツ

|        |       |  |
|--------|-------|--|
| 進行状況バー | ウィンドウ |  |
|--------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'WIN1':'PBStepSize')=10
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'WIN1'      SETATR  10          'PBSTEPSIZE'
```

## Platform

アプリケーションが稼動しているプラットフォーム名を戻します。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**戻される値**

次の 1 つです。

**WINDOWS**

ウィンドウ

**JAVA** Java

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      app1=%getatr('*Component':'*Component':'Platform')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      '*COMPONENT' GETATR      'PLATFORM'      APP1
*
```

# PlugCmd

## \*制約事項:

1. この属性は Java アプリケーションではサポートされていません。
2. この属性は VisualAge RPG で作成されたプラグインに適用されるだけです。

GUI Designer のプラグイン・コンポーネントにコマンドを渡します。**PlugCmd** を使用する前に、**PlugDLL** および **PlugID** 属性を設定する必要があります。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

## 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

## 使用可能な値

プラグイン・コマンド

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C          EVAL      %setatr('*Component': '*Component': 'PlugCmd')
C          = cmdstring
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C      '*Component' SETATR      cmdstring      'PLUGCMD'
*
```

# PlugDLL

## \*制約事項:

1. この属性は Java アプリケーションではサポートされていません。
2. この属性は VisualAge RPG で作成されたプラグインに適用されるだけです。

プラグイン DLL ユーティリティー名を設定します。**PlugCmd** を使用する前に、**PlugDLL** および **PlugID** 属性を設定する必要があります。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

## 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

## 使用可能な値

プラグイン DLL ユーティリティー名

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('*Component': '*Component': 'PlugDLL')
C          = PlugDLL
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          '*Component' SETATR    DLLname          'PLUGDLL'
*
```



# PlugID

## \*制約事項:

1. この属性は Java アプリケーションではサポートされていません。
2. この属性は VisualAge RPG で作成されたプラグインに適用されるだけです。

プラグインに渡される GUI Designer の ID を設定します。**PlugCmd** を使用する前に、**PlugDLL** および **PlugID** 属性を設定する必要があります。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

GUI Designer の ID

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('*Component': '*Component': 'PlugID')
C          = idname
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C      '*Component' SETATR    idname      'PLUGID'
*
```

# PlugRC

## \*制約事項:

1. この属性は Java アプリケーションではサポートされていません。
2. この属性は VisualAge RPG で作成されたプラグインに適用されるだけです。

最後のプラグイン・コマンドからの戻りコードを取得します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

## 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      code = %getatr('*Component': '*Component':
C          'PlugRC')
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          '*Component' GETATR  'PLUGRC'      code          2 0
*
```

# PlugResult

## \*制約事項:

1. この属性は Java アプリケーションではサポートされていません。
2. この属性は VisualAge RPG で作成されたプラグインに適用されるだけです。

最後のプラグイン・コマンドからの結果ストリングを取得します。

タイプ ストリング

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

## 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C                               EVAL      Result = %getatr('*Component': '*Component':
C                               'PlugResult')
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      '*Component' GETATR      'PLUGRESULT' Result
*
```

## Poke

データをサーバー・アプリケーションに送信します。データをサーバー・アプリケーションに Poke するためには、DDE 会話がこの DDE パーツのために正常に開始されていなければなりません。また、**DDEItem** 属性を設定して、どのサーバー項目が探し出されたデータを受信するかを指示しなければなりません。

サーバー・アプリケーションがポーク要求を処理すると、このアプリケーションは **PokeAck** イベントで応答します。ユーザーの **PokeAck** アクション・サブルーチンの **%PokeAck** 属性をチェックして、そのポークが正常に完了したかどうかを判別しなければなりません。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL    %setatr('win01': 'DDE1': 'DDEItem') = 'A1'
C          EVAL    %setatr('win01': 'DDE1': 'Poke') = '456.78'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          'DDE1'    SETATR    '123'    'POKE'
*
```

## Position

この属性に戻される値は、パーツのタイプによって異なります。

メディア・パーツの場合には、この属性はオーディオ・ファイルの現在の位置を戻します。この値はミリ秒単位です。オーディオ・ファイルをまだ処理していない場合には、値 0 が戻されます。

メディア・パネル・パーツでは、この属性は位置スライダーの位置を戻します。この値は 0-100 の範囲になります。

横または縦方向のスクロール・バー・パーツの場合には、この属性は、**Range** 属性値に基づいてスクロール・ボックスの位置を戻します。

### タイプ 数値

### 適用可能なパーツ

|               |      |          |
|---------------|------|----------|
| 水平方向のスクロール・バー | メディア | メディア・パネル |
| 縦方向のスクロール・バー  |      |          |

### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

### 使用可能な値

メディア・パネル・パーツの場合には、**Position** 値 (0 から 100 まで) はパーセントを表します。横または縦方向スクロール・バー・パーツの場合には、これは **Range** 値に基づく絶対値です。

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C                               EVAL      oldpos = %getatr('win01': 'MMP1': 'Position')
*
C                               EVAL      %setatr('win01': 'MMP1': 'Position') = 5
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'MMP1'      GETATR   'POSITION'   OldPos      3 0
*
C   'MMP1'      SETATR   5           'POSITION'
```

## PrevLine

この属性を 1 に設定すると、スクロール・バー・スライダーは、項目のリストで前の行を表す位置に移動します。スライダーは、スクロール・バー・パーツに対して設定された **Range** に対応して移動します。

たとえば、サブファイル・パーツがあり、コマンド・キーまたはカーソル移動キーが押された時に、レコードのセットにおける位置を示したい場合があります。サブファイルに割り当てられたスクロール・バー・パーツは、サブファイル中のレコードの現在数に基づいてユーザーの位置を示すだけです。**VKeyPress** イベントを取り込むことによって、該当するスライダー属性を設定することができます。スライダーが新しい位置に移動すると、**Scroll** イベントが通知されて、**%Position** イベントが更新されます。

**PageSize** 属性は、表示されている項目についての可視ページ・サイズを判別します。スクロールされる項目の数の上限は、**Range** 属性で決定されます。

**タイプ** 数値

適用可能なパーツ

|               |              |  |
|---------------|--------------|--|
| 水平方向のスクロール・バー | 縦方向のスクロール・バー |  |
|---------------|--------------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'VSC1':'PrevLine')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'VSC1'      SETATR      1          'PREVLINE'
*
```

## PrevPage

この属性を 1 に設定すると、スクロール・バー・スライダーは、項目のリストで前のページを表す位置に移動します。スライダーは、スクロール・バー・パーツに対して設定された **Range** に対応して移動します。

たとえば、サブファイル・パーツがあり、コマンド・キーまたはカーソル移動キーが押された時に、レコードのセットにおける位置を示したい場合があります。サブファイルに割り当てられたスクロール・バー・パーツは、サブファイル中のレコードの現在数に基づいてユーザーの位置を示すだけです。**VKeyPress** イベントを取り込むことによって、該当するスライダー属性を設定することができます。スライダーが新しい位置に移動すると、**Scroll** イベントが通知されて、**%Position** イベントが更新されます。

**PageSize** 属性は、表示されている項目についての可視ページ・サイズを判別します。スクロールされる項目の数の上限は、**Range** 属性で決定されます。

**タイプ** 数値

適用可能なパーツ

|               |              |  |
|---------------|--------------|--|
| 水平方向のスクロール・バー | 縦方向のスクロール・バー |  |
|---------------|--------------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'VSC1':'PrevPage')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'VSC1'      SETATR      1          'PREVPAGE'
```

## Print

ウィンドウのクライアント域またはイメージを印刷できるようにします。ストリング値は、印刷ジョブの名前およびタイトルを表します。この属性を設定すると、Windows の「プリンター」ダイアログが表示されるので、プリンターを選択することができます。

**タイプ** ストリング

**適用可能なパーツ**

|      |       |              |
|------|-------|--------------|
| イメージ | ウィンドウ | キャンバス付きウィンドウ |
|------|-------|--------------|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'ima2':'Print')='Image2'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'WIN01'      SETATR  'IMAGE2'      'PRINT'
*
```



## PrintAsIs

イメージを印刷し、縦横比を維持します。string値は、印刷ジョブの名前およびタイトルを表します。この属性を設定すると、Windows の「プリンター」ダイアログが表示されるので、プリンターを選択することができます。

タイプ string

適用可能なパーツ

|      |       |              |
|------|-------|--------------|
| イメージ | ウィンドウ | キャンバス付きウィンドウ |
|------|-------|--------------|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'ima2':'PrintAsIs')='Image2'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'WIN01'    SETATR  'IMAGE2'    'PRINTASIS'
```

## Printer

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

「プリンター」ダイアログで選択されたプリンター名を戻します。Windows のシステム・プリンター・ダイアログは、**SelPrinter** 属性を設定すると表示されます。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      prtname = %getatr('*Component': '*Component':
C          'Printer')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C '*Component' GETATR 'PRINTER' prtname
*
```

## ProgresBar

単純進行状況バーを拡張機能としてウィンドウに追加したり、ウィンドウから除去します。状況ウィンドウが存在する場合には、進行状況バーは、ウィンドウの反対側の位置に移動されます。より多くの種類の進行状況バーが必要な場合には、進行状況バー・パーツを使用してください。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**ProgresBar** は、次のいずれかとすることができます。

- 1 進行状況バーを作成します
- 0 進行状況バーを除去します

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'WIN1':'ProgresBar')=1
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C 'WIN1'      SETATR  1          'PROGRESBAR'
*
```

## Range

スクロールする項目の上限界を決定します。この値は、リストに表示される項目の数を表すことができます。たとえば、プログラムが 100 個のファイルのリストを表示している場合には、この目的のために上界を 100 に設定したいことがあります。

**PageSize** 属性は、表示されている項目についての可視ページ・サイズを判別します。

**タイプ** 数値

**適用可能なパーツ**

|               |              |  |
|---------------|--------------|--|
| 水平方向のスクロール・バー | 縦方向のスクロール・バー |  |
|---------------|--------------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**Range** は 1 から 32,767 までとすることができます。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'HScroll1':'Range')=100
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'HSCROLL1' SETATR 100          'RANGE'
*
```

## ReadOnly

パーツの入力フィールド部分を読み取り専用とするかどうかを決定します。読み取り専用を設定した時には、入力フィールドに入力することはできません。

この属性をキャンバス・パーツに適用すると、そのキャンバス上のすべての入力フィールドは、**ReadOnly** 属性の設定を継承することになります。

タイプ 数値

適用可能なパーツ

|         |           |         |
|---------|-----------|---------|
| キャンバス   | 組み合わせボックス | コンテナー   |
| 入力フィールド | 複数行編集     | スピン・ボタン |

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: コンテナー・パーツを除き、GUI Designer で プロパティ・ノートブック を使用して設定することができます。組み合わせボックス・パーツでは、組み合わせボックス・スタイルのドロップダウン・リスト・タイプを選択することによって、これを行ってください。その他のパーツでは、**読み取り専用**スタイルを選択することによって、これを行ってください。

使用可能な値

フィールドの **ReadOnly** 状況は次の通りとすることができます。

- 0 入力フィールドは読み取り専用ではありません。
- 1 入力フィールドは読み取り専用です。

## フリー・フォームの例

入力フィールドの読み取り専用の切り替え:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      rc = %getatr('win01': 'EF1': 'ReadOnly')
*
C   rc          IFEQ      1
C          EVAL      %setatr('win01': 'EF1': 'ReadOnly') = 0
C          ELSE
C          EVAL      %setatr('win01': 'EF1': 'ReadOnly') = 1
C          ENDIF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'EF1'      GETATR   'READONLY'   rc          1 0
*
C   rc          IFEQ      1
C   'EF1'      SETATR   0          'READONLY'
C          ELSE
C   'EF1'      SETATR   1          'READONLY'
C          ENDIF
*
```

## RecordID

他の属性によって参照されるコンテナ・レコードの識別を設定します。たとえば、**GetRcdFld** 属性を使用する前に、最初に **RecordID** 属性を設定しなければなりません。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

コンテナ・レコードの ID

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CN1': 'RecordID') = id
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CN1'          SETATR  id          'RECORDID'
*
```

## RefAttr

コンポーネント参照パーツ内の参照中のパーツの属性。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'COMP1': 'RefAttr') = 'Text'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'COMP1'      SETATR  'Text'      'REFATTR'
```



## RefParent

ターゲット・パーツの親ウィンドウの名前。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'COMP1': 'RefParent')
C          = 'WIN01'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'COMP1'   SETATR  'WIN01'   'REFPARENT'
```

## RefPart

参照中のターゲット・コンポーネント内のパーツの名前。

タイプ スtring

適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('window': 'COMP1': 'RefPart') = 'EF01'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'COMP1'      SETATR  'EF01'      'REFPART'
*
```

## Refresh

パーツを再ペイントまたは再ドロースさせます。

タイプ 数値

適用可能なパーツ

|                    |           |                |
|--------------------|-----------|----------------|
| ActiveX            | カレンダー     | キャンバス          |
| チェック・ボックス          | 組み合わせボックス | コンテナ           |
| 入力フィールド            | グラフ       | グラフィック・プッシュボタン |
| グループ・ボックス          | イメージ      | リスト・ボックス       |
| 複数行編集              | ノートブック    | ノートブック・ページ     |
| ODBC/JDBC インターフェース | 外枠        | プッシュボタン        |
| ラジオ・ボタン            | スライダー     | スピン・ボタン        |
| 静的テキスト             | ウィンドウ     |                |

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**Refresh** 属性は 1 にしか設定できません。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'LB1': 'Refresh') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'LB1'      SETATR  1          'REFRESH'
*
```

## RemoveItem

項目をリストから除去します。*index* パラメーターは、除去する項目を決定します。すべての項目をリストから除去するためには、索引値として 0 を指定します。

サブファイル・パーツの場合には、この属性は、その妥当性検査タイプとして定義された値を持つサブファイル入力フィールドに適用されます。この属性は、サブファイル入力フィールドの妥当性検査リストから値を除去するために使用します。最初に、**ColNumber** 属性を設定して、処理する列を識別します。

スピン・ボタン・パーツの場合には、すべての項目を除去することだけができます。つまり、索引値 0 (すべての除去) を指定することが必要です。スピン・ボタンから特定の項目を除去することはできません。

**タイプ** 数値

**適用可能なパーツ**

|           |          |         |
|-----------|----------|---------|
| 組み合わせボックス | リスト・ボックス | スピン・ボタン |
| サブファイル    |          |         |

**命令コード**

|        |          |
|--------|----------|
|        | 実行時に操作可能 |
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

除去される項目の索引値

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'LB1': 'RemoveItem') = 3
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'          SETATR  3          'REMOVEITEM'
```

## RemoveLink

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

このパーツと *part* パラメーターによって示されたパーツの間のリンクを除去します。パーツが現在リンクされているパーツでない場合には、アクションは取られません。

**タイプ** スtring

**適用可能なパーツ**

|           |         |          |
|-----------|---------|----------|
| チェック・ボックス | 入力フィールド | イメージ     |
| リスト・ボックス  | メディア    | メディア・パネル |
| スライダー     | タイマー    |          |

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

除去されるパーツの名前

これは次の形式でなければなりません。

'window|part'

この場合に、*window* はパーツが入っているウィンドウで、*part* はパーツ名です。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'EF1':'RemoveLink')='WIN1|EF1'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'EF1'      SETATR  'WIN1|EF1'  'REMOVELINK'
*
```

## RemoveMsg

メッセージをメッセージ・サブファイル・パーツから除去します。除去するメッセージは、*index* パラメーターによって決定されます。すべてのメッセージを除去するためには、*index* 値として 0 を指定します。

タイプ 数値

適用可能なパーツ

|              |  |  |
|--------------|--|--|
| メッセージ・サブファイル |  |  |
|--------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

除去されるメッセージの索引番号

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'MSG1': 'RemoveMsg') = 3
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'MSG1'      SETATR      3      'REMOVEMSG'
*
```

## RemoveRcd

コンテナからレコードを除去します。すべてのレコードを除去するためには、ゼロの値を指定してください。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

除去するレコードの ID

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CN1': 'RemoveRcd') = id
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C  'CN1'      SETATR  id          'REMOVERCD'
```

## Request

サーバー・アプリケーション中の項目の現在の値を要求します。要求を行う前に、**AppName** および **Topic** 属性が設定され、DDE 会話が正常に開始されていなければなりません。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'DDE1': 'Request') = 'A1'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'DDE1'      SETATR  'A1'          'REQUEST'
```



## ReturnVal

メソッド呼び出しの値を戻します。

タイプ スtring

適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('win01':'ocxlist':'method')='ReturnPtr'
C          EVAL    ptrstr=%getatr('win01':'ocxlist':'ReturnVal')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          'OCXLIST'  SETATR  'METHOD'    'RETURNPTR'
C          'OCXLIST'  GETATR  'RETURNVAL' 'PTRSTR'
*
```

## RmvSrcEvt

コンポーネント参照パーツによってモニターされるイベントのリストからソース・イベントを除去します。この属性は、**AddSrcEvt** と組み合わせて、1 つのコンポーネント参照パーツで複数のイベントをモニターできるようにします。新しいソース・イベントを追加するには、**AddSrcEvt** 属性を使用してください。

除去するソース・イベントを識別するには、コンポーネント、ウィンドウ、パーツ、およびイベント名を全部一緒に指定してください。このstringの形式は次の通りです。

```
'COMPONENTNAME | WINDOWNAME | PARTNAME | EVENTNAME'
```

コンパイラーは妥当性検査を行わないので、それぞれのソース・イベントは正確に識別する必要があります。

**タイプ** string

**適用可能なパーツ**

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D Event          S          100
*
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSR01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                  EVAL      Event='compb|win02|ef02|change'
C                  EVAL      %setatr('win01': 'CRP1':
C                  'RmvSrcEvt')=Event
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D Event          S          100
*
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSR01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                  EVAL      Event='compb|win02|ef02|change'
C 'CRP1'           SETATR    Event      'RMVSRCEVT'
*
```

## RmvEvent

イベント・フィルター・リストからイベント名を除去します。特殊値 \*ALL を使用して、ActiveX または Java Bean パーツに、リストに保管されているイベントの VARPG プログラムへの経路指定を停止させることができます。

タイプ ストリング

適用可能なパーツ

|         |           |  |
|---------|-----------|--|
| ActiveX | Java Bean |  |
|---------|-----------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'AX1':'RmvEvent')='EVT2'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AX1'      SETATR  'EVT2'      'RMVEVENT'
```

## RowBGClr

サブファイル行の背景カラーを決定する数値。**Index** 属性を使用して、カラーが設定される行を識別する必要があります。

**RowBGMix** 属性を使用して背景カラーを設定した場合には、戻される背景カラーが混合カラーに最も近いカラーです。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

このカラーは次の 1 つでなければなりません。

- 1 デフォルト (Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      RBcolor = %getatr('win01': 'SFL1': 'RowBGCLr')
*
C          EVAL      %setatr('win01': 'SFL1': 'Index') = 1
C          EVAL      %setatr('win01': 'SFL1': 'RowBGCLr') = 5
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'SFL1'      GETATR  'ROWBGCLR'  'RBcolor'      2 0
*
C 'SFL1'      SETATR  1          'INDEX'
C 'SFL1'      SETATR  5          'ROWBGCLR'
```

## RowBGMix

サブファイル行の背景カラーの混合を照会または設定します。**Index** 属性を使用して、カラー混合が照会または設定される行を識別する必要があります。

**タイプ** スtring

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

背景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      RBclrMx = %getatr('win01': 'SFL1': 'RowBGMix')
*
C  RBclrMx  IFNE     '10:23:200'
C          EVAL      %setatr('win01': 'SFL1': 'RowBGMix') = '10:23:200'
C          ENDIF
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'SFL1'   SETATR   '10:23:200'  'ROWBGMIX'
*
```

## RowFGClr

サブファイル行の前景カラーを決定する数値。**Index** 属性を使用して、カラーが設定される行を識別する必要があります。

**RowFGMix** 属性を使用して前景カラーを設定した場合には、戻される前景カラーが混合カラーに最も近いカラーです。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

このカラーは次の 1 つでなければなりません。

- 1 デフォルト (Java アプリケーションの場合には適用不可)
- 0 白
- 1 黒
- 2 青
- 3 赤
- 4 ピンク
- 5 緑
- 6 青緑
- 7 黄色
- 8 暗灰色
- 9 暗青色
- 10 暗赤色

- 11 暗桃色
- 12 暗緑色
- 13 暗青緑色
- 14 茶色
- 15 淡灰色

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Rfcolor = %getatr('win01': 'SFL1': 'RowFGClr')
*
C          EVAL      %setatr('win01': 'SFL1': 'Index') = 1
C          EVAL      %setatr('win01': 'SFL1': 'RowFGClr') = 5
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'SFL1'      GETATR  'ROWFGCLR'  'Rfcolor'      2 0
*
C 'SFL1'      SETATR  1          'INDEX'
C 'SFL1'      SETATR  5          'ROWFGCLR'
*
```



## RowFGMix

サブファイル行の前景カラーの混合を照会または設定します。**Index** 属性を使用して、カラー混合が照会または設定される行を識別する必要があります。

**タイプ** スtring

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

前景カラーの混合は次の形式でなければなりません。

```
'red:green:blue'
```

ここで、red、green、および blue は 0 - 255 の範囲の値であり、それぞれのカラーの輝度を表します。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      RfClrMx = %getatr('win01': 'SFL1': 'RowFGMix')
*
C  RfClrMx  IFNE     '10:23:200'
C          EVAL      %setatr('win01': 'SFL1': 'RowFGMix') = '10:23:200'
C          ENDIF
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'SFL1'   SETATR   '10:23:200'  'ROWFGMIX'
*
```

## Rows

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

データ・ソース内の行数を戻します。

**注:** この属性は、ODBC ドライバーがバージョン 3 以降である場合にのみ使用します。これより前のドライバーは、予測できない結果を戻します。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL      Nrws=%getatr('win01':'ODBC1':'Rows')
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          'ODBC1'    GETATR  'ROWS'    NRWS
*
```

## SBIndex

状況バー・ペインの索引値を設定します。この値は、どのペインがアクティブ・ペインであるかを決定します。ペインの数と **SBIndex** 値が一致しない場合には、この値にはデフォルトの 1 が使用されます。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| 状況バー |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**SBIndex** 値は 1 から 5 までの値とすることができます。デフォルトは 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'WIN1':'SBIndex')=3
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'WIN1'      SETATR      3          'SBINDEX'
*
```

## SBLabel

状況バーまたは状況バー・ペインに現れるテキスト・ストリング。状況バー・パーツの場合には、**SBIndex** 値は、どのペインがストリングを受け取るかを決定します。テキスト・ストリングは、メッセージ・ファイルからのメッセージ ID とすることもできます。

**タイプ** ストリング

**適用可能なパーツ**

|      |       |  |
|------|-------|--|
| 状況バー | ウィンドウ |  |
|------|-------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'WIN1':'SBLabel')='Wait...'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'WIN1'      SETATR  'Wait...'  'SBLABEL'
*
```

## SBPanes

状況バー・パーツのアクティブ・ペインの数を検索します。**SBIIndex** 属性は、活動化するペインを設定するために使用します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| 状況バー |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Oopcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      NPanes=%getatr('win01':'WIN1':'SBPanes')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Oopcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'WIN1'      GETATR      'SBPANES'      NPANES
*
```

## SBPosition

状況バーの位置を決定します。デフォルトの位置は、ウィンドウの最下部です。進行状況バーが最下部にある場合には、進行状況バーは最上部に移動されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**SBPosition** は、次のいずれかとすることができます。

- 1 状況バーを最上部に置きます
- 2 状況バーを最下部に置きます

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'WIN1':'SBPosition')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'WIN1'      SETATR  1          'SBPOSITION'
```

## SBStyle

状況バーがどのようにドロースされるかを決定します。押し下げ (デフォルト)、手前に持ち上げ、またはフラットが可能です。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**SBStyle** は、次のいずれかとすることができます。

- 1 押し下げ (デフォルト)
- 2 手前に持ち上げ
- 3 フラット

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'WIN1':'SBStyle')=2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'WIN1'      SETATR  2          'SBSTYLE'
```

## Scale

プログラマーが文字 "W" の幅のパーセントによって列の幅を調整できるようにします。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'SF11': 'Scale') = 8
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'SF11'      SETATR      8          'SCALE'
```



## Search

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

リスト内の指定されたストリングを検索します。検索は、**Index** 属性によって指定された項目から開始されます。該当する項目が見つかった場合には **Index** 値が戻され、見つからなかった場合には 0 が戻されます。

**Case** および **SearchType** 属性を使用することによって検索の実行を制御することができます。

**タイプ** ストリング

**適用可能なパーツ**

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CB1': 'Search') = 'Str1'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CB1'          SETATR      'STR1'          'SEARCH'
*
```

## SearchType

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

**Search** 属性の使用時に、検索ストリングによって行われる比較を決定します。

タイプ 数値

適用可能なパーツ

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**SearchType** は次の 1 つとすることができます。

- 1 項目のプレフィックスを比較します。
- 2 項目全体を比較します。
- 3 完全一致 (デフォルト)

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'CB1': 'SearchType') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      'CB1'          SETATR      1          'SEARCHTYPE'
*
```

## Selected

パーツのリスト部分の特定項目を選択するかどうかを示します。チェックする項目は、最初に適用可能なパーツの **Index** または **RecordID** 属性を設定して指定します。

タイプ 数値

適用可能なパーツ

|              |        |          |
|--------------|--------|----------|
| 組み合わせボックス    | コンテナ   | リスト・ボックス |
| メッセージ・サブファイル | サブファイル |          |

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

使用可能な値

**Selected** 状況は次のようにすることができます。

- 0** 項目は選択されません。
- 1** 項目が選択されます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C                               EVAL      Sel = %getatr('win01': 'LB1': 'Selected')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'LB1'          GETATR  'SELECTED'  Sel          1 0
*
```

## SelectIdx

タブおよびグループ・ウィンドウ内の選択されたラジオ・ボタンの位置を指示する指標値を戻します。ラジオ・ボタンは論理グループ内で作成されており、1つのボタンを選択すると、他のすべてのボタンが選択解除されることになります。ラジオ・ボタンが選択されていないと、値 0 か、またはそのグループ内にあるラジオ・ボタンの数より大きい数値が戻されます。

グループ内の任意のラジオ・ボタンの名前をパーツ名として使用することができます。

**タイプ** 数値

**適用可能なパーツ**

|         |  |  |
|---------|--|--|
| ラジオ・ボタン |  |  |
|---------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      sel = %getatr('win01': 'RB1': 'SelectIdx')
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'RB1'          GETATR 'SELECTIDX' sel          4 0
*
```

## SelectItem

パーツのリスト部分の項目を選択します。選択する項目は *index* パラメーターで指定します。リストのすべての項目を選択するためには、**Index** 属性をゼロに設定してください。しかし、リスト・ボックス・パーツの場合には、**Index** 属性がゼロであれば **SelectItem** は影響を与えません。

リストで単一選択だけが可能な場合には、1 つの項目を選択すると現在選択されているすべての項目が選択解除されます。

**注:** 項目を選択すると、**Select** イベントが生成されます。

**タイプ** 数値

**適用可能なパーツ**

|           |          |        |
|-----------|----------|--------|
| 組み合わせボックス | リスト・ボックス | サブファイル |
|-----------|----------|--------|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

項目の索引値。

**フリー・フォームの例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'LB1': 'SelectItem') = 3
*
```

**固定形式の例**

要求があればリスト・ボックスのすべての項目を選択します。

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'CB01'      GETATR  'SELECTED'  IsChecked      1 0
C      IsChecked  IFEQ      1
C      'LB1'      SETATR  0          'SELECTITEM'
C      ENDIF
*
```

## SelectList

リスト・ボックスまたはサブファイル・パーツ中の選択されたレコードの索引のリストを戻します。このリストは、一連の 6 桁の値からなる文字ストリングとして戻されます。各値は、選択されたレコードの索引を表します。

タイプ ストリング

適用可能なパーツ

|          |        |  |
|----------|--------|--|
| リスト・ボックス | サブファイル |  |
|----------|--------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      List=%getatr('Win01':'LB1':'SelectList')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'LB1'      GETATR  'SELECTLIST' LIST
*
```

## SelectList の例

次のコード・セグメントは、**SelectList** 属性を使用してリスト・ボックスで選択した一連のレコードを処理する 1 つの方法を示したものです。この例では、最大 10 個のレコードを選択できることを想定しています。

```
*
D                               DS
D theList                       60A
D T                               6 0 Dim(10) Overlay(theList)
*
C   'ListBox' Getatr 'NbrOfSel' N   2 0
*
C           If N > 0
C   'ListBox' Getatr 'SelectList' theList
*
C           Do N           I           2 0
C   'ListBox' Setatr T(I)       'Index'
C   'ListBox' Getatr 'GetItem'  theItem 60
*
*   ....
*   Process the item
*   ....
C           EndDo
*
C           EndIf
```

## SelectRcd

コンテナ中のレコードを選択します。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

選択するレコードのレコード ID

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CN1': 'SelectRcd') = id
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'CN1'          SETATR      id          'SELECTRCD'
*
```



## SelPrinter

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

Windows システムの「プリンター選択」ダイアログを表示します。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

使用可能な値は 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('*Component': '*Component':
C          'SelPrinter') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C '*Component' SETATR 1          'SELPRINTER'
*
```

## Sequence

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

**InsertItem** または **SetItem** 属性の設定時にパーツのリスト部分に項目を追加する順序を決定します。

**Sequence** 属性を変更しても、現在のリストの順序には影響しません。順序は、次回に **InsertItem** または **SetItem** 属性が設定された時に変更されます。

**タイプ** 数値

**適用可能なパーツ**

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

リストに追加する項目の順序は次のようになります。

- 1 項目は昇順に挿入されます
- 2 項目は降順に挿入されます
- 3 項目は **Index** 属性で指定された位置に挿入されます
- 4 項目はリストの終わりに挿入されます

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'LB1': 'Sequence') = 1
*
C          EVAL      rc = %getatr('win01': 'LB1': 'Sequence')
```

## 固定形式の例

リスト・ボックスの順序の変更:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      GETATR  'SEQUENCE'  sequence      1 0
*
C  sequence   IFEQ    1
C  'LB1'      SETATR  2          'SEQUENCE'
*
C          ELSE
C  sequence   IFEQ    2
C  'LB1'      SETATR  3          'SEQUENCE'
*
C          ELSE
C  'LB1'      SETATR  2          'SEQUENCE'
C          ENDIF
*
C          ENDIF
*
```

## SetItem

リスト中の既存の項目を置き換えます。置き換えられる項目は、このパーツの現在の **Index** 属性の値によって決定されます。

リストに新しい項目を挿入するためには、**InsertItem** 属性を使用してください。

**Sequence** 属性がソート順序 (すなわち、昇順または降順) を指定している場合には、更新される項目の位置は **Index** 属性で指示された位置と異なる場合があります。

**タイプ** スtring

適用可能なパーツ

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

新規作成項目のテキスト

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'LB1': 'Index') = 5
C          EVAL      %setatr('win01': 'LB1': 'SetItem') = 'new'
*
```

固定形式の例

リスト・ボックス項目の現在の値の変更:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'LB1'      SETATR   5          'INDEX'
C 'LB1'      SETATR   'New'     'SetItem'
*
```

## SetRcdFld

コンテナ・レコードの中のフィールドにデータを設定します。この属性を使用する前に、**RecordID** 属性を設定してコンテナ・レコードを指示し、**ColNumber** 属性を設定してそのレコードのどのフィールド番号を更新するかを指示する必要があります。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

コンテナ・フィールド中に設定するデータ

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CN1': 'SetRcdFld') = 'Test'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CN1'      SETATR  'Test'      'SetRcdFld'
*
```

## SetRcdIcon

ツリー・ビューまたはアイコン・ビュー内でこのレコードを表すためにアイコンにファイル名を設定します。Java アプリケーションでは、この属性がすべてのビューに適用されます。どのレコードを使用するかを指示するためには、**RecordID** 属性を設定しなければなりません。

**タイプ** スtring

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| コンテナー |  |  |
|-------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

アイコンのファイル名

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CN1': 'SetRcdIcon')
C          = 'C:\TEST.ICO'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CN1'          SETATR  'C:\TEST.ICO' 'SetRcdIcon'
*
```

## SetRcdText

ツリー・ビューでコンテナが表示される時に、テキストをアイコン付きで表示するよう設定します。どのレコードを使用するかを指示するためには、**RecordID** 属性を設定しなければなりません。

タイプ スtring

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

レコード・テキスト

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'CN1': 'SetRcdText') = text
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C   'CN1'          SETATR  text          'SETRCDTEXT'
```

## SetTop

指定された項目がリストの最上部にスクロールされるように、パーツ内でリストをセットします。

リスト・ボックスがスクロール・バーを持っていない (すなわち、リスト中のすべての項目が現在表示されている) 場合には、リストはスクロールされません。

**タイプ** 数値

**適用可能なパーツ**

| 組み合わせボックス* | コンテナ | リスト・ボックス |
|------------|------|----------|
| サブファイル     |      |          |

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

リストの最上部に移動される項目の索引。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'LB1': 'SetTop') = 5
*
```

**固定形式の例**

最初に選択した項目をリストの最上部にスクロール:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      GETATR  'FIRSTSEL'  index      2 0
*
C  index      IFGT    *ZERO
C  'LB1'      SETATR  index      'SETTOP'
C  ENDIF
*
```



## TopItem

現在、リスト・ボックスの上部にある項目の項目索引を戻します。

リスト・ボックスに項目がない場合、0 を戻します。

**タイプ** 数値

**適用可能なパーツ**

|          |  |  |
|----------|--|--|
| リスト・ボックス |  |  |
|----------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**戻り値** リストの上部にある項目の索引 (リスト・ボックスが空の場合は 0)。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      item=%getatr('win01': 'LB1': 'TopItem')
*
```

**固定形式の例**

リストの上部にある項目の索引を取得します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'LB1'      GETATR  'TopItem'   index          2 0
*
C          ENDIF
*
```

## SflNxtChg

サブファイル・レコードが変更されたかどうかを示します。影響を受けるレコードを指示するためには、**Index** 属性を設定する必要があります。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

レコードが設定されていることを指示するためには、1 に設定してください。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('window': 'SFX': 'Index') = 12
C          EVAL    %setatr('window': 'SFX': 'SflNxtChg') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          'SFX'    SETATR  12          'INDEX'
C          'SFX'    SETATR   1          'SFLNXTCHG'
*
```

## ShData

設定された共用データ・スペースから読み出される (またはそこに書き込まれる) ユーザー・データを指定します。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('*Component':'*Component':'ShData')='DTA'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      '*Component' SETATR      'DTA'          'SHDATA'
*
```

## ShDataLen

共用データから使用するデータのサイズを指定します。デフォルトは 1024 バイトです。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('*Component':'*Component':'ShDataLen')=100
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      '*Component' SETATR  100      'SHDATALEN'
*
```

## ShDataName

アプリケーション間で共用できるグローバル・データの名前を指定します。各共用データ・サイズは 1024 バイトに固定されています。

**ShDataName** を設定すると、システムは名前付き共用データ・スペースが存在するかどうかを検査します。存在していない場合には、それが作成されます。

**注:** 使用できる名前付き共用メモリのインスタンスは 1 つだけです。プロセス間通信の場合に共用メモリを使用できるのは、コンポーネントでなく、アプリケーションだけです。コンポーネントの場合には、プロセス間通信はパラメータを使用すれば効果的に実行されます。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('*Component': '*Component': 'ShDataName')='DTAN'
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      '*Component' SETATR      'DTAN'      'SHDATANAME'
*
```

## ShDataPos

参照される共用データ内の位置を指定します。デフォルトの開始位置は 1 です。  
**ShDataLen** と **ShDataPos** の値は、両方で 1024 を超えることはできません。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('*Component':'*Component':'ShDataPos')=100
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C      '*Component' SETATR  100          'SHDATAPOS'
*
```

## ShowMsgID

DSPLY 命令コードが指定された時に、メッセージ・ボックス内のメッセージ ID (MSGnnnn) の表示を使用可能または使用不能にします。

タイプ 数値

適用可能なパーツ

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

**ShowMsgID** は次の 1 つとすることができます。

- 0      メッセージ ID を表示しません。
- 1      メッセージ ID を表示します。デフォルト。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('*Component':'*Component':'ShowMsgID')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      '*COMPONENT' SETATR      1          'SHOWMSGID'
*
```

## ShowProp

パーツにオブジェクトのプロパティ・ページ・ダイアログを表示させるには、この属性を 1 に設定してください。

タイプ 数値

適用可能なパーツ

|         |           |  |
|---------|-----------|--|
| ActiveX | Java Bean |  |
|---------|-----------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'AX1':'ShowProp=') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AX1'      SETATR  1          'SHOWPROP'
*
```



## ShowRects

ユーザー定義の強調表示長方形が表示されるかどうかを示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**ShowRects** は、次のいずれかとすることができます。

**0** 長方形は表示されません

**1** 長方形は表示されます

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      isshown = %getatr('win01': 'CAL1': 'ShowRects')
*
C          EVAL      %setatr('win01': 'CAL1': 'ShowRects') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'CAL1'    GETATR   'SHOWRECTS'  isshown          1 0
*
```

## ShowTabs

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

この属性を 0 に設定すると、ノートブック・パーツ中のすべてのタブは隠されます。

**タイプ** 数値

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| ノートブック |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      %setatr('win01': 'WIN1': 'ShowTabs') = 0
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'WIN1'          SETATR      0          'SHOWTABS'
*
```

## ShowText

ユーザー定義のテキストが表示されるかどうかを示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**ShowText** は、次のいずれかとすることができます。

- 0 テキストは表示されません
- 1 テキストは表示されます

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      isshown = %getatr('win01': 'CAL1': 'ShowText')
*
C          EVAL      %setatr('win01': 'CAL1': 'ShowText') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'CAL1'   GETATR   'SHOWTEXT'   shown           1 0
*
```

## ShowTips

ツール・ヒントを特定のパーツに対して表示するか、ウィンドウ内のすべてのパーツに対して表示するかを決定します。**ShowTips** をウィンドウに対して設定すると、この値はウィンドウ内の特定のパーツに設定された値に優先します。たとえば、ウィンドウに対して **TipText** の表示を使用不可にした場合には、その設定はウィンドウ内のすべてのパーツにも適用されます。

### タイプ 数値

#### 適用可能なパーツ

|           |                |        |
|-----------|----------------|--------|
| チェック・ボックス | 組み合わせボックス      | コンテナ   |
| 入力フィールド   | グラフィック・プッシュボタン | イメージ   |
| リスト・ボックス  | メッセージ・サブファイル   | 複数行編集  |
| プッシュボタン   | ラジオ・ボタン        | スライダー  |
| スピン・ボタン   | 静的テキスト         | サブファイル |
| ウィンドウ     |                |        |

### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

### 使用可能な値

**ShowTips** は、次のいずれかとすることができます。

- 0 ウィンドウ内の特定のパーツまたはすべてのパーツについてツール・チップは表示されません。
- 1 **TipText** が定義されたウィンドウ内のパーツについてツール・チップが表示されます。

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      isshown = %getatr('win01': 'Win01': 'ShowTips')
*
C                               EVAL      %setatr('win01': 'Win01': 'ShowTips') = 1
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'Win01'      GETATR  'SHOWTIPS'  shown          1 0
*
```

## SizeToFit

リスト内の最後の項目が完全に可視になるように、パーツを自動的にサイズ変更します。

タイプ 数値

適用可能なパーツ

|            |          |        |
|------------|----------|--------|
| 組み合わせボックス* | リスト・ボックス | サブファイル |
|------------|----------|--------|

**\*制約事項:** この属性は、このパーツに対する Java アプリケーションでは実行時にサポートされません。

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**SizeToFit** は次のいずれかとすることができます。

**0** パーツをサイズ変更しません。

**1** パーツを自動的にサイズ変更します。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'CB1': 'SizeToFit') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'CB1'      SETATR  1          'SIZETOFIT'
```

## SortAsc

コンテナまたはサブファイル内のレコードを昇順にソートします。テキスト・データを持つ列はアルファベット順にソートされます。

**タイプ** 数値

**適用可能なパーツ**

|      |        |  |
|------|--------|--|
| コンテナ | サブファイル |  |
|------|--------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

ソートされる列番号。

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CTR1': 'SortAsc') = 2
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'CTR1'      SETATR  2          'SORTASC'
*
```

## SortDesc

コンテナまたはサブファイル内のレコードを降順にソートします。テキスト・データを持つ列はアルファベット順にソートされます。

タイプ 数値

適用可能なパーツ

|      |        |  |
|------|--------|--|
| コンテナ | サブファイル |  |
|------|--------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

ソートされる列番号。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C          EVAL      %setatr('win01': 'CTR1': 'SortDesc') = 2
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C  'CTR1'      SETATR  2          'SORTDESC'
*
```

## SQLError

直前の演算命令が成功したかどうかを示します。0 の値が戻された場合には、直前の演算命令は成功しました。これ以外の戻り値は、エラーが起こったことを示します。

タイプ 数値

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      Err=%getatr('win01':'ODBC1':'SQLError')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'ODBC1'      GETATR  'SQLERROR'  ERR
*
```



## SQLMsgBox

テーブルの処理中にエラーが起こった時に、ODBC メッセージ・ボックスを抑制または表示します。

この属性を 0 に設定してメッセージ・ボックスを抑制すれば、**SQLError** 属性を使用して、エラーが起こったかどうかを判別することができます。これにより、**Handle** 属性を使用し、ODBC API を呼び出して追加の情報を戻すことができます。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

**SQLMsgBox** は、次のいずれかとすることができます。

- 0**      メッセージ・ボックスを表示しません。
- 1**      メッセージ・ボックスを表示します。これがデフォルトです。

**フリー・フォームの例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                    EVAL      %setatr('win01':'ODBC1':'SQLMsgBox')=0
*
```

**固定形式の例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C    'ODBC1'        SETATR    0                    'SQLMSGBOX'
*
```

## SQLQuery

**ExecuteSQL** 属性が 0 に設定された時に実行する SQL ステートメントを定義します。

タイプ スtring

適用可能なパーツ

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Query = 'Select * From "Customers"'
C          EVAL      %setatr('win01':'ODBC1':'SQLQuery')=Query
C          EVAL      %setatr('win01':'ODBC1':'ExecuteSQL')=1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
*
D Query          C          'Select * From "Customers"'
*
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'ODBC'        Setatr  Query      'SQLQuery'
C  'ODBC'        Setatr  1         'ExecuteSQL' *
```

## StartAt

READC 演算命令を、指定したサブファイルで開始するように強制します。この属性は、サブファイル内のレコードを検査して変更されたレコードを戻すために READC を対話式に使用するプログラムに有用です。444 ページの『StartAt の例』を参照してください。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

開始するレコード位置

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('win01': 'SFL1': 'StartAt')=4
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SFL1'    SETATR  4          'STARTAT'
```

## StartAt の例

次のコード・セグメントは、サブファイル内のレコードを検査して、変更されたすべてのレコードを戻して処理するために READC 演算命令を対話式に使用するものです。OPT フィールドは、入力が正しいかどうかをモニターします。OPT 値が正しくない場合には、プログラムは SflNxtChg 属性を使用して正しくないレコードに変更済みのフラグを立てます。StartAt 属性は、エラーがあったレコードの後のレコードから次の READC 演算命令を開始させます。

デフォルトでは、READC は最初のサブファイル・レコードから変更済みレコードの検索を開始します。変更されたレコードが見つかってプログラムに戻されると、その状況が 未変更 にリセットされます。この例では、StartAt 属性も、READC 演算命令による SflNxtChg でフラグを立てられたレコードからのループを防止しています。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
* サブファイルには OPT と呼ばれるフィールドがあって、その値は
* 0 より大きくなければならないとします。
*
C          ReadC      SFL1                99
C          Dow        NOT *IN99
*
C          If         Opt > 0
*
* レコードを処理します。
C          Else
*
* レコードは無効です。したがって、次の反復で読み取るように SFLNXTCHG を設定します。
C          'SFL1'     Getatr   'Index'      X
C          'SFL1'     Setatr   X            'Index'
C          'SFL1'     Setatr   1            'SFLNXTCHG'
*
* エラーがあったレコードの後から読み取りを開始するように STARTAT を設定します。
C          Eval      X = X + 1
C          'SFL1'     Setatr   X            'StartAt'
C          ReadC     SFL1                99
C          EndDo *
```

## StartNew

この属性を 1 に設定すると、既存のデータはわきに寄せられて、グラフを描き続けるために保存されます。設定された新しいデータは **UseData** 属性が設定されるまで、グラフには使用されません。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

使用可能な値は 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'GRA1': 'StartNew') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GRA'      SETATR  1          'STARTNEW'
*
```

## StatusBar

簡単な 1 ペイン進行状況バーをウィンドウに追加したり、ウィンドウから除去したりします。デフォルトの位置は、ウィンドウの最下部です。複数のペインおよびアクション情報を持つ各種の状況バーが必要な場合には、状況バー・パーツを使用してください。

**タイプ** 数値

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

**StatusBar** は、次のいずれかとすることができます。

- 1** 状況バーを作成します
- 0** 状況バーを除去します

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'WIN1':'StatusBar')=1
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C 'WIN1'      SETATR  1          'STATUSBAR'
*
```

## SwitchTo

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

この属性が設定された時にプログラムのインスタンスをもう 1 つ開始すると、このプログラム・フォーカスの現在のインスタンスが与えられます。**SwitchTo** 属性は、フォーカスを取得するプログラム内のウィンドウのハンドルを、その値として受け入れます。ウィンドウがアイコン化されている場合には、それが復元されます。

このハンドルは 32 ビットの値で、長さが 10 で小数点以下の桁数が 0 のタイプ I (符号なし) として D 仕様で宣言する必要があります。

**タイプ** 数値

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++ResulT+++++Len++D+HiLoEq---
*
C          EVAL      hWnd=%getatr('Win1':'Win1':'Handle')
C          EVAL      %setatr('*Component':'*Component':'SwitchTo')=hWnd
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++ResulT+++++Len++D+HiLoEq---
*
C      'WIN1'      GETATR  'HANDLE'      HWND
C      '*COMPONENT' SETATR  HWND            'SWITCHTO'
```

## TabImage

ノートブック・ページ・タグのラベルの横に表示するイメージの名前を指定します。サポートされるイメージ・タイプのリストについては、**FileName** 属性を参照してください。\*BLANK が指定されたかまたはファイル名が指定されない場合には、イメージは除去されます。

**タイプ** スtring

適用可能なパーツ

|            |                   |  |
|------------|-------------------|--|
| ノートブック・ページ | キャンバス付きノートブック・ページ |  |
|------------|-------------------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'NP1':'TabImage') = 'N1.ICO'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'NP1'      SETATR  'N1.ICO'    'TABIMAGE'
*
```



## TabLabel

ノートブック・ページ・タブに表示されるラベルを定義します。影響を受けるページを指示するためには、最初に **PageNumber** 属性を設定しなければなりません。

タイプ スtring

適用可能なパーツ

|            |  |  |
|------------|--|--|
| ノートブック・ページ |  |  |
|------------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。タブ・テキスト・フィールドに値を指定することによって、これを行ってください。

使用可能な値

ノートブック・ページのタブに表示されるテキスト

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'NBP1': 'TabLabel')
C          = 'Section A'
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'NBP1'      SETATR  'Section A'  'TABLABEL'
*
```

## Text

パーツの入力フィールド部分のテキスト値を設定するかまたは戻します。

**タイプ** スtring

**適用可能なパーツ**

|           |         |       |
|-----------|---------|-------|
| 組み合わせボックス | 入力フィールド | 複数行編集 |
| スピン・ボタン   |         |       |

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** 複数行編集またはスピン・ボタン・パーツを除き、GUI Designer でプロパティ・ノートブックを使用して設定することができます。組み合わせボックス・パーツでは、**データ**・タブを使用し、値を入力することによって、これを行ってください。入力フィールド・パーツでは、「**一般**」タブを使用し、「**テキスト**」フィールドに値を入力することによって、これを行ってください。

**使用可能な値**

入力フィールドの値

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      oldstring = %getatr('win01': 'EF1': 'Text')
*
C          EVAL      %setatr('win01': 'EF1': 'Text') = 'Newstring'
*
```

**固定形式の例**

入力フィールドの値を保管してからそれを変更:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'EF1'      GETATR  'TEXT'      oldstring      64
C  'EF1'      SETATR  'Newstring'  'TEXT'
```

## TextEnd

最後に選択された文字の文字オフセットを決定します。テキストが選択されていない場合には、値 0 が戻されます。テキストのマウス・ポインターを位置指定してから、マウス・ボタンを押して、マウス・ポインターをテキストを横切って移動することによってテキストが選択されます。

この属性は **TextStart** と一緒に使用して、パーツの特定のテキスト部分を選択し、選択したテキストの長さを決定することができます。

**タイプ** 数値

**適用可能なパーツ**

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'TextEnd') = value
*
C          EVAL      value = %getatr('win01': 'MLE1': 'TextEnd')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MLE1'    GETATR  'TEXTEND'    endoffset      3 0
*
```

## TextLength

パーツ中のテキストの長さを戻します。

タイプ 数値

適用可能なパーツ

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      length = %getatr('win01': 'EF1': 'TextLength')
*
```

固定形式の例

入力フィールドのテキストの長さを取得:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'EF1'      GETATR  'TEXTLENGTH' length          3 0
*
```

## TextSelect

パーツから選択されたテキストを戻します。テキストが選択されていない場合には、空文字列が戻されます。

フィールドのテキストを選択するためには、**TextStart** および **TextEnd** 属性を設定してください。

**タイプ** スtring

**適用可能なパーツ**

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      value = %getatr('win01': 'MLE1': 'TextSelect')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'MLE1'      GETATR  'TEXTSELECT'  seltext      100
*
```

## TextStart

選択されたテキスト・フィールドの最初の文字を判別します。テキストが選択されていない場合には、値 0 が戻されます。

この属性は **TextEnd** と一緒に使用して、パーツの特定のテキスト部分を選択することができます。

タイプ 数値

適用可能なパーツ

|         |       |  |
|---------|-------|--|
| 入力フィールド | 複数行編集 |  |
|---------|-------|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'MLE1': 'TextStart') = value
*
C          EVAL      value = %getatr('win01': 'MLE1': 'TextStart')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'MLE1'    GETATR   'TEXTSTART'  endoffset      3 0
*
```

## TextString

複数行編集パーツ内のすべてのテキストを戻し、すべての復帰改行文字をブランクに変換します。

タイプ スtring

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      ATxt=%getatr('Win01':'ML1':'TextString')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      'ML1'      GETATR      'TEXTSTRING' ATXT
*
```

## TickLabel

文字列をスライダー・パーツの目盛りマークに割り当てます。**TickNumber** 属性は、どの **TickLabel** が参照されるかを決定します。

**タイプ** 文字列

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| スライダー |  |  |
|-------|--|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。「追加」プッシュボタンを選択し、「目盛りテキストの追加」ダイアログ・ボックスに値を入力することによって、これを行ってください。

**使用可能な値**

スライダーの目盛りに割り当てられたラベル。

**フリー・フォームの例**

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          Z-ADD    0          percent
*
C          DO      5          ticknbr
C          EVAL    %setatr('win01':'SLIDER1':'TickNumber')=ticknbr
C          EVAL    %setatr('win01':'SLIDER1':'TickLabel')=percent
C          ADD     25          percent
C          ENDDO
*
```

**固定形式の例**

スライダーの目盛りのテキストを変更:

```
*...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          Z-ADD    *ZERO     value          2 0
C  1      DO      11         tick            2 0
C  'SLIDER1' SETATR tick      'TICKNUMBER'
C  'SLIDER1' SETATR value    'TICKLABEL'
C          ADD     10         value
C          ENDDO
*
```



## TickNumber

次の **TickLabel** テキストを入れるスライダー目盛りを指定します。

番号は、1 からノートブックの設定でスライダーに指定された目盛りの数までの範囲の値になります。

**タイプ** 数値

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| スライダー |  |  |
|-------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

目盛り番号

**フリー・フォームの例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      x = %getatr('win01': 'SLIDER1': 'TickNumber')
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'SLIDER1'  GETATR  'TICKNUMBER'  OldTick          3 0
*
C  'SLIDER1'  SETATR  NewTick      'TICKNUMBER'
```

## Tile

1 に設定されていると、キャンバス上のイメージがタイル表示されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| キャンバス |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**Tile** は、次のいずれかとすることができます。

**0** イメージはタイル表示されません。

**1** イメージがタイル表示されます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CAN': 'Tile') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CAN1'      SETATR  1          'TILE'
*
```

## TimeOut

この値は、**DDEMode** を *Initiate* に設定した後で、DDE 会話を確立するために DDE クライアント側がどれだけ待機するかを決定します。この値は秒数で表されます。

この時間内に DDE 会話が完了しないと、**Timeout** イベントがユーザー・プログラムに警告します。

**タイプ** 数値

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'DDE1': 'TimeOut') = 5
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'DDE1'      SETATR  5          'TIMEOUT'
```

## TimerMode

タイマー・パーツの現在の操作モードを決定します。このモードが *Start* に設定されると、タイマーは **Tick** イベントの生成を開始します。このモードを *Stop* に設定すると、**Tick** イベントが生成されます。 **TimerMode** が *Pause* に設定されると、タイマーは **Tick** イベントの生成を中止しますが、**TimerTicks** 属性の更新を続行します。

**タイプ** 数値

**適用可能なパーツ**

|      |  |  |
|------|--|--|
| タイマー |  |  |
|------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**使用可能な値**

**TimerMode** は、次のいずれかとすることができます。

- 1 Start
- 2 停止
- 3 休止

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C   PB1           BEGACT   PRESS
C                   EVAL     %setatr('win01': 'TIMER01': 'TimerMode') = 1
C                   ...
C                   ENDACT
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C   'TIMER1'     SETATR   1           'TIMERMODE'
*
```

## TimerTicks

タイマーの現在の目盛り値を決定します。**TimerTicks** の値は、タイマーが **Tick** イベントを生成した回数です。

**Interval** と **Multiplier** 属性も tick イベントの数を決定することに注意してください。**TimerTicks** 属性は次のようにして決定されます。

$$\text{TimerTicks} = (\text{経過時間 (ミリ秒)}) / (\text{インターバル} * \text{乗数})$$

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| タイマー |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

現在の時間目盛りカウント

フリー・フォームの例

タイマーの目盛り値の変更:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      count=%getatr('win01':'TIMER01':'TimerTicks')
*
C  count      IFGT      1000
C          EVAL      %setatr('win01':'TIMER01':'TimerTicks')=0
C          ENDIF
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'TIMER01'  GETATR   'TIMERTICKS'  count          6 0
*
C  count      IFGT      1000
C  'TIMER01'  SETATR   0          'TIMERTICKS'
C          ENDIF
*
```

## TipText

ツール・ヒント制御にテキストを表示するよう指定します。

グラフ・パーツの場合には、**TipText** をデータ・ポイントと関連付けます。たとえば、バー・タイプの場合には、データ・ポイントを表すバーの横にツール・ヒント・テキストが表示されます。

カレンダー・パーツの場合には、**TipText** を **Dateldx** 属性と関連付けます。ツール・ヒントはその日付と関連付けられます。

**タイプ** スtring

適用可能なパーツ

|                |           |           |
|----------------|-----------|-----------|
| カレンダー          | チェック・ボックス | 組み合わせボックス |
| コンテナ           | 入力フィールド   | グラフ       |
| グラフィック・プッシュボタン | イメージ      | リスト・ボックス  |
| メッセージ・サブファイル   | 複数行編集     | プッシュボタン   |
| ラジオ・ボタン        | スライダー     | スピン・ボタン   |
| 静的テキスト         | サブファイル    |           |

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | いいえ      |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

最大 80 文字のテキストまたはメッセージ ID (\*MSGnnn)

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'MLE1': 'TipText')='*msg0002'
*/xmp>
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C  'MLE1'      SETATR  'Hello'      'TipText'
*
```

## Title

グラフのタイトル・テキストを決定します。

タイプ スtring

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      value = %getatr('win01': 'GRA1': 'Title')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'GRA1'      GETATR  'TITLE'      seltitle      1 0
*
```

## TitlePlace

グラフ内のどこにタイトルを入れるかを示します。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

**TitlePlace** は、次のいずれかとすることができます。

- 0** タイトルなし
- 1** グラフの上
- 2** グラフの下

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'GRA1': 'TitlePlace') = 3
*
C          EVAL      where = %getatr('win01': 'GRA1': 'TitlePlace')
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'GRA1'    GETATR   'TITLEPLACE' where          1 0
*
```



## Top

パーツまたはオブジェクトの画面の上の座標をピクセル数で指定します。

**注:** 原点は左上隅にあり、ゼロで始まります。

**タイプ** 数値

**適用可能なパーツ**

|               |                |                    |
|---------------|----------------|--------------------|
| ActiveX       | アニメーション制御      | カレンダー              |
| キャンバス*        | チェック・ボックス      | 組み合わせボックス          |
| コンテナ          | DDE クライアント     | 入力フィールド            |
| グラフ           | グラフィック・プッシュボタン | グループ・ボックス          |
| 水平方向のスクロール・バー | イメージ           | Java Bean          |
| リスト・ボックス      | メディア           | メディア・パネル           |
| 複数行編集         | ノートブック         | ODBC/JDBC インターフェース |
| 外枠            | 進行状況バー         | プッシュボタン            |
| ラジオ・ボタン       | スライダー          | スピン・ボタン            |
| 静的テキスト        | サブファイル         | タイマー               |
| 縦方向のスクロール・バー  | ウィンドウ          |                    |

\* **制約事項:** キャンバス・パーツの場合には、この属性は実行時にのみ取り出すことができます。

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\***注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

ピクセルの数

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      otop = %getatr('win01': 'Image1': 'Top')
C  otop    SUB       200          newtop
C          EVAL      %setatr('win01': 'Image1': 'Top') = newtop
*
```

## 固定形式の例

そのウィンドウ内のイメージ 200 ピクセルを移動します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'IMAGE1'  GETATR  'TOP'      OldTop      4 0
C  OldTop    SUB     200      NewTop      4 0
C  'IMAGE'   SETATR  NewTop    'TOP'
*
```

## Topic

**Topic** 属性は、DDE 会話のサーバー・アプリケーションに会話の主題を通知するために設定します。会話を開始する前に、**AppName** 属性も設定する必要があります。

DDE トピックはサーバー・アプリケーションが処理できるデータ集合です。スプレッドシート・アプリケーションの場合には、これはスプレッドシート・ファイルになります。

サーバー・プログラムが VisualAge RPG コンポーネントの場合には、この属性の構文は次のようになります。

```
'appname|appname'
```

この場合に、*appname* は VisualAge RPG コンポーネントの名前です。たとえば、サーバー・プログラムが *server* という名前の VisualAge RPG コンポーネントの場合には、この属性は次のように設定されます。

```
EVAL %setatr('WIN1':'DDE1':'Topic') = 'SERVER|SERVER'
```

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL    %setatr('win01': 'DDE1': 'AppName') = 'APP01'
C          EVAL    %setatr('win01': 'DDE1': 'Topic') = 'APP01.WG1'
C          EVAL    %setatr('win01': 'DDE1': 'DDEMode') = 1
C          ...
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          'DDE1'      SETATR  'APP01.WG1'  'TOPIC'
C          *
```

## TopLine

複数行編集フィールドの先頭にする行番号を示します。**TopLine** がゼロに設定されると、複数行編集は最終ページに表示されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

行番号

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'MLE1': 'TopLine') = 0
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'MLE1'      SETATR  0          'TOPLINE'
```

## TopRecord

指定された索引項目がリストの最上部にスクロールされるか、あるいは最上部の可視行の索引を取得する (戻す) ように、サブファイル・パーツ内のリストを設定します。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      itemtop = %getatr('win01': 'SFI1': 'TopRecord')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C   'SFI1'      GETATR  'TOPRECORD'  itemtop
*
```

# Treble

注: この属性は Windows にのみ適用されます。

**Treble** 属性は、オーディオ・ファイルのトレブル・レベルを設定します。このレベルは、0 - 100 に設定することができます。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| メディア |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

0 - 100 の値でなければなりません

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'AUD1': 'Treble') = 42
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'AUD1'      SETATR  42          'TREBLE'
```

## UnBind

**\*制約事項:** この属性は、Java アプリケーションにのみ適用されます。

前にバインドされたパーツをアンバインドするには、この属性を 1 に設定します。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'OD1':'UnBind')=1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'OD1'      SETATR  1          'UNBIND'
*
```

## UnderPoint

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

この属性を、グラフ・パーツの左上隅からの相対位置を表す文字列 (たとえば "25 10") に設定します。次に **HitItem**、**DataPoint**、および **DataGroup** 属性の値を使用して、与えられた点の下にグラフのどのパーツがあるかを調べてください。

**タイプ** 文字列

**適用可能なパーツ**

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

**使用可能な値**

形式 "x y" の文字列

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      xystring = %getatr('win01': 'GPH1': 'UnderPoint')
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'GPH1'      GETATR  'UNDERPOINT' xystring
*
```



## Undo

行った直前の変更をやり直すには、この属性を 1 に設定します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01': 'ML1': 'Undo') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq----
*
C 'ML1'      SETATR  1          'UNDO'
*
```

## UpdateRow

結果セット内の更新する行番号を決定します。先に行を取り出して更新する必要はありません。この属性によって、いずれかの行が更新されます。

取り出した行を更新するには、**CurrentRow** 属性を使用してその行番号を取得してください。

**注:** Java アプリケーションでは、この属性に適切な JDBC 2.0 承諾ドライバーが必要です。

**タイプ** 数値

**適用可能なパーツ**

|                    |  |  |
|--------------------|--|--|
| ODBC/JDBC インターフェース |  |  |
|--------------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C          EVAL      %setatr('win01':'ODBC1':'UpdateRow')=4
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+Hi LoEq----
*
C 'ODBC1'    SETATR  4          'UPDATEROW'
```

## UseData

最後に **StartNew** が設定されてから後でこのグラフに設定されたデータを使用して開始するためには、この属性を 1 に設定してください。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

使用可能な値は 1 です。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C          EVAL      %setatr('win01': 'GRA1': 'UseData') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+Hi LoEq---
*
C  'GRA1'      SETATR  1          'USEDATA'
```

## UseDelim

区切り文字が関連付けられているデータを識別するには、この属性を 1 に設定します。区切り文字 (あれば) を指定するには、**DelimChar** 属性を使用します。

タイプ 数値

適用可能なパーツ

|           |          |  |
|-----------|----------|--|
| 組み合わせボックス | リスト・ボックス |  |
|-----------|----------|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CB1': 'UseDelim') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CB1'      SETATR  1          'USEDELIM'
```

## UserData

ユーザー定義のストリングをパーツと関連付けることができます。このストリングには、長さの制限事項はありません。

**タイプ** ストリング

**適用可能なパーツ**

\*Component パーツを除くすべてのパーツ

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

パーツと関連付けるデータ。

**フリー・フォームの例**

入力フィールドの値を保管してからそれを変更:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL    oldstring = %getatr('win01': 'EF1': 'UserData')
*
C          EVAL    %setatr('win01': 'EF1': 'UserData') = 'Newstring'
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'EF1'      GETATR  'USERDATA'  oldstring      64
C 'EF1'      SETATR  'Newstring' 'USERDATA'
```

## Validate

この属性が設定された時、適用されるボタンを押すと、フィールドの妥当性検査の設定があつてボタンと同じウィンドウにあるパーツに対してフィールドの妥当性検査が行われます。フィールドの妥当性検査には、範囲検査および値検査などが含まれます。

フィールドの妥当性検査中にエラーが見つかり、メッセージが表示されて **Press** イベントは知らされません。

エラーがない場合、あるいはフィールドの妥当性検査をともしないフィールドがない場合には、press イベント・アクション・サブルーチンが呼び出されます。

**注:** ウィンドウのプッシュボタンまたグラフィック・プッシュボタンに **Validate** 属性が設定されていない場合には、フィールドの属性検査は行われません。

**タイプ** 数値

**適用可能なパーツ**

|                |         |  |
|----------------|---------|--|
| グラフィック・プッシュボタン | プッシュボタン |  |
|----------------|---------|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。フィールド妥当性検査の実行スタイルを選択することによって、これを行なってください。

**使用可能な値**

**Validate** は、次のいずれかとすることができます。

- 0** 入力の妥当性検査を実行しません。
- 1** 入力の妥当性検査を実行します。

## フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      validate = %getatr('win01': 'EF01': 'Validate')
*
C          EVAL      %setatr('win01': 'EF01': 'Validate') = 1
*
```

## 固定形式の例

入力フィールドの値を保管してからそれを変更:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'EF1'      GETATR  'VALIDATE'  validate      1 0
C 'EF1'      SETATR  1          'VALIDATE'
*
```

## Value

スライダー・パーツの値を戻します。この値はスライダーに割り当てられた最小値より大きいか等しくて、最大値より小さいか等しくなります。

スピン・ボタンの場合には、入力フィールドの現在の内容が戻されます。

**タイプ** 数値

**適用可能なパーツ**

|       |         |  |
|-------|---------|--|
| スライダー | スピン・ボタン |  |
|-------|---------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

スライダーまたは入力フィールドの値

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      oldvalue = %getatr('win01': 'SLIDER1': 'Value')
*
C          EVAL      %setatr('win01': 'SLIDER1': 'Value') = 5
*
```

**固定形式の例**

入力フィールドの値を保管してからそれを変更:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          'SLIDER1'  GETATR  'VALUE'      OldValue      3 0
C          'SLIDER1'  SETATR   5           'VALUE'
*
```



## VCoIFRDCol

**ColNumber** 属性によって指定された対応するデータ列の表示列を戻します。

タイプ 数値

適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'SF1':'ColNumber') = 4
C          EVAL      ViewCol4 = %getatr('win01':'SF1':'VCoIFRDCol1')
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'SF1'      SETATR   4          'COLNUMBER'
C 'SF1'      GETATR   'VCoIFRDCol' VIEWCOL4
*
```

## View

コンテナ中のレコードが表示される方法を決めます。コンテナでは、レコードをアイコン、ツリー、詳細、またはテキスト・ツリー・ビューで表示することができます。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

**\*制約事項:** Java アプリケーションでは、このパーツの場合にのみ実行時にこの属性を取り出すことができます。

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

コンテナ・ビューは次の 1 つとすることができます。

- 1 アイコン・ビュー
- 2 ツリー・ビュー
- 3 詳細ビュー
- 4 テキスト・ツリー・ビュー

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL      %setatr('win01': 'CN1': 'View') = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'CN1'          SETATR  1          'VIEW'
*
```

## ViewColumn

サブファイルのデータ列が表示される順序を示します。この属性は、**ColNumber** (データ列) および **MapViewCol** 属性と一緒に使用され、データ列が表示される順序を変更します。デフォルトでは、データ列が「サブファイル」プロパティ・ノートブックの「フィールド・リスト」ページにリストされたのと同じ順序で表示されます。

**タイプ** 数値

**適用可能なパーツ**

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

データ列 2 (ColNumber) を表示可能列 4 (ViewColumn) として表示します。たとえばデータ列が以下のように定義されていたとします。

```
データ列
1      2      3      4
Custno Name  Address  City
```

この演算命令の後の列の表示は以下のようになります。

```
後の表示:
1      2      3      4
Custno  Address  City  Name
```

ビュー中の他のすべての列がビュー中で左にシフトされるということに注意してください。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                               EVAL    %setatr('win01': 'SF1': 'ColNumber') = 2
C                               EVAL    %setatr('win01': 'SF1': 'ViewColumn') = 4
C                               EVAL    %setatr('win01': 'SF1': 'MapViewCol') = 1
*
```

**固定形式の例**

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++0opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C   'SF1'          SETATR  2          'COLNUMBER'
C   'SF1'          SETATR  4          'VIEWCOLUMN'
C   'SF1'          SETATR  1          'MAPVIEWCOL'
*
```

## Visible

パーツが可視であるかないかを決定します。

タイプ 数値

適用可能なパーツ

|               |                |                    |
|---------------|----------------|--------------------|
| ActiveX       | アニメーション制御      | カレンダー              |
| チェック・ボックス     | 組み合わせボックス      | コンポーネント参照          |
| コンテナ          | DDE クライアント     | 入力フィールド            |
| グラフ           | グラフィック・プッシュボタン | グループ・ボックス          |
| 水平方向のスクロール・バー | イメージ           | Java Bean          |
| リスト・ボックス      | メディア           | メディア・パネル           |
| メニュー項目        | メッセージ・サブファイル   | 複数行編集              |
| ノートブック        | ノートブック・ページ     | ODBC/JDBC インターフェース |
| 外枠            | ポップアップ・メニュー*   | 進行状況バー             |
| プッシュボタン       | ラジオ・ボタン        | スライダー              |
| スピン・ボタン       | 静的テキスト         | 状況バー               |
| サブファイル        | タイマー           | 縦方向のスクロール・バー       |
| ウィンドウ         |                |                    |

**\*制約事項:** Windows アプリケーションの場合には、照会 (GETATR) でこの属性に 0 の値が返されます。Java アプリケーションの場合に、この属性はポップアップ・メニューの状態を戻します。

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**Visible** は、次のいずれかとすることができます。

- 0** パーツは非可視です
- 1** パーツは可視です

**注:** Java アプリケーションのポップアップ・メニューの場合には、**InvPname** (呼び出し側の親の名前) と **InvName** (呼び出し側の名前) 属性および呼び出し側の座標空間のポップアップの x-y 座標も指定しなければなりません。

メニュー・バー・パーツでは少なくとも 1 つのメニュー項目が可視でなければなりません。メニュー項目を表すのにビットマップ・イメージを使用する場合には、すべてのメニュー項目がビットマップ・イメージでなければなりません。さもないと、**Visible** 属性がリセットされた時に、リフレッシュの問題が起こります。

## フリー・フォームの例

パーツの可視状態を切り替えます。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C                               EVAL      isVisible = %getatr('win01': 'EF1': 'Visible')
*
C      isVisible               IFEQ       1
C                               EVAL      %setatr('win01': 'EF1': 'Visible') = 0
*
C                               ELSE
C                               EVAL      %setatr('win01': 'EF1': 'Visible') = 1
C                               ENDIF
*
```

## 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C      'EF1'                   GETATR    'VISIBLE'    isVisible    1 0
C      isVisible               IFEQ     1
C      'EF1'                   SETATR    0            'VISIBLE'
*
C                               ELSE
C      'EF1'                   SETATR    1            'VISIBLE'
C                               ENDIF
*
```

## VisTitle

コンテナ・タイトルが可視であるかどうかを判別します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

次の 1 つです。

**0** タイトルを表示しません

**1** タイトルを表示します

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CN1': 'VisTitle) = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CN1'      SETATR  1          'VisTitle'
*
```

## VisTitlSep

コンテナ・タイトルの区切り記号が可視であるかどうかを判別します。

タイプ 数値

適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

使用可能な値

次の 1 つです。

- 0** タイトル区切り記号を表示しません
- 1** タイトル区切り記号を表示します

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'CN1': 'VisTitlSep) = 1
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'CN1'      SETATR  1          'VisTitlSep'
*
```



## Volume

**注:** この属性は Windows にのみ適用されます。

メディア・パーツ、システムの出カ波、およびシンセサイザーのボリュームを設定します。

メディア・パネルの場合には、ボリューム・スライダの設定を判別します。この値の範囲は 0 - 100 です。

**タイプ** 数値

**適用可能なパーツ**

|      |          |  |
|------|----------|--|
| メディア | メディア・パネル |  |
|------|----------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

ボリューム設定。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      curvol = %getatr('win01': 'MMP1': 'Volume')
*
C          EVAL      %setatr('win01': 'MMP1': 'Volume') = 5
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++OpcodE)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'MMP1'    GETATR  'VOLUME'  curvol      3 0
*
C  'MMP1'    SETATR  5          'VOLUME'
```

## VRule

サブファイル・データまたはカレンダー行の各列の間に縦方向に罫線を追加します。少なくとも 1 行のデータがなければなりません。

**タイプ** 数値

**適用可能なパーツ**

|       |        |  |
|-------|--------|--|
| カレンダー | サブファイル |  |
|-------|--------|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**VRule** は次の 1 つとすることができます。

- 0** 罫線を消します
- 1** 罫線を表示します

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      Vtrule = %getatr('win01': 'SF1': 'VRule')
*
C  Vtrule      IFEQ      0
C          EVAL      %setatr('win01': 'SF1': 'VRule') = 1
C          ELSE
C          EVAL      %setatr('win01': 'SF1': 'VRule') = 0
C          ENDIF
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'SF1'      GETATR  'VRULE'      Vtrule      1 0
*
C  'SF1'      SETATR  1          'Vtrule'
```

## WeekDay

選択された曜日を示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

戻される値

次の 1 つです。

- 0** Sunday
- 1** Monday
- 2** Tuesday
- 3** Wednesday
- 4** Thursday
- 5** Friday
- 6** Saturday

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      index = %getatr('win01': 'CAL1': 'WeekDay')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'CAL1'      GETATR  'WEEKDAY'      02
*
```

## WeekDayIdx

**YearIdx**、**MonthIdx**、および **DayIdx** の設定と対応する曜日を戻します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

戻される値

次の 1 つです。

- 0** Sunday
- 1** Monday
- 2** Tuesday
- 3** Wednesday
- 4** Thursday
- 5** Friday
- 6** Saturday

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      index = %getatr('win01': 'CAL1': 'WeekDayIdx')
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C 'CAL1'      GETATR  'WEEKDAYIDX' INDEX
*
```

## Width

パーツの幅をピクセル数で指定します。ウィンドウの幅を変更すると、**ReSize** イベントが生成されます。

**注:** **Width** 属性は、使用されるディスプレイ解像度によって決まる最大幅を超えることはできません。たとえば、640 ピクセル幅の画面で作業している時に幅が 800 ピクセルのパーツを作成することはできません。

**タイプ** 数値

**適用可能なパーツ**

|           |               |                    |
|-----------|---------------|--------------------|
| ActiveX   | カレンダー         | キャンバス              |
| チェック・ボックス | 組み合わせボックス     | コンテナ               |
| 入力フィールド   | グラフ           | グラフィック・プッシュボタン     |
| グループ・ボックス | 水平方向のスクロール・バー | イメージ               |
| Java Bean | リスト・ボックス      | メディア・パネル           |
| 複数行編集     | ノートブック        | ODBC/JDBC インターフェース |
| 外枠        | 進行状況バー        | プッシュボタン            |
| ラジオ・ボタン   | スライダー         | スピン・ボタン            |
| 静的テキスト    | サブファイル        | 縦方向のスクロール・バー       |
| ウィンドウ     |               |                    |

**命令コード**

|         |          |
|---------|----------|
|         | 実行時に操作可能 |
| SETATR* | はい       |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

パーツの幅 (ピクセル数)

## フリー・フォームの例

パーツの幅を大きくする:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+D+HiLoEq---
*
C          EVAL      width = %getatr('win01': 'EF1': 'Width')
*
C          EVAL      %setatr('win01': 'EF1': 'Width') =
                    width + 100
*
```

## 固定形式の例

プッシュボタンを 200 ピクセルだけ広げる:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+D+HiLoEq---
*
C  'EF1'      GETATR  'WIDTH'      OldWidth      2 0
C  OldTop     ADD      200          NewWidth      2 0
C  'EF1'      SETATR  NewWidth     'WIDTH'
*
```

## WindowMode

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

ウィンドウを表示する方法を決定します。

**タイプ** 数値

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

**使用可能な値**

**WindowMode** は、次のいずれかとすることができます。

- 1 最大化** - ウィンドウを可能なその最大サイズで表示します
- 2 最小化** - ワークスペースからウィンドウを除去して、ワークスペースのアイコンとして表示します
- 3 復元** - ウィンドウを最小化または最大化した前のサイズに戻します

**注:** ウィンドウが最大化状態から復元されると、**WindowMode** 属性は、ウィンドウがまだ最大化された状態 (そうでなくても) にあることを示す場合があります。

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'win01': 'WindowMode') = 1
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C  'WIN01'      SETATR  1          'WINDOWMODE'
```

## WordWrap

複数編集パーツに入力されたテキストを語の境界で折り返す必要があるかどうかを決定します。 **WordWrap** が使用不能の場合には、テキストは同じ行に続けられます。

**注:** 横のスクロール・スタイルが設定されている場合には、この属性は機能しません。

**タイプ** 数値

**適用可能なパーツ**

|       |  |  |
|-------|--|--|
| 複数行編集 |  |  |
|-------|--|--|

**命令コード**

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | いいえ      |
| GETATR  | はい       |

**\*注:** GUI Designer でプロパティ・ノートブックを使用して設定することができます。

**使用可能な値**

**WordWrap** は、次のいずれかとすることができます。

**0**      ワード・ラップは使用不能です

**1**      ワード・ラップが使用可能です

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C                    EVAL      rc = %getatr('win01': 'MLE1': 'WordWrap')
*
```

**固定形式の例**

複数行編集の wordwrap 状態の切り替え:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C    'MLE1'            GETATR    'WORDWRAP'    break            1 0
*
```



## WrkStnName

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

ワークステーションの名前を戻します。

**タイプ** スtring

**適用可能なパーツ**

|            |  |  |
|------------|--|--|
| *Component |  |  |
|------------|--|--|

### 命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | いいえ      |
| GETATR | はい       |

### フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      WRKN=%getatr('*Component':'*Component':'WrkStnName')
*
```

### 固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C      '*Component' GETATR      'WRKSTNNAME' WRKN
*
```

# X

**\*制約事項:** Java アプリケーションにのみ適用されます。

呼び出し側の座標空間内のポップアップ・メニューの x 座標 (ピクセル数) を指定します。この属性は、**InvName**、**InvPname**、および **Y** 属性と一緒に使用されます。

**タイプ** 数値

**適用可能なパーツ**

|             |  |  |
|-------------|--|--|
| ポップアップ・メニュー |  |  |
|-------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01':'POP1':'X')=35
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq----
*
C  'POP1'      SETATR  35          'X'
*
```

## XAxisLabel

X 軸のラベルとして使用されるテキスト。

タイプ スtring

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'GRA1': 'XAxisLabel') = label
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'GRA'      SETATR  'label'      'XAxisLabel'
*
```

# Y

**\*制約事項:** Java アプリケーションにのみ適用されます。

呼び出し側の座標空間内のポップアップ・メニューの y 座標 (ピクセル数) を指定します。この属性は、**InvName**、**InvPname**、および **X** 属性と一緒に使用されます。

**タイプ** 数値

**適用可能なパーツ**

|             |  |  |
|-------------|--|--|
| ポップアップ・メニュー |  |  |
|-------------|--|--|

**命令コード**

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | いいえ      |

**フリー・フォームの例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01':'POP1':'Y')=45
*
```

**固定形式の例**

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C  'POP1'      SETATR  45          'Y'
*
```

## YAxisLabel

Y 軸のラベルとして使用されるテキスト。

タイプ スtring

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'GRA1': 'YAxisLabel') = label
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C 'GRA1'      SETATR  'label'      'YAxisLabel'
*
```

## Year

カレンダーの現在選択されている年を示します。これが設定されると、カレンダー・パーツの選択された日付が、変更を反映するように更新されます。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

年

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'Year') = 1998
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul+++++Len++D+HiLoEq---
*
C 'CAL1'      SETATR  1998      'YEAR'
*
```

## YearIdx

現在のカレンダー・ページに表示されない日付に関するデータをユーザーが設定できるようにします。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

使用可能な値

索引値

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'YearIdx') = 1998
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq---
*
C  'CAL1'      SETATR  1998      'YEARIDX'
*
```

## YearLen

年を表示するための形式を示します。

タイプ 数値

適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

命令コード

|         | 実行時に操作可能 |
|---------|----------|
| SETATR* | はい       |
| GETATR  | はい       |

\*注: GUI Designer でプロパティ・ノートブックを使用して設定することができます。

使用可能な値

**YearLen** は、次のいずれかとすることができます。

- 1 全 4 桁の年
- 2 2 個の数字

注: Windows は、2 桁埋め込みオプションとして 2 桁の数字を表示します。区別は行われません。

- 3 1 個の数字しかない年の場合にゼロが埋め込まれる 2 個の数字

フリー・フォームの例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C          EVAL      %setatr('win01': 'CAL1': 'YearLen') = 3
*
```

固定形式の例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq---
*
C  'CAL1'      GETATR  'YEARLEN'      YearLen          1 0
*
```



## Yinc

Y 軸で使用する均等間隔のティック・マークの数を設定します。Y 軸は、許可される最小値と最大値に基づいて、等しい増分数に分けられます。

タイプ 数値

適用可能なパーツ

|     |  |  |
|-----|--|--|
| グラフ |  |  |
|-----|--|--|

命令コード

|        | 実行時に操作可能 |
|--------|----------|
| SETATR | はい       |
| GETATR | はい       |

フリー・フォームの例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C          EVAL      %setatr('win01': 'GRA1': 'YInc') = 20
*
```

固定形式の例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq----
*
C  'GRA1'      SETATR  20          'YInc'
*
```



---

## 第 3 章 パーツ・イベント

パーツがイベント (たとえば、ユーザーがプッシュボタンを選択した場合) を認識すると、ユーザー・プログラムの特定の BEGACT および ENDACT 命令コード内で定義したアクション・サブルーチンが呼び出されます。イベントにアクション・サブルーチンが定義されていない場合には、そのイベントは無視されます。アクション・サブルーチン・ウィンドウには、どのパーツ・イベントでどのアクション・サブルーチンが呼び出されるかが表示されます。

場合によっては、イベントも関連データが入っている **event** 属性を使用できることがあります。これらのイベント属性は、ユーザー・プログラムでそれらを参照してアクセスします。

この章ではパーツ・イベントについて説明します。それぞれの説明には以下が含まれています。

- イベントが適用されるパーツのリスト
- イベントの一般的な説明
- イベントに適用されるイベント属性のリスト (適用可能な場合)
- 例 (あれば)

---

### Activate

ウィンドウがアクティブ・ウィンドウになった時に、このイベントが発生します。ユーザーがマウス・ポインターでウィンドウをクリックするか、ウィンドウのパーツを選択すると、ウィンドウはアクティブになります。

#### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

#### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

### BeanEvent

\* **制約事項:** このイベントは Windows アプリケーションではサポートされていません。

このイベントは、Java Bean イベントが発生するたびに発生します。

#### 適用可能なパーツ

|           |  |  |
|-----------|--|--|
| Java Bean |  |  |
|-----------|--|--|

## イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## Change

パーツの内容または値が変更された時に、このイベントが起こります。それぞれのパーツ・タイプで、このイベントは次の変更が行われた時に現れます。

### 組み合わせボックス

ユーザーが、入力フィールド域に入力するか、ドロップダウン・リストから項目を選択した時。

### 入力フィールド

ユーザーが、入力フィールドに入力した時。

### 複数行編集

ユーザーが、複数行編集パーツに入力した時。

### メディア・パネル

ユーザーが、パネルのスライダー位置を変更した時。

### スライダー

ユーザーが、スライダー位置を変更した時。

### スピン・ボタン

ユーザーはスピン・ボタンの 1 つを選択するか、あるいは入力フィールドに値を入力します。

### サブファイル

ユーザーはサブファイル・セルのデータを変更してそのセルを終了します。

特定の属性を設定した結果として、パーツの値も変更できることに注意してください。たとえば、パーツはターゲット・パーツとして設定され、**Link** イベントに応答します。

### 適用可能なパーツ

|                 |         |          |
|-----------------|---------|----------|
| 組み合わせボックス       | 入力フィールド | メディア・パネル |
| 複数行編集           | スライダー   | スピン・ボタン  |
| 48 ページの『サブファイル』 |         |          |

## イベント属性

|            |            |        |
|------------|------------|--------|
| %ColNumber | %EventName | %Index |
| %Part      | %Window    |        |

---

## Click

ユーザーがパーツ上にマウス・ポインターを合わせ、マウス・ボタンの 1 つを押して離れた時に、このイベントが発生します。

このイベントを受け取るためには、このパーツが使用可能になっていなければなりません。

**注:** コンテナ・パーツが詳細ビューの場合には、Click イベントは通知されません。

#### 適用可能なパーツ

|         |        |      |
|---------|--------|------|
| カレンダー   | キャンバス  | コンテナ |
| 入力フィールド | グラフ    | イメージ |
| 複数行編集   | 静的テキスト |      |

#### イベント属性

|         |            |         |
|---------|------------|---------|
| %Button | %EventName | %MouseX |
| %MouseY | %Part      | %Window |

---

## Close

このイベントが現れるのは、システム・メニューを使用してウィンドウがクローズされる時だけです。CLSWIN 命令コードには影響を与えません。

#### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

#### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## Collapsed

このイベントは、ユーザーがツリー・ビューでコンテナ内のノードを縮小した時に発生します。このイベントを使用して、レコードの状態を表示するようにレコードのアイコンを変更することができます。

#### 適用可能なパーツ

|      |  |  |
|------|--|--|
| コンテナ |  |  |
|------|--|--|

#### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## ColSelect

このイベントは、ユーザーがクリックによって列見出しを選択した時に発生します。コンテナ・パーツは詳細ビュー内になければなりません。%ColNumber イベント属性は、選択された列の列番号を戻します。

### 適用可能なパーツ

|      |        |  |
|------|--------|--|
| コンテナ | サブファイル |  |
|------|--------|--|

### イベント属性

|            |            |       |
|------------|------------|-------|
| %ColNumber | %EventName | %Part |
| %Window    |            |       |

---

## OpenEdit

このイベントは、編集のためにサブファイル・セルがオープンされる直前に発生します。 %index イベント属性は、セルの索引番号を戻します。 %ColNumber イベント属性は、セルの列番号を戻します。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |            |        |
|------------|------------|--------|
| %ColNumber | %EventName | %Index |
| %Part      | %Window    |        |

---

## Complete

このイベントは、メディア・パーツがオーディオ・ファイルを処理した時に発生します。 %Complete イベント属性は、操作が正常に行われたかどうかを示します。

### 適用可能なパーツ

|      |  |  |
|------|--|--|
| メディア |  |  |
|------|--|--|

### イベント属性

|           |            |       |
|-----------|------------|-------|
| %Complete | %EventName | %Part |
| %Window   |            |       |

---

## Create

このイベントは、パーツが初めて作成された時に起こります。

ウィンドウ・パーツの場合には、**Create** イベントが行われるのはそのウィンドウのすべてのパーツが作成され、それらの **Create** イベントが行われた後になります。

この時点にパーツの属性を設定することが可能です。たとえば、入力フィールドがウィンドウに表示される前にその入力フィールドのデフォルト値を変更したい場合があります。

## 適用可能なパーツ

|              |                    |                |
|--------------|--------------------|----------------|
| ActiveX      | アニメーション制御          | カレンダー          |
| キャンバス        | チェック・ボックス          | 組み合わせボックス      |
| コンポーネント参照    | コンテナー              | DDE クライアント     |
| 入力フィールド      | グラフ                | グラフィック・プッシュボタン |
| グループ・ボックス    | 水平方向のスクロール・バー      | イメージ           |
| Java Bean    | リスト・ボックス           | メディア           |
| メディア・パネル     | メニュー・バー            | メニュー項目         |
| メッセージ・サブファイル | 複数行編集              | ノートブック         |
| ノートブック・ページ   | ODBC/JDBC インターフェース | 外枠             |
| 進行状況バー       | プッシュボタン            | ラジオ・ボタン        |
| スライダー        | スピン・ボタン            | 静的テキスト         |
| 状況バー         | サブファイル             | サブメニュー         |
| タイマー         | 縦方向のスクロール・バー       | ウィンドウ          |

## イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## Data

DDE 会話でサーバー・アプリケーションからデータが使用可能になった時に、このイベントが起こります。これは *Request* コマンドの結果として生成されるか、あるいはデータが変更されたかまたは DDE クライアント・パーツが **AllowLink** 属性を介して別のパーツとホット・リンク会話で設定された場合に生成されます。

## 適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

## イベント属性

|       |            |       |
|-------|------------|-------|
| %Data | %EventName | %Item |
| %Part | %Window    |       |

---

## DbIClick

ユーザーが連続して 2 回、マウス・ボタンを押して離すと、このイベントが発生します。

## 適用可能なパーツ

|         |       |       |
|---------|-------|-------|
| カレンダー   | キャンバス | コンテナー |
| 入力フィールド | グラフ   | イメージ  |

|       |        |  |
|-------|--------|--|
| 複数行編集 | 静的テキスト |  |
|-------|--------|--|

### イベント属性

|         |            |         |
|---------|------------|---------|
| %Button | %EventName | %MouseX |
| %MouseY | %Part      | %Window |

---

## DeActivate

ウィンドウがアクティブでない時に、このイベントが発生します。別のパーツがアクティブになった時に、ウィンドウが非アクティブとなります。

このイベントは、CLSWIN 命令コードが実行された時に起こります。

### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## Destroy

このイベントは、パーツが入っているウィンドウが、ウィンドウのシステム・メニューからクローズ・メニュー・オプションを選択するか、あるいは CLSWIN 命令コードを使用してクローズされた時に行われます。

このイベントはまた、プログラムが終了中である時にも、プログラムの各パーツごとに起こります。

### 適用可能なパーツ

|              |                    |                |
|--------------|--------------------|----------------|
| ActiveX      | アニメーション制御          | カレンダー          |
| キャンバス        | チェック・ボックス          | 組み合わせボックス      |
| コンポーネント参照    | コンテナ               | DDE クライアント     |
| 入力フィールド      | グラフ                | グラフィック・プッシュボタン |
| グループ・ボックス    | 水平方向のスクロール・バー      | イメージ           |
| Java Bean    | リスト・ボックス           | メディア           |
| メディア・パネル     | メニュー・バー            | メニュー項目         |
| メッセージ・サブファイル | 複数行編集              | ノートブック         |
| ノートブック・ページ   | ODBC/JDBC インターフェース | 外枠             |
| 進行状況バー       | プッシュボタン            | ラジオ・ボタン        |
| スライダー        | スピン・ボタン            | 静的テキスト         |
| 状況バー         | サブファイル             | サブメニュー         |



|      |              |       |
|------|--------------|-------|
| タイマー | 縦方向のスクロール・バー | ウィンドウ |
|------|--------------|-------|

#### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

## Drop

オブジェクトがドラッグされ、このパーツにドロップされた時に、このイベントが起こります。このイベントに応答するためには、このパーツに対して **DropEnable** 属性が設定されていなければなりません。**%Data** イベント属性をチェックして、ドロップされたパーツに関連した値を照会することができます。

#### 適用可能なパーツ

|              |         |           |
|--------------|---------|-----------|
| 組み合わせボックス*   | 入力フィールド | リスト・ボックス* |
| メッセージ・サブファイル | 複数行編集   | 静的テキスト    |

**\*制約事項:** このイベントは、このパーツに対する Java アプリケーションでは実行時にサポートされません。

#### イベント属性

|         |            |       |
|---------|------------|-------|
| %Data   | %EventName | %Part |
| %Window |            |       |

## DropDown

**\*制約事項:** このイベントは Java アプリケーションではサポートされていません。

組み合わせボックスについて次のことが行われると、このイベントが発生します。

- ユーザーがボックスの右側にある下矢印をクリックする。

#### 適用可能なパーツ

|           |  |  |
|-----------|--|--|
| 組み合わせボックス |  |  |
|-----------|--|--|

#### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

## Enter

組み合わせボックス、コンテナー、リスト・ボックス、メッセージ・サブファイル、またはサブファイルの場合には、このイベントは次のいずれかが起こった時に行われます。

- ユーザーがパーツをダブルクリックした時。

- パーツにフォーカスがあり、項目が選択され、ユーザーが Enter キーを押した時。

ラジオ・ボタンの場合には、このイベントはパーツをダブルクリックした時に起こります。

#### 適用可能なパーツ

|              |         |           |
|--------------|---------|-----------|
| 組み合わせボックス    | コンテナー   | リスト・ボックス* |
| メッセージ・サブファイル | ラジオ・ボタン | サブファイル    |

**\*制約事項:** このイベントは、このパーツに対する Java アプリケーションでは実行時にサポートされません。

#### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

## ExecuteAck

このイベントは、サーバー・アプリケーションが execute コマンドを処理した時に起こります。**%Complete** イベント属性は、実行が正常であったかどうかを示します。

#### 適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

#### イベント属性

|           |         |            |
|-----------|---------|------------|
| %Complete | %Data   | %EventName |
| %Part     | %Window |            |

## Expanded

このイベントは、ユーザーがツリー・ビューでコンテナー内のノードを拡大した時に発生します。このイベントを使用して、レコードの状態を表示するようにレコードのアイコンを変更することができます。

#### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| コンテナー |  |  |
|-------|--|--|

#### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## FirstRec

このイベントは、ナビゲーター・バーで「最初のレコード」ボタンが選択された時に行われます。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## GotFocus

このイベントは、パーツが入力フォーカスを受け取った時に起こります。これは、ユーザーがカーソルを使用可能パーツに移動したか、またはフォーカスが **Focus** 属性を使用してこのパーツに設定された場合に行われます。このイベントを受け取るためには、そのパーツが使用可能になっていなければならないことに注意してください。

### 適用可能なパーツ

|                |          |         |
|----------------|----------|---------|
| 組み合わせボックス      | コンテナ     | 入力フィールド |
| グラフィック・プッシュボタン | リスト・ボックス | 複数行編集   |
| プッシュボタン        | スライダー    | スピン・ボタン |
| サブファイル         |          |         |

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## KeyPress

このイベントは、キーが押され、パーツに入力フォーカスが当てられている時に起こります。このイベントを受け取るためには、このパーツが使用可能になっていなければなりません。

**注:** **ReadOnly** 属性が設定されてはいけません。

このイベントは文字キーについてだけ発生します。非文字キー (前ページや次ページなどの) によって生成されたイベントを処理する場合には、**VKeyPress** イベントを参照してください。

### 適用可能なパーツ

|           |       |         |
|-----------|-------|---------|
| 組み合わせボックス | コンテナ  | 入力フィールド |
| リスト・ボックス  | 複数行編集 | サブファイル  |

## イベント属性

|            |            |          |
|------------|------------|----------|
| %Alt       | %Character | %Control |
| %EventName | %Part      | %Shift   |
| %Window    |            |          |

---

## LastRec

このイベントは、ナビゲーター・バーで「最後のレコード」ボタンが押された時に行われます。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

## イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## LClickTray

このイベントは、ユーザーがアプリケーションのシステム・トレイ・アイコンを左クリックした時に起こります。このアプリケーションは、Windows の Shell\_NotifyIconA API を使用してシステム・トレイ・アプリケーションとして登録されていなければなりません。

### 適用可能なパーツ

|       |              |  |
|-------|--------------|--|
| ウィンドウ | キャンバス付きウィンドウ |  |
|-------|--------------|--|

## イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## Link

**\*制約事項:** このイベントは Java アプリケーションではサポートされていません。

このイベントは、パーツが **AddLink** 属性を使用してソース・パーツにリンクされて、ソース・パーツが変更された場合に行われます。ソース・パーツがこのイベントを通知するためには、**AllowLink** 属性が設定されていなければなりません。

### 適用可能なパーツ

|          |       |         |
|----------|-------|---------|
| 入力フィールド  | イメージ  | メディア    |
| メディア・パネル | スライダー | スピン・ボタン |
| 静的テキスト   | タイマー  |         |

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## LostFocus

このイベントは、パーツに入力フォーカスが当てられなくなった時に起こります。

### 適用可能なパーツ

|                |          |         |
|----------------|----------|---------|
| 組み合わせボックス      | コンテナー    | 入力フィールド |
| グラフィック・プッシュボタン | リスト・ボックス | 複数行編集   |
| プッシュボタン        | スライダー    | スピン・ボタン |
| サブファイル         |          |         |

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## MenuSelect

マウスでクリックすることによって、またはアクセラレーター・キーを使用することによって、ユーザーがメニュー項目を選択すると、このイベントが発生します。

このイベントは、メニュー項目がチェックされたことを意味しないことに注意してください。メニュー項目がチェックされているかどうかを調べるためには、その項目の **Checked** 属性を照会してください。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| メニュー項目 |  |  |
|--------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## MouseDown

マウス・ボタンが押された時に、このイベントが起こります。

### 適用可能なパーツ

|         |       |       |
|---------|-------|-------|
| カレンダー   | キャンバス | コンテナー |
| 入力フィールド | グラフ   | 複数行編集 |
| 静的テキスト  |       |       |

## イベント属性

|         |            |         |
|---------|------------|---------|
| %Button | %EventName | %MouseX |
| %MouseY | %Part      | %Window |

---

## MouseEnter

このイベントは、マウス・ポインターが初めてパーツに移動した時に発生します。

### 適用可能なパーツ

|                |              |           |
|----------------|--------------|-----------|
| カレンダー          | チェック・ボックス    | 組み合わせボックス |
| コンテナ           | 入力フィールド      | グラフ       |
| グラフィック・プッシュボタン | イメージ         | リスト・ボックス  |
| メディア・パネル       | メッセージ・サブファイル | 複数行編集     |
| プッシュボタン        | ラジオ・ボタン      | スライダー     |
| スピン・ボタン        | 静的テキスト       | サブファイル    |

## イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## MouseExit

このイベントは、マウス・ポインターが初めてパーツから移動した時に発生します。

### 適用可能なパーツ

|                |              |           |
|----------------|--------------|-----------|
| カレンダー          | チェック・ボックス    | 組み合わせボックス |
| コンテナ           | 入力フィールド      | グラフ       |
| グラフィック・プッシュボタン | イメージ         | リスト・ボックス  |
| メディア・パネル       | メッセージ・サブファイル | 複数行編集     |
| プッシュボタン        | ラジオ・ボタン      | スライダー     |
| スピン・ボタン        | 静的テキスト       | サブファイル    |

## イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## MouseMove

マウス・ポインターがパーツの上を移動した時に、このイベントが起こります。パーツの機能をユーザーに通知する情報域を作成するために、このイベントを使用することができます。

### 適用可能なパーツ

|           |                |              |
|-----------|----------------|--------------|
| カレンダー     | キャンバス          | チェック・ボックス    |
| 組み合わせボックス | コンテナ           | 入力フィールド      |
| グラフ       | グラフィック・プッシュボタン | イメージ         |
| リスト・ボックス  | メディア・パネル       | メッセージ・サブファイル |
| 複数行編集     | プッシュボタン        | ラジオ・ボタン      |
| スライダー     | スピン・ボタン        | 静的テキスト       |
| サブファイル    |                |              |

### イベント属性

|            |         |         |
|------------|---------|---------|
| %EventName | %MouseX | %MouseY |
| %Part      | %Window |         |

### 例

マウス・ポインターがパーツ上を移動した時に静的テキスト・パーツの **Label** 属性を変更して、情報域を作成します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
C    PB1          BEGACT   MOUSEMOVE
*
C    ST1          SETATR   'Ends program' LABEL
*
C                ENDACT
*
```

---

## MouseUp

マウス・ボタンが解放された時に、このイベントが起こります。

### 適用可能なパーツ

|         |       |       |
|---------|-------|-------|
| カレンダー   | キャンバス | コンテナ  |
| 入力フィールド | グラフ   | 複数行編集 |
| 静的テキスト  |       |       |

### イベント属性

|         |            |         |
|---------|------------|---------|
| %Button | %EventName | %MouseX |
| %MouseY | %Part      | %Window |

---

## Moved

このイベントは、ユーザーがウィンドウを移動するたびに発生します。

### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## MthChange

このイベントは、ユーザーが月矢印の 1 つをクリックすることによって新しい月を選択した時に発生します。

### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## NextRec

このイベントは、ナビゲーター・バーで「次のレコード」ボタンが選択された時に行われます。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## Notify

コンポーネント参照パーツのプロパティ・ノートブックで、指定したウィンドウの特定のパーツに関連したイベントを指定することができます。指定したウィンドウの指定したパーツでその名前が **NotSrcEvt** 属性の設定値と一致するイベントが現れると、コンポーネント参照パーツに **Notify** イベントが送られます。

シグナルの送られる Notify イベントの場合には、モニター中のイベント用のアクション・サブルーチンが存在していなければなりません。このサブルーチンはコードを含んでいる必要はありません。アクション・サブルーチンがなければ Notify イベントは現れません。

### 適用可能なパーツ

|           |  |  |
|-----------|--|--|
| コンポーネント参照 |  |  |
|-----------|--|--|



## イベント属性

|             |              |              |
|-------------|--------------|--------------|
| %EventName  | %Part        | %SrcCompName |
| %SrcEvtName | %SrcPartName | %SrcWinName  |
| %Window     |              |              |

---

## OCXEvent

\* **制約事項:** このイベントは Java アプリケーションではサポートされていません。

このイベントは、ActiveX 制御イベントが発生するたびに発生します。

### 適用可能なパーツ

|         |  |  |
|---------|--|--|
| ActiveX |  |  |
|---------|--|--|

## イベント属性

|            |       |           |
|------------|-------|-----------|
| %EventName | %Part | %RealName |
| %Window    |       |           |

---

## PageDown

このイベントは、ユーザーが PageDown キーを押した時、縦方向のスクロール・バーの下の空の領域をクリックした時、あるいは PageDown ボタン・ナビゲーション・バーを選択した時に発生します。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

## イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## PageEnd

このイベントは、ユーザーが PageDown キーまたは下矢印キーを押した時、あるいはサブファイルの最終レコードが表示されている時にユーザーが縦方向のスクロール・バーの下の空の領域をクリックするかまたはスクロール・バーのスライダーをサブファイル・リストの最下部までドラッグした場合に、発生します。この場合には、リストの最下部がすでに表示されているので、PageDown イベントは送られません。代わりに PageEnd イベントが送られます。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## PageSelect

このイベントが起こるのは、ユーザーがタブを選択することによって、新しいノートブック・セクション内のノートブック・ページを選択する時です。

**%Page** イベント属性は、選択したページのページ番号を戻します。

### 適用可能なパーツ

|            |  |  |
|------------|--|--|
| ノートブック・ページ |  |  |
|------------|--|--|

### イベント属性

|            |       |       |
|------------|-------|-------|
| %EventName | %Page | %Part |
| %Window    |       |       |

---

## PageTop

このイベントは、ユーザーが PageUp キーまたは上矢印キーを押した時、あるいはユーザーが縦方向のスクロール・バーの上の空の領域をクリックするかまたはスクロール・バーのスライダーをサブファイル・リストの最上部までドラッグした場合に、発生します。この場合には、**PageUp** イベントは送られません。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## PageUp

このイベントは、ユーザーが PageUp キーを押した時、あるいはユーザーが縦方向のスクロール・バーの上の空の領域をクリックするかまたは PageUp ナビゲーション・ボタンを選択した場合に、発生します。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

## PokeAck

DDE 会話で、サーバー・アプリケーションが poke コマンドを処理した時に、このイベントが起こります。poke が正常に実行されたかどうかは、**%Complete** イベントが示します。

### 適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

### イベント属性

|           |       |            |
|-----------|-------|------------|
| %Complete | %Data | %EventName |
| %Item     | %Part | %Window    |

### 例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
C   DDE1           BEGACT   POKEACK
*
C   %COMPLETE     IFNE     *ZERO
C   'Poke failed' DSPLY    style      X
*
C                   ENDIF
*
C                   ENDACT
*
```

## Popup

マウス・ポインターがパーツ上に位置付けされ、右マウス・ボタンが押された時に、このイベントが発生します。

### 適用可能なパーツ

|          |           |                |
|----------|-----------|----------------|
| キャンバス    | チェック・ボックス | 組み合わせボックス      |
| コンテナ     | 入力フィールド   | グラフィック・プッシュボタン |
| リスト・ボックス | メディア・パネル  | メッセージ・サブファイル   |
| 複数行編集    | プッシュボタン   | ラジオ・ボタン        |
| スライダー    | スピン・ボタン   | 静的テキスト         |
| サブファイル   |           |                |

### イベント属性

|            |            |        |
|------------|------------|--------|
| %ColNumber | %EventName | %Index |
| %MouseX    | %MouseY    | %Part  |
| %Window    |            |        |

---

## Press

このイベントは、ユーザーがマウスでそのパーツをクリックした場合、またはそのパーツに入力フォーカスがある時に Enter キーを押した場合に起こります。

### 適用可能なパーツ

|                |          |         |
|----------------|----------|---------|
| グラフィック・プッシュボタン | メディア・パネル | プッシュボタン |
|----------------|----------|---------|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## PrevRec

このイベントは、ナビゲーター・パネルで「前のレコード」ボタンが選択された時に行われます。

### 適用可能なパーツ

|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## RClickTray

このイベントは、ユーザーがアプリケーションのシステム・トレイ・アイコンを右クリックした時に起こります。このアプリケーションは、Windows の Shell\_NotifyIconA API を使用してシステム・トレイ・アプリケーションとして登録されていなければなりません。

### 適用可能なパーツ

|       |              |  |
|-------|--------------|--|
| ウィンドウ | キャンバス付きウィンドウ |  |
|-------|--------------|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## ReSize

このイベントは、パーツのサイズが変更された時に起こります。たとえば、ユーザーがサイズ変更枠を使用してウィンドウのサイズを変更したか、あるいは **Height** または **Width** 属性を設定してパーツのサイズを変更した時に現れます。

ユーザーがウィンドウをサイズ変更した後で、ウィンドウ上でパーツを再配置または移動するために、このイベントを使用することができます。

**ReSize** イベントの実行中には、**Height** および **Width** 属性にはウィンドウの現在のサイズが入っていて、**%NewHeight** および **%NewWidth** イベント属性には **ReSize** イベントが完了した時にウィンドウのサイズが入っています。

#### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

#### イベント属性

|            |            |           |
|------------|------------|-----------|
| %EventName | %NewHeight | %NewMode* |
| %NewWidth  | %Part      | %Window   |

\*注: 制約事項については、属性の説明を参照してください。

## Scroll

このイベントは、ユーザーが横または縦方向のスクロール・バー・パーツのスクロール矢印を押した時に発生します。**%Position** イベント属性には、ユーザーがスクロールを停止した時のスクロール・ボックスの位置が入っています。

#### 適用可能なパーツ

|               |              |  |
|---------------|--------------|--|
| 水平方向のスクロール・バー | 縦方向のスクロール・バー |  |
|---------------|--------------|--|

#### イベント属性

|            |       |           |
|------------|-------|-----------|
| %EventName | %Part | %Position |
| %Window    |       |           |

## Select

マウスでパーツ上をクリックすることにより、または矢印キーを使用することにより、ユーザーがパーツを選択した時に、このイベントが起こります。組み合わせボックス、リスト・ボックス、メッセージ・サブファイル、またはサブファイルについては、リストの項目が選択された時に、このイベントが発生します。

#### 適用可能なパーツ

|           |              |         |
|-----------|--------------|---------|
| チェック・ボックス | 組み合わせボックス    | コンテナー   |
| リスト・ボックス  | メッセージ・サブファイル | ラジオ・ボタン |
| サブファイル    |              |         |

#### イベント属性

|            |            |        |
|------------|------------|--------|
| %ColNumber | %EventName | %Index |
| %Part      | %Window    |        |

---

## SelfPending

\* **制約事項:** このイベントは Windows アプリケーションでしかサポートされていません。

このイベントが起こるのは、ユーザーがタブを選択することによって新規ノートブック・セクション内のノートブック・ページを選択し、そのページが表示されるまでの間です。

### 適用可能なパーツ

|            |  |  |
|------------|--|--|
| ノートブック・ページ |  |  |
|------------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## ShutDown

このイベントは、GUI ビルダーが終了し、NotifyOnClose プラグイン・コマンドにウィンドウ・ハンドルが指定されていた時に発生します。このイベントを使用して、GUI ビルダーから実行しているプラグインをモニターし、それをクローズさせることもできます。

### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| ウィンドウ |  |  |
|-------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## SpinDown

このイベントは、ユーザーがスピン・ボタン・パーツで下矢印ボタンを選択した時に発生します。

### 適用可能なパーツ

|         |  |  |
|---------|--|--|
| スピン・ボタン |  |  |
|---------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## SpinEnd

このイベントは、ユーザーがスピン・ボタン・パーツで上矢印または下矢印ボタンを解放した時に発生します。

### 適用可能なパーツ

|         |  |  |
|---------|--|--|
| スピン・ボタン |  |  |
|---------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## SpinUp

このイベントは、ユーザーがスピン・ボタン・パーツで上矢印ボタンを選択した時に発生します。

### 適用可能なパーツ

|         |  |  |
|---------|--|--|
| スピン・ボタン |  |  |
|---------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

---

## Terminate

このイベントは、DDE 会話のサーバーが会話を終了するように要求するか、または DDE クライアント・パーツがその **DDEMode** 属性を *Terminate* に設定した時に現れます。

### 適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|

### 例

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
CSRNO1Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
C   PB1           BEGACT   TERMINATE
*
C   'Terminating' DSPLY   STYLE     X
*
C               ENDACT
*
```

---

## Tick

このイベントはタイマーの間隔値が経過するたびに起こります。 **Multiplier** 属性も、 **Tick** イベントが生成される時点を決めます。

### 適用可能なパーツ

|      |  |  |
|------|--|--|
| タイマー |  |  |
|------|--|--|

### イベント属性

| %EventName | %Part | %Window |
|------------|-------|---------|
|------------|-------|---------|

例 静的テキスト・パーツをタイマー・パーツの現在の目盛り値で更新します。

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
C   TIMER1      BEGACT   TICK
*
C   'TIMER1'    GETATR   TIMERTICKS   timerticks   6 0
C   ST1         SETATR   timerticks   'LABEL'
*
C                   ENDACT
*
```

---

## Timeout

このイベントは、**TimeOut** 属性で定義された時間内に DDE 会話が確立されなかった時に現れます。

### 適用可能なパーツ

|            |  |  |
|------------|--|--|
| DDE クライアント |  |  |
|------------|--|--|

### イベント属性

| %EventName | %Part | %Window |
|------------|-------|---------|
|------------|-------|---------|

例

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+
CSRN01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
C   DDE1        BEGACT   TIMEOUT
*
C   'DDE Failed!' DSPLY   STYLE   X
C                   ENDACT
*
```

---

## VKeyPress

このイベントはコマンド・キーが押された時に発生します。コマンド・キーの例は、前ページ( Page Up )、Pause、および Ctrl です。戻される値については、**%Character** イベント属性を参照してください。このイベントをキャンバス・パーツに関連付けた場合には、このイベントに応答できるすべてのパーツに 1 つのアクション・サブルーチンをコーディングするだけで済みます。

### 適用可能なパーツ

|         |           |       |
|---------|-----------|-------|
| キャンバス   | 組み合わせボックス | コンテナ  |
| 入力フィールド | リスト・ボックス  | 複数行編集 |



|        |  |  |
|--------|--|--|
| サブファイル |  |  |
|--------|--|--|

### イベント属性

|            |            |          |
|------------|------------|----------|
| %Alt       | %Character | %Control |
| %EventName | %Part      | %Shift   |
| %Window    |            |          |

### 例

入力フィールドに入力フォーカスがある時に F5 が押されると、入力フィールドを消去します。

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+
CSRNO1Factor1+++++Opcod(E)+Factor2+++++Resul t+++++Len++D+HiLoEq
*
C   EF1           BEGACT   VKEYPRESS
*
C   %character   IFEQ     '29'
C   'EF1'        SETATR   *BLANKS   'TEXT'
C                   ENDIF
*
C                   ENDACT
*
```

---

## YearChange

このイベントは、ユーザーが年矢印の 1 つをクリックするか、あるいは年が変更されるように月矢印をクリックすることによって新規の年を選択した時に起こります。

### 適用可能なパーツ

|       |  |  |
|-------|--|--|
| カレンダー |  |  |
|-------|--|--|

### イベント属性

|            |       |         |
|------------|-------|---------|
| %EventName | %Part | %Window |
|------------|-------|---------|



## 第 4 章 イベント属性

Event 属性には、処理されるイベントに関連したデータが入ります。これらはユーザーのアプリケーションでは変更できません。これらは定義仕様で正しいタイプと長さが定義されなければなりません。次のテーブルに、event 属性とそれらの対応データ・タイプおよび長さをリストします。

| Event 属性     | データ・タイプ | 長さ             |
|--------------|---------|----------------|
| %Alt         | 数値      | 1              |
| %Button      | 数値      | 1              |
| %Character   | ストリング   | 1 <sup>1</sup> |
| %ColNumber   | 数値      | *              |
| %Complete    | 数値      | 1              |
| %Control     | 数値      | 1              |
| %Data        | ストリング   | *              |
| %EventName   | ストリング   | 10             |
| %Index       | 数値      | *              |
| %Item        | ストリング   | *              |
| %MouseX      | 数値      | 4              |
| %MouseY      | 数値      | 4              |
| %NewHeight   | 数値      | 4              |
| %NewMode     | 数値      | 1              |
| %NewWidth    | 数値      | 4              |
| %Page        | 数値      | 3              |
| %Part        | ストリング   | 10             |
| %Position    | 数値      | 4              |
| %RealName    | ストリング   | 10             |
| %Shift       | 数値      | 1              |
| %SrcCompName | ストリング   | 10             |
| %SrcEvtName  | ストリング   | 10             |
| %SrcPartName | ストリング   | 10             |
| %SrcWinName  | ストリング   | 10             |
| %Window      | ストリング   | 10             |

\* 必要に応じて長さを定義します。

<sup>1</sup>**VKeyPress** の場合には、長さは 2 です。

例:

```
DName+++++++ETDsFrom+++To/L+++IDc.Keywords+++++++
*
* Declare the %Button event attribute
D %Button      S              1P 0
```

```

*
* Declare the %Data event attribute
D %Data          S          32A

```

この章ではイベント属性について説明します。それぞれの説明には以下が含まれています。

- event 属性のデータ・タイプ
- event 属性の一般的な説明
- 属性が適用されるイベントのリスト

## %Alt

イベントが現れた時に Alt キーが押されたかどうかを示します。

**0** Alt キーは押されていません

**1** Alt キーが押されました

タイプ 数値

適用可能なイベント

|          |           |  |
|----------|-----------|--|
| KeyPress | VKeyPress |  |
|----------|-----------|--|

## %Button

イベント中にどちらのマウス・ボタンが押されたかを示します。

**1** ボタン 1

**2** ボタン 2

**3** ボタン 3

**4** ボタン 1 およびボタン 2

タイプ 数値

適用可能なイベント

|         |         |           |
|---------|---------|-----------|
| Click   | DbClick | MouseDown |
| MouseUp |         |           |

## %Character

イベントによって生成される文字コード。

**VKeyPress** イベント中に、非文字キーを押したことを表す値として **%Character** イベント属性が戻されます。値は次の通りです。

|              |              |            |          |
|--------------|--------------|------------|----------|
| 00 - Esc     | 13 - End     | 25 - Break | 37 - F14 |
| 01 - タブ右     | 14 - Page Up | 26 - F2    | 38 - F15 |
| 02 - タブ左     | 15 - 次ページ    | 27 - F3    | 39 - F16 |
| 03 - スペース    | 16 - 左矢印     | 28 - F4    | 40 - F17 |
| 04 - バックスペース | 17 - 右矢印     | 29 - F5    | 41 - F18 |

|                          |                  |          |          |
|--------------------------|------------------|----------|----------|
| 05 - Enter (数字キー<br>パッド) | 18 - 上矢印         | 30 - F6  | 42 - F19 |
| 06 - Enter               | 19 - 下矢印         | 31 - F7  | 43 - F20 |
| 07 - Shift               | 20 - Caps Lock   | 32 - F8  | 44 - F21 |
| 08 - Ctrl                | 21 - Num Lock    | 33 - F9  | 45 - F22 |
| 09 - Alt                 | 22 - Scroll Lock | 34 - F11 | 46 - F23 |
| 10 - Insert              | 23 - Pause       | 35 - F12 | 47 - F24 |
| 11 - Delete              | 24 - SysRq       | 36 - F13 | 56 - Alt |
| 12 - Home                |                  |          |          |

タイプ スtring

適用可能なイベント

|          |           |  |
|----------|-----------|--|
| KeyPress | VKeyPress |  |
|----------|-----------|--|

## %ColNumber

コンテナまたはサブファイル・パーツの列番号を戻します。

タイプ 数値

適用可能なイベント

|        |           |          |
|--------|-----------|----------|
| Change | ColSelect | OpenEdit |
| Popup  | Select    |          |

## %Complete

前のオプションが正常に完了したかどうかを示します。

0 異常

1 正常

タイプ 数値

適用可能なイベント

|          |            |         |
|----------|------------|---------|
| Complete | ExecuteAck | PokeAck |
|----------|------------|---------|

## %Control

イベントが現れた時に Ctrl キーが押されたかどうかを示します。

0 Ctrl キーは押されませんでした。

1 Ctrl キーが押されました。

タイプ 数値

適用可能なイベント

|          |           |  |
|----------|-----------|--|
| KeyPress | VKeyPress |  |
|----------|-----------|--|

---

## %Data

*Request* コマンドに応答してサーバー・アプリケーションが送信したデータが入っています。このデータの内容と構造は、サーバー・アプリケーションによって決定されます。

このイベント属性も **PokeAck** イベントで使用可能です。これには元の poke されたデータが入れます。

**タイプ** スtring

**適用可能なイベント**

| Data    | Drop | ExecuteAck |
|---------|------|------------|
| PokeAck |      |            |

---

## %EventName

イベントの名前を戻します。

**タイプ** String

**適用可能なイベント**

|          |  |  |
|----------|--|--|
| すべてのイベント |  |  |
|----------|--|--|

---

## %Index

**Popup** または **Select** が起こったときに、マウス・ポインターが置かれているレコードの索引番号を戻すか、または **OpenEdit** イベントが起こったときに、サブファイル・セルの索引番号を戻します。レコードをフォーカスする必要はありません。

**タイプ** 数値

**適用可能なイベント**

| Change | OpenEdit | Popup |
|--------|----------|-------|
| Select |          |       |

---

## %Item

DDE 会話で現在呼び出されている項目の名前を含みます。

**タイプ** String

**適用可能なイベント**

|      |         |  |
|------|---------|--|
| Data | PokeAck |  |
|------|---------|--|

---

## %MouseX

イベントが現れた時のパーツ内のマウス・ポインターの X 座標の現在の場所。これはピクセルで表されます。それは、ゼロとこのパーツの幅の間の値となります。ゼロの値は、このパーツの左の端を表します。

**注: MouseMove** イベントの場合には、この座標はデスクトップに関連がありません。

タイプ 数値

適用可能なイベント

|           |         |           |
|-----------|---------|-----------|
| Click     | DbClick | MouseDown |
| MouseMove | MouseUp | Popup     |

---

## %MouseY

イベントが現れた時のパーツ内のマウス・ポインターの Y 座標の現在の場所。これはピクセルで表されます。それは、ゼロとこのパーツの高さの間の値となります。ゼロの値は、このパーツの下の端を表します。

**注: MouseMove** イベントの場合には、この座標はデスクトップに関連がありません。

タイプ 数値

適用可能なイベント

|           |         |           |
|-----------|---------|-----------|
| Click     | DbClick | MouseDown |
| MouseMove | MouseUp | Popup     |

---

## %NewHeight

サイズ変更後のパーツの高さ。

**注:** パーツの **Height** 属性は、**ReSize** イベントのアクション・サブルーチンが完了するまで変更されません。

タイプ 数値

適用可能なイベント

|        |  |  |
|--------|--|--|
| ReSize |  |  |
|--------|--|--|

---

## %NewMode

**\*制約事項:** この属性は Java アプリケーションではサポートされていません。

ウィンドウが次のどのモードに設定されようとしているかを示します。

1 最大化

2 最小化

3 復元

タイプ 数値

適用可能なイベント

|        |  |  |
|--------|--|--|
| ReSize |  |  |
|--------|--|--|

---

## %NewWidth

サイズ変更後のパーツの幅。

注: パーツの **Width** 属性は、**ReSize** イベントのアクション・サブルーチンが完了するまで変更されません。

タイプ 数値

適用可能なイベント

|        |  |  |
|--------|--|--|
| ReSize |  |  |
|--------|--|--|

---

## %Page

**PageSelect** イベントが現れた時に選択されたページのページ番号を戻します。

タイプ 数値

適用可能なイベント

|            |  |  |
|------------|--|--|
| PageSelect |  |  |
|------------|--|--|

---

## %Part

イベントを生成するパーツの名前を含みます。

タイプ スtring

適用可能なイベント

|          |  |  |
|----------|--|--|
| すべてのイベント |  |  |
|----------|--|--|

---

## %Position

ユーザーがスクロールを停止した時に、スクロール・ボックスの現在位置を戻します。

タイプ 数値

適用可能なイベント

|        |  |  |
|--------|--|--|
| Scroll |  |  |
|--------|--|--|



---

## %RealName

発生する ActiveX イベントの実際の名前を戻します。

タイプ スtring

適用可能なイベント

|          |  |  |
|----------|--|--|
| OCXEvent |  |  |
|----------|--|--|

---

## %Shift

イベントが現れた時に Shift キーが押されたかどうかを示します。

0 Shift キーは押されていない。

1 Shift キーが押された。

タイプ 数値

適用可能なイベント

|          |           |  |
|----------|-----------|--|
| KeyPress | VKeyPress |  |
|----------|-----------|--|

---

## %SrcCompName

**Notify** イベントがコンポーネント参照パーツに送られると、このイベント属性はその通知イベントのコンポーネントの名前を戻します。1 つのコンポーネント参照パーツが複数のイベントのモニター用に使用されている場合には、このイベント属性は **%SrcCompName** を **%SrcWinName**、**%SrcPartName**、および **%SrcEvtName** 属性と一緒に使用して、Notify イベントを起動したイベントの正確なソースを判別することができます。

タイプ String

適用可能なイベント

|        |  |  |
|--------|--|--|
| Notify |  |  |
|--------|--|--|

---

## %SrcEvtName

**Notify** イベントがコンポーネント参照パーツに送られると、このイベント属性は、**NotSrcEvt** 属性で設定されたイベントの名前を特定のパーツおよびウィンドウと一緒に戻して、コンポーネント参照パーツに対して **Notify** イベントを起動します。

タイプ String

適用可能なイベント

|        |  |  |
|--------|--|--|
| Notify |  |  |
|--------|--|--|

---

## %SrcPartName

**Notify** イベントがコンポーネント参照パーツに送られると、このイベント属性は **NotSrcPart** 属性で設定されたパーツの名前を特定のイベントおよびウィンドウと一緒に戻して、コンポーネント参照パーツに対して **Notify** イベントを起動します。

タイプ スtring

適用可能なイベント

|        |  |  |
|--------|--|--|
| Notify |  |  |
|--------|--|--|

---

## %SrcWinName

**Notify** イベントがコンポーネント参照パーツに送られると、このイベント属性は **NotSrcWin** 属性で設定されたウィンドウの名前を特定のパーツおよびイベントと一緒に戻して、コンポーネント参照パーツに対して **Notify** イベントを起動します。

タイプ スtring

適用可能なイベント

|        |  |  |
|--------|--|--|
| Notify |  |  |
|--------|--|--|

---

## %Window

イベントを生成したパーツが入っているウィンドウの名前を示します。

タイプ スtring

適用可能なイベント

|          |  |  |
|----------|--|--|
| すべてのイベント |  |  |
|----------|--|--|



---

## 用語集

この用語集には、次のものから引用した用語とその定義が含まれています。

- 米国情報システム用辞書 ANSI X3.172-1990、著作権 1990 年 米国規格協会 (ANSI) コピーは米国規格協会、1430 Broadway, New York, New York, 10018 から購入することができます。これらの定義は定義の後のシンボル (A) によって明示されています。
- 国際標準化機構および国際電気工学委員会 (ISO/IEC JTC1/SC1) の分科会 1、合同技術委員会 1 によって開発された 情報技術用語。この用語集の出版済み部分の定義は、定義の後のシンボル (I) によって示されています。国際標準草案、委員会草案、および ISO/IEC JTC1/SC1 によって開発中の作業文書から引用した定義は、定義の後のシンボル (T) によって示され、これは SC1 の関係国際本部間でいまだ最終合意に至っていないことを示します。
- IBM 計算用ディクショナリー, New York: McGraw-Hill, 1994.
- オブジェクト指向インターフェース設計 IBM 共通ユーザー・インターフェース・ガイドライン, SC34-4399-00, Carmel, IN: Que Corporation, 1992.

### [ア行]

**アイコン.** オブジェクトのグラフィカル表現であり、イメージ、イメージの背景、およびラベルで構成されます。

**アイコン・ビュー.** コンテナに含まれている各オブジェクトがアイコンとして表示される標準的な目次ビュー。

**アクション.** (1) アクション・サブルーチンの同義語。(2) プロジェクトのパーツを操作するために使用したり、あるいはビルドにかかわるために使用する実行可能なプログラムまたはコマンド・ファイル。

**アクション・サブルーチン.** 特定のイベントに応答するために作成するロジック。

**アクティブ・ウィンドウ.** 現在ユーザーが対話しているウィンドウ。これがキーボード入力を受け取るウィンドウです。

**アニメーション制御パーツ.** Windows で AVI という拡張子を持つビデオ・ファイルの再生、または Java アプリケーションでアニメーション GIF シーケンスの再生を可能にするパーツ。

**アプリケーション.** コンピューター上で特定のユーザー・タスクを実行するために使用するソフトウェア・コンポーネントの集合。

**アプリケーション・プログラミング・インターフェース (API).** オペレーティング・システムによって提供される機能インターフェースまたは個別に注文するライセンス・プログラムであり、高水準言語で書かれたアプリケーション・プログラムがオペレーティング・システムまたはライセンス・プログラムの特定のデータまたは機能を使用できるようにするもの。

**アプレット.** Java で書かれ、Java 互換ブラウザーまたは appletviewer の内部で実行されるプログラム。

**アンカー.** 他のパーツを調整、サイズ決め、およびスペーシングする場合の参照点として使用する任意のパーツ。

**イベント.** パーツの状態に対する変更の結果として生成されるシグナル。たとえば、ボタンを押すと *Press* イベントが生成されます。

**イメージ・パーツ.** ウィンドウ上で、BMP または ICO ファイルからピクチャーを表示するのに使用するパーツ。

**インポート.** AS/400 表示装置ファイル・オブジェクトを該当する VARPG パーツに変換する機能。エクスポートと対比。

**ウィンドウ・パーツ.** 可視境界を持つ領域で、オブジェクトのビューを表したり、あるいはこれを使用してユーザーはコンピューター・システムでダイアログを管理します。パーツ・パレットまたはパーツ・カタログ内のウィンドウ・パーツを指してクリックし、それをプロジェクト・ウィンドウ上に置くことができます。

**ウェーブ・ファイル.** ウェーブ形式装置のオーディオ・サウンド用に使用するファイル。

**エクスポート.** アプリケーションの外側で使用するために内部ファイルを何らかの標準ファイル形式に変換する機能。インポートと対比。

**エラー・ログ.** エラー・ログにエラーのトラックを保存します。エディターを使用すると、ソース内でエラーが起こった場所を表示できます。

**オブジェクト.** (1) それ自身について記述し、場合によってはデータについて記述する一組のプロパティから成る名前の付いた記憶スペース。オブジェクトはストレージの中に存在し、そのスペースを占有するもので、それについて操作を実行することができます。オブジェクトの例としては、プログラム、ファイル、ライブラリー、およびフォルダーなどがあります。(2) ユーザー・インターフェースのビジュアルなコンポーネントであり、ユーザーはタスクを実行するためにこれで作業することができます。オブジェクトはテキストまたはアイコンとして表示されることがあります。

**オブジェクト指向プログラミング.** 他のオブジェクトと対話することのできるオブジェクトのデータおよび操作について記述するための階層的に編成されたクラスとして、プログラムを構造化するためのメソッド。(T)

**オブジェクト-アクション・パラダイム.** 対話の 1 つのパターンであり、このパターンでユーザーはオブジェクトを選択し、そのオブジェクトに適用するアクションを選択します。

**オブジェクト・プログラム.** 実行に適したターゲット・プログラム。オブジェクト・プログラムは、リンクを必要とする場合と、そうでない場合があります。(T)

**オペレーティング・システム.** コンピューター・システムの全体の操作を制御するシステム・プログラムの集合。

## [力行]

**カーソル.** ユーザーとキーボードの対話によって現れる可視位置の指示。

**[隠す] ボタン.** タイトル・バーにあるボタンで、ユーザーがウィンドウをクローズしないでワークスペースからそのウィンドウを除去する場合にクリックします。ウィンドウが隠されると、ウィンドウの状態 (ウィンドウ・リストに表示される) が変わります。最大化ボタンおよび最小化ボタンと対比。

**カラー・パレット.** アプリケーションの GUI 中の任意のパーツのカラーを変更するために使用するカラーのセット。

**カレンダー・パーツ.** カレンダーを追加するパーツであり、ユーザーはテキスト、カラー、およびその他の属性を組み込むためにこれを変更することができます。

**簡略記号.** 選択項目のテキスト内の 1 つの文字で、その文字の下に下線を引いて示されます。略号の選択も参照してください。

**キャンバス付きウィンドウ・パーツ.** ウィンドウ・パーツとキャンバス・パーツの組み合わせ。ウィンドウ・パーツおよびキャンバス・パーツも参照してください。

**キャンバス・パーツ.** 他のいろいろなパーツを指示してクリックし、それらを位置指定して、グラフィカル・ユーザー・インターフェースを作成するようにそれらを編成するパーツ。キャンバス・パーツは、ウィンドウ・パーツまたは

ノートブック・ページ・パーツのいずれかのクライアント域を占めます。キャンバス・パーツ付きノートブック・ページおよびキャンバス・パーツ付きウィンドウも参照してください。

**キャンバス・パーツ付きノートブック・ページ。** ノートブック・ページ・パーツとキャンバス・ページ・パーツの組み合わせ。ノートブック、キャンバス・パーツも参照してください。

**強調。** カラー変更を強調表示したり、あるいはオブジェクトまたは選択項目と対話するユーザーの能力に影響するこれらのオブジェクトまたは選択項目に関連する条件の可視な指示を強調表示すること。また、強調により、選択項目またはオブジェクトの状態についての追加情報がユーザーに与えられます。

**共通ユーザー・アクセス・アーキテクチャー (CUA アーキテクチャー)。** 人間とワークステーションまたは端末装置の間のダイアログの指針。

**共用コンポーネント。** 複数のプロジェクトがアクセスできるコンポーネント。

**組み合わせボックス。** 入力フィールドとリスト・ボックスの機能を結合する制御。組み合わせボックスには、入力フィールドを完成させるためにユーザーがスクロールして選択することのできるオブジェクトのリストが入っています。これがない場合は、ユーザーはテキストを直接入力フィールドに入力することができます。VisualAge RPG では、パーツ・パレットまたはパーツ・カタログ内の組み合わせボックス・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**クライアント。** (1) サーバーにデータを提供するために、サーバーに依存しているシステム。(2) VARPG アプリケーションが実行される PWS。DDE クライアントも参照してください。

**クライアント域。** ユーザーのワークスペースであるウィンドウ内の部分であり、ここでユーザーは情報を入力したり、選択フィールドから選択項目を選択します。プライマリー・ウィンドウにおいて、ユーザーが作業するオブジェクトをアプリケーション・プログラマーが提供する領域。

**クライアント/サーバー。** 分散データ処理における相互作用の形式であり、一方のサイトのプログラムが別のサイトのプログラムに要求を送信し、応答を待機します。要求側のプログラムをクライアントと呼び、応答側のプログラムをサーバーと呼びます。クライアント、サーバー、DDE クライアント、DDE サーバーも参照してください。

**グラフィカル・ユーザー・インターフェース (GUI)。** 高解像度グラフィックスを利用したユーザー・インターフェースのタイプ。グラフィカル・ユーザー・インターフェースには、グラフィックスの組み合わせ、オブジェクト - アクション・パラダイム、ポインティング・デバイスやメニュー・バーおよびその他のメニューの使用、オーバーラップ・ウィンドウ、およびアイコンが組み込まれています。

**グラフィック・プッシュボタン・パーツ。** グラフィックスのラベルが付いたプッシュボタンで、ユーザーがこれを選択した時に開始されるアクションを表します。プッシュボタン・パーツと対比。

**グラフ・パーツ。** ユーザーが GUI にグラフを追加するために使用できるパーツ。使用可能なグラフ・スタイルは折れ線グラフ、棒グラフ、折れ線と棒グラフ、または円グラフです。

**クリック。** 選択項目またはオブジェクトからポインターを外さないでマウス・ボタンを押して放すこと。ダブルクリックも参照してください。

**クリップボード。** データを一時的に保持するためにシステムによって提供された記憶領域。クリップボード中のデータは他のアプリケーションにも使用可能です。

**グループ・ボックス・パーツ。** 制御グループの回りを取り囲む四角形のフレームであり、これらの制御が関連していることを示し、そのグループに対して任意指定のラベルを提供します。VisualAge RPG では、パーツ・パレットまたはパーツ・カタログ内のグループ・ボックス・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**グループ・マーカー。** パーツをグループ内の最初のものであるとして認識するマーク。パーツのグループ内でカーソルを移動している時に、最後のパーツに達すると、カーソルはグループ内の最初のパーツに戻ります。

**コールド・リンク会話.** DDE において、クライアント・プログラムからサーバー・プログラムに対して行われる明示的な要求。サーバー・プログラムはこの要求に応答します。ホット・リンク会話と対比。

**構成.** 情報処理システムのハードウェアおよびソフトウェアが編成され、相互接続される方法 (T)。

**構文検査.** ソースの編集中に、各行の構文が正しいことを確認します。確認することによって、コンパイル・エラーを回避することができます。このオプションは、オンまたはオフに設定することができます。一定の仕様タイプ (C 仕様など) または特定のストリングを持つ行だけを表示することができます。

**項目.** 動的データ交換におけるデータの単位。たとえば、スプレッドシートにおける 左上のセルの位置は、行 1、列 1 です。このセル位置は、項目 RIC1 と呼ばれることもあります。

**コンテナー・パーツ.** 関連レコードを保管するパーツであり、それらを詳細、アイコン、またはツリー・ビューで表示します。

**コンパイル.** ソース・プログラムを実行可能プログラム (オブジェクト・プログラム) に変換すること。

**コンポーネント.** プロジェクト内の関連ファイルの機能上のソート。コンポーネントは、制御仕様書に NOMAIN および EXE キーワードがない時に作成されます。

**コンポーネント参照パーツ.** 1 つのコンポーネントが VARPG アプリケーション中の別のコンポーネントと通信できるようにするパーツ。

## [サ行]

**サーバー.** ネットワーク中にあり、他のシステム (クライアントと呼ばれる) の要求を処理するシステム。

**サーバーの別名.** サーバー名の代わりに使用することができるユーザー定義の名前。

**最小化ボタン.** タイトル・バーの右端のボタンの隣にあるボタンで、ウィンドウを最小サイズに縮小します。最大化ボタンおよび[隠す] ボタンと対比。

**最大化ボタン.** タイトル・バーの右端にあるボタンで、ウィンドウをその最大サイズに拡大するためにクリックします。最小化ボタン、[隠す] ボタンと対比。

**索引.** リスト・ボックスまたは組み合わせボックスなどの VARPG パーツ中の項目の ID 。

**サブファイル・パーツ.** レコードのリストを表示するために使用するパーツであり、沢山のフィールドから成っています。このパーツは AS/400 サブファイルに類似しています。サブファイル・フィールドも参照してください。

**サブファイル・フィールド.** サブファイル・パーツ内のフィールドを定義するために使用するフィールド。サブファイル・パーツも参照してください。

**サブプロシージャー.** サブプロシージャーはメイン・ソース・セクションの後に指定されるプロシージャーです。メイン・ソース・セクションの定義仕様の中に対応するプロトタイプが入っていなければなりません。

**サブメニュー.** 別のメニューの中のカスケード選択項目から表示され、それに関連した選択項目が入っているメニュー。サブメニューはプルダウン・メニューまたはポップアップ・メニューの長さを縮小するために使用します。サブメニュー・パーツも参照してください。

**サブメニュー・パーツ.** メニュー項目または既存のメニューからサブメニューを開始したり、あるいはメニュー・バー上のメニュー項目からプルダウン・メニューを開始するために使用するパーツ。サブメニューおよびメニュー項目・パーツも参照してください。

**参照解除.** パーツと AS/400 データベース・フィールドとの間の関連を除去するアクション。



**参照フィールド.** AS/400 のデータベース・フィールドであり、入力フィールド・パーツがそのプロパティを継承することができます。

**状況バー.** ウィンドウのパーツであり、現行のビューまたはオブジェクトの状態を示す情報を表示します。情報域も参照してください。

**状況バー・パーツ.** ウィンドウ上にあり、そのウィンドウの処理またはアクションについての追加情報を表示できるパーツ。

**詳細ビュー.** オブジェクトについての説明情報を提供するテキストが小アイコンに結合されている標準目次ビュー。

**情報域.** カーソルが置かれているオブジェクトまたは選択項目についての情報が表示されるウィンドウのパーツ。情報域には、処理の通常の完了についてのメッセージも表示されます。状況バーも参照してください。

**情報表示機能 (IPF).** プログラマブル・ワークステーション上でオンライン・ヘルプを作成するのに使用するツール。

**情報表示機能 (IPF) ファイル.** アプリケーションのヘルプ・ソースが保管されるファイル。

**進行状況バー・パーツ.** ファイルのコピー、データベースのロード、などの処理の進行をグラフィカルに示すために使用できるパーツ。

**進行状況表示.** 処理の進行状況をユーザーに知らせるために使用される 1 つまたは複数の制御。

**水平方向のスクロール・バー・パーツ.** ウィンドウに水平方向のスクロール・バーを追加するパーツ。このパーツによって、ユーザーは、情報のペインを左から右または右から左にスクロールできるようになります。

**スクロール・バー.** 特定の方向にまだ情報があることをユーザーに示し、マウスまたはページ・キーを使用して移動しその情報を表示できることを示すパーツ。

**スピン・ボタン・パーツ.** 入力フィールドの 1 つのタイプであり、関連があるが互いに排他的な選択項目のリングを表示し、ユーザーはこれをスクロールして 1 つの選択項目を選択することができます。ユーザーは入力フィールドに正しい選択を入力することもできます。パーツ・パレットまたはパーツ・カタログ内のスピン・ボタン・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**スライダー・アーム.** スライダー内の可視標識であり、ユーザーは数値を変更するためにこれを移動することができます。

**スライダー・パーツ.** ユーザー・インターフェースのビジュアル・コンポーネントであり、数量を表し、その数量に使用できる値の範囲に対するリレーションシップを表します。ユーザーはこの数量の値を変更することもできます。パーツ・パレットまたはパーツ・カタログ内のスライダー・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**スレッド.** プロセス中で実行される操作の最小の単位。

**静的テキスト・パーツ.** 他のパーツのラベルとして使用されるパーツ。入力フィールド・パーツに対するプロンプトなど。

**セカンダリー・ウィンドウ.** プライマリー・ウィンドウの中の情報に依存した情報が入っているウィンドウで、プライマリー・ウィンドウ中の対話を補足するために使用されます。プライマリー・ウィンドウも参照してください。ポップアップ・ウィンドウの同義語。

**設計ウィンドウ.** GUI Designer 中のウィンドウであり、ユーザー・インターフェースを作成するためのパーツが置かれています。

**選択ボタン.** 左マウス・ボタン を参照してください。

**選択枠.** VARPG パーツまたはオーダーメイド・パーツの回りに現れる可視枠で、マウスまたはキーボードを使用して移動できます。

**ソース・ディレクトリー.** VARPG アプリケーションのすべてのソース・ファイルが保管されているディレクトリー。

**ソース・パーツ.** ソース・パーツの状態が変更された時には常にターゲット・パーツに通知できるパーツ。ソース・パーツは複数のターゲットを持つことができます。

**操作ボタン.** 右マウス・ボタン を参照してください。

**外枠パーツ.** パーツのグループの回りを囲む長方形ボックスのパーツで、その中のすべてのパーツが関連していることを示します。

## [夕行]

**ターゲット・ディレクトリー.** ビルド後にコンパイル済みの VARPG アプリケーションが保管されるディレクトリー。ターゲット・フォルダーと対比。

**ターゲット・パーツ.** ソース・パーツの状態が変更された時には常に、ソース・パーツからリンク・イベントを受信するパーツ。

**ターゲット・フォルダー.** VARPG アプリケーションを表すアイコンが入っているオブジェクト。

**ターゲット・プログラム.** プロジェクトによってビルドされるオブジェクト。ダイナミック・リンク・ライブラリー (DLL) など。

**タイトル・バー.** 各ウィンドウの最上部にある領域で、システム・メニュー・シンボルが入っています。

**ダイナミック・リンク・ライブラリー (DLL).** リンク時ではなくロード時または実行時にプログラムにバインドされる実行可能コードおよびデータが入っているファイル。ダイナミック・リンク・ライブラリー中のコードおよびデータは複数のアプリケーションで同時に共用することができます。

**タイマー・パーツ.** 2 つのイベント間の時間間隔を追跡し、その間隔が経過した時に 2 番目のイベントを起動するために使用するパーツ。

**縦方向のスクロール・バー・パーツ.** ウィンドウに縦方向のスクロール・バーを追加するパーツ。このパーツによって、ユーザーは、情報のペインを縦方向にスクロールできるようになります。

**ダブルクリック.** マウス・ボタンを素早く 2 回押すこと。

**タブ・ストップ.** ユーザーがインターフェースの中を移動するためにタブ・キーを使用する時に、パーツにフォーカスできるように、そのパーツのタブ・ストップを設定するために使用する属性。

**チェック・ボックス・パーツ.** 選択項目を表している関連テキストを持つ正方形のボックス。ユーザーが選択項目を選択すると、その選択項目が選択されたことを示す標識がチェック・ボックスの中に現れます。その選択項目を再び選択することによって、チェック・ボックスを取り消すことができます。VisualAge RPG では、パーツ・パレットまたはパーツ・カタログ内のチェック・ボックス・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**直接編集.** ユーザーがオブジェクトをマウスでドラッグしたり、オブジェクトのポップアップ・メニューと対話することによって、そのオブジェクトを処理することのできる手法を使用すること。

**ツールバー.** ユーザーがマウスを使用して実行することのできるアクションを表す 1 つまたは複数のグラフィカル選択項目が入っているメニュー。

**ツリー・ビュー.** オブジェクトの目次を階層的に表示する方法。

**データベース.** (I) 複数ユーザーのためのデータをオンデマンドで受け入れ、保管し、および提供するための指定の構造体を持つデータの集合。(T) (2) システム中に保管されているすべてのデータ・ファイル。

**データ・オブジェクト.** テキスト、グラフィックス、オーディオ、またはビデオなどの情報を運ぶオブジェクト。

**デフォルト.** ユーザーによって値が指定されていない時に、システムまたはプログラムによって自動的に供給または想定される値。デフォルト値はプッシュボタンまたはグラフィック・プッシュボタンに割り当てることができます。

**デフォルト・アクション.** 何らかのアクション (たとえば Enter キーを押す) が取られた時に実行されるアクション。

**トークン強調表示.** コードの読みやすさを向上させます。異なるカラーまたはフォントを使用して、異なる言語コンポーネントの強調表示を構成して、プログラム構造を識別することができます。トークンの強調表示はオンにしたりオフにしたりすることができます。

**動的データ交換 (DDE).** プログラム間またはプログラムとデータ・ファイル・オブジェクトの間のデータの交換。1つのプログラムまたはセッションの中の情報に対して行われた変更が、他のプログラムによって作成された同じデータに適用されます。*DDE 会話*、*DDE クライアント*、*DDE サーバー*も参照してください。

**トピック.** 動的データ交換 (DDE) において、DDE 会話のサブジェクトとなるデータのセット。

**ドラッグ.** マウスを使用してオブジェクトを移動またはコピーすること。たとえば、マウス・ボタンを押したままで移動することにより、ウィンドウ・ボーダーをドラッグしてそのウィンドウを大きくすることができます。ドラッグ・アンド・ドロップも参照してください。

**ドラッグ・アンド・ドロップ.** マウスを使用してオブジェクトを直接移動し、別の場所に置くこと。

**ドロップダウン組み合わせボックス.** 組み合わせボックスの変形であり、ユーザーが明示的にアクションを実行して可視にするまでは、リスト・ボックスは隠されています。

**ドロップダウン・リスト.** 現在の選択項目だけが可視になっている単一の選択フィールド。他の選択項目が入っているリスト・ボックスを表示するためのアクションを明示的に実行するまでは、他の選択項目は隠されたままです。

## [ナ行]

**ナビゲーション・パネル.** サブファイル中のレコードの可視選択を制御するために使用するボタンのグループ。

**入出力 (I/O).** コンピューターに提供されるデータ、またはコンピューター処理の結果としてのデータ。

**入力フィールド・パーツ.** ユーザーが情報を入力できる画面上の領域 (そのフィールドが読み取り専用でない場合)。通常、入力フィールドの境界が指示されています。VisualAge RPG では、パーツ・パレットまたはパーツ・カタログ内の入力フィールド・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**入力フォーカス.** キーボードまたはマウスのどちらからでもユーザーの対話が可能なウィンドウの領域。

**ノートブック・パーツ.** ノートブックのグラフィカル表現。ノートブック・ページをノートブック・パーツに追加し、その後でそれらのページをタブ付きの区切り記号によって分離されたセクションにグループ化することができます。Windows では、ノートブックのことを Windows タブ制御と呼ぶことがあります。ノートブック・ページ・パーツ、キャンバス・パーツ付きノートブック・ページも参照してください。

**ノートブック・ページ・パーツ.** ノートブック・パーツにページを追加するために使用するパーツ。ノートブックも参照してください。

## [ハ行]

**パーツ.** VARPG アプリケーションの GUI を構成するオブジェクト。

**パーツ・カタログ.** VARPG アプリケーション用のグラフィカル・ユーザー・インターフェースを作成するために使用するすべてのパーツのための記憶スペース。

**パーツ・パレット.** アプリケーション用の現在のグラフィカル・ユーザー・インターフェースをビルドするのに最も適しているパーツの集合。1つのGUIを終了したら、パレットをきれいにふき取り、次のアプリケーションに必要なパーツをパーツ・カタログから追加することができます。

**パッケージ.** VARPG アプリケーションのすべてのパーツを配布用に収集するために使用する機能。

**非活動ウィンドウ.** 指定の瞬間にキーボード入力を受け入れることができないウィンドウ。

**左マウス・ボタン.** デフォルトとして、選択のために使用するマウス上の左のボタン。

**ビルド.** VARPG アプリケーションのコンポーネントを構成するいろいろなソース・コードが、実行可能なバージョンのアプリケーションを作成するためにコンパイルされ、リンクされるプロセス。

**ファイル.** 関連データの集合であり、割り当てられた名前でも保管されたり検索されます。ファイルには、プログラム (プログラム・ファイル・オブジェクト) を開始する情報、テキストまたはグラフィック (データ・ファイル・オブジェクト) を含む情報、あるいは一連のコマンド (バッチ・ファイル) を処理する情報を組み込むことができます。

**フィールド.** (1) ウィンドウ内の識別可能な領域であり、たとえば、ユーザーがテキストを入力する入力フィールドなど。(2) 名前や容量などの関連したバイトのグループであり、レコード中の単位として扱われます。

**フォーカス.** 入力フォーカスの同義語。

**フォント・パレット.** アプリケーション GUI 中のパーツのフォントを変更するために使用できるフォントのセット。

**復元ボタン.** ウィンドウが最大化された後に、タイトル・バーの右端の隅に現れるボタン。復元ボタンを選択すると、ウィンドウは最大化される前のサイズと位置に戻ります。最大化ボタンも参照してください。

**複数行編集 (MLE) パーツ.** ユーザーが複数行のテキストを入力することのできる入力フィールドを表すパーツ。

**プッシュボタン・パーツ.** ユーザーがプッシュボタンを選択した時に開始されるアクションを表すテキストがラベルとして付いているボタン。パーツ・パレットまたはパーツ・カタログ内のプッシュボタン・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。「グラフィック・プッシュボタン・パーツ」も参照してください。

**プライマリー・ウィンドウ.** ユーザーとアプリケーションの間の主たる会話が行われるウィンドウ。メイン・ウィンドウ同義。

**プラグイン.** ユーザーまたは外部のベンダーによって作成された機能で、VARPG プログラムで使用することができます。

**プルダウン・メニュー.** メニュー・バーの選択した項目またはシステム・メニュー・シンボルから拡張されるメニュー。プルダウン・メニューの中の選択項目はある意味で互いに関連しています。

**プログラマブル・ワークステーション (PWS).** ある程度の処理能力を持っているワークステーションで、ユーザーがその機能を変更することができるワークステーション。

**プロシージャー.** プロシージャーは、CALLP 命令コードで呼び出すことのできるコードの任意の部分です。

**プロシージャー・インターフェース定義.** プロシージャー・インターフェース定義は、プロシージャーの定義の中のプロトタイプ情報の反復です。これは、プロシージャーの入力パラメーターを宣言したり、プロシージャーの内部定義が外部定義 (プロトタイプ) と矛盾しないことを保証するために使用します。

**プロジェクト.** ダイナミック・リンク・ライブラリー (DLL) または実行可能ファイル (EXE) などの単一のターゲットをビルドするのに必要なデータおよびアクションの完全なセット。

**プロトタイプ.** プロトタイプは、呼び出しインターフェースの定義です。これには次のような情報が組み込まれています：呼び出しがバインド (プロシージャー) であるかダイナミック (プログラム) であるか；外部名；パラメーターの数および特長；どのパラメーターを渡さなければならないか；戻り値のデータ・タイプ (プロシージャーの場合)。

**プロパティ・ノートブック.** タブ区分ページで複数のセクションに分離されたページを含むバウンド・ノートブックに類似したグラフィカル表現。1つのセクションから別のセクションに移動するためには、ノートブックのタブを選択してください。

**プロンプト.** (1) ユーザーの応答を要求するために、プログラムによって送信される視覚または音によるメッセージ。(T) (2) ユーザーからの入力を要求したり、操作情報を提供する表示記号またはメッセージ。ユーザーは続行するためには、このプロンプトに応答しなければなりません。

**ポイント/クリック.** (1) パーツをパーツ・パレットまたはパーツ・カタログから GUI 設計ウィンドウ、アイコン・ビュー、またはツリー・ビューにコピーするために使用される選択方式。(2) パーツを希望するいずれかのビューに入れるためには、そのパーツを指示してクリックし、次にカーソルを選択したウィンドウに移動して、そのカーソルを指示して、パーツを表示したい位置でクリックします。アイコンおよびツリー・ビューでは、パーツは親パーツ上に置かれるので、ユーザーはそれを設計ウィンドウ中の希望する位置に移動しなければなりません。

**ぼかし表示.** ユーザーがそのパーツを選択したり直接操作することができないことを示すために、明度が減光されて表示されているもの。

**ボタン.** (1) アクションを要求したり開始するために使用するマウスなどのポインティング・デバイス上のメカニズム。(2) ウィンドウ内のグラフィカル・メカニズムであり、これを選択するとアクションが実行されます。ボタンの1つの例に [OK] プッシュボタンがあり、これを選択するとアクションが開始されます。

**ホット・リンク会話.** DDE において、サーバー上でデータが変更された時の、サーバー・プログラムによるクライアント・プログラムの自動更新。コールド・リンク会話と対比。

**ポップアップ・ウィンドウ.** 固定サイズの移動可能なウィンドウであり、アプリケーションがユーザーの要求の処理を続行できるように、アプリケーションに必要な情報をユーザーがこのウィンドウに提供します。セカンダリー・ウィンドウと同義。

**ポップアップ・メニュー.** オブジェクトに関連するメニューを要求した時に、そのオブジェクトの横に表示されるメニュー。これには、オブジェクトの現在のコンテキストに適した選択項目が含まれています。

**ポップアップ・メニュー・パーツ.** オブジェクトに関して要求した時に、そのオブジェクトの横に表示されるパーツ (そのパーツがユーザー・インターフェース上のオブジェクトに追加されている場合)。パーツ・パレットまたはパーツ・カタログ内のポップアップ・メニュー・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

## [マ行]

**マイグレーション.** (1) 変更された操作環境に移動することであり、通常は新しいリリースまたはバージョンのシステムに移動します。(2) ある階層のストレージから別の階層のストレージにデータを移動すること。

**マウス.** キーボードを使用しないで画面上のポインターを位置付けるために使用し、1つまたは複数のプッシュボタンを持つ装置。実行する選択項目または機能を選択するために使用したり、画面上で操作を実行するために使用します。たとえば、ある場所から別の場所に線をドラッグしたり描いたりします。

**マウス・ポインター.** カーソルの同義語。

**マウス・ボタン.** 選択項目を選択したり、アクションを開始したり、あるいはポインターでオブジェクトを操作するために使用するマウス上の機構。左マウス・ボタン および右マウス・ボタン も参照してください。

**右マウス・ボタン.** デフォルトとして、操作のために使用するマウス上の右のボタン。

**メイン・ウィンドウ.** プライマリー・ウィンドウを参照してください。

**メイン・ソース・セクション.** VARPG プログラムでは、メイン・ソース・セクションには、モジュールのすべてのグローバル定義が含まれています。コンポーネントの場合には、このセクションには、アクションおよびユーザー・サブルーチンも含まれます。

**メイン・プロシージャ.** メイン・プロシージャは、プログラムの入り口プロシージャとして指定でき、最初に呼び出された時に制御を受け取るサブプロシージャです。メイン・プロシージャは EXE の作成時にしか作成されません。EXE モジュールを参照してください。

**メッセージ.** (1) ユーザーによって要求された情報ではなく、予期しないイベントにตอบสนองしてプロダクトによって表示されるか、あるいは何か不都合なことが起こった時に表示される情報。(2) ある人またはプログラムから別の人またはプログラムに対して送信される通信。

**メッセージ・サブファイル・パーツ.** プログラム・ロジックに提供されている事前定義メッセージまたはテキストを表示できるパーツ。

**メッセージ・ファイル.** アプリケーション・メッセージが入っているファイル。このファイルは、ビルド・プロセス時にメッセージ・ソース・ファイルから作成されます。ビルドも参照してください。

**メディア・パーツ.** プログラムに、サウンド・ファイルおよびビデオ・ファイルを処理する機能を与えるパーツ。

**メディア・パネル・パーツ.** ユーザーに他のパーツ全般の制御を与えるために使用するパーツ。たとえば、メディア・パネル・パーツは、メディア・パーツのボリュームを制御するために使用することができます。

**メニュー.** オブジェクトに適用することのできる選択項目のリスト。メニューには、一定のコンテキストの下では選択できない選択項目も含まれることがあります。これらの選択項目はぼかし表示されます。

**メニュー項目パーツ.** メニュー上のグラフィカルまたはテキスト項目であるパーツ。ユーザーは何らかの方法でオブジェクトを処理するためのメニュー項目を選択します。

**メニュー・バー・パーツ.** ウィンドウの上部近くで、タイトル・バーの下で、ウィンドウの他の部分より上にある領域で、他のメニューにアクセスできる選択項目が含まれています。VisualAge RPG では、パーツ・パレットまたはパーツ・カタログ内のメニュー・バー・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

## [ヤ行]

**ユーザー定義パーツ.** ユーザーがカスタマイズした 1 つまたは複数のパーツから成るパーツであり、再使用のためにこれをパーツ・パレットまたはパーツ・カタログに保管します。パレットまたはカタログの中にある時には、他の VARPG パーツと同様に、このパーツをポイントおよびクリックして設計ウィンドウに入れることができます。

**ユーティリティ DLL.** *NOMAIN* モジュールを参照してください。

## [ラ行]

**ラジオ・ボタン・パーツ.** 横にテキストが付いている円。ラジオ・ボタンは、固定した選択項目のセットから 1 つの選択項目を選択できる組み合わせです。選択項目が選択されると、円の一部が塗りつぶされます。パーツ・パレットまたはパーツ・カタログ内のラジオ・ボタン・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**リスト・ボックス・パーツ.** ユーザーが選択できるスクロール可能な選択項目が入っている制御。VisualAge RPG では、パーツ・パレットまたはパーツ・カタログ内のリスト・ボックス・パーツを指してクリックし、それを設計ウィンドウ上に置くことができます。

**略号の選択.** 選択項目の簡略記号を入力することにより、その選択項目を選択する選択手法。

**リンク・イベント.** ソース・パーツの状態が変更された時に、常にターゲット・パーツが受け取るイベント。

**例外.** (1) プログラム言語において、実行時に起こる可能性のある異常な状態、通常の実行順序から逸脱する原因となる可能性のある異常な状態であり、それを定義したり、引き起こしたり、認識、無視、および処理するための機能がプログラム言語中に存在しています。(I) (2) VisualAge RPG において、ユーザーが要求したアクションが予定通りに完了するのを妨げたり、妨げる可能性のあるイベントまたは状態。例外は、プロダクトがユーザーの入力を解釈できない時に起こります。

## [ワ行]

**ワークエリア.** ユーザーのタスクにしたがってオブジェクトを編成するために使用する領域。ユーザーがワークエリアをクローズすると、このワークエリアにあるオブジェクトからオープンされたすべてのウィンドウがワークスペースから除去されます。

**ワークステーション.** ユーザーが作業することのできる装置。プログラマブル・ワークステーションも参照してください。

**ワークスペース.** 画面全体を埋める領域で、ユーザー・インターフェースを構成するすべてのオブジェクトを保持する領域。

**枠のサイズ変更.** マウスまたはキーボードを使用してパーツ (またはパーツのセット) をサイズ変更するために選択するパーツ (またはパーツのセット) の回りの枠またはフレーム。

## [数字]

**1 バイト文字セット (SBCS).** 各文字が 1 バイトのコードで表される文字セット。2 バイト文字セット (DBCS) と対比。

**2 バイト文字セット (DBCS).** 各文字が 2 バイトで表される文字のセット。256 個のコード・ポイントでは表しきれない記号が入っている、日本語、中国語、および韓国語、などの言語は 2 バイト文字セットを必要とします。各文字が 2 バイトを必要とするので、DBCS 文字の入力、表示、および印刷では、DBCS をサポートするハードウェアおよびソフトウェアが必要です。システムは次の 4 つの 2 バイト文字セットをサポートします: 日本語、韓国語、中国語 (簡体字)、および中国語 (繁体字)。1 バイト文字セット (SBCS) と対比。

## A

**activeX パーツ.** プロジェクトに ActiveX 制御オブジェクトを追加するパーツ。これにより、VARPG アプリケーションは、その属性にアクセスして、イベントをモニターすることができます。

**API.** アプリケーション・プログラミング・インターフェース。

**ASCII (情報交換用米国標準コード).** 7 ビット (パリティ検査を組み込む場合は 8 ビット) のコード化文字から成るコード化文字セットを使用する標準コードであり、データ処理システム、データ通信システム、および関連機器の間で情報交換用に使用されます。ASCII セットは制御文字とグラフィック文字から成ります。(A)

## B

**BMP.** ビットマップ・ファイルのファイル拡張子。

## C

**\*component パーツ.** コンポーネントの『パーツ表現』であるパーツ。各コンポーネントごとに 1 つの \*component パーツが自動的に作成され、それは目には見えません。

**CONFIG.SYS.** DOS、OS/2、または Windows オペレーティング・システムのブート・ドライブのルート・ディレクトリーにある構成ファイル。この中には、ハードウェアおよびソフトウェアをインストールし実行するために必要な情報が入っています。

**CUA アーキテクチャー.** 共通ユーザー・アクセス・アーキテクチャー。

## D

**DBCS.** 2 バイト文字セット。

**DDE.** 動的データ交換。

**DDE 会話.** DDE クライアントと DDE サーバーの間のデータの交換。コールド・リンク会話およびホット・リンク会話も参照してください。

**DDE クライアント.** DDE 会話を開始するアプリケーション。DDE サーバーと対比。DDE クライアント・パーツ、DDE 会話も参照してください。

**DDE クライアント・パーツ.** 動的データ交換 (DDE) プロトコルをサポートする他のアプリケーション (スプレッドシート・アプリケーションなど) とデータを交換するために使用するパーツ。

**DDE サーバー.** データを別の DDE 可能アプリケーションに提供するアプリケーション。DDE クライアントと対比。DDE 会話も参照してください。

**DLL.** ダイナミック・リンク・ライブラリー。

## E

**EBCDIC.** 拡張 2 進化 10 進コード。256 個の 8 ビット文字のコード化文字セット。

**EXE.** 実行可能ファイルの拡張子。

**EXE モジュール.** EXE モジュールはメイン・プロシージャとサブプロシージャから成ります。これは、制御仕様に EXE キーワードがある場合に作成されます。すべてのサブルーチン (BEGSR) はプロシージャに対してローカルでなければなりません。EXE には、名前がソース・ファイルの名前と一致するプロシージャが入っていなければなりません。これが EXE のメイン入り口点であり、すなわちメイン・プロシージャとなります。

## G

**GUI Designer.** パーツをパーツ・パレットから設計ウィンドウにドラッグ/ドロップすることによってインターフェースを作成するのに使用する一組のツール。

## I

**ICO.** アイコン・ファイルのファイル拡張子。

**INI.** OS/2 または Windows オペレーティング・システムにおいて、アプリケーションから別のアプリケーションを呼び出すために保持する必要があるアプリケーション固有の情報が入っているファイルのファイル拡張子。

**IPF.** 情報表示機能



## J

**JAR ファイル (.jar).** Java では、Java ARchive の省略形。多くのファイルを 1 つのファイルに集めるために使用されるファイル形式。

**Java.** リモート・オブジェクト間の対話をサポートする移植可能解釈コードのためのオブジェクト指向プログラミング言語。Java は、Sun Microsystems, Incorporated によって開発され、明示されたものです。

**Java 2 Software Development Kit (J2SDK).** Sun Microsystems 社が Java 開発者のために配布するソフトウェア。このソフトウェアには、Java インタープリター、Java クラス、および Java 開発ツールが組み込まれています。開発ツールには、コンパイラー、デバッガー、ディスアセンブラー、AppletViewer、スタブ・ファイル生成プログラム、および文書生成プログラムが組み込まれています。

**Java Bean パーツ.** VARPG アプリケーションが Sun Microsystem の JavaBeans にアクセスできるようにするパーツ。

**Java 仮想マシン (JVM).** Java 実行時環境 (JRE) の一部で、Java バイトコードの解釈を受け持ちます。

**Java 実行時環境 (JRE).** JRE を再配布したいエンド・ユーザーおよび開発者のための Java Developer Kit のサブセット。JRE は、Java 仮想マシン、Java コア・クラス、およびサポート・ファイルからなります。

**Java データベース接続性 (JDBC).** Java と広範囲のデータベースの間のデータベースから独立した接続のための業界標準。JDBC は、SQL 基本データベース・アクセスのための呼び出しレベルのアプリケーション・プログラミング・インターフェース (API) を提供します。

**Java ネイティブ・インターフェース (JNI).** Java 仮想マシン (JVM) の内部で実行する Java コードが他のプログラム言語で書かれた機能と相互に関連し合って機能できるようにするプログラミング・インターフェース。

**JavaBeans.** Java では、移植可能で、プラットフォームから独立した再使用可能コンポーネント・モデルです。

## M

**MID.** MIDI ファイルのファイル拡張子。

**MIDI ファイル.** 電子楽器デジタル・インターフェース・ファイル。

## N

**NOMAIN モジュール.** サブプロシージャだけが入っているモジュール。この中には、アクションまたはスタンドアロン・ユーザー・サブルーチンはありません。NOMAIN モジュールは、制御仕様の中に NOMAIN キーワードがある場合に作成されます。

## O

**odbc/jdbc パーツ.** VARPG アプリケーションが Windows ODBC API または Sun Microsystem 社の JDBC API をサポートするデータベース・ファイルにアクセスして、それを処理できるようにするパーツ。

## P

**PWS.** プログラマブル・ワークステーション。

## S

**SBCS.** 1 バイト文字セット。

**secure sockets layer (SSL).** Netscape Communications Corp. および RSA Data Security, Inc によって開発された一般的なセキュリティ機構。SSL によって、クライアントはサーバーを認証できるようになり、すべてのデータおよび要求が暗号化できるようになります。SSL によって保護されたセキュア・サーバーの URL は、http ではなく、https で始まっています。

**SSL.** Secure sockets layer。

## W

**WAV.** ウェーブ・ファイルのファイル拡張子。

---

## 参考文献

WebSphere Development Studio Client と関連したトピックの追加情報については、以下の IBM 資料を参照してください。

WebSphere Development Studio Client マニュアル:

- *WebSphere Development Studio Client for iSeries* ご使用に際して、SD88-5054-03。WebSphere Development Studio Client for iSeries についての情報を提供し、各種コンポーネントの概要、それらがどのように関連して機能するか、およびそれを使用することによる業務上の利点を示します。

VisualAge RPG マニュアル:

- *VisualAge for RPG プログラミング*, SC88-5607-05 には、VisualAge RPG を使用してアプリケーションを作成する場合の具体的な情報が記載されています。これには、アプリケーション開発サイクルのすべての段階で実行する必要があるステップが、設計からパッケージおよび分配まで説明されています。VARPG アプリケーション開発の概念と工程を明らかにするために、プログラミング例が収録されています。
- *ADTS/CS VisualAge for RPG パーツ解説書*, SD88-5040-03 には、VARPG の各パーツ、パーツ属性、パーツ・イベント、およびイベント属性が説明されています。本書は、VisualAge RPG を使用してアプリケーションを開発しているすべての方々の参照資料です。
- *VisualAge for RPG WINDOWS 版 言語解説書*, SC88-5608-04 には、VARPG 言語およびコンパイラーに関する参照情報が提供されます。
- *Java for RPG プログラマー*。Java 言語 (および RPG IV) を RPG 言語と比較しながら紹介しています。これは、Java を使って様々なことを経験するための最初のステップとして格好なものです。これには、Java および VisualAge for Java に関する対話式 CD チュートリアルも MINDQ 別に含まれています。
- *Experience RPG IV Tutorial*。RPG IV および ILE について肩の凝らない段階的方法によって学習するための対話式 CD チュートリアルです。この資料は、この興味深い RPG の新規バージョンでの実技経験を得る上で役立つ質問および演習が記載されたハンドブックです。
- VisualAge RPG ユーザーの関心を引く IBM 以外の資料に *VisualAge for RPG by Example* があります。.

インターネットにアクセスできる場合には、以下の Web サイトから iSeries および AS/400e についての最新の情報および資料を入手できます。

<http://www.ibm.com/eserver/iseries/infocenter>

Application Development Manager マニュアル:

- *AS/400 V3 適用業務開発ツール (ADT) セット /400 適用業務開発管理 /400 入門 および計画の手引き*, GC88-5245-00 には、Application Development Manager 機能を効率的に使用するために必要な基本概念および計画が説明されています。

- *DTS/ Application Development Manager User's Guide*, SC09-2133-02 には、Application Development Manager 機能に対して定義されるプロジェクトを作成し管理する方法が説明されています。
- *AS/400 アドバンスド・シリーズ V3.6 適用業務開発ツール (ADT) セット OS/400 用 適用業務開発管理 OS/400 用 自習書*, SC88-5417-00。Application Development Manager 機能を使用する実践的な体験を得ることができます。この手引書には、ステップ順に練習することによって、Application Development Manager 機能の使用法が説明されています。
- *ADTS/400: Application Development Manager API Reference*, SC09-2180-00。アプリケーション・プログラマーが Application Development Manager 機能に対する固有のインターフェースを作成する方法が説明されています。

情報表示機能マニュアル:

- *Information Presentation Facility Programming Guide* G25H-7110 には、情報表示機能 (IPF) を構成するエレメントが説明されています。IPF は、オンライン文書機能およびオンライン・ヘルプ機能の設計および開発をサポートするツールです。

SQL マニュアル:

- *IBM SQL Reference Version 2* (SC26-8416) Volume 2 には、以下に示すものの機能比較があります。
  - DB2
  - SQL/DS™
  - DB2/400™
  - DB2/6000™
  - IBM SQL
  - ISO-ANSI (SQL92E)
  - X/Open™ (XPG4-SQL).
- *DB2 Universal Database Administration Guide* S10J-8157 には、DB2 製品の使用および管理に必要な情報が示されています。
- *DB2 Universal Database Embedded SQL Programming Guide* S10J-8158 には、DB2 クライアント/サーバーのファミリー・サーバー (DB2 または DB2/400) にアクセスするアプリケーション・プログラムの設計およびコーディングの方法が説明されています。これには、構造化照会言語 (SQL) の使用およびアプリケーションでの API 呼び出しに関する詳細情報が説明されています。





Printed in Japan

SD88-5040-04



日本アイ・ビー・エム株式会社  
〒106-8711 東京都港区六本木3-2-12