

Oracle Reports Developer

Report Builder リファレンス

リリース 6*i*

2000 年 4 月

部品番号 : J01123-01

Oracle Reports Developer Report Builder リファレンス リリース 6i

部品番号: J01123-01

原本名: Oracle Reports Reference Release 6i

原本部品番号: A73174-01

Copyright © Oracle Corporation 1997, 1999. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	xiii
前提条件	xiv
表記規則	xiv
関連資料	xiv
ビルトイン	1
ビルトイン・パッケージ	2
SRW.SET_ATTR属性	4
ACTION	5
AFTCODE	6
AFTFORM_ESCAPE	6
AFTPAGE_ESCAPE	7
AFTREPORT_ESCAPE	8
BBCOLOR	10
BEFCODE	10
BEFFORM_ESCAPE	11
BEFPAGE_ESCAPE	11
BEFREPORT_ESCAPE	13
BFCOLOR	14
BOOKMARK	15
BORDERWIDTH	17
BORDPATT	17
FACE	17
FBCOLOR	18
FFCOLOR	18
FILLPATT	19
FORMATMASK	19
GCOLOR	19
GSPACING	20
HJUST	20
HYPERLINK	21
LINKTAG	23
PRINTER_INTRAY	25
STYLE	26
SZ	26
TEXT	27
WEIGHT	28
Report Builder PL/SQLパッケージ (SRW)	29
SRW.ADD_DEFINITION	29
SRW.APPLY_DEFINITION	30
SRW.BREAK	30

SRW.CONTEXT_FAILURE	31
SRW.DO_SQL	32
SRW.DO_SQL_FAILURE.....	34
SRW.FILE_ESCAPE	35
SRW.GETERR_RUN.....	35
SRW.GET_PAGE_NUM	35
SRW.INTEGER_ERROR	36
SRW.MAXROW_INERR	37
SRW.MAXROW_UNSET	38
SRW.MESSAGE.....	38
SRW.NULL_ARGUMENTS	39
SRW.PROGRAM_ABORT	40
SRW.REFERENCE.....	41
SRW.RUN_REPORT	42
SRW.RUN_REPORT_BATCHNO.....	45
SRW.RUN_REPORT_FAILURE	45
SRW.SET_ATTR.....	46
SRW.TEXT_ESCAPE.....	51
SRW.TRACE_ADD_OPTION	51
SRW.TRACE_END	52
SRW.TRACE_REM_OPTION	52
SRW.TRACE_START.....	52
SRW.TRACEOPTS.MASK	53
SRW.TRUNCATED_VALUE	54
SRW.UNKNOWN_QUERY	55
SRW.UNKNOWN_USER_EXIT	55
SRW.USER_EXIT	56
SRW.USER_EXIT20	57
SRW.USER_EXIT_FAILURE.....	58
設定属性プロシージャ	59
SRW.SET_AFTER_FORM_HTML	60
SRW.SET_AFTER_PAGE_HTML	61
SRW.SET_AFTER_PRINTING_CODE.....	62
SRW.SET_AFTER_REPORT_HTML.....	63
SRW.SET_BACKGROUND_BORDER_COLOR	64
SRW.SET_BACKGROUND_FILL_COLOR.....	65
SRW.SET_BEFORE_FORM_HTML.....	65
SRW.SET_BEFORE_PAGE_HTML.....	66
SRW.SET_BEFORE_PRINTING_CODE	68
SRW.SET_BEFORE_REPORT_HTML	68
SRW.SET_BOOKMARK	69
SRW.SET_BORDER_PATTERN.....	71
SRW.SET_BORDER_WIDTH	72
SRW.SET_CHARMODE_TEXT.....	72
SRW.SET_CUSTOM_SPACING	73
SRW.SET_DISPLAY_NAME	73
SRW.SET_FIELD	74
SRW.SET_FIELD_CHAR	75
SRW.SET_FIELD_DATE.....	76
SRW.SET_FIELD_NUM.....	77

SRW.SET_FILL_PATTERN.....	78
SRW.SET_FONT_FACE.....	78
SRW.SET_FONT_SIZE.....	79
SRW.SET_FONT_STYLE.....	79
SRW.SET_FONT_WEIGHT.....	80
SRW.SET_FOREGROUND_BORDER_COLOR.....	81
SRW.SET_FOREGROUND_FILL_COLOR.....	82
SRW.SET_FORMAT_MASK.....	82
SRW.SET_HYPERLINK.....	83
SRW.SET_HYPERLINK_ATTRS.....	85
SRW.SET_JUSTIFICATION.....	86
SRW.SET_LINKTAG.....	86
SRW.SET_MAXROW.....	87
SRW.SET_PAGE_NAVIGATION_HTML.....	89
SRW.SET_PDF_ACTION.....	90
SRW.SET_PRINTER_TRAY.....	91
SRW.SET_SPACING.....	92
SRW.SET_TEXT_COLOR.....	92
トリガー.....	93
使用する「レポート・トリガー」の選択.....	94
「レポート・トリガー」の実行順序.....	94
グループ・フィルタ.....	97
式.....	98
妥当性チェック・トリガー.....	99
フォーマット・トリガー.....	102
アクション・トリガー.....	105
参照カーソル問合せ.....	106
After Parameter Formトリガー.....	109
After Reportトリガー.....	110
Before Parameter Formトリガー.....	110
Before Reportトリガー.....	111
Between Pagesトリガー.....	111
プロパティ.....	113
Oracle8使用上の注意.....	114
プロパティ・パレットについて.....	116
プロパティ・リスト.....	121
アンカー・プロパティ.....	129
子の枠の割合(%).....	131
子の枠の種類.....	131
子オブジェクト名.....	131
水平方向に縮小.....	132
垂直方向に縮小.....	132
親の枠の割合(%).....	135

親の枠の種類	135
親オブジェクト名	136
ボイラープレート・プロパティ	137
HTMLタグを含む	138
最小孤立行	139
ソース・ファイルの書式	141
ソース・ファイル名	142
型	143
枠付きの行伸長	143
ボタン・プロパティ	146
ラベルの型	147
テキスト	148
アイコン名	149
型	149
マルチメディア・ファイル	150
マルチメディア・ファイル・タイプ	151
マルチメディア列	151
マルチメディア列の型	152
PL/SQLトリガー	153
チャート・プロパティ	154
チャート・ファイル名	156
チャート・ハイパーリンク	157
チャート・パラメータと列プロパティ	159
チャート・パラメータ	159
レポート列 (チャート・パラメータ対応)	160
チャート問合せ	160
レポート・グループ	161
チャート列	162
レポート列 (チャート列対応)	162
共通の列プロパティ	164
ブレイク順序	164
列タイプ	167
コメント	167
データベース列名	168
データ型	168
ファイル形式	170
ファイルからの読み込み	170
NULL時の値	172
幅	173
ブレイク順序の設定	174
共通のレイアウト・オブジェクト・プロパティ	176
コメント	178
水平拡張度	178
アンカー・オブジェクトと連動	181

名前	183
後で改ページ	184
前で改ページ	187
ページ保護	188
基本印刷オン	191
印刷オブジェクト・オン	192
前のプリンタ・コード	195
後のプリンタ・コード	196
垂直拡張度	197
ハイパーリンク	202
ハイパーリンクの宛先	203
ブックマーク	204
アプリケーション・コマンド・ライン (PDF)	205
条件付き書式	206
表示名	206
追加のハイパーリンク属性	207
レイアウト・セクションの終了	208
データベース列プロパティ	209
フィールド・プロパティ	210
ソース・データ型	211
書式マスク	212
DATEおよびTIME書式マスク構文	213
数値書式マスク構文	214
可視	217
ページ番号付け	219
ソース	223
式列プロパティ	225
粹プロパティ	226
グループ・プロパティ	227
子ディメンション	227
フィルタ・タイプ	230
名前	231
コメント	232
レコード数	232
PL/SQLフィルタ	233
データ・リンク・プロパティ	234
子列	236
子問合せ	237
条件	237
親列	238
親グループ	239
SQL句	239
マトリックス・プロパティ	241
クロス積グループ	241

水平繰返し枠	243
垂直繰返し枠	244
データベース列オブジェクト・プロパティ	245
OLE2プロパティ	245
パラメータ・プロパティ	246
システム・パラメータ	246
追加	247
追加の属性 (HTML)	247
コメント	248
データ型	248
第1列を隠す	249
初期値	250
入力マスク	250
削除	251
事前定義済み値に対するリストを制限	252
SELECT文/スタティック値	252
有効化トリガー	253
値	253
幅	253
値リスト	254
パラメータ・フォーム・ボイラープレート・プロパティ	255
名前	255
型	255
パラメータ・フォーム・フィールド・プロパティ	256
データ型	256
名前	256
ソース	257
ブレースホルダ列プロパティ	258
PL/SQL式	260
列ソート・プロパティ	261
問合せプロパティ	262
コメント	262
外部問合せソース・ファイル	263
最大取出し行	264
名前	265
SQL問合せ文	265
型	268
データベース参照列プロパティ	269
繰返し枠プロパティ	269
列モード	270
枠間の水平間隔	273
ページ当りの最大レコード数	274
最小孤立レコード数	275
印刷方向	276

ソース	280
枠間の垂直間隔	280
レポート・プロパティ	282
サイズの単位	283
最大水平ボディ・ページ	284
最大垂直ボディ・ページ	284
パネルの印刷順序	285
方向	286
配布	287
タイトル	287
ロール名	287
幅 (パラメータ・フォーム・ウィンドウ)	288
高さ (パラメータ・フォーム・ウィンドウ)	289
ページ数	290
ページ・ナビゲーション・コントロール・タイプ	290
ページ・ナビゲーション・コントロール値	291
レポート前型	292
レポート前値	293
レポート後型	294
レポート後値	295
ページ前型	296
ページ前値	296
ページ後型	297
ページ後値	298
フォーム前型	299
フォーム前値	300
フォーム後型	301
フォーム後値	302
キャラクタ単位で設計	303
プレビューアのヒント行を使用	303
プレビューアのヒント行のテキスト	304
プレビューアのステータス行を使用	304
プレビューアのステータス行のテキスト	305
ビットマップ・オブジェクトを含む	305
境界線を含む	306
ホスト・メニュー項目を使用不可にする	306
画面分割キーを使用不可にする	307
ズーム・キーを使用不可にする	307
ズームで起動	307
プレビューア・タイトルの抑止	308
参照カーソル問合せプロパティ	309
PL/SQL文	309
セクション・プロパティ	310
配布	310

高さ	310
ページ当りの水平パネル	312
方向	313
レポート高	314
レポート幅	314
ページ当りの垂直パネル	315
幅	316
サマリー列プロパティ	317
計算位置	317
ファンクション	320
積順序	322
リセット位置	327
ソース	329
テンプレート・プロパティ	330
整列	331
サマリーをフィールドの位置に合わせ	333
バックグラウンド・カラー	333
フィールドとラベル間（水平）	334
枠とフィールド間（水平）	335
枠とフィールド間（垂直）	335
マスターとディテール間（水平）	336
マスターとディテール間（垂直）	337
ページと枠間（水平）	337
ページと枠間（垂直）	338
同等枠間（水平）	339
同等枠間（垂直）	339
境界線	340
文字揃え	341
破線	341
日付揃え	342
枠のフォアグラウンド・カラー	343
枠のバックグラウンド・カラー	343
枠のパターン	344
行当りのフィールド数	345
塗りパターン	345
フォント	346
フォアグラウンド・カラー	347
イメージ	348
フィールド間（水平）	348
フィールド間（垂直）	349
枠間（水平）	349
枠間（垂直）	350
位置揃え	351
数値揃え	351

フィールドの上にラベルを置く	352
位置	353
スタイル	353
テキスト	354
テキスト・カラー	355
垂直間隔の使用	355
XMLプロパティ	357
XMLの設定	357
XMLタグ	357
XMLタグ属性	358
XML出力から除外	359
XML Prolog型	360
XML Prolog値	360
外部XMLタグ	361
外部XML属性	362
XMLタグを含む	363
実行プログラム	377
実行プログラム名	378
実行プログラムの起動	378
コマンド・ライン・オプションのヘルプ	378
キーワードの使用法	380
データベースへのログイン	381
明示的ログイン	381
ユーザーID	381
自動ログイン	382
リモート・ログイン	382
RWBLD60	383
RWBLD60コマンド・ライン引数	383
RWCGI60	416
RWCLI60	416
RWCLI60コマンド・ライン引数	416
RWCON60	424
RWCON60コマンド・ライン引数	425
RWMTS60	434
RWRQM60	434
RWRQV60	435
RWOWS60	435
RWRBE60	435
RWRUN60	436
RWRUN60コマンド・ライン引数	437

索引 473

はじめに

Oracle Reports Developer Report Builderリファレンス リリース6iによるこそ。

このリファレンス・ガイドでは、Report Builderを効果的に利用できるようにするための情報と、次の項目に関する詳細な情報が説明されています。

- ビルトイン
- トリガー
- プロパティ
- 実行プログラム

ここでは、このガイドの構成を説明し、Report Builderを使用する際に参考になるその他の情報源を紹介します。

前提条件

まず、ご使用のコンピュータおよびそのオペレーティング・システムについて精通している必要があります。たとえば、ファイルの削除およびコピーのコマンドの知識があり、検索パス、サブディレクトリおよびパス名をの概念を理解していなければなりません。詳細は、各オペレーティング・システムの製品マニュアルを参照してください。

アプリケーション・ウィンドウの要素などのMicrosoft Windowsの基本要素も理解している必要があります。エクスプローラ、タスクバー、タスクマネージャ、またはレジストリなどのプログラムに精通している必要があります。

表記規則

このマニュアルでは、次のような表記上の規則を使用しています。

規則	意味
固定幅フォント	固定幅フォントのテキストは、表示されたとおりに入力するコマンドを示します。PCに入力するテキストでは、特に断りのない限り大文字と小文字を区別しません。 コマンドでは、大カッコと縦線以外の句読点は表示されているとおりに正確に入力する必要があります。
小文字	コマンド文の小文字は変数を表します。適切な値に置き換えてください。
大文字	テキスト内の大文字は、コマンド名、SQL予約語、キーワードを表します。
ゴシック・テキスト	メニュー・選択項目やボタンなど、ユーザー・インタフェース項目を示すには、ゴシック・テキストが使用されます。
C>	C>はDOSプロンプトを表します。実際とは異なる場合があります。

関連資料

次のOracleマニュアルを参照することもできます。

タイトル	部品番号
『Oracle Forms Developer and Oracle Reports Developer アプリケーション作成ガイド リリース6i』	J00449-01

ビルトイン

ビルトイン・パッケージ

Oracleでは、PL/SQLベースのアプリケーションを作成したりデバッグする際に使用できるパッケージ・プロシージャをいくつか提供しています。

PL/SQLコードから、次のクライアント側ビルトイン・パッケージのプロシージャ、ファンクション、例外を利用できます。

- Report Builderのビルトイン・パッケージ
- その他のビルトイン・パッケージ

Report Builderビルトイン・パッケージ

Report Builderには、ビルトイン・パッケージが1つのみあります。

- SRW

SRWのビルトイン・パッケージ

SRW.ADD_DEFINITION
SRW.APPLY_DEFINITION
SRW.BREAK
SRW.CONTEXT_FAILURE
SRW.DO_SQL
SRW.DO_SQL_FAILURE
SRW.GETTERR_RUN
SRW.GET_PAGE_NUM
SRW.INTEGER_ERROR
SRW.MAXROW_INERR
SRW.MAXROW_UNSET
SRW.MESSAGE
SRW.NULL_ARGUMENTS

SRW.PROGRAM_ABORT
SRW.REFERENCE
SRW.RUN_REPORT
SRW.RUN_REPORT_BATCHNO
SRW.RUN_REPORT_FAILURE
SRW.SETAttributes
SRW.TRACE_ADD_OPTION
SRW.TRACE_END
SRW.TRACE_REM_OPTION
SRW.TRACE_START
SRW.TRACEOPTS.MASK
SRW.TRUNCATED_VALUE
SRW.UNKNOWN_QUERY
SRW.UNKNOWN_USER_EXIT
SRW.USER_EXIT_
SRW.USER_EXIT20
SRW.USER_EXIT_FAILURE

SRW.SET_ATTR属性

ACTION

AFTCODE

AFTFORM_ESCAPE

AFTPAGE_ESCAPE

AFTREPORT_ESCAPE

BBCOLOR

BEFCODE

BEFFORM_ESCAPE

BEFPAGE_ESCAPE

BEFREPORT_ESCAPE

BFCOLOR

BOOKMARK

BORDERWIDTH

BORDPATT

FACE

FBCOLOR

FFCOLOR

FILLPATT

FORMATMASK

GCOLOR

GSPACING

HJUST

HYPERLINK

LINKTAG

PRINTER_INTRAY

STYLE

SZ

TEXT

WEIGHT

ACTION

新機能：

この属性は、SRW.SET_PDF_ACTIONプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

PDFビューア内のオブジェクトがクリックされたとき、ローカル・マシン上で実行されるコマンドです。

注意: アクションに関連付けられたオブジェクトを、Webリンクのソースにすることはできません。

値

ローカル・マシン上の任意の有効なコマンド・ライン (c:\orawin95\bin\rwrwin60
userid=scott/tiger report=example.rdfなど)

ACTIONの制限

- ACTIONは、PDF出力専用です。
- ACTIONの設定は、次のトリガーに限られます。

フォーマット・トリガー

ACTIONの例

注意: この例では、ACTION属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_PDF_ACTIONプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The format trigger below defines an "action" (in
```

```
** this case a call to the Reports Runtime) that
** should be executed when the user clicks on the
** boilerplate object B_2.
*/

function B_2FormatTrigger return boolean is
begin
    srw.attr.mask := srw.ACTION_ATTR;
    srw.attr.action := 'c:¥orawin¥bin¥rwrwin60' ||
        'userid=scott/tiger ' ||
        'report=example.rdf';
    srw.set_attr(0,srw.attr);
    return (TRUE);
end;
```

AFTCODE

新機能：

この属性は、SRW.SET_AFTER_PRINTING_CODEプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトの各行の後で実行されるプリンタ・エスケープ・シーケンスを指します。オブジェクトの印刷がトリガーされた後、実際に印刷される前に、プリンタ・エスケープ・シーケンスが挿入されます。AFTCODEは、キャラクタ・モードでレポートを実行しているときにしか使用されません。

値

&番号形式の文字列。番号は、パッケージReport Builderプリンタ・エスケープ・シーケンスかまたは自分で作成したプリンタ・エスケープ・シーケンスに割り当てられた番号です。

AFTFORM_ESCAPE

新機能：

この属性は、SRW.SET_AFTER_FORM_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

HTMLパラメータ・フォームの下に表示する任意のテキスト、図形またはHTMLコマンドです。この属性は、ロゴまたは標準リンクをパラメータ・フォーム上に配置するのに便利です。

値

この属性には、次の2つの項目を指定する必要があります。

- AFTFORM_ESCAPE_TYPE は、SRW.FILE_ESCAPEまたはSRW.TEXT_ESCAPEに設定されません。
- AFTFORM_ESCAPE_VALUEには、AFTFORM_ESCAPE_TYPEに指定したものに依りて、ファイル名または有効HTMLを含むテキスト文字列を指定できます。ファイルを指定した場合、ファイル検索メソッドが検索に使用されます。

AFTPAGE_ESCAPE

新機能：

この属性は、SRW.SET_AFTER_PAGE_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

ドキュメントのページの最後に表示させたい任意のテキスト、図形またはHTMLコマンドです。この属性は、ロゴまたは標準リンクをHTML文書の各ページの下端に配置するのに便利です。

値

この属性には、次の2つの項目を指定する必要があります。

- AFTPAGE_ESCAPE_TYPEは、SRW.FILE_ESCAPEまたはSRW.TEXT_ESCAPEに設定可能です。
- AFTPAGE_ESCAPE_VALUEには、AFTPAGE_ESCAPE_TYPEに指定したものに依りて、ファイル名または有効HTMLを含むテキスト文字列を指定できます。ファイルを指定した場合、ファイル検索メソッドが検索に使用されます。

AFTPAGE_ESCAPEの制限

- この属性のSRW.SET_ATTRを指定する場合、object_idとしてSRW.REPORT_IDを使用する必要があります。

```
SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
```

- ページの下端に含まれるデフォルトHTMLを以下に示します。ページ間に表示するデフォルトのセパレータ・ラインが不要な場合は、これは不要です。

```
<hr size=5 noshade>
```

- AFTPAGE_ESCAPEをレポートのすべてのページに適用する場合は、Before Reportトリガーのように、レポートがフォーマットを開始する前に起動するトリガーの中で、それを指定する必要があります。
- BEFPAGE_ESCAPEを現行のページのみ適用する場合は、そのページ上のオブジェクトに対してフォーマット・トリガーの中で指定する必要があります。

グラフィック・ページ・セパレータの例

注意: この例では、AFTPAGE_ESCAPE属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_AFTER_PAGE_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The example below inserts a GIF file as a
** page separator.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.ATTR.MASK := SRW.AFTPAGE_ESCAPE_ATTR;
    SRW.ATTR.AFTPAGE_ESCAPE_TYPE := SRW.TEXT_ESCAPE;
    SRW.ATTR.AFTPAGE_ESCAPE_VALUE := '<CENTER>' ||
    '<IMG ALT="Ruler" SRC="line.gif" VSPACE=10>' ||
    '<BR></CENTER>';
    SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
end if;
return (TRUE);
end;
```

AFTREPORT_ESCAPE

新機能:

この属性は、SRW.SET_AFTER_REPORT_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

文書の終わりに表示する任意のテキストまたは図形、HTMLコマンドです。この属性は、ロゴまたは標準リンクをHTML文書の終わりに配置するのに便利です。

値

この属性には、次の2つの項目を指定する必要があります。

- AFTREPORT_ESCAPE_TYPEは、SRW.FILE_ESCAPEまたはSRW.TEXT_ESCAPEに設定してください。
- AFTREPORT_ESCAPE_VALUEには、AFTREPORT_ESCAPE_TYPEに指定したものに依じて、ファイル名または有効HTMLを含むテキスト文字列を指定できます。ファイルを指定した場合、ファイル検索メソッドが検索に使用されます。

AFTREPORT_ESCAPEの制限

- この属性のSRW.SET_ATTRを指定する場合、object_idとしてSRW.REPORT_IDを使用する必要があります。

```
SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
```

- レポートの下端に含まれるデフォルトHTMLを以下に示します。AFTREPORT_ESCAPEを使用してこのテキストを置換する場合、等価のHTMLコマンドが含まれていることを確認する必要があります。

```
</body></html>
```

- Before Reportトリガーのように、レポートがフォーマットされる前に起動するトリガーの中で、AFTREPORT_ESCAPEを設定する必要があります。

ホーム・ページへのリンクの例

注意: この例では、AFTREPORT_ESCAPE属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_AFTER_REPORT_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The example below inserts two links to home pages
** at the end of the report output.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.ATTR.MASK := SRW.AFTREPORT_ESCAPE_ATTR;
    SRW.ATTR.AFTREPORT_ESCAPE_TYPE := SRW.TEXT_ESCAPE;
```

```
SRW.ATTR.AFTREPORT_ESCAPE_VALUE := '<CENTER>' ||
'A HREF="http://www.oracle.com/">' ||
'Oracle Corporation</A> -' ||
chr(10) ||
'A HREF="http://home.netscape.com/">' ||
'Netscape</A> </CENTER>' ||
'</BODY> </HTML>';
SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
end if;
return (TRUE);
end;
```

BBCOLOR

新機能：

この属性は、SRW.SET_BACKGROUND_BORDER_COLOR プロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトのバックグラウンド境界色です。この属性はキャラクタ・モードでは無視されます。

値

有効な色の名前。

BEFCODE

新機能：

この属性は、SRW.SET_BEFORE_PRINTING_CODE プロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトの各行の前で実行されるプリンタ・エスケープ・シーケンスを指します。オブジェクトの印刷がトリガーされた後、実際に印刷される前に、プリンタ・エスケープ・シーケンスが挿入されます。BEFCODEは、キャラクタ・モードでレポートを実行しているときにしか使用されません。

値

&番号形式の文字列。番号は、パッケージReport Builderプリンタ・エスケープ・シーケンスかまたは自分で作成したプリンタ・エスケープ・シーケンスに割り当てられた番号です。

BEFFORM_ESCAPE

新機能：

この属性は、SRW.SET_BEFORE_FORM_HTML プロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

HTMLパラメータ・フォームの上に表示する任意のテキストまたは図形、HTMLコマンドです。この属性は、ロゴまたは標準リンクをパラメータ・フォーム上に配置するのに便利です。

値

この属性には、次の2つの項目を指定する必要があります。

- BEFFORM_ESCAPE_TYPEは、SRW.FILE_ESCAPEまたはSRW.TEXT_ESCAPEに設定してください。
- BEFFORM_ESCAPE_VALUEには、BEFFORM_ESCAPE_TYPEに指定したものに依じて、ファイル名または有効HTMLを含むテキスト文字列を指定できます。ファイルを指定した場合、ファイル検索メソッドが検索に使用されます。

BEFPAGE_ESCAPE

新機能：

この属性は、SRW.SET_BEFORE_PAGE_HTML プロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

ドキュメントにあるページの最初に表示する任意のテキスト、図形またはHTMLコマンドです。この属性は、ロゴまたは標準リンクをHTML文書の各ページの先頭に配置するのに便利です。

値

この属性には、次の2つの項目を指定する必要があります。

- BEFPAGE_ESCAPE_TYPEは、SRW.FILE_ESCAPEまたはSRW.TEXT_ESCAPEに設定してください。
- BEFPAGE_ESCAPE_VALUEには、BEFPAGE_ESCAPE_TYPEに指定したものにに応じて、ファイル名または有効HTMLを含むテキスト文字列を指定できます。ファイルを指定した場合、ファイル検索メソッドが検索に使用されます。

BEFPAGE_ESCAPEの制限

- この属性のSRW.SET_ATTRを指定する場合、object_idとしてSRW.REPORT_IDを使用する必要があります。

```
SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
```
- BEFPAGE_ESCAPEをレポートのすべてのページに適用する場合は、Before Reportトリガーのように、レポートがフォーマットを開始する前に起動するトリガーの中で、それを指定する必要があります。
- AFTPAGE_ESCAPEを現行のページのみ適用する場合は、そのページ上のオブジェクトに対してフォーマット・トリガーの中で指定する必要があります。

BEFPAGE_ESCAPEとAFTPAGE_ESCAPEの例

注意: この例では、BEFPAGE_ESCAPE属性およびAFTPAGE_ESCAPE属性の設定にSRW.SET_ATTRが使用されています。これらの属性は、SRW.SET_BEFORE_PAGE_HTMLプロシージャおよびSRW.SET_AFTER_PAGE_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The example below centers the document in
** the browser.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.ATTR.MASK := SRW.BEFPAGE_ESCAPE_ATTR +
                    SRW.AFTPAGE_ESCAPE_ATTR;
    SRW.ATTR.BEFPAGE_ESCAPE_TYPE := SRW.TEXT_ESCAPE;
    SRW.ATTR.BEFPAGE_ESCAPE_VALUE := '<CENTER>';
    SRW.ATTR.AFTPAGE_ESCAPE_TYPE := SRW.TEXT_ESCAPE;
    SRW.ATTR.AFTPAGE_ESCAPE_VALUE := '</CENTER>';
    SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
end if;
```

```
return (TRUE);  
end;
```

BEFREPORT_ESCAPE

新機能：

これらの属性は、SRW.SET_BEFORE_PAGE_HTMLプロシージャおよびSRW.SET_AFTER_PAGE_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

文書の始めに表示する任意のテキストまたは図形、HTMLコマンドです。この属性は、ロゴまたは標準リンクをHTML文書の先頭に配置するのに便利です。

値

この属性には、次の2つの項目を指定する必要があります。

- BEFREPORT_ESCAPE_TYPE は、SRW.FILE_ESCAPEまたはSRW.TEXT_ESCAPEに設定可能です。
- BEFREPORT_ESCAPE_VALUEには、BEFREPORT_ESCAPE_TYPE用に指定したものに依じて、ファイル名または有効HTMLを含むテキスト文字列を指定できます。ファイルを指定した場合、ファイル検索メソッドが検索に使用されます。

BEFREPORT_ESCAPEの制限

- この属性のSRW.SET_ATTRを指定する場合、object_idとしてSRW.REPORT_IDを使用する必要があります。

```
SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
```

- レポートの先頭に含まれるデフォルトHTMLを以下に示します。BEFREPORT_ESCAPEを使用してこのテキストを置換する場合、等価のHTMLコマンドが含まれていることを確認する必要があります。

```
<html>  
<body bgcolor="#ffffff">
```

- Before Reportトリガーのように、レポートがフォーマットを開始する前に起動するトリガーの中で、BEFREPORT_ESCAPEを設定する必要があります。

BEFREPORT_ESCAPEとAFTREPORT_ESCAPEの例

注意: この例では、BEFREPORT_ESCAPE属性およびAFTREPORT_ESCAPE属性の設定にSRW.SET_ATTRが使用されています。これらの属性は、SRW.SET_BEFORE_REPORT_HTMLプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The example below sets both BEFREPORT_ESCAPE and
** AFTREPORT_ESCAPE in the Before Report trigger.
** The HTML for BEFREPORT_ESCAPE is located in a file
** named logo.htm.The HTML for AFTREPORT_ESCAPE is
** specified within the PL/SQL itself.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.ATTR.MASK := SRW.BEFREPORT_ESCAPE_ATTR +
                    SRW.AFTREPORT_ESCAPE_ATTR;
    SRW.ATTR.BEFREPORT_ESCAPE_TYPE := SRW.FILE_ESCAPE;
    SRW.ATTR.BEFREPORT_ESCAPE_VALUE := 'logo.htm';
    SRW.ATTR.AFTREPORT_ESCAPE_TYPE := SRW.TEXT_ESCAPE;
    SRW.ATTR.AFTREPORT_ESCAPE_VALUE := '<ADDRESS>' ||
    ' Questions?- ' ||
    '<A HREF=mailto:webmaster@xyztech.com>' ||
    'webmaster@xyztech.com</A>' ||
    '</ADDRESS>' ||
    '</body></html>';
    SRW.SET_ATTR(SRW.REPORT_ID, SRW.ATTR);
end if;
return (TRUE);
end;
```

BFCOLOR

新機能：

この属性は、SRW.SET_BACKGROUND_FILL_COLORプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトの塗りカラーです。この属性はキャラクタ・モードでは無視されます。

値

有効な色の名前。

BOOKMARK

新機能：

この属性は、SRW.SET_BOOKMARKプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

マスターHTMLドキュメントのフレーム内またはPDFビューア内に現れる文字列です。ブックマークをクリックすると、ウィンドウの一番上に対応付けられたオブジェクトが表示されます。

値

インデントおよび順序付けに関する情報がないテキスト文字列。ブックマークは、レポートがオブジェクトを渡す順に、ブックマーク・リストに表示されます。

順序付けおよびインデントを明示したx#book_mark_name形式のテキスト文字列で、xにはアウトライン番号が入ります。シャープ記号(#)とアウトライン番号は、「ブックマーク」ウィンドウには表示されませんが、順序とインデントを決めるのに使用されます。たとえば、次のように指定します。

```
1#Expense Summary Section
2#Expense Detail Section
2.1#Expenses for the Administration Department
2.2#Expenses for the Engineering Department
2.3#Expenses for the Sales Department
2.3.1#Expenses for the Eastern Sales Region
2.3.2#Expenses for the Northern Sales Region
2.3.3#Expenses for the Southern Sales Region
2.3.4#Expenses for the Western Sales Region
```

BOOKMARKの制限

- 同じアウトライン番号が何回も使用される場合、エントリはすべて表示されますが、順序はレポートがオブジェクトを渡した順で定義されます。
- 番号がとんでいる場合、次の2つのうちの1つが発生します。同等レベルの番号がとんでいる場合は、目に見える影響はありません（たとえば、1.3.1と1.3.3で、1.3.2がない場合、同じインデント・レベルで並んで表示されます）。上位の番号と低位の番号の間がとんでいる場合、中間のレベルが必要に応じて生成されます（たとえば、1.0の後が2.1.1ならば、見かけ上の2

と2.1のエントリが定義されます。そのタイトルは、後の実際のエントリと同じになります。

- BOOKMARKの設定は、次のトリガーに限られます。

フォーマット・トリガー

BOOKMARKの例

注意: この例では、BOOKMARK属性の設定にSRW.SET ATTRが使用されています。この属性は、SRW.SET_BOOKMARKプロシーダを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシーダを使用することをお勧めします。

```
/* The format trigger below defines a bookmark for
** the boilerplate object B_2. Notice the use of
** explicit ordering information (1#) in this example.
** If you did not want the ordering information, you
** could omit 1#.
*/

function B_2FormatTrigger return boolean is
begin
  srw.attr.mask := srw.BOOKMARK_ATTR;
  srw.attr.bookmark := '1#Expense Summary Section';
  srw.set_attr(0,srw.attr);
  return (TRUE);
end;
```

動的BOOKMARKの例

注意: この例では、BOOKMARK属性の設定にSRW.SET ATTRが使用されています。この属性は、SRW.SET_BOOKMARKプロシーダを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシーダを使用することをお勧めします。

```
/* The format trigger below defines a bookmark for
** the boilerplate object B_2. Notice that the name
** of the bookmark is dynamic.CATEGORY is a column
** value that is concatenated with the string Expense
** Summary Section for each execution of the format
** trigger.In this case, CATEGORY could contain
** ordering information (e.g., 1#) or perhaps a string
** that makes the bookmark unique within the report.
*/

function B_2FormatTrigger return boolean is
begin
  srw.attr.mask := srw.BOOKMARK_ATTR;
  srw.attr.bookmark := :category ||
  ' Expense Summary Section';
  srw.set_attr(0,srw.attr);
```

```
    return (TRUE);
end;
```

BORDERWIDTH

新機能：

この属性は、SRW.SET_BORDER_WIDTHプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトの枠幅。

値

0	境界線がないことを意味します。
任意の正の整数	ビットマップ・レポートの場合はそのパイカ数(1パイカは12ポイント)の、キャラクタ・モード・レポートの場合は1字の枠幅を意味します。

BORDPATT

新機能：

この属性は、SRW.SET_BORDER_PATTERNプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトの枠の塗りパターンです。この属性はキャラクタ・モードでは無視されます。

値

有効パターン名。

FACE

新機能：

この属性は、SRW.SET_FONT_FACEプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHARまたはDATE、NUMBER型フィールドのフェースです。この属性はキャラクタ・モードでは無視されます。

値

レポートが実行されるマシン上の有効フォント名。

FBCOLOR

新機能：

この属性は、SRW.SET_FOREGROUND_BORDER_COLORプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトのフォアグラウンドの境界色です。この属性はキャラクタ・モードでは無視されません。

値

有効な色の名前。

FFCOLOR

新機能：

この属性は、SRW.SET_FOREGROUND_FILL_COLORプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトのフォアグラウンドの塗りカラーです。この属性はキャラクタ・モードでは無視されます。

値

有効な色の名前。

FILLPATT

新機能：

この属性は、SRW.SET_FILL_PATTERNプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトの塗りパターンです。この属性はキャラクタ・モードでは無視されます。

値

有効パターン名。

FORMATMASK

新機能：

この属性は、SRW.SET_FORMAT_MASKプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

DATEまたはNUMBER型フィールドの書式マスクです。

値

任意の有効書式マスク。

GCOLOR

新機能：

この属性は、SRW.SET_TEXT_COLORプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHAR、DATEまたはNUMBER型フィールドのグローバルな文字色です。この属性はキャラクタ・モードでは無視されます。

値

有効な色の名前。

GSPACING

新機能：

この属性は、SRW.SET_CUSTOM_SPACINGプロシージャまたはSRW.SET_SPACINGプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHARまたはDATE、NUMBER型フィールドまたはボイラープレート・テキストのグローバルなテキスト行間。この属性はキャラクタ・モードでは無視されます。

値

srw.single_spacing

srw.onehlf_spacing

srw.double_spacing

srw.custom_spacing

非常に小さいVGS単位で間隔を指定します。srw.custom_spacingを使用する場合は、次にsrw.attr.customを使用してVGS単位のカスタム数を指定します。たとえば、次のように指定します。

```
srw.attr.mask := SRW.GSPACING_ATTR;  
srw.attr.gspacing := SRW.CUSTOM_SPACING;  
srw.attr.custom := 200;  
srw.set_attr (0, srw.attr);
```

HJUST

新機能：

この属性は、SRW.SET_JUSTIFICATIONプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHAR、DATEまたはNUMBER型フィールドの水平方向です。

値

srw.left_hjust

srw.center_hjust

srw.right_hjust

srw.flush_hjust ビットマップ・レポートでは無視されます。

HYPERLINK

新機能：

この属性は、SRW.SET_HYPERLINKプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

URL Webリンクの仕様です。

注意: Webリンクのソースであるオブジェクトをアクションに関連付けることもできません。

値

有効リンク:

同じマシン上の別の文書(たとえば、file:/private/mynewdoc.pdf または file:///C:/temp/mynewdoc.pdf)

別のマシン上の別の文書(たとえば、http://www.newmach.com/newdoc.pdf)

現行文書内の宛先(たとえば、#my_dest_name)

ローカル文書内の宛先(たとえば、file:/private/somedoc.pdf#a_dest_name)

リモート文書内の宛先(たとえば、http://www.newmach.com/newdoc.pdf#some_dest_name)

すべてのURL (たとえば、http://www.newmach.com/newdoc.html, ftp://www.reposit.com/filetoget.example, http://www.somemch.com/cgi-bin/webmenu?choice1)

HYPERLINKの制限

- HYPERLINKの設定は、次のトリガーに限られます。

フォーマット・トリガー

- PDFビューアからリモート・サーバーまたはHTML文書へのWebリンクをたどるには、Webブラウザと連動するようにPDFビューアを構成する必要があります(たとえば、ヘルパー・アプリケーションとして構成したり、プラグインとしてWebブラウザにインストールする)。

HYPERLINKの例

注意: この例では、HYPERLINK属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_HYPERLINKプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The format trigger below defines a Web link
** to a destination (mytarget) in another document
** (newdoc.pdf) for the boilerplate object B_2.
** Note:If the target were in the same document,
** you would omit http://www.newmach.com/newdoc.pdf.
*/

function B_2FormatTrigger return boolean is
begin
  srw.attr.mask := srw.HYPERLINK_ATTR;
  srw.attr.hyperlink :=
    'http://www.newmach.com/newdoc.pdf' ||
    '#mytarget';
  srw.set_attr(0,srw.attr);
  return (TRUE);
end;
```

動的HYPERLINKの例

注意: この例では、HYPERLINK属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_HYPERLINKプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The format trigger below defines a Web link
** for the boilerplate object B_3. Notice how the
** destination of the link is determined dynamically
** based upon the values of SRCDIR and CF_PICKVIDEO.
** For example, if the value of SRCDIR were
** http://www.newmach.com and CF_PICKVIDEO were good.avi,
** this function would assign the following Web
** link to the object:
** http://www.newmach.com/webdemo/src/good.avi.
*/
```

```

function B_3FormatTrigger return boolean is
begin
  srw.attr.mask := srw.HYPERLINK_ATTR;
  srw.attr.hyperlink := :srcdir||:cf_pickvideo;
  if ( upper(:cf_pickvideo) like '%GOOD%' ) then
    srw.attr.mask := srw.attr.mask +
      srw.FBCOLOR_ATTR +
      srw.BBCOLOR_ATTR;
    srw.attr.fbcolor := 'green';
    srw.attr.bbcolor := 'green';
  end if;
  srw.set_attr(0,srw.attr);
  return (TRUE);
end;

/* SRCDIR is a parameter whose value is determined at
** runtime by the following After Parameter Form trigger
** */

function AfterPForm return boolean is
begin
  :srcdir := :web_server || '/webdemo/src/';
  :docsdire := :web_server || '/webdemo/docs/';
  return (TRUE);
end;

/* CF_PICKVIDEO is a formula column whose value is
** determined by the following function
** */

function CF_pickvideoFormula return Char is
begin
  if ( :avg_h_div < .80 )
    then return ('bad.avi');
    else return ('good.avi');
  end if;
end;

```

LINKTAG

新機能：

この属性は、SRW.SET_LINKTAGプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

オブジェクトの識別子であり、Webリンクの連結先として使用することが可能です。

値

ASCIIの26個の英文字(大文字および小文字)、数字またはアンダースコアのみを使った、有効で、一意の名前。他の特殊文字は、アンダースコア文字に自動的に変換されます。

LINKTAGの制限

- LINKTAGの設定は、次のトリガーに限られます。

フォーマット・トリガー

LINKTAGの例

注意: この例では、LINKTAG属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_LINKTAGプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The format trigger below assigns an identifier
** (mytarget) to the boilerplate object B_500.
** This means that the object can now be the destination
** of a Web link.
*/

function B_500FormatTrigger return boolean is
begin
  srw.attr.mask := SRW.LINKTAG_ATTR;
  srw.attr.linktag := 'mytarget';
  srw.set_attr(0, SRW.ATTR);
  return (TRUE);
end;
```

動的LINKTAGの例

注意: この例では、LINKTAG属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_LINKTAGプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```
/* The format trigger below assigns an identifier to
** the field F_Dept by concatenating some static text
** (dept_details_) with the value of the source column
** (DEPTNO).This function ensures that a unique
** identifier is assigned to each instance of F_Dept.
*/

function F_DeptFormatTrigger return boolean is
begin
  srw.attr.mask := SRW.LINKTAG_ATTR;
  srw.attr.linktag := 'dept_details_' ||
```

```

LTRIM(TO_CHAR(:deptno));
srw.set_attr(0, SRW.ATTR);
return (TRUE);
end;
```

PRINTER_INTRAY

新機能:

この属性は、SRW.SET_PRINTER_TRAYプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

有効なプリンタ・トレイ名で、レポートのフォーマットの際に、異なるプリンタ・トレイへ切り替えるのに使用されます。

値

「ページ設定」ダイアログで、プリンタに定義した有効で一意的な名前。

PRINTER_INTRAYの制限

- PRINTER_INTRAY設定は、次のトリガーに限られます。

Between Pagesトリガー

Before Reportトリガー

フォーマット・トリガー

- Motif版のPCLプリンタにレポートを出力する場合、PRINTER_INTRAYは無視されます。

PRINTER_INTRAYの例

注意: この例では、PRINTER_INTRAY属性の設定にSRW.SET_ATTRが使用されています。この属性は、SRW.SET_PRINTER_TRAYプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

```

/* The example below sets the printer tray in the Between Pages trigger.
*/
```

```

function BetweenPages return boolean is
begin
  srw.attr.mask           := SRW.PRINTER_INTRAY_ATTR;
  srw.attr.printer_intray := 'letterhead';
```

```
    srw.set_attr(SRW.REPORT_ID, srw.attr);  
return (TRUE);  
end;
```

STYLE

新機能:

この属性は、SRW.SET_FONT_STYLEプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHAR、DATEまたはNUMBER型フィールドのフォント・スタイルです。この属性はビットマップ・レポートでは無視されます。

値

srw.plain_style
srw.italic_style
srw.oblique_style
srw.underline_style
srw.outline_style
srw.shadow_style
srw.inverted_style
srw.overstrike_style
srw.blink_style

SZ

新機能:

この属性は、SRW.SET_FONT_SIZEプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHARまたはDATE、NUMBER型フィールドのフォント・サイズです。この属性はキャラクタ・モードでは無視されます。

値

レポートを実行するマシン上の指定フォントの有効サイズ。

TEXT

新機能:

この属性は、SRW.SET_CHARMODE_TEXTプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHARまたはDATE、NUMBER型フィールドのキャラクタ・モード・テキスト特性です。この属性はビットマップ・レポートでは無視されます。

値

srw.plain_texta

srw.reverse_texta

srw.bold_texta

srw.reversebold_texta

srw.underline_texta

srw.underlinereverse_texta

srw.underlinebold_texta

srw.reverseboldunderline_texta

WEIGHT

新機能:

この属性は、SRW.SET_FONT_WEIGHTプロシージャを使用して設定するほうが適しています。互換性を保つために、SRW.SET_ATTRでもこの属性を設定できるようになっていますが、新しい簡素化されたプロシージャを使用することをお勧めします。

説明

CHARまたはDATE、NUMBER型フィールドのフォントの太さです。この属性はキャラクタ・モードでは無視されます。

値

srw.ultralight_weight

srw.extralight_weight

srw.light_weight

srw.demilight_weight

srw.medium_weight

srw.demibold_weight

srw.bold_weight

srw.extrabold_weight

Report Builder PL/SQLパッケージ (SRW)

時間を節約するために、Report Builderには、どのライブラリやレポートでも参照できるファンクションおよびプロシージャ、例外が多数入ったパッケージ（つまり、PL/SQL構成体を集めたもの）が付属しています。Report Builderのパッケージの名前はSRWです。そのため、SRWパッケージの構成体を参照するときには、「SRW.DO_SQL」のように、必ずその構成体の前に「SRW」を付ける必要があります。

注意: 別の製品(SQL*Plusなど)から、SRWパッケージ内の構成体を参照することはできません。

パッケージ内にある構成体は、「パッケージ・ファンクション」、「パッケージ・プロシージャ」、「パッケージ例外」というように、通常「パッケージ」を付けて呼ばれます。

SRW.ADD_DEFINITION

説明

このプロシージャでは、列または変数内に保管されているXMLレポート定義が取り出され、文書バッファに追加されます。レポート定義をすべてバッファに入れるには、SRW.ADD_DEFINITIONを複数回実行する必要があります。バッファ内に定義が入ると、SRW.APPLY_DEFINITIONを使用してレポートに適用できます。

構文

```
SRW.ADD_DEFINITION(:column_name | variable);
```

使用上の注意

- XMLレポート定義を適用するには、SRW.APPLY_DEFINITIONビルトインを使用します。たとえば、次のように指定します。

```
srw.add_definition (:xml_col);  
srw.apply_definition;
```

- XMLレポート定義がファイル内に配置されている場合は、SRW.APPLY_DEFINITIONのみを使用してファイルを参照できます。たとえば、次のように指定します。

```
srw.apply_definition('d:¥xml_reps¥web.xml');
```

SRW.APPLY_DEFINITION

説明

このプロシージャでは、文書バッファまたはファイル・システムに保管されているXMLレポート定義が取り出され、レポートに追加されます。文書バッファからの定義が適用された場合、SRW.APPLY_DEFINITIONでは引数を使用しません。ファイルから定義が適用された場合、SRW.APPLY_DEFINITIONでは名前およびファイルのパス名が引数として使用されます。

構文

```
SRW.APPLY_DEFINITION[(filename.xml)];
```

使用上の注意

- 複数のXMLレポート定義を適用したい場合は、適用する順序でSRW.APPLY_DEFINITION文を別々に並べます。
- XMLレポート定義がファイル内にある場合は、SRW.APPLY_DEFINITIONのみを使用してファイルを参照できます。たとえば、次のように指定します。

```
srw.apply_definition('d:¥xml_reps¥web.xml');
```

- 文書バッファからXMLレポートを適用するには、最初にSRW.ADD_DEFINITIONを使用してバッファ内に定義を入れておく必要があります。たとえば、次のように指定します。

```
srw.add_definition (:xml_col);  
srw.apply_definition;
```

SRW.BREAK

説明

このプロシージャは、レポートの実行を一時的に停止し、すべての列とパラメータの現在の設定値を表示します(読込み専用)。読込み専用画面を適用すると、レポートの実行が再開されます。

構文

```
SRW.BREAK;
```

SRW.BREAKの制限

- このプロシージャは、フォーム前またはフォーム後トリガーでは無効です。これは、レポートのデータが、Before Reportトリガー(フォーム後トリガー実行後に起動)が起動するまでフェッチされないからです。
- すべての列とパラメータの値が、読込み専用スクリーンに表示されます。PL/SQL変数やその

他のオブジェクトの値は表示されません。

SRW.CONTEXT_FAILURE

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1426: <construct_name>は無効なコンテキストから実行されています。

構文

```
SRW.CONTEXT_FAILURE;
```

使用上の注意

Report Builderでは、Report Builderのパッケージ・ファンクションまたはプロシージャが誤ったコンテキストで使用されると、この例外を呼び出します(下表を参照)。

この表では、「NO」はそのコンテキストではそのファンクションまたはプロシージャをコールできないことを、「YES」はコールできることを意味します。

名前	パラメータ・フォーム	データ・モデル	フォーマット・トリガー	レポート・トリガー
srw.break	NO	YES	YES	NO
srw.do_sql	YES	YES	YES	YES
srw.geterr_run	YES	YES	YES	YES
srw.get_page_num	NO	NO	YES	NO
srw.message	YES	YES	YES	YES
srw.reference	YES	YES	YES	YES
srw.run_report	YES	YES	YES	YES
srw.set_attr	NO	NO	YES	NO
srw.set_field_char	NO	NO	YES	NO
srw.set_field_date	NO	NO	YES	NO
srw.set_field_num	NO	NO	YES	NO
srw.set_maxrow	NO	YES	YES	YES
srw.user_exit	YES	YES	YES	YES

SRW.CONTEXT_FAILUREの例

```
/* Suppose you want your own error message raised,
** instead of the default error message.
** You could handle this exception in the following way:
*/
```

```

EXCEPTION
when SRW.CONTEXT_FAILURE then
  srw.message(4000, 'Contact the Application
    Development group regarding SRW.CONTEXT_FAILURE. ');
  raise srw.program_abort;

```

SRW.DO_SQL

説明

このプロシージャは、Report Builder内から指定したSQL文を実行します。SQL文はDDL（データを定義する文）でも、DML（データを操作する文）でもかまいません。DML文は、通常、SRW.DO_SQLよりもPL/SQLでのほうが高速に実行できます。

PL/SQLではDDL文を実行できないので、SRW.DO_SQLパッケージ・プロシージャは、ユーザー・イグジットを介さずにReport Builder内でDDL文を実行するのに特に便利です。DDLまたはDML文の詳細は、『Oracle8 Server SQLリファレンス』を参照してください。

構文

```
SRW.DO_SQL (sql_statement CHAR);
```

パラメータ

sql_statement	任意の有効SQL文。Report Builderオブジェクト名の前には、必ずコロンの(:)を付けてください。
---------------	--

SRW.DO_SQLの制限

- レポート・トリガーの実行順序で、SET TRANSACTION READONLYがどこで発生するかに注意してください。
- バインド変数の値は、最大64,000バイトまで。（値がこの制限を超えると、最初の64,000バイト分を超過した分は切り捨てられます。）
- INTO句に文字列の宛先としてパラメータを使用する場合、パラメータに選択値が入るだけの幅があることを確認する必要があります。たとえば、下に示すSRW.DO_SQL文の場合、宛先パラメータ(my_ename)には、ENAME列の最大幅に相当する幅が必要です。これは、選択値の後ろには、その値の仮定サイズまで空白が入るからです。パラメータの幅が十分でないと、切捨て例外が発生します。SELECTリスト項目の最大幅がわからない場合は、2000をパラメータの幅として使用してください。

```
srw.do_sql('SELECT ENAME INTO :my_ename FROM EMP');
```

SRW.DO_SQLの例

```

/* Suppose you want your report to create a table named CHECK
** just before the Runtime Parameter Form is displayed.

```

```

** Because CREATE TABLE is a SQL DDL statement (and PL/SQL
** cannot perform DDL statements), you need to use SRW.DO_SQL.
** Therefore, your PL/SQL could look like this in the Before Form trigger:
*/
/* Additional Information:If you use a table created in this way for your
** report output, the table must exist before you create your query in
the
** data model.Otherwise, Report Builder would not be able to parse your
query.
*/
FUNCTION CREATETAB RETURN BOOLEAN IS
BEGIN
SRW.DO_SQL('CREATE TABLE CHECK (EMPNO NUMBER NOT NULL
          PRIMARY KEY, SAL NUMBER (10,2)) PCTFREE 5
          PCTUSED 75');
RETURN(TRUE);
EXCEPTION
WHEN SRW.DO_SQL_FAILURE THEN
SRW.MESSAGE(100, 'ERROR WHILE CREATING CHECK TABLE. ');
SRW.MESSAGE(50, 'REPORT WAS STOPPED BEFORE THE RUNTIME
PARAMETER FORM. ');
RAISE SRW.PROGRAM_ABORT;
END;

/* Suppose you want to create a "table of contents" by getting the
** first character of a column's value, and page number on which its
** field fires to print.Assume that you want to put the "table of contents"
** into a table named SHIP.You could write the following construct:
*/

DECLARE
PAGE_NO      NUMBER;
PAGE_FOR     INDEX NUMBER;
SORT_CHAR    CHAR(1);
CMD_LINE     CHAR(200);
BEGIN
SORT_CHAR := :SORT_NAME ;
IF :CALLED = 'Y' THEN
SRW.GET_PAGE_NUM(PAGE_FOR);
SRW.USER_EXIT('RWECOP PAGE_FOR
              P_START_PAGENO');
SRW.MESSAGE(2,TO_CHAR(:P_START_PAGENO));
END IF;
SRW.GET_PAGE_NUM(PAGE_NO);
CMD_LINE := 'INSERT INTO SHIP VALUES
            ('''||SORT_CHAR||''', '''||TO_CHAR(PAGE_NO)||''')';
SRW.MESSAGE(2,CMD_LINE);
SRW.DO_SQL(CMD_LINE);
COMMIT;
EXCEPTION

```

```
WHEN DUP_VAL_ON_INDEX THEN
    NULL;
WHEN SRW.DO_SQL_FAILURE THEN
    SRW.MESSAGE(1, 'FAILED TO INSERT ROW INTO SHIP
TABLE');
WHEN OTHERS THEN
    COMMIT;
END;
```

SRW.DO_SQL_FAILURE

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1425: DO_SQLパッケージ実行中にエラーが発生しました。 - REP-msg ORA-msg

この場合:

REP-msg: Report Builderのメッセージ。

ORA-msg: オプションのOracleメッセージで、Report Builderメッセージに関する追加情報。

構文

```
SRW.DO_SQL_FAILURE;
```

使用上の注意

Report Builderでは、SRW.DO_SQLパッケージ・プロシージャに失敗したとき（ユーザーにDDL権限がないのに、SRW.DO_SQLで表を作成しようとした場合など）、この例外を呼び出します。

SRW.DO_SQL_FAILUREの例

```
/* Suppose you want your own error message raised,
** instead of the default error message.
** You could handle this exception in the following way:
*/

EXCEPTION
when SRW.DO_SQL_FAILURE then
    srw.message(1000, 'Error occurred while creating
table CHECKS.');
```

SRW.FILE_ESCAPE

HTMLがファイル内にあることを示します。SRW.FILE_ESCAPEを指定する場合は、SRW.xxxx_ESCAPE_VALUE (xxxxには、BEFREPORTなどのエスケープの名前が入ります) のファイル名を入力する必要があります。

SRW.GETERR_RUN

説明

このファンクションは、Report BuilderでSRW.RUN_REPORTプロシージャの実行中にエラーが検出された場合に、エラー・メッセージを戻します。

構文

```
SRW.GETERR_RUN;
```

戻り値

エラー・メッセージ。

SRW.GETERR_RUNの例

```
/* Suppose you are sending parts of a report to users via Oracle*Mail.
** For more information, see "SRW.RUN_REPORT". Also,
** suppose that if SRW.RUN_REPORT fails, you want to display a message
** that explains why it failed.Your PL/SQL could look like this:
*/

BEGIN
DECLARE TMP CHAR(100);
begin
    srw.run_report('batch=yes report=send.rdf
        destype=file desname=send.lis desformat=dflt');
    exception when srw.run_report_failure then
        tmp := srw.geterr_run;
        srw.message(1000, tmp);
    end;
END;
```

SRW.GET_PAGE_NUM

説明

このプロシージャは、現行のページ番号を戻します。フィールドのフォーマット・トリガー・プロパティでそのページ番号を使用する場合に便利です。

構文

```
SRW.GET_PAGE_NUM (page_num);
```

パラメータ

page_num	現行のページ番号を取得する変数。
----------	------------------

戻り値

現行のページ番号

SRW.GET_PAGE_NUMの制限

- SRW.GET_PAGE_NUMが有効なのは、フォーマット・トリガーにおいてのみです。他の場所に入力しても効果はありません。

SRW.GET_PAGE_NUMの例

```
/* Suppose you want to perform a computation based upon a page number.
** In the field's Format Trigger, you could use SRW.GET_PAGE_NUM:
*/

BEGIN
DECLARE PAGE_NUM NUMBER;
  begin
    srw.get_page_num (page_num);
    srw.set_field_num (0, page_num + 3);
  end;
END;
```

SRW.INTEGER_ERROR

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1417: <SRW.MESSAGEまたはSRW.SET_MAXROW>に無効な整数引数が渡されました。

構文

```
SRW.INTEGER_ERROR;
```

使用上の注意

Report Builderでは、内部エラーの結果、SRW.MESSAGEまたはSRW.SET_MAXROWが非整数によって使用された場合、この例外を呼び出します。(内部エラーがない場合は、PL/SQLがPL/SQL例外のVALUE_ERRORによって型の不一致を見つけます。)

SRW.INTEGER_ERRORの例

```
/* Suppose you want your own error message raised,
** instead of the default error message.
** You could handle this exception in the following way:
*/

EXCEPTION
when SRW.INTEGER_ERROR then
    srw.message(4000, 'Contact the Application
        Development group regarding SRW.INTEGER_ERROR. ');
    raise srw.program_abort;
```

SRW.MAXROW_INERR

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1424: SRW.SET_MAXROWの実行中に内部エラーが発生しました。

このエラーが発生した場合は、サポート担当に連絡してください。

構文

```
SRW.MAXROW_INERR;
```

使用上の注意

Report Builderでは、SRW.SET_MAXROWパッケージ・プロシージャの実行中に内部エラーを検出した場合、この例外を呼び出します。

SRW.MAXROW_INERRの制限

- SRW.MAXROW_INERRを処理する場合、そのレポートには内部に問題があるので、必ずSRW.PROGRAM_ABORTを呼び出してレポートの実行を終了してください。

SRW.MAXROW_INERRの例

```
/* Suppose you want your own error message raised,
** instead of the default error message.
** You could handle this exception in the following way:
*/

EXCEPTION
when SRW.MAXROW_INERR then
    srw.message(1000, 'Contact Oracle's customer
        support: SRW.MAXROW_INERR');
    raise srw.program_abort;
```

SRW.MAXROW_UNSET

説明 この例外は、一時的にレポートの実行を停止し、次のエラー・メッセージを呼び出します。メッセージが呼び出され、これをアクセプトすると、レポートは引き続き実行されます。

REP-1423: 問合せの実行開始後は、その最大行を変更できません。

構文

```
SRW.MAXROW_UNSET;
```

使用上の注意

Report Builderでは、問合せのレコードがフェッチされた後でSRW.SET_MAXROWパッケージ・プロシージャが使用された場合、この例外を呼び出します。

SRW.MAXROW_UNSETの例

```
/* Suppose you want your own error message raised,
** instead of the default error message.
** You could handle this exception in the following way:
*/

EXCEPTION
when SRW.MAXROW_UNSET then
  srw.message(1000, 'Data was fetched before
    SRW.SET_MAXROW was called.');
```

SRW.MESSAGE

説明

このプロシージャは、指定するメッセージ番号とテキストを持つメッセージを表示します。メッセージは、次の書式で表示されます。メッセージが呼び出され、これをアクセプトすると、レポートは引き続き実行されます。

MSG-msg_number: msg_text.

構文

```
SRW.MESSAGE (msg_number NUMBER, msg_text CHAR);
```

パラメータ

msg_number	メッセージ・ラインに表示される1~10桁の番号。番号が5桁に満たないときは、5桁になるように0で埋められます。たとえば、123を指定すると、SRW-00123のように表示されます。
msg_text	メッセージ・ラインに表示される180バイト以下の文字。

SRW.MESSAGEの制限

- Report Builderのエラー・メッセージの検出や変更はできません。
- SRW.MESSAGE@命令は、レポートの実行を終了できません。メッセージを呼び出してからレポートを終了する場合は、SRW.PROGRAM_ABORTを使用します。
- メッセージ文字列中に余分な空白があると、メッセージに表示されます。余分な空白は、Report Builderでは削除されません。

SRW.MESSAGEの例

```

/* Suppose you have a user exit named MYEXIT to which you want to
** pass the values of the SAL column. Suppose, also, that you want
** to raise your own error if the user exit is not found (e.g., because
** it is not linked, compiled, etc.). To do these things, you could
** write the following PL/SQL in the Format Trigger of the F_SAL field:
*/
/* This trigger will raise your message as follows:
** MSG-1000: User exit MYEXIT failed. Call Karen Smith x3455.
*/

FUNCTION FOO RETURN BOOLEAN IS
BEGIN
    srw.reference(:SAL);
    srw.user_exit('myexit sal');
EXCEPTION
    when srw.unknown_user_exit then
        srw.message(1000, 'User exit MYEXIT failed.
Call Karen Smith x3455. ');
        raise srw.program_abort;
RETURN (TRUE);
END;
```

SRW.NULL_ARGUMENTS

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1418: <SRW.DO_SQLまたはSRW.MESSAGE、SRW.RUN_REPORT、SRW.SET_MAXROW、SRW.USER_EXIT>にNULL引数が渡されました。

構文

```
SRW.NULL_ARGUMENTS;
```

使用上の注意

Report Builderでは、欠落引数でパッケージ・ファンクションまたはパッケージ・プロシージャが使用された場合、この例外を呼び出します。引数値をパッケージ・ファンクションまたはパッケージ・プロシージャに渡すときや、渡された値がNULLでないことを確認するときに便利です（式列がSRW.USER_EXITをコールし、ユーザー・イグジット文字列が PL/SQLライブラリから渡された場合など）。

次の場合、この例外が呼び出される可能性があります。

- SRW.DO_SQL
- SRW.MESSAGE
- SRW.RUN_REPORT;
- SRW.SET_MAXROW
- SRW.USER_EXIT;

SRW.NULL_ARGUMENTSの例

```
/* Suppose you want your own error message raised,  
** instead of the default error message.  
** You could handle this exception in the following way:  
*/  
  
EXCEPTION  
when SRW.NULL_ARGUMENTS then  
    srw.message(4000, 'Contact Application Development  

```

SRW.PROGRAM_ABORT

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1419: PL/SQLプログラムが異常終了しました。

SRW.PROGRAM_ABORTを呼び出すと、レポートの実行が停止します。

構文

```
SRW.PROGRAM_ABORT;
```

使用上の注意

この例外は、PL/SQL内から呼び出す必要があります。

SRW.PROGRAM_ABORTの例

```

/* Suppose you want to put a border around the salary if it is greater
than 0.
** Suppose, also, that if the report fetches a salary less than 0, you
want to
** raise a customized error message (i.e., "FOUND A NEGATIVE SALARY..."),
** then terminate the report execution.To do so, you could write the
** following format trigger for F_SAL.
*/

FUNCTION foo return boolean is
BEGIN
if :sal >= 0 then
srw.attr.mask      := SRW.BORDERWIDTH_ATTR;
srw.attr.borderwidth := 1;
srw.set_attr (0, srw.attr);
else
srw.message(100, 'FOUND A NEGATIVE SALARY.
CHECK THE EMP TABLE. ');
raise srw.program_abort;
end if;
RETURN (TRUE);
END;
```

SRW.REFERENCE

説明

このプロシージャにより、ReportBuilderでは、参照オブジェクトをPL/SQL構成体の依存性リストに追加します。これによってReport Builderは、PL/SQL構成体を起動する直前に、参照オブジェクトの値を判断します。ユーザー・イグジットに渡された列値が、最後に計算またはフェッチされた値であることを確認するときに便利です。

構文

```
SRW.REFERENCE (:object CHAR|DATE|NUMBER);
```

パラメータ

object	Report Builderのパラメータあるいは列。PL/SQL構成体を起動させる前に、値を確認する必要があります。
--------	--

SRW.REFERENCEの制限

- オブジェクト名の前には、コロンが必要です。
- SRW.REFERENCEは、オブジェクトがすでに現行のPL/SQL構造体内で参照されている場合には不要です。

SRW.REFERENCEの例

```
/* Suppose you want to pass the temperature and pressure values
** to a user exit called SUPERHEAT. Suppose, also, that if the
** temperature is too low, you want to raise a customized error message.
** To do so, you could write the following formula:
*/

FUNCTION EXIT RETURN BOOLEAN IS
BEGIN
  if :temp > 0 then
    srw.reference (:temp); -- unnecessary reference
    srw.reference (:pressure);
    srw.user_exit('superheat temp pressure');
  else srw.message(1000, 'Temperature is below
    normal.Is machine off?');
    raise srw.program_abort;
  end if;
RETURN(TRUE);
END;
```

SRW.RUN_REPORT

説明

このプロシージャは、指定する文字列によってRWRUN60を起動します。このプロシージャは次の場合に便利です。

- ドリルダウン・レポートを実行する（つまり、ボタンのアクション・トリガーからレポートをコールすること）。
- レポートの一部を異なる受信者に送る（e-mailによって、各マネージャに、担当グループのデータのみを付けて、レポートを送る場合など）。
- レポートの一部を異なるプリンタに送る（各マネージャのレポートを、各自のプリンタに送る場合など）。
- 1つのレポートから複数のレポートを実行する。

SRW.RUN_REPORTは指定したRWRUN60コマンドを実行します。

構文

```
SRW.RUN_REPORT (command_line CHAR);
```

パラメータ

command_line	有効なRWRUN60コマンド。
--------------	-----------------

SRW.RUN_REPORTの制限

- ランタイム・パラメータ・フォームに入力したパラメータ値をRWRUN60文字列に渡す場合は、フォーム前トリガーの後でSRW.RUN_REPORTをコールする必要があります。
- このプロシージャに指定または渡される文字列は、ご使用のオペレーティング・システムの構文と大小文字の区別のルールに従ってください。
- SRW.RUN_REPORTプロシージャに対しては、ユーザーIDを指定しません。ユーザーIDは、「コール側」レポートによって継承されます。
- BDESTYPEには、FileまたはPrinter、Sysout、Mailを指定することができます。
- SRW.RUN_REPORTがPL/SQLでボタンに使用されている場合、そのボタンが選択されても、ランタイム・パラメータ・フォームはデフォルトでは表示されません。ランタイム・パラメータ・フォームを表示する場合は、SRW.RUN_REPORTの呼出しでPARAFORM=YESを指定する必要があります。
- パスを指定しないと、Report Builderではファイル・パス検索順序 を使用して、レポートを検索します。

SRW.RUN_REPORTの例

```
/* Suppose you have the following two reports:
** MGR_RUN, which queries manager names, and invokes a second report named
MAIL_IT
** MAIL_IT, which queries employee names for the manager that MGR_RUN passes
it,
** and sends the report output to the manager via e-mail.
** The description of MGR_RUN could be as follows:
** Query:
SELECT ENAME, EMPNO FROM EMP WHERE JOB='MANAGER'
** Group Filter:
*/

FUNCTION FOO RETURN BOOLEAN IS
BEGIN
  srw.run_report('report=MAIL_IT
  desname='||:ename ||' desformat=dflt batch=yes
```

```
        mgr_no='|| TO_CHAR(:empno) );
RETURN(TRUE);
EXCEPTION
when srw.run_report_failure then
    srw.message(30, 'Error mailing reports. ');
    raise srw.program_abort;
END;
/* This PL/SQL invokes MAIL_IT, specifies that MAIL_IT's output
** should be sent to the manager via Oracle Mail, and passes the
** manager number, so that the MAIL_IT report can query only the
** manager's employees.
** Note:EMPNO's values must be converted to characters
** (TO_CHAR in the PL/SQL above), because SRW.RUN_REPORT
** requires a character string.
** Layout:None is needed, because this report only fetches data,
** then passes it to a second report.
** The description of MAIL_IT could be as follows:
** Query:
    SELECT DEPTNO, ENAME, SAL FROM EMP WHERE MGR=:MGR_NO
** Layout:Master/Detail
*/

/* Suppose that you have three reports that you almost always run together.
** The reports are named SALARY, COMMISS, and TAXES.To run these reports
** with one RWRUN60 command, you create a driver report named PAYROLL.
** The description of PAYROLL could be as follows:
** Query:
    SELECT DEPTNO FROM DEPT
** Before Report Trigger:
*/

FUNCTION FOO RETURN BOOLEAN IS
BEGIN
    srw.run_report('batch=yes report=SALARY
        destype=file desformat=dflt desname=salary.lis');
    srw.run_report('batch=yes report=COMMISS
        destype=file desformat=dflt desname=comiss.lis');
    srw.run_report('batch=yes report=TAXES
        destype=file desformat=dflt desname=comiss.lis');
RETURN(TRUE);
END;
/* Layout:  Tabular
** When you run PAYROLL from the designer or RWRUN60, the other three
```

```
** reports will all be run. (Note that, in this case, the query and  
** the layout for Payroll could be anything. They are only used here  
** in order to make it possible to run PAYROLL.)  
*/
```

SRW.RUN_REPORT_BATCHNO

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

```
REP-1429: <program unit name>: batch=noでは<procedure name>を起動できません。
```

構文

```
SRW.RUN_REPORT_BATCHNO;
```

使用上の注意

Report Builderでは、SRW.RUN_REPORTプロシージャがBATCH=NOを含むコマンド・ライン文字列によってコールされた場合、この例外を呼び出します。

SRW.RUN_REPORT_BATCHNOの例

```
/* Suppose you want your own error message raised,  
** instead of the default error message (above).  
** You could handle this exception in the following way:  
*/
```

```
EXCEPTION  
when SRW.RUN_REPORT_BATCHNO then  
  srw.message(4000, 'Contact Application Dev.Services  
  regarding SRW.RUN_REPORT_BATCHNO.');
```

```
raise srw.program_abort;
```

SRW.RUN_REPORT_FAILURE

説明

SRW.RUN_REPORTパッケージ・プロシージャが失敗した場合、この例外はレポートの実行を停止し、次のエラー・メッセージを呼び出します。

```
REP-1428: SRW.RUN_REPORTを実行中にエラーが発生しました。
```

構文

```
SRW.RUN_REPORT_FAILURE;
```

使用上の注意

SRW.RUN_REPORTパッケージ・プロシージャが失敗すると、Report Builderではこの例外を呼び出します。

SRW.RUN_REPORT_FAILUREの例

```
/* Suppose you want your own error message raised, instead of the default
** error message (above).The following PL/SQL code raises a customized
** error message when SRW.RUN_REPORT_FAILURE is raised.
*/

EXCEPTION
  WHEN srw.run_report_failure then
    srw.message(30, 'Error mailing reports. ');
    raise srw.program_abort;
```

SRW.SET_ATTR

新機能：

Set Attributes Proceduresを使った属性の設定が、より便利になりました。互換性を保つため、まだSRW.SET_ATTRもサポートされています。しかし、この新しく簡単になったプロシージャの使用を強くお勧めします。

説明

SRW.SET_ATTRは、レイアウト・オブジェクトにフォントまたはサイズ、カラーなどの属性を設定します。SRW.SET_ATTRを使ってフォーマット属性を指定する手順は、次のとおりです：

- 1 SRW.ATTR.MASKを設定して、設定する属性（境界線など）を指定します。
- 2 1で指定した各属性（1文字の幅など）の値を指定します。
- 3 SRW.SET_ATTRプロシージャをコールして、オブジェクトにマスクを適用します。

注意：ほとんどの場合、オブジェクトのプロパティ・パレットでWebリンク(HTMLかPDF)とHTML属性を定義できます。条件付きの設定など、さらに複雑なインプリメントが必要な場合は、SRW.SET_ATTRを使用する必要があります。

- Webリンクを指定するには、ACTIONおよびBOOKMARK、HYPERLINK、LINKTAG属性を使用します。

- HTML文書およびページ・ヘッダーとフッターを定義するには、BEFPAGE_ESCAPEおよびAFTPAGE_ESCAPE①ABEFREPORT_ESCAPE②AAFTREPORT_ESCAPE属性を使用します。
- HTMLパラメータ・フォームのヘッダーとフッターを定義するには、BEFFORM_ESCAPEおよびAFTFORM_ESCAPEを使用します。

構文

```
SRW.ATTR.MASK := SRW.attr_ATTR [+ SRW.attr_ATTR.. .];
SRW.ATTR.attr := value; [SRW.ATTR.attr := value;.. .]
SRW.SET_ATTR (object_id, SRW.ATTR);
```

パラメータ

attr	設定する属性の名前(たとえば、BORDERWIDTH)。
object_id	またはSRW.REPORT_ID、BEFREPORT_ESCAPEおよびAFTREPORT_ESCAPE、BEFPAGE_ESCAPE、AFTPAGE_ESCAPEの値を設定する場合は、SRW.REPORT_IDです。その他の場合はすべて0です。
value	属性の有効値(たとえば、1はBORDERWIDTHの有効値になります)。

SRW.SET_ATTRの制限

- 属性 (SRW.FILLPATT_ATTRなど) を指定する場合は、その値を指定する必要があります。指定しないと、その属性は予想できないものになります。
- Report Builderパッケージにない値を指定する場合は、文字列 (一重引用符で囲まれたもの) にしてください。たとえば、SRW.ATTR.FORMATMASKの値は、任意の値を指定できるので、引用符で囲む必要があります。SRW.ATTR.HJUSTの値は、Report Builderにある値から指定する必要があるため、引用符で囲まないでください。
- SRW.SET_ATTRを使用しないと、フィールドにテキストの属性を設定することはできません。SRW.SET_ATTRは、ポイラプレート・テキストには使用できません。ポイラプレート・テキストのテキスト属性を変更するには、レイアウト・エディタの「書式」メニューまたはツール・パレットを使用します。

SRW.SET_ATTRキャラクタ・モードの例

```
/* Here's an example of using all of the formatting
** attributes valid for character-mode reports.
** Notice that there are two values applied to the
** horizontal justification attribute (HJUST).
** Note:For the printer codes &1 and &2 to be
** meaningful, the printer definition file for
** this report must define what the printer should
** do for &1 and &2.
**/

function F_SALFormatTrigger return boolean is
begin
```

```
IF :SAL > 2000 THEN
  SRW.ATTR.MASK      := SRW.BEFCODE_ATTR  +
                      SRW.AFTCODE_ATTR  +
                      RW.TEXT_ATTR      +
                      RW.FILLPATT_ATTR  +
                      RW.BORDERWIDTH_ATTR +
                      RW.FORMATMASK_ATTR +
                      RW.HJUST_ATTR;

  SRW.ATTR.BEFCODE   := '&1';
  SRW.ATTR.AFTCODE   := '&2';
  SRW.ATTR.TEXT      := SRW.BOLD_TEXTA;
  SRW.ATTR.FILLPATT  := 'TRANSPARENT';
  SRW.ATTR.BORDERWIDTH := 1;
  SRW.ATTR.FORMATMASK := 'DD-MON-YY';
  SRW.ATTR.HJUST     := SRW.CENTER_HJUST +
                      SRW.FLUSH_HJUST;

END IF;
SRW.SET_ATTR(0, SRW.ATTR);
RETURN(TRUE);
end;
```

SRW.SET_ATTR書式マスクの例

```
/* If the salary is 2000, the following code
** segment sets the format mask to -99990.
*/
function F_SALFormatTrigger return boolean is
begin
  if :sal = 2000 then
    srw.attr.mask      := SRW.FORMATMASK_ATTR;
    srw.attr.formatmask := '-99990';
    srw.set_attr (0, srw.attr);
  end if;
  RETURN(TRUE);
end;
```

SRW.SET_ATTRフィールド・カラー設定の例

```
/* If the salary is 2000, this function sets the
** following attributes: border width to 1, foreground
** border color to "red," background border color
** to "blue," border pattern to "checker," foreground
** fill color to "yellow," background fill color to
** "green," and the fill pattern to "crisscross."
**
```

```

** Note: When you run the report with this PL/SQL
** construct in character mode, only the border width
** (of 1) and fill pattern (of solid) will be set;
** all other settings will be ignored. When you run
** the report with this PL/SQL construct in bitmap,
** all of the attributes will be set. (Be aware that
** the border background color cannot be set on some
** GUIs.)
*/
function F_SALFormatTrigger return boolean is
begin
  if :sal = 2000 then
    srw.attr.mask      := SRW.BORDERWIDTH_ATTR +
                        SRW.FBCOLOR_ATTR      +
                        SRW.BBCOLOR_ATTR      +
                        SRW.BORDPATT_ATTR     +
                        SRW.FFCOLOR_ATTR      +
                        SRW.BFCOLOR_ATTR      +
                        SRW.FILLPATT_ATTR;

    srw.attr.borderwidth := 1;
    srw.attr.fbcolor     := 'red';
    srw.attr.bbcolor     := 'blue';
    srw.attr.bordpatt    := 'checker';
    srw.attr.ffcolor     := 'yellow';
    srw.attr.bfcolor     := 'green';
    srw.attr.fillpatt    := 'crisscross';
    srw.set_attr (0, srw.attr);
  end if;
  RETURN(TRUE);
end;

```

SRW.SET_ATTRフィールド・テキスト属性設定の例

```

/* If the salary is 2000, this function
** sets the following text attributes:text
** to bold, font to TIMES, size to 18 points,
** style to underline, weight to bold, text color
** to "blue," and the justification to center.
**
** Note:When you run the report with this PL/SQL
** construct in character mode, only the text will
** be set to bold (SRW.TEXT_ATTR); all other
** attributes will be ignored. When you run the
** report with this PL/SQL construct in bitmap,
** all of the attributes--except SRW.TEXT_ATTR--will
** be set.
*/

```

```
function F_SALFormatTrigger return boolean is
begin
  if :sal = 2000 then
    srw.attr.mask      := SRW.TEXT_ATTR      +
                        SRW.FACE_ATTR       +
                        SRW.SZ_ATTR         +
                        SRW.WEIGHT_ATTR     +
                        SRW.STYLE_ATTR      +
                        SRW.GCOLOR_ATTR     +
                        SRW.HJUST_ATTR;
    srw.attr.text      := SRW.BOLD_TEXTA;
    srw.attr.face      := 'times';
    srw.attr.sz        := 18;
    srw.attr.weight    := SRW.BOLD_WEIGHT;
    srw.attr.style     := SRW.UNDERLINE_STYLE;
    srw.attr.gcolor    := 'blue';
    srw.attr.hjust     := SRW.CENTER_HJUST;
    srw.set_attr (0, srw.attr);
  end if;
  RETURN(TRUE);
end;
```

SRW.SET_ATTR垂直間隔設定の例

```
/* If the salary is 2000, this function sets
** the vertical spacing between lines to a custom
** size of 200 VGS units. (One inch is equivalent
** to 8192 VGS units.)
*/
```

```
function F_SALFormatTrigger return boolean is
begin
  if :sal = 2000 then
    srw.attr.mask      := SRW.GSPACING_ATTR;
    srw.attr.gspacing  := SRW.CUSTOM_SPACING;
    srw.attr.custom    := 200;
    srw.set_attr (0, srw.attr);
  end if;
  RETURN(TRUE);
end;
```

SRW.SET_ATTRの例 (プリンタ・トレイの切り替え)

```
/* The example below sets the printer tray in a Between Pages trigger.
```

```

*/
function BetweenPages return boolean is
begin
    srw.attr.mask           := SRW.PRINTER_INTRAY_ATTR;
    srw.attr.printer_intray := 'letterhead';
    srw.set_attr(SRW.REPORT_ID, srw.attr);
    return (TRUE);
end;

```

SRW.TEXT_ESCAPE

HTMLがSRW.xxxx_ESCAPE_VALUE(xxxxには、BEFREPORTなどのエスケープの名前が入ります) で指定されていることを示します。SRW.TEXT_ESCAPEを指定する場合は、SRW.xxxx_ESCAPE_VALUEに引用符付き文字列を入力する必要があります。

SRW.TRACE_ADD_OPTION

説明

SRW.TRACE_ADD_OPTIONを使用すると、SRW.TRACE_STARTによるトレース情報のロギング開始後、新規のトレース・オプションの指定が可能になります。

次の例では、マスクに3つのトレース・オプションを指定してから、SRW.TRACE_ADD_OPTIONのコールの中でそのマスクを起動します。

```

SRW.TRACEOPTS.MASK := SRW.TRACE_ERR +
                      SRW.TRACE_BRK +
                      SRW.TRACE_PRF;
SRW.TRACE_ADD_OPTION(SRW.TRACEOPTS);

```

構文

```
SRW.TRACE_ADD_OPTION (SRW.TRACEOPTS);
```

パラメータ

SRW.TRACE_OPTS	SRW.TRACEOPTS.MASKで前に定義したトレース・オプションのマスクを適用します。SRW.TRACE_APPENDとSRW.TRACE_REPLACEは、トレース・オプションではありません。SRW.TRACE_ADD_OPTIONと一緒に使用できないことに注意してください。
----------------	--

SRW.TRACE_END

説明

SRW.TRACE_ENDは、レポートの実行中に、トレース情報のファイルへのログギングを終了します。ログギングを開始するには、SRW.TRACE_STARTを使用します。

構文:

```
SRW.TRACE_END ();
```

SRW.TRACE_REM_OPTION

説明

SRW.TRACE_REM_OPTIONは、トレース・オプションを削除します。

次の例では、マスクに3つのトレース・オプションを指定してから、SRW.TRACE_REM_OPTIONのコールの中でそのマスクを起動します。

```
SRW.TRACEOPTS.MASK := SRW.TRACE_ERR +
                      SRW.TRACE_BRK +
                      SRW.TRACE_PRF;
SRW.TRACE_REM_OPTION(SRW.TRACEOPTS);
```

構文

```
SRW.TRACE_REM_OPTION (SRW.TRACEOPTS);
```

パラメータ

SRW.TRACE_OPTS	SRW.TRACEOPTS.MASKによって前に定義したマスクにおけるトレース・オプションを削除します。SRW.TRACE_APPENDとSRW.TRACE_REPLACEは、トレース・オプションではありません。SRW.TRACE_REM_OPTIONと一緒に使用しないことに注意してください。
----------------	---

SRW.TRACE_START

説明

SRW.TRACE_STARTにより、レポートの実行中に、トレース情報のファイルへのログギングを開始することができます。ログギングを停止するには、SRW.TRACE_ENDを使用します。ログ・ファイルにどの情報を入れるかを指定することもできます。この情報は、レポートのデバッグやパフォーマンスのボトルネックを見つけ出すのに役立つ場合があります。マスクを定義して、どのトレース・オプションを適用するかを指定します。PL/SQLでトレース・オプションを設定するには、次の作業を行います。

- 1 起動させたいオプションを指定します。
- 2 SRW.TRACE_STARTプロシージャをコールするときに、SRW.TRACEOPTSを使用してオプションをオブジェクトに適用します。

次の例では、3つのトレース・オプションを指定してから、SRW.TRACE_STARTのコールでそれらのオプションを起動します。

```
SRW.TRACEOPTS.MASK := SRW.TRACE_ERR +
                      SRW.TRACE_BRK +
                      SRW.TRACE_PERF;
SRW.TRACE_START(test.dat, SRW.TRACE_APPEND, SRW.TRACEOPTS);
```

構文

```
SRW.TRACEOPTS.MASK := SRW.TRACE_opts [+ SRW.TRACE_opts . . .];
SRW.TRACE_START (filename, {SRW.TRACE_REPLACE|SRW.TRACE_APPEND},
SRW.TRACEOPTS);
```

パラメータ

filename	Report Builderがログ情報を格納するファイルの名前。
SRW.TRACE_APPEND	新しい情報を現在のファイルの内容と置き換えます。
SRW.TRACE_REPLACE	新しい情報を現在のファイルの内容と置き換えます。
SRW.TRACE_OPTS	SRW.TRACEOPTS.MASKで前に定義したトレース・オプションのマスクを適用します。
Opts	有効トレース・オプション。

SRW.TRACEOPTS.MASK

トレース・マスクに適用可能なオプションとその意味は、次のとおりです。

マスク・オプション	説明
SRW.TRACE_ALL	ログ・ファイルに可能な限りのトレース情報を取得します。
SRW.TRACE_APP	ログ・ファイルにすべてのレポート・オブジェクトのトレース情報を取得します。
SRW.TRACE_BRK	ログ・ファイルにブレークポイントをリストします。
SRW.TRACE_DST	ログ・ファイルに配布リストをリストします。この情報を使って、どのセクションがどの宛先に送られたかを調べることができます。
SRW.TRACE_ERR	ログ・ファイルにエラー・メッセージと警告を取得します。
SRW.TRACE_PLS	ログ・ファイルにすべてのPL/SQLオブジェクトのトレース情報を取得します。
SRW.TRACE_PERF	ログ・ファイルにパフォーマンスの統計を取得します。
SRW.TRACE_SQL	ログ・ファイルにすべてのSQLのトレース情報を取得します。

SRW.TRACE_STARTの制限

- ログ情報は、*rdf*ファイルの実行時にのみ生成されます。repファイルの実行時に、ロギングの指定はできません。
- 一度に指定できるのは1つのログ・セッションのみです。SRW.TRUNCATED_VALUEを実行すると、自動的に他のログ・セッションは終了されます。

SRW.TRUNCATED_VALUE

説明

この例外は、レポートの実行を停止し、2つのエラー・メッセージのうちのどちらかを呼び出します。エラーがPL/SQLの割当てが原因で発生した場合、次のメッセージが表示されます。

REP-1435: 列<parameter/column name>の値が切り捨てられました。

エラーがユーザー・イグジットIAF PUTが原因で発生した場合、次のメッセージが表示されます。

REP-1416: ユーザー・イグジット<name> IAF PUT:列<parameter/column name>の値が切り捨てられました。

SRW.TRUNCATED_VALUE例外が処理される場合、その値は切り捨てられ、パラメータまたは列に割り当てられたままになります。

構文

```
SRW.TRUNCATED_VALUE;
```

使用上の注意

ユーザー・イグジットまたはPL/SQL構成体が、オブジェクトの最大幅より大きい値をパラメータまたは列に割り当てようとすると、Reportsはこの例外を呼び出します。

SRW.TRUNCATED_VALUEの例

```
/* Suppose you want your own error message raised, instead of the
** default error message (above). You could handle this exception
** in the following way:
*/
```

```
EXCEPTION
when SRW.TRUNCATED_VALUE then
srw.message(2000, 'Contact Dan Brown:
           SRW.TRUNCATED_VALUE. ');
```

SRW.UNKNOWN_QUERY

説明

この例外は、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1427: SRW.SET_MAXROWで参照されている問合せ名が存在しません。

構文

```
SRW.UNKNOWN_QUERY;
```

使用上の注意

SRW.SET_MAXROWパッケージ・プロシージャが不明な問合せでコールされた場合に、Reportsではこの例外を呼び出します。

SRW.UNKNOWN_QUERYの例

```
/* Suppose you want your own error message raised,  
** instead of the default error message (above).  
** You could handle this exception in the following way:  
*/
```

```
EXCEPTION  
when SRW.UNKNOWN_QUERY then  
    srw.message(4000, 'Contact Per Jensen:  
    SRW.UNKNOWN_QUERY.');
```

```
    raise srw.program_abort;
```

SRW.UNKNOWN_USER_EXIT

説明

ユーザー・イグジットが不明な場合 (リンクされていないなどの理由で)、この例外はレポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1415: ユーザー・イグジットは不明です。

構文

```
SRW.UNKNOWN_USER_EXIT;
```

使用上の注意

SRW.USER_EXITまたはSRW.USER_EXIT20パッケージ・プロシージャに指定されたユーザー・イグジットが見つからない場合(ユーザー・イグジットがリンクされていなかったなどの理由で)、Reportsではこの例外を呼び出します。

SRW.UNKNOWN_USER_EXITの例

Suppose you want your own error message raised, instead of the default error message (above).The following PL/SQL will raise a customized error message for SRW.UNKNOWN_USER_EXIT.

```
EXCEPTION
  WHEN srw.unknown_user_exit then
    srw.message(200, 'PASS USER EXIT WAS UNKNOWN.
      CHECK IF IT'S LINKED.');
```

SRW.USER_EXIT

説明

このプロシージャは、user_exit_stringで指定したユーザー・イグジットをコールします。レポートの実行中に制御を3GLプログラムに渡すときに便利です。

構文

```
SRW.USER_EXIT (user_exit_string CHAR);
```

パラメータ

user_exit_string	コールするユーザー・イグジットの名前と、ユーザー・イグジット・プログラムに渡す任意の列またはパラメータです。
------------------	--

SRW.USER_EXITの制限

- ユーザー・イグジットは、移植性がありません。レポートに移植性を持たせ、条件論理を追加する必要がある場合は、PL/SQLを使用します。
- ユーザー・イグジット文字列により列またはパラメータをユーザー・イグジット・プログラムに渡す場合は、このプロシージャの前にSRW.REFERENCEをコールする必要があります。

SRW.USER_EXITの例

```
/* Suppose you have a user exit named STORE to which you want
** to pass salary values from Report Builder.To do so, you
** could write the following formula.For more information on
** how to call user exits, see Calling a user exit.
*/
```

```
FUNCTION FOO RETURN BOOLEAN IS
BEGIN
IF :SAL >= 0 THEN
    SRW.REFERENCE(:SAL);
    SRW.USER_EXIT('STORE SAL');
ELSE
SRW.MESSAGE(100, 'FOUND A NEGATIVE SALARY.CHECK THE
    EMP TABLE. ');
END IF;
EXCEPTION
    WHEN SRW.UNKNOWN_USER_EXIT THEN
SRW.MESSAGE(200, 'STORE USER EXIT WAS UNKNOWN.
    CHECK IF IT'S LINKED. ');
    WHEN SRW.USER_EXIT_FAILURE THEN
SRW.MESSAGE(200, 'STORE USER EXIT FAILED.
    CHECK ITS CODE. ');
RETURN(TRUE);
END;
```

SRW.USER_EXIT20

説明

このプロシージャは、コールするユーザー・イグジットの引数が2つではなく5つであるという点を除いて、SRW.USER_EXITと同じです。これにより、2つではなく5つの引数を渡すOracle Formsなどの他のOracle製品と、ユーザー・イグジットを共有することができます。

ユーザー・イグジットをコールするとき、SRW.USER_EXITによって次のものが渡されます。

- 指定した文字列
- 文字列の長さに対するポインタ

ユーザー・イグジットをコールするとき、SRW.USER_EXIT20によって次のものが渡されます。

- 指定した文字列
- 文字列の長さに対するポインタ
- エラー文字列（この引数は、Report Builderでは使用されず、常に' 'の値を渡します）。
- エラー文字列の長さに対するポインタ（この引数は、Report Builderでは使用されず、常に0の値を渡します）。

- 問合せ内引数（この引数は、Report Builderでは使用されず、常に0の値を渡します）。

構文

```
SRW.USER_EXIT20 (user_exit_string CHAR);
```

パラメータ

user_exit_string	コールするユーザー・イグジットの名前と、ユーザー・イグジット・プログラムに渡す任意の列またはパラメータです。
------------------	--

SRW.USER_EXIT_FAILURE

説明

コールしたユーザー・イグジットが失敗したとき、この例外が呼び出されます。この例外が呼び出されると、レポートの実行を停止し、次のエラー・メッセージを呼び出します。

REP-1416: ユーザー・イグジット: <exit_name>が失敗しました。

構文

```
SRW.USER_EXIT_FAILURE;
```

使用上の注意

SRWERBバッファが空でないとき、Reportsではこの例外を呼び出します。

SRW.USER_EXIT_FAILUREの例

```
/* Suppose you want your own error message raised, instead of the
** default error message (above).The following PL/SQL code will
** raise a customized error message for SRW.USER_EXIT_FAILURE.
*/
```

```
EXCEPTION
  when SRW.USER_EXIT_FAILURE then
    SRW.MESSAGE(200, 'PASS user exit failed.
    Check its code.');
```

設定属性プロシージャ

SRW.SET_AFTER_FORM_HTML
SRW.SET_AFTER_PAGE_HTML
SRW.SET_AFTER_PRINTING_CODE
SRW.SET_AFTER_REPORT_HTML
SRW.SET_ATTR 属性
SRW.SET_BACKGROUND_BORDER_COLOR
SRW.SET_BACKGROUND_FILL_COLOR
SRW.SET_BEFORE_FORM_HTML
SRW.SET_BEFORE_PAGE_HTML
SRW.SET_BEFORE_PRINTING_CODE
SRW.SET_BEFORE_REPORT_HTML
SRW.SET_BOOKMARK
SRW.SET_BORDER_PATTERN
SRW.SET_BORDER_WIDTH
SRW.SET_CHARMODE_TEXT
SRW.SET_CUSTOM_SPACING
SRW.SET_DISPLAY_NAME
SRW.SET_FIELD
SRW.SET_FIELD_CHAR
SRW.SET_FIELD_DATE
SRW.SET_FIELD_NUM
SRW.SET_FILL_PATTERN
SRW.SET_FONT_FACE
SRW.SET_FONT_SIZE

SRW.SET_FONT_STYLE
SRW.SET_FONT_WEIGHT
SRW.SET_FOREGROUND_BORDER_COLOR
SRW.SET_FOREGROUND_FILL_COLOR
SRW.SET_FORMAT_MASK
SRW.SET_HYPERLINK
SRW.SET_HYPERLINK_ATTRS
SRW.SET_JUSTIFICATION
SRW.SET_LINKTAG
SRW.SET_MAXROW
SRW.SET_PAGE_NAVIGATION_HTML
SRW.SET_PDF_ACTION
SRW.SET_PRINTER_TRAY
SRW.SET_SPACING
SRW.SET_TEXT_COLOR

SRW.SET_AFTER_FORM_HTML

説明

このプロシージャは、HTMLパラメータ・フォームの下端に表示するテキスト、グラフィックまたはHTMLコマンドを挿入します。この属性は、ロゴまたは標準リンクをパラメータ・フォーム上に配置するのに便利です。

構文

```
SRW.SET_AFTER_FORM_HTML(type, 'string');
```

パラメータ

type	SRW.FILE_ESCAPE またはSRW.TEXT_ESCAPE。文字列パラメータがファイル名か挿入されるテキストかを示します。
文字列	typeパラメータの指定内容に依存した、ファイル名または挿入されるテキスト。

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「フォーム後型」および「フォーム後値」プロパティを設定します。

SRW.SET_AFTER_FORM_HTMLの例

```

/* This example inserts two links to home pages
** at the bottom of the HTML Parameter Form.
*/

function BeforeForm return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.SET_AFTER_FORM_HTML(SRW.TEXT_ESCAPE, '<CENTER>' ||
        '<A HREF="http://www.oracle.com/">' ||
        'Oracle Corporation</A> -' ||
        '<BR>' ||
        '<A HREF="http://home.netscape.com/">' ||
        'Netscape</A> </CENTER>' ||
        '</BODY> </HTML>');
end if;
return (TRUE);
end;

```

SRW.SET_AFTER_PAGE_HTML

説明

このプロシージャは、文書のページの下端に表示するテキスト、グラフィックまたはHTMLコマンドを挿入します。この属性は、ロゴまたは標準リンクをHTML文書の各ページの先頭に配置するのに便利です。

HTMLページ・ストリームを使用する場合、このプロシージャはレポート出力のすべてのページに適用されます。レポートの最初（ヘッダー）または最後（フッター）のページにのみHTMLを指定するには、SRW.SET_BEFORE_REPORT_HTML または SRW.SET_AFTER_REPORT_HTML PL/SQLプロシージャをそれぞれ使用します。

構文

```
SRW.SET_AFTER_PAGE_HTML(type, 'string');
```

パラメータ

type	SRW.FILE_ESCAPE またはSRW.TEXT_ESCAPE。文字列パラメータがファイル名が挿入されるテキストかを示します。
string	typeパラメータの指定内容に依存した、ファイル名または挿入されるテキスト。

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「ページ後型」および「ページ後値」プロパティを設定します。

SRW.SET_AFTER_PAGE_HTMLの制限

- ページの下端に含まれるデフォルトHTMLを以下に示します。ページ間に表示するデフォルトのセパレータ・ラインが不要な場合は、これは不要です。

```
<hr size=5 noshade>
```

- SRW.SET_AFTER_PAGE_HTMLをレポートのすべてのページに適用する場合、Before Report トリガーなど、レポートのフォーマットが始まる前に起動するトリガーに指定する必要があります。
- SRW.SET_AFTER_PAGE_HTMLを現行ページのみ適用する場合、そのページのオブジェクトのフォーマット・トリガーに指定する必要があります。

SRW.SET_AFTER_PAGE_HTMLの例

```
/* The example below inserts a GIF file as a
** page separator.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.SET_AFTER_PAGE_HTML(SRW.TEXT_ESCAPE, '<CENTER>' ||
        '<IMG ALT="Ruler" SRC="line.gif" VSPACE=10>' ||
        '<BR></CENTER>');
end if;
return (TRUE);
end;
```

SRW.SET_AFTER_PRINTING_CODE

説明

このプロシージャは、オブジェクトの各行の後に実行されるプリンタ・エスケープ・シーケンスへの参照を挿入します。オブジェクトの印刷がトリガーされた後、実際に印刷される前に、プリンタ・エスケープ・シーケンスが挿入されます。プリンタ・コードは、キャラクタ・モードでレポートを実行しているときのみ使用します。

構文

```
SRW.SET_AFTER_PRINTING_CODE('code');
```

パラメータ

code	DESFORMAT用に指定された.prtファイルに定義された有効なプリンタ・コード。
------	--

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「後のプリンタ・コード」プロパティを設定します。

SRW.SET_AFTER_REPORT_HTML

説明

このプロシージャは、HTML文書の終わりに表示するテキスト、グラフィックまたはHTMLコマンドを挿入します。この属性は、ロゴまたは標準リンクをHTML文書の各ページの先頭に配置するのに便利です。

HTMLページ・ストリームを使用する場合、このプロシージャはレポート出力の最初のページに適用されます。HTMLがレポート全体(バックグラウンド・カラーやイメージ、その他の <body> HTML属性)に適用されるように指定するには、SRW.SET_BEFORE_PAGE_HTML PL/SQL プロシージャを使用する必要があります。

構文

```
SRW.SET_AFTER_REPORT_HTML(type, 'string');
```

パラメータ

type	SRW.FILE_ESCAPEまたはSRW.TEXT_ESCAPE。文字列パラメータがファイル名が挿入されるテキストかを示します。
string	typeパラメータの指定内容に依存した、ファイル名または挿入されるテキスト。

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「レポート後型」および「レポート後値」プロパティを設定します。

SRW.SET_AFTER_REPORT_HTMLの制限

- レポートの下端に含まれるデフォルトHTMLを以下に示します。
SRW.SET_AFTER_REPORT_HTMLを使用してこのテキストを置換する場合、等価なHTMLコマンドを必ず含んでいる必要があります。
`</body></html>`
- SRW.SET_AFTER_REPORT_HTMLを、Before Reportトリガーなど、レポートのフォーマット前に起動するトリガーに設定する必要があります。

SRW.SET_AFTER_REPORT_HTMLの例

```
/* The example below inserts two links to home pages
** at the end of the report output.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.SET_AFTER_REPORT_HTML(SRW.TEXT_ESCAPE, '<CENTER>' ||
    '<A HREF="http://www.oracle.com/">' ||
    'Oracle Corporation</A> -' ||
    chr(10) ||
    '<A HREF="http://home.netscape.com/">' ||
    'Netscape</A> </CENTER>' ||
    '</BODY> </HTML>');
end if;
return (TRUE);
end;
```

SRW.SET_BACKGROUND_BORDER_COLOR

説明

このプロシージャは、オブジェクトのバックグラウンド境界色を指定します。この属性はキャラクタ・モードでは無視されます。使用するパターンによっては、バックグラウンド塗りカラーを表示することができません。たとえば、パターンが実線の場合、バックグラウンド塗りカラーはパターン全体に表示されません。この場合、フォアグラウンド・カラーのみが全体表示されます。

構文

```
SRW.SET_BACKGROUND_BORDER_COLOR('color');
```

パラメータ

color	カラー・パレットからの有効なカラー。
-------	--------------------

使用上の注意

- このプロシージャは枠のパターンをサポートしていないので、Windowsプラットフォームに適応されません。結果として、境界のフォアグラウンドのみがWindowsで示されます。

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「枠のバックグラウンド・カラー」プロパティを設定します。

ユーザー・インタフェース

バックグラウンド境界色をユーザー・インタフェースで設定する方法は、「カラーの変更」を参照してください。

SRW.SET_BACKGROUND_FILL_COLOR

説明

このプロシージャは、オブジェクトの塗りカラーを指定します。この属性はキャラクタ・モードでは無視されます。使用するパターンによっては、バックグラウンド塗りカラーを表示することができません。たとえば、パターンが実線の場合、バックグラウンド塗りカラーはパターン全体に表示されません。この場合、フォアグラウンド・カラーのみが全体表示されます。

構文

```
SRW.SET_BACKGROUND_FILL_COLOR('color');
```

パラメータ

color	カラー・パレットからの有効なカラー。
-------	--------------------

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「バックグラウンド・カラー」プロパティを設定します。

ユーザー・インタフェース

バックグラウンド塗りカラーをユーザー・インタフェースで設定する方法は、「カラーの変更」を参照してください。

SRW.SET_BEFORE_FORM_HTML

説明

このプロシージャは、HTMLパラメータ・フォームの上端に表示するテキスト、グラフィックまたはHTMLコマンドを挿入します。この属性は、ロゴまたは標準リンクをパラメータ・フォーム上に配置するのに便利です。

構文

```
SRW.SET_BEFORE_FORM_HTML(type, 'string');
```

パラメータ

type	SRW.FILE_ESCAPE または SRW.TEXT_ESCAPE。文字列パラメータがファイル名か挿入されるテキストかを示します。
文字列	typeパラメータの指定内容に依存した、ファイル名または挿入されるテキスト。

プロパティ・パレット

プロパティ・パレットを使ってこの属性を定義するには、「フォーム前型」および「フォーム前値」プロパティを設定します。

SRW.SET_BEFORE_FORM_HTMLの例

```

/* The example below sets both the before and after
** form escapes in the Before Form trigger.
** The HTML for the before escape is located in a file
** named logo.htm.The HTML for the after report escape
** is specified within the PL/SQL itself.
*/

function BeforeForm return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.SET_BEFORE_FORM_HTML(SRW.FILE_ESCAPE, 'logo.htm');
    SRW.SET_AFTER_FORM_HTML(SRW.TEXT_ESCAPE, <B> ||
        '<FONT FACE="Arial, Helvetica">' ||
        '<FONT COLOR="#FF0000">' ||
        '<Click <IMG SRC="RUN.GIF" HEIGHT=18 WIDTH=18>' ||
        'to run the report using the parameters you have' ||
        'specified above.' ||
        '</FONT></FONT></B>');
end if;
return (TRUE);
end;

```

SRW.SET_BEFORE_PAGE_HTML

説明

このプロシージャは、文書のページの先頭に表示するテキスト、グラフィックまたはHTMLコマンドを挿入します。この属性は、ロゴまたは標準リンクをHTML文書の各ページの先頭に配置するのに便利です。

HTMLページ・ストリームを使用する場合、このプロシージャはレポート出力のすべてのページ（バックグラウンド・カラーやイメージ、その他の <body> HTML属性）に適用されます。レポートの最初（ヘッダー）または最後（フッター）のページにのみHTMLを指定するには、

SRW.SET_BEFORE_REPORT_HTML または SRW.SET_AFTER_REPORT_HTML PL/SQLプロシージャをそれぞれ使用します。

構文

```
SRW.SET_BEFORE_PAGE_HTML(type, 'string');
```

パラメータ

type	SRW.FILE_ESCAPE またはSRW.TEXT_ESCAPE。文字列パラメータがファイル名が挿入されるテキストかを示します。
string	typeパラメータの指定内容に依存した、ファイル名または挿入されるテキスト。

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「ページ前型」および「ページ前値」プロパティを設定します。

SRW.SET_BEFORE_PAGE_HTMLの制限

- SRW.SET_BEFORE_PAGE_HTMLをレポートのすべてのページに適用する場合、Before Report トリガーなど、レポートのフォーマットが始まる前に起動するトリガーに指定する必要があります。
- SRW.SET_BEFORE_PAGE_HTMLを現行ページのみ適用する場合、そのページのオブジェクトのフォーマット・トリガーに指定する必要があります。

SRW.SET_BEFORE_PAGE_HTMLの例

```
/* The example below centers the document in
** the browser.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.SET_BEFORE_PAGE_HTML(SRW.TEXT_ESCAPE, '<CENTER>');
    SRW.SET_AFTER_PAGE_HTML(SRW.TEXT_ESCAPE, '</CENTER>');
end if;
return (TRUE);
end;
```

SRW.SET_BEFORE_PRINTING_CODE

説明

このプロシージャは、オブジェクトの各行の前に実行されるプリンタ・エスケープ・シーケンスへの参照を挿入します。オブジェクトの印刷がトリガーされた後、実際に印刷される前に、プリンタ・エスケープ・シーケンスが挿入されます。プリンタ・コードは、キャラクタ・モードでレポートを実行しているときのみ使用します。

構文

```
SRW.SET_BEFORE_PRINTING_CODE('code');
```

パラメータ

code	DESFORMAT用に指定された.prtファイルに定義された有効なプリンタ・コード。
------	--

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「前のプリンタ・コード」プロパティを設定します。

SRW.SET_BEFORE_REPORT_HTML

説明

このプロシージャは、文書の先頭に表示するテキスト、グラフィックまたはHTMLコマンドを挿入します。この属性は、ロゴまたは標準リンクをHTML文書の先頭に配置するのに便利です。

HTMLページ・ストリームを使用する場合、このプロシージャはレポート出力の最初のページに適用されます。HTMLがレポート全体(バックグラウンド・カラーやイメージ、その他の <body> HTML属性)に適用されるように指定するには、SRW.SET_BEFORE_PAGE_HTML PL/SQL プロシージャを使用する必要があります。

構文

```
SRW.SET_BEFORE_REPORT_HTML(type, 'string');
```

パラメータ

type	SRW.FILE_ESCAPE またはSRW.TEXT_ESCAPE。文字列パラメータがファイル名か挿入されるテキストかを示します。
文字列	typeパラメータの指定内容に依存した、ファイル名または挿入されるテキスト。

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「レポート前型」および「レポート前値」プロパティを設定します。

SRW.SET_BEFORE_REPORT_HTMLの制限

- レポートの先頭に含まれるデフォルトHTMLを以下に示します。
SRW.SET_BEFORE_REPORT_HTMLを使用してこのテキストを置換する場合、等価なHTMLコマンドを必ず含んでいる必要があります。

```
<html>
<body bgcolor="#ffffff">
```

- SRW.SET_BEFORE_REPORT_HTMLを、Before Reportトリガーなど、レポートのフォーマットが始まる前に起動するトリガーに設定する必要があります。

SRW.SET_BEFORE_REPORT_HTMLの例

```
/* The example below sets both the before and after
** report escapes in the Before Report trigger.
** The HTML for the before escape is located in a file
** named logo.htm.The HTML for the after report escape
** is specified within the PL/SQL itself.
*/

function BeforeReport return boolean is
begin
if ((upper(:MODE) != 'CHARACTER') and
    (upper(:DESFORMAT) = 'HTML')) then
    SRW.SET_BEFORE_REPORT_HTML(SRW.FILE_ESCAPE, 'logo.htm');
    SRW.SET_AFTER_REPORT_HTML(SRW.TEXT_ESCAPE, '<ADDRESS>' ||
    ' Questions?-' ||
    '<A HREF=mailto:webmaster@xyztech.com>' ||
    'webmaster@xyztech.com</A>' ||
    '</ADDRESS>' ||
    '</body></html>');
end if;
return (TRUE);
end;
```

SRW.SET_BOOKMARK

説明

このプロシージャは、ブックマークをオブジェクトに関連付けて、マスターHTML文書またはPDF文書のブックマーク枠に表示する文字列を指定します。ブックマーク領域をオープンし、ブックマークをクリックすると、ウィンドウ上端にオブジェクトが表示されます。

構文

```
SRW.SET_BOOKMARK('bookmark');
```

パラメータ

bookmark	<p>次のどれかである:</p> <ul style="list-style-type: none"> ■ インデント/順序の情報が指定されていないテキスト文字列。ブックマークは、レポートがオブジェクトを渡す順に、ブックマーク・リストに表示されます。 ■ 順序付けおよびインデントを明示したx#book_mark_name形式のテキスト文字列で、xにはアウトライン番号が入ります。シャープ記号 (#) とアウトライン番号は、「ブックマーク」ウィンドウには表示されませんが、順序とインデントを決めるのに使用されます。たとえば、次のように指定します。 <ul style="list-style-type: none"> 1#Expense Summary Section 2#Expense Detail Section 2.1#Expenses for the Administration Department 2.2#Expenses for the Engineering Department 2.3#Expenses for the Sales Department 2.3.1#Expenses for the Eastern Sales Region 2.3.2#Expenses for the Northern Sales Region 2.3.3#Expenses for the Southern Sales Region 2.3.4#Expenses for the Western Sales Region
----------	--

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「ブックマーク」プロパティを設定します。

SRW.SET_BOOKMARKの制限

- 同じアウトライン番号が何回も使用する場合、エントリはすべて表示されますが、順序はレポートがオブジェクトを渡した順で定義されます。
- 番号がとんでいる場合、次の2つのうちの1つが発生します。同等レベルの番号がとんでいる場合は、目に見える影響はありません（たとえば、1.3.1と1.3.3で、1.3.2がない場合、同じインデント・レベルで並んで表示されます）。上位の番号と低位の番号の間がとんでいる場合、中間のレベルが必要に応じて生成されます（たとえば、1.0の後が2.1.1ならば、見かけ上の2と2.1のエントリが定義されます。そのタイトルは、後の実際のエントリと同じになります）。
- SRW.SET_BOOKMARKは次のトリガーのみに設定する必要があります。

Format

SRW.SET_BOOKMARKの例

```
/* The format trigger below defines a bookmark for
** the boilerplate object B_2. Notice the use of
** explicit ordering information (1#) in this example.
** If you did not want the ordering information, you
```

```

** could omit 1#.
*/

function B_2FormatTrigger return boolean is
begin
  srw.set_bookmark('1#Expense Summary Section');
  return (TRUE);
end;

```

動的SRW.SET_BOOKMARKの例

```

/* The format trigger below defines a bookmark for
** the boilerplate object B_2. Notice that the name
** of the bookmark is dynamic.CATEGORY is a column
** value that is concatenated with the string Expense
** Summary Section for each execution of the format
** trigger.In this case, CATEGORY could contain
** ordering information (e.g., 1#) or perhaps a string
** that makes the bookmark unique within the report.
*/

function B_2FormatTrigger return boolean is
begin
  srw.set_bookmark(:category ||
  ' Expense Summary Section');
  return (TRUE);
end;

```

SRW.SET_BORDER_PATTERN

説明

このプロシージャは、オブジェクト境界線の塗りパターンを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_BORDER_PATTERN('pattern');
```

パラメータ

pattern	境界線の塗りパターン(実線など)。
---------	-------------------

使用上の注意

- このプロシージャは枠のパターンをサポートしていないので、Windowsプラットフォームに適用されません。

プロパティ・パレット

(テンプレートのみ) プロパティ・パレットを使ってテンプレートにこの属性を定義するには、「枠のパターン」プロパティを設定します。

ユーザー・インタフェース

境界線の属性をユーザー・インタフェースで設定する方法は、「オブジェクト境界線属性の変更」を参照してください。

SRW.SET_BORDER_WIDTH

説明

このプロシージャは、オブジェクト境界線の幅をピカ単位で指定します。

構文

```
SRW.SET_BORDER_WIDTH(width);
```

パラメータ

幅	0(ゼロ)または正の整数。ゼロは境界線がないことを示し、正の整数は境界線の幅をピカ単位で示します。
---	---

ユーザー・インタフェース

境界線の属性をユーザー・インタフェースで設定する方法は、「オブジェクト境界線属性の変更」を参照してください。

SRW.SET_CHARMODE_TEXT

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドのフォント・スタイルを指定します。この属性はビットマップ・レポートでは無視されます。

構文

```
SRW.SET_CHARMODE_TEXT('style');
```

パラメータ

style	次のどれかである: SRW.REVERSE_TEXTA SRW.BOLD_TEXTA SRW.REVERSEBOLD_TEXTA SRW.UNDERLINE_TEXTA SRW.UNDERLINEREVERSE_TEXTA SRW.REVERSEBOLDUNDERLINE_TEXTA SRW.PLAIN_TEXTA
-------	---

ユーザー・インタフェース

フォントの属性をユーザー・インタフェースで設定する方法は、「テキストの属性の変更」を参照してください。

SRW.SET_CUSTOM_SPACING

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドに先行するグローバル・テキストまたはボイラープレート・テキストを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_CUSTOM_SPACING(spacing);
```

パラメータ

spacing	VGS単位で先行を示す数であり、非常に小さい。
---------	-------------------------

SRW.SET_DISPLAY_NAME

説明

このプロシージャは、HTMLまたはHTMLCSS形式のレポート出力のイメージ・オブジェクト上にカーソルが移動したときに、ポップアップとして表示されるテキストを指定します。

構文

```
SRW.SET_DISPLAY_NAME('text_string');
```

パラメータ

text_string	ASCIIの26個の英文字(大文字および小文字)、数字またはアンダースコアのみを使った任意のテキスト文字列。他の特殊文字は、アンダースコア文字に自動的に変換されます。
-------------	---

プロパティ・パレット

プロパティ・パレットを使ってこの属性を定義するには、「ディスプレイ名」プロパティを設定します。

SRW.SET_DISPLAY_NAMEの制限

- SRW.SET_DISPLAY_NAMEは、次のトリガーにのみ設定する必要があります。

フォーマット・トリガー

SRW.SET_DISPLAY_NAMEの例

```
/* The format trigger below assigns a text string
** ('Click here to fill your shopping basket') to the image
** object BASKET.In the HTML report, this popup will display
** when the cursor moves over the BASKET object.
*/

function BASKETFormatTrigger return boolean is
begin
  SRW.SET_DISPLAY_NAME('Click here to fill your shopping basket');
  return (TRUE);
end;
```

SRW.SET_FIELD

説明

このプロシージャは、文字、数値または日付フィールドの値を設定します。これは、フィールドの値を条件付きで変更するのに便利です。

構文

```
SRW.SET_FIELD (object_id, text CHAR|number NUM|date DATE);
```

パラメータ

object_id	常に0(ゼロ)。(オブジェクトは常に自身の属性を設定する必要があります。)
text number date	フィールドに表示する文字、数値または日付の文字列。

使用上の注意

- 日付値の場合は、2番目の引数をTO_DATEファンクションで日付値に変換する必要があります。たとえば、次のように指定します。

```
srw.set_field(0, to_date('01-JAN-99'));
```

- 値の両側の引用符を省略すると、その値は数値と見なされます。たとえば、次のように指定します。

```
srw.set_field(0, 99);
```

あるいは、TO_NUMBERファンクションを使用すれば、引用符を使用することもできます。たとえば、次のように指定します。

```
srw.set_field(0, to_number('99'));
```

- ファンクションなしで引用符を使用すると、その値は文字列と見なされます。あるいは、数値や日付値と統一するために、TO_CHARファンクションを使用することもできます。

```
srw.set_field(0, to_char('my string'));
```

SRW.SET_FIELDの例

```
/* Suppose you want to conditionally change the
** number of a field, based on each employee's salary.
** In the format trigger for the field, you could
** type the following:
**/
```

```
FUNCTION CHGFIELD RETURN BOOLEAN IS
TMP    NUMBER;
BEGIN
  if :sal >= 2000 then
    tmp := :sal * 1.08;
    srw.set_field (0, tmp);
  else
    srw.set_field (0, 2500);
  end if;
RETURN(TRUE);
END;
```

SRW.SET_FIELD_CHAR

説明

このプロシージャは、文字フィールドの値を設定します。これは、フィールドの文字値を条件付きで変更するのに便利です。

構文

```
SRW.SET_FIELD_CHAR (object_id, text CHAR);
```

パラメータ

object_id	常に0(ゼロ)。(オブジェクトは常に自身の属性を設定する必要があります。)
text	フィールドに表示する文字列。

SRW.SET_FIELD_CHARの制限

- SRW.SET_FIELD_CHARは、データ型が文字のフィールドのフォーマット・トリガーでのみ意味を持ちます。他の場所に入力しても影響はありません。

SRW.SET_FIELD_CHARの例

```
/* Suppose you want to conditionally change the value of a
** Character field, based on each employee's salary.
** In the format trigger for the field, you could type the following:
*/
```

```
FUNCTION CHGFIELD RETURN BOOLEAN IS
BEGIN
    if :sal >= 2000000 then
        srw.set_field_char (0, 'HIGH SALARY');
    end if;
    RETURN(TRUE);
END;
```

SRW.SET_FIELD_DATE

説明

このプロシージャは、日付フィールドの値を設定します。これは、フィールドの日付値を条件付きで変更するのに便利です。

構文

```
SRW.SET_FIELD_DATE (object_id, date DATE);
```

パラメータ:

object_id	常に0(ゼロ)。(オブジェクトは常に自身の属性を設定する必要があります。)
date	フィールドに表示する日付。

SRW.SET_FIELD_DATEの制限

- SRW.SET_FIELD_DATEは、日付フィールドのフォーマット・トリガーでのみ意味を持ちます。他の場所に入力しても影響はありません。

SRW.SET_FIELD_DATEの例

```

/* Suppose you want to conditionally change the date of the reunion
** invitation, based on each student's graduation year.In the format
** trigger for the field, you could type the following:
*/

FUNCTION CHGFIELD RETURN BOOLEAN IS
BEGIN
  if :graduation >= 1975 then
    srw.set_field_date (0, '02-JUL-95');
  else
    end if;
RETURN (TRUE);
END;
```

SRW.SET_FIELD_NUM

説明

このプロシージャは、数値フィールドの値を設定します。これは、フィールドの数値を条件付きで変更するのに便利です。

構文

```
SRW.SET_FIELD_NUM (object_id, number NUM);
```

パラメータ

object_id	常に0(ゼロ)。(オブジェクトは常に自身の属性を設定する必要があります。)
number	フィールドに表示する数値。

SRW.SET_FIELD_NUMの制限

- SRW.SET_FIELD_NUMは、データ型がNUMBERのフィールドのフォーマット・トリガーでのみ意味を持ちます。他の場所に入力しても影響はありません。

SRW.SET_FIELD_NUMの例

```

/* Suppose you want to conditionally change the number of a field,
** based on each employee's salary.In the format trigger for the
** field, you could type the following:
*/

FUNCTION CHGFIELD RETURN BOOLEAN IS
TMP  NUMBER;
BEGIN
```

```
if :sal >= 2000 then
  tmp := :sal * 1.08;
  srw.set_field_num (0, tmp);
else
  srw.set_field_num (0, '2500');
end if;
RETURN(TRUE);
END;
```

SRW.SET_FILL_PATTERN

説明

このプロシージャは、オブジェクトの塗りパターンを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_FILL_PATTERN('pattern');
```

パラメータ

pattern	オブジェクトの塗りパターン(実線など)。
---------	----------------------

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「塗りパターン」プロパティを設定します。

ユーザー・インタフェース

パターンの属性をユーザー・インタフェースで設定する方法は、「パターンの変更」を参照してください。

SRW.SET_FONT_FACE

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドのフォントを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_FONT_FACE('face');
```

パラメータ

face	フォント。 arialなど。
------	----------------

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「フォント」プロパティを設定します。

ユーザー・インタフェース

フォントの属性をユーザー・インタフェースで設定する方法は、「テキストの属性の変更」を参照してください。

SRW.SET_FONT_SIZE

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドのフォント・サイズを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_FONT_SIZE(size);
```

パラメータ

size	フォント・サイズ。 9(ポイント)など。
------	----------------------

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「フォント」プロパティを設定します。

ユーザー・インタフェース

フォントの属性をユーザー・インタフェースで設定する方法は、「テキストの属性の変更」を参照してください。

SRW.SET_FONT_STYLE

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドのフォント・スタイルを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_FONT_STYLE(style);
```

パラメータ

style	次のどれかである： SRW.ITALIC_STYLE SRW.OBLIQUE_STYLE SRW.UNDERLINE_STYLE SRW.OUTLINE_STYLE SRW.SHADOW_STYLE SRW.INVERTED_STYLE SRW.BLINK_STYLE SRW.PLAIN_STYLE
-------	--

使用上の注意

- このプロシージャを同じフィールドで複数回使った場合、フォント・スタイルは蓄積されません。たとえば、次の場合はフィールドに下線が引かれイタリックに設定されます：

```
SRW.SET_FONT_STYLE(SRW.UNDERLINE_STYLE);  
SRW.SET_FONT_STYLE(SRW.ITALIC_STYLE);
```

蓄積されたすべてのスタイルを否定するには、PLAIN_STYLE を使用します。

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「フォント」プロパティを設定します。

ユーザー・インタフェース

フォントの属性をユーザー・インタフェースで設定する方法は、「テキストの属性の変更」を参照してください。

SRW.SET_FONT_WEIGHT

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドのフォントの太さを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_FONT_WEIGHT(weight);
```

パラメータ

weight	次のどれかである: SRW.ULTRALIGHT_WEIGHT SRW.EXTRALIGHT_WEIGHT SRW.LIGHT_WEIGHT SRW.DEMILIGHT_WEIGHT SRW.DEMIBOLD_WEIGHT SRW.BOLD_WEIGHT SRW.EXTRABOLD_WEIGHT SRW.MEDIUM_WEIGHT
--------	--

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「フォント」プロパティを設定します。

ユーザー・インタフェース

フォントの属性をユーザー・インタフェースで設定する方法は、「テキストの属性の変更」を参照してください。

SRW.SET_FOREGROUND_BORDER_COLOR

説明

このプロシージャは、オブジェクトのフォアグラウンド境界色を指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_FOREGROUND_BORDER_COLOR('color');
```

パラメータ

color	カラー・パレットからの有効なカラー。
-------	--------------------

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「枠のフォアグラウンド・カラー」プロパティを設定します。

ユーザー・インタフェース

フォアグラウンド境界色をユーザー・インタフェースで設定する方法は、「カラーの変更」を参照してください。

SRW.SET_FOREGROUND_FILL_COLOR

説明

このプロシージャは、オブジェクトのフォアグラウンドのバックグラウンド色を指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_FOREGROUND_FILL_COLOR('color');
```

パラメータ

color	カラー・パレットからの有効なカラー。
-------	--------------------

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「フォアグラウンド・カラー」プロパティを設定します。

ユーザー・インタフェース

フォアグラウンド塗りカラーをユーザー・インタフェースで設定する方法は、「カラーの変更」を参照してください。

SRW.SET_FORMAT_MASK

説明

このプロシージャは、DATEまたはNUMBERフィールドの書式マスクを指定します。

構文

```
SRW.SET_FORMAT_MASK('mask');
```

パラメータ

mask	有効な書式マスク。日付書式マスク構文または数値書式マスク構文を参照してください。
------	--

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「書式マスク」プロパティを設定します。

SRW.SET_HYPERLINK

説明

このプロシージャは、URL Webリンクを指定します。

注意: Webリンクのソースであるオブジェクトをアクションに関連付けることもできません。

構文

```
SRW.SET_HYPERLINK('hyperlink');
```

パラメータ

hyperlink	<p>有効なリンク:</p> <ul style="list-style-type: none"> ■ 同一マシン上の他の文書(たとえば、file:/private/mynewdoc.pdf または file:///C:/temp/mynewdoc.pdf) ■ 別のマシン上の別の文書(たとえば、http://www.newmach.com/newdoc.pdf) ■ 現行文書内の宛先(たとえば、#my_dest_name) ■ ローカル文書内の宛先(たとえば、file:/private/somedoc.pdf#a_dest_name) ■ リモート文書内の宛先(たとえば、http://www.newmach.com/newdoc.pdf#some_dest_name) ■ すべてのURL (たとえば、http://www.newmach.com/newdoc.html, ftp://www.reposit.com/filetoget.example, http://www.somemch.com/cgi-bin/webmenu?choice1)
-----------	--

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「ハイパーリンク」プロパティを設定します。

SRW.SET_HYPERLINKの制限

- SRW.SET_HYPERLINKは次のトリガーのみに設定する必要があります:

フォーマット・トリガー

- PDFビューアからリモート・サーバーまたはHTML文書へのWebリンクをたどるには、Webブラウザと連動するようにPDFビューアを構成する必要があります(たとえば、ヘルパー・アプリケーションとして構成したり、プラグインとしてWebブラウザにインストールします)。

SRW.SET_HYPERLINKの例

```
/* The format trigger below defines a Web link
** to a destination (mytarget) in another document
** (newdoc.pdf) for the boilerplate object B_2.
** Note:If the target were in the same document,
** you would omit http://www.newmach.com/newdoc.pdf.
```

```
*/  
  
function B_2FormatTrigger return boolean is  
begin  
  SRW.SET_HYPERLINK('http://www.newmach.com/newdoc.pdf' ||  
    '#mytarget');  
  return (TRUE);  
end;
```

動的SRW.SET_HYPERLINKの例

```
/* The format trigger below defines a Web link  
** for the boilerplate object B_3. Notice how the  
** destination of the link is determined dynamically  
** based upon the values of SRCDIR and CF_PICKVIDEO.  
** For example, if the value of SRCDIR were  
** http://www.newmach.com and CF_PICKVIDEO were good.avi,  
** this function would assign the following Web  
** link to the object:  
** http://www.newmach.com/webdemo/src/good.avi.  
*/  
  
function B_3FormatTrigger return boolean is  
begin  
  SRW.SET_HYPERLINK(:srcdir||:cf_pickvideo);  
  if ( upper(:cf_pickvideo) like '%GOOD%' ) then  
    SRW.SET_FOREGROUND_BORDER_COLOR('green');  
    SRW.SET_BACKGROUND_BORDER_COLOR('green');  
  end if;  
  return (TRUE);  
end;
```

```
/* SRCDIR is a parameter whose value is determined at  
** runtime by the following After Parameter Form trigger  
*/  
  
function AfterPForm return boolean is  
begin  
  :srcdir := :web_server || '/webdemo/src/';  
  :docsdire := :web_server || '/webdemo/docs/';  
  return (TRUE);  
end;
```

```
/* CF_PICKVIDEO is a formula column whose value is  
** determined by the following function  
*/  
  
function CF_pickvideoFormula return Char is  
begin  
  if ( :avg_h_div < .80 )  
    then return ('bad.avi');  
    else return ('good.avi');  
  end if;
```

```
end;
```

SRW.SET_HYPERLINK_ATTRS

説明

このプロシージャは、プロパティ・パレットまたはSRW.SET_HYPERLINKプロシージャで指定されたハイパーリンクに追加のHTMLコマンドを適用します。

構文

```
SRW.SET_HYPERLINK_ATTRS('string');
```

パラメータ

HTML_string	適用されるHTMLコマンドを含む文字列。
-------------	----------------------

プロパティ・パレット

プロパティ・パレットを使ってこの属性を定義するには、「追加のハイパーリンク属性」プロパティを設定します。

SRW.SET_HYPERLINK_ATTRSの制限

- SRW.SET_HYPERLINK_ATTRSは、次のトリガーにのみ設定する必要があります。

フォーマット・トリガー

SRW.SET_HYPERLINK_ATTRSの例

```
/* The format trigger below generates a status
** of information at the bottom of your Web browser
** when the mouse cursor passes over a Shopping Basket
** object.
*/

function BASKETFormatTrigger return boolean is
begin
SRW.SET_HYPERLINK_ATTRS('onMouseover "window.status="Click here to see
what is in your shopping basket";return true"|| 'onMouseout
="window.status=" ";return true"');
return (TRUE);
end;
```

SRW.SET_JUSTIFICATION

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドの水平文字位置を指定します。

構文

```
SRW.SET_JUSTIFICATION(justification);
```

パラメータ

justification	次のどれかである: SRW.LEFT_HJUST SRW.RIGHT_HJUST SRW.FLUSH_HJUST SRW.CENTER_HJUST.
---------------	--

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義する場合、文字フィールドでは「文字揃え」プロパティ、日付フィールドでは「日付揃え」プロパティ、数値フィールドでは「数値揃え」プロパティを設定します。

ユーザー・インタフェース

水平文字位置をユーザー・インタフェースで設定する方法は、「テキストの属性の変更」を参照してください。

SRW.SET_LINKTAG

説明

このプロシージャは、オブジェクトの識別子を指定したり、Webリンクの宛先として使用することができます。

構文

```
SRW.SET_LINKTAG('linktag');
```

パラメータ

linktag	ASCIIの26個の英文字(大文字および小文字)、数字またはアンダースコアのみを使用した、有効で、一意の名前。他の特殊文字は、アンダースコア文字に自動的に変換されます。
---------	--

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「ハイパーリンクの宛先」プロパティを設定します。

SRW.SET_LINKTAGの制限

- SRW.SET_LINKTAGは次のトリガーのみに設定する必要があります:

フォーマット・トリガー

SRW.SET_LINKTAGの例

```
/* The format trigger below assigns an identifier
** (mytarget) to the boilerplate object B_500.
** This means that the object can now be the destination
** of a Web link.
*/

function B_500FormatTrigger return boolean is
begin
  SRW.SET_LINKTAG('mytarget');
  return (TRUE);
end;
```

動的SRW.SET_LINKTAGの例

```
/* The format trigger below assigns an identifier to
** the field F_Dept by concatenating some static text
** (dept_details_) with the value of the source column
** (DEPTNO).This function ensures that a unique
** identifier is assigned to each instance of F_Dept.
*/

function F_DeptFormatTrigger return boolean is
begin
  SRW.SET_LINKTAG('dept_details_' ||
  LTRIM(TO_CHAR(:deptno)));
  return (TRUE);
end;
```

SRW.SET_MAXROW

説明

このプロシージャは、指定した問合せの際にフェッチするレコードの最大数を設定します。これは、レポートが、フェッチする問合せ(1つ以上)よりも少ないレコードをフォーマットする(つまり、表示する)場合に便利です。したがって、SRW.SET_MAXROWを使用して、レポート用にフェッチするデータを条件付きで制限できるので、レポートのパフォーマンスが向上します。

構文

```
SRW.SET_MAXROW (query_name CHAR, maxnum PLS_INTEGER);
```

パラメータ

query_name	フェッチするレコードが制限される問合せ。
maxnum	問合せによりフェッチするレコードの最大数。

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「最大取出し行」プロパティを設定します。

SRW.SET_MAXROWの制限

- SRW.SET_MAXROWは、Before Reportトリガーでのみ意味を持ちます(つまり、問合せが解析された後)。Before Reportトリガーの後にSRW.SET_MAXROW が呼び出された場合(つまり、問合せが実行された後)、SRW.MAXROW_UNSET パッケージ例外が発生します。
- このプロシージャは指定した数のレコードのみをフェッチするので、問合せの"フェッチされない"レコードは計算などに使用されません。
- レコードをフェッチしないように指定しても、問合せが解析されます。

SRW.SET_MAXROWの例

```
/* Suppose your report has two queries, Q_Stocks and Q_Bonds.
** Suppose also, that you have a user-created parameter, named
** WHICHDATA, that enables users to specify which data they want
** the report to display:either stocks or bonds.In the
** Before Report trigger, you could use the SRW.SET_MAXROW
** procedure to ensure that only one query's data is fetched:
*/
```

```
FUNCTION FETCHIT RETURN BOOLEAN IS
BEGIN
  if :whichdata != 1 then
    srw.set_maxrow ('Q_Stocks', 0);
  else
    srw.set_maxrow ('Q_Bonds', 0);
  end if;
RETURN(TRUE);
END;
```

SRW.SET_PAGE_NAVIGATION_HTML

説明

このプロシージャは、ページ・ストリームHTML/HTMLCSSレポート出力のナビゲーション・コントロール用のスクリプトを指定します。

構文

```
SRW.SET_PAGE_NAVIGATION_HTML(type, 'string');
```

パラメータ

type	SRW.FILE_ESCAPE またはSRW.TEXT_ESCAPE.stringパラメータがファイル名なのか、挿入するHTMLスクリプト・テキストなのかを示します。
文字列	ファイル名または挿入するHTMLスクリプト・コード。 typeパラメータに指定された値によって変わります。空白にした場合は、デフォルトのReport Builder JavaScriptを使って、レポート出力の各ページのナビゲーション・コントロールが定義されます。

使用上の注意

ナビゲーション・コントロールに自分で作成したスクリプトを使用する場合、スクリプトに次の3つの変数を含める必要があります。

- &TotalPages (合計ページ数)
- &file_name (出力宛先ファイルの名前。 例 myreport)
- &file_ext (ファイル拡張子の名前。 例 htm)

Report Builderによって、書式設定時にこれらの変数が割り当てられます。

ページ・ナビゲーション・コントロール枠の高さは、スクロール・バーなしで70ポイントにハードコードされます。枠の幅は可変です。この枠の中でアイコンやイメージを使用する場合は、これらの寸法を覚えておいてください。

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「ページ・ナビゲーション・コントロール・タイプ」および「ページ・ナビゲーション・コントロール値」プロパティを設定します。

SRW.SET_PDF_ACTION

説明

このプロシージャは、PDFビューアでオブジェクトがクリックされたときにローカル・マシンで実行されるコマンド行を指定します。

注意: アクションに関連付けられたオブジェクトを、Webリンクのソースにすることはできません。

構文

```
SRW.SET_PDF_ACTION('action');
```

パラメータ

action	ローカル・マシン上の任意の有効なコマンド行(たとえば、c:¥orawin¥bin¥rwr60 userid=scott/tiger report=example.rdfまたはusr/local/bin/phone smith)。
--------	---

プロパティ・パレット

プロパティ・パレットを使用してこの属性を定義するには、「アプリケーション・コマンド・ライン(PDF)」プロパティを設定します。

SRW.SET_PDF_ACTIONの制限

- SRW.SET_PDF_ACTIONは、PDF出力でのみ使用できます。
- SRW.SET_PDF_ACTIONは、次のトリガーのみに設定する必要があります:

Format

SRW.SET_PDF_ACTIONの例

```
/* The format trigger below runs a
** report named example.rdf when the
** button object U_Report is clicked
** in the PDF viewer.
*/

function U_ReportFormatTrigger return boolean is
begin
  SRW.SET_PDF_ACTION ||
  ('c:¥orawin¥bin¥rwr60' ||
  'userid=scott/tiger ' ||
  'report=example.rdf');
end;
```

SRW.SET_PRINTER_TRAY

説明

このプロシージャは、有効なプリンタ・トレイ名を指定したり、レポート書式に応じて異なるプリンタ・トレイに切り替えるために使用することができます。

構文

```
SRW.SET_PRINTER_TRAY('tray');
```

パラメータ

tray	有効なプリンタ・トレイ名であり、レポート書式に応じて異なるプリンタ・トレイに切り替えるために使用することができる。
------	---

使用上の注意

- オブジェクトのフォーマット・トリガーでSRW.SET_PRINTER_TRAYを使用する場合、オブジェクトが印刷されるページによってプリンタ・トレイは変更されます。次のトリガーで変更するまで、指定したプリンタ・トレイは維持されます。

SRW.SET_PRINTER_TRAYの制限

- SRW.SET_PRINTER_TRAYは次のトリガーのみに設定する必要があります:

- After Reportトリガー
- Before Reportトリガー
- Between Pagesトリガー
- Formatトリガー

Before FormトリガーまたはAfter FormトリガーでSRW.SET_PRINTER_TRAYを使用する場合、無効コンテキスト・エラーが発生します。

- Motif版のPCLプリンタにレポートを出力する場合、PRINTER_INTRAYは無視されます。

SRW.SET_PRINTER_TRAYの例

```
/* The example below sets the printer
** tray in a Between Pages trigger.
*/

function BetweenPages return boolean is
begin
  SRW.SET_PRINTER_TRAY('letterhead');
  return (TRUE);
end;
```

SRW.SET_SPACING

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドに先行するグローバル・テキストまたはボイラプレート・テキストを指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_SPACING (spacing);
```

パラメータ

spacing	次のどれかである： SRW.SINGLE_SPACING SRW.DOUBLE_SPACING SRW.ONEHLF_SPACING
---------	---

SRW.SET_TEXT_COLOR

説明

このプロシージャは、CHAR、DATEまたはNUMBERフィールドのグローバル文字色を指定します。この属性はキャラクタ・モードでは無視されます。

構文

```
SRW.SET_TEXT_COLOR ('color');
```

パラメータ

color	カラー・パレットからの有効なカラー。
-------	--------------------

プロパティ・パレット

(テンプレートのみ)プロパティ・パレットを使用してテンプレートにこの属性を定義するには、「テキスト・カラー」プロパティを設定します。

ユーザー・インタフェース

テキスト・カラーをユーザー・インタフェースで設定する方法は、「カラーの変更」を参照してください。

トリガー

使用する「レポート・トリガー」の選択

一般的なルールとして、レポートによって取り出されるデータに影響する処理は、「BEFORE PARAMETER FORMトリガー」または「AFTER PARAMETER FORMトリガー」で実行する必要があります。（これら2つは、解析やフェッチが行われる前に起動する「レポート・トリガー」です。）レポートによって取り出されたデータに影響しない処理は、その他のトリガー内で実行できます。

Report Builderには、5つの「グローバル・レポート・トリガー」があります。新規の「グローバル・レポート・トリガー」作成はできません。トリガー名は、どの時点でトリガーが起動するかを示しています。

BEFORE REPORT	問合せが解析された後、レポートが実行される前に起動します。
AFTER REPORT	プレビューを終了した後、またはレポート出力がファイル、プリンタなどの指定した宛先に送られた後に起動します。このトリガーは、表の削除などの完了した初期処理をクリーン・アップするのに使用できます。ただし、このトリガーは、レポートが正常に完了したかどうかにかかわらず、必ず起動することに注意してください。
BETWEEN PAGES	レポートの最初のページを除く各ページがフォーマットされる前に起動します。このトリガーは、カスタマイズされたページのフォーマットにも使用できます。プレビューでは、このトリガーは、あるページに初めて進むときにのみ起動します。後からそのページに戻っても、トリガーは起動しません。
BEFORE PARAMETER FORM	ランタイム・パラメータ・フォームが表示される前に起動します。このトリガーから、パラメータおよびPL/SQLグローバル変数、レポート・レベルの列の値にアクセスし、変更できます。ランタイム・パラメータ・フォームが非表示であっても、このトリガーは起動します。したがって、このトリガーは、コマンド・ライン・パラメータの妥当性チェックに使用できます。
AFTER PARAMETER FORM	ランタイム・パラメータ・フォームが表示された後に起動します。このトリガーから、パラメータへアクセスし、その値を確認できます。このトリガーは、パラメータ値を変更したり、エラーが発生した場合にランタイム・パラメータ・フォームに戻るのにも使用できます。このトリガーから、データ・モデルの列にはアクセスできません。ランタイム・パラメータ・フォームが非表示であっても、「AFTER PARAMETER FORMトリガー」は起動します。したがって、このトリガーは、コマンド・ライン・パラメータやその他のデータの妥当性チェックに使用できます。

「レポート・トリガー」の実行順序

レポートが実行される際のイベントの順序は次のようになります。

- 1 「BEFORE PARAMETER FORMトリガー」が起動されます。

- 2 ランタイム・パラメータ・フォームが表示されます（非表示でない場合）。
- 3 「AFTER PARAMETER FORMトリガー」が起動されます（ユーザーがランタイム・パラメータ・フォームから取り消さない場合）。
- 4 レポートが"コンパイル"されます。
- 5 問合せが解析されます。
- 6 「Before Reportトリガー」が起動されます。
- 7 SET TRANSACTION READONLYが実行されます（READONLY引数または設定によって指定した場合）。
- 8 レポートが実行され、最終ページ以外の各ページに対してBetween Pagesトリガーが起動されます（レポートのフォーマット中にはいつでもデータをフェッチすることができます）。この間ユーザー・イグジットでDDLを使用している場合、またはSRW.DO_SQLでDDLを使っている場合、ONFAILURE=COMMITと指定されていてレポートが失敗した場合、のうちのいずれかの場合にCOMMITが発生します。
- 9 COMMITが実行されて、（READONLYが指定されている場合は）トランザクションが終了します。
- 10 「After Reportトリガー」が起動されます。
- 11 ONSUCCESS引数または設定に指定した内容に基づいて、COMMIT、ROLLBACK、NOACTIONのいずれかが実行されます。

注意

- ステップ4から9では、レポートの基になる表を変更するようなDDL文は避けてください。ステップ3で表のスナップショットが取られ、これはレポートの実行中ずっと有効である必要があります。ステップ7から9では、レポートの基になる表の内容を変更するようなDML文は避けてください。問合せは、どのような順序で実行してもかまいませんが、（そのレポートで使用されていない表で実行されない限り）DML文は信頼できないものになります。
- READONLYを指定した場合、DDLと一緒に指定しないでください。DDL文を実行すると（たとえば、SRW.DO_SQLまたはユーザー・イグジットを介して）、COMMITが自動的に発行されます。READONLYを使っている場合は、SET TRANSACTION READONLYで開始されたトランザクションを途中で終了してしまいます。

「レポート・トリガー」の制限

- レポート出力をランタイム・プレビューまたはライブ・プレビューに送っている場合、レポート出力が表示される前に、「レポート・トリガー」のいくつかまたはすべてが起動される場合があることに注意してください。たとえば、SRW.MESSAGEを使って、ある条件が満たされた場合に「BETWEEN PAGESトリガー」でメッセージを発行するとします。レポー

トに前方参照がある場合（最終ページまでに表示されるページの総数など）、Report Builderは前方参照を計算する前にフォーマットを行う必要がある場合があります。そのため、まだページが表示されていない場合でも、そのページはすでにフォーマットされ、トリガーが起動している可能性があります。

- 「レポート・トリガー」では、レポート・レベルの列およびパラメータの値を使用できます。たとえば、ある条件（IF :COUNT1 = 10など）でCOUNT1というパラメータの値を使用する必要があります。しかし、ページ依存の列（「リセット位置」が「Page」に設定されている列）またはページ依存の列に依存する列を参照できないことに注意してください。
- 「BEFORE PARAMETER FORMトリガー」および「AFTER PARAMETER FORMトリガー」と「Before Reportトリガー」および「After Reportトリガー」では、パラメータの値を設定できます（例 代入文:COUNT1 = 15で値を指定）。「Before Reportトリガー」および「After Reportトリガー」では、レポート・レベル、ブレースホルダなどの列の値も設定できます。
- 「BETWEEN PAGESトリガー」には、データ・モデル・オブジェクトの値を設定できません。PL/SQLのグローバル変数を使って、間接的に列やパラメータの値を設定することもお薦めしません。これを行うと、予期しない結果になる可能性があります。
- Report Builderランタイムからレポートを実行する場合（つまり、コマンド・ラインやSRW.RUN_REPORTではなく）、レポートを実行する前に、「BEFORE PARAMETER FORMトリガー」および「AFTER PARAMETER FORMトリガー」、妥当性チェック・トリガーで行ったデータベースの変更をコミットする必要があります。この方法で実行している場合、これらのトリガーは、親プロセスのデータベース接続を共有します。ただし、レポートが実際に実行されると、個々のデータベース接続が確立されます。
- 「AFTER PARAMETER FORMトリガー」が起動すると、字句参照を使用して追加のバインド変数を作成できません。たとえば、次のような問合せがあるとしみます（WHERE句は、字句参照に置き換わることに注意）。

```
SELECT ENAME, SAL FROM EMP
&where_clause
```

WHERE_CLAUSEパラメータの値にバインド変数への参照が含まれている場合、「AFTER PARAMETER FORMトリガー」内またはそれ以前に値を指定する必要があります。「Before Reportトリガー」のパラメータに次の値を指定した場合、エラーが発生します。同じ値を「AFTER PARAMETER FORMトリガー」内で指定すると、レポートは実行されます。

```
WHERE SAL = :new_bind
```

が表示されます。

グループ・フィルタ

説明

グループ・フィルタは「フィルタ・タイプ」プロパティが「PL/SQL」である場合に、グループにどのレコードを挿入するかを判断するPL/SQLファンクションです。ファンクションは、ブール値（TRUEまたはFALSE）を戻す必要があります。ファンクションがTRUEまたはFALSEのどちらを戻すかによって、レポートに対して現行のレコードが挿入されたり除外されたりします。グループ・フィルタには、オブジェクト・ナビゲータまたはプロパティ・パレット（「PL/SQL フィルタ」プロパティ）、PL/SQLエディタからアクセスできます。

定義レベル

グループ

失敗時

グループから現行のレコードを除外します。

グループ・フィルタの例

```
function filter_comm return boolean is
begin
  if :comm IS NOT NULL then
    if :comm < 100 then
      return (FALSE);
    else
      return (TRUE);
    end if;
  else
    return (FALSE); -- for rows with NULL commissions
  end if;
end;
```

グループ・フィルタの制限

- 「フィルタ・タイプ」が「最初」または「最後」の場合、グループ・フィルタはグループに追加できません。
- グループ・フィルタは、クロス積グループに追加できません。
- グループ・フィルタでは、Report Builderの正しい頻度の列およびパラメータの値を読み込むことができますが（次のルールを参照）、これらの値を直接設定することはできません。たとえば、ある条件（IF :COUNT1 = 10 など）でCOUNT1というパラメータの値を使用することはできますが、代入文（:COUNT1 = 10など）に直接その値を設定することはできません。PL/SQLのグローバル変数を使って、間接的に列やパラメータの値を設定することもできない点にも

注意してください。これを行うと、予期しない結果になる可能性があります。ページ依存の列（つまり、「リセット位置」が「ページ」に設定されている列）や、グループ・フィルタのページ依存の列に依存する列を参照することもできません。

- グループ・フィルタに入力するファンクションは、次の列にのみ依存できます。
 - グループの問合せまたはデータ・モデル階層でそれより上のレベルにある問合せが所有するデータベース列。
 - 関係付けられていない問合せ（つまり、グループおよびその上位グループ、下位グループの列に依存しない計算列）に依存する計算列（式またはサマリー）。

式

説明

式は、式列またはブレースホルダ列に挿入するPL/SQLファンクションです。式のPL/SQLには、オブジェクト・ナビゲータまたはPL/SQLエディタ、プロパティ・バレット（つまり、「PL/SQL式」プロパティ）からアクセスできます。

データ型がNumberである列には、データ型Numberの値を戻す式のみを使用できます。データ型がDateである列には、データ型Dateの値を戻す式のみを使用できます。データ型がCharacterである列には、データ型CharacterまたはVARCHAR、VARCHAR2の値を戻す式のみを使用できます。

定義レベル

列

失敗時

列に値が戻されません。

値を追加する式の例

```
function salcomm return NUMBER is
begin
    return(:sal + :comm);
end;
```

条件付き式の例

```
function calcomm return NUMBER is
    temp number;
begin
    if :comm IS NOT NULL then
        temp := :sal + :comm;
    else
```

```
    temp := :sal;  
end if;  
return (temp);  
end;
```

式の制限

- 式では、正しい頻度の列およびブレースホルダ列、パラメータ（次のルールを参照）を読み込み、値を割り当てることができます。たとえば、ある条件（IF :COMP = 10など）でCOMPという列の値を使用し、代入文（:COMP := 15など）に直接その値を設定できます。
- 式は、グループ階層においてその列と同じか、それより高いレベルの列のみを参照できます。たとえば、レポート・レベルの列の式は、その他のレポート・レベルの列のみを参照できません。
- 式は、式で最初に参照される列が最初に計算されるように実行されます。そのために、Report Builderは依存性リストを作成し、計算が正しい順序で行われることを保証します。参照している列によって参照される循環依存は、直接、間接を問わず許されていないことに注意してください。
- ユーザー・イグジットが変更できるのは、ブレースホルダ列の値のみです。
- SRW.DO_SQLを使っている場合、同じレポートで更新または挿入されたデータベースの値を読み込むことはお薦めしません。出力のフォーマットのためにReport Builderがデータベースからレコードをフェッチする正確な時刻は、保証されていません。Report Builderは、パフォーマンスを最適化するために、内部で「データの先取り」を行います。このように、特定のレコードに対して更新を発行する前に、すでにそのレコードにアクセスしている場合があります。Report Builderは、ユーザー・イグジットの起動やサマリーの計算などのイベントが正しい順序で発生することを保証する内部依存性リストを作成します。ただし、Report Builderでは、これらのイベントが内部データ・アクセスやデータのフォーマットと同期して発生するかどうかは保証できません。

妥当性チェック・トリガー

説明

妥当性チェック・トリガーは、パラメータ値をコマンド・ラインで指定する際およびランタイム・パラメータ・フォームを確定する際に実行されるPL/SQLファンクションです（これは、各妥当性チェック・トリガーが、レポート実行時に2回起動する可能性があるということを示しています）。妥当性チェック・トリガーは、パラメータの「初期値」プロパティの妥当性チェックにも使用されます。ファンクションは、ブール値（TRUEまたはFALSE）を戻す必要があります。ファンクションがTRUEまたはFALSEのどちらを戻すかによって、ユーザーはランタイム・パラメータ・フォームに戻ります。妥当性チェック・トリガーには、オブジェクト・ナビゲータまたはPL/SQLエディタ、プロパティ・パレット（「妥当性チェック・トリガー」プロパティ）からアクセスできます。

定義レベル

パラメータ

失敗時

ユーザーは、ランタイム・パラメータ・フォームのパラメータ値に戻り、値を変更するか、ランタイム・パラメータ・フォームを取り消すことができます。

妥当性チェック・トリガーの例

```
/* This function prevents the runtime user from
** sending report output anywhere except a printer.
** The user will be returned to the Runtime Parameter
** Form unless PRINTER is specified as the destination
** type (DESTYPE).
*/

function DESTYPEValidTrigger return boolean is
begin
  IF UPPER(:DESTYPE) = 'PRINTER' THEN
    RETURN (TRUE);
  ELSE
    RETURN (FALSE);
  END IF;
end;
```

妥当性チェック・トリガーの制限

- 妥当性チェック・トリガーのPL/SQLには、最大32Kバイトまでの文字を使用できます。上限は、オペレーティング・システムによって異なる場合があります。
- 妥当性チェック・トリガーでは、Report Builderパラメータを読み込み、値を割り当てることができます。列を読み込んで値を割り当てることはできません。たとえば、ある条件(IF :COUNT1 = 10など)でCOUNT1というパラメータの値を使用することができ、代入文(:COUNT1 = 15など)に直接その値を設定できます。場合によっては、1つのパラメータで、何回も妥当性チェック・トリガーを起動させることができます。このため、通常は、After Parameter Form トリガーで、パラメータ値を割り当てるのが最善の方法です。PL/SQLのグローバル変数を使って、間接的に列の値を設定することはできない点にも注意してください。これを行うと、予期しない結果になる可能性があります。
- 妥当性チェック・トリガーでは、DDLは使用しないでください。
- 他のレポートから発生したプロセスであるレポート(BACKGROUND=YESで実行した場合など)では、レポートを実行する前に、BEFORE PARAMETER FORMトリガーおよびAFTER PARAMETER FORMトリガー、妥当性チェック・トリガーで行ったデータベースの変更をコミットする必要があります。発生したプロセスは、その親プロセスのデータベース接続をBEFORE PARAMETER FORMトリガーおよびAFTER PARAMETER FORMトリガー、妥当性

チェック・トリガーに使用します。発生したプロセスがレポートを実行する場合には、個々のデータベース接続が確立されます。したがって、子レポートの実行時までコミットされなかったデータベースの変更は失われます。

フォーマット・トリガー

説明

フォーマット・トリガーは、オブジェクトがフォーマットされる前に実行されるPL/SQLファンクションです。トリガーは、オブジェクトのフォーマット属性を動的に変更するのに使用できます。ファンクションは、ブール値（TRUEまたはFALSE）を戻す必要があります。ファンクションがTRUEまたはFALSEのどちらを戻すかによって、オブジェクトの現行のインスタンスがレポート出力に対して挿入されたり除外されたりします。フォーマット・トリガーには、オブジェクト・ナビゲータまたはプロパティ・パレット、PL/SQLエディタからアクセスできます。

定義レベル

レイアウト・オブジェクト

失敗時

出力からオブジェクトの現行のインスタンスを除外します。

使用上の注意

- フォーマット・トリガーは、レポートによって取り出されるデータには影響しません。たとえば、フォーマット・トリガーがフィールドにFALSEを戻すと、出力にはフィールドは表示されませんが、フィールドのデータは取り出されます。
- フォーマット・トリガーによってレポートの最終ページでレポート出力が非表示になる場合でも、最終ページはフォーマット設定され、適切な出力に送信されます。ページは合計ページ数に含まれます。

フォーマット・トリガーの例（値をハイライト表示する）

```
/* Suppose that you are building a banking report and
** would like it to indicate if a customer is overdrawn.
** To do so, you give the repeating frame around the
** customer information a format trigger that causes
** it to have a border only if a customer's account
** balance is less than 0 (or the required minimum
**balance).
**/

function my_formtrig return BOOLEAN is
begin
if :bal < 0 then
    srw.attr.mask := SRW.BORDERWIDTH_ATTR;
    srw.attr.borderwidth := 1;
    srw.set_attr (0, srw.attr);
```

```
end if;
return (true);
end;
```

フォーマット・トリガーの例 (ラベルを非表示にする)

```
/* Suppose that you are building a master/detail report
** and, if no detail records are retrieved for a master
** record, you do not want the boilerplate labels to
** appear.To do this, you first create a summary
** column called MYCOUNT with a Function of Count in
** the source group of the master repeating frame.
** In the format trigger for the group frame that
** surrounds the detail repeating frame and its labels,
** you enter the following:
*/

function my_formtrig return BOOLEAN is
begin
if :mycount = 0 then
    return (false);
else
    return (true);
end if;
end;
```

フォーマット・トリガーの例 (値を非表示にする)

```
/* Suppose that you are building a salary report and
** you only want to see salaries above $2500. In the
** report summary of all salaries, however, you want all
** salaries to be included.To do this, you create a
** data model for the report without restricting the records
** retrieved.Then, you enter the following format trigger
** for the repeating frame.The report output will only
** show salaries greater than $2500, but the summary that
** calculates the sum of all salaries will include all the
** salaries retrieved from the database.
*/

function my_formtrig return BOOLEAN is
begin
if :sal > 2500 then
    return (true);
else
    return (false);
end if;
end;
```

フォーマット・トリガーの例 (フィールド間にカンマを配置する)

```
/* Suppose that you want to create a comma-separated
```

```
** list for the names of employees.First, you create
** a field inside a repeating frame with a Print
** Direction of Across.Next to the field, you create
** a boilerplate text object that contains a comma
** followed by a space.To ensure that the comma does
** not appear after the last name in the list, enter the
** following format trigger for the boilerplate object.
** LASTNAME is a summary column with a Source of ENAME,
** a Function of Last, and Reset At of Report.
*/

function my_formtrig return BOOLEAN is
begin
if :ename <> :lastname then
    return (true);
else
    return (false);
end if;
end;
```

フォーマット・トリガーの制限

注意: フォーマット・トリガーのPL/SQLは、Report Builderがレイアウト・オブジェクトをフォーマットするたびに実行されます。このため、フォーマット・トリガーには、フォーマット属性(カラーやハイライト表示など)を設定するPL/SQLプログラム単位のみが含まれている必要があります。トリガーがいつ、何回起動するかは予測できないので、表へのデータの挿入など、その他の操作を実行しないでください。たとえば、オブジェクトに「ページ保護」が設定されている場合、そのオブジェクトは、トリガーが起動される論理ページ上にフォーマットされない可能性があります。さらに、トリガーが2回以上実行される可能性があります。

- フォーマット・トリガーがレポートの最終ページのオブジェクトについて偽を返し、最終ページ上の他のオブジェクトが書式設定されない場合、最終ページはブランク・ページになります。
- PL/SQLコードに直接挿入するコメントには、PL/SQLコメント・デリミタを使用してください。
- フォーマット・トリガーでは、Report Builderの正しい頻度の列およびパラメータの値を読み込むことができますが(次のルールを参照)、これらの値を直接設定することはできません。たとえば、ある条件(IF:COUNT1=10など)でCOUNT1というパラメータの値を使用することはできませんが、代入文(:COUNT1=10など)に直接その値を設定することはできません(この制限は、フォーマット・トリガーからコールされたユーザー・イグジットにも適用されます)。PL/SQLのグローバル変数を使って、間接的に列やパラメータの値を設定することもできない点にも注意してください。これを行うと、予期しない結果になる可能性があります。
- オブジェクトのフォーマット・トリガー内の列または変数の頻度が、オブジェクトの頻度と異なる場合、参照できません。たとえば、マスター/ディテール・レポートを作成する場合、親グループの繰返し枠は、子グループの値に依存するフォーマット・トリガーを持つことはできません。各親に対して複数の子がある場合があります。そのため、親のレコード・レベ

ルでは、Report Builderは、どの子レコードを使ったらよいかを判断できません。また、ページ依存の列（つまり、「リセット位置」が「Page」に設定されている列）や、フォーマット・トリガーのページ依存の列に依存する列を参照することもできません。これを行うと、結果として依存性が循環するためです。つまり、ページ依存の列はレポートがフォーマットされるまで計算できず、レポートは、フォーマット・トリガーが実行されるまでフォーマットできません。

- フォーマット・トリガーがFALSEを戻し、オブジェクトがフォーマットされない場合、その他のオブジェクトは印刷されません。たとえば、繰返し枠がフォーマットされない場合、それが囲んでいるオブジェクト（フィールドまたはボイラプレート、枠、その他の繰返し枠）もフォーマットされません。
- 繰返し枠では、フォーマット・トリガーは繰返し枠の各インスタンスに対して実行されます。繰返し枠のすべてのインスタンス上で一度に動作するフォーマット・トリガーを作成するには、繰返し枠の周囲に枠を作成し、その枠のフォーマット・トリガーを入力します。フォーマット・トリガーが、論理ページの繰返し枠のすべてのインスタンスにFALSEを戻した場合、繰返し枠は、論理ページ上に領域を確保されず、その他のオブジェクトへのアンカーは縮小します（指定した場合）。
- フォーマット・トリガーのPL/SQLは、同じオブジェクトに対して首尾一貫した結果を戻す必要があります。たとえば、ある条件を満たした場合に、FALSEを戻すフォーマット・トリガーを持つ枠があるとします。枠が2ページにまたがっている場合、フォーマット・トリガーは、実際には2回起動します（枠がフォーマットされる各ページにつき1回ずつ）。PL/SQLの条件は、フォーマット・トリガーが起動する2回とも同じ結果を戻す必要があります。値が異なっていると、枠の一部しかフォーマットされません（最初のページの枠部分はフォーマットされても、2ページ目の枠部分はフォーマットされないなど）。
- マトリックス・レポートの繰返し枠にあるフォーマット・トリガーがFALSEを戻した場合、マトリックスの行または列はまったくフォーマットされません。たとえば、横方向繰返し枠のインスタンスがフォーマットされない場合、マトリックスの列はまったくフォーマットされません。
- マトリックスのセルを条件付きで変更する場合、マトリックス内のフィールドの周囲に枠を配置し、その枠に対してフォーマット・トリガーを使用してください。

アクション・トリガー

説明

アクション・トリガーは、ランタイム・プレビューアでボタンが選択されたときに実行されるPL/SQLプロシージャです。トリガーは、別のレポートを動的にコール（ドリルダウン）したり、他のPL/SQLを実行するのに使用できます。アクション・トリガーには、オブジェクト・ナビゲータまたはプロパティ・パレット（「PL/SQLトリガー」プロパティ）、PL/SQLエディタからアクセスできます。

定義レベル

ボタン

使用上の注意

- PL/SQLアクション・トリガーは、ライブ・プレビューアではテストできません。これは、ライブ・プレビューアのボタンがアクティブではないためです。ランタイム・プレビューアを使用する必要があります（ライブ・プレビューアから「表示」→「ランタイム・プレビューア」を選択）。
- PL/SQLインタプリタを使ってアクション・トリガーをデバッグできません。PL/SQLインタプリタはランタイム・プレビューアから使用できず、ライブ・プレビューアのボタンをアクティブにできないためです。別の方法として、アクション・トリガー・コードをレポート・トリガーに移動して、ライブ・プレビューアからそのコードをテストできます。

アクション・トリガーの例

```
/* When the button is clicked, the action trigger
** defined by this procedure displays the Parameter
** Form for a report named web4.rdf, then displays
** the resulting report output on your screen.
*/

procedure U_1ButtonAction is
begin
  srw.run_report('module=web4.rdf destype=Screen paramform=yes');
end;
```

参照カーソル問合せ

説明

参照カーソル問合せは、レポートに対するデータをフェッチするためにPL/SQLを使用します。参照カーソル問合せでは、カーソル変数からカーソル値を返すPL/SQL関数を指定します。

定義レベル

問合せ

失敗時

問合せにデータは返されません。

使用上の注意

- 参照カーソル問合せをデータ・リンクの子に指定すると、グループ間のみでリンクされます。

列間ではリンクされません。

- 参照カーソルをインプリメントするのにストアド・プログラム単位を使用すると、Oracleデータベースにプログラム単位で格納されるという利点もあります。

参照カーソルを使用したパッケージの例

```
/* This package spec defines a ref cursor
** type that could be referenced from a
** ref cursor query function.
** If creating this spec as a stored
** procedure in a tool such as SQL*Plus,
** you would need to use the CREATE
** PACKAGE command.
**/

PACKAGE cv IS
  type comp_rec is RECORD
    (deptno number,
     ename varchar(10),
     compensation number);
  type comp_cv is REF CURSOR return comp_rec;
END;
```

参照カーソルと関数を使用したパッケージ例

```
/* This package spec and body define a ref
** cursor type as well as a function that
** uses the ref cursor to return data.
** The function could be referenced from
** the ref cursor query, which would
** greatly simplify the PL/SQL in the
** query itself.If creating this spec
** and body as a stored procedure in a
** tool such as SQL*Plus, you would need
** to use the CREATE PACKAGE and CREATE
** PACKAGE BODY commands.
**/

PACKAGE cv IS
  type comp_rec is RECORD
    (deptno number,
     ename varchar(10),
     compensation number);
  type comp_cv is REF CURSOR return comp_rec;
  function emprefc(deptno1 number) return comp_cv;
END;
```

```
PACKAGE BODY cv IS
function emprefc(deptno1 number) return comp_cv is
    temp_cv cv.comp_cv;
begin
    if deptno1 > 20 then
        open temp_cv for select deptno, ename,
            1.25*(sal+nvl(comm,0)) compensation
        from emp where deptno = deptno1;
    else
        open temp_cv for select deptno, ename,
            1.15*(sal+nvl(comm,0)) compensation
        from emp where deptno = deptno1;
    end if;
    return temp_cv;
end;
END;
```

参照カーソル問合せの例

```
/* This ref cursor query function would be coded
** in the query itself.It uses the cv.comp_cv
** ref cursor from the cv package to return
** data for the query.
*/

function DS_3RefCurDS return cv.comp_cv is
    temp_cv cv.comp_cv;
begin
    if :deptno > 20 then
        open temp_cv for select deptno, ename,
            1.25*(sal+nvl(comm,0)) compensation
        from emp where deptno = :deptno;
    else
        open temp_cv for select deptno, ename,
            1.15*(sal+nvl(comm,0)) compensation
        from emp where deptno = :deptno;
    end if;
    return temp_cv;
end;
```

関数をコールする参照カーソル問合せの例

```
/* This ref cursor query function would be coded
** in the query itself.It uses the cv.comp_cv
** ref cursor and the cv.emprefc function from
** the cv package to return data for the query.
** Because it uses the function from the cv
```

```
** package, the logic for the query resides
** mainly within the package.Query
** administration/maintenance can be
** done at the package level (e.g.,
** modifying SELECT clauses could be done
** by updating the package).You could also
** easily move the package to the database.
** Note this example assumes you have defined
** a user parameter named deptno.
*/

function DS_3RefCurDS return cv.comp_cv is
  temp_cv cv.comp_cv;
begin
  temp_cv := cv.emprefc(:deptno);
  return temp_cv;
end;
```

After Parameter Formトリガー

説明

「AFTER PARAMETER FORMトリガー」は、ランタイム・パラメータ・フォームが表示された後に起動します。このトリガーから、パラメータへアクセスし、その値を確認できます。このトリガーは、パラメータ値を変更したり、エラーが発生した場合にランタイム・パラメータ・フォームに戻るのにも使用できます。このトリガーから、データ・モデルの列にはアクセスできません。ランタイム・パラメータ・フォームが非表示であっても、「AFTER PARAMETER FORMトリガー」は起動します。したがって、このトリガーは、コマンド・ライン・パラメータやその他のデータの妥当性チェックに使用できます。

定義レベル

レポート

失敗時

ランタイム・パラメータ・フォームに戻ります。フォームが非表示の場合は、レポートを実行した場所に戻ります。

After Report トリガー

説明

「After Reportトリガー」は、ランタイム・プレビューを終了した後、あるいはレポート出力がファイル、プリンタ、メールIDなどの指定した宛先に送られた後に起動します。このトリガーは、表の削除などの完了した初期処理をクリーン・アップするのに使用できます。ただし、このトリガーは、レポートが正常に完了したかどうかにかかわらず、必ず起動することに注意してください。

定義レベル

レポート

失敗時

レポートが完了しているため、フォーマットの実行には影響しません。しかし、メッセージを呼び出して、レポートが正しく実行されなかったことを示すことができます。たとえば、「Before Reportトリガー」の変数にシステム時間を入れ、「After Reportトリガー」のシステム時間と比較できます。レポートの実行にかかった時間が一定の時間を超えた場合に、エラーを呼び出すことができます。

使用上の注意

- ライブ・プレビューでは、「After Reportトリガー」は起動されません。

Before Parameter Form トリガー

説明

「BEFORE PARAMETER FORMトリガー」は、ランタイム・パラメータ・フォームが表示される前に起動します。このトリガーから、パラメータおよびPL/SQLグローバル変数、レポート・レベルの列の値にアクセスし、変更できます。ランタイム・パラメータ・フォームが非表示であっても、このトリガーは起動します。したがって、このトリガーは、コマンド・ライン・パラメータの妥当性チェックに使用できます。

定義レベル

レポート

失敗時

エラー・メッセージを表示してから、レポートを実行した場所に戻ります。

Before Reportトリガー

説明

「BEFORE REPORTトリガー」は、問合せが解析されデータがフェッチされた後、レポートが実行される前に起動します。

定義レベル

レポート

失敗時

エラー・メッセージを表示してから、レポートを実行した場所に戻ります。

Between Pagesトリガー

説明

「BETWEEN PAGESトリガー」は、レポートの最初のページを除く各ページがフォーマットされる前に起動します。このトリガーは、カスタマイズされたページのフォーマットにも使用できます。ランタイム・プレビューまたはライブ・プレビューでは、このトリガーは、あるページに初めて進むときにのみ起動します。後からそのページに戻っても、トリガーは起動しません。

定義レベル

レポート

失敗時

トリガーがFALSEを戻したページに進もうとすると、エラー・メッセージが表示されます。FALSEが戻されたページに続くページはフォーマットされません。トリガーが最後のページでFALSEを戻しても、レポートのフォーマットは完了しているので何も起こりません。「BETWEEN PAGESトリガー」は、最初のページより前には起動しません。トリガーが最初のページでFALSEを戻すと、最初のページは表示されますが、2ページ目に進もうとすると、エラー・メッセージが表示されます。

プロパティ

Oracle8使用上の注意

列オブジェクトとREF列

- 属性列または列オブジェクトのメソッドを参照するには、表の別名または相関変数で名前を修飾する必要があります。たとえば、次のように指定します。

```
select e.empno, e.address.street, e.address.city from emp e
```

- 列オブジェクトやREF列を全体として移動することも、個々の属性列を移動することもできます。関連する列オブジェクトの外部に属性列を移動すると、その列オブジェクトから切り離されて、他の列と同じように扱われます。列オブジェクトとの関連付けを示すために名前（ADDRESS_ZIPなど）のみが更新されます。
- 列オブジェクトをレイアウト・オブジェクトのソースにすることはできませんが、その属性列をレイアウト・オブジェクトのソースとして使用することはできます。たとえば、STREET、CITY、STATEおよびZIPで構成されるADDRESSという名前の列オブジェクトがあるとします。ADDRESSはフィールドのソースになりませんが、STREET、CITY、STATEまたはZIPはフィールドのソースになります。
- 列オブジェクトやREF列は、その属性列の1つが存在する最下位のグループに存在する必要があります。これは、列オブジェクトやREF列の頻度が、最高頻度の属性列と同じ必要があるからです。列オブジェクトやREF列が属性列の1つより上のグループにあると、頻度が低くなりすぎてエラーが発生します。たとえば、列オブジェクトADDRESSがG_MASTERという名前のグループにあり、その属性列の1つがZIPであるとします。G_DETAILというG_MASTERの子にZIPをドラッグする場合は、ADDRESSもG_DETAILに移動してください。
- 列オブジェクトをバインド変数（つまり:column_name）として使用することはできませんが、その属性列をバインド変数として使用することはできます。
- 列オブジェクトやREF列でブレイクすることはできません。したがって、列オブジェクトやREF列をブレイク・グループの唯一の列にすることはできません。ただし、列オブジェクトの属性列でブレイクすることはできます。たとえば、列オブジェクトとそのすべての属性列をブレイク・グループにドラッグし、必要な数の属性列にブレイク順序を割り当てることができます。
- データ・モデルのリンクで列オブジェクトやREF列を使用することはできません。リンクで使用できるのは、その個々の属性列のみです。たとえば、STREET、CITY、STATEおよびZIPで構成されるADDRESSという名前の列オブジェクトがあるとします。ADDRESSの任意の属性列（CITYなど）を使って、リンクを作成できます。

サマリー、式およびブレースホルダ

- 式列またはブレースホルダ列のデータ型は、Oracle7ビルトイン・データ型（CHAR、VARCHAR

など)の1つにしてください。Oracle8で新しく追加されたデータ型は、まだサポートされていません。

- 列オブジェクトやREF列をソースとして持つサマリー列では、First、LastおよびCountファンクションのみを使用できます。

可変配列およびネストした表

- 可変配列およびネストした表のタイプはまだサポートされていません。問合せでこれらのタイプの列オブジェクトを選択すると、データ・モデル表示の列オブジェクトの横に、サポートされていないことを示す特殊なアイコンが表示されます。

ラージ・オブジェクト(LOB)

- LOBはまだサポートされていません。問合せでLOBを選択すると、データ・モデル表示のLOBの横に、サポートされていないことを示す特殊なアイコンが表示されます。

プロパティ・パレットについて

プロパティ・パレットには、オブジェクト・プロパティの表示および検索、設定ができる一括表示型インタフェースがあります。プロパティ・パレットには、次の機能があります。

- 拡張と縮小が可能なノード
- その場でのプロパティ編集
- 検索機能
- 複数選択
- 複合プロパティ・ダイアログ
- プロパティ・パレットの複数インスタンスを起動する能力

エディタまたはオブジェクト・ナビゲータでオブジェクトを選択すると、プロパティ・パレットが更新され、選択したオブジェクトのプロパティが表示されます。デフォルトでは、異なるオブジェクトを選択するたびに、パレットに表示されるプロパティ・リストが同期化されます。プロパティ・パレットのツールバーで「**固定解除**」ボタンをクリックして、特定のパレットの同期化をオンまたはオフに切り替えられます。

オブジェクトのプロパティを比較する必要がある場合、「**停止**」ボタンをクリックして、現行のプロパティ・パレットを固定し、必要に応じてその他のプロパティ・パレットを起動します。

プロパティ・パレットには、次の機能があります。

プロパティ・リスト

プロパティ・リストは、プロパティ名とプロパティ値の2列のリストです。オブジェクトが選択されていない場合、リストにプロパティは表示されません。プロパティは、「一般情報」プロパティや「フィールド」プロパティのような、機能ヘッダーまたはオブジェクト・ヘッダーのもとにグループ化されます。クリックするか、上/下の矢印キーによってナビゲートし、リストのプロパティを選択します。プロパティ・リスト・ペインで、プロパティを選択し、必要に応じて入力またはダブルクリックして、プロパティを設定することができます。

名前

「名前」フィールドには、ナビゲータまたはエディタで現在選択されているオブジェクトの名前が表示されます。「名前」フィールドは、オブジェクトが選択されていない場合は表示されません。同時に複数のオブジェクトが選択されている場合、名前のリストがプロパティ・パレットの一番上に表示されます。

プロパティ制御の設定

プロパティ・リストでプロパティが選択されている場合、プロパティの横に次のようにテキスト・フィールドまたはポップリスト、コンボ・ボックス、「...」ボタンが表示されます。

現行のプロパティをテキスト値の入力によって設定できる場合、**テキスト・フィールド**が表示されます。さらに長いテキスト値を必要とするプロパティの場合は、テキスト・フィールドの横にあるアイコン・ボタンを押して、モーダル・エディタを起動します。

プロパティに、TRUE/FALSEまたは一連の固定有効値のいずれかがある場合、**ポップリスト**が表示されます。ダブルクリックを繰り返して、値を循環させることができます。

プロパティを、値の入力またはポップリストからの選択によって設定できる場合、**コンボ・ボックス**が表示されます。

プロパティに、さらに複雑な値が必要な場合、「...」ボタンが表示されます。このボタンを選択すると、必要な情報を入力するためのダイアログが起動されます。

プロパティ・パレットは、作成したオブジェクトのプロパティを文書で設定する場所です。

プロパティ・パレットでの複数選択について

ナビゲータまたはエディタで、[Shift]を押しながらクリックするか、[Ctrl]を押しながらクリックすると、複数のオブジェクトを同時に選択できます。複数のオブジェクトが選択されると、オブジェクト名のリストがプロパティ・パレットの一番上に表示されます。

2つ以上のオブジェクトが選択されている場合、プロパティ・パレットのツールバーの「論理積/論理和」コマンドによって、プロパティ・リストにどのプロパティが表示されるかが決まります。論理積と論理和を切り替えると、プロパティ・パレットに表示されるプロパティのリストは変更されますが、プロパティの設定には影響ありません。

論理積 デフォルト。すべての選択されたオブジェクトに共通のプロパティのみが表示されます。

論理和 選択された各オブジェクトのプロパティがすべて表示されます。

複数選択の2つ以上のオブジェクトに共通するプロパティは1度のみ表示され、プロパティの設定は次のように表示されます。

- 選択されたオブジェクトがすべて共通のプロパティを持ち、その設定が同じである場合、その共通の設定が表示されます。
- 共通のプロパティを持つオブジェクト間でその設定が異なる場合、文字列*****が表示されます。

プロパティ・パレットのツールバー・コマンドについて

ツールバーを使用すると、コマンドに迅速にアクセスできます。実行するコマンドのアイコンをクリックします。

継承	テンプレートの「デフォルト・セクション」および「見出し」、「ボディ」、「集計」のみに適用されます。現行のプロパティをデフォルトに設定します。複数のプロパティを選択して、「継承」をクリックすることができます。
ローカライズ	テンプレートの「デフォルト・セクション」および「見出し」、「ボディ」、「集計」のみに適用されます。現行のプロパティをローカルにします（つまり、継承しません）。複数のプロパティを選択して、「ローカライズ」をクリックすることができます。
「検索」フィールド	プロパティ・パレットで検索する文字列です。
順方向検索 	「検索」フィールドの文字列を順方向検索します。
逆方向検索 	「検索」フィールドの文字列を逆方向検索します。
論理積/論理和 	プロパティ・パレットの結合表示モードと分岐表示モードの切替えを行います。同時に複数のオブジェクトが選択されている場合（複数選択）、このオプションによってどのプロパティが表示されるかが決まります。分岐モード（デフォルト）では、選択されたオブジェクトに共通のプロパティのみが表示されます。結合モードでは、現在選択されているすべてのオブジェクトのプロパティが表示されます。
固定/固定解除 	プロパティ・パレットの同期化のオンとオフを切り替えます。「停止」がオフ（デフォルト）の場合、ナビゲータおよびその他のウィンドウで選択するオブジェクトのプロパティを表示するように、プロパティ・リストが更新されます。「停止」がオンの場合、プロパティ・リストは固定されて更新されないため、その他のプロパティ・リストと比較することができます。

プロパティ・パレットの表示

オブジェクトのプロパティ・パレットを表示するには、次のいずれかを行います。

- オブジェクトをダブルクリックする。2つ目のプロパティ・パレットを起動するには、現行のプロパティ・パレットを固定し、オブジェクト・アイコンをダブルクリックしてください。
- オブジェクトをクリックし、「ツール」→「プロパティ・パレット」を選択します。
- ポップアップ・メニュー（Windowsで右クリック）から「プロパティ・パレット」を選択します。
- オブジェクト・ナビゲータで、オブジェクトのプロパティ・アイコンをダブルクリックします。

プロパティ・パレットのクローズ

プロパティ・パレットをクローズするには:

右上の隅にあるクローズ・ボックスをクリックします。

プロパティ・パレットを使用したプロパティの設定

プロパティ・パレットを使用してプロパティを設定するには:

- 1 「プロパティ・パレットを表示」します。
- 2 設定するプロパティを選択します。

プロパティを選択するには、クリックするか、上/下の矢印キーによってナビゲートします。プロパティが選択されると、テキスト・フィールドまたはポップリスト、コンボ・ボックス、「...」ボタンが、そのプロパティの横に表示されます。

- 3 プロパティを希望どおりに設定します。

複数選択でのプロパティの設定

複数選択してプロパティを設定するには:

- 1 オブジェクト・ナビゲータまたはエディタで設定するプロパティを持つオブジェクトを選択します。オブジェクトは、タイプが異なるものでも、異なるレポートにあるものでもかまいません。
- 2 「ツール」→「プロパティ・パレット」を選択します。
- 3 プロパティ・パレットで、すべてのオブジェクトのすべてのプロパティを参照する場合は、 (「論理和」ボタン) をクリックします。または、オブジェクトに共通なプロパティのみを参照する場合は  (「論理積」ボタン) をクリックします。
- 4 プロパティを希望どおりに設定します。

プロパティの設定に加えた変更は、現在選択されているオブジェクトのうちで、そのプロパティが適用されるすべてのオブジェクトに適用されます。

複数選択でのプロパティの設定の例

レポートには複数のフィールドが組み込まれており、各フィールドには日付が表示されます。各フィールドの「データ型」プロパティはDATEであるため、日付はデフォルトのOracle書式YY-MM-DDで表示されます。

アプリケーションで全体を通してこれとは異なる日付書式を使用するには、日付を表示する各フィールドの「書式マスク」プロパティを設定する必要があります。各フィールドの「書式マスク」プロパティを個々に設定するのではなく、すべての項目を選択してから「書式マスク」プロパティを設定すれば、選択した各項目の書式マスクが一度に変更されます。

オブジェクトどうしのプロパティの比較

オブジェクトどうしのプロパティを比較するには:

- 1 オブジェクト・ナビゲータまたはエディタで最初のオブジェクトをダブルクリックし、プロパティ・パレットにそのオブジェクトのプロパティを表示します。
- 2 プロパティ・パレットでをクリックして、同期化をオフにし、プロパティ・パレットを「停止」します。
- 3 オブジェクト・ナビゲータで、2番目のオブジェクトをクリックしてから、ポップアップ・メニューから「**プロパティ・パレット**」を選択します。

2つ目のプロパティ・パレットが表示されます。2つ目のウィンドウが最初のウィンドウの上に重なったときは、ドラッグして最初のウィンドウの横に並べます。

プロパティ・リスト

追加

追加の属性 (HTML)

追加のハイパーリンク属性

フォーム後型

フォーム後値

ページ後型

ページ後値

レポート後型

レポート後値

サマリーをフィールドの位置に合わせる

整列

アプリケーション・コマンド・ライン (PDF)

バックグラウンド・カラー

基本印刷オン

フォーム前型

フォーム前値

ページ前型

ページ前値

レポート前型

レポート前値

フィールドとラベル間 (水平)

枠とフィールド間 (水平)

枠とフィールド間 (垂直)
マスターとディテール間 (水平)
マスターとディテール間 (垂直)
ページと枠間 (水平)
ページと枠間 (垂直)
同等枠間 (水平)
同等枠間 (垂直)
ブックマーク
境界線
ブレイク順序
文字揃え
チャート列
チャート・ファイル名
チャート・ハイパーリンク
チャート・パラメータ
チャート問合せ
子列
子の枠の割合 (%)
子の枠の種類
子オブジェクト名
子問合せ
水平方向に縮小
垂直方向に縮小
列モード
列タイプ

コメント
計算位置
条件
条件付き書式
XMLタグを含む
クロス積グループ
破線
データ型 (列)
データ型 (パラメータ)
データ型 (パラメータ・フォーム・フィールド)
日付揃え
キャラクタ単位で設計
方向
配布 (セクション)
配布 (レポート)
ホスト・メニュー項目を使用不可にする
画面分割キーを使用不可にする
ズーム・キーを使用不可にする
表示名
枠のバックグラウンド・カラー
枠のフォアグラウンド・カラー
枠のパターン
XML出力から除外
外部問合せソース・ファイル
行当りのフィールド数

ファイル形式
塗りパターン
フォント
フォアグラウンド・カラー
フィルタ・タイプ
書式マスク
フォーマット・トリガー
ファンクション
高さ (パラメータ・フォーム・ウィンドウ)
高さ (セクションの物理ページ)
第1列を隠す
枠間の水平間隔
水平拡張度
ページ当りの水平パネル
水平繰返し枠
ハイパーリンク (HTML/PDF)
ハイパーリンクの宛先
アイコン名
イメージ
ビットマップ・オブジェクトを含む
境界線を含む
初期値
入力マスク
フィールド間 (水平)
フィールド間 (垂直)

枠間（水平）
枠間（垂直）
位置揃え
アンカー・オブジェクトと連動
ラベルの型
枠付きの行伸長
値リスト
最大水平ボディ・ページ
最大垂直ボディ・ページ
ページ当りの最大レコード数
最大取出し行
最小孤立行
最小孤立レコード数
マルチメディア列の型
マルチメディア列
マルチメディア・ファイル・タイプ
マルチメディア・ファイル
名前
ページ数
レコード数
数値揃え
方向
外部XMLタグ
外部XMLタグ属性
後で改ページ

前で改ページ

ページ・ナビゲーション・コントロール・タイプ

ページ・ナビゲーション・コントロール値

ページ番号付け

ページ保護

パネル印刷順序

親列

親の枠の割合 (%)

親の枠の種類

親グループ

親オブジェクト名

PL/SQLフィルタ

PL/SQL式 (式およびブレースホルダ列)

PL/SQL文

PL/SQLトリガー

フィールドの上にラベルを置く

プレビューアのヒント行のテキスト

プレビューアのステータス行のテキスト

印刷方向

印刷オブジェクト・オン

後のプリンタ・コード

前のプリンタ・コード

積順序

ファイルからの読み込み

削除

レポート列 (チャート列)
レポート列 (チャート・パラメータ)
レポート・グループ
レポート高
レポート幅
リセット位置
ロール名
事前定義済み値に対するリストを制限
SELECT文/スタティック値リスト
ブレイク順序の設定
ソース・データ型
ソース・ファイルの形式
ソース・ファイル名
ソース (サマリー列)
ソース (繰返し枠)
ソース (パラメータ・フォーム・フィールド)
ソース (フィールド)
SQL句
SQL問合せ文
ズームで起動
スタイル
プレビューア・タイトルの抑止
テキスト (ボタン)
テキスト (テンプレート)
文字色

タイトル
型 (ボイラープレート)
型 (ボタン)
型 (問合せ)
サイズの単位
プレビューアのヒント行を使用
プレビューアのステータス行を使用
垂直間隔の使用
妥当性チェック・トリガー
NULL時の値
値
枠間の垂直間隔
垂直拡張度
ページ当りの垂直パネル
垂直繰返し枠
可視
幅 (列)
幅 (パラメータ)
幅 (パラメータ・フォーム・ウィンドウ)
幅 (セクションの物理ページ)
XML Prologタイプ
XML Prolog値
XMLタグ
XMLタグ属性

アンカー・プロパティ

子の枠の割合 (%)

子の枠の種類

子オブジェクト名

水平方向に縮小

垂直方向に縮小

コメント

名前

親の枠の割合 (%)

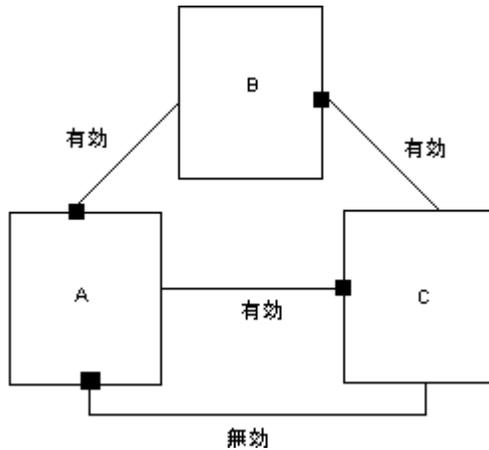
親の枠の種類

親オブジェクト名

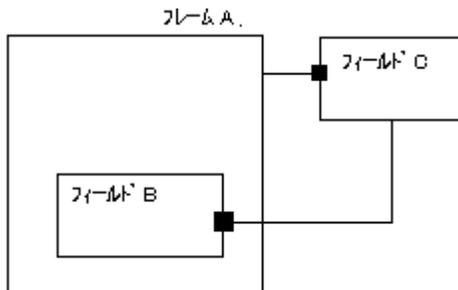
アンカーの制限

- オブジェクトは、ただ1個に限り他のオブジェクトにアンカーされることがあります。
- マトリックス・オブジェクトおよびアンカー、マージンは、何に対してもアンカーできません(つまり、これらはアンカーについて親オブジェクトにも子オブジェクトにもなれません)。
- あるマトリックスの垂直繰返し枠または水平繰返し枠となっている繰返し枠は、別のオブジェクトにはアンカーできませんが、その他のオブジェクトはこれらの繰返し枠にアンカーできます(つまり、これらの繰返し枠はアンカーについて親オブジェクトにはなれますが、子オブジェクトにはなれません)。
- 非表示オブジェクト(「可視」が「いいえ」に設定されているオブジェクト)には、何もアンカーできません。
- アンカーを移動すると、アンカーされている2つのオブジェクトも一緒に移動します。
- オブジェクトの依存関係が循環した形になるようにオブジェクト同士をアンカーすることはできません。たとえば、次のように指定します。
 - オブジェクトAとオブジェクトBが互いにアンカーされており、オブジェクトAが親であるとし、オブジェクトBは、オブジェクトAとオブジェクトB間の別のアンカーについて親にはなれません。さらに、オブジェクトCがオブジェクトAへのアンカーについて親であ

る場合、オブジェクトBはオブジェクトCへのアンカーについて親にはなれません。



- 枠Aの中にフィールドBがあり、レイアウト・モデルでフィールドBが枠Aの上にあるとします。また、枠Aが、枠Aの外側にあるフィールドCにアンカーされており、フィールドCが親であるとします。この場合、フィールドCは、フィールドBへのアンカーについて親にはなれません。それは、フィールドBが枠Aの中にあるため、フィールドCが親の場合はフィールドBをフィールドCにアンカーできないからです。



- アンカーをコピーするには、アンカーとアンカーされている2つのオブジェクトを選択してください。アンカーのみを選択した場合は、何もコピーされません。アンカーと一方のオブジェクトを選択した場合は、選択されたオブジェクトのみがコピーされます。
- アンカーのサイズは変更できません。
- アンカーは、必ずアンカーされるオブジェクトより上にしてください（つまり、アンカーは、親オブジェクトと子オブジェクトよりも上のレイヤーに存在する必要があります）。Report Builderでは、親オブジェクトと子オブジェクトより下のレイヤーにアンカーを移動できないようになっています。
- 「配置」メニューの「オブジェクト整列」および「オブジェクト・サイズ設定」をアンカー

に対して使用することはできません。

子の枠の割合 (%)

説明

「子の枠の割合 (%)」プロパティは、アンカーが配置されている子オブジェクトの枠の縦または横（上から下または左から右）のパーセンテージを示します。この設定によって、オブジェクトが挿入されたときの枠上のアンカーの位置が決まります。

値

0 ~ 100の整数

適用対象

アンカー

子の枠の種類

説明

「子の枠の種類」プロパティは、アンカーが配置されている子オブジェクトの枠を示します。

値

上、下、左、右

適用対象

アンカー

子オブジェクト名

説明

「子オブジェクト名」プロパティは、子オブジェクトの名前を示します。このフィールドは、読み込み専用です。

適用対象

アンカー

水平方向に縮小

説明

「水平方向に縮小」プロパティは、親オブジェクトがなんらかの理由で印刷されない場合（たとえば、「印刷オブジェクト・オン」または「基本印刷オン」、「フォーマット・トリガー」プロパティによって）、アンカーの幅が0（ゼロ）である必要があるかどうかを示します。親オブジェクトが印刷されない場合、子オブジェクトは、親オブジェクトが印刷されるはずであった領域に水平移動します。

適用対象

アンカー

垂直方向に縮小

説明

「垂直方向に縮小」プロパティは、親オブジェクトがなんらかの理由で印刷されない場合（たとえば、「印刷オブジェクト・オン」または「基本印刷オン」、「フォーマット・トリガー」プロパティによって）、アンカーの高さが0（ゼロ）である必要があるかどうかを示します。親オブジェクトが印刷されない場合、子オブジェクトは、親オブジェクトが印刷されるはずであった領域に垂直移動します。

適用対象

アンカー

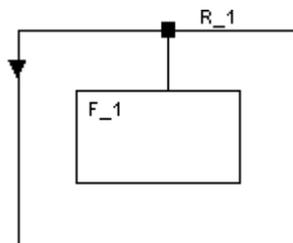
フィールドを水平に中央揃えする例

「水平拡張度」が「可変」に設定された繰返し枠内で、フィールドを水平に中央揃えするとします。また、このフィールドを繰返し枠の枠左部から一定の間隔をおいて固定するとします。フォーマットされた繰返し枠の大きさがわからないので、フィールドを中央揃えするにはアンカーを使用してください。

繰返し枠内でフィールドを水平に中央揃えするには、次の手順に従います。

- 繰返し枠上部中央のX座標がフィールド上部中央のX座標と一致するように、繰返し枠の中央にフィールドを配置する。下の図を参照してください。
- フィールドの上部中央から繰返し枠上部中央までアンカーを描く。

これで、このフィールドは繰返し枠内で水平に中央揃えされます。

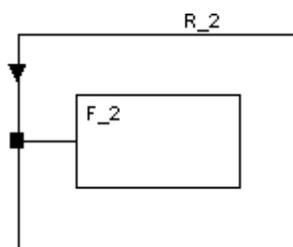


フィールドを垂直に中央揃えする例

「垂直拡張度」が「可変」に設定された繰返し枠内で、フィールドを垂直に中央揃えするとします。また、このフィールドを繰返し枠の枠左部から一定の間隔をおいて固定するとします。フォーマットされた繰返し枠の大きさがわからないので、フィールドを中央揃えするにはアンカーを使用してください。

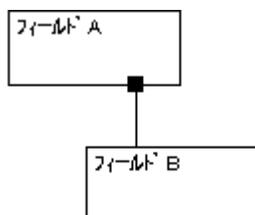
繰返し枠内でフィールドを垂直に中央揃えするには、フィールド枠の左側から繰返し枠左側までアンカーを描きます。

これで、このフィールドは繰返し枠内で垂直に中央揃えされます。



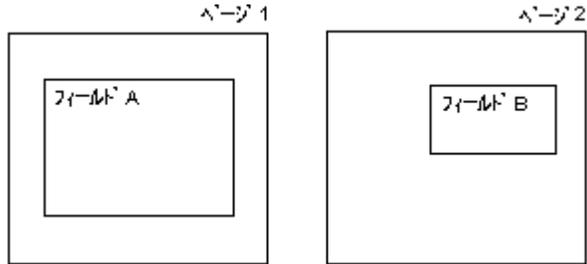
垂直方向に縮小する例（フィールド）

フィールドBをフィールドAにアンカーし、「垂直方向に縮小」を「はい」に設定したとします。



レポートの最初の論理ページにフィールドAが印刷され、ページの領域をフィールドAが大量に使用しているため、フィールドBが同じ論理ページ内に収まらないとします。そこで、フィール

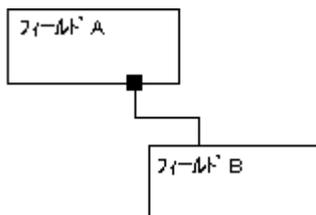
ドBは、フィールドAが印刷されない論理ページの2ページ目に印刷されます。「垂直方向に縮小」が「はい」に設定されているので、アンカーはY軸方向に縮小します。フィールドBは、フィールドAが論理ページに実際に印刷されなくても、フィールドAに対するX軸方向の相対的位置を維持します。



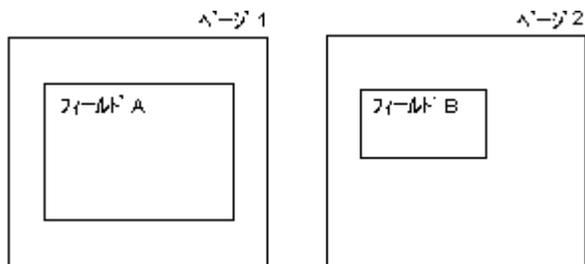
水平および垂直方向に縮小する例（フィールド）

フィールドBがフィールドAにアンカーされているときに、フィールドBを水平と垂直の双方向に移動して、フィールドAが印刷されるはずであった領域に配置するとします。そのためには、次の手順に従います。

- フィールドBの上部からフィールドAの下部にアンカーを描きます。
- 「水平方向に縮小」と「垂直方向に縮小」の両方を「はい」に設定します。



レポートの最初の論理ページにフィールドAが印刷され、ページの領域をフィールドAが大量に使用しているため、フィールドBは同じページ内に収まりません。そこで、フィールドBは、フィールドAが印刷されない論理ページの2ページ目に印刷されます。アンカーは縮小し、フィールドAが描画として印刷されるはずであった位置にフィールドBが移動します。



フィールドAとBが同一の論理ページに印刷される場合、アンカーは縮小されず、これらのフィールドは互いに相対的な位置を維持します。

親の枠の割合 (%)

説明

「親の枠の割合 (%)」プロパティは、アンカーが配置されている親オブジェクトの枠の縦または横 (上から下または左から右) のパーセンテージを示します。この設定によって、オブジェクトが挿入されたときの枠上のアンカーの位置が決まります。

値

0 ~ 100の整数

適用対象

アンカー

親の枠の種類

説明

「親の枠の種類」プロパティは、アンカーが配置されている親オブジェクトの枠を示します。

値

上、下、左、右

適用対象

アンカー

親オブジェクト名

説明

「親オブジェクト名」プロパティは、親オブジェクトの名前を示します。このフィールドは、読み専用です。

適用対象

アンカー

ボイラープレート・プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

コメント

HTMLタグを含む

表示名

フォーマット・トリガー

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アンカー・オブジェクトと連動

枠付きの行伸長

最小孤立行

名前

後で改ページ

前で改ページ

ページ保護

印刷オブジェクト・オン

後のプリンタ・コード

前のプリンタ・コード

ソース・ファイルの書式

ソース・ファイル名

型

垂直拡張度

ボイラープレートの制限

- ボイラープレート・テキストの各パラグラフのサイズは64Kバイト以下、もしくはパラグラフの数は64,000個以下にしてください。この制限は、オペレーティング・システムによって異なることがあります。
- ボイラープレートは、ボイラープレートが含まれる繰返し枠の前にしてください。

注意: ボイラープレート・テキストのサイズを変更する場合は、そのテキストがオブジェクト内に必ず収まるようにしてください。フォントのディセンダ (g、qなどの文字で並び線より下の部分のために残されたスペース) が収まらない場合、そのテキスト行はレポート・エディタには表示されませんが、「水平拡張度」と「垂直拡張度」の設定によっては、実行されたレポートの出力に表示されないことがあります。

HTMLタグを含む

説明

「HTMLタグを含む」プロパティは、ボイラープレートまたはフィールド・オブジェクトがHTMLタグをインクルードしていることを示しています。Report Builderでは、オブジェクトを垂直にも水平にも固定でき、標準的なテキスト・オブジェクトとしてそのオブジェクトをフォーマットします (つまり、どのフィールド参照も解析されます)。パラメータとしてオブジェクトの幅と高さを渡す必要がある場合は、例で示したObjectWidthとObjectHeight変数が使用されます。

適用対象

ボイラープレートとフィールド

必須/オプション

オプション

デフォルト

いいえ

ObjectWidthとObjectHeightのHTMLへの出力例

ボイラープレート・テキスト・オブジェクトには、オブジェクトの幅と高さをパラメータとして使用するNervousText.classという次のJava Appletを入力することができます。

```
<base href=http://cagney.uk.oracle.com/java/NervousText/>
```

```
<applet code="NervousText.class" width=&ObjectWidth
height=&<ObjectHeight>>
<param name=text value="&deptno">
</applet>
```

HTMLへ出力する場合は次のように入力します。

```
<base href=http://cagney.uk.oracle.com/java/NervousText/>
<applet code="NervousText.class" width=84 height=72>
<param name=text value="10">
</applet>
```

ボイラープレート・テキスト・オブジェクト例(ハイパーリンク)

この例は、ハイパーリンクとしてタグ付けしたボイラープレート・テキスト・オブジェクトです。

ボイラープレート・テキスト・オブジェクトとして次のものを作成します。

```
<a href=http://your_webserver/reports/my_report.html><img
src=oracle.gif> </a>
```

ボイラープレート・オブジェクトを選択し、「プロパティ・パレット」をオープンします。「Webの設定」で、「HTMLタグを含む」プロパティを「はい」に設定します。

「HTMLタグを含む」の制限

- 「HTMLタグを含む」の使用は、テキスト・ボイラープレートおよびテキストのソース・ファイル書式を持つリンク・ファイル・ボイラープレート、キャラクタのソース・データ型を持つフィールドに限られます。

最小孤立行

説明

「最小孤立行」プロパティは、オブジェクトがプリントを開始する論理ページ上に現れるボイラープレートまたはフィールド・テキストの最小行数です。このプロパティに設定された行数が論理ページに収まらない場合は、テキストのすべての行が次のページに移動されます。

注意: このプロパティは、ボイラープレートまたはフィールド・オブジェクト内のテキストの最初の数行にのみ適用されます。これによって、個々のパラグラフの孤立保護が行われるわけではありません。

値

0 ~ 32Kの正の整数

適用対象

ボイラープレートとフィールド

必須/オプション

オプション

デフォルト

0 (次のページに続く前に、論理ページに収まるテキストが印刷されます。)

「最小孤立行」の制限

- 「最小孤立行」プロパティの使用は、テキスト・ボイラープレートおよびテキストのソース・ファイル書式を持つリンク・ファイル・ボイラープレート、キャラクタのソース・データ型を持つフィールドに限られます。

最小孤立行の設定例 (設定値1の場合と設定値2の場合)

「最小孤立行」を1に設定すると、図1のような状態になります。つまり、ある論理ページの最終行にフィールドの第1行が印刷され、残りは次の論理ページに印刷されます。「最小孤立行」を2 (またはそれ以上) に設定すると、図2のように、フィールドの1行分のみ論理ページの最後に印刷されません。

最小ウィンドウ行数 = 1

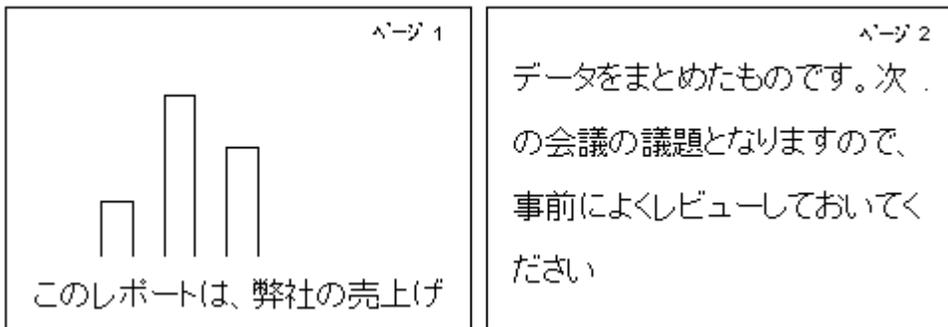
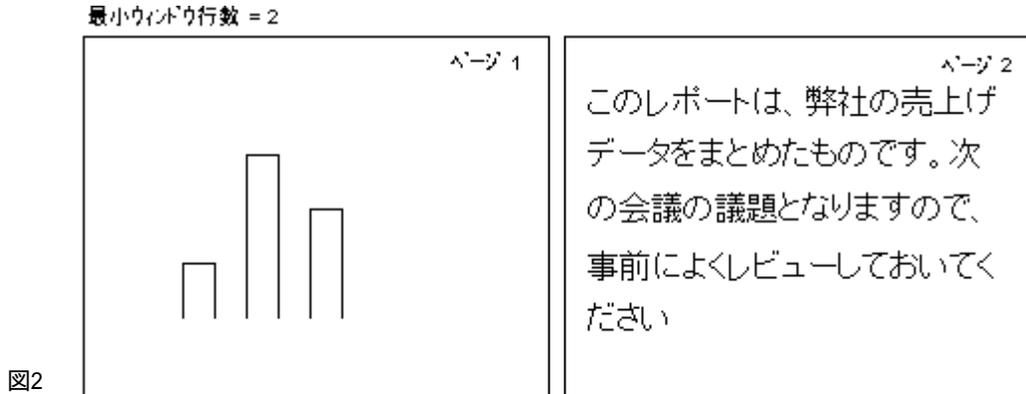


図1



ソース・ファイルの書式

説明

「ソース・ファイルの書式」プロパティは、ボイラープレートがリンクしている外部ファイルの書式です。

値

テキスト	ASCII形式のテキストです。
イメージ	ビットマップ・イメージです。
CGM	CGM形式での線の描画です。
Oracle Drawing Format	Oracle形式での線の描画です。
イメージURL	イメージのURLリンクです。

注意: 他の全てのイメージ書式は、リリース2.0内にレポートをオープンするときのために提供されます。リリース2.0内にレポートをオープンまたは実行する場合には、ボイラープレートの特別なイメージ書式を選択してください。それ以外の場合は、上記リストの中から選択できます。

適用対象

リンク・ファイル

必須/オプション

必須

ソース・ファイル名

説明

ソース・ファイル名 プロパティは、ボイラープレートがリンクしている外部ファイルまたはイメージURLの名前です。ファイルにリンクしておくこと、変更される可能性があるファイルの内容を組み込む場合に便利です。次のいずれかの処理を行うと、ボイラープレートは自動的に更新され、ファイルの最新の内容が反映されます。

- プロパティ・パレットの確定
- オブジェクトのペースト
- オブジェクトのインポート（Oracle Formatの場合）
- レポートのオープン
- レポートの実行

ボイラープレート・オブジェクトにリンクされたファイルの内容は、レポート・エディタに表示されますが、レポート・エディタ内では編集できません。

値

1K以下の長さの有効なファイル名またはイメージURLを入力します。ファイル名の前にパスを付けることができます。ファイル名の前にパスが指定されていない場合は、Report Builderのファイル・パス検索順序に従って、ファイルが検索されます。

適用対象

リンク・ファイル

必須/オプション

オプション

デフォルト

ブランク

使用上の注意

- レポートの移植を確保するには、REPORTS60_PATHというReport Builder固有の環境変数を使用します。この変数を使用して、レポート内で使用される外部ファイル（外部問合せ、ボイラープレート、PL/SQLなど）をReport Builderが検索するためのデフォルト・ディレクトリを指定できます。これによって、レポートにディレクトリ・パスをハードコードする必要がなくなります。

- 「ファイルからの読み込み」プロパティを使用して、レポートにファイルを組み込むこともできます。

「ソース・ファイル名」の制限

- 指定するファイルの最大サイズは、オペレーティング・システムによって異なります。
- ソース・ファイル名は、オペレーティング・システムに依存します。レポートを別のオペレーティング・システムに移動すると、ファイル名を更新する必要があることがあります。
- このフィールドは、リンク・ファイル・ツールによって作成されたオブジェクトにのみ表示されます。
- レイアウト・オブジェクトとそのプロパティを含むファイル（つまり、エクスポートされたレイアウト・オブジェクトを含むファイル）を指定した場合は、このファイルは名前のないグラフィック・オブジェクトとして扱われます（つまり、オブジェクトのプロパティは無視されます）。オブジェクトのプロパティを使用する場合、ファイルをリンクするかわりにそれをインポートする必要があります。

型

説明

型プロパティは、ボイラープレート・オブジェクトの書式（例、テキスト）です。このフィールドは、読み込み専用です。

適用対象

ボイラープレート

枠付きの行伸長

説明

「枠付きの伸長」プロパティは、ボイラープレート線が関係付けられた枠または繰り返し枠の名前です。このプロパティが設定されている場合、その線は両方の終点で関係付けられた枠にアンカーされます。したがって、レポートのフォーマット時に関係付けられた枠が拡張されるにつれて、この線も同じだけ伸ばされます。

縦線の場合、上の終点は枠の上部にアンカーされ、下の終点は枠の下部にアンカーされます。横線の場合、左側の終点は枠の左側にアンカーされ、右側の終点は枠の右側にアンカーされます。

値

レポート内に現在含まれている枠および繰返し枠のリストから、枠または繰返し枠を選択します。値がNULLの場合は、プロパティが設定されていません。また、レポートのフォーマット時にボーラプレート線のサイズは変化しません。

適用対象

ボーラプレート線

必須/オプション

オプション

デフォルト

NULL

使用上の注意

- 「枠付きの行伸長」は、レイアウト位置にかかわらず、すべてのボーラプレート線に設定できます。
- このプロパティを使って線を繰返し枠に関係付ける場合、線は繰返し枠内に完全に収める必要があります。

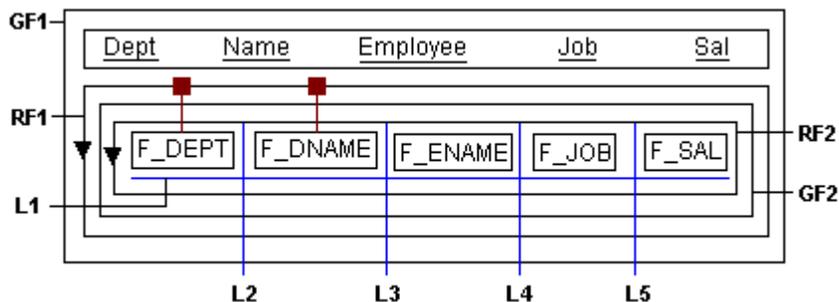
「枠付きの行伸長」の例（格子表示）

格子表示を使ってグループ左レポートの行と列を分割するとします。

<u>Dept</u>	<u>Name</u>	<u>Employee</u>	<u>Job</u>	<u>Sal</u>
10	Sales	Clerk	Manager	2450
		King	President	5000
		Miller	Clerk	1200
20	Accounting	Smith	Clerk	800
		Adams	Clerk	1100
		Ford	Analyst	3000
		Scott	Analyst	3000

アンカー、ボイラープレート線、「枠付きの行伸長」を使用して、次のようにしてレイアウトを編集します。

- 1番内側のグループ枠（GF2）とディテール繰返し枠（RF2）を拡張し、すべてのフィールドがディテール繰返し枠で囲まれるように、2つのグループ・フィールドを囲みます。
- 各グループ・フィールドをマスター繰返し枠（RF1）にアンカーします。
- ディテール繰返し枠（RF2）内のフィールドの下にボイラープレート線L1を作成します。
- ボイラープレート線L2～L5を作成し、「枠付きの行伸長」を使用して、外側のグループ枠（GF1）と関係付けます。



レポートがフォーマットされると、明示的なアンカーによってグループ・フィールドがマスター繰返し枠と同じ頻度で表示されます。線L1は、ディテール繰返し枠内の各レコードに対して繰り返す、線L2～L5の線は、最も外側にあるグループ枠の全長が入るように伸ばされます。

「枠付きの行伸長」の制限

- 「枠付きの行伸長」では、垂直方向および水平方向の線のみがサポートされます。対角線にこのプロパティを設定すると、レポートのフォーマット時に、縦線または横線になります。
- このプロパティが設定されていると、「暗黙のアンカーリング」アルゴリズムが適用されません。

ボタン・プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

コメント

フォーマット・トリガー

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アイコン名

アンカー・オブジェクトと連動

ラベルの型

マルチメディア列

マルチメディア列の型

マルチメディア・ファイル

マルチメディア・ファイル・タイプ

名前

後で改ページ

前で改ページ

ページ保護

PL/SQLトリガー

印刷オブジェクト・オン

後のプリンタ・コード

前のプリンタ・コード

テキスト

型

垂直拡張度

ボタンの制限

- ボタンは、印刷されません。また、ボタンが含まれるレポートをキャラクタ・モードで実行した場合も、ボタンは表示されません。
- ボタンは、使用しているプラットフォームの本来のボタンの特性を持ち、「ライブ・プレビューア」に表示されます。そのため、ボタンの外観に影響する変更（たとえば、境界線を破線に変更）を「レイアウト・モデル」で行っても、出力結果に現れないことがあります。「レイアウト・モデル」では、境界線は破線に変更されますが、「ライブ・プレビューア」では変更されません。「レイアウト・モデル」では、境界線は破線に変更されますが、「ライブ・プレビューア」では変更されません。
- ユーザー定義のボタンに様々なカラーとパターンを割り当てられるかどうかは、プラットフォームによって異なります。
- ボタンを回転することはできません。
- ボタンは、常に「ライブ・プレビューア」のオブジェクトの最上位のレイヤーに表示されます。ただし、ボタンが「レイアウト・モデル」の上に表示されないこともあります。たとえばボタンが「レイアウト・モデル」で繰り返し枠に入っている場合、ボタンは「ライブ・プレビューア」の繰り返し枠内のオブジェクトの上に表示されますが、繰り返し枠ごとには表示されません。

ラベルの型

説明

「ラベルの型」プロパティは、ボタン上に表示する識別子のタイプを示します。

値

テキスト	ボタン・ラベルをテキスト文字列として使用する場合に指定します。ラベルにテキスト文字列を入力するには、「テキスト」プロパティを使用してください。
アイコン	ボタン・ラベルをアイコンとして使用する場合に指定します。ラベルにアイコンを指定するには、「アイコン名」プロパティを使って、ラベルにアイコンが含まれたファイル名を指定します（ファイルは、UI_ICON環境変数で指定されたディレクトリにあります）。

適用対象

ボタン

必須/オプション

必須

デフォルト

テキスト

「ラベルの型」の制限

- ラベルとして使われるアイコンは、ご使用のプラットフォームの標準のアイコン形式である必要があります。拡張子は、.icoです。アイコンは、移植できません。

テキスト

説明

「テキスト」プロパティは、「ラベルの型」で「テキスト」を選択した場合にボタンに表示されるテキスト・ラベルを示します。指定されたテキスト文字列は、ボタンの水平方向と垂直方向の両方で中央揃えになります。

値

2Kまでの長さのテキスト文字列。しかし、実際の長さは、ボタンのサイズで決まります。

適用対象

ボタン

必須/オプション

オプション

デフォルト

ブランク

使用上の注意

- 「書式」メニューの選択肢を使用して、ラベル・テキストのフォントおよびサイズ、フォントの太さ、スタイルを変更できます。しかし、このメニューの選択肢の一部（間隔および整列など）は、ボタンのテキストには効果がありません。

「テキスト」の制限

- テキストがボタン上に収まらない場合、収まらないテキストが2行目に折り返されることはなく、テキストの左端と右端が切れてしまいます。

アイコン名

説明

「アイコン名」プロパティは、「ラベルの型」で「アイコン」を選択した場合に、ボタンに表示されるアイコンを含むファイル名を示します。

値

ご使用のプラットフォームの標準のアイコン形式で、.icoという拡張子の有効なアイコン・ファイル。ファイルは、UI_ICON環境変数で指定されたディレクトリにあります。

適用対象

ボタン

必須/オプション

オプション

デフォルト

ブランク

型

説明

「型」プロパティは、ボタンを押したときに実行されるアクションです。

値

マルチメディア・ファイル	アクションがオペレーティング・システム内のファイルに含まれることを意味します。「マルチメディア・ファイル」プロパティで指定します。
マルチメディア列	アクションがデータベース列に含まれることを意味します。「マルチメディア列」プロパティで指定します。
PL/SQL	アクションが、ボタンと関連付けられたPL/SQLプログラム単位によって制御されることを意味します。「PL/SQLトリガー」プロパティで指定します。

適用対象

ボタン

必須/オプション

必須

デフォルト

マルチメディア・ファイル

「型」の制限

- 「PL/SQL」を指定し、PL/SQLでSRW.RUN_REPORTが使われている場合、ボタンを選択したときのランタイム・パラメータ・フォームはデフォルトでは表示されません。ランタイム・パラメータ・フォームを表示したい場合は、SRW.RUN_REPORTの呼出しでPARAFORM=YESを指定する必要があります。
- 「PL/SQL」を指定し、ボタンのアクション・トリガーでSRW.PROGRAM_ABORTを呼び出す場合、ボタンによって呼び出された「プレビューア」ウィンドウはすべてクローズされます。

マルチメディア・ファイル

説明

「マルチメディア・ファイル」プロパティは、マルチメディア（サウンド、ビデオまたはイメージ）・ファイルの名前を示します。

値

有効なサウンド・ファイル、ビデオ・ファイルまたはイメージ・ファイル

適用対象

ボタン

必須/オプション

「型」が「マルチメディア・ファイル」の場合、必須。

デフォルト

ブランク

使用上の注意

- レポートの移植を確保するには、REPORTS60_PATHというReport Builder固有の環境変数を使用します。この変数を使用して、レポート内で使用される外部ファイル（外部問合せ、ボイラープレート、PL/SQLなど）をReport Builderに検索させるためのデフォルト・ディレクトリを指定できます。これによって、レポートにディレクトリ・パスをハードコードする必要がなくなります。

「マルチメディア・ファイル」の制限

- 「マルチメディア・ファイル」を選択した場合、表示されているテキスト・フィールドにファイル名を入力してください。ファイルのパスを入力しない場合は、Report Builderのファイル・パス検索順序に従ってファイルを検索します。

マルチメディア・ファイル・タイプ

説明

「マルチメディア・ファイル・タイプ」プロパティは、「マルチメディア・ファイル」プロパティによって指定されたファイルの形式を示します。

値

イメージまたはビデオ、サウンド

適用対象

ボタン

必須/オプション

必須

デフォルト

イメージ

マルチメディア列

説明

「マルチメディア列」プロパティは、マルチメディア（サウンドまたはビデオ、イメージ）・オブジェクトが含まれるレポート内の、データベースの列の名前を示します。

値

マルチメディア・オブジェクトが含まれるレポート内の、有効なデータベースの列

適用対象

ボタン

必須/オプション

「型」が「マルチメディア列」の場合、必須。

デフォルト

NULL

「マルチメディア列」の制限

- ボタンのアクションに「マルチメディア列」を指定した後、レポートからその列を削除した（つまり、イメージ列を選択して、ボタンに対してその列を「マルチメディア列」として指定した後、その列を削除またはマルチメディア以外の形式に変更した）場合、そのボタンから表示するデータにはアクセスできなくなり、「マルチメディア列」は未定義になります。

「マルチメディア列」が未定義になるときにこのボタンのプロパティ・パレットがオープンされている場合、別の列を選択してからそのプロパティ・パレットを確定する必要があります。また、「マルチメディア列」が未定義になるときにこのプロパティ・パレットがクローズされている場合、レポートを実行してボタンをアクティブにしようとするまで、警告は出ません。この処理は、エラーになります。

マルチメディア列の型

説明

「マルチメディア列の型」プロパティは、「マルチメディア列」プロパティによって指定されたデータベースの列に格納される、マルチメディア・オブジェクトの形式を示します。

値

イメージまたはビデオ、サウンド

適用対象

ボタン

必須/オプション

必須

デフォルト

「マルチメディア列」が指定された場合、適切なマルチメディア列タイプが自動的に選択されます。それ以外の場合、デフォルト値はイメージです。

PL/SQLトリガー

説明

「PL/SQLトリガー」プロパティは、「ライブ・プレビューア」でボタンが選択されたときに実行されるPL/SQLプロシージャを示します。トリガーは、別のレポートを動的にコール（ドリルダウン）したり、他のPL/SQLを実行するのに使用できます。

適用対象

ボタン

必須/オプション

ボタンの「型」が「PL/SQL」の場合、必須。

デフォルト

ブランク

チャート・プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

チャート・ファイル名

チャート・ハイパーリンク

コメント

フォーマット・トリガー

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アンカー・オブジェクトと連動

名前

後で改ページ

前で改ページ

ページ保護

パラメータおよび列

印刷オブジェクト・オン

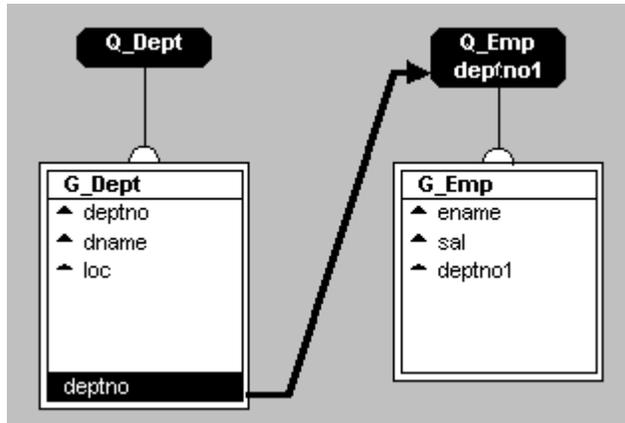
後のプリンタ・コード

前のプリンタ・コード

垂直拡張度

グループ・レポート内のチャートの例

次のデータ・モデルを持つグループ・レポートがあるとしています。



Graphics Builderで、各従業員の給与を示す棒グラフを表示するGraphics Builder図表 (empsal.ogd) を作成しており、レポートに組み込みたいとします。「レイアウト・モデル」内のチャート・ツールを使って、まず、マスター繰返し枠 (R_Dept) 用の繰返し枠の中にチャート・オブジェクトを描きます。これによって、各部門別の棒グラフが表示されます。

その後、そのチャート・オブジェクトのプロパティ・パレットに移動し、「表示名」プロパティに empsal.ogd を設定します。「表示名」へのパス指定は任意ですが、このケースでは、REPORTS60_PATH環境変数に指定したパスをReport Builderで使用するので、パスは指定しません。

プロパティ・パレットのプロパティの「パラメータおよび列」で次のように設定します。

レポート・グループ	G_Emp
チャート問合せ	Q_OG_Emp
レポート列/チャート列	ENAME/OG_NAME SAL/OG_SALARY

「レポート・グループ」は、図表によって要約されているグループ (G_Emp) です。

「チャート問合せ」は、図表内の問合せの名前です。Q_OG_Empには、次のSELECT文が含まれています。

```
SELECT ENAME OG_NAME, SAL OG_SALARY FROM EMP
```

「チャート問合せ」にQ_OG_Empを指定したので、この問合せは、データベースではなくReport BuilderからENAMEとSALの値を取り出します。

「レポート列」と「チャート列」は、レポートの列を図表の列にマップします。「レポート列」の下のリストから、ENAMEとSALを選択します。「チャート列」内のENAMEとSALに対応する図表の列の名前を入力してください。(「レポート列」が指定されている場合、「チャート列」を空白にすることはできません。)

チャートの制限

- データを渡すGraphics Builderの図表またはチャートは、そのソース・グループがマトリックスの次元でない場合は「レイアウト・モデル」の任意の場所に配置できます。Graphics Builderの図表またはチャートのソース・グループが、次元グループ（つまり、クロス積グループ内のグループ）の場合、そのソース・グループの上位項目（上位項目がある場合）の繰返し枠の中に表示してください。
- パラメータを渡すGraphics Builderの図表またはチャートは、ソース・グループが、渡される列またはパラメータを含むグループ（または、グループの下位項目）となる繰返し枠の中に配置してください。渡される列が、クロス積グループによって所有されるサマリーの場合、Graphics Builderの図表またはチャートは、ソースがサマリー列の積順序内のグループである、繰返し枠の集合の中に配置してください。

チャート・ファイル名

説明

「チャート・ファイル名」プロパティは、Graphics Builderの図表またはチャートが含まれるファイルの名前を示します。

値

Graphics Builderの図表またはチャート（拡張子である.ogdはオプション）の有効なファイル名を入力します。ファイル名の長さは、512バイト以下にしてください。

適用対象

チャート・オブジェクト

必須/オプション

必須

デフォルト

ブランク

使用上の注意

- オプションでファイル名の前にパスを付けることができます。ファイル名の前にパスが指定されていない場合は、Report Builderのファイル・パス検索順序によって、ファイルが検索されます。
- レポートの移植を確保するには、REPORTS60_PATHというReport Builder固有の環境変数を使

用します。この変数を使用して、レポート内で使用される外部ファイル（外部問合せ、ポイラプレート、PL/SQLなど）をReport Builderに検索させるためのデフォルト・ディレクトリを指定できます。これによって、レポートにディレクトリ・パスをハードコードする必要がなくなります。

「チャート・ファイル名」の制限

- ファイルに格納された図表またはチャートのみを参照できます。データベースに格納された図表またはチャートは、「チャート・ファイル名」では参照できません。

チャート・ハイパーリンク

説明

「チャート・ハイパーリンク」プロパティは、チャート・セクションごとに異なるように指定できるハイパーリンクです。注意

値

レポート出力内の任意のオブジェクト、他のレポートまたは任意の有効なハイパーリンク宛先

適用対象

チャート・オブジェクト

必須/オプション

オプション

デフォルト

ブランク

使用上の注意

- 他のオブジェクトをその宛先として指定する、チャートの「チャート・ハイパーリンク」プロパティを設定する場合は、「ハイパーリンクの宛先」プロパティを指定する必要があります。
- 「チャート・ハイパーリンク」プロパティによって定義されたリンクに追加のHTML属性を適用する場合、「追加のハイパーリンク属性」プロパティは使用できません。
- レポート出力から2つ目のレポートを実行することにより、他のレポートを「チャート・ハイパーリンク」プロパティとして設定できます。
- フォーマット・トリガーはチャートにつき1度しか実行されないため、PL/SQLを使用してこの

プロパティを設定できません。その代わりに、「チャート・ハイパーリンク」プロパティとして<column_name>を指定できます。

チャート・パラメータと列プロパティ

チャート列
 チャート・パラメータ
 チャート問合せ
 レポート列 (チャート列)
 レポート列 (チャート・パラメータ)
 レポート・グループ

チャート・パラメータ

説明

「チャート・パラメータ」プロパティは、「レポート列」で指定した列またはパラメータに対応する、Graphics Builderの図表またはチャート内で使用されるパラメータを示します。

注意: 1つのレポート列またはパラメータは、複数のチャート・パラメータに対応させることができますが、各チャート・パラメータは、1つのレポート列またはパラメータのみに対応します。

値

Graphics Builderの図表またはチャート内の有効なパラメータの名前。名前の長さは、1Kバイト以下にしてください。Graphics Builderの図表またはチャート内の有効なパラメータ名のリストを入力することもできます。カンマ(,)を入力して、パラメータ名を区切ってください(たとえば、parm1,parm2,parm3)。

適用対象

チャート・オブジェクト

必須/オプション

オプション

デフォルト

Report Builderのパラメータ名

レポート列 (チャート・パラメータ対応)

説明

「レポート列 (チャート・パラメータ対応)」プロパティは、Graphics Builderの図表またはチャートのパラメータに値を指定するために使用される、レポート内の列とパラメータのリストを示します。列またはパラメータを選択することは、図表またはチャート内に対応するパラメータがあるということであり、図表またはチャート・パラメータは選択済みレポート列またはパラメータから値を取得するという事です。図表またはチャートのパラメータ名がレポート内の列またはパラメータと異なる場合、対応する「チャート・パラメータ」フィールドにその名前を指定してください。

値

レポートの中から1つ以上の有効な列またはパラメータを選択します。

適用対象

チャート・オブジェクト

必須/オプション

オプション

デフォルト

選択されていません。

「レポート列 (チャート・パラメータ対応)」の制限

- ページレベルのサマリー列 (つまり、「リセット位置」または「計算位置」が「Page」に設定されている場合のサマリー) は、Graphics Builderの図表またはチャートに渡すことはできません。同様に、ページレベルのサマリーに依存するサマリー列または式列も、図表またはチャートに渡すことはできません。

チャート問合せ

説明

「チャート問合せ」プロパティは、Report Builderからデータを取得するGraphicBuilderの図表またはチャートの問合せの名前です。問合せ名を指定すると、その問合せはデータベースからではなくReport Builderからデータを取り出します。これにより、レポートでデータに対して行ったすべてのフィルタあるいは集計を図表またはチャートに反映させることができます。「チャート問

「合わせ」は、図表またはチャートで使用されたデータがレポートで使用されたデータと同じであることを確認します。

注意: ここで問合せ名を指定する場合は、どのレポート列あるいはパラメータが図表またはチャートの問合せに使用される値を持つかも指定してください。

値

チャート・ファイル名で指定された、Graphics Builderの図表またはチャートの有効な問合せ名を入力します。問合せ名の長さは、512バイト以下にしてください。

適用対象

チャート・オブジェクト

必須/オプション

オプション

レポート・グループ

説明

「レポート・グループ」プロパティは、Graphics Builderの図表またはチャートによってサマリー関数値を計算した、レポート内のグループを示します。

値

値リストからグループ名を選択します。

適用対象

チャート・オブジェクト

必須/オプション

「チャート問合せ」に値が入力されている場合、必須。

「レポート・グループ」の制限

- 「レポート・グループ」には、クロス積グループを指定することはできません。
- 「レポート列（チャート列対応）」を介してGraphics Builderの図表またはチャートに渡す列は、いずれも、レポート・グループ内に存在するか、その上位項目にしてください。
- 「チャート問合せ」を選択せず「レポート・グループ」を選択した場合、プロパティ・シー

トを確定したときに「レポート・グループ」は選択解除されます。

チャート列

説明

「チャート列」プロパティは、「レポート列」で指定した列またはパラメータに対応する、Graphics Builderの図表またはチャート内で使用される列または列のリストを示します。

注意: 1つのレポート列またはパラメータは、複数のチャート・パラメータに対応させることができますが、各チャート列は、1つのレポート列またはパラメータのみに対応します。

値

Graphics Builderの図表またはチャート内の有効な列の名前。名前の長さは、1K以下にしてください。Graphics Builderの図表またはチャート内の有効な列名のリストを入力することもできます。カンマ(,)を入力して、列名を区切ってください(たとえば、col1,col2,col3)。

適用対象

チャート・オブジェクト

必須/オプション

オプション

デフォルト

Report Builderの列名

レポート列 (チャート列対応)

説明

「レポート列 (チャート列対応)」プロパティは、Graphics Builderの図表またはチャートのパラメータに値を指定するために使用される、レポート内の列とパラメータのリストを示します。列またはパラメータを選択するということは、Graphics Builderの図表またはチャート内に対応するパラメータがあるということであり、図表またはチャート列は選択済みレポート列またはパラメータから値を取得するということです。図表またはチャートの列名がレポート内の列またはパラメータと異なる場合、対応する「チャート列」フィールドにその名前を指定してください。

値

レポートの中から1つ以上の有効な列またはパラメータを選択します。

適用対象

チャート・オブジェクト

必須/オプション

「チャート問合わせ」に値が入力されている場合は必須

「レポート列（チャート列対応）」の制限

- 「レポート列（チャート列対応）」を介してGraphics Builderの図表またはチャートに渡す列は、いずれも、「レポート・グループ」内に存在するか、その上位項目にしてください。
- ページレベルのサマリー列（つまり、「リセット位置」または「計算位置」が「Page」に設定されている場合のサマリー）は、Graphics Builderの図表またはチャートに渡すことはできません。同様に、ページレベルのサマリーに依存するサマリー列または式列も、Graphics Builderの図表またはチャートに渡すことはできません。

共通の列プロパティ

ブレイク順序
コメント
列のデータ型
データ型
ファイル形式
名前
ファイルからの読み込み
ブレイク順序の設定
NULL時の値
幅

ブレイク順序

説明

「ブレイク順序」プロパティは、列値の表示順序を示します。このプロパティは、ユーザーが作成したグループ（つまり、ブレイク・グループ）の固有の値を識別する列にのみ適用されます。デフォルトのグループ内の列値の順序は、SQL問合せのORDER BY句およびExpressへの問合せのソート列によって決まります。しかし、ユーザーが作成したグループ内の列値の場合、ブレイク列の値の順序を指定するため、「ブレイク順序」を使用してください。

値

なし
昇順
降順

適用対象

列

必須/オプション

「ブレイク順序の設定」が「はい」の場合、必須。

デフォルト

データベースおよびソート列は「昇順」。ブレイク・グループを使用してOracle Express問合せ内のディメンションを表示するデータベース列は「なし」。

ブレイク順序の例（降順）

次の問合せによってレポートを作成するとします。

```
SELECT DEPTNO, JOB, ENAME, SAL FROM EMP
ORDER BY SAL
```

その後、2つのグループG_DEPTおよびG_JOBを作成します。G_DEPTにはDEPTNO列が含まれ、G_JOBにはJOB列が含まれます。DEPTNO列の「ブレイク順序」に「降順」を指定し、JOB列には「昇順」を指定した場合、出力は次のようになります（表スタイルを使用することを想定）。

Deptno	Job	Ename	Sal
30	CLERK	JAMES	950.00
	MANAGER	BLAKE	2850.00
	SALESMAN	WARD	1250.00
	MARTIN		1250.00
	TURNER		1500.00
	ALLEN		1600.00
20	ANALYST	SCOTT	3000.00
		FORD	3000.00
	CLERK	SMITH	800.00
		ADAMS	1100.00
	MANAGER	JONES	2975.00
10	CLERK	MILLER	1300.00
	MANAGER	CLARK	2450.00
	PRESIDENT	KING	5000.00

ブレイク順序の例 (ORDER BY)

DEPTNOおよびENAMEのあるグループ左レポートを作成するとします。問合せでは、ORDER BY DEPTNOを使用します。また、DEPTNO列の「ブレイク順序」には「降順」を指定してあります。この場合、出力は次のようになります。

```
Deptno Ename
```

```
-----
```

```
30 JAMES
```

```
WARD
```

```
MARTIN
```

```
TURNER
```

```
ALLEN
```

```
BLAKE
```

```
20 SMITH
```

```
ADAMS
```

```
JONES
```

```
SCOTT
```

```
FORD
```

```
10 MILLER
```

```
CLARK
```

```
KING
```

「ブレイク順序」プロパティがORDER BY句より優先することに注意してください(つまり、部門番号が降順になっています)。

「ブレイク順序」の制限

- 「ブレイク順序」は、特定の問合せの最下位のグループに属する列には影響を与えません。「ブレイク順序」は、問合せの最下位の子グループより上のグループの列のみに影響を与えます。
- 問合せの最下位の子グループより上のグループは、「ブレイク順序」が設定された列を少なくとも1つ持っている必要があります。
- 「ブレイク順序」は、「データ型」が「LONG」または「LONG RAW」の列には指定でき

ません。

- 「ブレイク順序」はブレイク対象の列の順序にのみ影響を与えます。ブレイク・グループの内部の列の順序には影響を与えません。たとえば、DEPTNOでブレイクし、各部門についてENAMEをリストするとします。「ブレイク順序」が「昇順」の場合、部門番号は昇順で印刷されますが、各部門の従業員の名前は問合せによって指定されている順序で印刷されます。(SQL問合せでその順序を変更するには、SELECT文でORDER BY句を使用する必要があります。また、Expressへの問合せでその順序を変更するには、SelectorのSortツールを使用する必要があります。)
- サマリー列は、ブレイク列にすることはできません。また、「ブレイク順序」を設定することはできません。
- サマリー列に依存する式列は、ブレイク列にすることはできません。また、「ブレイク順序」を設定することはできません。

列タイプ

説明

「列タイプ」プロパティは、列が属するカテゴリです。

値

データベース - オブジェクト

データベース - 参照

データベース - スカラー

データベース - 不明 (列がサポートされていないタイプであることを示す)

式

ブレースホルダ

ソート(Oracle Express対応のみ)

サマリー

コメント

説明

「コメント」プロパティは、列についてのコメントを書き込むテキスト・フィールドです。

値

64K以下のテキストを入力します。

適用対象

列

必須/オプション

オプション

デフォルト

ブランク

データベース列名

説明

「データベース列名」プロパティは、列を識別するために使用される列の正式名称です。たとえば、CITYという名前の列を含むCOUNTRYという名前の列オブジェクトが含まれるADDRESSという名前の列オブジェクトがあると想定します。この場合、CITYのデータベース列名は、ADDRESS.COUNTRY.CITYとなります。

適用対象

列

必須/オプション

必須

データ型

説明

「データ型」プロパティは、列の内容の型を示します。

値

Character

Date

LONG

LONG RAW

Nested Table of <データ型>

Number

RAW

Ref of <データ型>

Rowid

VARCHAR

VARCHAR2

Trusted Oracleのデータ型（MLSLABELとROWLABEL）はReport Builderでサポートしています。Trusted Oracleのインストールされているオペレーティング・システム上でReport Builderを実行すると、これらの列はTrusted Oracleのデータ型として扱われます。つまり、これらの列は適切にソートされることを意味しています。Trusted Oracleのインストールされていない（たとえば、Windows95/NTなどがインストールされている）オペレーティング・システム上でReport Builderを実行すると、これらの列は文字データ型として扱われます。

適用対象

列

必須/オプション

必須

デフォルト

データベースの列の場合、デフォルトの「データ型」はデータベースから導出されます。「ファンクション」が「カウント」に設定されたサマリー列の場合、デフォルトの「データ型」は、「NUMBER」です。その他のサマリー列の場合、デフォルトは「ソース」から導出されます。式列、プレースホルダ列およびソート列の場合、デフォルトの「データ型」は、「NUMBER」です。

データ型の制限

- 式列またはプレースホルダ列の「データ型」のみは編集できます。式列およびプレースホルダ列の場合は、「データ型」を「CHARACTER」および「DATE」、「NUMBER」に設定できます。
- 「データ型」が「LONG」または「LONG RAW」、「RAW」の場合、「ファイル形式」フィー

ルドでその他の値を指定しなければ、列はテキストとして処理されます。

- 列に別名があり、問合せを変更すると列のデータ型が変更される場合、このデータ型の変更は、問合せを確定したときに自動的に反映されます。

ファイル形式

説明

「ファイル形式」プロパティは、データベースから取り出されたか、列に保存されたファイル名で識別されるオブジェクトの形式です。

値

テキスト	ASCII形式のテキストです。
イメージ	ビットマップ・イメージです。
CGM	CGM形式での線の描画です。
Oracle Drawing Format	Oracle形式での線の描画です。
サウンド	longraw形式のサウンド・オブジェクトです。
ビデオ	longraw形式のビデオ・オブジェクトです。
OLE2	レポートに埋め込まれるリンク・オブジェクトです。
イメージURL	イメージへのURLリンクです。

適用対象

列

「ファイル形式」の制限

- 「サウンド」および「ビデオ」のソースにアクセスできるのは、ユーザーが作成したボタンからのみです。

ファイルからの読み込み

説明

「ファイルからの読み込み」プロパティは、列にファイル名またはイメージのURLが含まれていることを示します。「ファイルからの読み込み」を「はい」に設定する場合は、「ファイル形式」プロパティにファイル名を指定してください。

「ファイルからの読み込み」は、列によって指し示されるイメージまたは図形、テキスト・ファイルがある場合に役立ちます。「ファイルからの読み込み」を「はい」に設定した場合、ファイルの内容がインポートされ、列の値として使用されます。「ファイルからの読み込み」を「いいえ」に

設定した場合は、データベースから値を取得します。その結果、列にファイル名が含まれ、「ファイルからの読み込み」が「いいえ」に設定されている場合、ファイルの内容ではなく、ファイル名が印刷されます。

「ファイルからの読み込み」は、データベース列またはサマリー列、式列、ブレースホルダ列に対して使用することができます（列値がファイル名の場合）。ファイルを、ポイラプレートとしてレポートに組み込むこともできます。

適用対象

列

必須/オプション

オプション

使用上の注意

- 列内のファイル名の前にパスを付けることができます。ファイル名の前にパスが指定されていない場合は、Report Builderのファイル・パス検索順序によって、ファイルが検索されます。
- レポートの移植を確保するには、REPORTS30_PATHというReport Builder固有の環境変数を使用します。この変数を使用して、レポート内で使用される外部ファイル（外部問合せ、ポイラプレート、PL/SQLなど）をReport Builderに検索させるためのデフォルト・ディレクトリを指定できます。これによって、レポートにディレクトリ・パスをハードコードする必要がなくなります。

ファイルからの読み込み例（イメージ）

各従業員用に、従業員の写真が含まれる.bmpファイルがあるとします。従業員表には、その.bmpファイルの名前と従業員の写真を含むPICTURE列があります。

従業員の写真を表示するレポートを生成するには、問合せでPICTURE列を選択します。PICTURE列の「列」プロパティ・シートでは、「ファイルからの読み込み」を「はい」に設定し、「ファイル形式」を「イメージ」に設定します。すると、Report Builderによって、PICTURE列によって指し示されたファイルは、レイアウト内のPICTURE列を参照した場所に表示されます。

注意:「ファイルからの読み込み」を「いいえ」に設定した場合は、写真ではなく、ファイル名が印刷されます。

ファイルからの読み込み例（式列）

ファイルからテキストを各従業員の情報の横に取り込んだ従業員リストを作成するとします。従業員の給与が2000ドルを超える場合、その従業員の横には、*salary1.txt*というファイルからテキストを表示するとします。従業員の給与が2000ドル未満の場合、*salary2.txt*というファイルからその従業員の横にテキストを表示するとします。

「データ型」を「CHARACTER」に、「ファイルからの読み込み」を「はい」（「ファイル形式」が「テキスト」の場合）に設定した式列を作成します。式列で、次のPL/SQLを入力します。

```
if :sal > 2000 then
    return('/home/jsmith/txt/salary1.txt');
else
    return('/home/jsmith/txt/salary2.txt');
end if;
```

「ファイルからの読み込み」の制限

- 各ファイルの最大サイズは、4GBです。（一部のオペレーティング・システムでは、最大64Kです。）
- 列の「データ型」が文字互換型（つまり、CHARACTERまたはVARCHAR、VARCHAR2）の場合のみ、「ファイルからの読み込み」を「はい」に設定できます。
- サマリー列または式列について「ファイルからの読み込み」を「はい」に設定した場合、サマリーまたは式は、ファイルが読み込まれる前に計算されます。同様に、プレースホルダ列について「ファイルからの読み込み」が「はい」に設定されている場合も、列値は最初に計算されます。

NULL時の値

説明

「NULL時の値」プロパティは、列のNULL値に代入される値です。たとえば、このフィールドにキャラクタ・タイプの列としてXを入力した場合、列にフェッチされたNULL値には、Xが表示されます。ブランクのままにした場合、NULL値には何も代入されません。

値

列のデータ型に合う有効な値を入力します。「NULL時の値」の最大長は1Kです。

適用対象

列

必須/オプション

オプション

デフォルト

ブランク

使用上の注意

- ブレーク列について「NULL時の値」に文字列を入力すると、Report Builderによって、ブレーク列のグループ内のすべての行が事前フェッチされます。これによって、レポート実行時の性能が落ちることがあります。事前フェッチを避けるには、「NULL時の値」に文字列を入力しないで、問合せでNVLファンクションを使用します。

「NULL時の値」の制限

- 文字データ型（たとえば、VARCHAR）を持つ列の場合、「NULL時の値」に入力した文字列の幅が列の幅を超えると、文字列の一部は切り捨てられます。
- 問合せの中の字句参照を介して列が参照されている場合は、問合せの妥当性検査のためにその列の「NULL時の値」の値が使用されます。
- 「NULL時の値」に入力する値のデータ型は、列のデータ型と同じにしてください。たとえば、列の「データ型」が「NUMBER」の場合、「NULL時の値」に「XX」を指定することはできません。この場合は、数値を指定してください。
- 「NULL時の値」に日付値を入力する場合、Report Builderの日付用の内部マスク（YY-MM-DD）に対して妥当性検査が行われます。（数字や文字の値については、妥当性検査は行われません。）

幅

説明

「幅」プロパティは、列値の最大文字数です。この「幅」とは、Report Builderによってその列に対して内部的に使用される幅のことです。サマリーおよび式の計算などに影響があります。しかし、列のデータ表示に使用されるサイズは、「レイアウト・モデル」内の関係付けられたフィールドのサイズおよび「垂直拡張度」プロパティ、「水平拡張度」プロパティによって決まります。

値

1 ~ 64Kの数を入力します。

2つの数字は「a,b」という形式で入力します。「a」は全体の桁数を示し、「b」は小数点以下の桁数を示しています。（この形式は、数字の桁数を決めるためのみに使用されます。）

適用対象

列

必須/オプション

必須

デフォルト

データベースの列の場合、デフォルトの「幅」はデータベースから導出されます。ソート列の場合、デフォルトの「幅」は、「38」です。サマリー列および式列の場合は、デフォルトの「幅」は、「10.0」です。サマリー列の場合は、Report Builderでは列の「ソース」および「ファンクション」として入力した値に基づいて「幅」を更新します。

使用上の注意

- 「列タイプ」が「データベース - スカラー」で、「データ型」が「CHARACTER」または「DATE」、「NUMBER」、「RAW」の場合、「幅」を編集することはできません。しかし、レポート出力の列値の幅を制御できないということではありません。「デフォルト・レイアウト」ダイアログ・ボックスまたは「フィールド」プロパティ・シートでフィールド幅を変更することによって、出力値の幅を変更できます。
- 「列タイプ」が「データベース - スカラー」または「データベース - プレースホルダ」で、「データ型」が「LONG」または「LONG RAW」の場合、「幅」に値を入力できます。「LONG」または「LONG RAW」の列に値を指定しない場合、Report Builderによって列の最大幅が割り当てられます。式列の場合も「幅」を入力できます。
- 「列タイプ」が「データベース - 参照」の場合、「幅」を編集することはできません。
- 「列タイプ」が「ソート」の場合は、「幅」を編集できます。

幅の制限

注意: データベース列で「データ型」が「LONG」または「LONG RAW」の列の値が、列に指定された「幅」より大きい場合、値は切り捨てられます。

- プレースホルダまたは式列の値が、列に指定された「幅」より大きい場合、例外状況が発生します。
- 列に別名があり、問合せを変更すると列の幅が変更される場合、この幅の変更は、問合せを確定したときに自動的に反映されます。

ブレイク順序の設定

説明

「ブレイク順序の設定」プロパティは、「ブレイク順序」プロパティを使用して、列値の表示順序を設定するかどうかを示します。

適用対象

列

必須/オプション

オプション

デフォルト

はい

共通のレイアウト・オブジェクト・プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

コメント

条件付き書式

表示名

フォーマット・トリガー

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アンカー・オブジェクトと連動

名前

後で改ページ

前で改ページ

ページ保護

印刷オブジェクト・オン

前のプリンタ・コード

後のプリンタ・コード

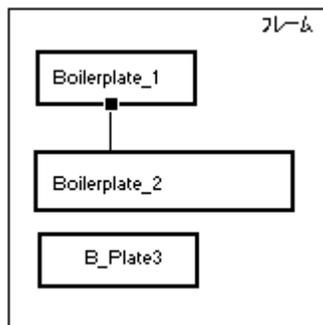
垂直拡張度

共通のレイアウト・オブジェクトの制限

- 同じオブジェクトでも、異なるオブジェクトでも、プロパティが相互に矛盾しないように設定してください。レイアウト・プロパティの機能上、別のプロパティと矛盾するプロパティ

の指定は可能です。同一のオブジェクトについて、次のようにレイアウト・プロパティを組み合わせるのは避けてください。同一のオブジェクトに次の組合せを指定しようとすると、エラーになります。

- 「基本印刷オン」を「インクローズ・オブジェクト」に、「印刷オブジェクト・オン」を「最終ページ」に設定
- 「基本印刷オン」を「インクローズ・オブジェクト」に、「印刷オブジェクト・オン」を「最後のページ以外の全ページ」に設定
- 「印刷オブジェクト・オン」を「全ページ」に、「前で改ページ」を「はい」に設定
- 「印刷オブジェクト・オン」を「最初のページ以外の全ページ」に、「前で改ページ」を「はい」に設定
- 「印刷オブジェクト・オン」を「最後のページ以外の全ページ」に、「前で改ページ」を「はい」に設定
- 「アンカー・オブジェクトと連動」および「前で改ページ」を「はい」に設定（アンカー・オブジェクトの後に配置されるオブジェクトの場合）。または、「アンカー・オブジェクトと連動」および「後で改ページ」を「はい」に設定（アンカー・オブジェクトの前に位置するオブジェクトの場合）。
- 同様に、異なるオブジェクト間でプロパティの組合せが矛盾していると、予期しない結果が発生したり、レポートが無限に出力されることがあります。たとえば、枠の中で互いにアンカーされた2つのボイラプレート・オブジェクトがあるとします。親オブジェクトの「印刷オブジェクト・オン」が「全ページ」に、「基本印刷オン」が「インクローズ・オブジェクト」に設定されています。子オブジェクトの「印刷オブジェクト・オン」が「第1ページ」に、「基本印刷オン」が「アンカー・オブジェクト」に、「前で改ページ」が「はい」に設定されています。下の図は、このレイアウトを示しています。レポートを実行しても、子オブジェクトは出力されません。「前で改ページ」が「はい」に設定されているので、子オブジェクトの「印刷オブジェクト・オン」（「第1ページ」）および「基本印刷オン」（「アンカー・オブジェクト」）が満たされることはありません。しかし、子オブジェクトの下にあるオブジェクト（B_Plate3）は出力されることに注意してください。



コメント

説明

「コメント」プロパティには、オブジェクトを説明するテキスト・フィールドが含まれています。64KB以下のテキストを入力します。

適用対象

オブジェクト

必須/オプション

オプション

デフォルト

ブランク

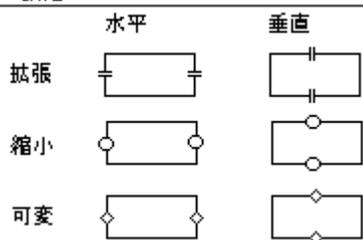
水平拡張度

説明

「水平拡張度」プロパティは、実行時にオブジェクトまたはデータをその中に収めるためにオブジェクトの水平サイズを変更する方法を示します。

- 枠および繰返し枠の場合、このプロパティは、枠または繰返し枠のサイズをその中のオブジェクトに合わせて変更する必要があるかどうかを定義します。
- テキストが含まれるオブジェクトの場合、このプロパティは、フィールドまたはボイラープレートがテキストのサイズに合わせて変更される必要があるかどうかを定義します。サイズ固定のテキストは、定義されたオブジェクト・サイズの範囲内でラップされますが、十分な空間がなければ切り捨てられる場合があります。数値データまたは日付データを定義されたサイズ内に収められない場合、アスタリスク(*)が表示されます。
- イメージおよび描画、チャート・オブジェクトの場合、Report Builderによって、比例したスケール変更が行われます。イメージおよび描画、チャート・オブジェクトの拡張度オプションによって、スケールが決まります。

サイズ設定



値

縮小	その中のフォーマットされたオブジェクトまたはデータの幅が狭い場合、オブジェクトの水平サイズは小さくなります。データの切捨てが行われる場合があります。例を参照してください。（このオプションは、「縮小のみ行われ、拡張されない」ことを意味しています。）
拡張	フォーマットされたオブジェクトまたはデータの幅が広い場合、オブジェクトの水平サイズは大きくなります。（このオプションは、「拡張のみ行われ、縮小されない」ことを意味しています。）
固定	オブジェクトの幅は、その中のオブジェクトまたはデータのサイズに関係なく、各論理ページで同じです。データの切捨てが行われる場合があります。例を参照してください。オブジェクトの幅は、レポート・エディタ内のオブジェクトの幅に定義されます。
可変	オブジェクトは水平方向に拡張または縮小され、その中のオブジェクトまたはデータは適切なサイズに調整されます。つまり、実行時のオブジェクトの幅は、レポート・エディタ内に示される幅には影響されないということです。

適用対象

レイアウト・オブジェクト

必須/オプション

必須

デフォルト

オブジェクトにより異なる。

使用上の注意

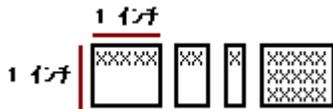
- 「垂直拡張度」および「水平拡張度」を「固定」に設定して、レポート内にチャート・オブジェクトを作成した場合、Report Builderはオブジェクトに定義したサイズに合わせて図表をスケールリングします。

水平拡張度の例（フィールド）

次の特性を持つフィールドがあるとします。

特性	値
幅	1インチ
水平拡張度	縮小
高さ	1インチ
垂直拡張度	固定

データ幅が1インチより短い場合は、フィールドのサイズはデータを収めるのに最低限必要なサイズに縮小されることに注意してください。また、データが1インチより長い場合、フィールドを垂直または水平方向に拡張できないため、データが切り捨てられることに注意してください。

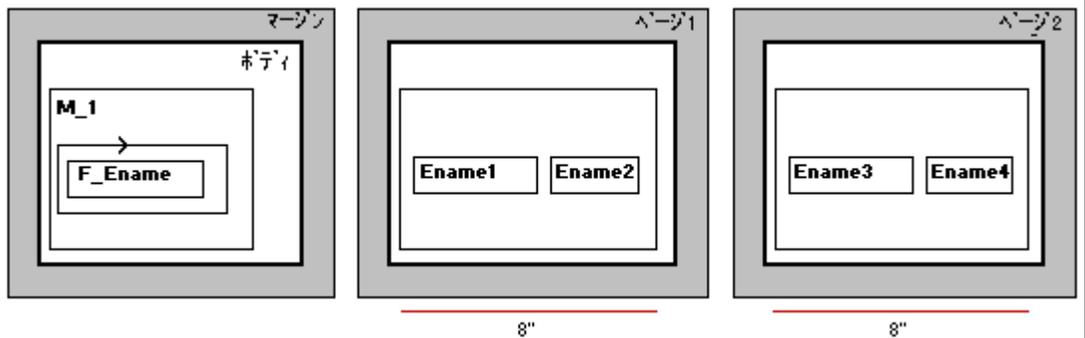


水平拡張度の例（枠）

次の特性を持つ、M_1という枠があるとします。

特性	値
水平拡張度	可変
高さ	7インチ
垂直拡張度	固定

また、レポートのプロパティ・シートで「幅」を8インチに定義したとします。M_1の中には、フィールドF_Enameを持つ繰返し枠があります。F_Enameの最初の値の幅が4インチ、次の値の幅は3インチ、その次の値の幅は4インチ、そして、最後の値の幅は4インチです。次の図は、枠がどのようにフォーマットされるかを示しています。



フィールドF_Enameのインスタンスのうち最初2つは論理ページの1ページ目に印刷されますが、1ページ目に十分な空間がないため、残りの2つのインスタンスは2ページ目に印刷されることに

注意してください。また、「水平拡張度」が「可変」ではなく、「縮小」または「固定」に設定されている場合、枠は、1論理ページに完全に収まる場合のみ印刷されることに注意してください。

アンカー・オブジェクトと連動

説明

「アンカー・オブジェクトと連動」プロパティは、1つのオブジェクトと、そのオブジェクトにアンカーされているオブジェクトを同じ論理ページに保持するかどうかを示します。「アンカー・オブジェクトと連動」を「はい」に設定すると、オブジェクトとアンカー・オブジェクトが同じ論理ページに収まらない場合、次のページに移動します。

繰り返し枠の「アンカー・オブジェクトと連動」を「はい」に設定した場合、繰り返し枠の最初のインスタンスは、アンカー・オブジェクトと同じページに収めてください。そうしておかないと、「アンカー・オブジェクトと連動」の条件は満たされないこととなります。繰り返し枠以外のレイアウト・オブジェクトの「アンカー・オブジェクトと連動」を「はい」に設定した場合、繰り返し枠の最初のインスタンスは、アンカー・オブジェクトと同じページに収めてください。

2つのオブジェクト間のアンカーは、明示的でも暗黙的でもかまいません。そのため、「アンカー・オブジェクトと連動」は、2つのオブジェクト間に明示的にアンカーを作成しなくても有効になることがあります。

適用対象

レイアウト・オブジェクト

必須/オプション

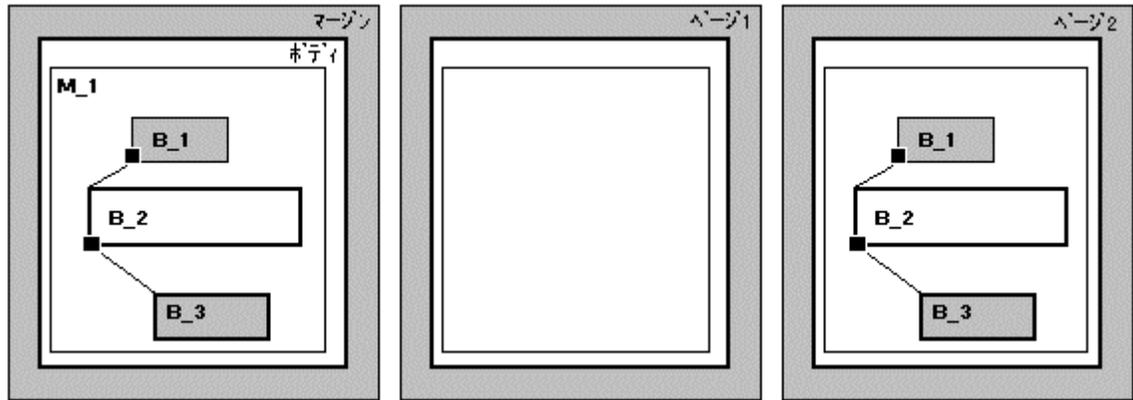
オプション

デフォルト

いいえ

「アンカー・オブジェクトと連動」の例

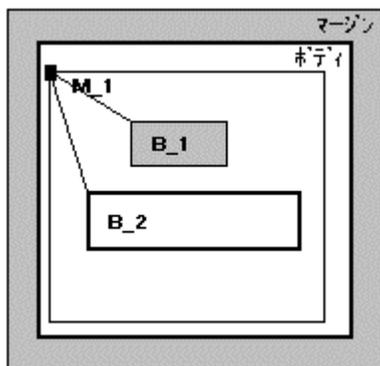
3つのボイラープレート・オブジェクトを作成したとします。



B_2およびB_3の「アンカー・オブジェクトと連動」は「はい」に設定されています。Report Builderが最初にこれらのオブジェクトを印刷しようとした論理ページには、B_1およびB_2を収めることはできますが、B_3を収めることはできません。B_3の「アンカー・オブジェクトと連動」が「はい」に設定されていて、アンカー・オブジェクトがB_2なので、B_2およびB_3は次の論理ページに移動します。B_2の「アンカー・オブジェクトと連動」も「はい」に設定されており、アンカー・オブジェクトがB_1なので、B_1も次の論理ページに移動します。

「アンカー・オブジェクトと連動」の制限

- 「アンカー・オブジェクトと連動」は、オブジェクトがフォーマットされる最初の論理ページにのみ適用されます。「アンカー・オブジェクトと連動」は後続のページでは、無視されます。
- オブジェクトとそのアンカー・オブジェクトを次の論理ページに移動すると、現行の論理ページがブランクになったり、繰返しオブジェクト（たとえば、ページ・ヘッダー）のみになる場合、「アンカー・オブジェクトと連動」は適用されません。これによって、レポートに不要なブランク・ページができなくなります。
- アンカー・オブジェクトが繰返し枠であり、現行のオブジェクトがその繰返し枠の外側にある場合、アンカー・オブジェクトとは、繰返し枠全体（繰返し枠のすべてのインスタンス）になります。
- 2つのオブジェクトが、それらを囲む枠にアンカーされており、その枠の「垂直拡張度」または「水平拡張度」（またはその両方）が「縮小」または「可変」の場合、「アンカー・オブジェクトと連動」の条件を満たすには、これら3つのオブジェクトがすべて同じページでフォーマットする必要があります。2つのオブジェクトが、枠ではなく、繰返し枠に囲まれている場合、このルールは、その繰返し枠の各インスタンスに適用されます。この図では、「アンカー・オブジェクトと連動」の条件を満たすには、B_1およびB_2、M_1が、すべて同じページ上でフォーマットする必要があります。



名前

説明

「名前」プロパティは、Report Builderが現行のオブジェクトの識別に使用する名前です。30バイト以内の有効な名前を入力します。名前についての詳細を以下に示します。

適用対象

オブジェクト

必須/オプション

必須

デフォルト

ボイラープレート	B_ <i>n</i> 。この <i>n</i> は、レポートのボイラープレート・オブジェクト名の中で一意の数値です。
ボタン・オブジェクト	U_ <i>n</i> 。この <i>n</i> は、レポートのフィールド名の中で一意の数値です。
チャート・オブジェクト	D_ <i>n</i> 。この <i>n</i> は、レポートのチャート・オブジェクト名の中で一意の数値です。
データベースの列	SQL問合せ（表示のみ）のSELECT文で参照される列名。
フィールド	F_ <i>n</i> 。この <i>n</i> は、レポートのフィールド名の中で一意の数値です。
式列	CF_ <i>n</i> 。この <i>n</i> は、レポートの式名の中で一意の数値です。
枠	M_ <i>n</i> 。この <i>n</i> は、レポートの枠名の中で一意の数値です。
問合せ	Q_ <i>n</i> 。この <i>n</i> は、レポートの問合せ名の中で一意の数値です。

グループ	G_queryname。このquerynameは、問合せの名前で、グループの中で一意の数値です。問合せがデフォルト名（たとえば、Q_1）の場合、デフォルト・グループ名は、G_nです。このnは、レポートのグループ名の中で一意の数値です。グループが問合せに関係付けられていない（つまり、クロス積グループ）場合のデフォルト・グループ名も、G_nです。
マトリックス・オブジェクト	X_n。このnは、レポートのマトリックス名の中で一意の数値です。
OLE2オブジェクト	B_n。このnは、レポートのOLE2オブジェクト名の中で一意の数値です。
パラメータ	P_n。このnは、レポートの列名の中で一意の数値です。問合せでバインド参照を作成することによってパラメータを作成する場合、問合せで使用される名前はデフォルトになります。
プレースホルダ列	CP_n。このnは、レポートのプレースホルダ名の中で一意の数値です。
繰返し枠	R_n。このnは、レポートの繰返し枠名の中で一意の数値です。
ソート列	(Oracle Expressのみ) S_dimension。このdimensionはソート列が参照するデータベース列の名前です。
サマリー列	CS_n。このnは、レポートのサマリー名の中で一意の数値です。

名前の制限

- データベース列の場合、名前は編集できません。
- Report Builderでは、データベースの列のデフォルト名として、問合せから列名および別名、式のテキストを使用します。
- システム・パラメータの名前は、変更できません。

後で改ページ

説明

「後で改ページ」プロパティは、オブジェクトのすべての子オブジェクトを次のページに移動する場合に指定します。つまり、このオブジェクトに対するアンカー（暗黙的または明示的）の子オブジェクトである、すべてのオブジェクトは、「前で改ページ」が「はい」に設定されているかのように処理されます。「後で改ページ」が設定されたオブジェクトの下にあるすべてのオブジェクトが、次のページに移動するわけではないことに注意してください。

適用対象

レイアウト・オブジェクト

必須/オプション

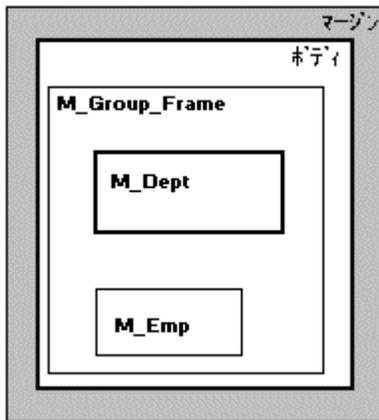
オプション

デフォルト

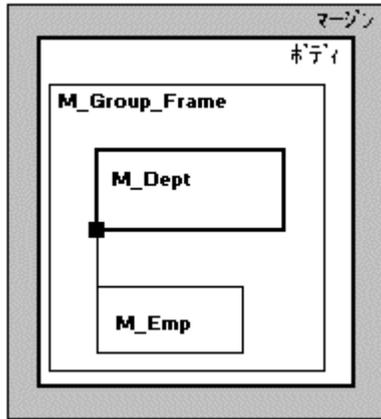
いいえ

「後で改ページ」の制限

- 「後で改ページ」は、オブジェクトの子オブジェクトのフォーマットを遅らせます。同位オブジェクトの場合Report Builderがフォーマット順序の判別に使用できる階層（つまり、親子の関係）がないので、同位オブジェクトのフォーマットは遅れません。したがって、「後で改ページ」が有効になるように、オブジェクトは親として他のオブジェクトに（暗黙的または明示的に）アンカーされている必要があります。

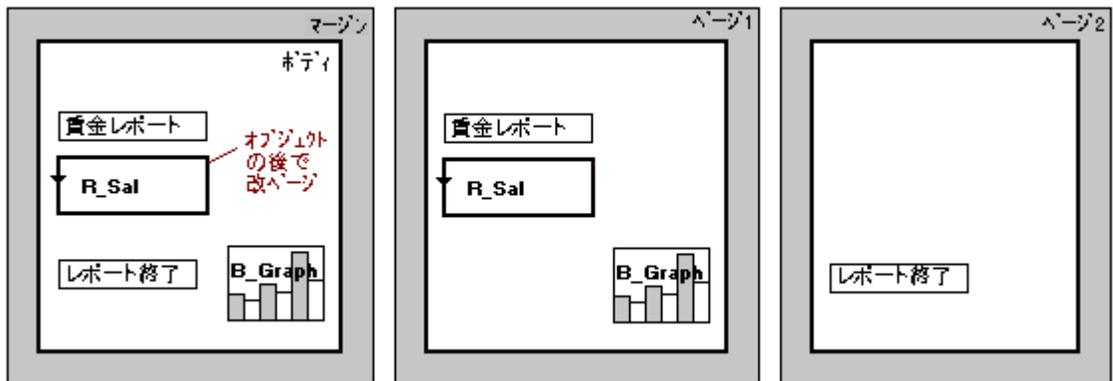


この図では、枠M_Deptの「後で改ページ」が「はい」に設定されています。M_DeptとM_Empは両方とも、「垂直拡張度」および「水平拡張度」が「固定」に設定されています。M_DeptおよびM_Empは親枠M_Darentの同位項目なので、この場合、「後で改ページ」は何も変わりません。



この図は、前の例と同じですが、M_Deptが親として明示的にM_Empにアンカーされている点が異なります。M_DeptはM_Empの親であるため、この場合、「後で改ページ」はM_Empのフォーマットを遅らせます。

- 繰り返し枠の場合、「後で改ページ」によって、論理ページの改ページは、繰り返し枠全体を印刷した後で行われます。繰り返し枠の各インスタンスの後で改ページするには、繰り返し枠のプロパティ・シートで「ページ当りの最大レコード数」を1に設定することも必要です。
- 「後で改ページ」が「はい」に設定されているオブジェクトの下にあるオブジェクトは、次のページに移動しないことがあります。オブジェクトが、「後で改ページ」が「はい」に設定されているオブジェクトの子オブジェクトではない場合、そのページに収まれば、「後で改ページ」が「はい」に設定されているオブジェクトと同じページに印刷されることがあります。



レイアウト中のR_Salの下に表示されるB_Graphの印刷が実際に終わるのは、R_Salが印刷される1ページ目であることに注意してください。これは、B_GraphがR_Salの子ではないために発

生じます。B_GraphがR_Salに対する子として明示的にアンカーされている場合、B_Graphは1ページ目ではなく、2ページ目に出力されます。

- オブジェクトが、外部オブジェクトから（明示的または暗黙的に）アンカーされていない場合、「後で改ページ」は効果がありません。

前で改ページ

説明

「前で改ページ」プロパティは、初めに印刷しようとするページの次のページで、オブジェクトをフォーマットする場合に設定します。「前で改ページ」が設定されたオブジェクトの下にあるすべてのオブジェクトが、次のページに移動するわけではないことに注意してください。

適用対象

レイアウト・オブジェクト

必須/オプション

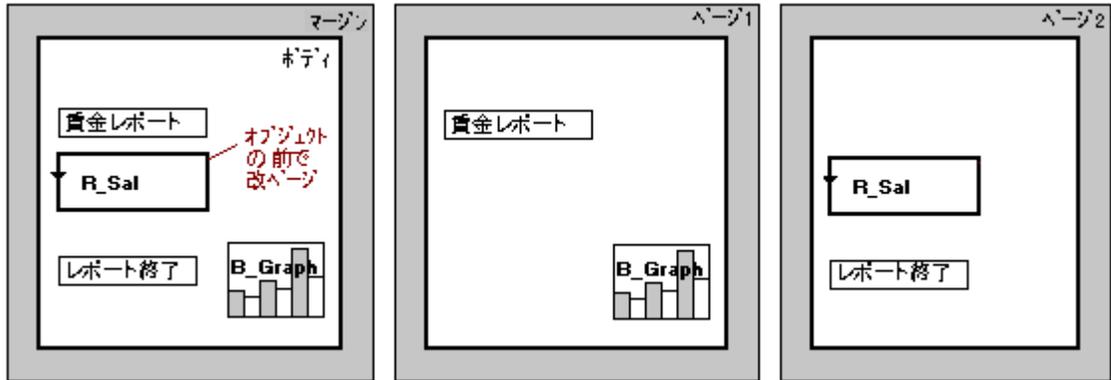
オプション

デフォルト

いいえ

「前で改ページ」の制限

- 繰返し枠の場合、「前で改ページ」によって、論理ページの改ページは、繰返し枠を印刷する前に行われます。繰返し枠の各インスタンスの前で改ページするには、繰返し枠のプロパティ・シートで「ページ当りの最大レコード数」を1に設定することも必要です。
- 「前で改ページ」が「はい」に設定されているオブジェクトの下にあるオブジェクトは、次のページに移動しないことがあります。オブジェクトがあるページに印刷されるよう設定されており、「前で改ページ」によって次のページに移動する場合、十分な空白があれば、そのオブジェクトが印刷されるはずであった位置に、その他のオブジェクトが印刷されることがあります。



レイアウト中のR_Salの下に表示されるB_Graphの印刷が実際に終わるのは、2ページ目ではなく、1ページ目であることに注意してください。

ページ保護

説明

「ページ保護」プロパティは、同じ論理ページにオブジェクト全体とその内容を保持するかどうかを示します。「ページ保護」を「はい」に設定すると、オブジェクトの内容が現行の論理ページに収まらない場合は、オブジェクトおよびその内容のすべては次の論理ページに移動します。

適用対象

レイアウト・オブジェクト

必須/オプション

オプション

デフォルト

いいえ

ページ保護の例 (小さなオブジェクト)

さまざまな列や集計を表示するフィールドを数多く持つ枠M_1を作成したとします。読みやすくするために、M_1内のすべてのフィールドを同じ論理ページに印刷し、列および集計結果を一緒に参照できるようにします。

M_1に「ページ保護」を指定します。 Report Builderが印刷しようとする最初の論理ページに、M_1およびすべてのフィールドが収まらない場合、Report Builderは、次の論理ページに印刷しようとしています。

ページ・プロテクトなし

Figure 1: Page1 (Page 1 of 2) - No page protection. The page contains two tables. The first table is titled '資金レポート' and lists departments. The second table is titled 'M_1' and lists employee details.

Deptno	Dname	Loc
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

M_1	Empno	Sal
	7369	800
	7499	1600
	7521	1250

Figure 2: Page2 (Page 2 of 2) - No page protection. The page contains only the 'M_1' table.

Empno	Sal
7566	2975
7654	1250
7698	2850
7782	2450
7788	3000

ページ・プロテクトあり

Figure 3: Page1 (Page 1 of 2) - Page protection. The page contains two tables. The first table is titled '資金レポート' and lists departments. The second table is titled 'M_1' and lists employee details.

Deptno	Dname	Loc
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

M_1	Empno	Sal
	7369	800
	7499	1600
	7521	1250
	7566	2975
	7654	1250
	7698	2850
	7782	2450
	7788	3000

Figure 4: Page2 (Page 2 of 2) - Page protection. The page contains only the 'M_1' table.

Empno	Sal
7369	800
7499	1600
7521	1250
7566	2975
7654	1250
7698	2850
7782	2450
7788	3000

ページ保護の例 (大きなオブジェクト)

さまざまな列や集計を表示するフィールドを数多く持つ枠M_1を作成したとします。読みやすくするために、M_1内のすべてのフィールドを同じ論理ページに印刷し、列および集計結果を一緒に参照できるようにします。

M_1に「ページ保護」を指定します。Report Builderが印刷しようとする最初の論理ページに、M_1およびすべてのフィールドが収まらない場合、Report Builderは、次の論理ページに印刷しようとしています。

しかし、Report Builderが印刷しようとする論理ページの2ページ目にM_1およびすべてのフィールドが収まらないとします。Report Builderは、論理ページに収まるのみM_1を印刷し、残りは次の論理ページに印刷します。

ページ保護の例 (グループ・レポート)

グループ・レポートがあるとします。できる限り同じページにすべてのマスターおよびディテールを表示するとします。その場合は、マスター繰返し枠(最も外側の繰返し枠)の「ページ保護」を指定します。印刷しようとする最初のページに、ディテールおよびマスターが収まらない場合、次のページに印刷するように設定されます。

「ページ保護」の制限

- 「ページ保護」は、オブジェクトがフォーマットされる最初の論理ページのみ適用されます。「ページ保護」は後続のページでは、無視されます。
- 繰返し枠の場合、「ページ保護」は、繰返し枠の各インスタンスに適用されます。繰返し枠の各インスタンスおよび囲まれたオブジェクトは、「ページ保護」によってともに保持されます。ただし、繰返し枠の最初のページを除くページに表示される最初のインスタンスには、ページ保護は適用されません。たとえば、下の図では、ページ保護は、インスタンス1および2、4には適用されますが、インスタンス3には適用されません。インスタンス3は、繰返し枠の2ページ目でフォーマットされる最初のインスタンスです。この場合、ページ保護がインスタンス3に適用されていれば、2ページ目ではなく3ページ目から開始します。2ページ目にはほとんど何も印刷されません。

論理ページ1		論理ページ2	
Deptno	Status	Deptno	Status
10	XXXXXXXXXX XXXXXXXXXX	30	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX
	Instance1		Instance3
20	XXXXXXXXXX XXXXXXXXXX		
	Instance2		
論理ページ3		論理ページ4	
Deptno	Status	Deptno	Status
30	XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	40	XXXXXXXXXX
	Instance3		Instance4

- 「ページ保護」は、マトリックスの一部である繰返し枠には対応していません。マトリックス・オブジェクトによって繰返し枠が関係付けられている場合、「ページ保護」を設定しても効果がありません。
- オブジェクトとその内容を次の論理ページに移動すると、現行の論理ページが空白になったり、繰返しオブジェクト（たとえば、ページ・ヘッダー）のみになる場合、「ページ保護」は適用されません。これによって、レポートに不要な空白・ページができなくなります。
- フォーマット時に「ページ保護」によって、オブジェクトが次の論理ページに移動する場合、ページ保護されたオブジェクトの横にあるオブジェクトは、一緒に移動することもあれば、移動しないこともあります。ページ保護されたオブジェクトの横にあるオブジェクトが、ページ保護されたオブジェクトに（明示的または暗黙的に）アンカーされていない場合、そのオブジェクトが、ページ保護されたオブジェクトが収まりきらない領域に収まれば、その論理ページに印刷されます。それ以外の場合は、そのオブジェクトはページ保護されたオブジェクトとともに次の論理ページに移動します。

基本印刷オン

説明

「基本印刷オン」プロパティは、現行のオブジェクトの「印刷オブジェクト・オン」プロパティに基づいたオブジェクトです。たとえば、「印刷オブジェクト・オン」を「全ページ」に、「基本印刷オン」を「アンカー・オブジェクト」に設定した場合、現行のオブジェクトは、アンカー・オブジェクト（親オブジェクト）が表示される論理ページごとに印刷されるようにトリガーされます。

値

アンカー・オブジェクト	現行のオブジェクトが明示的または暗黙的にアンカーされる親オブジェクトです。
インクローズ・オブジェクト	現行のオブジェクトを囲むオブジェクトです。

適用対象

レイアウト・オブジェクト

必須/オプション

必須

デフォルト

アンカー・オブジェクト

印刷オブジェクト・オン

説明

「印刷オブジェクト・オン」プロパティは、オブジェクトがレポートに出力される頻度です。「印刷オブジェクト・オン」オプションは、オブジェクトが「基本印刷オン」のオブジェクトについて印刷される論理ページを示しています。

注意: オブジェクトが論理ページに印刷されるように設定されていても、その論理ページに印刷されるとは限りません。別の設定（たとえば、「前で改ページ」）またはそのページで使用できる空白の量によって、Report Builderは、最初のトリガー時には印刷されないはずであったページにオブジェクトを印刷することがあります。

これらのオプションを適用する場合、Report Builderは、オブジェクトの1ページ目を、オブジェクトの一部が印刷される最初の論理ページと見なします。同様に、最後のページを、オブジェクトの一部が印刷される最後の論理ページと見なします。たとえば、「印刷オブジェクト・オン」を「第1ページ」に、「基本印刷オン」を「インクローズ・オブジェクト」に設定した場合、オブジェクトは、インクローズ・オブジェクトが表示される最初の論理ページに印刷されます。

値

全ページ	オブジェクトおよびその内容のすべては「基本印刷オン」のオブジェクトのすべての論理ページに印刷されます。オブジェクトは、「基本印刷オン」のオブジェクトのすべてのオーバーフロー・ページ上で繰り返され、必要があれば論理ページの境界線で切り捨てられます。
最初のページ以外の全ページ	オブジェクトおよびその内容のすべては、「基本印刷オン」のオブジェクトの最初の論理ページを除くすべての論理ページに印刷されます。オブジェクトは、「基本印刷オン」のオブジェクトのオーバーフロー・ページ上でフォーマットされ、必要があれば論理ページの境界線で切り捨てられます。
最後のページ以外の全ページ	オブジェクトおよびその内容のすべては、「基本印刷オン」のオブジェクトの最後の論理ページを除くすべての論理ページに印刷されます。オブジェクトは、「基本印刷オン」のオブジェクトの最後のページを除くすべてのオーバーフロー・ページ上に繰り返され、必要があれば論理ページの境界線で切り捨てられます。
デフォルト	Report Builderでは、「印刷オブジェクト・オン」が「第1ページ」または*「最終ページ」のいずれかに設定されるようにオブジェクトが配置されます。（アスタリスク（*）は、Report Builderによって設定された値であることを示します。）
第1ページ	オブジェクトおよびその内容のすべては、「基本印刷オン」のオブジェクトの最初の論理ページだけに印刷されます。オブジェクトは、フォーマットされ、必要があれば後続のページにオーバーフローします。
最終ページ	オブジェクトおよびその内容のすべては、「基本印刷オン」のオブジェクトの最後の論理ページだけに印刷されます。オブジェクトは、「基本印刷オン」のオブジェクトの後にフォーマットされ、必要があれば後続のページにオーバーフローします。

適用対象

レイアウト・オブジェクト

必須/オプション

必須

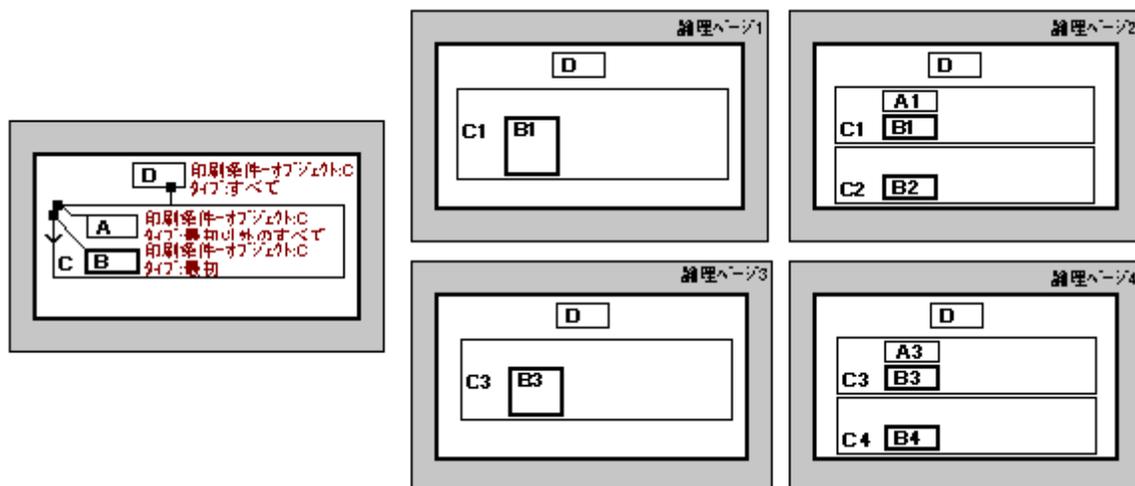
デフォルト

オブジェクトにより異なる。

印刷オブジェクト・オンの例（すべて）

グループGroup_1内の列の値を合計したサマリー列Col_Sumを作成したとします。 Col_Sumは、Group_1からの値を持つ繰返し枠（R_1）内で囲まれたフィールド（F_1）のソースです。 F_1の「基本印刷オン」を「全ページ」に、「基本印刷オン」を「インクローズ・オブジェクト」に指定した場合、各論理ページのR_1のすべてのインスタンスの中に印刷されます。

印刷オブジェクト・オンの例（オーバーフロー）



この図は、レイアウトおよび出力結果を示しています。

ボイラープレート・オブジェクトDが繰返し枠Cの外側にあるので、ボイラープレート・オブジェクトDの「基本印刷オン」の設定は、繰返し枠全体に適用されます。ボイラープレート・オブジェクトAおよびBの「基本印刷オン」の設定は、繰返し枠の中にあるので、繰返し枠Cの各インスタンスに適用されます。

オブジェクトDは、繰返し枠Cが出力されるすべての論理ページに出力されます。しかし、オブジェクトAは、繰返し枠Cのインスタンスがオーバーフローしたときにのみ印刷されます。この場合「印刷オブジェクト・オン」の「最初のページ以外の全ページ」を指定すると、オブジェクトは、インスタンスが印刷される最初のページを除いて、繰返し枠の各インスタンスの各論理ページに印刷されるように設定されます。

繰返し枠Cの中にあるのは、2つのオブジェクト（ボイラープレート・オブジェクトAおよびB）のみなので、ボイラープレート・オブジェクトBがオーバーフローした場合に2ページ目にオーバーフローするのみです。オブジェクトAおよびBの「印刷オブジェクト・オン」の設定（「最初のページ以外の全ページ」および「第1ページ」）を見ると、これらのオブジェクト2つは同じページに印刷されることはあり得ない、と思うかもしれませんが、しかし、この場合、この2つのオブジェクトは同じページにしか出力できません。

「印刷オブジェクト・オン」の制限

注意： オブジェクトの内容が印刷を開始する論理ページ内に収まらない場合、「印刷オブジェクト・オン」を「全ページ」または「最初のページ以外の全ページ」、「最後のページ以外の全ページ」に設定したオブジェクトは、後続のページにはオーバーフローできません。この場合、オブジェクトの内容は切り捨てられ、オブジェクトの内容のうち最初の論理ページに収まった部分のみが、印刷するように設定されているすべての論理ページに印刷されます。

- オブジェクトの内容が印刷を開始する論理ページ内に収まらない場合、「印刷オブジェクト・オン」を「第1ページ」または「最終ページ」に設定したオブジェクトは、後続のページにオーバーフローできます。その結果、最初または最後のページのみに印刷されるようにトリガーされていても、最初または最後のページの次のページにオーバーフローすることがあります。オーバーフローすると、「基本印刷オン」の「第1ページ」と「最初のページ以外の全ページ」は、必ずしも互いに排他であるとは限りません。たとえば、あるオブジェクトの「印刷オブジェクト・オン」が「第1ページ」であり、もう1つのオブジェクトは「最初のページ以外の全ページ」で、「基本印刷オン」の設定がどちらも同じ場合、2つのオブジェクトが同じページに表示されることはあり得ないと思うかもしれませんが、しかし、「第1ページ」に設定されたオブジェクトが2ページ目にオーバーフローした場合、この2つのオブジェクトは同じページに出力されることがあります。
- 囲み枠または繰返し枠に明示的または暗黙的にアンカーされたオブジェクトは、「印刷オブジェクト・オン」を「最終ページ」または「最後のページ以外の全ページ」に設定することはできません。囲み枠または繰返し枠内で別のオブジェクトに明示的または暗黙的にアンカーされたオブジェクトは、「基本印刷オン」を「インクローズ・オブジェクト」にして、「印刷オブジェクト・オン」を「最終ページ」または「最後のページ以外の全ページ」に設定することはできません。
- 「基本印刷オン」に設定されたオブジェクトがそのオブジェクトを囲む繰返し枠の場合、この設定はオブジェクトがアンカーされている繰返し枠の各インスタンスに適用されます。
- 「基本印刷オン」に設定されたオブジェクトが繰返し枠であり、そのオブジェクトが繰返し枠の外側にある場合、この設定は繰返し枠全体に適用されます。たとえば、「印刷オブジェクト・オン」の「第1ページ」は、繰返し枠が印刷されるようにトリガーされている最初の論理ページのことです。
- 「基本印刷オン」の設定の適用対象が繰返し枠全体であるかまたは繰返し枠のインスタンスであるかにかかわらず、「印刷オブジェクト・オン」の設定は、論理ページに必ず適用されます。たとえば、「基本印刷オン」のオブジェクトが繰返し枠であり、そのオブジェクトが

繰返し枠の外側にあるとします。この場合、「基本印刷オン」は、繰返し枠のインスタンスに適用されます。しかし、「印刷オブジェクト・オン」の「第1ページ」は、繰返し枠の最初のインスタンスのことではありません。それは、繰返し枠が印刷されるように設定されている各インスタンスの最初の論理ページのことです。

- マージン領域にあるオブジェクトは、各物理ページで再表示されます。その結果、一部の「印刷オブジェクト・オン」の設定は、マージン領域のオブジェクトには意味がありません。オブジェクトは、各ページで再表示されるので、常にオブジェクトの最初のページが出力されているような状態になります。「第1ページ」は、「全ページ」と同じです。「最終ページ」および「最後のページ以外の全ページ」は、「最後」になることがないため、無効です。「最初のページ以外の全ページ」の場合、オブジェクトは「最初」の次になることがないため、オブジェクトは出力されません。

前のプリンタ・コード

説明

「前のプリンタ・コード」および「後のプリンタ・コード」は、オブジェクトに対する特別な印刷指示(たとえば、特別なフォント・サイズまたは反転表示)を定義するプリンタ・エスケープ・シーケンスの参照元です。「前のプリンタ・コード」プロパティは、オブジェクトの各行の前に実行されるプリンタ・エスケープ・シーケンスを参照します。プリンタ・エスケープ・シーケンスは、オブジェクトの印刷が準備されてから、実際に印刷するまでに挿入されます。

値

256バイトまでの有効なプリンタ・コード参照を入力します。入力値は、&number形式の文字列です。このnumberは、パッケージReport Builderプリンタ・コードかまたは自分で作成したプリンタ・コードに割り当てられた番号です。

適用対象

レイアウト・オブジェクト

使用上の注意

- 大抵の場合、レイアウト・オブジェクト用にPL/SQLを記述すれば、どのような特別なフォーマットも扱えます。

PL/SQL PL/SQLを使用してこの属性を定義するには、SRW.SET BEFORE PRINTING CODEプロシージャを使用します。

後のプリンタ・コード

説明

「後のプリンタ・コード」および「前のプリンタ・コード」は、オブジェクトに対する特別な印刷指示(たとえば、特別なフォント・サイズまたは反転表示)を定義するプリンタ・エスケープ・シーケンスの参照元です。「後のプリンタ・コード」プロパティは、オブジェクトの各行の後に実行されるプリンタ・エスケープ・シーケンスを参照します。プリンタ・エスケープ・シーケンスは、オブジェクトの印刷が準備されてから、実際に印刷するまでに挿入されます。

値

256バイトまでの有効なプリンタ・コード参照を入力します。入力値は、&number形式の文字列です。このnumberは、パッケージReport Builderプリンタ・コードかまたは自分で作成したプリンタ・コードに割り当てられた番号です。

適用対象

レイアウト・オブジェクト

使用上の注意

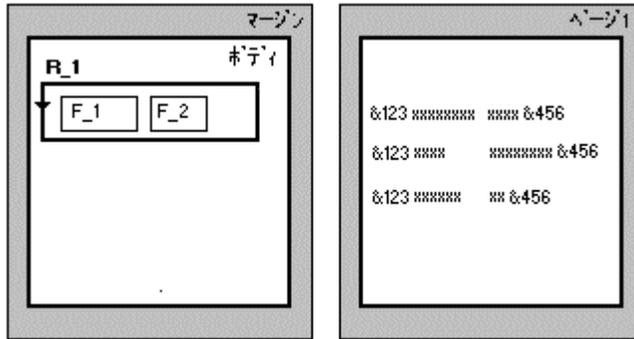
- 大抵の場合、レイアウト・オブジェクト用にPL/SQLを記述すれば、どのような特別なフォーマットも扱えます。

PL/SQL PL/SQLを使用してこの属性を定義するには、SRW.SET AFTER PRINTING CODEプロシージャを使用します。

プリンタ・コードの特別なフォントの例

プリンタ定義ファイルに、2つのプリンタ・コード&123および&456を作成したとします。プリンタは、コード&123によって特別なフォントを印刷し、コード&456によって標準のフォントを印刷します。繰返し枠(R_1)内のすべてのフィールドを特別なフォントを使用して印刷とします。プリンタ・コードは、各フィールドに使用することも、繰返し枠全体に使用することもできます。後者の方が、作成およびメンテナンスが簡単です。

R_1繰返し枠のプロパティ・シートでは、「前のプリンタ・コード」に&123、「後のプリンタ・コード」に&456を入力します。R_1が印刷されると、R_1の各行印刷前および後に、プリンタ・コードが実行されます。



この図は、各プリンタ・コードが起動されるポイントを示しています。

プリンタ・コードの制限

- プリンタ・コードは、キャラクタ・モードでレポートを実行しているときのみ使用されます。ビットマップ・モードで実行している場合には、使用されません。

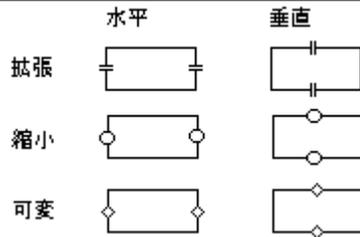
垂直拡張度

説明

「垂直拡張度」プロパティは、実行時にオブジェクトまたはデータをその中に収めるためにオブジェクトの垂直サイズを変更する方法を示します。

- 枠および繰返し枠の場合、このプロパティは、枠または繰返し枠のサイズをその中のオブジェクトに合わせて変更する必要があるかどうかを定義します。
- テキストが含まれるオブジェクトの場合、このプロパティは、フィールドまたはボイラールートがテキストのサイズに合わせて変更される必要があるかどうかを定義します。サイズ固定のテキストは、定義されたオブジェクト・サイズの範囲内でラップされますが、十分な空間がなければ切り捨てられる場合があります。数値データまたは日付データを定義されたサイズ内に収められない場合、アスタリスク (*) が表示されます。
- イメージおよび描画、チャート・オブジェクトの場合、Report Builderによって、比例したスケール変更が行われます。イメージおよび描画、チャート・オブジェクトの拡張度オプションによって、スケールが決まります。

サイズ設定



値

縮小	フォーマットされたオブジェクトまたはデータの高さが低い場合、オブジェクトの垂直サイズは小さくなります。データの切捨てが行われる場合があります。例を参照してください。（このオプションは、「縮小のみ行われ、拡張されない」ことを意味しています。）
拡張	フォーマットされたオブジェクトまたはデータの高さが高い場合、オブジェクトの垂直サイズは大きくなります。（このオプションは、「拡張のみ行われ、縮小されない」ことを意味しています。）
固定	オブジェクトの高さは、その中のオブジェクトまたはデータのサイズに関係なく、各論理ページで同じです。データの切捨てが行われる場合があります。例を参照してください。オブジェクトの高さは、レポート・エディタ内のオブジェクトの高さに定義されます。
可変	オブジェクトは垂直方向に拡張または縮小され、その中のオブジェクトまたはデータは適切なサイズに調整されます。つまり、実行時のオブジェクトの高さは、レポート・エディタ内に示される高さには影響されないということです。

適用対象

レイアウト・オブジェクト

必須/オプション

必須

デフォルト

オブジェクトにより異なる。

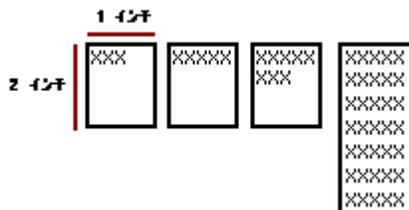
使用上の注意

- 「垂直拡張度」および「水平拡張度」を「固定」に設定して、レポート内にチャート・オブジェクトを作成した場合、Report Builderによってオブジェクトに定義したサイズに合わせて図表をスケーリングします。

垂直拡張度の例（拡張）

次の特性を持つフィールドがあるとします。

特性	値
幅	1インチ
水平拡張度	固定
高さ	2インチ
垂直拡張度	拡張



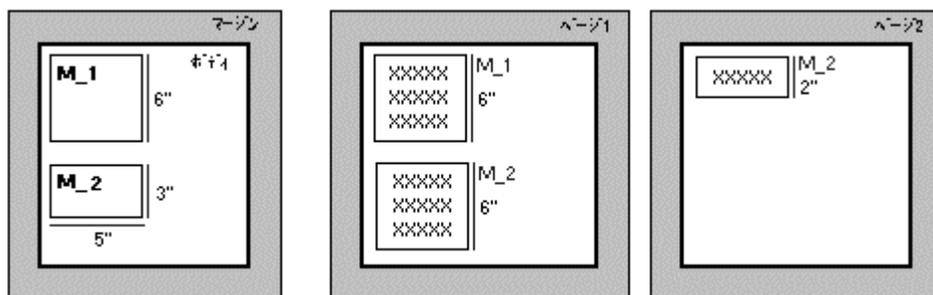
垂直拡張度の例（可変）

次の特性を持つ、M_2という枠があるとします。

特性	値
幅	5インチ
水平拡張度	固定
垂直拡張度	可変

また、レポートの「高さ」が12インチだとします。その中のオブジェクトのサイズにより、M_2をフォーマットするには垂直方向に8インチ必要です。別の枠M_1は、論理ページ上でM_2の前にあり、6インチ必要だとします。

次の図は、枠がどのようにフォーマットされるかを示しています。



M_2全体は、最初の論理ページには収まりません。「垂直拡張度」が「可変」なので、M_2の最初の6インチ分は、最初の論理ページに印刷され、残りの2インチ分は論理ページの2ページ目に

印刷されます。「垂直拡張度」を「固定」にしていた場合は、M_2全体が論理ページの2ページ目に印刷されます。

「垂直拡張度」の制限

注意: オブジェクトの内容が定義された高さを超える場合、「垂直拡張度」および「水平拡張度」によって、内容が次ページにオーバーフローするのかがまたは切り捨てられるのかが判断されます。イメージおよび描画、チャート・オブジェクトは、オーバーフローできません。イメージおよび描画、Graphics Builder図表の詳細は、次のルールを参照してください。

オブジェクト	固定/縮小	拡張	可変
繰返し枠 インスタンスおよび枠 (内容 <= 定義された高さ)	オーバーフローなし	オーバーフローなし	オーバーフローなし
繰返し枠 インスタンスおよび枠 (内容 > 定義された高さ)	オーバーフローなし	オーバーフローなし (ページに収まらない場合)	オーバーフローなし (ページに収まらない場合)
フィールドおよびボイラープレート (内容 <= 定義された高さ)	オーバーフローなし	オーバーフローなし	オーバーフローなし
フィールドおよびボイラープレート (内容 > 定義された高さ)	切捨て	オーバーフローなし (ページに収まらない場合)	オーバーフローなし (ページに収まらない場合)

* 切捨てが発生しないようにするには、フィールドまたはボイラープレートの「垂直拡張度」を「可変」にして、「垂直拡張度」が「固定」の枠で囲んでください。

追加情報: データを切り捨てる場合、問合せでSQLのSUBSTRファンクションを使用してください。たとえば、列の最初の4文字のみを使用する場合、SELECT文でSUBSTRを使用して、データベースからその文字のみを取り出します。

- オブジェクトのフォーマット時に、「垂直拡張度」が「固定」に設定されている場合、論理ページにオブジェクト全体を含むことができる十分な空白があれば、Report Builderは論理ページ上にオブジェクトをフォーマットするのみです。
- 「垂直拡張度」が「拡張」に設定されている場合、論理ページに定義されたオブジェクトの高さ(つまり、最小値)が収まるのみの十分な空白があれば、Report Builderは論理ページ上でオブジェクトをフォーマットするのみです。フォーマットが完了しない場合、オブジェクトは次ページにオーバーフローします。
- 「垂直拡張度」が「可変」または「縮小」に設定されている場合、Report Builderは、最初の論理ページにできるのみフォーマットし、必要があれば、次の論理ページ上でオブジェクトのフォーマットを完了します。
- 数値がフィールドに収まらない場合は、値のかわりにアスタリスク(*)がフィールドに表示されます。値を表示するには、オブジェクトのサイズを変更するかまたはサイズ変更を「拡張」または「可変」に変更します。

- 次のサイズ・ルールは、イメージおよび描画、Graphics Builder図表を含むオブジェクト（たとえば、フィールドまたはボイラプレート）に適用されます。
 - レイアウト・オブジェクトのサイズが可変（つまり、「垂直拡張度」および「水平拡張度」の両方が「可変」）の場合、その内容は、ソース（たとえば、ソース・ファイル）と同じサイズで表示されます。
 - レイアウト・オブジェクトのサイズが固定（つまり、「垂直拡張度」および「水平拡張度」の両方が「固定」）の場合、その内容は、比例してスケール変更されます。これによって、内容とオブジェクトの境界線の間の領域が空白になる、つまり、「隙間」ができることがあります。たとえば、オブジェクトの「垂直拡張度」および「水平拡張度」の両方が「固定」であり、イメージがオブジェクトに対して大きすぎる場合、そのイメージの一方のみはそのままのサイズになります。その他の方向では、そのイメージはオブジェクトより短くなります。このような隙間ができないようにするには、サイズの設定を変更（つまり、水平方向または垂直方向、その両方を「可変」に）できます。または、手動でオブジェクトのサイズを変更できます。
 - イメージおよび描画、Graphics Builder図表を含むオブジェクトが一方のみ「固定」に設定されている場合、Report Builderは常に「固定」のサイズまで内容をサイズ変更します。
 - オブジェクトのサイズ変更の一方が「縮小」で、他方が「固定」の場合、内容はクリップされることがあります。このようにクリップされないようにするには、サイズの設定を変更（たとえば、水平方向および垂直方向の両方を「可変」に）します。または、手動でオブジェクトのサイズを変更します。
 - オブジェクトのサイズ変更が可変（つまり、「垂直拡張度」および「水平拡張度」が「可変」）で、オブジェクトが囲みオブジェクト（たとえば、枠または論理ページ）に収まらない場合、オブジェクトをフォーマットできないので、レポートは無限ループします。
- レポート出力に表示される値がレポートのフォーマットに依存する場合、値を表示するフィールドまたはボイラプレート・オブジェクトは「垂直拡張度」および「水平拡張度」が「固定」に設定されている必要があります。（これは、ページ依存参照と呼ばれます。）サイズが固定されていない場合、Report Builderは実行時に固定に変更します。

ページ依存参照の典型的な例は、「ソース」が「総ページ数」に設定されたフィールドです。論理ページの合計値は、レポート全体がフォーマットされるまでわかりません。しかし、Report Builderは、レポート終了前にフィールドをフォーマットする必要があり、レポートが完全にフォーマットされて作成するフィールドの大きさが決まるまで待つことはできません。そのためにはフィールドのサイズが固定されている必要があるため、フィールドのソース値が決まる前に、Report Builderはフィールド用に確保するスペースの大きさを認識しています。

次はページ依存参照のケースです。フィールドまたはボイラプレート・オブジェクトのソース値がレポートのフォーマットによって決まるため、そのサイズは固定にしてください。

- サマリー列が「リセット位置」または「計算位置」が「ページ」の場合、その値を表示するフィールドまたはボイラプレート・オブジェクトのサイズは固定にしてください。

- 式列またはサマリー列が、「リセット位置」または「計算位置」が「ページ」のサマリー列に依存する場合、その値を表示するフィールドまたはボイラプレート・オブジェクトのサイズは固定にしてください。
- フィールドの「ソース」が「ページ番号」または「パネル番号」、「物理ページ番号」、「総ページ数」、「総パネル数」、「総物理ページ数」の場合、フィールド（フィールド自体またはボイラプレート・オブジェクトのいずれか）を表示するオブジェクトのサイズは、固定にしてください。
- フィールドが「ソース」としてページ番号を持っており、「ページ番号付け」ダイアログ・ボックスの「リセット位置」が繰返し枠の場合、値を表示するレイアウト・オブジェクトのサイズは、固定にしてください。

注意: ページ依存参照を含むフィールドの「可視」が「いいえ」に設定されている場合、このフィールドの「水平拡張度」は「可変」または「拡張」、「縮小」にしてください。そのフィールドを参照するどのボイラプレート・オブジェクトも、サイズは固定にしてください。

ハイパーリンク

説明

「ハイパーリンク (HTML/PDF)」プロパティは、WebブラウザまたはPDFビューア内でオブジェクトをクリックしたときに、文書または文書内の宛先を表示するURLリンク仕様部です。

値

次の宛先に対する有効なリンクを指定します。

- 現行文書内の宛先（#my_dest_nameなど）
- ローカルの文書内の宛先（file:/private/somedoc.pdf#a_dest_nameなど）
- ローカル・マシン上の文書（file:/private/mynewdoc.pdfまたは file:///C:/temp/mynewdoc.pdfなど）
- リモート・マシン上の文書（http://www.newmach.com/newdoc.pdfまたは http://www.newmach.com/newdoc.html、ftp://www.reposit.com/filetoget.example、http://www.somemch.com/cgi-bin/webmenu?choice1など）
- リモートの文書内の宛先（http://www.newmach.com/newdoc.pdf#some_dest_nameなど）

適用対象

レイアウト・オブジェクト

デフォルト

ブランク

使用上の注意

- 「ハイパーリンク (HTML/PDF)」プロパティが枠に対して指定されている場合、このプロパティは「ハイパーリンク (HTML/PDF)」プロパティが指定されていないすべての子 (内部) オブジェクトに転送されます。プロパティを子オブジェクトに転送できない場合、枠の「ハイパーリンク (HTML/PDF)」の値は失われます。
- PDF形式のレポート出力は、ハイパーリンクとアプリケーション・コマンド・ライン・リンクの両方をインクルードできます。「アプリケーション・コマンド・ライン」プロパティがオブジェクトに設定されている場合は、そのプロパティがオブジェクトに適用されます。それ以外の場合は、「ハイパーリンク」プロパティが適用されます。
- 「ハイパーリンク (HTML/PDF)」プロパティが枠内のオブジェクトについて指定されている場合、オブジェクトをクリックすると、デフォルトで、ハイパーリンクの宛先を示すウィンドウ全体に、枠が置き換わります。枠のみを置き換えるには、"target=ファイル名"というHTMLを「追加のハイパーリンク属性」プロパティ設定に追加します。ファイル名は、置き換える枠の名前です。

PL/SQL

PL/SQLで出力するデータを制限するには、SRW.SET HYPERLINK プロシージャを使用します。

「ハイパーリンク (HTML/PDF)」の制限

- PDFビューアからリモート・サーバーまたはHTML文書へのWebリンクをたどるには、Webブラウザと連動するようにPDFビューアを構成する必要があります (たとえば、ヘルパー・アプリケーションとして構成したり、プラグインとしてWebブラウザにインストールする)。

ハイパーリンクの宛先

説明

「ハイパーリンクの宛先」プロパティは、オブジェクト用の一意識別子で、Webリンクの宛先として使用できます。

値

26の大文字または小文字のASCII文字または数字、アンダースコア文字のみを使用する有効な一意の名前。他の特殊文字は、アンダースコア文字に自動的に変換されます。

適用対象

レイアウト・オブジェクト

必須/オプション

宛先に自分を指定するオブジェクトの「ハイパーリンク (HTML/PDF)」プロパティを設定する場合、必須。

デフォルト

ブランク

使用上の注意

- 「ハイパーリンクの宛先」プロパティが枠に対して指定されている場合、プロパティは枠の上部左角に最も近い可視オブジェクトに転送されます。

PL/SQL

PL/SQLで出力するデータを制限するには、SRW.SET LINKTAGプロシージャを使用します。

ブックマーク

説明

「ブックマーク」プロパティは、マスターHTML文書のブックマーク枠またはPDFビューアのブックマーク領域に表示されるWebリンクです。ブックマークをクリックすると、ウィンドウの一番上に対応付けられたオブジェクトが表示されます。

値

インデントおよび順序付けに関する情報がないテキスト文字列。ブックマークは、レポートがオブジェクトを渡す順に、ブックマーク・リストに表示されます。

順序付けおよびインデントを明示したx#book_mark_name形式のテキスト文字列で、xにはアウトライン番号が入ります。シャープ記号 (#) とアウトライン番号は、「ブックマーク」ウィンドウには表示されませんが、順序とインデントを決めるのに使用されます。

適用対象

レイアウト・オブジェクト

デフォルト

ブランク

使用上の注意

- 「ブックマーク」が枠に対して指定されている場合、プロパティは枠の上部左角に最も近い可視オブジェクトに転送されます。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_BOOKMARKプロシージャを使用します。

ブックマークの例（インデントおよび順序付け済み）

```
1#Expense Summary Section
2#Expense Detail Section
2.1#Expenses for the Administration Department
2.2#Expenses for the Engineering Department
2.3#Expenses for the Sales Department
2.3.1#Expenses for the Eastern Sales Region
2.3.2#Expenses for the Northern Sales Region
2.3.3#Expenses for the Southern Sales Region
2.3.4#Expenses for the Western Sales Region
```

「ブックマーク」の制限

- 同じアウトライン番号が何回も使用される場合、エントリはすべて表示されますが、順序はレポートがオブジェクトを渡した順で定義されます。
- 番号がとんでいる場合、次の2つのうちの1つが発生します。同等レベルの番号がとんでいる場合は、目に見える影響はありません（たとえば、1.3.1と1.3.3で、1.3.2がない場合、同じインデント・レベルで並んで表示されます）。上位の番号と低位の番号の間がとんでいる場合、中間のレベルが必要に応じて生成されます（たとえば、1.0の後が2.1.1ならば、見かけ上の2と2.1のエントリが定義されます。そのタイトルは、後の実際のエントリと同じになります）。

アプリケーション・コマンド・ライン（PDF）

説明

「アプリケーション・コマンド・ライン（PDF）」プロパティは、PDF文書内のオブジェクトがPDFビューアでクリックされたときに、ローカル・マシン上で実行されるコマンド・ラインです。

値

ローカル・マシン上の任意の有効なコマンド・ライン（`c:\¥orawin95¥bin¥rwrwn60 userid=scott/tiger report=example.rdf`など）

適用対象

レイアウト・オブジェクト

デフォルト

ブランク

- PDF形式のレポート出力は、ハイパーリンクとアプリケーション・コマンド・ライン・リンクの両方をインクルードできます。「アプリケーション・コマンド・ライン」プロパティがオブジェクトに設定されている場合は、そのプロパティがオブジェクトに適用されます。それ以外の場合は、「ハイパーリンク」プロパティが適用されます。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_PDF ACTIONプロシージャを使用します。

「アプリケーション・コマンド・ライン (PDF)」の制限

- 「アプリケーション・コマンド・ライン」プロパティは、PDF出力専用です。

条件付き書式

説明

「条件付き書式」プロパティは、「条件書式」ダイアログ・ボックスで指定した条件がTRUEの場合、選択したレイアウト・オブジェクトの出力書式をその条件で設定します。

適用対象

レイアウト・オブジェクト

必須/オプション

オプション

表示名

説明

「表示名」プロパティは、HTMLまたはHTMLCSS形式のレポート出力に含まれるイメージ・オブジェクト上にカーソルを移動したときに、ポップアップ形式で表示されるテキストです。

値

256文字以下のテキスト文字列

適用対象

HTMLレポートのイメージ・オブジェクト

デフォルト

ブランク

使用上の注意

- レポートをHTMLまたはHTMLCSS形式で出力するとき、「表示名」プロパティ値は、HTML ALTタグを使用してIMGタグに挿入されます。
- イメージ・オブジェクトに対して定義されたハイパーリンクがある場合、「追加のハイパーリンク属性」プロパティを使用して、情報のステータス行を生成することもできます。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET DISPLAY NAMEプロシージャを使用します。

追加のハイパーリンク属性

説明

「追加のハイパーリンク属性」プロパティは、「ハイパーリンク (HTML/PDF)」プロパティによって定義されたリンクに適用される、追加のHTML属性を指定します。

値

「ハイパーリンク (HTML/PDF)」プロパティによって定義されたリンクについて有効な任意のHTMLタグ

適用対象

「ハイパーリンク (HTML/PDF)」プロパティによって定義されたリンクを持つレイアウト・オブジェクト。リンクが定義されていない場合、このプロパティは効果がありません。

デフォルト

ブランク

使用上の注意

- 「ハイパーリンク (HTML/PDF)」プロパティが枠内のオブジェクトについて指定されている場合、オブジェクトをクリックすると、デフォルトで、ハイパーリンクの宛先を示すウイン

ドゥ全体に、枠が置き換わります。枠のみを置き換えるには、"target=ファイル名"というHTMLを「追加のハイパーリンク属性」プロパティ設定に追加します。ファイル名は、置き換える枠の名前です。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET HYPERLINK ATTRプロシージャを使用します。

レイアウト・セクションの終了

説明

「レイアウト・セクションの終了」プロパティでは、ライブ・プレビューアが、整列したオブジェクトの検索をどこで終了するか決定します。このプロパティはおもにキャラクタ・モードで使用し、一致するオブジェクトの検索範囲を制限します。

値

はい	ライブ・プレビューアは、一致するオブジェクトの検索をフレームの終わりで終了します。
いいえ	ライブ・プレビューアは、同じ方向に整列しているすべてのオブジェクトを検索します。

適用対象

フレーム

必須/オプション

必須

デフォルト

オブジェクトにより異なる。

データベース列プロパティ

ブレイク順序

列のデータ型

コメント

データベース列名

データ型

ファイル形式

名前

ファイルからの読み込み

NULL時の値

幅

データベース列の制限

- データベースの列を作成できるのは、Report Builderのみです(問合せの選択リスト内の各項目について1つずつ作成されます)。データベースの列を作成するには、問合せの選択リストに項目を追加してください。すると、Report Builderがデータベースの列を作成します。
- データベースの列を削除するには、関係付けられた項目を選択リストから除去するしかありません。

フィールド・プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

コメント

HTMLタグを含む

書式マスク

フォーマット・トリガー

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アンカー・オブジェクトと連動

最小孤立行

名前

後で改ページ

前で改ページ

ページ番号付け

ページ保護

印刷オブジェクト・オン

後のプリンタ・コード

前のプリンタ・コード

ソース

ソース・データ型

垂直拡張度

可視

フィールドの制限

- フィールド内のテキストの各パラグラフのサイズは64Kバイト以下、パラグラフの数は64,000個以下にしてください。この制限は、オペレーティング・システムによって異なることがあります。
- フィールドの内容は、単語の境界（つまり、スペースおよびハイフン、タブ）で折り返されます。その他の句読点は、直前の単語の最後の文字と見なされます。フィールドの内容の中に改行文字があると改行されます。作成可能な行数についての制限はありません。
- レイアウト・モデルでは、フィールドは、ソース・グループがフィールドのソース列を含むグループ（または、そのグループの下位グループ）である繰返し枠の中に配置してください。フィールドのソースがクロス積グループによって所有される集計の場合、そのフィールドは、ソースがサマリー列の積順序内のグループである繰返し枠の集合の中に配置してください。
- フィールドは、フィールドを含む繰返し枠の前面にする必要があります。
- 数値がフィールドに収まらない場合は、値のかわりにアスタリスク（*）がフィールドに表示されます。値を表示するには、オブジェクトのサイズを変更するかまたはサイズ変更を「拡張」または「可変」に変更します。

ソース・データ型

説明

「ソース・データ型」プロパティは、フィールドのソースのデータ型を表示する読み専用フィールドです。

適用対象

フィールド

使用上の注意

- 「ソース・データ型」が「LONG」または「LONG RAW」、「RAW」の場合、「列」プロパティ・パレットの「ファイル形式」フィールドでその他の値を指定しなければ、フィールドはテキストとして処理されます。

書式マスク

説明

「書式マスク」プロパティは、フィールドにDATEおよびNUMBERの値を表示する方法を定義します。数値および日付用のカスタム・マスクを入力することも、値リストから選択することもできます。文字値には、書式マスクはありません。

最初にリストに表示される値は、フィールドのデータ型によって決まります（つまり、フィールドの「ソース・データ型」が「NUMBER」の場合、リストには数値マスクのみが表示されます）。「書式マスク」のリストにないものを入力し、「フィールド」プロパティ・パレットを確定した場合、新しい書式マスクが値リストに表示されます。

値

長さ1Kバイト以内の有効な数値または日付書式マスク。 ブランクを入力すると、Report Builderによって、そのフィールドのソースのデータ型に基づいて値がフォーマットされます。

適用対象

フィールド

必須/オプション

オプション

デフォルト

ブランク

使用上の注意

- このフィールドをブランクにしておき、このフィールドのソースの「データ型」が「DATE」の場合、使っている言語のデフォルトの書式マスクになります（たとえば、アメリカでは、デフォルトは「DD-MON-YY」）。フィールドのソースの「データ型」が「NUMBER」の場合、値はソースの幅（下の表を参照）に従ってフォーマットされます。

次の表では、「NUMBER」値がフィールドの「ソース」の幅に従ってどのようにフォーマットされるかを示すいくつかの例が挙げられています。

値	ソースの幅	出力結果
9.99999	4	10.0
9.99999	3	10.
9.99999	2	10
9.99999	1	.
-9.99999	4	-10.

-9.99999	3	-10
-9.99999	2	**
400	2	**
4.563	5	4.563
4.563	4	4.56
4.563	3	4.6
0.001	5	.001
0.001	4	.001
0.001	3	.00
0.001	2	.0
0.001	1	.
-0.01	4	-.01
-0.01	3	-.0
-.01	2	-.

デフォルトでは、SQL*ReportWriter Version 1.1と同じ動作をします。

- カスタム書式マスクは、現行のReport Builderセッション中のみ有効です。Report Builderを終了して再起動すると、「書式マスク」フィールドの値リストには、前回のReport Builderセッションで作成した書式マスクは含まれていません（これらのマスクが使用されたレポートをオープンしない場合）。リスト内に組み込んだマスクが複数セッション間で保存されるようにするには、「環境設定」ダイアログ・ボックス（「ツール」→「作業環境」）の「書式マスク」を使用します。

PL/SQL

PL/SQLを使ってこの属性を定義するには、SRW.SET FORMAT MASKプロシージャを使用します。

DATEおよびTIME書式マスク構文

次の表に、日付用の書式マスク、それに追加可能な接尾辞および時刻の書式マスクを説明します。

書式マスク	説明
SCCまたはCC	世紀。「S」によって、紀元前の場合「-」が付きます。
YYYYまたはYYYY	年。「S」によって、紀元前の場合「-」が付きます。
I、YまたはIYY	年の下3桁、下2桁または下1桁。
Y、YYまたはYYY	年の下3桁、下2桁または下1桁。
Y,YYY	カンマ付きの年。
SYEARまたはYEAR	綴りで表した年。「S」によって、紀元前の場合「-」が付きます。
RRRRまたはSRRRR	年。「S」によって、紀元前の場合「-」が付きます。

RR	年の下2桁。
BC、ADまたはB.C.、A.D.	世紀を示す記号。
Q	四半期（1月から3月が四半期「1」）。
MM	数字表記の月（1月は「01」）。
MONTHまたはMON	月の名前または月の略称（英語の場合3文字）。
WWまたはIW	年間の何週目かを示します。
W	ユリウス暦で何週目かを示します。
J	ユリウス暦日付。紀元前4712年1月1日からの日数。
DDD、DDまたはD	1年の何日目かまたは月の何日目か、週の何日目かを示します。
DAY	曜日（例:月曜日）。
DY	曜日の略称（例:月）。
AM、PMまたはA.M.、P.M.	正午標識。
HHまたはHH12	1日の時間の表記（1～12時）。
HH24	1日の時間の表記（0～23時）。
MI	分。
SSおよびSSSSS	1分間の秒および1日の秒。
FM	日付、数値またはワードの前か間にある複数のスペースを1つのスペースに置換する充填モードをトグルします。

次の接尾辞は、書式マスクに追加できます。

接尾辞	説明
TH	序数（「4th」の場合は「DDth」）
SP	綴りで表した数値（「FOUR」の場合は「DDSP」）
SPTHまたはTHSP	綴りで表した序数（「FOURTH」の場合は「DDSPTH」）

数値書式マスク構文

次の表に、数値書式マスクの作成に使用できるトークンを説明します。

フォーマット・トークン	説明
0	1桁分の数字を印刷。
N	小数点左側の先行0ゼロまたは小数点右側の後続0以外の数字の場合、1桁分の数字を印刷。
*	小数点左側の先行0以外の数字の場合、1桁分の数字が印刷され、先行0の場合はアスタリスク（*）が印刷されます。小数点右側の後続0は、印刷されません。
9	小数点左側の先行0以外の数字の場合、1桁分の数字が印刷され、先行0の場合はスペースが印刷されます。小数点右側の後続0は、印刷されます。
+	正の値の前にはプラス記号（+）が印刷され、負の値の前にはマイナス記号（-）が印刷されます。このトークンは、マスクの前にあることが必要です。

-	負の値の前にはマイナス記号(-)が印刷されますが、正の数の場合は何も印刷されません。このトークンは、その他のトークンの前にあることが必要です。
MI	負の値の後にはマイナス記号(-)、正の値の後にスペースを印刷する。このトークンは、その他のトークンより後にあることが必要です。
S	負の値にはマイナス記号(-)、正の値にはプラス記号(+)を印刷します(マスク内のSの位置に印刷)。このトークンは、その他のトークンより前または後にあることが必要です。
PR	負の値は山がた(<>)で囲み、正の値はスペースで囲んで印刷します。このトークンは、その他のトークンより後にあることが必要です。
()	負の値は丸カッコ()で囲み、正の値はスペースで囲んで印刷します。カッコでその他のトークンを囲む必要があります。
DB	正の値の後に、「DB」を印刷します。このトークンは、その他のトークンより後にあることが必要です。
CR	負の値の後に、「CR」を印刷します。このトークンは、その他のトークンより後にあることが必要です。
Y	負の数でも正の数でも、符号記号は印刷しません。
V	数値を10のN乗倍します。Nは、Vの右側にあるトークン(0または9、*、S)の数。
EEEE	科学表記法で数値を印刷します。すべての数字トークンは、仮数の数字を示します。小数点の左側には、その表示、非表示にかかわらず、数字がちょうど1桁必要です。トークンEEEEは、縦座標が正または0の場合、Eの後にプラス記号(+)を印刷します。縦座標が負の場合、Eの後にマイナス記号(-)と縦座標を表す2桁の数字を印刷(たとえば、E-99)します。
"文字列"	文字列を二重引用符で囲んで印刷します。文字列の中に二重引用符を含めたい場合、二重引用符を続けて2つ(" ")入力します。
. (ピリオド)	数値の整数部と小数部を分けるピリオド(.)を印刷します。
D	数値の整数部と分数部を分けず、ローカル的小数点文字を印刷します。
, (カンマ)	グループ・セパレータまたは3桁セパレータを示す文字としてカンマ(,)を印刷します。
G	ローカルのグループ・セパレータまたは3桁セパレータを印刷します。
\$	ドル記号(\$)を印刷します。
L	地域別通貨記号を印刷します。
C	ISO基準の通貨記号を印刷します。
%	パーセント記号(%)を印刷します。
" "	ブランクを印刷します(マスクに引用符を含めてはいけません)。
v	他のトークンに関係なく、すべての0をスペースで印刷します。
K	ブランクを印刷します。
<>	小数点揃えする領域(つまり、書式マスク中の小数点に合わせる数値の部分)の開始および終了を示します。山がたは、その数値が常に同じ大きさの領域を使用することを示します。必要に応じて、小数点左側または右側にスペースが埋め込まれます。
RN, rn	値をローマ数字(RNの場合は大文字、rnの場合小文字)で印刷します。このトークンは、他のトークンとともに使用することはできません。

書式マスクの例

次の表に、書式マスクの例をいくつか示します。

日付書式の例	表示される日付
MM/DD/RR	03/04/85
DD MON RRRR	04 MAR 1985
Mon.DD, RRRR	Mar. 4, 1985
MonthDD"日" Day AM HH:MI	11月04日 月曜日 午前 11:35

数値書式の例	数値	表示される数値
-0000	7934	"7934"
	-7934	"-7934"
-00000	7934	"07934"
-NNNN	7639	"7639"
	535	"535"
-NNN	7639	"*****"
_****	7902	"7902"
_*****	7902	"*7902"
+NNNN	100	"+100"
	-99	"-99"
(NNNN)	1600	"1600"
	-800	"(800)"
NNNNPR	1600	"1600"
	-800	"<800>"
NNNNM I	1600	"1600 "
	-800	"800-"
NNNVNN	343	"34300"
N.NNEEEE	7369	"7.37E+03"
"SRW"-0000	7782	"SRW7782"
-\$NNNN.NN	800	"\$800"
	1600	"\$1600"
-%NNN	10	"%10"
-NN NNN.NN	3000	"3 000"
+KKNNNNN.00	1950	"+ 1950.00"
	900	"+ 900.00"
\$<NNNNN.00>	1200	"\$ 1200.00"
	500	"\$ 500.00"
\$<NNNNN.00> DB	1200	"\$ 1200.00 DB"
	-500	"\$ 500.00"
\$<NNNNN.00> CR	1200	"\$ 1200.00"

	-500	"\$ 500.00 CR"
--	------	----------------

* 引用符は出力されません。この表では、先行または後続する空白の位置を明示するために使用されています。

「書式マスク」の制限

- 数値書式マスクの場合、実際の値が指定された書式マスクより長ければ、フィールドの幅に関係なく、アスタリスク(*)の文字列がレポート出力に表示されます。たとえば、フィールドの固定幅が8で、値が1234567の場合、書式マスクが<NNNNNN>ならば、*****と出力されます。

同様に、数値書式マスクによって、フィールドの幅より値が大きくなる場合は、値のかわりにアスタリスク(*)が出力されます。たとえば、フィールドの固定幅が6で、値が1234、書式マスクが-99999999の場合、*****が出力されます。これは、フォーマット・トークン9によって、小数点左側の先行0にスペースが印刷される指定になっているためです。その結果、値が長すぎるため、幅が6のフィールドには表示できません。

- 書式マスクに符号トークンを指定しない場合、正の値の前にはスペース、負の値の前にはマイナス記号(-)が表示されます。この動作は、SQL*ReportWriter Version 1.1のデフォルトの動作とは異なります。書式マスク・トークンYを指定すると、Version 1.1のデフォルトの動作になります。
- つまり、数値書式マスクはソースが数値の場合に表示され、日付書式マスクはソースが日付のときに表示されます。
- 0をスペースに置き換えて印刷する書式マスク(たとえば、9)を使用すると、1ページ当たりのバイト数が増加し、出力に時間がかかります。

可視

説明

「可視」プロパティは、フィールドのフォーマットの有無を示します。このプロパティを「いいえ」に設定すると、フィールドは非表示になります(フォーマットされません)。これは、ポイラープレート・テキストで参照するのみのフィールドに使用します。非表示フィールドは、フォーム・レター・レポートに最もよく使用されます。フォーム・レター・レポートでは、フィールドは、ポイラープレート・テキストで参照されます。非表示フィールドを参照するには、ポイラープレート・テキスト・オブジェクト内に&フィールド名を入力します。

適用対象

フィールド

必須/オプション

オプション

デフォルト

はい

使用上の注意

- 非表示フィールドを参照する場合、ボイラープレート・テキスト内ならどこでも表示できます。フィールド参照をその他のテキストの横に配置し、それらの間にスペースがない場合は、フィールド名を山がた (<>) で囲みます (たとえば、&<フィールド名>Kg)。それ以外の場合、Report Builderでは、アンパサンド (&) と次のスペースの間にあるテキストはフィールド名の一部であると見なされます。しかし、間にスペースを入れたり、山がたを使用したりしなくても、複数の参照を並べることができる (たとえば、&フィールド1&フィールド2&フィールド3) ことに注意してください。

非表示フィールドの参照例

フォーム・レター・レポートを作成しており、そのレターに受取人の名前および住所、電話番号を書くとしてます。受取人名用にNAME1、住所用にADDRESS、電話番号用にPHONEという名前のフィールドを作成します。これらのフィールドは、ボイラープレート・テキスト内に表示されるのみなので、「いいえ」を選択します。これらの値を表示するボイラープレート・テキスト内に、参照を次のように挿入します。

Dear &NAME1,

Our records show that your address and phone are the

following:

&ADDRESS

&PHONE

出力すると、次のようなレターになります。

Dear J. Smith,

Our records show that your address and phone are the

following:

234 Summer Lane

San Francisco, CA 11104

415-363-1234

非表示フィールドの制限

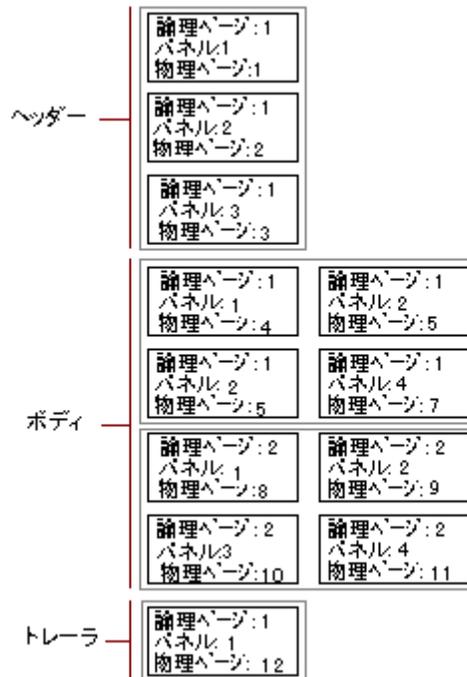
非表示フィールドの次のプロパティおよび属性は、非表示フィールドの参照に適用されます。

- 「フォントの指定」ダイアログの各属性
- 水平拡張度
- 書式マスク
- ページ番号（「ページ番号付け」ダイアログ・ボックスにあります）
- フォーマット・トリガー

ページ番号付け

説明

「ページ番号付け」プロパティによって、「物理ページ番号」または「総物理ページ数」、「パネル番号」、「総パネル数」の「ソース」を持つフィールドについてページ番号の計算方法が判別されます。



図には、論理ページおよび物理ページ、パネルの番号付けの違いが示されています。各領域には複数の論理ページが含まれる可能性があることに注意してください。さらに、各論理ページ内には、複数のパネルが含まれていることがあります。図では、物理ページは、ヘッダー・セクションのページから開始して連続した番号が付けられます。これは、「ページ番号付け」ダイアログ・ボックスで、ヘッダー、ボディおよびトレーラの各セクションを選択したと想定しています。

注意: 各ページの設定は、フィールドの「ソース」によって指定された論理ページまたは物理ページに基づいています。

ページ番号付けの例（ボディ・ページの場合）

レポートのメイン・セクションの物理ページに番号を付けるとします。

- 1 F_PAGEという名前のフィールドを作成し、「ソース」プロパティを「物理ページ番号」に設定します。
- 2 F_PAGEの「フィールド」プロパティ・パレットで、「ページ番号付け」ダイアログ・ボックスに次のように入力し、「ページ番号付け」プロパティを設定します。
 - メイン・セクションのページのみに番号を付けるので、「メイン」セクションをチェックして、ヘッダー・セクションとトレーラ・セクションはチェックしません。
 - 「起動位置」の設定に、1を入力します。
 - 「増分」の設定に、1を入力します。
 - メイン・セクションのすべてのページに連続した番号を付けるので、「リセット位置」の設定に「&Report」を入力します。
- 3 レイアウト・モデル内でF_PAGEを配置します。

ページ番号付けの例（「Nページ/全ページ」形式の場合）

レポートの各ページの最上部にページ番号を付けるとします。「Nページ/Mページ」という形式にするとします。

- 1 F_PAGEという名前のフィールドを作成し、「ソース」プロパティを「物理ページ番号」に設定します。
- 2 F_PAGEの「フィールド」プロパティ・パレットで、「ページ番号付け」ダイアログ・ボックスに次のように入力し、「ページ番号付け」プロパティを設定します。
 - メイン・セクションのページのみに番号を付けるので、「メイン」セクションをチェックして、ヘッダー・セクションとトレーラ・セクションはチェックしません。
 - 「起動位置」の設定に、1を入力します。

- 「増分」の設定に、1を入力します。
 - レポートが完了するまでページ番号が0にリセットされないように、「リセット位置」に「&Report」を入力します。
- 3 F_TOTPGEという名前のフィールドを作成し、「ソース」プロパティを「総物理ページ数」に設定します。
 - 4 次のいずれかの操作を行います。
 - これらのフィールドを非表示に（つまり、プロパティ・シートで「可視」を「いいえ」に設定）して、ボイラープレート・テキスト内でそれらを参照します。
ページ &F_PAGE of &F_TOTPGE
 - または、F_PAGEとF_TOTPGEを並べて配置します。テキスト「ページ」を含むボイラープレートでF_PAGEの左に挿入します。テキスト「/」を含むボイラープレートをF_PAGEとF_TOTPGEの間に挿入します。

ページ番号付けの例（繰返し枠のインスタンスの場合）

複数ページにまたがるインスタンスを持つ繰返し枠（R_1）があり、この各インスタンスがまたがる物理ページにページ番号を付けるとします。また、「Nページ/Mページ」という書式にするとします。

- 1 F_PAGEという名前のフィールドを作成し、「ソース」プロパティを「物理ページ番号」に設定します。
- 2 F_PAGEの「フィールド」プロパティ・パレットで、「ページ番号付け」ダイアログ・ボックスに次のように入力し、「ページ番号付け」プロパティを設定します。
 - メイン・セクションのページのみ番号を付けるので、「メイン」セクションをチェックして、ヘッダー・セクションとトレーラ・セクションはチェックしません。
 - 「起動位置」の設定に、1を入力します。
 - 「増分」の設定に、1を入力します。
 - 繰返し枠のインスタンスごとにページ番号をリセットするので、「リセット位置」に「&R_1」を入力します。
- 3 F_TOTPGEという名前のフィールドを作成し、「ソース」プロパティを「総物理ページ数」に設定します。
- 4 F_TOTPGEの「フィールド」プロパティ・パレットで、「ページ番号付け」ダイアログ・ボックスに次のように入力し、「ページ番号付け」プロパティを設定します。
 - 繰返し枠のインスタンスごとにページ番号をリセットするので、「リセット位置」に

「&R_1」を入力します。

- 5 R_1のすべての論理ページに印刷されるように、F_PAGEおよびF_TOTPGEの印刷条件を設定します。

- 「印刷オブジェクト・オン」を「全ページ」に設定します。
- 「基本印刷オン」を「インクローズ・オブジェクト」に設定します。

- 6 次のいずれかの操作を行います。

- これらのフィールドを非表示にして（つまり、プロパティ・シートで「可視」を「いいえ」に設定）、次のテキストを持つボイラープレート・テキスト・オブジェクトをR_1内に作成します。

```
Page &F_PAGE of &F_TOTPGE
```

- または、レイアウト・モデルを使用して、R_1内にF_PAGEとF_TOTPGEを並べて配置します。テキスト「ページ」を含むボイラープレートをF_PAGEの左に挿入します。テキスト「/」を含むボイラープレートをF_PAGEとF_TOTPGEの間に挿入します。

「ページ番号付け」の制限

「リセット位置」の値が繰返し枠の場合、繰返しフィールドが表示されないページには、フィールドを表示しないようにしてください。フィールドの値は繰返し枠によって決まるため、繰返し枠が表示されないページでは、結果が予測できなくなります。この状況避けるには、次のいずれかの作業を行うことができます。

- フィールドの値がリセットされる繰返し枠の内部にフィールドを配置する。これによって、フィールドは、繰返し枠が表示されるときのみ表示されます。
- フィールドの値が有効なページ（つまり、繰返し枠の1つのインスタンスが表示されるページ）にのみフィールドが表示されるように、フィールドの「フォーマット・トリガー」を作成する。

同様に、フィールドがリセットする繰返し枠の内部になく、複数のインスタンスが同じページに表示される場合、Report Builderではフィールドの値が判別されず、この場合も結果が予測できなくなります。この状況避けるには、次のいずれかの作業を行うことができます。

- フィールドの値がリセットされる繰返し枠の内部にフィールドを配置する。
- フィールドの値がリセットされる繰返し枠の「ページ当りの最大レコード数」を 1 に設定する。

ソース

説明

「ソース」プロパティは、フィールドの値の格納場所（通常は、列またはパラメータ）です。

注意: 「ソース」がページ番号値（たとえば、「パネル番号」）の場合、どのようにページ番号付けされるかを制御する「ページ番号付け」を選択できます。

値

有効な列またはパラメータの名前

現在の日付	ランタイム・パラメータ・フォーム終了後、レポートが実行されるときオペレーティング・システムの日付
ページ番号	論理ページでの番号付けによる現行のページ番号
パネル番号	論理ページ内の現行のパネル番号
物理ページ番号	物理ページでの番号付けによる現行のページ番号
総ページ数	論理ページでの番号付けによる合計ページ数
総パネル数	論理ページ内のパネルの合計数
総物理ページ数	物理ページでの番号付けによる合計ページ数

適用対象

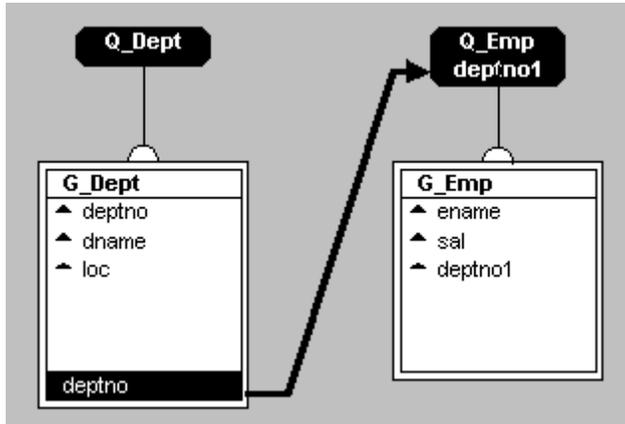
フィールド

必須/オプション

必須

ソースの例

次の図のようなデータ・モデルを持つレポートがあるとします。さらに、G_Deptは繰返し枠R_Deptの「ソース」で、G_Empは繰返し枠R_Empの「ソース」であるとします。



この例では、DEPTNOおよびDNAMEは、R_DeptまたはR_Emp内のフィールドの「ソース」にすることができます。ENAMEおよびSALは、R_Emp内のフィールドの「ソース」にしかできません。

ソースの制限

- フィールドが繰返し枠に囲まれている場合、そのソースとして使用される列は、フィールドを囲む繰返し枠のソース・グループの中またはソース・グループの上位項目の中に存在する必要があります。

図には、グループ・ツリーが示されています。繰返し枠のソース・グループが6の場合、フィールド「ソース」は、1または3、6になります。フィールドを囲む繰返し枠がない場合、列はレポート・レベルの列にしてください。つまり、レポートによって所有されている列である必要があります。
- 列に「Page」という「リセット位置」が設定されている場合、それをソースとして使用する任意のフィールドは、その列のグループをソースとして使用する繰返し枠の中にある必要があります。さらに、フィールドを参照するボイラプレート・オブジェクトも、列のグループをソースとして使用する繰返し枠によって囲まれている必要があります。
- フィールドの「ソース」を指定した後、レポートからそのソースを削除した（つまり、サマリー列を作成して、それをフィールドのソースに指定した後そのサマリー列を削除した）場合、フィールドは表示するデータにアクセスできなくなり、「ソース」フィールドにNULLが表示されます。
- ページ数」、「総パネル数」、「総物理ページ数」の場合、フィールド（フィールド自体またはボイラプレート・オブジェクトのいずれか）を表示するレイアウト・オブジェクトのサイズは、必ず固定にしてください。
- フィールドは、「ソース」に「サウンド」または「ビデオ」を指定することはできません。

式列プロパティ

ブレイク順序

列のデータ型

コメント

データ型

ファイル形式

名前

PL/SQL式

積順序

ファイルからの読み込み

ブレイク順序の設定

NULL時の値

幅

式列の制限

- プレースホルダ列または式列に設定できるデータ型は、CharacterおよびNumber、Dateです。

枠プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

コメント

フォーマット・トリガー

レイアウト・セクションの終了

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アンカー・オブジェクトと連動

名前

後で改ページ

前で改ページ

ページ保護

印刷オブジェクト・オン

後のプリンタ・コード

前のプリンタ・コード

垂直拡張度

グループ・プロパティ

子ディメンション

コメント

フィルタ・タイプ

名前

レコード数

PL/SQLフィルタ

グループの制限

- 問合せによって所有されている最下位のグループ以外は、「ブレイク順序」が指定された列を少なくとも1つ含んでいる必要があります。
- クロス積グループに関するルールは、「マトリックスの制限」を参照してください。

子ディメンション

説明

子ディメンションは、グループがネストされたマトリックス・データ・モデル内の子ディメンション・グループであるかどうかを示します。どのグループが子ディメンションであるかを識別することにより、マトリックス内の空の行または列（あるいはその両方）を排除します。

値

はい	グループがネストされたマトリックス・レポート内で子ディメンションであることを示します。
いいえ	グループがネストされたマトリックス・レポート内で子ディメンションでないことを示します。

適用対象

1つの問合せのマトリックス・データ・モデル内のクロス積・グループ

必須/オプション

オプション

デフォルト

いいえ

使用上の注意

- 子ディメンションが「いいえ」に設定されると、クロス積グループ内のグループ間の線は点線になります。子ディメンションが「はい」に設定されると、グループとその親の間の線は実線になります。

「いいえ」に設定された子ディメンションの例

次のマトリックスでは、deptno (G_DEPTNO)を含むグループは子ディメンションが「いいえ」に設定されています。値が存在しない場合にすべての行がどのように出力されているかに注意してください。

Year	JobANALYST		CLERK	MANAGER
	Deptno	Sum Sal	Sum Sal	Sum Sal
80	10			
	20		800	
	30			
81	10			2450
	20	3000		2975
	30		950	2850
82	10		1300	
	20	3000		
	30			
83	10			
	20		1100	
	30			

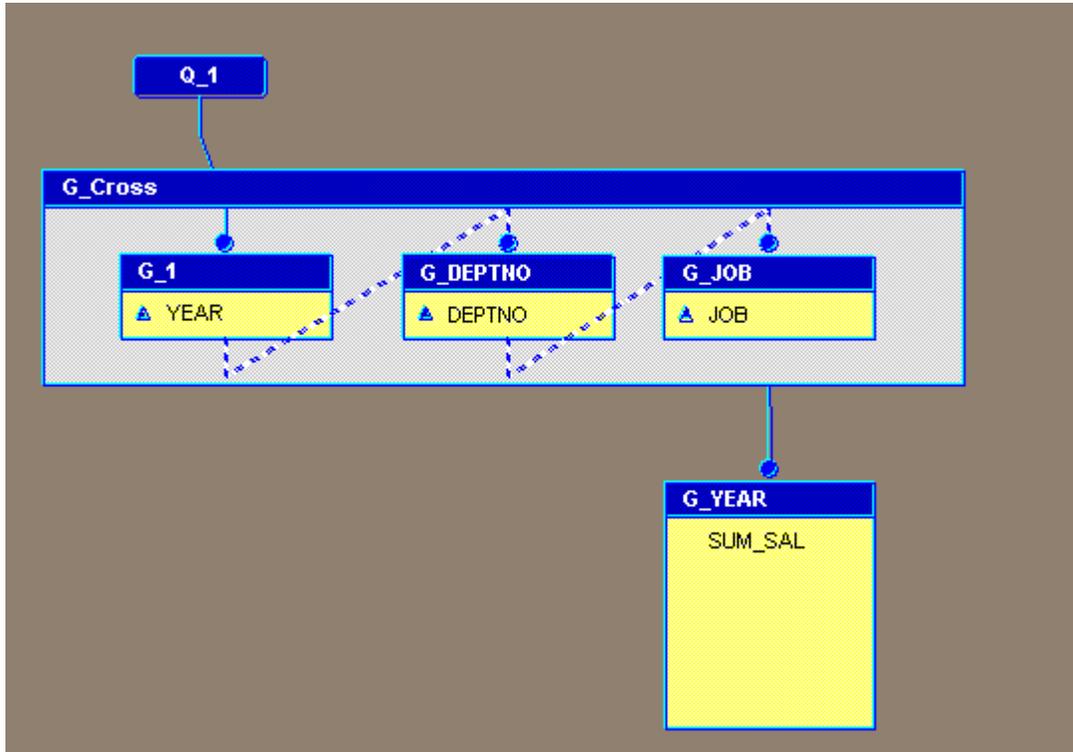
「はい」に設定された子ディメンションの例

次のマトリックスでは、deptno (G_DEPTNO)を含むグループは子ディメンションが「はい」に設定されています。値が存在する行のみ出力されていることに注意してください。

Year	JobANALYST		CLERK	MANAGER
	Deptno	Sum Sal	Sum Sal	Sum Sal
80	20		800	
81	10			2450
	20	3000		2975
	30		950	2850
82	10		1300	
	20	3000		
83	20		1100	

子ディメンションの制限

- 子ディメンションは、1つの問合せのマトリックス・データ・モデル内のクロス積・グループにのみ有効です。
- 子ディメンションは、データ・モデル内の一番上のグループには無効です。
- 次のデータ・モデル例で、子ディメンションはG_1、G_CrossおよびG_YEARには無効です。G_DEPTNOおよびG_JOBには有効です。



フィルタ・タイプ

説明

「フィルタ・タイプ」によって、Report Builderがグループにフェッチされるレコードに対して使用するフィルタ・タイプが指定されます。

値

なし	グループにフィルタを使用していないことを意味します。
最初	取り出された最初のn個のレコードがグループに含まれることを意味する。nは、「レコード数」プロパティで指定された数値です。
最後	取り出された最後のn個のレコードがグループに含まれることを意味する。nは、「レコード数」プロパティで指定された数値です。
PL/SQL	グループに組み込むレコードを判別するPL/SQLファンクションが定義される（参照される）ことを意味します。

適用対象

グループ

必須/オプション

オプション

デフォルト

なし

「フィルタ・タイプ」の例

2つの列 (ENAMEおよびSAL) を含む、EMPという名前のグループがあるとします。EMPの親グループはDEPTで、DEPTNOのみが含まれています。DEPTの「フィルタ・タイプ」に「最初」、「レコード数」に3を指定し、EMPの「フィルタ・タイプ」を「最初」、「レコード数」に5を指定します。この場合、Report Builderによって、最初の3つの部門番号が、それぞれ最初の5つの名前および給与とともに取り出されます。

フィルタ・タイプの制限

- 「フィルタ・タイプ」が「最初」または「最後」の場合、対応付けられたフィールドにNUMBER型の正の整数を指定する必要があります。数値のサイズは、最長64Kまでです。
- クロス積グループには、フィルタを使用することはできません。
- フィルタによってグループから除外されたレコードは、計算には使用されません。データを計算には使用するがレポートには出力しない場合、フォーマット・トリガーを使用して出力するデータを制限します。

名前

説明

「名前」プロパティは、Report Builderにより現行のグループの識別に使用される名前です。30バイト以内の有効なグループ名を入力します。

適用対象

グループ

必須/オプション

必須

デフォルト

G_queryname。このquerynameは、問合せの名前で、グループの中で一意の数値です。問合せがデフォルト名(たとえば、Q_1)の場合、デフォルト・グループ名は、G_nです。このnは、レポートのグループ名の中で一意の数値です。グループが問合せに関係付けられていない(つまり、クロス積グループ)場合のデフォルト・グループ名も、G_nです。

コメント

説明

「コメント」プロパティは、グループを説明するテキストを含みます。たとえば、ブレイク・グループの用途を記述しておけば、他の人が見てそのブレイク・グループで何が処理されるかがすぐにわかります。64KB以下のテキストを入力します。

適用対象

グループ

必須/オプション

オプション

デフォルト

ブランク

レコード数

説明

「レコード数」プロパティは、「フィルタ・タイプ」が「最初」または「最後」に設定されている場合に、グループに組み込むレコードの数です。たとえば、「フィルタ・タイプ」が「最初」、「レコード数」が20に設定されている場合、取り出された最初の20レコードがグループに組み込まれます。

値

長さが64K以下の、正の整数を入力します。

適用対象

グループ

必須/オプション

「フィルタ・タイプ」プロパティが「最初」または「最後」に設定されている場合、必須。

デフォルト

0

PL/SQLフィルタ**説明**

「PL/SQLフィルタ」プロパティは、PL/SQLエディタを表示するボタンです。エディタは、グループに組み込むレコードを判別する際のPL/SQLファンクションを入力できます。

適用対象

グループ

必須/オプション

「フィルタ・タイプ」プロパティが「PL/SQL」に設定されている場合は必須。

データ・リンク・プロパティ

子列

子問合せ

条件

親列

親グループ

SQL句

WHERE句のリンク例

次の2つの問合せによって、マスター/ディテール・レポートを作成するとします。

```
Q_1: SELECT DISTINCT ENAME, EMPNO
      FROM EMP
      WHERE JOB = 'SALESMAN'
```

```
Q_2: SELECT REPID, AMOUNT
      FROM SALES
      ORDER BY REPID
```

この2つの問合せを結合するには、プロパティ・パレットが次のように設定されたリンクを作成します。

SQL句	親列	条件	子列
WHERE	EMPNO	=	REPID

これによって、次のような1つの問合せを作成した場合と同様のリンクが作成されます。

```
SELECT DISTINCT ENAME, EMPNO, REPID, AMOUNT
FROM EMP, SALES
WHERE EMPNO = REPID (+) AND JOB = 'SALESMAN'
ORDER BY REPID
```

ヒント

表スタイルを使用すると、次の図のような出力になります。

```

      Ename      Empno      Repid      Amount
-----
      ALLEN      7499      7499      3000
```

	7499	810	
	7499	847	
	7499	24	
	7499	1500	
	7499	340	
	7499	240	
	7499	400	
	7499		180
	7499		280
	7499		250
MARTIN	7654	7654	16569
	7654	2301	
	7654	3306	
	7654	5600	
TURNER	7844	7844	350
	7844	485	
	7844	293	
	7844	50	
	7844	1703	

「START WITH」を使用したリンクの例

管理階級の役職順に従業員をリストしたレポートを作成するとします。次のような問合せを使用することができます。

```
Q_1: SELECT EMPNO FROM EMP
      WHERE JOB = 'MANAGER'
```

```
Q_2: SELECT ENAME, JOB, EMPNO, MGR, LEVEL
      FROM EMP
      CONNECT BY PRIOR EMPNO = MGR
```

この2つの問合せを結合するには、プロパティ・パレットが次のように設定されたリンクを作成します。

SQL句	親列	条件	子列
START WITH	EMPNO	=	EMPNO1

デフォルト・レイアウトの表スタイルを使用すると、次の図のような出力になります。

Empno	Ename	Job	Empno1	Mgr	Level
-----	-----	-----	-----	-----	-----
7566	JONES	MANAGER	7566	7839	1
	SCOTT	ANALYST	7788	7566	2
	ADAMS	CLERK	7876	7788	3
	FORD	ANALYST	7902	7566	2
	SMITH	CLERK	7369	7902	3
7698	BLAKE	MANAGER	7698	7839	1
	ALLEN	SALESMAN	7499	7698	2
	WARD	SALESMAN	7521	7698	2
	MARTIN	SALESMAN	7654	7698	2
	TURNER	SALESMAN	7844	7698	2
	JAMES	CLERK	7900	7698	2
7782	CLARK	MANAGER	7782	7839	1
	MILLER	CLERK	7934	7782	2

データ・リンクの制限

- 列を指定してリンクを作成する場合、親と子両方の列を持っており、それらの列が互換性のあるデータ型であることが必要です。たとえば、親列が文字データ型（たとえば、VARCHAR）の場合、子列も文字データ型（たとえば、CHAR）であることが必要です。

子列

説明

「子列」プロパティは、親グループ内の列（つまり、親列）に関連する子問合せで使用される列の名前です。「子列」は、データベースの列であることが必要です。サマリーまたは式にすることはできません。また、字句参照に依存することもできません（「子列」の制限を参照してください）。

適用対象

リンク

「子列」の制限

- Report Builderのリンクは、字句参照に依存してはいけません。つまり、リンクの子列または表名のどちらも、字句参照によって判別してはいけません。この機能を達成するには、列を指定しないでリンクを作成し、そのリンクに対するSQL句（WHEREなど）を問合せに直接入力する必要があります。たとえば、親および子問合せを次のように書きます。

```
Parent Query:  SELECT DEPTNO FROM EMP
Child Query:   SELECT &PARAM_1 COL_1, &PARAM2
               COL_2
               FROM EMP
               WHERE &PARAM_1 = :DEPTNO
```

親問合せで選択されたDEPTNOへのバインド参照が、WHERE句でどのように作成されるかに注意してください。また、この例では、列を指定しない状態で、データ・モデル内で問合せ間のリンクをすでに作成していることが想定されています。

子問合せ

説明

「子問合せ」プロパティは、データ・モデルでのリンク作成時に子として定義された問合せの名前です。このフィールドは、読み込み専用です。

適用対象

リンク

「子問合せ」の制限

- 「子問合せ」は、「親グループ」を1つのみ保有できます。1つの子に対して複数の親を定義することはできません。

条件

説明

「条件」プロパティは、「親列」と「子列」のリレーションを定義するSQL演算子です。SQL演算子の詳細は、『Oracle8 Server SQLリファレンス』を参照してください。

値

=	等価
<	より小さい
<=	以下
<>	等しくない
>	より大きい
>=	以上
Like	1つの列の値がその他の列のパターンと一致する場合に、条件が真であることです。パターンには、ワイルドカード文字としてパーセント記号 (%) およびアンダースコア (_) を含むことができます。
Not Like	1つの列の値がその他の列のパターンと一致しない場合に、条件が真であることです。パターンには、ワイルドカード文字としてパーセント記号 (%) およびアンダースコア (_) を含むことができます。

適用対象

リンク

必須/オプション

「親列」が指定されている場合、必須。

デフォルト

= (等価符号)

親列

説明

「親列」プロパティは、子問合せ内の列（つまり、子列）に関連する親問合せで使用される列です。「親列」には、親グループ内のデータベースの列またはサマリー列、式列を指定できます。「親列」がサマリー列または式列の場合、子問合せによって所有される列に直接的にも間接的にも依存できません。このフィールドは、読み込み専用です。

適用対象

リンク

親グループ

説明

「親グループ」プロパティは、データ・モデルでのリンク作成時に親として定義されたグループの名前です。このフィールドは、読み込み専用です。

適用対象

リンク

使用上の注意

- リンク先の列を指定しなくても、親グループから子問合せへのリンクを作成できます。これによって、2つのオブジェクト間のリレーションを作成できます。しかし、すべてのディテール・レコードが親グループの各マスター・レコードについて表示されます。

「親グループ」の制限

- 「親グループ」は、上位項目の1つとして「子問合せ」を持つことはできません。つまり、リンクは循環依存（つまり、子問合せが親グループに依存し、親グループは子問合せに依存する状況）できません。

SQL句

説明

「SQL句」プロパティは、子問合せに対して親グループを関係付ける句のタイプです。詳細は、『Oracle8 Server SQLリファレンス』を参照してください。

値

HAVING

START WITH

WHERE

適用対象

リンク

必須/オプション

「親列」が指定されている場合、必須。

デフォルト

WHERE

使用上の注意

- 「SQL句」が「START WITH」の場合、子問合せのSELECT文には、CONNECT BY句が含まれている必要があります(このプロパティに含まれているため、START WITH句は不要です)。同様に、「SQL句」が「HAVING」の場合、子問合せのSELECT文には、GROUP BY句が含まれている必要があります(このプロパティに含まれているため、HAVING句は不要です)。

マトリックス・プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

ブックマーク

コメント

クロス積グループ

水平繰返し枠

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

名前

垂直繰返し枠

クロス積グループ

説明

「クロス積グループ」プロパティは、「水平繰返し枠」および「垂直繰返し枠」のソース・グループを含むグループです。クロス積グループによって、1つのグループの中または複数のグループ間で互いに値が関係付けられて、マトリックス内のデータが作成されます。

値

有効なクロス積グループの名前を入力します。

適用対象

マトリックス

必須/オプション

必須

デフォルト

ブランク

マトリックスの例

データベースの列DEPTNOから値を取得するC_DEPTNOという列が含まれる、Group1という名前のグループがあるとします。Group2というグループには、データベースの列JOBから値を取得する列C_JOB、およびGroup1の間合せへのリンクに使用される列C_DEPTNO1が含まれています。Group3というグループには、データベースの列SALの合計であるSUMSALという列が含まれています。

	Job			
	Analyst	Clerk	Manager	
	10	\$1300	\$2450	
Dept	20	\$6000	\$1900	\$2975
	30	\$ 950	\$2850	

この例では、次のようになっています。

- 「垂直繰返し枠」は、Group1（部門番号）を含む繰返し枠です。
- 「水平繰返し枠」は、Group2（ジョブ・タイトル）を含む繰返し枠です。
- 「クロス積グループ」は、Group4（Group1およびGroup2の親であるグループ）です。

より複雑なマトリックスを作成する必要がある場合、Group1およびGroup2に列を追加すれば可能です。たとえば、Group1が部門番号のみを含むようにするのではなく、部門の位置（LOC）も含むようにすることもできます。この場合、マトリックスは、次のようになります。

	Job			
Loc Dept	Analyst	Clerk	Manager	
New York	10	\$1300	\$2450	
Dallas	20	\$6000	\$1900	\$2975
Chicago	30	\$ 950	\$2850	

マトリックスの制限

- マトリックス内では、縦方向への繰返し枠は横方向の繰返し枠の下にあることが必要です。
- マトリックス・オブジェクトは、必ず、マトリックスを形成する繰返し枠の一番上にあるこ

とが必要です（つまり、マトリックスは、水平繰返し枠および垂直繰返し枠よりも上のレイヤーに存在する必要があります）。Report Builderでは、マトリックスを水平繰返し枠および垂直繰返し枠より下に移動できません。

- マトリックスを移動すると、その2つの繰返し枠も移動します。
- マトリックス・オブジェクトを他のオブジェクトにアンカーしたり、また、他のオブジェクトをマトリックス・オブジェクトにアンカーできません（つまり、マトリックス・オブジェクトは、あるアンカーについて親オブジェクトになることも子オブジェクトになることもできません）。
- マトリックスをコピーするには、そのマトリックスおよびその2つの繰返し枠を選択してください。マトリックスのみを選択した場合は、ペースト・バッファには何もコピーされません。マトリックスと一方の繰返し枠を選択した場合は、その繰返し枠のみがペースト・バッファにコピーされます。
- マトリックス・オブジェクトのサイズを変更するには、関係付けられた繰返し枠のサイズを変更するしかありません。
- 「配置」メニューの「オブジェクト整列」、「オブジェクト・サイズ設定」をマトリックス・オブジェクトに対して使用することはできません。
- マトリックスの列および行を構成する繰返し枠のソース・グループは、同じクロス積グループから取得される必要があります。
- ソース・グループが同じ「系列」の階層に属する（つまり、相互に上位項目または下位項目である）繰返し枠の「印刷方向」は同じにしてください。マトリックス内でネストを作成する場合、クロス積グループ内の親子リレーションが使用されます。その結果、そのようなグループと関係付けられた繰返し枠は、ページ上で同じ方向に印刷されることが必要です。
- 他のオブジェクトと同様に、マトリックス・オブジェクトに対しても境界線を書くことができます。しかし、幅は必ず最小幅になります。マトリックス・レイアウト内ではオブジェクトの間が狭いため、境界線を太くすることはできません。

水平繰返し枠

説明

「水平繰返し枠」プロパティは、列のヘッダーおよびサマリーの作成、およびマトリックスのクロス積の計算に使用される値を持つ繰返し枠です。

値

有効な繰返し枠名を入力します。

適用対象

マトリックス

必須/オプション

必須

デフォルト

ブランク

「水平繰返し枠」および「垂直繰返し枠」の制限

- 「水平繰返し枠」プロパティおよび「垂直繰返し枠」プロパティに指定された繰返し枠のソース・グループは、同じクロス積グループ内にある必要があります。

垂直繰返し枠

説明

「垂直繰返し枠」プロパティは、行のヘッダーおよびサマリーの作成、およびマトリックスのクロス積の計算に使用される値を持つ繰返し枠です。

値

有効な繰返し枠名を入力します。

適用対象

マトリックス

必須/オプション

必須

デフォルト

ブランク

データベース列オブジェクト・プロパティ

コメント

列タイプ

データ型

名前

OLE2プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

コメント

フォーマット・トリガー

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アンカー・オブジェクトと連動

名前

後で改ページ

前で改ページ

ページ保護

印刷オブジェクト・オン

後のプリンタ・コード

前のプリンタ・コード

垂直拡張度

パラメータ・プロパティ

追加

追加の属性 (HTML)

コメント

データ型

第1列を隠す

初期値

入力マスク

値リスト

名前

削除

事前定義済み値に対するリストを制限

SELECT文/スタティック値

妥当性チェック・トリガー

値

幅

システム・パラメータ

BACKGROUND	レポートをフォアグラウンドとバックグラウンドのどちらで実行するかを設定します。
COPIES	レポートの印刷時に、レポートを印刷する部数を指定します。
CURRENCY	通貨インジケータの記号 (たとえば、ドル記号「\$」)。
DECIMAL	小数点の記号 (たとえば、ピリオド「.」)。
DESFORMAT	出力デバイスの形式 (たとえば、プリンタのLandscapeモード) を定義します。このパラメータは、レポートをキャラクタ・モード環境で実行する場合およびビットマップ・レポートをファイル送信する (たとえば、PDFまたはHTML出力を作成する) 場合に使用されます。

DESNAME	出力デバイスの名前(たとえば、ファイル名、プリンタ名、メール・ユーザーID)です。
DESTYPE	レポート出力の送信先となるデバイスのタイプ (screenまたはfile、mail、printer、preview)。
MODE	レポートをキャラクタ・モードとビットマップのどちらで実行するかを設定します。
ORIENTATION	レポートの印刷方向 (Landscape、Portrait、デフォルト)。
PRINTJOB	レポート実行前に「印刷ジョブ」ダイアログ・ボックスを表示するかどうかを指定します。
THOUSANDS	3桁区切り文字の記号 (たとえば、カンマ「,」)。

追加

説明

「追加」ボタンを押すと、「値」フィールドの値がスタティック値としてリストに追加されます。

適用対象

パラメータ

必須/オプション

スタティック値を追加する場合、必須。

追加の属性 (HTML)

説明

「追加の属性 (HTML)」プロパティには、JavaScriptイベント・ハンドラが含まれます。これにより、Webを介してレポートが実行されている場合、さまざまな入力のトリガーまたはイベントの選択(たとえば、無効な入力に対するエラー・メッセージの表示または入力値のフォーマット)をランタイム・パラメータ・フォームで行います。

値

有効なJavaScriptイベント・ハンドラ

適用対象

パラメータ・フォーム・フィールド

デフォルト

ブランク

コメント

説明

「コメント」プロパティは、パラメータを説明するテキストを含みます。たとえば、パラメータの用途を記述しておけば、他の人が見てそのパラメータで何が処理されるかがすぐにわかります。64KB以下のテキストを入力します。

適用対象

パラメータ

必須/オプション

オプション

デフォルト

ブランク

データ型

説明

「データ型」プロパティは、パラメータの値（たとえば、数値）の型を示します。

値

Character

Date

Number

適用対象

パラメータ

必須/オプション

必須

デフォルト

Character

データ型の制限

- パラメータに入力する値は、実行時でも必ずパラメータの「データ型」に一致する必要があります。
- 「データ型」は、パッケージ・パラメータ（たとえば、DESTYPE）では変更できません。
- 文字データ型のパラメータは、明示的にSQL文に埋め込まない限り、埋め込まれません。

第1列を隠す

説明

「第1列を隠す」プロパティは、パラメータの値リストに選択リストの第1列を表示するかどうかを示すチェックボックスです。この設定は、ユーザーが知らない値または知られたくない値を隠す場合に便利です。

適用対象

パラメータ

デフォルト

チェックあり

第1列を隠す場合の例

実行時にユーザーが部門を指定して、従業員をリストするレポートを作成するとします。この場合に使用する表には、プライマリ・キーまたは外部キーとしてDEPTNOがあります。このレポートのユーザーは、部門番号を知りません。ユーザーが知っているのは、部門名のみです。レポートの問合せは、次のようになります。

```
SELECT * FROM EMP
WHERE DEPTNO = :MYDEPT
```

「パラメータ」プロパティ・シートの「データ/選択」タブでは、「SELECT文」を選択して値リストに次のSQL文を入力します。

```
SELECT DEPTNO, DNAME FROM DEPT
```

その後、「第1列を隠す」をチェックします。これによって、DEPTNOは、レポート実行時に値リストに表示されません。DNAMEのみが表示されます。ユーザーがリストからDNAMEの値を選択すると、それに対応するDEPTNOの値がパラメータMYDEPTに使用されます。

初期値

説明

「初期値」プロパティには、パラメータのデフォルト値を指定します。コマンド・ラインまたはランタイム・パラメータ・フォームで上書きしなければ、この値が使用されます。そのパラメータの「データ型」に合う値を入力します。「初期値」の最大長は、1Kバイトです。「初期値」をブランクのままにした場合、そのパラメータのデフォルト値はありません。

適用対象

パラメータ

必須/オプション

オプション

デフォルト

DESTYPEの場合、「Preview」。DESFORMATの場合、「dflt」。COPIESの場合、「1」。その他のパラメータの場合、ブランク。

初期値の制限

- 「初期値」は、パラメータの「データ型」および「幅」によって妥当性チェックが行われます。また、次のように、「初期値」は書式マスクに対しても妥当性チェックが行われます。
 - あるデータに初期値を指定すると、「書式マスク」に設定されたマスクが使用されます。
 - 「書式マスク」がブランクで値が日付である場合、Report Builderの内部マスク(たとえば、DD-MON-YY)が使用されます(数字や文字の値については、「書式マスク」を指定しない限り妥当性チェックは行われません)。
- 値を指定せず、パラメータがSQLまたはPL/SQLで参照される場合、SQLまたはPL/SQLの解析にはNULLが使用されます。NULLによって、SQLまたはPL/SQL構成体が正しく解析されない場合、エラーが発生します。

入力マスク

説明

「入力マスク」プロパティは、コマンド・ラインまたはランタイム・パラメータ・フォームで入力されたパラメータの値についてReport Builderが妥当性チェックに使用する書式マスクです。パラメータの「初期値」の妥当性チェックにも使用されます。128バイト以下の有効な書式マスクを入力するか、リストから値を選択します。

適用対象

パラメータ

必須/オプション

オプション

デフォルト

ブランク。つまり、「データ型」が「Date」の場合、「入力マスク」は、ご使用の言語のデフォルト値（たとえば、アメリカならばデフォルトは「DD-MON-YY」）になり、「データ型」が「Number」の場合、「入力マスク」は-9999になります。

入力マスクの制限

- 最初に「入力マスク」の値リストに表示される値は、パラメータのデータ型によって決まります（たとえば、パラメータの「データ型」が「Number」の場合、リストには数値マスクのみが表示されます）。新しい書式マスクを入力してプロパティ・シートを確定した場合、新しい値がリストに表示されます。しかし、これらの書式マスクが有効なのは、現行のReport Builderセッション中のみです。Report Builderを終了して再起動すると、値リストには、前回のReport Builderセッションで作成した書式マスクは含まれていません（これらのマスクが使用されたレポートをオープンしない場合）。リストに追加した書式マスクが複数セッション間で保存されるようにするには、「作業環境」ダイアログの「マスク編集」ボタンを使用します。
- 「入力マスク」プロパティは、「データ型」が「Character」に設定されたパラメータには適用されません。

削除

説明

「削除」ボタンは、スタティック値リストから選択された値を削除します。

適用対象

パラメータ

必須/オプション

スタティック値を削除する場合、必須。

事前定義済み値に対するリストを制限

説明

「事前定義済み値に対するリストを制限」は、パラメータの値を値リストに含むかどうかを示すチェックボックスです。たとえば、「事前定義済み値に対するリストを制限」をチェックして、「初期値」が値リストの値に一致しない場合、「パラメータ」プロパティ・シートの確定時にエラーが発生します。同様に、コマンド・ラインまたはコマンド・スクリプトに指定したパラメータが値リストの値に一致しない場合、レポート実行時にエラーが発生します。「事前定義済み値に対するリストを制限」をチェックした場合、値リストがランタイム・パラメータ・フォームに表示されます。このパラメータをチェックしない場合は、コンボ・ボックスが表示されます（つまり、リストから選択することも、値を入力することもできます）。

適用対象

パラメータ

デフォルト

チェックあり

SELECT文/スタティック値

説明

「SELECT文/スタティック値」プロパティは、パラメータの値リストへの挿入に、SELECT (SQL 問合せ) 文とスタティック・リストのどちらを使用するかを示すラジオ・ボタンです。

値

SELECT文	パラメータの値リストに挿入するためにSELECT文を入力することを示します。このオプションを選択した場合、「SELECT文」ラジオ・ボタンの下の領域にSELECT文を入力する必要があります。
スタティック値	パラメータの値リストに挿入するために自分で値を入力することを示します。このオプションを選択した場合、自分で値を入力して値リストを作成してください。この場合、「値」フィールドに値を入力して、「追加」ボタンを選択します。リストに載せる値すべてについてこの処理を行います。値を削除するには、削除する値をリストから選択して「削除」ボタンを選択します。

適用対象

パラメータ

必須/オプション

必須

有効化トリガー

説明

「有効化トリガー」プロパティは、妥当性チェックをするPL/SQLの入力および（必要があれば）パラメータの修正が可能なPL/SQLエディタを表示するボタンです。

適用対象

パラメータ

値

説明

「値」プロパティは、「追加」ボタンを選択したときに、パラメータ値のスタティック・リストに追加される値です。パラメータに有効な値を入力して「追加」ボタンを選択し、スタティック値リストにその値を追加します。

適用対象

パラメータ

必須/オプション

スタティック・リストに値を追加する場合、必須。

幅

説明

「幅」プロパティは、コマンド・ラインまたはランタイム・パラメータ・フォームに入力するパラメータ値の最大文字数です。1～64000の数を入力します。

適用対象

パラメータ

必須/オプション

必須

デフォルト

40文字

幅の制限

- 実行時に入力されたパラメータ値が「幅」より長い場合、Report Builderによって値が切り捨てられます。
- 「幅」には、パッケージ・パラメータ（たとえば、DESNAME）を指定できません。

値リスト

追加

第1列を隠す

削除

事前定義済み値に対するリストを制限

SELECT文/スタティック値

値

パラメータ・フォーム・ボイラープレート・プロパティ

コメント

名前

型

名前

説明

「名前」プロパティは、Report Builderが現行のボイラープレートの識別に使用する名前です。

値

有効なパラメータ・フォーム・ボイラープレート名を入力してください。

適用対象

パラメータ・フォーム・ボイラープレート

必須/オプション

必須

デフォルト

PB_*n*。この*n*は、レポートのパラメータ・フォーム・ボイラープレート名の中で一意の数値です。

型

説明

「型」プロパティには、ボイラープレート・オブジェクトの書式（たとえば、テキスト）が表示されます。このフィールドは、読み込み専用です。

パラメータ・フォーム・フィールド・プロパティ

コメント

データ型

名前

ソース

データ型

説明

「データ型」プロパティは、パラメータ・フォーム・フィールドの「ソース」のデータ型を表示するフィールドです。

名前

説明

「名前」プロパティは、Report Builderが現行のフィールドの識別に使用する名前です。

値

有効なパラメータ・フォーム・フィールド名を入力します。

適用対象

パラメータ・フォーム・フィールド

必須/オプション

必須

デフォルト

PF_*n*。この*n*は、レポートのパラメータ・フォーム・フィールド名の中で一意の数値です。

ソース

説明

「ソース」プロパティは、パラメータ・フォーム・フィールドの値を持つパラメータです。この値リストには、レポート内のすべてのパラメータが含まれています。

値

有効なパラメータ名を入力します。

適用対象

パラメータ・フォーム・フィールド

必須/オプション

必須

デフォルト

ブランク

ソースの制限

- パラメータ・フォーム・フィールドの「ソース」は、データ・モデル・パラメータであることが必要です。
- 1つの「ソース」に対して複数のパラメータ・フォーム・フィールドを指定できません。

プレースホルダ列プロパティ

ブレイク順序

列タイプ

コメント

データ型

ファイル形式

名前

PL/SQL式

積順序

ファイルからの読み込み

ブレイク順序の設定

NULL時の値

幅

特殊な集計計算のプレースホルダの例

各部門の最も給与の高い従業員をリストしたマスター/ディテール・レポートを作成するとします。たとえば、次のような出力結果が得られます。

```
Dname ACCOUNTING          Loc NEW YORK      Highest Paid
```

```
Employee(s):
```

```
  Ename      Job          Sal
```

```
-----
```

```
CLARK      MANAGER      2450.00      KING
```

```
KING       PRESIDENT    5000.00
```

```
MILLER     CLERK        1300.00
```

```
Dname RESEARCH           Loc DALLAS       Highest Paid
```

```
Employee(s):
```

```

Ename      Job          Sal
-----
SMITH      CLERK        800.00      SCOTT and FORD
JONES      MANAGER      2975.00
SCOTT      ANALYST      3000.00
ADAMS      CLERK        1100.00
FORD       ANALYST      3000.00

```

給与の最も高い従業員の名前を取得するには、データ型が「Character」に設定された HIGHEST_PAID_EMP というブレースホルダ列を作成します。HIGHEST_PAID_EMP の値は、データ型が「Character」に設定された SET_HIGHEST_PAID_EMP という式列によって設定されます。R_COUNT は、「カウント」ファンクションによるサマリー列です。SET_HIGHEST_PAID_EMP は、式に次の PL/SQL 関数を含んでいます。

```

function set_plch_high_emp return character is
begin
  /* Check if we are at first record in child, if so,
     then reset placeholder */
  if (:r_count <= 1) then
    :highest_paid_emp := ' ';
  end if;
  /* Does this child record compare to the max sal
     for the dept ? If yes, and placeholder already
     set (i.e. we have more than one max sal) set the
     placeholder to the concatenated string. Else
     set the placeholder to the employee's name */
  if (:sal = :max_dept_sal) then
    if (:highest_paid_emp <> ' ') then
      :highest_paid_emp := :highest_paid_emp || ' and '
        || :ename;
    else
      :highest_paid_emp := :ename;
    end if;
  end if;
  return (:highest_paid_emp);
end;

```

ブレースホルダ列の制限

- ブレースホルダ列または式列に設定できるデータ型は、Character、Number および Date です。
- ブレースホルダ列の値を割り当てることができるのは、1つのグループのみです。複数のグループにブレースホルダ列の値を割り当ててはできません。

PL/SQL式

説明

「PL/SQL式」プロパティは、式の入力が可能なPL/SQLエディタを表示するボタンです。

適用対象

式列およびプレースホルダ列

列ソート・プロパティ

ブレイク順序

列のデータ型

コメント

データベース列名

データ型

ファイル形式

名前

ファイルからの読み込み

NULL時の値

幅

「列ソート」の制限

- 列ソートは、Expressへの問合せにのみ使用できます。
- Expressのディメンションのみ対応するソート列を持ち、単位は指定できません。
- 「列ソート」プロパティの編集は、データ・モデル表示またはオブジェクト・ナビゲータで行います。
- ディメンション値をデータ・モデルの別のグループに移動する場合は、対応するソート列もいっしょに移動する必要があります。
- ソート列は、データ・ウィザードおよびレポート・ウィザード、レイアウト・モデル表示およびライブ・プレビューアには表示されません。
- Report Builderは、Expressへの問合せ内のディメンションごとにソート列を生成します。
- セレクタのソート・ツールを使って、ディメンションの順序を変更します。

問合せプロパティ

コメント

外部問合せソース・ファイル

最大取出し行

名前

SQL問合せ文

型

コメント

説明

「コメント」プロパティには、問合せを説明するテキストが含まれます。たとえば、問合せの用途を記述しておけば、他の人が見てその問合せで何が処理されるかがすぐにわかります。64KB以下のテキストを入力します。

適用

問合せ

必須/オプション

オプション

デフォルト

ブランク

使用上の注意

- Oracleの標準SQLコメント・デリミタ（たとえば、`/* comment */`）を使用して、SQLのSELECT文に直接コメントを挿入することもできます。詳細は、『Oracle8 Server SQLリファレンス』を参照してください。

外部問合せソース・ファイル

説明

「外部問合せソース・ファイル」プロパティは、データベース、または現行の問合せのSELECT文として使用する必要のあるSELECT文を持つファイルに格納された問合せの名前です。1KB以内の有効な外部問合せソース・ファイル名を入力します。参照する外部問合せソース・ファイルの長さは、32KB以内である必要があります。32KBを超えると、ファイルは切り捨てられ、警告が出されます。その問合せがファイルに格納されている場合は、問合せ名の前にパスをつけることができます。ファイル名の前にパスが指定されていない場合は、Report Builderのファイル・パス検索順序に従って、ファイルが検索されます。問合せがデータベースに格納されている場合、格納場所を示すために問合せの前に「DB:」を付けます。たとえば、次のように指定します。

```
DB:myquery
```

問合せ名に接頭辞「DB:」を付けると、Report Builderによって直接データベースがアクセスされ、問合せが検索されます。データベースに接続されていない場合、問合せを見つけられず、その結果を知らせるメッセージが発行されます。

注意: レポートの移植性を確保するには、REPORTS60_PATHという環境変数を使用します。この変数を使用して、レポート内で使用される外部ファイル（外部問合せ、ポイラープレート、PL/SQLなど）を検索するためのデフォルト・ディレクトリを指定できます。これによって、レポートにディレクトリ・パスをハードコードする必要がなくなります。

適用

問合せ

必須/オプション

オプション

デフォルト

ブランク

使用上の注意

- このフィールドから参照される外部問合せソース・ファイルのSELECT文についての妥当性チェックは、この問合せのプロパティ・シートの確定時に行われます。
- 外部問合せソース・ファイルを更新する場合、更新によってレポート定義が無効になることがないようにする必要があります。たとえば、レポートにDEPTNO用の列があり、外部問合せソース・ファイルの選択リストからDEPTNOを削除する場合、レポートを実行できません。
- 外部問合せソース・ファイル名を指定した場合、外部問合せのSELECT文が「SQL問合せ文」

フィールドに読み専用として表示されます。SELECT文を変更する場合は、外部問合せソース・ファイルを開いてください。

最大取出し行

説明

「最大取出し行」プロパティは、問合せで取り出すことができるデータの最大行数です。Report Builderによって、このプロパティで指定した行数のみが問合せで取り出されます。レポートを設計している場合、どのようなレイアウトになるかを見るために何度もReport Builderを実行する必要があります。レポートで多数のデータを取り出す場合、この作業にはかなりの時間がかかってしまいます。このプロパティの指定により、取り出されてフォーマットされるデータ量を減らせば、パフォーマンスが改善されます。

値

取り出される行数を、1～32,000の正の数で指定。空白のままにすると、すべての行が取り出されます。

すべての問合せグループではなく、一部の問合せグループによって行が読み込まれるように制限する場合は、「グループ」プロパティ・シートの「フィルタ」を使用します。

適用

問合せ

必須/オプション

オプション

デフォルト

空白

使用上の注意

- 最大行数を超えたために取り出されなかった行は、Report Builder固有の計算では使用されません。たとえば、子問合せで「最大取出し行」を5に設定した場合、その親グループに含まれる子のサマリーには、その問合せによって取り出される5レコードのみが使用されます。データを計算には使用するがレポートには出力しない場合、「フォーマット・トリガー」を使用して出力するデータを制限します。

PL/SQL

PL/SQLで出力するデータを制限するには、SRW.SET MAXROWプロシージャを使用します。

名前

説明

「名前」プロパティは、Report Builderによって現行の問合せの参照に使用される名前です。30バイト以内の有効な名前を入力します。

適用

問合せ

必須/オプション

必須

デフォルト

Q_n。このnは、レポートの問合せ名の中で一意の数値です。

SQL問合せ文

説明

「SQL問合せ文」プロパティは、データベースからレポートにデータを取り出すSQLのSELECT文です。64K以内の有効なSELECT文を入力します。この上限は、オペレーティング・システムによって異なります。SELECT文の機能は、INTO句およびFOR UPDATE句を除いて、すべてサポートされています。SELECT文の作成には、次の作業を行います。

- 「表/列」ダイアログ・ボックスを使用
- コメントを挿入
- バインド参照および字句参照を挿入
- SQLエラーのチェック

適用

問合せ

必須/オプション

必須

SQL問合せ文の例

次にSELECT文の例を示します。

```
SELECT O.CUSTID, P.DESCRIP, I.ITEMTOT, O.ORDERDATE,
       I.ORDID
FROM ORD O, PRODUCT P, ITEM I
WHERE O.ORDID = I.ORDID AND
      I.PRODID = P.PRODID
ORDER BY O.CUSTID, P.PRODID, O.ORDERDATE
/* For each customer, show the products they bought */
/* (DESCRIP), the amount they spent on that product */
/* (ITEMTOT), the date they ordered the product */
/* (ORDERDATE), and the order id (ITEM.ORDID) of the */
/* order in which they bought the product. */
```

バインド参照および字句参照を含むSQL問合せ文の例

次に、バインド参照および字句参照を使用するSELECT文の例を示します。

```
SELECT CUSTID, SUM(TOTAL) TOTAL
FROM &FROMGROUP
HAVING SUM(TOTAL) > :MINTOTAL
```

この場合:

&FROMGROUP	ORD GROUP BYCUSTID (または、実行時に入力するその他の値)です。この場合、指定された「初期値」を持つFROMGROUPという名前のパラメータを定義することが必要なことに注意してください。SELECT文の解析時に値&FROMGROUPがNULLの場合、この文ではFROMの後に表名が指定されていないため、エラーになります。
:MINTOTAL	注文の合計が最小になるよう顧客を選択するために使用される別の問合せからの列です。

字句参照を含むSQL問合せ文の例

次に、パラメータに字句参照を使用するSELECT文の例を示します。

```
SELECT ENAME, EMPNO
FROM EMP
WHERE ENAME LIKE UPPER (&NAME)
```

この場合:

&NAME デフォルト値を持つパラメータを参照します。このパラメータの値は実行時にも指定できることに注意してください。たとえば、コマンド・ラインから、次のように入力できます。

```
RWRUN60 REP SCOTT/TIGER NAME='SM%'
```

SQL問合せ文の制限

注意: レポート作成後にデータベースの列の幅が増えた場合、その変更を反映するためには、データ・モデルとレイアウト・モデルの両方を更新する必要があります。更新しない場合は、切捨てが行われる場合があります。問合せの再解析（たとえば、「SQL問合せ文」にスペースを入力して「OK」をクリック）時に、データベースの列は自動的に更新されます。ソース列の幅が大きくなった集計、式およびブレースホルダについては、手動でそれぞれの幅を更新する必要があります。同様に、レイアウト内のフィールドのサイズを変更し、広くなったソース列の幅に合わせてサイズ設定を変更する必要があります。

- SELECT文は、すべてのSQL構文ルールに従っている必要があります。詳細は、『Oracle8 Server SQLリファレンス』を参照してください。「SQL問合せ文」の終了時およびレポートの生成または実行時に、Report Builderによって構文がチェックされます。
- セミコロン、スラッシュまたはその他の文字をSELECT文の終わりに使用することはできません。
- SELECT文を有効にするためにSELECT文で参照される表およびビューに関する適切な権限を保持している必要があります。
- 「SQL句」のSTART WITHとのリンクにおいて子になっている問合せの「SQL問合せ文」には、CONNECT BY句が含まれている必要があります（データ・モデル・リンクに含まれているため、START WITH句は不要です）。同様に、「SQL句」のHAVINGとのリンクにおいて、子になっている問合せの「SQL問合せ文」には、GROUP BY句が含まれている必要があります（データ・モデル・リンクに含まれているため、HAVING句は不要です）。
- SELECT式が問合せに追加されたり、問合せ内で改名されたり、問合せから削除された場合、それに対応する列はReport Builderによって自動的に作成または削除されます。「レイアウト・モデル」内の影響を受けたオブジェクトは手動で更新するか、またはそのレポート用に新規のデフォルト・レイアウトを作成する必要があります。Report Builderによってレイアウト・モデルが自動更新されない理由は、デフォルト・レイアウトに対して手動で行った更新が、誤って失われる可能性があるためです。
- 問合せによって選択された列のデータ型または名前をデータベース内で変更する場合、変更された列を「ソース」として使用するReport Builderの列およびフィールドに必要な変更は、すべて手動で行なってください。
- SELECT文内の列の順序によって最初の列の順序が決まります。SELECT文内の列の順序を後で変更しても、列の順序は変更されません。SELECT文に新規の列を追加した場合、その列は、問合せに属する最下位の子グループの終わりに付加されます。列の順序は、デフォルト・レイアウト内の関係付けられたフィールドの順序決定に使用されます。

型

説明

「型」プロパティは、問合せの型を記述する読み専用のフィールドです。

適用

問合せ

データベース参照列プロパティ

ブレイク順序

コメント

列のデータ型

データ型

名前

幅

繰返し枠プロパティ

追加のハイパーリンク属性

アプリケーション・コマンド・ライン (PDF)

基本印刷オン

ブックマーク

列モード

コメント

フォーマット・トリガー

枠間の水平間隔

水平拡張度

ハイパーリンク (HTML/PDF)

ハイパーリンクの宛先

アンカー・オブジェクトと連動

ページ当りの最大レコード数

最小孤立レコード数

名前

後で改ページ

前で改ページ

ページ保護

印刷オブジェクト・オン

後のプリンタ・コード

前のプリンタ・コード

印刷方向

ソース

枠間の垂直間隔

垂直拡張度

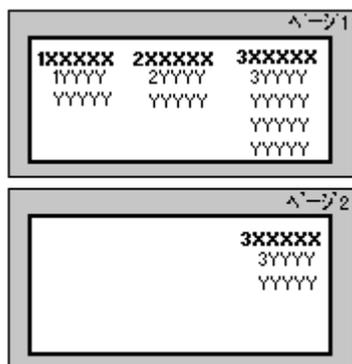
繰返し枠の制限

- 繰返し枠は、そのソース・グループがマトリックスでない場合は「レイアウト・モデル」の任意の場所に配置できます。
- 繰返し枠は、それを含むオブジェクト（たとえば、フィールドまたはボイラープレート）の背面にある必要があります。

列モード

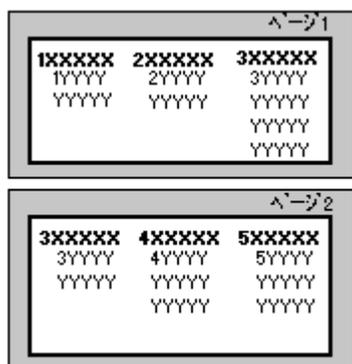
説明

「列モード」プロパティは、Report Builderによって、繰返し枠のインスタンス用のデータがどのようにフェッチおよびフォーマットされるかをコントロールします。「列モード」を「はい」に設定すると、前のインスタンスが完了する前に、繰返し枠の次インスタンスのフォーマットを開始できます。



この図では、「列モード」が設定されていない場合のインスタンスの位置が示されています。この場合、インスタンスがオーバーフローすると、ページ間の整列は行なわれません。インスタンス3は1ページ目右側でフォーマットされ、オーバーフローは2ページ目の左側でフォーマットされます。

「列モード」を「いいえ」に設定すると、前のインスタンスが完了する前に、繰返し枠の次インスタンスのフォーマットを開始できません。



この図では、「列モード」が繰返し枠のインスタンスのフォーマットにどのような影響を与えるかが示されています。インスタンス3のフォーマットは、1ページ目で開始し、2ページ目で終了します。さらに、インスタンスが別のページにもオーバーフローした場合はページ間で整列されます。この図では、インスタンス3は、2ページ目でも1ページ目に表示された位置と同じ相対位置に表示されます。

「列モード」が使用されるのは、主に、マスター繰返し枠が、垂直または水平方向に拡張（つまり、サイズ変更が「可変」または「拡張」）される可能性があるフィールドを含む繰返し枠です。

適用対象

繰返し枠

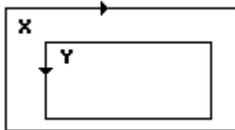
必須/オプション

オプション

デフォルト

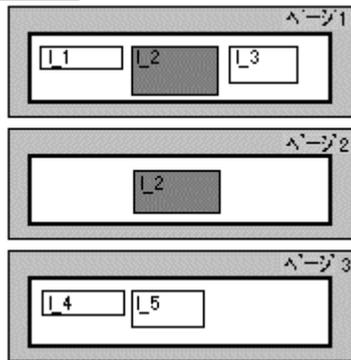
いいえ

列モードの例

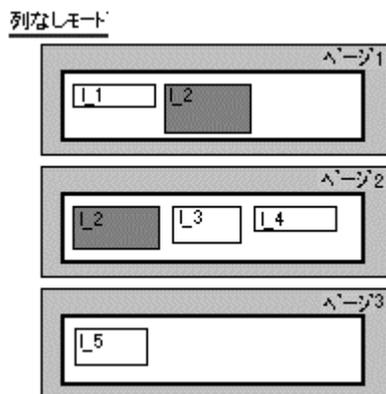


この図では、「列モード」を使用した場合のレイアウト例が示されています。（Xの「水平拡張度」は「固定」であると想定されています。）

列なしモード



この図では、繰り返し枠Xの「列モード」が「はい」に設定された場合のインスタンス1~3が示されています。インスタンス2が論理ページの2ページ目で完了したとしても、インスタンスの位置は、論理ページの1ページ目と同じです。論理ページ幅によって、1ページ目でフォーマットを開始できるインスタンスが制限されることに注意してください。同様に、3ページ目でインスタンス4および5がフォーマットされます。



この図では、繰返し枠Xの「列モード」が「はい」に設定されていない場合のインスタンス1~5の位置が示されています。この例では、インスタンス2は、論理ページの2ページ目では異なる位置で完了します。また、インスタンス3は、論理ページの1ページ目で完了しなかったため、論理ページの2ページ目では、1ページ目とは異なる位置にあります。

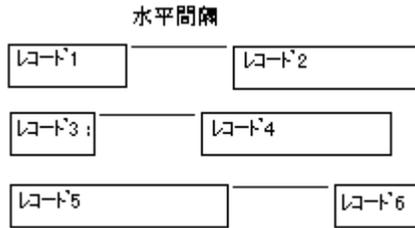
列モードの制限

- 次のいずれかを持つ繰返し枠についてのみ「列モード」を指定できます。
 - 「印刷方向」が「横」および「水平拡張度」が「固定」
 - 「印刷方向」が「縦」および「垂直拡張度」が「固定」
- 「印刷方向」が「右/下」または「下/右」に設定されている繰返し枠には、列モードを指定できません。

枠間の水平間隔

説明

「枠間の水平間隔」プロパティは、繰返し枠のインスタンスの水平間隔の空白の量（そのレポートのサイズの単位による）です。0以上の数値を入力します。この最大値は、サイズの単位によって異なります。単位がインチの場合、最大値は512インチです。センチメートルでは、1.312センチメートルです。パイカの場合、36.864パイカです。



適用対象

繰返し枠

必須/オプション

必須

デフォルト

0

枠間の水平間隔の制限

- 「印刷方向」が「縦」の場合、この設定には効果がありません。
- 「印刷方向」が「下/右」の場合、横並びに表示される2つのインスタンス間の空白に対してこの設定は有効です。

ページ当りの最大レコード数

説明

「ページ当りの最大レコード数」プロパティは、1論理ページ上でフォーマットされる繰返し枠のインスタンス数の最大値です。多数のインスタンスを持つ繰返し枠があるとします。レポートを見やすくするために、繰返し枠のインスタンスが1論理ページに最大3つまで含まれるようにするとします。1論理ページに含まれるインスタンスが3つを超えないようにするには、「ページ当りの最大レコード数」を3に設定します。

値

1～32Kまでの整数	1論理ページ上にフォーマットできるインスタンスの最大数です。
ブランク	1論理ページ上にできる限り多数のインスタンスをフォーマットします。

適用対象

繰返し枠

必須/オプション

オプション

デフォルト

ブランク

ページ当りの最大レコード数の例

1論理ページに1つの繰返し枠のインスタンスを1つのみ含むようにするとします。まず、1つのインスタンスのみが各論理ページに表示されるように、繰返し枠の「ページ当りの最大レコード数」を1に設定します。その後、最初のインスタンスが新しい論理ページから表示されるように「前で改ページ」および「後で改ページ」を指定します。

最小孤立レコード数

説明

「最小孤立レコード数」プロパティは、繰返し枠の印刷を開始する論理ページに表示されるインスタンスの最小数です。このプロパティに設定された数のインスタンスが、最初に印刷が設定されていた論理ページに収まらない場合は、繰返し枠のフォーマットは次のページで開始します。このプロパティを1に設定した場合は、最小値はなく、繰返し枠は、1つのインスタンスが収まる論理ページであればどのページでもフォーマットを開始します。

レポートを見やすくするために、繰返し枠の最初のインスタンスが論理ページの一番下に印刷されないようにするとします。その論理ページに少なくとも2つのインスタンスがあるようにするには、「最小孤立レコード数」を2に設定します。

値

1～32Kまでの整数	1論理ページ上にフォーマットできるインスタンス数の最小値です。1を設定した場合、最小値はありません。
ブランク	最小値はありません。

適用対象

繰返し枠

必須/オプション

必須

デフォルト

ブランク

最小孤立レコード数の例

1論理ページに繰返し枠のすべてのインスタンスが含まれるようにするとします。この場合、「最小孤立レコード数」には、かなり大きな値を指定します。繰返し枠が印刷されるように設定されている最初の論理ページにすべてのインスタンスが収まらない場合、そのページにはどのインスタンスもフォーマットされません。そのかわり、次の論理ページにすべてフォーマットされます。しかし、次のページおよび後続のページでは、「最小孤立レコード数」は無視されます。空白のページで印刷を開始しているため、すべてのインスタンスがその論理ページに収まらない場合はどの論理ページにも収まりません。

最小孤立レコード数の制限

- 「最小孤立レコード数」は、オブジェクトが印刷されるように設定されている最初の論理ページのみ適用されます。「最小孤立レコード数」は後続のページでは、無視されます。
- 「最小孤立レコード数」が繰返し枠のインスタンスの合計より大きい場合、繰返し枠がフォーマットされるように設定されている最初の論理ページにすべてのインスタンスが収まっている必要があります。繰返し枠がフォーマットされるように設定されている最初の論理ページにすべてのインスタンスが収まらない場合、繰返し枠のフォーマットはその次のページで開始します。

印刷方向

説明

「印刷方向」プロパティは、繰返し枠の後続のインスタンスが表示される方向です。

値

横	繰返し枠の2番目以降の各インスタンスは、その論理ページの前インスタンスの右に印刷されます。
右/下	繰返し枠の2番目以降の各インスタンスは、その論理ページの前インスタンスより右のマージンに1つのインスタンス全体が入りきらなくなるまで、前のインスタンスの右に印刷されます。その時点でその論理ページの垂直方向に印刷できる空白が残っていれば、Report Builderによって論理ページの左端のインスタンスの下にインスタンスが印刷されます。
縦	繰返し枠の2番目以降の各インスタンスは、その論理ページの前インスタンスより下に印刷されます。

下/右	繰返し枠の2番目以降の各インスタンスは、その論理ページの前のインスタンスより下のマージンに1つのインスタンス全体が入りきらなくなるまで、前のインスタンスの下に印刷されます。その時点でその論理ページの水平方向に印刷できる空白が残っていれば、Report Builderによって論理ページの上端のインスタンスの右にインスタンスが印刷されます。
-----	---

適用対象

繰返し枠

必須/オプション

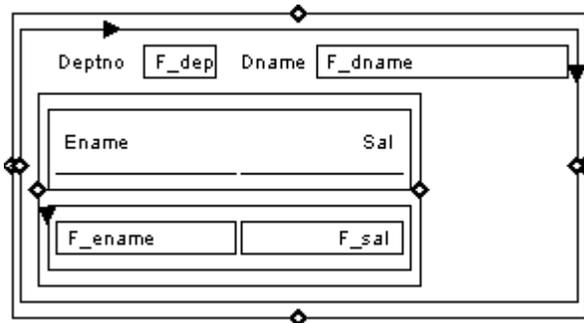
必須

デフォルト

縦

印刷方向の例（「右/下」の場合）

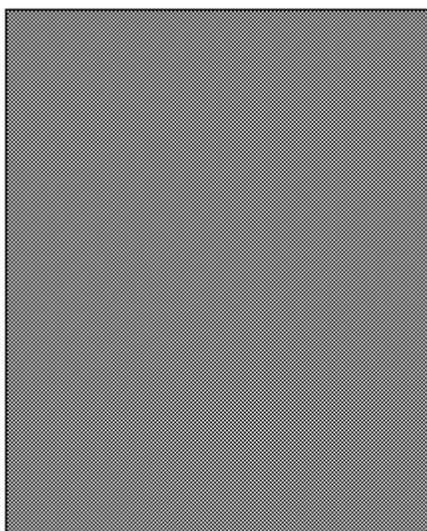
「印刷方向」が「右/下」に設定された繰返し枠があるとしてします。その繰返し枠のインスタンスは、左から右、上から下へ印刷されます。



Deptno 10	Dname ACCOUNTING	Deptno 20	Dname RESEARCH
<u>Ename</u>	<u>Sal</u>	<u>Ename</u>	<u>Sal</u>
CLARK	2450	SMITH	800
KING	5000	JONES	2975
MILLER	1300	SCOTT	3000
		ADAMS	1100
		FORD	3000

Deptno 30	Dname SALES	Deptno 40	Dname OPERATIONS
<u>Ename</u>	<u>Sal</u>	<u>Ename</u>	<u>Sal</u>
ALLEN	1600		
WARD	1250		
MARTIN	1250		
BLAKE	2850		
TURNER	1500		
JAMES	950		

しかし、オブジェクト（たとえば、図形）があるために、そのページで繰返し枠の2つのインスタンスを右方向へ配置できない場合、繰返し枠のインスタンスは、ページの下方向に印刷されることがあります。複数のインスタンスをページの右方向に印刷する空間がある場合は、繰返し枠のインスタンスがまず右方向に印刷され、次に下方向に印刷されます。



Deptno 10 Dname ACCOUNTING

Ename	Sal
CLARK	2450
KING	5000
MILLER	1300

Deptno 20 Dname RESEARCH

Ename	Sal
ADAMS	1100
FORD	3000

Deptno 30 Dname SALES

Ename	Sal
ALLEN	1600
WARD	1250
MARTIN	1250

印刷方向の制限

- ネストされた繰返し枠の「印刷方向」が「横」で、それを囲む繰返し枠の「印刷方向」が「縦」に設定されている場合、横の繰返し枠は、現行のページ幅よりも幅が広ければ、オーバーフローします。外側の次インスタンス、つまり下の繰返し枠は、右の繰返し枠の前のインスタンスが終わるまで印刷されません。

注意: 1ページ目のA3を2ページ目にオーバーフローしたA3と揃える場合は、「列モード」を使用します。

- クロス積グループ内でソース・グループが同じ「系列」の階層に属する（つまり、相互に上位項目または下位項目である）マトリックスの繰返し枠は、「印刷方向」が同じである必要

があります。マトリックス内でネストを作成する場合、クロス積グループ内の親子リレーションが使用されます。その結果、そのようなグループと関係付けられた繰返し枠は、ページ上で同じ方向に印刷される必要があります。

ソース

説明

「ソース」プロパティは、繰返し枠のデータを所有するグループです。有効なグループ名を選択します。

適用対象

繰返し枠

必須/オプション

必須

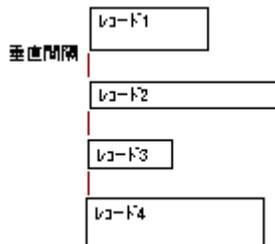
使用上の注意

- 「ソース」には、クロス積グループを指定できません。

枠間の垂直間隔

説明

「枠間の垂直間隔」プロパティは、繰返し枠のインスタンスの垂直間隔の空白の量（そのレポートのサイズの単位による）です。0以上の数値を入力します。この最大値は、サイズの単位によって異なります。単位がインチの場合、最大値は512インチです。センチメートルでは、1.312センチメートルです。パイカの場合、36,864パイカです。



適用対象

繰返し枠

必須/オプション

必須

デフォルト

0

枠間の垂直間隔の制限

- 「印刷方向」が「横」の場合、この設定には効果がありません。
- 「印刷方向」が「右/下」の場合、上下に並んで表示される2つのインスタンス間の空白に対してこの設定は有効です。

レポート・プロパティ

フォーム後型

フォーム後値

ページ後型

ページ後値

レポート後型

レポート後値

フォーム前型

フォーム前値

ページ前型

ページ前値

レポート前型

レポート前値

コメント

キャラクタ単位で設計

方向

ホスト・メニュー項目を使用不可にする

画面分割キーを使用不可にする

ズーム・キーを使用不可にする

配布

高さ(パラメータ・フォーム・ウィンドウ)

境界線を含む

ビットマップ・オブジェクトを含む

最大水平ボディ・ページ

最大垂直ボディ・ページ
名前
ページ数
ページ・ナビゲーション・コントロール・タイプ
ページ・ナビゲーション・コントロール値
パネル印刷順序
プレビューアのヒント行のテキスト
プレビューアのステータス行のテキスト
ロール名
プレビューア・タイトルの抑止
ズームで起動
タイトル
サイズの単位
プレビューアのヒント行を使用
プレビューアのステータス行を使用
幅（パラメータ・フォーム・ウィンドウ）

サイズの単位

説明

「サイズの単位」プロパティは、レポートおよびパラメータ・フォームのサイズ指定に使用されるサイズの標準です。

値

センチメートル

インチ

ポイント 1パイカを構成するプリンタの印字サイズです。1パイカは12ポイントで、1インチの6分の1に相当します。

適用対象

レポート

必須/オプション

必須

デフォルト

インチ

最大水平ボディ・ページ

説明

「最大水平ボディ・ページ」プロパティは、レポートに表示する幅でのボディ・ページの最大数です。

値

1 ~ 999の整数

適用対象

レポート

必須/オプション

必須

デフォルト

10

最大垂直ボディ・ページ

説明

「最大垂直ボディ・ページ」プロパティは、レポートに表示する高さでのボディ・ページの最大数です。

値

1 ~ 999の整数

適用対象

レポート

必須/オプション

必須

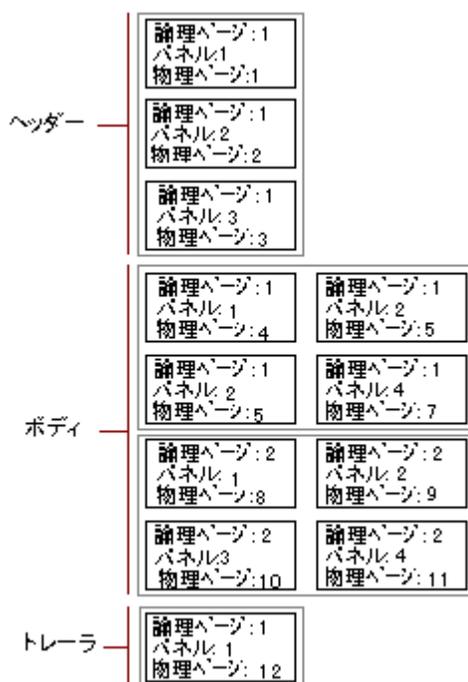
デフォルト

10

パネルの印刷順序

説明

パネル印刷順序は、論理ページを導出する物理ページが印刷される順序です。また、論理ページ内でのパネル番号の順序を決めます。図は、論理ページおよび物理ページ、パネルを示しています。



値

右/下	レポート・ボディの物理ページが左から右に印刷されてから、上から下に印刷されます。
下/右	レポート・ボディの物理ページが上から下に印刷されてから、左から右に印刷されます。

適用対象

レポート

必須/オプション

必須

デフォルト

右/下

方向

説明

「方向」プロパティでは、レポート全体の読み込み順序を指定できます。「方向」の値は、レポートにあるレイアウト・オブジェクト（たとえば、フィールド、ボイラープレートなど）のデフォルトとなります。また、レポート・エディタの方向を制御します（たとえば、「方向」が「右から左」である場合は、レポート・エディタの方向は右から左になります）。つまり、レポート・エディタが右から左になるように、手動でオブジェクトを移動する必要はありません。

値

左から右

右から左

適用対象

レポート

必須/オプション

必須

デフォルト

使用言語の自然な記述方向。

「方向」の制限

- レポートまたは任意のオブジェクトの方向の変更は、双方向プラットフォーム上での実行時にのみ有効です。

配布

説明

「配布」プロパティでは、「配布」ダイアログ・ボックスで指定された値を使用して、レポート全体の配布を定義します。

適用対象

レポート

必須/オプション

オプション

タイトル

説明

「タイトル」は、ライブ・プレビューアでレポートの一番上に表示されるテキストです。デフォルトのタイトルを変更するには、フィールドに希望するテキストを入力します。

値

256バイト以内の英数字文字列を入力します。デフォルトのタイトルは、自分で入力したタイトルのテキストを削除すれば、復元できます。

適用対象

レポート

ロール名

説明

「ロール名」プロパティは、実行時にそのレポート用に設定されるデータベースのロールを指定します。このオプションを使用して、エンド・ユーザーが、通常はアクセス権を持っていないデータベース表への問合せを行うレポートを実行することができます。Reports Runtimeは、デー

データベースに付与されているロールを設定できるだけです。Reports Runtimeではロールを設定できない場合は、そのレポートは実行されません。

値

30文字以内の有効なデータベース・ロールを入力します。

適用対象

レポート

必須/オプション

オプション

デフォルト

データベースでユーザーに対して指定されたデフォルトのロールです。これは、ユーザーがデータベースに接続すると設定されます。

使用上の注意

- レポートのプロパティ・パレットで「ロール名」プロパティを設定できます。ロール・パスワードを設定する（オプション）には、「ロール名」値フィールドのボタンをダブルクリックして、「ロール設定」ダイアログを表示します。
- Reports Runtime (RWRUN60) でコマンド・ラインを使用してロールを指定する場合、レポートのロール設定が上書きされます。
- 1つのレポートには、1つのロールしか設定できません。複数のロールを設定するには、他のロールが付与されたロールを設定する必要があります。

「ロール名」の制限

- PL/SQLトリガーにロールを設定すると矛盾が起きる可能性があるため、これはお薦めできません。

幅 (パラメータ・フォーム・ウィンドウ)

説明

「幅」プロパティは、「サイズの単位」での1パラメータ・フォーム・ページの幅です。

値

0以上の有効な幅を入力します。最大幅は、「サイズの単位」によって異なります。インチの場合、最大幅は512インチになります。センチメートルでは、1,312センチメートルです。ポイントの場合は36,864ポイントになります。

適用対象

レポート

必須/オプション

必須

デフォルト

4インチまたは10センチメートル、あるいは300ポイント

「幅」の制限 (パラメータ・フォーム・ウィンドウ)

- 「幅」を指定後に「サイズの単位」を変更する場合は、「幅」は新しい「サイズの単位」の適切な幅に自動的に変更されます。

高さ (パラメータ・フォーム・ウィンドウ)

説明

「高さ」プロパティは、「サイズの単位」での1パラメータ・フォーム・ページの高さです。

値

0以上の有効な高さを入力します。最大高は、「サイズの単位」によって異なります。インチの場合、最大高は512インチになります。センチメートルでは、1,312センチメートルです。ポイントの場合は36,864ポイントになります。

適用対象

レポート

必須/オプション

必須

デフォルト

4インチまたは10センチメートル、あるいは300ポイント

「高さ」の制限（パラメータ・フォーム・ウィンドウ）

- 「高さ」を指定後に「サイズの単位」を変更する場合は、「高さ」は新しい「サイズの単位」の適切な高さに自動的に変更されます。

ページ数

説明

「ページ数」プロパティは、ランタイム・パラメータ・フォームで希望する総ページ数です。

値

1～9,999の整数

適用対象

レポート

必須/オプション

必須

デフォルト

1

ページ・ナビゲーション・コントロール・タイプ

説明

「ページ・ナビゲーション・コントロール・タイプ」プロパティは、HTMLページ・ストリームのレポート出力のナビゲーション・ボタンについて、スクリプトの定義方法を指定します。

値

ファイル	ページ・ナビゲーション・コントロールのHTMLスクリプト・コードを含む、任意の有効なテキスト・ファイルです。
テキスト	ページ・ナビゲーション・コントロールのHTMLスクリプト・コードを含む、任意のテキスト文字列です。

適用対象

HTMLまたはHTMLCSSページ・ストリームのレポート出力

必須/オプション

オプション

デフォルト

テキスト

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_PAGE_NAVIGATION_HTML プロシージャを使用します。

ページ・ナビゲーション・コントロール値

説明

「ページ・ナビゲーション・コントロール値」プロパティは、HTMLページ・ストリームのレポート出力のナビゲーション・ボタンについて、ファイル名またはスクリプト・コードを指定します。

値

任意の有効なファイル名、または有効なHTMLコマンドを含む任意のテキスト文字列。「ページ・ナビゲーション・コントロール・タイプ」に指定したものによって決まります。必要に応じて、値フィールド (?ラベル) をクリックして、デフォルトのビルトインJavaScriptを変更できます。

適用対象

HTMLまたはHTMLCSSページ・ストリームのレポート出力

必須/オプション

オプション

デフォルト

Report BuilderのビルトインJavaScript

使用上の注意

ナビゲーション・コントロールに自分で作成したスクリプトを使用する場合、スクリプトに次の3つの変数を含める必要があります。

- &TotalPages (合計ページ数)
- &file_name (出力宛先ファイルの名前。myreportなど)
- &file_ext (ファイル拡張子。htmなど)

Report Builderによって、書式設定時にこれらの変数が割り当てられます。

ページ・ナビゲーション・コントロール枠の高さは、スクロール・バーなしで70ポイントにハードコードされます。枠の幅は可変です。この枠の中でアイコンやイメージを使用する場合は、これらの寸法を覚えておいてください。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_PAGE_NAVIGATION_HTML プロシージャを使用します。

レポート前型

説明

「レポート前型」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるたびに、文書の始めに表示されるヘッダーの型を指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力の先頭ページに適用されます。レポート全体(バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など)にHTMLを適用するように指定するには、「ページ前」プロパティを使用する必要があります。

値

テキスト	有効なHTMLコマンドを含む任意のテキスト文字列です。
ファイル	任意の有効なテキスト・ファイルまたはグラフィック・ファイルです。

適用対象

レポート

必須/オプション

必須(「レポート前値」を指定した場合)

デフォルト

テキスト

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET BEFORE REPORT HTMLプロシージャを使用します。

レポート前値

説明

「レポート前値」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、文書の始めに表示されるヘッダーを指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力の先頭ページに適用されます。レポート全体（バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など）にHTMLを適用するように指定するには、「ページ前」プロパティを使用する必要があります。

値

任意の有効なファイル名、または有効なHTMLコマンドを含む任意のテキスト文字列。「レポート前型」に指定したものによって決まります。

適用対象

レポート

必須/オプション

必須（「レポート前型」を「File」に指定した場合）

デフォルト

次のようなデフォルトのHTMLコマンドが、レポートの始めに組み込まれています。

```
<html>  
<body bgcolor="#ffffff">
```

使用上の注意

- このプロパティは、HTML文書の始めに、1つのロゴまたは複数の標準リンクを入れるのに便利です。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET BEFORE REPORT HTMLプロシーダを使用します。

「レポート前値」の制限

- デフォルトのテキストを置換する場合は、等価のHTMLコマンドが含まれていることを確認する必要があります。

レポート後型

説明

「レポート後型」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、文書の終わりに表示されるトレーラの型を指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力の最終ページに適用されます。レポート全体（バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など）にHTMLを適用するように指定するには、「ページ前」プロパティを使用する必要があります。

値

テキスト	有効なHTMLコマンドを含む任意のテキスト文字列です。
ファイル	任意の有効なテキスト・ファイルまたはグラフィック・ファイルです。

適用対象

レポート

必須/オプション

必須（「レポート後値」を指定した場合）

デフォルト

テキスト

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET AFTER REPORT HTMLプロシーダを使用します。

レポート後値

説明

「レポート後値」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、文書の終わりに表示されるトレーラを指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力の最終ページに適用されます。レポート全体（バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など）にHTMLを適用するように指定するには、「ページ前」プロパティを使用する必要があります。

値

任意のファイル名、または有効なHTMLコマンドを含む任意のテキスト文字列。「レポート後型」に指定したものによって決まります。

適用対象

レポート

必須/オプション

必須（「レポート後型」を「File」に指定した場合）

デフォルト

次のようなデフォルトのHTMLコマンドが、レポートの終わりに組み込まれています。

```
</body></html>
```

使用上の注意

- このプロパティは、HTML文書の終わりに、1つのロゴまたは複数の標準リンクを入れるのに便利です。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET AFTER REPORT HTMLプロシージャを使用します。

「レポート後値」の制限

- デフォルトのテキストを置換する場合は、等価のHTMLコマンドが含まれていることを確認してください。

ページ前型

説明

「ページ前型」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、文書の各ページの始めに表示されるヘッダーの型を指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力のすべてのページに適用されます(バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など)。レポート出力の先頭(ヘッダー)ページまたは最終(フッター)ページにのみHTMLを指定するには、「レポート前」プロパティまたは「レポート後」プロパティをそれぞれ使用します。

値

テキスト	有効なHTMLコマンドを含む任意のテキスト文字列です。
ファイル	任意の有効なテキスト・ファイルまたはグラフィック・ファイルです。

適用対象

レポート

必須/オプション

必須(「ページ前値」を指定した場合)

デフォルト

テキスト

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET BEFORE PAGE HTMLプロシージャを使用します。

ページ前値

説明

「ページ前値」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、文書の各ページの始めに表示されるヘッダーを指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力のすべてのページに適用されます(バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など)。レポート出力の先頭(ヘッダー)ページまたは最終(フッター)ページにのみHTMLを指定するには、「レポート前」プロパティまたは「レポート後」プロパティをそれぞれ使用します。

値

任意のファイル名、または有効なHTMLコマンドを含む任意のテキスト文字列。「ページ前型」に指定したものによって決まります。

適用対象

レポート

必須/オプション

必須（「ページ前型」を「File」に指定した場合）

デフォルト

ブランク

使用上の注意

- このプロパティは、HTML文書の各ページの始めに、1つのロゴまたは複数の標準リンクを入れるのに便利です。
- この値を現行のページのみ適用する場合は、そのページのオブジェクトに対する「フォーマット・トリガー」に値を指定します。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET BEFORE PAGE HTMLプロシージャを使用します。

ページ後型

説明

「ページ後型」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、文書の各ページの終わりに表示されるフッターの型を指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力のすべてのページに適用されます（バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など）。レポート出力の先頭（ヘッダー）ページまたは最終（フッター）ページにのみHTMLを指定するには、「レポート前」プロパティまたは「レポート後」プロパティをそれぞれ使用します。

値

テキスト	有効なHTMLコマンドを含む任意のテキスト文字列です。
ファイル	任意の有効なテキスト・ファイルまたはグラフィック・ファイルです。

適用対象

レポート

必須/オプション

必須（「ページ後値」を指定した場合）

デフォルト

テキスト

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET AFTER PAGE HTMLプロシージャを使用します。

ページ後値

説明

「ページ後値」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、文書の各ページの終わりに表示されるフッターを指定します。

HTMLページ・ストリームを使用する場合、このプロパティはレポート出力のすべてのページに適用されます（バックグラウンド・カラー、イメージまたはその他の<body> HTML属性など）。レポート出力の先頭（ヘッダー）ページまたは最終（フッター）ページにのみHTMLを指定するには、「レポート前」プロパティまたは「レポート後」プロパティをそれぞれ使用します。

値

任意のファイル名、または有効なHTMLコマンドを含む任意のテキスト文字列。「ページ後型」に指定したものによって決まります。

適用対象

レポート

必須/オプション

必須（「ページ後型」を「File」に指定した場合）

デフォルト

次のようなデフォルトのHTMLコマンドが、ページの終わりに組み込まれています。

```
<hr size=5 noshade>
```

使用上の注意

- このプロパティは、HTML文書の各ページの終わりに、1つのロゴまたは複数の標準リンクを入れるのに便利です。
- この値を現行のページのみ適用する場合は、そのページのオブジェクトに対する「フォーマット・トリガー」に値を指定します。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET AFTER PAGE HTMLプロシージャを使用します。

「ページ後値」の制限

- デフォルトのテキストを置換する場合は、等価のHTMLコマンドが含まれていることを確認する必要があります。

フォーム前型

説明

「フォーム前型」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、ランタイム・パラメータ・フォームの始めに表示されるヘッダーの型を指定します。

値

テキスト	有効なHTMLコマンドを含む任意のテキスト文字列です。
ファイル	任意の有効なテキスト・ファイルまたはグラフィック・ファイルです。

適用対象

レポート

必須/オプション

必須（「フォーム前値」を指定した場合）

デフォルト

テキスト

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET BEFORE FORM HTMLプロシージャを使用します。

フォーム前値

説明

「フォーム前値」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、ランタイム・パラメータ・フォームの始めに表示されるヘッダーを指定します。

値

任意の有効なファイル名、または有効なHTMLコマンドを含む任意のテキスト文字列。「フォーム前型」に指定したものによって決まります。

適用対象

レポート

必須/オプション

必須（「フォーム前型」を「File」に指定した場合）

デフォルト

次のようなデフォルトのHTMLコマンドが、ランタイム・パラメータ・フォームの始めに組み込まれています。

```
<html>
<body bgcolor="#ffffff">
<form method=post>
<center>
<p><hr>
<table border=0 cellspacing=0 cellpadding=0>
<tr>
<td><input type=submit>
<td width=15>
<td><input type=reset>>
</table>
<p><hr><p>
```

使用上の注意

- このプロパティは、HTML文書のランタイム・パラメータ・フォームの始めに、1つのロゴまたは複数の標準リンクを入れるのに便利です。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET BEFORE FORM HTMLプロシージャを使用します。

「フォーム前値」の制限

- デフォルトのテキストを置換する場合は、等価のHTMLコマンドが含まれていることを確認してください。

フォーム後型

説明

「フォーム後型」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、ランタイム・パラメータ・フォームの終わりに表示されるフッターの型を指定します。

値

テキスト	有効なHTMLコマンドを含む任意のテキスト文字列です。
ファイル	任意の有効なテキスト・ファイルまたはグラフィック・ファイルです。

適用対象

レポート

必須/オプション

必須（「フォーム後値」を指定した場合）

デフォルト

テキスト

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_AFTER_FORM_HTMLプロシージャを使用します。

フォーム後値

説明

「フォーム後値」プロパティでは、レポートがHTMLファイルに出力されてWebブラウザに表示されるときに、ランタイム・パラメータ・フォームの終わりに表示されるフッターを指定します。

値

任意の有効なファイル名、または有効なHTMLコマンドを含む任意のテキスト文字列。「フォーム後型」に指定したものによって決まります。

適用対象

レポート

必須/オプション

必須（「フォーム後型」を「File」に指定した場合）

デフォルト

次のようなデフォルトのHTMLコマンドが、ランタイム・パラメータ・フォームの終わりに組み込まれています。

```
</center>  
</body>  
</form>  
</html>
```

使用上の注意

- このプロパティは、HTML文書のランタイム・パラメータ・フォームの終わりに、1つのロゴまたは複数の標準リンクを入れるのに便利です。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_AFTER_FORM_HTMLプロシージャを使用します。

「フォーム後値」の制限

- デフォルトのテキストを置換する場合は、等価のHTMLコマンドが含まれていることを確認する必要があります。

キャラクタ単位で設計

説明

「キャラクタ単位で設計」プロパティは、レポート・エディタおよびその格子表示でキャラクタ・セル単位の使用を指定します。このプロパティおよび「境界線を含む」を使用すると、レポート・エディタではオブジェクトのキャラクタ・モード境界線が0（ゼロ）以上の幅の線で表示されません。

デフォルト

いいえ

プレビューアのヒント行を使用

説明

プレビューアのヒント行は、画面の最終行から2行目に表示されるライブ・プレビューアの行です。デフォルトのヒント行を変更するには、「はい」を選択して、「プレビューアのヒント行のテキスト」プロパティに希望するテキストを入力します。

値

「はい」である場合は、「プレビューアのヒント行のテキスト」プロパティにテキストを入力します。

適用対象

レポート

デフォルト

いいえ

使用上の注意

- 「プレビューアのヒント行のテキスト」プロパティに入力したヒント・テキストを削除して、「プレビューアのヒント行を使用」プロパティを「いいえ」に設定すると、デフォルトのヒント行を復元できます。

プレビューアのヒント行のテキスト

説明

プレビューアのヒント行は、画面の最終行から2行目に表示されるライブ・プレビューアの手行です。

値

80バイト以内の英数字文字列を入力します。

適用対象

レポート

プレビューアのステータス行を使用

説明

プレビューアのステータス行は、ライブ・プレビューアの最終行です。デフォルトのステータス行を変更するには、「はい」を選択して、「プレビューアのステータス行のテキスト」プロパティに希望するテキストを入力します。

値

「はい」である場合は、「プレビューアのステータス行のテキスト」プロパティにテキストを入力します。

適用対象

レポート

デフォルト

いいえ

使用上の注意

- 「プレビューアのステータス行のテキスト」プロパティに入力したステータス・テキストを削除して、「プレビューアのステータス行を使用」プロパティを「いいえ」に設定すると、デフォルトのステータス行を復元できます。

「プレビューアのステータス行を使用」の制限

- プレビューアのステータス行を「いいえ」に設定すると、Report Builderではそのための行が予約されません。このため、画面当たり1行余分にレポートを表示できます。
- 「プレビューアのステータス行のテキスト」を変更するとき、レポート名を入力しないとステータス行には表示されません。たとえば、次のように入力します。

This report is for Human Resources personnel only (Report: deptinfo)

プレビューアのステータス行のテキスト

説明

プレビューアのステータス行は、ライブ・プレビューアの最終行です。

値

80バイト以内の英数字文字列を入力します。

適用対象

レポート

ビットマップ・オブジェクトを含む

説明

「ビットマップ・オブジェクトを含む」プロパティは、レポートがキャラクタ・モードで実行されるときに、すべてのビットマップ処理済みオブジェクト（つまり、テキストおよび線、四角形を除くGraphics Builder図表およびボイラープレート・オブジェクト）をボックスに変換します。「いいえ」に設定すると、レポートがキャラクタ・モードで実行されるときに、すべてのビットマップ処理済みオブジェクトおよびそれらに対する明示的アンカーが削除されます。

適用対象

レポート

デフォルト

いいえ

「ビットマップ・オブジェクトを含む」の制限

- 水平または垂直でない線は、キャラクタ・モードではこの設定に関わらず、水平または垂直の線に変換されます。同様に、回転された四角形は、キャラクタ・モードではこの設定に関

ならず、回転していない四角形に変換されます。

境界線を含む

説明

「境界線を含む」プロパティは、ビットマップで0(ゼロ)以上の線幅のオブジェクトに対して、キャラクタ・モードでの境界線をオンにします。「いいえ」に設定すると、キャラクタ・モードではすべての境界線がオフになります。

適用対象

レポート

デフォルト

いいえ

使用上の注意

- このプロパティおよび「キャラクタ単位で設計」プロパティを、「はい」に設定すると、レポート・エディタではオブジェクトのキャラクタ・モード境界線が0(ゼロ)以上の幅の線で表示されます。

ホスト・メニュー項目を使用不可にする

説明

「ホスト・メニュー項目を使用不可にする」プロパティは、ランタイム・パラメータ・フォームおよびライブ・プレビューアの両方にある「ファイル」メニューのホスト項目を使用不可にします。ホスト項目はキャラクタ・モードのReport Builderの「ファイル」メニューに表示されます。

適用対象

レポート

デフォルト

いいえ

画面分割キーを使用不可にする

説明

「画面分割キーを使用不可にする」は、「垂直分割」および「水平分割」ファンクション・キーを使用不可にします。

適用対象

レポート

デフォルト

いいえ

ズーム・キーを使用不可にする

説明

「ズーム・キーを使用不可にする」プロパティは、ライブ・プレビューア表示での「ズームイン/ズームアウト」ファンクション・キーを使用不可にします。

適用対象

レポート

デフォルト

いいえ

「ズーム・キーを使用不可にする」の制限

- この機能は、UNIXでのみ使用できます。
- 「ズーム・キーを使用不可にする」および「ズームで起動」が両方とも「はい」に設定されていると、ライブ・プレビューアではズームがオンで表示され、ズームをオフにすることはできなくなります。

ズームで起動

説明

「ズームで起動」プロパティは、ズームがオンになった状態でライブ・プレビューアを表示します（つまり、ライブ・プレビューアの周りに通常表示される境界線は見えなく、レポート出力のみが見えます）。

適用対象

レポート

デフォルト

いいえ

「ズームで起動」の制限

- この機能は、UNIXでのみ使用できます。
- 「ズーム・キーを使用不可にする」および「ズームで起動」が両方とも「はい」に設定されていると、ライブ・プレビューアではズームがオンで表示され、ズームをオフにすることはできなくなります。

プレビューア・タイトルの抑止

説明

「プレビューア・タイトルの抑止」プロパティは、キャラクタ・モードのプレビューアにある「タイトル」の表示を隠します。

適用対象

レポート

デフォルト

はい

参照カーソル問合せプロパティ

コメント

最大取出し行

名前

PL/SQL文

型

PL/SQL文

説明

「PL/SQL文」のプロパティは、参照カーソルを使用してデータを戻すために、PL/SQLファンクションを入力できるPL/SQLエディタを表示するボタンです。

適用

問合せ

必須/オプション

必須

セクション・プロパティ

配布

高さ

ページ当りの水平パネル

方向

レポート高

レポート幅

ページ当りの垂直パネル

幅

配布

説明

「配布」プロパティは、「配布」ダイアログ・ボックスで指定された値を使用して、現行のセクションの配布を定義します。

適用対象

セクション

必須/オプション

オプション

高さ

説明

「高さ」プロパティは、1物理ページ（マージンを含む）の「サイズの単位」の高さです（例11インチ）。

値

0以上の有効な高さを入力します。最大高は、「サイズの単位」によって異なります。インチの場合、最大高は512インチになります。センチメートルでは、1,312センチメートルです。ポイントの場合は36,864ポイントになります。

適用対象

セクション

必須/オプション

必須

デフォルト

11インチ

高さの制限

- ページは、セクションが入る大きさである必要があります。たとえば、セクション内のフレームを「高さ」以上のサイズに拡張すると、レポートは実行されません。
- すでに「高さ」を指定した後で「サイズの単位」を変更すると、「高さ」は新しい「サイズの単位」のおおよその高さに自動的に変更されます。
- 物理ページの「幅」または「高さ」を変更すると、論理ページのサイズがそれに応じて変更され、マージンのサイズが可能な限り保たれます。

注意: 物理ページの印刷可能領域が制限されているプリンタもあります。たとえば、プリンタ用紙は8.5 x 11インチでも、プリンタで印刷できる範囲が8 x 10.5インチしかない場合があります。プリンタの印刷可能な領域を超える「幅」や「高さ」を定義した場合、レポート出力でクリッピングが起きる可能性があります。クリッピングを回避するには、プリンタの印刷可能な領域を広げるか（オペレーティング・システムで設定できる場合）、ページの印刷可能な領域のサイズになるように「幅」と「高さ」を設定します。

- コマンド・ラインでPAGESIZEパラメータに別の値を指定すると、ページの大きさを上書きできます。
- 上下のデフォルトのマージンは、それぞれ0.5インチです。

ページ当りの水平パネル

説明

「ページ当りの水平パネル」プロパティは、物理ページ（プリンタ・ページ）単位の論理ページ（レポート・ページ）の幅です。たとえば、「ページ当りの水平パネル」のサイズが1であれば、各論理ページは1物理ページの幅になります。

値

1～50までの任意の整数

適用対象

セクション

必須/オプション

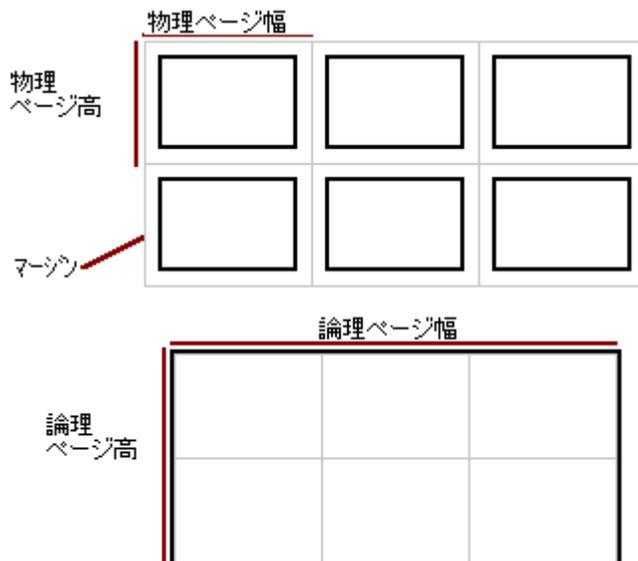
必須

デフォルト

1

ページ当りの水平/垂直パネルの例

パネルは、論理ページ内の物理ページを表します。この例では、1論理ページが6物理ページから成っています。論理ページは、幅が3物理ページで、高さが2物理ページです。したがって、「ページ当りの水平パネル」のサイズが3で、「ページ当りの垂直パネル」のサイズが2になります。論理ページの幅を2物理ページ、高さを3物理ページにする場合は、「ページ当りの水平パネル」のサイズを2、「ページ当りの垂直パネル」のサイズを3に指定します。



方向

説明

「方向」プロパティは、セクションの出力が縦方向か横方向かを指定します。この設定は、ORIENTATIONコマンド・ライン・パラメータ、システム・パラメータ、または「ページ設定」ダイアログ・ボックスで指定された方向を上書きします。

値

デフォルト	ORIENTATIONコマンド・ライン・パラメータ、システム・パラメータ、または「ページ設定」ダイアログ・ボックスで定義された方向を使用します。
縦	縦方向のセクション出力です。
横	横方向のセクション出力です。

適用対象

セクション

必須/オプション

オプション

デフォルト

デフォルト

「方向」の制限

- レポートをMotif版のPCLプリンタに出力する場合には、印刷方向の設定は無視されます。レポートはデフォルトの設定を使用して処理されます。

レポート高

説明

「レポート高」プロパティは、レポート・ページの文字単位の高さです。

値

1～9,999までの任意の整数

適用対象

セクション

必須/オプション

必須

デフォルト

80文字

レポート高の制限

- キャラクタ・モードのレポートをビットマップのライブ・プレビューア表示で表示する場合、Report Builderはこの設定を使用して表示フォントを選択します。フォントは実際の文字セル・サイズよりわずかに大きくなるか、小さくなる可能性があります。このため、文字の一部が切り捨てられる可能性があります。

レポート幅

説明

「レポート幅」プロパティは、レポート・ページの文字単位の幅です。

値

1 ~ 9,999までの任意の整数

適用対象

セクション

必須/オプション

必須

デフォルト

66文字

レポート幅の制限

- キャラクタ・モードのレポートをビットマップのライブ・プレビューア表示で表示する場合、Report Builderはこの設定を使用して表示フォントを選択します。フォントは実際の文字セル・サイズよりわずかに大きくなるか、小さくなる可能性があります。このため、文字の一部が切り捨てられる可能性があります。

ページ当りの垂直パネル

説明

「ページ当りの垂直パネル」プロパティは、物理ページ（プリンタ・ページ）単位の論理ページ（レポート・ページ）の高さです。たとえば、「ページ当りの垂直パネル」のサイズが1であれば、各論理ページは1物理ページの高さになります。

値

1 ~ 50までの任意の整数

適用対象

セクション

必須/オプション

必須

デフォルト

1

幅

説明

「幅」プロパティは、1物理ページ（マージンを含む）の「サイズの単位」の幅です（例8.5インチ）。

値

0以上の有効な幅を入力します。最大幅は、「サイズの単位」によって異なります。インチの場合、最大幅は512インチになります。センチメートルでは、1,312センチメートルです。ポイントの場合は36,864ポイントになります。

適用対象

セクション

必須/オプション

必須

デフォルト

8.5インチ

幅の制限

- ページは、セクションが入る大きさである必要があります。たとえば、セクション内のフレームを「幅」以上のサイズに拡張すると、レポートは実行されません。
- すでに「幅」を指定した後で「サイズの単位」を変更すると、「幅」は新しい「サイズの単位」のおおよその幅に自動的に変更されます。
- 物理ページの「幅」または「高さ」を変更すると、論理ページのサイズがそれに応じて変更され、マージンのサイズが可能な限り保たれます。

注意: 物理ページの印刷可能領域が制限されているプリンタもあります。たとえば、プリンタ用紙は8.5 x 11インチでも、プリンタで印刷できる範囲が8 x 10.5インチしかない場合があります。プリンタの印刷可能な領域を超える「幅」や「高さ」を定義した場合、レポート出力でクリッピングが起きる可能性があります。クリッピングを回避するには、プリンタの印刷可能な領域を広げるか（オペレーティング・システムで設定できる場合）、ページの印刷可能な領域のサイズになるように「幅」と「高さ」を設定します。

- コマンド・ラインでPAGESIZEパラメータに別の値を指定すると、ページの大きさを上書きできます。
- 左右のデフォルトのマージン・サイズは0です。

サマリー列プロパティ

ブレイク順序

列のデータ型

コメント

計算位置

データ型

ファンクション

ファイル形式

名前

積順序

ファイルからの読み込み

リセット位置

ブレイク順序の設定

ソース

NULL時の値

幅

計算位置

説明

「計算位置」プロパティは、「割合 (%)」サマリー列が計算されるグループのことです。「計算位置」は、「割合 (%)」のファンクションが設定された列に対してのみ使用します。パーセンテージを計算する場合、その値を合計で割ります(たとえば、SMITHの給与/部門の給与合計)。「計算位置」で、パーセンテージ計算の合計が定義されます。Report Builderでは、このフィールドを使用して、各ソース列値がそのパーセンテージとなる合計が決められます。マトリックス・レポートの場合、「計算位置」を複数のグループにすることができます。

値

Page	1ページの合計値のパーセンテージとして、ソース列を計算するために使用します。
Report	レポート内の合計値のパーセンテージとして、ソース列を計算するために使用します。
<names>	グループ内の合計値のパーセンテージとして、ソース列を計算するために使用します。

適用対象

サマリー列。

必須/オプション

「割合 (%)」のファンクションを持つサマリー列の場合は、必須。

デフォルト

サマリー列のグループの親グループ。

計算位置の例 (グループ・レポート)

次のようなグループ・レポートがあるとします。

```

Deptno 10  Dname      ACCOUNTING
Ename      Sal          %Dept      %Total
-----
MILLER     1300.00      14.86%     6.62%
CLARK      2450.00      28.00%     12.48%
KING       5000.00      57.14%     25.48%

Deptno 20  Dname      RESEARCH
Ename      Sal          %Dept      %Total
-----
SMITH      800.00       7.36%      4.08%
ADAMS     1100.00      10.11%     5.61%
JONES     2975.00      27.36%     15.16%
    
```

SCOTT	3000.00	27.59%	15.29%
FORD	3000.00	27.59%	15.29%

親グループのG_DEPTには、DEPTNOおよびDNAMEの列が含まれています。子グループのG_EMPには、ENAMEおよびSALの列が含まれています。

部門の給与合計 (Pct_Dept) に対するパーセンテージとして、各従業員の給与の合計を求めるには、次の設定でG_EMPにサマリー列を作成します。

ファンクション	リセット位置	計算位置
割合 (%)	G_EMP	G_DEPT

この場合、「リセット位置」は、各従業員ごとに集計がNULLに再設定されることを示しています。これは、各従業員に対して個別のパーセンテージを求めるためです。「計算位置」は、その部門の給与合計がパーセンテージを計算するための全体 (100%) となることを示しています。各従業員別のパーセンテージを表示するため、この集計がG_EMPグループに置かれていることにも注意する必要があります。

レポート全体 (Pct_Total) の給与合計のパーセンテージとして、各従業員の給与の合計を求めるには、次の設定でG_EMPにサマリー列を作成します。

ファンクション	リセット位置	計算位置
割合 (%)	G_EMP	Report

この場合、「計算位置」は、レポート全体の給与合計がパーセンテージを計算するための全体 (100%) となることを示しています。前述のPct_Dept集計の場合のように、「リセット位置」は、各従業員別のパーセンテージを表示するため、各従業員ごとに集計がNULLに再設定され、その集計がG_EMPグループに置かれることを示しています。

計算位置の例 (マトリックス・レポート)

次のようなマトリックス・レポートがあり、それにいくつかのパーセンテージの集計を追加するとします。

Dept	Analyst	Clerk	Manager	Total
10		1300	2450	3750
20	6000	1900	2975	10875
30		950	2850	3800
Total	6000	4150	8275	18425

グループG_JOBには列JOB、G_DEPTにはDEPTNO、G_SALにはSUMSALが含まれていて、G_CROSSはクロス積グループです（G_CROSSには、このマトリックスのすべての集計が含まれています）。

1部門あたりの各職種カテゴリの給与合計を、すべての部門におけるその職種の給与合計のパーセンテージとして計算する集計を求めるには、G_CROSSのサマリー列に次の設定を入力してください。

ファンクション	リセット位置	計算位置	積順序
割合 (%)	G_DEPT	G_JOB	G_JOB, G_DEPT

この場合、「リセット位置」は、各部門ごとに集計がNULLに再設定されることを示します。これは、各部門に対して個別のパーセンテージを求めるためです。「計算位置」は、各職種カテゴリの給与合計がパーセンテージを計算するための全体（100%）となることを示しています。このマトリックスに使用されるすべての集計はクロス積グループに属していなければならないので、この集計がG_CROSS内に置かれていることにも注意する必要があります。

注意: これはマトリックス・レポートなので、集計には「積順序」を使用する必要があります。「積順序」の詳細は、「積順序」を参照してください（その中の一例では、この例と同じ出力を使用して、「積順序」の使用方法がより詳しく説明されています）。

計算位置の制限

- 列がサマリー列であり、「ファンクション」が「割合 (%)」である場合にのみ、「計算位置」は編集できます。
- サマリー列のグループが「レポート」でない場合は、サマリー列の「計算位置」グループを、「レポート」および「ページ」を含めてサマリー列のグループもしくはそれより上位の任意のグループにすることができます。
- クロス積グループに属しているサマリー列には、「Page」という「計算位置」を設定できません。
- サマリーに「Page」という「計算位置」が設定されている場合は、それを表示するフィールドまたはボイラープレート・オブジェクトのサイズが固定されている必要があります。

ファンクション

説明

「ファンクション」プロパティは、「ソース」で指定された列の値に対して実行される計算です。累計集計を作成するには、該当するグループに列を入れ、「リセット位置」と「ファンクション」を組み合わせて使用します。

値

平均	リセット・グループ内の列の値の平均を計算します。
カウント	リセット・グループ内のレコード数をカウントします。
最初	リセット・グループにフェッチされた列の最初の値を表示します。
最後	リセット・グループにフェッチされた列の最後の値を表示します。
最大値	リセット・グループ内の列の最大値を計算します。
最小値	リセット・グループ内の列の最小値を計算します。
割合 (%)	リセット・グループ内の列の合計に対するパーセントを計算します。
標準偏差	リセット・グループの平方偏差の正の平方根を計算します。
合計	リセット・グループ内の列の値の合計を計算します。
分散	リセット・グループの列の平均値と各値との差を2乗し、その和を値の数より1小さい数で除算します。

適用対象

サマリー列。

必須/オプション

サマリー列の場合は必須。

デフォルト

ブランク。

ファンクションの制限

- 「ファンクション」は、列のタイプが「サマリー」である場合のみ、編集できます。
- 「合計」、「平均」、「割合 (%)」、「標準偏差」および「分散」は、「Number」のデータ型のフィールドに対してのみ、適用できます。
- Report Builderでは、集計関数値を計算される列のデータ型をベースにして、集計にデータ型が設定されます。
- 集計関数値を計算される列を変更すると、それに合わせて集計のデータ型も変更されます。
- レポート・レベルの集計には、「割合 (%)」の「ファンクション」を設定できません。レポート・レベルの列にこのファンクションを設定した場合、列値は常に100%となるので、設定する意味がありません。
- 「割合 (%)」または「合計」、「分散」として「ファンクション」が設定された列を、ブレイク列にすることはできません。また、この列にはブレイク順序を設定できません。
- 列の平均を計算するとき、NULL値は無視されます（つまり、「平均」の「ファンクション」では、NULL値と0（ゼロ）は区別されます）。

積順序

説明

「積順序」プロパティは、集計を出すためにクロス積で評価されるグループの順序のことです。「積順序」は、クロス積グループ内の集計、式またはプレースホルダの度数も定義します。つまり、集計、式、またはプレースホルダは、「積順序」におけるグループの各値の組合せに対して、それぞれ1つずつしか値を持ちません。「積順序」は、クロス積グループに属する列に対してのみ使用します。クロス積は複数のグループに関係するため、クロス積内のグループを多数の異なる順序で評価できます。したがって、クロス積の集計を作成する場合、Report Builderで「積順序」を使用して、評価する最初のグループ、2番目のグループというように指定する必要があります。また、「積順序」を使用して、クロス積内の集計、式またはプレースホルダの度数も指定する必要があります。

値

有効なグループの組合せ。

適用対象

集計、式、プレースホルダの列。

必須/オプション

クロス積グループに属するサマリー列または式列の場合は必須。

積順序の例（1つのグループ）

次のようなマトリックス・レポートがあるとします。

Job	ANALYST	CLERK	MANAGER	PRESIDENT	SALESMAN
	-----	-----	-----	-----	-----
Year	Dept				
----	----				
80	10				
	20	800			800
	30				

					800
81	10		2450	5000	7450
	20	3000	2975		5975
	30	950	2850		9400
				5600	-----
					22825
82	10		1300		1300
	20	3000			3000
	30				

					4300
83	10				
	20		1100		1100
	30				

					1100
		-----	-----	-----	-----
		6000	4150	8275	5000
					5600

グループG_JOBには列JOB、G_DEPTにはDEPTNO、G_YEARにはYEAR、G_SALにはSUMSALが含まれていて、G_CROSSはクロス積グループです（G_CROSSには、このマトリックスのすべての集計が含まれています）。

職種別の給与合計を求めるには、次の設定でG_CROSS内にサマリー列を作成します。

ファンクション	リセット位置	積順序
割合 (%)	G_JOB	G_JOB

この場合、G_JOBの各レコード（つまり、各職種カテゴリ）の合計を計算させようとしているので、「積順序」はG_JOBになります（この集計は、マトリックスの最下部に表示されます）。

年度別の給与合計を求めるには、次の設定でG_CROSS内にサマリー列を作成します。

ファンクション	リセット位置	積順序
合計	G_YEAR	G_YEAR

この場合、G_YEARの各レコード（つまり、各年度）の集計を計算させようとしているので、「積順序」はG_YEARになります。（この集計は、マトリックスの右枠に沿って、ボイラープレート・ラインのすぐ下に表示されます。）

各年度の部門別の給与合計を求めるには、次の設定でG_CROSS内にサマリー列を作成します。

ファンクション	リセット位置	積順序
合計	G_DEPT	G_YEAR, G_DEPT

この場合、集計は各年度の各部門（G_DEPTおよびG_YEAR）の合計を計算します。この集計では、G_YEARの度数が最も低く（つまり、G_DEPTの3つのレコードに対して1度しか変化しない）、G_DEPTの度数が最も高くなります（つまり、G_YEARの各レコードに対して3度変化する）。結果として、「積順序」には、最初にG_YEAR（最も度数の低いグループ）が表示され、2番目にG_DEPT（最も度数の高いグループ）が表示されます。

積順序の例（複数のグループ）

次のようなマトリックス・レポートがあり、それにいくつかのパーセンテージの集計を追加するとします。

Dept	Analyst	Clerk	Manager	Total
10		1300	2450	3750
20	6000	1900	2975	10875
30		950	2850	3800
Total	6000	4150	8275	18425

グループG_JOBには列JOB、G_DEPTにはDEPTNO、G_SALにはSUMSALが含まれていて、G_CROSSはクロス積グループです。

1部門あたりの各職種カテゴリの給与合計を、すべての部門におけるその職種の給与合計のパーセンテージとして計算する集計を求めるには、G_CROSSの集計列に次の設定を入力してください。

ファンクション	リセット位置	計算位置	積順序
割合 (%)	G_DEPT	G_JOB	G_JOB, G_DEPT

この場合、集計がマトリックスの縦方向を計算してから横方向を計算するようにしています。集計は初めは縦に向かうので、G_JOBの度数は最も低くなり（3つの部門で1度しか変化しない）、「積順序」で第1番目のグループになります。縦に向かった後、集計はマトリックスの横に向かうので、G_DEPTの度数は最も高くなり（1つの職種で3度変化する）、「積順序」では第2番目のグループになります。

Dept	Analyst	Pct_Job	Clerk	Pct_Job	Manager	Pct_Job	Total
10			1300	31.33%	2450	29.61%	3750
20	6000	100.00%	1900	45.78%	2975	35.95%	10875
30			950	22.89%	2850	34.44%	3800
Total	6000		4150		8275		18425
		100%	100%	100%			

職種カテゴリあたりの各部門の給与合計を、その部門の給与合計のパーセンテージとして計算する集計を求めるには、G_CROSSのサマリー列に次の設定を入力してください。

ファンクション	リセット位置	計算位置	積順序
割合 (%)	G_JOB	G_DEPT	G_DEPT, G_JOB

この場合、集計がマトリックスの横方向を計算してから縦方向を計算するようにしています。集計は初めは横に向かうので、G_DEPTの度数は最も低くなり(3つの職種で1度しか変化しない)、「積順序」で第1番目のグループになります。横に向かった後、集計はマトリックスの縦に向かうので、G_JOBの度数は最も高くなり(1つの部門で3度変化する)、「積順序」では第2番目のグループになります。

	Analyst	Pct_Job	Clerk	Pct_Job	Manager	Pct_Job	Total
Dept	-----	-----	-----	-----	-----	-----	-----
10			1300	34.67%	2450	65.33%	3750
							100%
20	6000	55.17%	1900	17.47%	2975	27.36%	10875
							100%
30			950	25.00%	2850	75.00%	3800
							100%
Total	6000		4150		8275		18425

職種カテゴリあたりの各部門の給与合計を、レポート内の給与合計のパーセンテージとして計算する集計を求めるには、G_CROSSのサマリー列に次の設定を入力してください。

ファンクション	リセット位置	計算位置	積順序
割合 (%)	G_JOB	Report	G_DEPT, G_JOB

この場合、集計はマトリックスの横方向を計算してから縦方向を計算することも、縦方向を計算してから横方向を計算することもあります。これは、レポート全体に対してこの集計を計算(たとえば、「計算位置」は「Report」)しているためです。「積順序」は、「G_DEPT、G_JOB」(横、縦)または「G_JOB、G_DEPT」(縦、横)のどちらでもかまいません。ただし、「リセット位置」は最も度数の高いグループである必要があることに注意してください。したがって、「G_JOB、G_DEPT」の場合はG_DEPT、「G_DEPT、G_JOB」の場合はG_JOBになります。

	Analyst	Pct_Job	Clerk	Pct_Job	Manager	Pct_Job	Total
Dept	-----	-----	-----	-----	-----	-----	-----
10			1300	7.06%	2450	13.30%	3750
20	6000	32.56%	1900	10.31%	2975	16.15%	10875
30			950	5.16%	2850	15.47%	3800
Total	6000		4150		8275		18425
							100%

積順序の例(式)

次のようなマトリックス・レポートがあり、それに1つの式を追加するとします。

Dept	Analyst	Clerk	Manager	President	Salesman
10		1300	2450	5000	

```

20      6000      1900      2975
30                950      2850                5600

```

グループG_JOBには列JOB、G_DEPTにはDEPTNO、G_SALにはSUMSALが含まれていて、G_CROSSはクロス積グループです。

各部門の課税後の給与合計を計算する式を追加するには、まず次の設定でG_CROSS内にサマリー列 (DEPTSUM) を作成します。

ファンクション	リセット位置	積順序
割合 (%)	G_DEPT	G_DEPT

次に、次の設定でG_CROSS内に式 (AFTERTAX) を作成します。

式	積順序
:deptsum * .9	G_DEPT

式はG_CROSS内にあるので、「積順序」を指定する必要があります。この場合、式の度数は、サマリー列 (DEPTSUM) と同じです。これによって、式は次のように計算されます。

```

Dept  Analyst  Clerk  Manager  President  Salesman  Comp
10                1300   2450     5000                7875
20      6000   1900   2975                9788
30                950    2850                5600   8460

```

積順序の制限

- このフィールドは、集計、式、またはブレースホルダ列のグループがクロス積である場合にのみ使用できます。
- 累計集計ではない集計（つまり、「リセット位置」が「Report」でない）には、そのソース列の「積順序」のサブセットである「積順序」が必要です。集計のソース列の度数は、集計の度数と同じか、それ以上にしてください。つまり、集計の「積順序」は、ソース列の「積順序」の一部が含まれたサブセットにしてください。
- 累計集計（つまり、「リセット位置」が「Report」）には、ソース列の「積順序」の前にある「積順序」が必要です。たとえば、ソース列の「積順序」がA、B、Cであるとすると、集計列の「積順序」はA（それ自体である場合）またはA、Bとなります。この集計の「積順序」を、B、Aとすることはできません。

リセット位置

説明

「リセット位置」プロパティは、サマリー列値が0（ゼロ）（「ファンクション」が「カウント」である場合）またはNULL（「ファンクション」が「カウント」ではない場合）、NULL時の値（列がそれを持っている場合）にリセットされるグループです。「リセット位置」には、有効なリセット・グループが含まれている値リストがあります。「リセット位置」では、集計が累計集計であるか、または周期的（たとえば、グループ・レベル）集計であるかについて判断されます。

値

Page	ページ・レベルの集計に使用します。集計はページ間でリセットされます。
Report	総計など、レポート全体に対して適用される集計に使用されます。集計は、レポートが終わるごとにリセットされます（つまり、絶対にリセットされません）。
<names>	有効なグループ名です。集計は、グループの各レコードが終わるごとにリセットされます。

適用対象

サマリー列。

必須/オプション

列タイプがサマリー列の場合は必須。

使用上の注意

- サマリー列のリセット・グループは、「レポート」および「ページ」を含めて、そのグループか、またはグループより上の任意のグループにすることができます。
- ページの集計を想定したとおりに実行するには、各レコードを確実に1ページに収めるようにします。収まらない場合は、新規レコードを次のページに印刷するようにします。

リセット位置の例（グループ）

次のようなグループ・レポートを希望しているとします。

DEPT	NAME	SAL	RESETSUM	RUNSUM
----	-----	----	-----	-----
10	SMITH	1000	1000	1000
	JONES	1000	2000	2000
	KING	1000	3000	3000
20	JOHNSON	1500	1500	4500
	WARD	1000	2500	5500

親グループG_DEPTには、DEPTNO列が含まれています。子グループG_EMPには、ENAMEおよびSALが含まれています。各部門の給与合計（RESETSUM）を求めるには、次の設定でG_EMPにサマリー列を作成します。

ファンクション	リセット位置
合計	G_DEPT

この場合、「リセット位置」は、各部門が終わるごとに集計がNULLに設定されることを示しています。

累計の給与合計（RUNSUM）を求めるには、次の設定でG_EMPにサマリー列を作成します。

ファンクション	リセット位置
合計	Report

この場合、「リセット位置」は、集計がNULLにリセットされないで、レポートの終わりまで累計加算されることを示しています。

リセット位置の制限

- 「リセット位置」は、サマリー列の場合にのみ編集できます。
- クロス積グループに属しているサマリー列では、「リセット位置」として「Page」を設定できません。
- 集計に「Page」という「リセット位置」が設定されている場合、それを表示するフィールドまたはボイラプレート・オブジェクトのサイズを固定してください。
- 列に「Page」という「リセット位置」が設定されている場合、それをソースとして使用する任意のフィールドは、その列のグループをソースとして使用する繰返し枠の中にある必要があります。
- 「Page」という「リセット位置」は、集計に必要なグループと関係付けられた複数の繰返し枠が入っているレポートに対しては、稼動しません。この一般的なケースが、マトリックス・レポートです。結果として、マトリックス・レポートにはページ・レベルの集計は使用できません。
- 同じ親グループを持つ子グループ内の複数のサマリー列の場合は、「Page」の「リセット位置」は間違った結果を生む可能性があります。
- データ・モデルに同じ親を持つ複数の子グループが含まれている場合（たとえば、マスター-ディテール-ディテール）、「Page」の「リセット位置」を持つ集計列により、予想外の結果になることがあります。

ソース

説明

「ソース」プロパティは、集計関数値を計算する列の名前です。

値

任意の有効な列名。

適用対象

サマリー列。

必須/オプション

サマリー列の場合は必須。

デフォルト

空白。

ソースの制限

- サマリー列のソースは、サマリー列のグループ内またはその下位にある（ファンクションに対して有効なデータ型の）列となります。
- 列に「ファイルからの読み込み」が指定されていない場合、サマリー列の「ソース」を変更すると、サマリー列のデータ型が新規のソース列のデータ型に合わせて変更されます。
- どの列も、複数のサマリー列の「ソース」にすることができます。
- サマリー列のソースを指定した後、レポートからソース列を削除すると、そのサマリー列からは集計関数値を計算するデータにアクセスできなくなります。

テンプレート・プロパティ

整列

サマリーをフィールドの位置に合わす

バックグラウンド・カラー

フィールドとラベル間（水平）

枠とフィールド間（水平）

枠とフィールド間（垂直）

マスターとディテール間（水平）

マスターとディテール間（垂直）

ページと枠間（水平）

ページと枠間（垂直）

同等枠間（水平）

同等枠間（垂直）

境界線

文字揃え

破線

日付揃え

枠のバックグラウンド・カラー

枠のフォアグラウンド・カラー

枠のパターン

行当りのフィールド数

塗りパターン

フォント

フォアグラウンド・カラー

イメージ
 フィールド間 (水平)
 フィールド間 (垂直)
 枠間 (水平)
 枠間 (垂直)
 位置揃え
 数値揃え
 フィールドの上にラベルを置く
 位置
 スタイル
 テキスト
 テキスト・カラー
 垂直間隔の使用

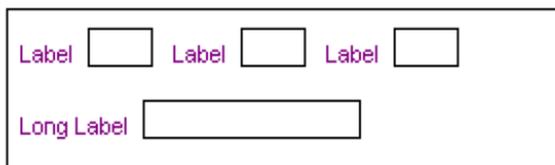
整列

説明

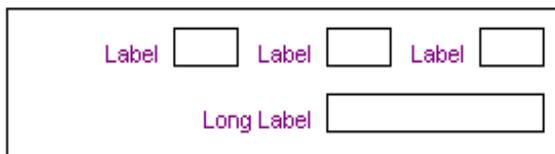
「整列」プロパティは、行にあるラベルおよびフィールドをページ内で位置づける方法を指定します。このプロパティは、「スタイル」が「フォーム」の場合にのみ適用されます。フィールドおよび列の数は「行当りのフィールド数」プロパティで決められます。

値

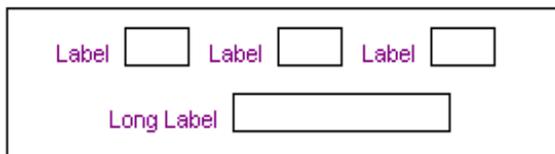
左	1番左のラベルまたはフィールドがページの左側に整列されます。
右	1番右のラベルまたはフィールドがページの右側に整列されます。
中央	ラベルおよびフィールドが、ページの中央に揃えられます。
フラッシュ	1番右のラベルまたはフィールドはページの右側に整列され、1番左のラベルまたはフィールドはページの左側に整列されて、ラベルおよびフィールドがページ内で等間隔に位置づけられます。
列	最大のラベルおよびフィールドの幅を基盤として、ページ内でラベルおよびフィールドが列上に等間隔で整列されます。



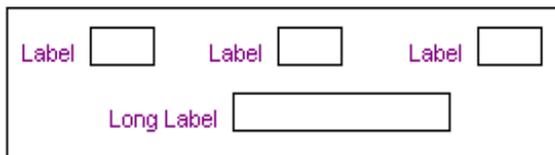
左



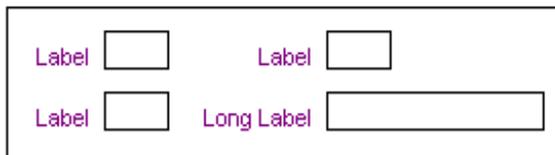
右



中央



フラッシュ



列

適用対象

テンプレート

デフォルト

左

サマリーをフィールドの位置に合わす

説明

「サマリーをフィールドの位置に合わす」プロパティが「はい」に設定されている場合には、ソース・フィールドの下にサマリー・フィールドが位置付けられます。このプロパティは、「スタイル」が「フォーム」の場合にのみ適用されます。

適用対象

テンプレート

デフォルト

はい

バックグラウンド・カラー

説明

「バックグラウンド・カラー」プロパティは、「塗りパターン」プロパティ設定のバックグラウンド・カラーを定義します。

値

カラー・パレットから有効なカラーを選択します。

適用対象

テンプレート

デフォルト

white

使用上の注意

- 「塗りパターン」プロパティが「transparent」に設定されていると、「バックグラウンド・カラー」および「フォアグラウンド・カラー」設定が無視されます。
- 「塗りパターン」プロパティが「ソリッド」に設定されていると、「バックグラウンド・カ

ラー」は無視されて、「フォアグラウンド・カラー」のみが参照できるように設定されます。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_BACKGROUND_FILL_COLORプロシージャを使用します。

フィールドとラベル間（水平）

説明

「フィールドとラベル間（水平）」プロパティは、1つのグループにあるフィールドと関係付けられたラベルの間にある水平領域の広さを定義します。このプロパティは、「スタイル」が「フォーム」の場合にのみ適用され、ラベルはフィールドの左側にあります。このように、このプロパティに対する垂直領域はありません。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

枠とフィールド間（水平）

説明

「枠とフィールド間（水平）」プロパティは、枠と枠が囲んでいるフィールドの間の水平領域の広さを定義します。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

枠とフィールド間（垂直）

説明

「枠とフィールド間（垂直）」プロパティは、枠と枠が囲んでいるフィールドの間の垂直領域の広さを定義します。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

マスターとディテール間（水平）

説明

「マスターとディテール間（水平）」プロパティは、マスター・グループ枠とディテール・グループ枠の間の水平領域の広さを定義します。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

マスターとディテール間（垂直）

説明

「マスターとディテール間（垂直）」プロパティは、マスター・グループ枠とディテール・グループ枠の間の垂直領域の広さを指定します。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

ページと枠間（水平）

説明

「ページと枠間（水平）」プロパティは、ページの枠と最上位グループ枠の間の水平領域の広さを定義します。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

ページと枠間（垂直）

説明

「ページと枠間（垂直）」プロパティは、ページの枠と最上位グループ枠の間の垂直領域の広さを定義します。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

同等枠間（水平）

説明

「同等枠間（水平）」プロパティは、同一レベルにあるグループの枠の間の水平領域の広さを定義します。グループは分かれている場合と、同じ親グループを共有している場合があります。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

同等枠間（垂直）

説明

「同等枠間（垂直）」プロパティは、同一レベルにあるグループの枠の間の垂直領域の広さを定義します。グループは分かれている場合と、同じ親グループを共有している場合があります。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

境界線

説明

「境界線」プロパティは、オブジェクトのどの境界線を表示するかを定義します。

値

すべて	すべての境界線を表示します。
上のみ	上の境界線のみを表示します。
下のみ	下の境界線のみを表示します。
左のみ	左の境界線のみを表示します。
右のみ	右の境界線のみを表示します。
上と下	上と下の境界線のみを表示します。
左と右	左と右の境界線のみを表示します。
上と右	上と右の境界線のみを表示します。
上と左	上と左の境界線のみを表示します。
下と右	下と右の境界線のみを表示します。
下と左	下と左の境界線のみを表示します。
上以外すべて	上の境界線以外のすべてを表示します。
下以外すべて	下の境界線以外のすべてを表示します。
左以外すべて	左の境界線以外のすべてを表示します。
右以外すべて	右の境界線以外のすべてを表示します。
なし	境界線を表示します。

適用対象

テンプレート

デフォルト

すべて

文字揃え

説明

フィールドの「ソース」のデータ型がCHARACTERである場合に、「文字揃え」プロパティは、フィールドまたはラベルのテキストの整列方法を定義します。

値

左	テキストが左揃えされることを意味します。
中央	テキストが中央に揃えされます。
右	テキストが右揃えされます。
始点	使用している言語が通常左右のどちらから書き始められるかによって、テキストが左または右揃えされます。たとえば、左から右に流れる言語の場合は、「始点」を指定するとテキストが左揃えされます。反対に、右から左に流れる言語の場合、「始点」を指定するとテキストが右揃えされます。
終点	使用している言語が通常左右のどちらで終わるかによって、テキストが左または右揃えされます。たとえば、左から右に流れる言語の場合は、「終点」を指定するとテキストが右揃えされます。反対に、右から左に流れる言語の場合は、「終点」を指定するとテキストが左揃えされます。

適用対象

テンプレート

デフォルト

始点

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_ JUSTIFICATIONプロシージャを使用します。

破線

説明

「破線」プロパティは、オブジェクトの線/線種スタイルを定義します。

値

リストから有効な線/線種パターンを選択します。

適用対象

テンプレート

デフォルト

実線

日付揃え

説明

フィールドの「ソース」のデータ型がDATEである場合に、「日付揃え」プロパティは、フィールドまたはラベルのテキストの整列方法を定義します。

値

左	テキストが左揃えされます。
中央	テキストが中央揃えされます。
右	テキストが右揃えされます。
始点	使用している言語が通常左右のどちらから書き始められるかによって、テキストが左または右揃えされます。たとえば、左から右に流れる言語の場合は、「始点」を指定するとテキストが左揃えされます。反対に、右から左に流れる言語の場合、「始点」を指定するとテキストが右揃えされます。
終点	使用している言語が通常左右のどちらで終わるかによって、テキストが左または右揃えされます。たとえば、左から右に流れる言語の場合は、「終点」を指定するとテキストが右揃えされます。反対に、右から左に流れる言語の場合は、「終点」を指定するとテキストが左揃えされます。

値

入力

適用対象

テンプレート

デフォルト

始点

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_ JUSTIFICATIONプロシージャを使用します。

枠のフォアグラウンド・カラー

説明

「枠のフォアグラウンド・カラー」は、「枠のパターン」プロパティ設定のフォアグラウンド・カラーを定義します。

値

カラー・パレットから有効なカラーを選択します。

適用対象

テンプレート

デフォルト

black

使用上の注意

- 「枠のパターン」プロパティが「transparent」に設定されていると、「枠のバックグラウンド・カラー」および「枠のフォアグラウンド・カラー」設定が無視されます。
- 「枠のパターン」プロパティが「ソリッド」に設定されていると、「枠のフォアグラウンド・カラー」設定のみが参照できます(「枠のバックグラウンド・カラー」設定は無視されます)。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_ FOREGROUND_ BORDER_ COLORプロシージャを使用します。

枠のバックグラウンド・カラー

説明

「枠のバックグラウンド・カラー」プロパティは、「枠のパターン」プロパティ設定のバックグラウンド・カラーを定義します。

値

カラー・パレットから有効なカラーを選択します。

適用対象

テンプレート

デフォルト

white

使用上の注意

- 「枠のパターン」プロパティが「transparent」に設定されていると、「枠のバックグラウンド・カラー」および「枠のフォアグラウンド・カラー」の両方の設定が無視されます。
- 「枠のパターン」プロパティが「ソリッド」に設定されていると、「枠のバックグラウンド・カラー」設定は無視され、「枠のフォアグラウンド・カラー」設定のみが参照されます。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_BACKGROUND_BORDER_COLORプロシージャを使用します。

枠のパターン

説明

「枠のパターン」プロパティは、オブジェクトの境界線に使用するパターンを定義します。「枠のフォアグラウンド・カラー」および「枠のバックグラウンド・カラー」プロパティを使用して、バックグラウンドおよびフォアグラウンドのカラーを定義できます。

値

パターン・パレットから有効なパターンを選択します。

適用対象

テンプレート

デフォルト

transparent

使用上の注意

- 「枠のパターン」プロパティが「transparent」に設定されていると、「枠のフォアグラウンド・カラー」および「枠のバックグラウンド・カラー」プロパティ設定が無視されます。
- 「枠のパターン」プロパティが「ソリッド」に設定されていると、「枠のフォアグラウンド・カラー」設定のみが参照できます。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_BORDER_PATTERNプロシージャを使用します。

行当りのフィールド数

説明

「行当りのフィールド数」プロパティは、行に表示される、空白を許すフィールド/ラベルの組合せの最大数を定義します。たとえば、2つのフィールド/ラベルの組合せに対して空白が1つのみあり、値が3に設定されている場合は、2つのフィールド/ラベルの組合せのみが表示されます。しかし、4つのフィールド/ラベルの組合せに対して空白があり、値が3に設定されている場合は、3が最大許可数であるために、3つの組合せのみが表示されます。このプロパティは、「スタイル」が「フォーム」の場合にのみ適用されます。

値

0（ゼロ）以上の値を入力します。0（ゼロ）は、最大値がなく、行に収まるのみのフィールド/ラベルの組合せが表示されることを示しています。

適用対象

テンプレート

デフォルト

0

塗りパターン

説明

「塗りパターン」プロパティは、オブジェクトで囲む領域に使用するパターンを定義します。「フォアグラウンド・カラー」および「バックグラウンド・カラー」プロパティを使用して、塗りパターンのバックグラウンドおよびフォアグラウンドのカラーを定義できます。

値

パターン・パレットから有効なパターンを選択します。

適用対象

テンプレート

デフォルト

transparent

使用上の注意

- 「塗りパターン」プロパティが「transparent」に設定されていると、「フォアグラウンド・カラー」および「バックグラウンド・カラー」プロパティ設定が無視されます。
- 「塗りパターン」プロパティが「ソリッド」に設定されていると、「フォアグラウンド・カラー」設定のみが参照できます。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_FILL_PATTERNプロシージャを使用します。

フォント

説明

「フォント」プロパティは、オブジェクトで使用するテキスト・フォントを定義します。

値

使用しているシステムのフォントのリストから有効なフォントを選択してください。

適用対象

テンプレート

デフォルト

Courier、10ポイント、標準

PL/SQL

PL/SQLを使用してフォントを定義するには、SRW.SET_FONT_FACEプロシージャを使用します。PL/SQLを使用してフォント・サイズを定義するには、SRW.SET_FONT_SIZEプロシージャを使用します。PL/SQLを使用してフォント・スタイルを定義するには、SRW.SET_FONT_STYLEプロシージャを使用します。

フォアグラウンド・カラー

説明

「フォアグラウンド・カラー」は、「塗りパターン」プロパティ設定のフォアグラウンド・カラーを定義します。

値

カラー・パレットから有効なカラーを選択します。

適用対象

テンプレート

デフォルト

black

使用上の注意

- 「塗りパターン」プロパティが「transparent」に設定されていると、「フォアグラウンド・カラー」および「バックグラウンド・カラー」設定が無視されます。
- 「塗りパターン」プロパティが「ソリッド」に設定されていると、「フォアグラウンド・カラー」設定のみが参照できます（「バックグラウンド・カラー」設定は無視されます）。

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_FOREGROUND_FILL_COLORプロシージャを使用します。

イメージ

説明

「イメージ」プロパティは、使用しているテンプレート内のリンク・ファイル(外部)・ボイラープレート・オブジェクトで参照されるイメージ用の代用ファイルです。

値

有効なテキスト・ファイルの名前を入力します。

適用対象

テンプレート

デフォルト

なし

フィールド間 (水平)

説明

「フィールド間 (水平)」プロパティは、1つのグループにあるフィールド間の水平領域の広さを定義します。「スタイル」が「フォーム」の場合、このプロパティは、1つのグループにあるフィールド/ラベルの組合せの終点と別のフィールド/ラベルの組合せの始点間の水平領域の広さを定義します。フィールドとラベル間の距離は、「フィールドとラベル間 (水平)」プロパティの設定で決められます。

値

0 (ゼロ) 以上の値を入力します。値はレポートの「サイズの単位」(インチ、センチメートルまたはポイント)になります。

適用対象

テンプレート

デフォルト

.174インチ

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイ

ントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。

- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

フィールド間（垂直）

説明

「フィールド間（垂直）」プロパティは、1つのグループにあるフィールド間の垂直領域の広さを定義します。

値

0（ゼロ）以上の値を入力します。値はレポートの「サイズの単位」（インチ、センチメートルまたはポイント）になります。

適用対象

テンプレート

デフォルト

0

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトが一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

枠間（水平）

説明

「枠間（水平）」プロパティは、1つのグループにある枠間の水平領域の広さを定義します。

値

0 (ゼロ) 以上の値を入力します。値はレポートの「サイズの単位」(インチ、センチメートルまたはポイント)になります。

適用対象

テンプレート

デフォルト

.0087インチ

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。
- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

枠間 (垂直)

説明

「枠間 (垂直)」プロパティは、1つのグループにある枠間の垂直領域の広さを定義します。

値

0 (ゼロ) 以上の値を入力します。値はレポートの「サイズの単位」(インチ、センチメートルまたはポイント)になります。

適用対象

テンプレート

デフォルト

.0136 インチ

使用上の注意

- 「格子指定」がオンの場合は、この設定に関係なく、オブジェクトは一番近い格子指定ポイントに指定されます。より細かい間隔制御をする場合は、レポートの作成時に「格子指定」

をオフにできます。しかし、これはオブジェクトが格子指定ポイントに指定されなくなることを意味します。

- レポートの「サイズの単位」を変更すると、それに従ってこの設定も変更する必要があります。

位置揃え

説明

「位置揃え」プロパティは、タイトルのテキストの整列方法を定義します。

値

左	テキストが左側に整列されます。
中央	テキストが中央に整列されます。
フラッシュ	テキストが左側と右側に整列されます。
右	テキストが右側に整列されます。
始点	使用している言語が通常左右のどちらから書き始められるかによって、テキストが左側または右側に整列されます。たとえば、左から右に流れる言語の場合は、「始点」を指定するとテキストが左側に整列されます。反対に、右から左に流れる言語の場合、「始点」を指定するとテキストが右側に整列されます。
終点	使用している言語が通常左右のどちらで終わるかによって、テキストが左側または右側に整列されます。たとえば、左から右に流れる言語の場合は、「終点」を指定するとテキストが右側に整列されます。反対に、右から左に流れる言語の場合は、「終点」を指定するとテキストが左側に整列されます。

適用対象

テンプレート

デフォルト

始点

数値揃え

説明

フィールドの「ソース」のデータ型がNUMBERである場合に、「数値揃え」プロパティは、フィールドまたはラベルのテキストの整列方法を定義します。

値

左	テキストが左側に整列されます。
---	-----------------

中央	テキストが中央に整列されます。
右	テキストが右側に整列されます。
始点	使用している言語が通常左右のどちらから書き始められるかによって、テキストが左側または右側に整列されます。たとえば、左から右に流れる言語の場合は、「始点」を指定するとテキストが左側に整列されます。反対に、右から左に流れる言語の場合、「始点」を指定するとテキストが右側に整列されます。
終点	使用している言語が通常左右のどちらで終わるかによって、テキストが左側または右側に整列されます。たとえば、左から右に流れる言語の場合は、「終点」を指定するとテキストが右側に整列されます。反対に、右から左に流れる言語の場合は、「終点」を指定するとテキストが左側に整列されます。

適用対象

テンプレート

デフォルト

始点

PL/SQL

PL/SQLを使用してこの属性を定義するには、SRW.SET_JUSTIFICATIONプロシージャを使用します。

フィールドの上にラベルを置く

説明

「フィールドの上にラベルを置く」プロパティは、「はい」に設定されている場合には、ラベルをフィールドの左側ではなくフィールドの上に位置決めします。このプロパティは、「スタイル」が「フォーム」の場合にのみ適用されます。

適用対象

テンプレート

デフォルト

いいえ

位置

説明

「位置」プロパティは、子オブジェクトに関する親オブジェクトの配置を定義します。

適用対象

テンプレート

デフォルト

親オブジェクトは、子オブジェクトの上に表示されます。

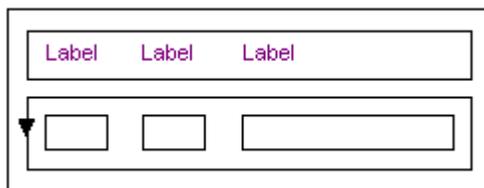
スタイル

説明

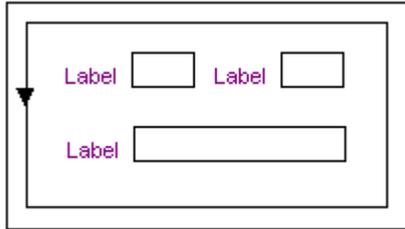
「スタイル」プロパティは、グループに「表」スタイル・レイアウトまたは「フォーム」スタイル・レイアウトのどちらが指定されるかを定義します。このプロパティを使用して、以下にリストされているデフォルトを上書きすることができます。

値

表	ラベルがフィールドの上にあり、フィールドを含む繰返し枠の外側にあることを意味します。
フォーム	ラベルがフィールドの左側にあり、フィールドを含む繰返し枠の内側にあることを意味します。



表



フォーム

適用対象

テンプレート

デフォルト

デフォルトの「スタイル」は、次のように、テンプレートのレイアウト・スタイルで決められません。

表	表
グループ左	表
グループ右	子グループのあるグループの場合は「フォーム」。 子グループのないグループの場合は「表」。
フォーム形式	フォーム
フォーム・レター	該当なし（上書き不可）。
メール・ラベル	該当なし（上書き不可）。
マトリックス	表（上書き不可）。
グループ別マトリックス	表（上書き不可）。

テキスト

説明

「テキスト」プロパティは、テンプレート内のボイラープレート・テキスト・オブジェクトで使用する代替ファイルです。

値

有効なテキスト・ファイルの名前を入力します。

適用対象

テンプレート

デフォルト

なし

テキスト・カラー

説明

「テキスト・カラー」プロパティは、オブジェクトで使用する文字色を定義します。

値

カラー・パレットから有効なカラーを選択します。

適用対象

テンプレート

デフォルト

black

PL/SQL

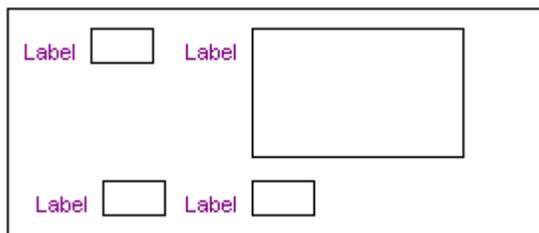
PL/SQLを使用してこの属性を定義するには、SRW.SET COLOR TEXTプロシージャを使用します。

垂直間隔の使用

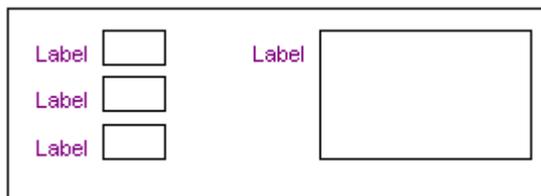
説明

「垂直間隔の使用」プロパティは、「はい」に設定されている場合は、オブジェクトを前行の未使用の領域に置きます。たとえば、後続の行にあるオブジェクトが前行の未使用領域に収まる場合は、オブジェクトは未使用領域を占めるように移動されます。

「いいえ」に設定されているときは、前行の未使用領域は保存されます。



「はい」に設定されているときは、オブジェクトは未使用領域を埋めるように移動されます。



このプロパティは、「スタイル」が「フォーム」の場合にのみ適用されます。

適用対象

テンプレート

デフォルト

はい

XMLプロパティ

- XMLタグ
- XMLタグ属性
- XML出力から除外
- XML Prolog型
- XML Prolog値
- 外部XMLタグ
- 外部XML属性
- XMLタグを含む

XMLの設定

説明

これらのプロパティを使用して、レポートについてのデフォルトのXML設定を操作することができます。プロパティにより、レポート・レベル、グループ・レベル、列レベルの3つのレベルの項目を制御できるもの、1つのレベルまたは2つのレベルの制御に限られるものがあります。

この見出しノードの下のプロパティのヘルプを表示するには、プロパティをクリックし、[F1]を押してください。プロパティ・パレット自体のヘルプを表示するには、[Ctrl]を押しながらプロパティをクリックしてプロパティの選択を解除し、[F1]または  を押してください。

XMLタグ

説明

「XMLタグ」プロパティには、現在選択されているレポート、グループあるいは列の要素名が含まれ、ユーザー定義が可能です。XMLタグの目的はオブジェクトを特定の型あるいはデータのカテゴリとして一意に識別することなので、任意に要素の名前を選択できます。XML出力を生成する理由の1つは、オブジェクトを外部のDTD仕様に合った構成にするためです。これにより、レポートとサード・パーティーのアプリケーションとの統合が可能になります。

適用対象

レポート、グループ、列

必須/オプション

オプション

デフォルト

オブジェクト名（「名前」プロパティに表示されるのと同じもの）

使用上の注意

- 任意の長さの文字列を含むことができます。
- XML要素の名前は、プロパティ・パレットのオブジェクト名に依存します。「名前」プロパティを変更すると、それに合わせて「XMLタグ」プロパティも変更されます。
- このオブジェクトについてはXMLを出力せずに子のタグおよびデータを出力する場合にはこのプロパティを空白にしてください。詳細は、XMLタグの例を参照してください。

XMLタグ属性

説明

「XMLタグ属性」プロパティを使用して、XMLの属性を入力できます。オブジェクトのタグに属性を追加するのは、そのタグについての詳細を記述したり、データ値を特定したりするためです。タグの属性の使用は、XML出力を編成する方法の1つです。この方法をとる場合は、このプロパティに値を入力する必要があります。もう1つの編成方法は、データの型（グループあるいは列）ごとに別のタグを割り当てる方法です。この場合は、プロパティを空白にしてください。

適用対象

レポート、グループ、列

必須/オプション

オプション

デフォルト

空白

構文

XMLの属性は、名前の値の組をスペースで区切って入力してください。次に例を示します。

XMLタグ属性	[attribute name]="&[report object name]"
---------	--

大カッコ () 内のテキストはユーザーが選択します。レポートのオブジェクト名には、レポート内の任意の列あるいはユーザー・パラメーターを使用できます。このプロパティには、任意の数の属性を追加できます。それぞれの属性の間にスペースを入れてください。

使用上の注意

- 他のXMLタグに属性として追加されたデータ・オブジェクトは、冗長性を少なくするため、XML出力から除外されます。詳細は、XMLタグ属性の例(グループ・レベル)を参照してください。
- 属性で参照する列あるいは変数は、同じあるいは少ない頻度で発生するものである必要があります。

XML出力から除外

説明

「XML出力から除外」プロパティを使用して、選択したグループまたは列についてXMLタグおよび関連データ(子データを含む)の出力を止めることができます。オブジェクトの除外を選択すると、XMLタグ、属性、データ値および子データ、たとえばそのグループ内の列が出力されません。このプロパティは、冗長なあるいは不必要と思われるデータを除外できるので、XMLドキュメントをフォーマットし直すのに便利です。

適用対象

レポート、グループ、列

必須/オプション

オプション

デフォルト

いいえ

値

はい

いいえ

使用上の注意

- オブジェクトを属性として別の要素に追加したあと、そのオブジェクトのデータをもうXML出力内の別の要素として表示しない場合に、このプロパティが特に便利です。
- レポート・レベルで「XML出力からの除外」プロパティを「あり」に設定した場合、すべてのデータがレポートの子データなので、レポートからのXML生成ができなくなります。

XML Prolog型

説明

「XML Prolog型」プロパティを使用して、外部のprologファイルまたはReport Builderによって生成されたテキストprologを選択できます。XML prologには、XML宣言、コメント、処理命令(PI)および文書型宣言(DTD)が含まれます。prologの詳細はXML Prolog値を参照してください。

適用対象

レポート

必須/オプション

必須

デフォルト

テキスト

値

テキスト	「XML Prolog値」プロパティで、prologをテキストとして入力してください。
ファイル	「XML Prolog値」プロパティを使用して、外部prologファイルへナビゲートします。

XML Prolog値

説明

「XML Prolog値」プロパティを使用して、適格なXML出力すべての最初に表示されるXML Prologの値を編集できます。XML prologにはXML宣言が含まれますが、その他にコメント、処理命令（スタイルシート[XSL]など）およびDTDも含むことができます。

適用対象

レポート

必須/オプション

適格なXML出力にはXML宣言が必須です。ただし、このプロパティ内の任意の英数字を使用して、Report Builder内にRaw XML出力を生成できます。

デフォルト

```
<?xml version="1.0"?>
```

構文

例

XML宣言:	<code><?xml version="1.0" encoding="UTF-8"?></code>
コメント:	<code><!-- this is how you do comments in XML --></code>
処理命令(PI):	<code><?xml:stylesheet type="text/xsl" href="emp.xsl" ?></code>
文書型宣言(DTD):	<code><!DOCTYPE EMP_TABULAR [<!ELEMENT EMP_TABULAR (EMP_ITEM)*> <!ELEMENT EMP_ITEM (EMPNO)> <!ELEMENT EMPNO (#PCDATA)>]></code>

外部XMLタグ

説明

「外部XMLタグ」フィールドは、選択されたグループの名前とLIST_prefixとを表示します。外部XMLタグの目的は、グループ・オブジェクトのインスタンス内のすべてのタグおよびデータを囲むことです。レポート内では、グループ・オブジェクトのインスタンスは繰り返されます(たとえば10、20、30のように出力するdeptnoなど)。外部XMLタグは、すべての繰り返しインスタンスのデータをカプセル化します。

適用対象

グループ

必須/オプション

オプション

デフォルト

```
<LIST_G_[groupname]>
```

使用上の注意

- このフィールドは空白のままでもかまいません。しかし、空白が便利なのは、レポートの問合せ内にグループのインスタンスが1つしかない場合のみです。

外部XML属性

説明

「外部XML属性」プロパティを使用して、外部のXMLタグについてのXMLの属性を入力することができます。外部のタグに属性を追加するのは、そのタグについての詳細を記述したり、データ値を特定したりするためです。

適用対象

グループ

必須/オプション

オプション

デフォルト

空白

構文

外部XMLの属性は、名前の値の組をスペースで区切って入力してください（HTML属性の場合と同じ）。次に例を示します。

外部XMLタグ属性:	[attribute name]="&[report object name]"
------------	--

ここで、大カッコ（[]）内のテキストはユーザーが選択します。このプロパティには、任意の数の属性を追加できます。それぞれの属性の間にスペースを入れてください。

使用上の注意

- 他のXMLタグに属性として追加されたデータ・オブジェクトは、冗長性を少なくするため、XML出力から除外されることに注意してください。詳細はXMLから除外を参照してください。
- 属性で参照する列あるいは変数は、より少ない頻度で発生するものである必要があります。

XMLタグを含む

説明

「XMLタグを含む」プロパティは、レポート列にXMLタグが含まれるかどうかを表示します。プロパティを「はい」に設定すると、Report Builderは<、>および&と<lt;, >および&などのXML構文関連の文字を変換しません。

適用対象

キャラクタのソース・データ型を持つ列

必須/オプション

オプション

デフォルト

いいえ

「XMLタグ属性」の例(グループ・レベル)

表レポートに次の問合せを使用しているとします。

```
SELECT * FROM DEPT
```

デフォルトのXML出力は次のようになります。

```
<MODULE2>
  <LIST_G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>10</DEPTNO>
      <DNAME>ACCOUNTING</DNAME>
      <LOC>NEW YORK</LOC>
    </G_DEPTNO>
  <G_DEPTNO>
    <DEPTNO>20</DEPTNO>
    <DNAME>RESEARCH</DNAME>
    <LOC>DALLAS</LOC>
  </G_DEPTNO>
```

与えられた部門番号(DEPTNO)に関係付けられたすべてのデータを、その部門番号<DEPTNO>の要素に収めているXML出力として生成するとします。これにより、多数の出力要素を減らすことができます。

データ・モデルから、DEPTNO列について次のプロパティを指定します。

XMLタグ	DEPTNO
XMLタグ属性	NAME="&DNAME" LOCATION="&LOC"

ここで&DNAMEは部門名、&LOCはその部門の場所です。この2つのオブジェクトは、DEPTNOと同じ頻度で繰り返されます。

新しいXML出力は次のようになります。

```
<G_DEPTNO>
  <DEPTNO NAME="ACCOUNTING" LOCATION="NEW YORK">10</DEPTNO>
  <DNAME>ACCOUNTING</DNAME>
  <LOC>NEW YORK</LOC>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="RESEARCH" LOCATION="DALLAS">20</DEPTNO>
  <DNAME>RESEARCH</DNAME>
  <LOC>DALLAS</LOC>
</G_DEPTNO>
```

DNAMEおよびLOCの値が今度は二度出力されているのがわかります。一度が属性として、もう一度がXML要素としての出力です。XML要素としての出力を削除するには、「XML出力から除外」プロパティも使用する必要があります。

データ・モデルでDNAME列およびLOC列についての「XML出力から除外」プロパティを「はい」に設定します。

この結果、XML出力は次のようになります。

```
<G_DEPTNO>
  <DEPTNO NAME="ACCOUNTING" LOCATION="NEW YORK">10</DEPTNO>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="RESEARCH" LOCATION="DALLAS">20</DEPTNO>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="SALES" LOCATION="CHICAGO">30</DEPTNO>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="OPERATIONS" LOCATION="BOSTON">40</DEPTNO>
</G_DEPTNO>
```

「XMLタグ属性」の例（レポート・レベル）

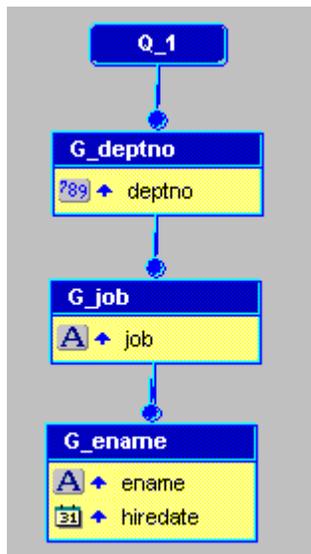
次の例では、レポート・レベルの属性がXMLにおいてどのように役に立つか示しています。

レポートに次の問合せを使用しているとします。

```
select deptno, job, ename, hiredate from emp
```

レポートはグループ左として設計され、DEPTNOが1番目のレベルのグループ、JOBが2番目のレベルのグループになっています。この例では、データ・モデルでDEPTNO列およびJOB列を選択し、「ブレイク順序」プロパティを「降順」に指定しておきます。データ・モデルのDEPTNO列およびJOB列の左向きの矢印が、これで下を向きます。

データ・モデルは次のようになっています。



XML出力は次のようになります。

```

<MODULE2>
  <LIST_G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>30</DEPTNO>
    </G_DEPTNO>
  </LIST_G_DEPTNO>
  <LIST_G_JOB>
    <G_JOB>
      <JOB>SALESMAN</JOB>
    </G_JOB>
  </LIST_G_JOB>
  <LIST_G_ENAME>
    <G_ENAME>
      <ENAME>ALLEN</ENAME>
      <HIREDATE>20-FEB-81</HIREDATE>
    </G_ENAME>
    <G_ENAME>
      <ENAME>WARD</ENAME>
      <HIREDATE>22-FEB-81</HIREDATE>
    </G_ENAME>
    <G_ENAME>
      <ENAME>MARTIN</ENAME>
      <HIREDATE>28-SEP-81</HIREDATE>
    </G_ENAME>
  </LIST_G_ENAME>
</MODULE2>

```

後で参考にするため、部門番号30内の最初の3人の従業員の雇用日が20-FEB-81から28-SEP-81であることを注意してください。

次に、レポートにユーザー・パラメータを追加して、雇用日によってデータを選択したり排除したりしてみます。データ・モデルの問合せ(Q_1)をダブルクリックします。

「SQL問合せ文」ダイアログ・ボックスで、入力されている問合せの最後に次のテキストを追加します。

```
where hiredate > :p_date
```

これで、問合せは次のようになります。

```
Select deptno, job, ename, hiredate from emp where hiredate > :p_date
```

このSQL文はReport Builderに、p_dateの名前を持つパラメータを作成するよう指示します。ただし、「データ型」プロパティを、手動でDateに変更する必要があります。

1. オブジェクト・ナビゲータで、データ・モデル下のユーザー・パラメータを確認します。ユーザー・パラメータの左側のプラス記号をクリックします。
2. オブジェクトP_DATEをダブルクリックします。
3. プロパティ・パレットで、「データ型」プロパティ・フィールドをクリックし、ドロップ・ダウン・メニューから「Date」を選択します。

次に、このユーザー・パラメータをレポート要素の属性に割り当てることを考えます。パラメータは、レポートのXML出力全体を通して様々な要素を含めたり除外したりする規則を提供します。

オブジェクト・ナビゲータのレポート・ノードをダブルクリックします。

プロパティ・パレットの「XMLタグ属性」プロパティ・フィールドにHiredateLaterThan="&p_date"と入力します。

ライブ・プレビューアでレポートを実行します。ランタイム・パラメータ・フォームが表示され、日付値(P Date)を要求されます。「1-JUL-81」と入力し、をクリックします。

これでXMLは次のようになります。

```
<MODULE2 HiredateLaterThan="01-JUL-81">
  <LIST_G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>30</DEPTNO>
    <LIST_G_JOB>
      <G_JOB>
        <JOB>SALESMAN</JOB>
      <LIST_G_ENAME>
        <G_ENAME>
          <ENAME>MARTIN</ENAME>
          <HIREDATE>28-SEP-81</HIREDATE>
        </G_ENAME>
      </G_JOB>
    </LIST_G_JOB>
  </G_DEPTNO>
</LIST_G_DEPTNO>
</MODULE2>
```

1-July-81より前のHIREDATEを持つグループG_ENAMEのインスタンスが、XMLで使用できなくなっていることに注意してください。このパラメータHiredateLaterThan="01-JUL-81"を定義する属性が、レポート要素MODULE2に表示されています。

「XML出力から除外」の例（データ出力の制限）

データをいくつかの指定した要素に制限するXML出力を生成するとします。現在、XML出力には部門番号、部門名および場所が含まれています。XML出力は次のようになっています。

```
<G_DEPTNO>
  <DEPTNO>10</DEPTNO>
  <DNAME>ACCOUNTING</DNAME>
  <LOC>NEW YORK</LOC>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO>20</DEPTNO>
  <DNAME>RESEARCH</DNAME>
  <LOC>DALLAS</LOC>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO>30</DEPTNO>
  <DNAME>SALES</DNAME>
  <LOC>CHICAGO</LOC>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO>40</DEPTNO>
  <DNAME>OPERATIONS</DNAME>
  <LOC>BOSTON</LOC>
</G_DEPTNO>
```

<LOC>タグおよび関係付けられたデータを削除することにします。

データ・モデル・ビューで、LOC列をダブルクリックし、プロパティ・パレットで次のプロパティを指定します。

XML出力から除外	はい
-----------	----

これでXML出力は次のようになります。

```

<LIST_G_DEPTNO>
  <G_DEPTNO>
    <DEPTNO>10</DEPTNO>
    <DNAME>ACCOUNTING</DNAME>
  </G_DEPTNO>
  <G_DEPTNO>
    <DEPTNO>20</DEPTNO>
    <DNAME>RESEARCH</DNAME>
  </G_DEPTNO>
  <G_DEPTNO>
    <DEPTNO>30</DEPTNO>
    <DNAME>SALES</DNAME>
  </G_DEPTNO>
  <G_DEPTNO>
    <DEPTNO>40</DEPTNO>
    <DNAME>OPERATIONS</DNAME>
  </G_DEPTNO>
</LIST_G_DEPTNO>

```

<LIST_G_DEPTNO>タグまたは<G_DEPTNO>タグを削除したいが、(DEPTNOまたはDNAMEなどの)子データは除外しない場合には、データ・モデルでグループG_DEPTNOを選択し、そのプロパティ・パレットをオープンします。「XMLタグ」プロパティ・フィールドからG_DEPTNOのXMLタグを削除し、「外部XMLタグ」プロパティ・フィールドからLIST_G_DEPTNOのXMLタグを削除します。詳細は、「外部XMLタグ」プロパティ を参照してください。

「XML出力から除外」の例（冗長性の制限）

現在次のようになっているXML出力の属性を使用するとします。

```

<G_DEPTNO>
  <DEPTNO NAME="ACCOUNTING" LOCATION="NEW YORK">10</DEPTNO>
  <DNAME>ACCOUNTING</DNAME>
  <LOC>NEW YORK</LOC>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="RESEARCH" LOCATION="DALLAS">20</DEPTNO>
  <DNAME>RESEARCH</DNAME>
  <LOC>DALLAS</LOC>
</G_DEPTNO>

```

冗長性をなくすには、XML出力からDNAMEおよびLOCの要素を除外する必要があります。

データ・モデル表示でDNAME列およびLOC列のプロパティ・パレットを表示し、次のプロパティを指定します。

XML出力から除外	はい
-----------	----

この結果、XML出力は次のようになります。

```

<G_DEPTNO>
  <DEPTNO NAME="ACCOUNTING" LOCATION="NEW YORK">10</DEPTNO>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="RESEARCH" LOCATION="DALLAS">20</DEPTNO>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="SALES" LOCATION="CHICAGO">30</DEPTNO>
</G_DEPTNO>
<G_DEPTNO>
  <DEPTNO NAME="OPERATIONS" LOCATION="BOSTON">40</DEPTNO>
</G_DEPTNO>

```

「XMLタグ」の例

レポート、グループ、問合せの列についてXML出力を作成するとします。問合せ文、

```
SELECT deptno, dname, loc FROM dept
```

が表レポート内にあり、これにより次のXML出力が生成されます。

```

<MODULE1>
  <LIST_G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>10</DEPTNO>
      <DNAME>ACCOUNTING</DNAME>
      <LOC>NEW YORK</LOC>
    </G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>20</DEPTNO>
      <DNAME>RESEARCH</DNAME>
      <LOC>DALLAS</LOC>
    </G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>30</DEPTNO>
      <DNAME>SALES</DNAME>
      <LOC>CHICAGO</LOC>
    </G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>40</DEPTNO>
      <DNAME>OPERATIONS</DNAME>
      <LOC>BOSTON</LOC>
    </G_DEPTNO>
  </LIST_G_DEPTNO>
</MODULE1>

```

出力にさまざまなタグが表示されないようにするには、「XMLタグ」プロパティ・フィールドのデフォルト名を削除し、ブランクのままにしておきます。これでタグの出力は削除されますが、子のタグに関係付けられたデータはすべて残ります。

たとえば、G_DEPTNOタグを削除すると、XML出力は次のようになります。

```
<MODULE1>
  <LIST_>
    <DEPTNO>10</DEPTNO>
    <DNAME>ACCOUNTING</DNAME>
    <LOC>NEW YORK</LOC>
    <DEPTNO>20</DEPTNO>
    <DNAME>RESEARCH</DNAME>
    <LOC>DALLAS</LOC>
    <DEPTNO>30</DEPTNO>
    <DNAME>SALES</DNAME>
    <LOC>CHICAGO</LOC>
    <DEPTNO>40</DEPTNO>
    <DNAME>OPERATIONS</DNAME>
    <LOC>BOSTON</LOC>
  </LIST_>
</MODULE1>
```

前に<LIST_G_DEPTNO>と呼ばれていたタグがここでは<LIST_>となっていることに注意してください。これは「外部XMLタグ」と呼ばれ、グループ・タグに関係付けられているのみです。外部XMLタグの名前は、グループのXMLタグの名前から選択されます。<LIST_>タグも削除する場合は、「外部XMLタグ」フィールドをブランクのままにします。詳細は、「外部XMLタグ」プロパティを参照してください。

「外部XMLタグ」の例

次の問合せを含むレポートがあるとします。

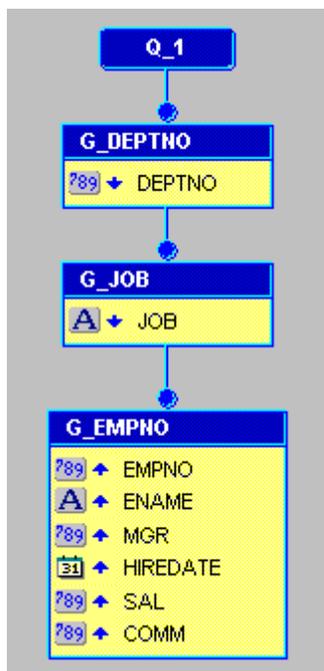
```
SELECT * from EMP
```

このデータをグループ左レポートとして表示することとし、DEPTNOを1番目のレベルのグループ、JOBを2番目のレベルのグループとします。

この例では、グループDEPTNOおよびJOBの出力順を昇順から降順に入れ替えることを考えます。

1. データ・モデルから列DEPTNOを選択し、プロパティ・パレットをオープンします。
2. 「ブレイク順序」プロパティを「昇順」から「降順」に変更します。
3. 1、2を列JOBについても繰り返します。

データ・モデルは次のようになっています。



次にレポートのグループについて、XML出力の階層構造を表示するとします。G_JOBをG_DEPTNOの子グループにしたので、外部グループ<LIST_G_DEPTNO>内のすべての要素を縮小してデータの構造を一度に1レベル拡張します。

```

- <MODULE1>
  - <LIST_G_DEPTNO>
    + <G_DEPTNO>
    + <G_DEPTNO>
    + <G_DEPTNO>
  </LIST_G_DEPTNO>
</MODULE1>

```

上のXML例では、グループG_DEPTNO（外部タグ<LIST_G_DEPTNO>で表示されている）に一意のDEPTNOデータのインスタンスが3つあることがわかります。次に、G_DEPTNOの子である次のレベルのグループ、G_JOBを見てみます。最初のXMLタグ<G_DEPTNO>を拡張すると、次のように表示されます。

```

- <MODULE1>
  - <LIST_G_DEPTNO>
    - <G_DEPTNO>
      <DEPTNO>30</DEPTNO>
    + <LIST_G_JOB>
      </G_DEPTNO>
    + <G_DEPTNO>
    + <G_DEPTNO>
      <LIST_G_DEPTNO>
    </MODULE1>

```

<LIST_G_DEPTNO>に<G_DEPTNO>の繰り返されるインスタンスが含まれるのと同じように、+<LIST_G_JOB>には<G_JOB>の繰り返されるインスタンスが含まれます。要素+<LIST_G_JOB>を拡張すると、次のXML出力がその場に表示されます。

```

- <LIST_G_JOB>
  + <G_JOB>
  + <G_JOB>
  + <G_JOB>
- </LIST_G_JOB>

```

最初のG_JOB要素(+<G_JOB>)を拡張すると、次のXML出力がその場に表示されます。

```

- <G_JOB>
  <JOB>SALESMAN</JOB>
+ <LIST_G_EMPNO>
</G_JOB>

```

<G_JOB>に、別の繰り返しグループ<G_EMPNO>のインスタンスが含まれていることがわかります。+<LIST_G_EMPNO>を拡張すると、<G_EMPNO>のインスタンスが表示されます。ここではもう子グループはありませんが、G_EMPNOで使用できる多くの列値があります。XML出力は次のようになります。

```

- <LIST_G_EMPNO>
- <G_EMPNO>
  <EMPNO>7499</EMPNO>
  <ENAME>ALLEN</ENAME>
  <MGR>7698</MGR>
  <HIREDATE>20-FEB-81</HIREDATE>
  <SAL>1600</SAL>
  <COMM>300</COMM>
</G_EMPNO>
</LIST_G_EMPNO>

```

外部の要素<LIST_[groupname]>は、繰り返しグループのコンテナとして便利です。

「外部XML属性」の例

レポートに次の問合せを使用しているとします。

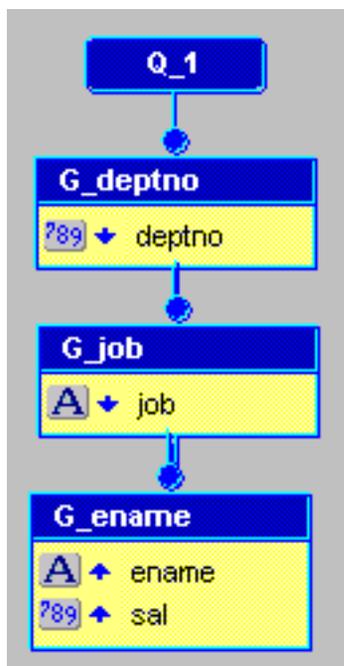
```
SELECT deptno, job, ename, sal FROM EMP
```

このデータを除外グループ・レポートとして表示することとし、DEPTNOを1番目のレベルのグループ、JOBを2番目のレベルのグループとします。

この例では、グループDEPTNOおよびJOBの出力順を昇順から降順に入れ替えるを考えます。

1. データ・モデルから列DEPTNOを選択し、プロパティ・パレットをオープンします。
2. 「ブレイク順序」プロパティを「昇順」から「降順」に変更します。
3. 1、2を列JOBについても繰り返します。

データ・モデルは次のようになっています。



ここでのXML出力は次のようになります。

```

<MODULE1>
  <LIST_G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>30</DEPTNO>
      <LIST_G_JOB>
        <G_JOB>
          <JOB>SALESMAN</JOB>
          <LIST_G_ENAME>
            <G_ENAME>
              <ENAME>ALLEN</ENAME>
              <SAL>1600</SAL>
            </G_ENAME>
            <G_ENAME>
              <ENAME>WARD</ENAME>
              <SAL>1250</SAL>
            </G_ENAME>
          </G_JOB>
        </LIST_G_JOB>
      </G_DEPTNO>
    </LIST_G_DEPTNO>
  </MODULE1>

```

この中で、<JOB>の一意の値が<G_ENAME>に少なくとも2回適用されています。具体的には、少なくとも2人の従業員（ALLENおよびWARD）がSALESMANのジョブを持っています。役職（<JOB>）を別の行の別のXMLタグにネストするかわりに、この一意の値を<LIST_G_ENAME>要素に収めます。<JOB>の値は<LIST_G_ENAME>ごとに1回発生するためです。

データ・モデルから、次のプロパティを指定します。

グループ・オブジェクト	プロパティ	アクション	値
G_ENAME	外部XML属性	入力	JOBTITLE="&JOB"
Job	XML出力から除外	選択	はい
G_JOB	XMLタグ	削除	G_JOB

これでXML出力は次のようになります。

```

<MODULE1>
  <LIST_G_DEPTNO>
    <G_DEPTNO>
      <DEPTNO>30</DEPTNO>
      <LIST_G_JOB>
        <LIST_G_ENAME JOBTITLE="SALESMAN">
          <G_ENAME>
            <ENAME>ALLEN</ENAME>
            <SAL>1600</SAL>
          </G_ENAME>
          <G_ENAME>
            <ENAME>WARD</ENAME>
            <SAL>1250</SAL>
          </G_ENAME>
        </LIST_G_ENAME>
      </LIST_G_JOB>
    </G_DEPTNO>
  </LIST_G_DEPTNO>
</MODULE1>

```

「XML Prolog値」の例 (Prolog型:テキスト)

XML出力の最初にテキストとしてprologを挿入するとします。

Report Builderでレポートをオープンした状態でオブジェクト・ナビゲータ内のレポート・アイコンを選択し、プロパティ・パレットを表示します。次のプロパティを指定します。

XML Prolog型	テキスト
-------------	------

次に、「XML Prolog値」プロパティ・フィールドを1回クリックし、 ボタンを表示します。このボタンをクリックして「XML Prolog値」ダイアログ・ボックスを表示します。prologのテキストを入力し、「OK」をクリックします。

次の例は、順にXML宣言、コメント、処理命令 (PI)、文書型宣言 (DTD) を含むprologテキストです。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is how you make comments in XML -->
<?xml:stylesheet type="text/xsl" href="emp.xsl" ?>
<!DOCTYPE EMP_TABULAR [
<!ELEMENT EMP_TABULAR (EMP_ITEM)*>
<!ELEMENT EMP_ITEM (EMPNO)>
<!ELEMENT EMPNO (#PCDATA)>
]>
```

「XML Prolog値」の例 (Prolog型:ファイル)

prologを外部のファイルに記述するとします。

Report Builderでレポートをオープンした状態で、次のプロパティをレポート・レベルで指定します。

XML Prolog型	ファイル
-------------	------

次に、「XML Prolog値」プロパティ・フィールドを1回クリックし、 ボタンを表示します。このボタンをクリックして、prologテキストが含まれている外部のファイルまでナビゲートします。

次の例は、順にXML宣言、コメント、処理命令 (PI)、文書型定義 (DTD) を含むprologテキストです。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is how you make comments in XML -->
<?xml:stylesheet type="text/xsl" href="emp.xsl" ?>
<!DOCTYPE EMP_TABULAR [
<!ELEMENT EMP_TABULAR (EMP_ITEM)*>
<!ELEMENT EMP_ITEM (EMPNO)>
<!ELEMENT EMPNO (#PCDATA)>
]>
```

「XMLタグを含む」の例

データベースにすでに列の値に含まれたXMLタグがあるとします。下の最初のインスタンスで、XMLタグ「DESTINATION」は、ユーザー指定のパラメータ&p_destを表します。「XMLタグを含む」プロパティが「はい」に設定されていると、Report Builderは<DESTINATION>をXMLタグと認識します。&p_destをユーザー指定に従って変数と解釈し、かわりにXML出力の&p_destにある一意の値を使用します。このことは、"&[report object name]"の構文を持つ列に含まれるどの値にも当てはまります。

XMLタグを持つ変数の列値:

列の元のテキスト文字列	<DESTINATION>&p_dest<DESTINATION>
「XMLタグを含む」=「はい」	<DESTINATION>FIJI<DESTINATION>
「XMLタグを含む」=「いいえ」	<DESTINATION>&p_dest<DESTINATION>

XMLタグを持つ変数でない列値:

列の元のテキスト文字列	<JOB>Clerk<JOB>
「XMLタグを含む」=「はい」	<JOB>Clerk<JOB>
「XMLタグを含む」=「いいえ」	<JOB>Clerk<JOB>

実行プログラム

実行プログラム名

実行プログラムのGUIモードが、システム固有のGUI以外のものかどうかを示すために、実行プログラム名に文字を追加します。たとえば、「a」はApple MacintoshのGUIモード、「m」はMotifのGUIモードを示します（文字が追加されていない場合は、実行プログラムのGUIモードがそのシステム固有のGUIであることを示します）。さらに、「x」は、ユーザー・イグジットがその実行プログラムにリンクされているかどうかを示すのに使用されます。

実行プログラムの起動

実行プログラムは、次の2つの方法のいずれかで起動できます。

- コマンド・ラインを使用します。
- 実行プログラムのアイコンをダブルクリックします。

実行引数 実行プログラムを起動するとき、実行プログラムの動作を制御する実行引数を入力できます。コマンド・ラインを介して実行プログラムを起動する場合、コマンド・ラインで、実行プログラム名の後に実行引数を入力します。

コマンド・ライン・オプションのヘルプ

コマンド・ラインを使用している場合、実行プログラム名の後に空白1つと疑問符1つを入力して[Return]キーを押すと、実行プログラムのコマンド・ライン・オプションを表示できます。たとえば、`rwrn60 "?"`のように入力します。

注意: 疑問符を引用符で囲む必要があるオペレーティング・システムもあります。たとえば、UNIXのCシェルでRWRUN60実行プログラムのコマンド・ライン・オプションを表示するには、`rwrn60 "?"`と入力する必要があります。

実行プログラム構文の制限

- コマンド・ラインに実行プログラムを入力する場合、一般には次のような形式になります。

```
executable_name keyword1=value1 keyword2=value2 ...
```

- オペレーティング・システムに使用可能なコマンド・ラインがなく（Apple MacintoshまたはWindowsなど）、アイコンを使用して実行プログラムを起動する場合、実行引数の値を入力できるダイアログ・ボックスが表示されるか、実行引数の指定に「作業環境」ダイアログ・ボックス（「ツール」→「作業環境」）の「ランタイム値」および「ランタイム設定」タブが使用されることが必要な場合があります。

- 各keyword=valueは、実行引数と呼ばれます。キーワードを毎回入力するのは面倒なので、省略できます。
- 引数の等号の前には、空白を入れません。
- 引数は、1つ以上の空白で区切ります。引数を区切るのにカンマを使用しないでください。
- 引数の値を一重または二重引用符で囲むことができます。一重または二重引用符の機能は、オペレーティング・システムによって異なります。
- すべての引数のkeyword=の部分は、大/小文字の区別をしません。SOURCEおよびDEST、CMDFILE、TERM、DESNAME、DESFORMAT、DNAME、SNAME引数の値の部分は、使用しているオペレーティング・システムによって、大/小文字の区別が必要な場合があります。その他すべての引数の値の部分は、大/小文字の区別をしません。
- キーワードは省略される場合があります。
- コマンド・ラインから一重引用符を渡すには、2つの引用符（1つはエスケープとして、もう1つは実際の引用符として）を入力する必要があります。次に例を示します。

```
rwrunc60 module=myrep destype=file desname=run.out batch=yes
p_value="Roy"'s Batch Report"
```

FormsのRUN_PRODUCTパッケージ・プロシージャを使用して一重引用符の付いたパラメータ値を渡すには、4つの一重引用符を入力する必要があります（PL/SQLやReport Builderでは、2つの引用符を1つと見なします）。たとえば、次のリテラル値をReportsのパラメータに渡す場合、次のように入力します。

```
where '='
```

RUN_PRODUCTでは、次の文字列を入力します。

```
'where ''''='''''
```

- 引数に値として空の文字列を指定するには、keyword=またはkeyword=""を入力します。このように空の文字列を入力すると、ランタイム・パラメータ・フォームでは、初期値のかわりに引数の空の文字列を表示します。
- すべてのファイルを参照するために、フル・パス名がサポートされています（desname=/revenues/q1/nwsalesなど）。フル・パスを指定しないと、ファイルの検索にはファイル・パス検索順序が使用されます。

注意: レポートの移植性を確実にするために、Reports固有の環境変数REPORTS60_PATHを使用して、レポート、外部問合せ、外部PL/SQLライブラリ、テキスト・ファイル、図形などを参照できます。REPORTS60_PATHを使用すると、自分のレポートにディレクトリ・パスをハードコーディングしなくても、Report BuilderまたはReports Runtimeの外部に格納されているファイルにアクセスできます。

- ランタイム・パラメータ・フォームから入力された値は、コマンド・ラインに入力された値

を上書きします。たとえば、コマンド・ラインで、RWRUN60をCOPIES=1にして指定しても、ランタイム・パラメータ・フォームでCOPIES=2と指定すると、レポートのコピーは2つ生成されます。

同様に、コマンド・ラインに入力された値は、コマンド・ファイルで指定された値を上書きします。たとえば、コマンド・ラインからRWRUN60をCOPIES=1およびCMDFILE=RUNONE（コマンド・ファイル）として指定するとします。RUNONEでは、COPIESはRWRUN60コマンドによって2に設定されています。この場合、生成されるレポートのコピーは1つのみです。

DESTYPE、DESNAME、DESFORMAT、ORIENTATIONおよびCOPIESの値は、いろいろな場所で指定できます。次のリストは、これらの値を指定する場所の優先順位を示しています。リストの最初の項目は他すべての項目よりも優先順位が高く、2番目の項目は最初の項目を除くすべての項目よりも優先順位が高い、というようになります。

- 「印刷ジョブ」ダイアログ・ボックス
- ランタイム・パラメータ・フォーム
- 「環境設定」ダイアログ・ボックスの「ランタイム値」タブおよび「ランタイム設定」タブ
- コマンド・ライン上のキーワード
- レポート定義で指定された値
- 「印刷」ダイアログ

キーワードの使用方法

各引数のkeyword=の部分は、引数が構文で指定される順で入力される限り、オプションです。引数の1つをスキップする場合は、スキップした引数の後にキーワードを使用する必要があります。

実行プログラムに4つの引数があり、1、2、4番目の引数のみを指定する場合、最初の2つはキーワードなしで指定できます。4番目の引数は、構文で指定された順で入力しないので、キーワードが必要です。

例（有効）:

次は、有効なコマンド・ライン・シーケンスの例です。

```
rwrun60 userid=scott/tiger paramform=no destype=mail  
desname=rmiller
```

```
rwrun60 scott/tiger no destype=mail desname=rmiller
```

この例では、キーワードDESTYPEを使用した後に、後続のすべての引数でもキーワードを使用していることに注意してください。一度キーワードを入力すると、コマンド・ラインの残りの引数もキーワードを使用する必要があります。

例（無効）:

次は、無効なコマンド・ライン・シーケンスの例です。

```
rwrunc60 paramform=no scott/tiger destype=mail  
desname=rmiller
```

前述のシーケンスは、scott/tigerの入力が順序どおりではなく、USERIDキーワードが使用されていないので無効です。

```
rwrunc60 scott/tiger mail rmiller no
```

前述のシーケンスは、値NOの入力が順序どおりではなく、PARAMFORMキーワードが使用されていないので無効です。

データベースへのログイン

次の方法でデータベースに接続できます。

- 明示的ログイン
- 自動ログイン
- リモート・ログイン

明示的ログイン

USERIDキーワードを使用するか、または「ファイル」→「接続」を選択して、明示的にデータベースにログインできます。

ユーザーID

説明

ユーザーIDは、オプションのデータベース名またはリモート・データベースにアクセスするためのNet8サービス名、ODBCデータ・ソース名（非Oracleデータ・ソースにアクセスする場合）を持つOracleのユーザー名およびパスワードです。

オプションのNet8サービス名の詳細は、「リモート・ログイン」を参照してください。自動ログインの詳細は、「自動ログイン」を参照してください。

値

データベース接続の文字列は次の形式のいずれかで、長さは512バイトを超えないようにしてください。

```
username[/password]
username[/password] [@database]
[user[/password]]@ODBC:datasource[:database] or
[user[/password]]@ODBC:*
```

自動ログイン

Oracleの自動ログイン機能により、明示的にユーザー名とパスワードを指定しなくてもデータベースに接続できます。

この方法で接続する場合、Oracleではデフォルトのログイン (OPSS\$など) を使用して、Oracleユーザー名を、ご使用のオペレーティング・システムのユーザー名と関連付けます。自動ログイン・プロシージャを起動するには、次のいずれか1つを実行します。

- コマンド・ラインでrwcon60 /を入力します。
- 「ログイン」画面に、ユーザー名としてスラッシュ (/) を入力します。
- コマンド・ラインでユーザー名およびパスワードを省略し、[Return]キーを2回押して「ログイン」フォームを表示します。

注意: この方法でログインする場合、スラッシュの後に@databaseを入力すると、リモート・ログインにアクセスできます。

デフォルトのログインの詳細は、『Oracle8 Server SQLリファレンス』を参照してください。

リモート・ログイン

Net8を介して、リモート・データベース (使用しているコンピュータ上にないもの) を指定できます。リモート・ログイン機能を使用すると、リモート・データベースは透過的になります。詳細は、Oracle Net8のマニュアルを参照してください。

使用上の注意

- リモート・アクセスを実行するには、ターゲット・データベースが常駐するリモート・コンピュータ・ノードのアカウントがあることが必要です。

RWBLD60

説明

RWBLD60は、Report Builderを起動し、レポートの作成およびメンテナンスを行います。

構文

```
RWBLD60 [ [keyword=]value ] ...
```

ここで、keyword=valueは、有効なコマンド・ライン引数です。

RWBLD60コマンド・ライン引数

```
[MODULE | REPORT=] modulename  
[USERID=] userid  
[PARAMFORM=] {YES | NO}  
[CMDFILE=] filename  
[ARRAYSIZE=] n  
[DESTYPE=] {SCREEN | FILE | PRINTER | PREVIEW | MAIL}  
[DESNAME=] desname  
[DESFORMAT=] desformat  
[COPIES=] n  
[CURRENCY=] currency_symbol  
[THOUSANDS=] thousands_symbol  
[DECIMAL=] decimal_symbol  
[READONLY=] {YES | NO}  
[BUFFERS=] n  
[PAGE SIZE=] width x height  
[PROFILE=] profiler_file  
[RUNDEBUG=] {YES | NO}
```

```
[ONSUCCESS=] { COMMIT | ROLLBACK | NOACTION }
[ONFAILURE=] { COMMIT | ROLLBACK | NOACTION }
[ERRFILE=] error_file
[LONGCHUNK=] n
[ACCESS=] { file | database }
[ORIENTATION=] { DEFAULT | LANDSCAPE | PORTRAIT }
[BACKGROUND=] { YES | NO }
[MODE=] { BITMAP | CHARACTER | DEFAULT }
[PRINTJOB] { YES | NO }
[TRACEFILE=] tracefile
[TRACEMODE=] { TRACE_APPEND | TRACE_REPLACE }
[TRACEOPTS=] { TRACE_ERR | TRACE_PRF | TRACE_BRK | TRACE_APP |
TRACE_PLS | TRACE_SQL | TRACE_TMS | TRACE_DST | TRACE_ALL | (opt1, opt2, ...) }
[AUTOCOMMIT=] { YES | NO }
[NONBLOCKSQL=] { YES | NO }
[ROLE=] rolename / [rolepassword]
[BLANKPAGES=] { YES | NO }
[MAXIMIZE=] { YES | NO }
[DISABLEPRINT=] { YES | NO }
[DISABLEMAIL=] { YES | NO }
[DISABLEFILE=] { YES | NO }
[DISABLENEW=] { YES | NO }
[DISTRIBUTE=] { YES | NO }
[DESTINATION=] filename.DST
[CACHELOB=] { YES | NO }
[DELIMITER=] value
[CELLWRAPPER=] value
```

```
[NUMBERFORMATMASK=] mask  
  
[DATEFORMATMASK=] mask  
  
[PAGESTREAM=] {YES | NO}  
  
[EXPRESS_SERVER=] "server=[server]/domain=[domain]/user=[userid]/password=[passwd] "  
  
[CUSTOMIZE]=filename.xml | (filename1.xml, filename2.xml, . . .)  
  
[SAVE_RDF]=filename.rdf  
  
<param>=value
```

MODULE|REPORT

説明

MODULE|REPORTは、Report Builderでオープンするモジュール（レポートまたは外部問合せ、外部PL/SQLライブラリ）です。（REPORTは、下位互換性のためのものです。）MODULE|REPORTを使用してレポートを指定しないと、Report Builderは、新規レポートの定義をオープンして起動します。

構文

```
[MODULE|REPORT=] runfile
```

値

任意の有効なレポートまたはXMLレポート定義(XMLファイル)、外部問合せ、外部PL/SQLライブラリ名です。Report Builderは、モジュールのACCESSキーワードで指定した場所を検索します。ACCESSを指定しないと、Report Builderではファイル・パス検索順序を使用します。モジュールが見つからないと、Report Builderは指定したモジュールをオープンしないで起動します。

使用上の注意

- ビットマップのRWBLD60を介してキャラクタ・モードのレポートをオープンする場合、Report Builderでは警告を表示してから、レポートを8.5 x 11インチのページ・サイズと7 x 6インチのフォーム・サイズを使用してオープンします。
- レポートを実行（たとえば、ライブ・プレビューアで表示）するには、レポート定義は完全に行う必要があります（それ自身のデータ定義およびレイアウト定義が含まれていることなど）。レポート定義の一部のみを実行することはできません。
- MODULE|REPORTキーワードを指定した場合、XMLレポート定義はXMLファイルの拡張子が指定されている必要があります。

PARAMFORM

説明

PARAMFORMは、レポートの実行時にランタイム・パラメータ・フォームを表示するかどうかを指定します。

構文

```
[PARAMFORM=] {YES|NO}
```

値

YESの場合、フォームを表示します。

NOの場合、フォームを非表示にします。

デフォルト

YES

CMDFILE

説明

CMDFILEは、RWBLD60コマンドの引数を含むファイルです。このオプションを使用すると、RWBLD60を起動するたびに多くの引数を指定しなくても、レポートを実行できます。

構文

```
[CMDFILE=] cmdfile
```

値

任意の有効なコマンド・ファイルです。

使用上の注意

- コマンド・ファイルは、別のコマンド・ファイルを参照できます。
- RWBLD60引数のコマンド・ファイル構文は、コマンド・ラインで使用される構文と同じです。
- コマンド・ラインに入力した値は、コマンド・ファイルで指定した値を上書きします。たとえば、コマンド・ラインからCOPIES = 1およびCMDFILE = RUNONE (コマンド・ファイル)としてRWBLD60を指定するとします。RUNONEでは、COPIESは2に設定されています。この場合、生成されるレポートのコピーは1つのみです。詳細は、実行可能プログラム構文の制限を参照してください。

- このキーワードの引数は、オペレーティング・システム固有の場合があります。

ARRAYSIZE

説明

ARRAYSIZEは、Oracle配列処理用の配列サイズで、キロバイトで指定します。

構文

```
[ARRAYSIZE=]n
```

値

1から9,999までの数値です。つまり、Report Builderでは、レポートの問合せ1つに対してこのキロバイト数のメモリーを使用できます。

デフォルト

デフォルトの配列サイズは10です。Oracleの配列処理の詳細は、『Oracle8 Server管理者ガイド』を参照してください。

DESTYPE

説明

DESTYPEは、レポート出力を受け取る宛先のデバイスのタイプです。

構文

```
[DESTYPE=] { SCREEN | FILE | PRINTER | PREVIEW | MAIL }
```

値

SCREEN	出力をプレビューアに送ります。
FILE	出力をDESTNAMEで指定したファイルに保存します。
PRINTER	出力をDESTNAMEで指定したプリンタに送ります。
PREVIEW	出力は対話形式で表示するためにプレビューアに送られます。ただし、Previewを指定すると、出力はPostScript出力としてフォーマットされます。プレビューアではDESTNAMEを使用して、出力の表示にどのプリンタのフォントを使用するかを判断します。
MAIL	出力をDESTNAMEで指定したメールIDに送ります。MAPI準拠またはサービス・プロバイダ・ドライバがインストールされているメール・システムであれば、メールを送ることができます。レポートは、添付ファイルとして送られます。

デフォルト

DESTYPEパラメータの初期値プロパティの設定値です。

使用上の注意

- SYSOUTは、RWBLD60の使用中は、DESTYPEの有効なオプションではありません。SYSOUTが使用するのは、BATCH=YESのときのRWRUN60のみです。
- DESTYPEをPRINTERまたはFILE、MAILに設定すると、非表示に指定していなければ、ランタイム・パラメータ・フォームを確定した後に、「印刷ジョブ」ダイアログ・ボックスが表示されます。「印刷ジョブ」ダイアログ・ボックスを適用すると、レポートが実行され、出力がプリンタ（「ファイル」→「プリンタ選択」で指定されている）またはファイル、メールIDに送られます。）オペレーティング・システムによって、ダイアログ・ボックスの外観は異なります。
- このパラメータは、使用しているオペレーティング・システムによって上書きされる場合があります。
- オペレーティング・システムによっては、一部のDESTYPE値がサポートされないことがあります。

DESNAME

説明

DESNAMEは、レポート出力の送り先のファイル、プリンタ、あるいは電子メールID（または配布リスト）の名前です。電子メールでレポート出力を送るには、電子メール・アプリケーション（Windows上のMAPI準拠のアプリケーションや、UNIX上の固有のメール・アプリケーションなど）の場合と同様に電子メールIDを指定します。名前をカンマで区切ってカッコで囲むことによって、複数のユーザーを指定できます（(name, name, ...name)など）。

構文

```
[DESNAME=] desname
```

値

長さが1Kを超えない任意の有効なファイル名またはプリンタ名、電子メールIDです。プリンタ名には、任意にポートを指定できます。次に例を示します。

```
DESNAME=printer,LPT1:
```

```
DESNAME=printer,FILE:
```

デフォルト

DESNAMEパラメータの初期値プロパティの設定値です。DESTYPE=FILEとDESNAMEが指定されていない場合、実行時にデフォルトで<reportname>.lisになります。

Usage Notes

- DESTYPEがScreenの場合、このキーワードは無視されます。
- DESTYPEがPreviewの場合、Report BuilderではDESNAMEを使用して、出力の表示にどのプリンタのフォントを使用するかを判断します。
- このキーワードの引数は、使用しているオペレーティング・システムによっては、大/小文字の区別が必要な場合があります。
- このパラメータは、使用しているオペレーティング・システムによって上書きされる場合があります。

DESFORMAT

説明

DESFORMATは、ビットマップ環境では、DESTYPEがFileの場合に使用するプリンタ・ドライバを指定します。キャラクタ・モード環境では、DESNAMEで指定したプリンタの特性を指定します。

構文

[DESFORMAT=] desformat

値

長さが1Kを超えない任意の有効な宛先フォーマットです。このキーワードの有効値の例として、hpl、hplwide、dec、decwide、decland、dec180、dflt、wideなどがあります。有効な宛先フォーマットのリストについては、システム管理者に問い合せてください。

PDF	レポート出力は、PDF viewerで読み込むことのできるファイルに送られます。PDF出力は、ご使用のシステム用に現在構成されているプリンタをベースにしています。現在選択されているプリンタのドライバが、出力の作成に使用されます。レポートを実行しているマシン用に、プリンタを構成する必要があります。
HTML	レポート出力は、HTML3.0準拠のブラウザ (Netscape 2.2など) で読み込むことのできるファイルに送られます。
HTMLCSS	ファイルに送られるレポート出力には、カスケード・スタイル・シートをサポートするHTML3.0準拠のブラウザで読み込むことのできるスタイル・シート拡張が含まれます。
HTMLCSSIE	ファイルに送られるレポート出力には、Microsoft Internet Explorer 3.xで読み込むことのできるスタイル・シート拡張が含まれます。

RTF	レポート出力は、標準のワード・プロセッサ (Microsoft Wordなど) で読み込むことのできるファイルに送られます。MS-WORDでファイルをオープンする場合、レポート内のすべての図形およびオブジェクトを表示するには、「表示」→「ページ・レイアウト」を選択する必要があります。
DELIMITED	レポート出力は、標準のスプレッドシート・ユーティリティ (Microsoft Excelなど) で読み込むことのできるファイルに送られます。デリミタを選択しないと、デフォルトのデリミタはタブになることに注意してください。
XML	レポート出力は、第5世代以上のWebブラウザを含む任意のXMLをサポートするアプリケーションで読み込むことのできるファイルに送られます。

デフォルト

DESFORMATパラメータの初期値プロパティの設定値です。ビットマップReport Builderでは、DESFORMATがブランクまたはデフォルトの場合、現行のプリンタ・ドライバ(「ファイル」→「プリンタの選択」で指定)が使用されます。「プリンタの選択」で何も選択していない場合、デフォルトでPostScriptが使用されます。

使用上の注意

- DESTYPEがScreenの場合、このキーワードは無視されます。
- このキーワードの値は、使用しているオペレーティング・システムによっては、大/小文字の区別が必要な場合があります。

COPIES

説明

COPIESは、印刷するレポート出力の部数です。

構文

[COPIES=] n

値

1 ~ 9,999までの任意の有効な整数です。

デフォルト

COPIESパラメータの初期値プロパティの設定値です。

使用上の注意

- DESTYPEがPrinterでない場合、このキーワードは無視されます。

- COPIESがランタイム・パラメータ・フォームで指定されていない場合、デフォルトで1に設定されます。

CACHELOB

説明

CACHELOBは、抽出されたOracle8ラージ・オブジェクトを一時ファイル・ディレクトリ（REPORTS60_TMPによって指定）内にキャッシュとして格納するかどうかを指定します。

値

YESの場合、LOBを一時ファイル・ディレクトリにキャッシュします。

NOの場合、LOBを一時ファイル・ディレクトリにキャッシュしません。

デフォルト

YES

使用上の注意

- このオプションは、コマンド・ラインからのみ設定できます。
- 一時ファイル・ディレクトリの場所に十分な空きディスク領域がない場合は、この値をNOに設定してください。ただし、値をNOに設定すると、サーバーから何度もLOBをフェッチすることになるため、パフォーマンスが低下する場合があります。

CURRENCY

説明

CURRENCYは、数値書式に使用される通貨記号です。

構文

```
[CURRENCY=]currency_symbol
```

値

長さが1Kを超えない任意の有効な英数字文字列です。

デフォルト

Oracleのデフォルトは、Oracle各国語サポート機能によって決まります。CURRENCYパラメータの初期値プロパティに、4文字までのデフォルトを設定することもできます。

使用上の注意

- プロパティ・パレットに入力されたCURRENCY値は、コマンド・ラインに入力されたCURRENCY値を上書きします。

THOUSANDS

説明

THOUSANDSは、数値書式に使用される3桁区切り記号です。

構文

```
[THOUSANDS=] thousands_symbol
```

値

任意の有効な英数字です。

デフォルト

Oracleのデフォルトは、Oracle各国語サポート機能によって決まります。THOUSANDSパラメータの初期値プロパティに、4文字までのデフォルトを設定することもできます。

使用上の注意

- パラメータ・プロパティ・シートに入力されたTHOUSANDS値は、コマンド・ラインに入力されたTHOUSANDS値を上書きします。
- THOUSANDS値として定義した英数字が、実際に返される値になります。たとえば、THOUSANDS値として "," を定義した場合は、"," が返されます。

DECIMAL

説明

DECIMALは、数値書式に使用される10進記号（インジケータ）です。

構文

```
[DECIMAL=] decimal_symbol
```

値

任意の有効な英数字です。

デフォルト

Oracleのデフォルトは、Oracle各国語サポート機能によって決まります。THOUSANDSパラメータの初期値プロパティに、デフォルトを設定することもできます。

使用上の注意

- パラメータ・プロパティ・シートに入力されたDECIMAL値は、コマンド・ラインに入力されたDECIMAL値を上書きします。
- DECIMAL値として定義した英数字が、実際に返される値になります。たとえば、DECIMAL値として "." を定義した場合は、 "." が返されます。

READONLY

説明

READONLYは、レポートの複数の問合せにおける読み一貫性を要求します。Oracleからデータにアクセスする場合、読み一貫性はSET TRANSACTION READ ONLY文によって達成されません (SET TRANSACTION READ ONLYの詳細は、『Oracle8 Server SQLリファレンス』を参照)。

構文

```
[READONLY=] {YES|NO}
```

値

YESの場合、読みの一貫性を要求します。

NOの場合、読みの一貫性は要求されません。

デフォルト

NO

使用上の注意

- Oracleでは、問合せが1つのレポートには、ロックせずに、読み一貫性を自動的に提供するので、このキーワードが役に立つのは、複数の問合せを使用するレポートのみです。

READONLYの制限

- レポート・トリガーの実行順序で、SET TRANSACTION READONLYがどこで発生するかに注

意してください。

BUFFERS

説明

BUFFERSは、仮想メモリー・キャッシュのサイズで、キロバイトで指定します。レポートの実行に十分な領域を確保する一方で、システムのリソースを使用し過ぎないように調整する必要があります。

構文

```
[BUFFERS=] n
```

値

1から9,999までの数値です。オペレーティング・システムによっては、上限がこれよりも低い場合があります。

デフォルト

640K

使用上の注意

- この設定がセッションの途中で変更されても、変更は次のレポートが実行されるまで有効になりません。

PAGESIZE

説明

PAGESIZEは、物理ページのサイズ（つまり、プリンタが出力するページのサイズ）です。ページには、レポートの内容が収まるだけの大きさが必要であることに注意してください。たとえば、レポートの枠サイズがページのサイズよりも大きくなると、レポートは実行されません。

構文

```
[PAGESIZE=]width x height
```

値

「ページ幅×ページの高さ」の形式で表される任意の有効なページ・サイズで、ページ幅およびページの高さは0（ゼロ）以上です。最大幅および高さは、サイズの単位によって異なります。

インチでは、最大幅および高さは512インチです。センチメートルでは、1,312センチメートルです。パイカの場合、36,864パイカです。

デフォルト

ビットマップでは、8.5 x 11インチです。キャラクタ・モードでは、80 x 66文字です。レポートがキャラクタ・モードに設計されており、ビットマップ上で実行または変換中の場合、何も指定されていないと、ページ・サイズを判断するのに次の計算式が使用されます。(デフォルトのページ・サイズ * キャラクタ・ページ・サイズ) / デフォルトのキャラクタ・ページ・サイズ。たとえば、キャラクタ・ページ・サイズが80 x 20の場合、ビットマップ・ページ・サイズは、 $(8.5 * 80) / 80 \times (11 * 20) / 66 = 8.5 \times 3.33$ となります。

使用上の注意

- 物理ページの印刷可能領域が制限されているプリンタもあります。たとえば、プリンタ用紙は8.5 x 11インチでも、プリンタで印刷できる範囲が8 x 10.5インチしかない場合があります。Report Builderで、プリンタの印刷可能領域よりも大きいページ幅 x ページの高さを定義すると、レポート出力でクリッピングが発生する可能性があります。クリッピングを防ぐには、プリンタの印刷可能領域を拡大するか（オペレーティング・システムで可能な場合）、ページ幅 x ページの高さをページの印刷可能な領域のサイズに設定します。
- このキーワードを使用すると、その値はレポート定義のページのサイズを上書きします。
- ランタイム・パラメータ・フォームに入力されたPAGESIZE値は、コマンド・ラインに入力されたPAGESIZE値を上書きします。

PROFILE

説明

PROFILEは、レポートの実行に関するパフォーマンスの統計を格納するファイルの名前です。ファイル名を指定すると、Report Builderでは、レポートの実行にかかった経過時間およびCPU時間の統計を計算します。PROFILEは、次の統計を計算します。

- **合計経過時間**は、RWBLD60を発行してからBuilderを終了するまでに経過した時間です。合計経過時間は、Report Builder時間とOracle時間の合計です。
- **時間**は、Report Builderでかかった時間です。
- **Oracle時間**は、データベースでかかった時間で、次のものから成り立っています。
 - **UPI**は、データベースへの接続およびSQLの解析、データのフェッチなどにかかった時間。
 - **SQL**は、SRW.DO_SQLの実行にかかった時間。
- **処理に使用した合計CPU時間**は、Builderの実行にかかったCPU時間。

注意: オペレーティング・システムには、データベースがReport Builderのプロセスに含まれているために、Report Builder時間にデータベース時間が含まれるものもあります。

構文

```
[PROFILE=]profiler_file
```

値

現行のディレクトリにある任意の有効なファイル名です。

RUNDEBUG

説明

RUNDEBUGは、レポートの論理エラーのチェックを行うかどうかを指定します。RUNDEBUGは、エラーではないが、望ましくない出力結果になる可能性があるものをチェックします。RUNDEBUGは、次のものをチェックします。

- 別のオブジェクトとオーバーラップしているが、囲んではいない枠または繰返し枠。これにより、オブジェクトが出力で他のオブジェクトを上書きする可能性があります。
- 固定のサイズ設定ではない、ページ依存の参照元を持つレイアウト・オブジェクト。Report Builderでは、そのようなオブジェクトを、「垂直拡張度」および「水平拡張度」プロパティにかかわらず、固定サイズにします。
- 誤った頻度で参照されるPL/SQLのバインド変数。

構文

```
[RUNDEBUG=] {YES|NO}
```

値

YESの場合、追加のランタイム・エラーのチェックを実行します。

NOの場合、追加のランタイム・エラーのチェックを実行しません。

デフォルト

YES

ONSUCCESS

説明

ONSUCCESSは、レポートの実行終了時に、COMMITまたはROLLBACKのどちらを実行するかを指定します。

構文

```
[ONSUCCESS=] { COMMIT | ROLLBACK | NOACTION }
```

値

COMMITの場合、レポートが終了するとCOMMITを実行します。

ROLLBACKの場合、レポートが終了するとROLLBACKを実行します。

NOACTIONの場合、レポートが終了しても何も行われません。

デフォルト

ユーザーIDが指定されている場合はCOMMITです。

ユーザーIDの指定がなく、外部ソース（Oracle Formsなど）からコールされる場合はNOACTIONです。

使用上の注意

- ONSUCCESSのCOMMITまたはROLLBACKは、「After Reportトリガー」の起動後に実行されます。その他のCOMMITおよびROLLBACKは、これより前に発生する場合があります。詳細は、「READONLY」を参照してください。

ONFAILURE

説明

ONFAILUREは、エラーが発生してレポートが終了できない場合に、COMMITまたはROLLBACKのどちらを実行するかを指定します。

構文

```
[ONFAILURE=] { COMMIT | ROLLBACK | NOACTION }
```

値

COMMITの場合、レポートが失敗するとCOMMITを実行します。

ROLLBACKの場合、レポートが失敗するとROLLBACKを実行します。

NOACTIONでは、レポートが失敗しても何も行いません。

デフォルト

ユーザーIDが指定されている場合は、ROLLBACKです。

ユーザーIDの指定がなく、外部ソース（Oracle Formsなど）からコールされる場合はNOACTIONです。

使用上の注意

- ONFAILUREのCOMMITまたはROLLBACKは、レポートが失敗した後に実行されます。その他のCOMMITおよびROLLBACKは、これより前に発生する場合があります。詳細は、「READONLY」を参照してください。

ERRFILE

説明

ERRFILEは、Report Builderでエラー・メッセージを格納するファイルの名前です。

構文

```
[ERRFILE=]error_file
```

値

任意の有効なファイル名です。

LONGCHUNK

説明

LONGCHUNKは、Report BuilderでLONG列値を取り出す変更分のサイズ（キロバイト）です。LONG値を取り出す場合、メモリのサイズに制限があるので、すべてを一度に取り出さずに、変更分に取り出せます。LONGCHUNKは、Oracle7およびOracle8にのみ適用されます。

構文

```
[LONGCHUNK=]n
```

値

1から9,999までの数値です。オペレーティング・システムによっては、上限がこれよりも低い場合があります。

デフォルト

10K

ACCESS

説明

ACCESSは、モジュールをオープンする位置です。Report Builderでは、MODULEで指定したモジュールがこの位置で検索されます。ACCESSを指定しないと、Report Builderではファイル・パス検索順序を使用して、ファイルを検索します。

構文

```
[ACCESS=] { FILE | DATABASE }
```

値

FILE

database

デフォルト

FILE

ORIENTATION

説明

ORIENTATIONは、レポートのページを印刷する方向を制御します。

構文

```
[ORIENTATION=] { DEFAULT | LANDSCAPE | PORTRAIT }
```

値

DEFAULTの場合、現行のプリンタの方向設定を使用します。

LANDSCAPEは横方向です。

PORTRAITは縦方向です。

デフォルト

DEFAULT

使用上の注意

- キャラクタ・モードのレポートに、ORIENTATION=LANDSCAPEを設定する場合、プリンタ定義ファイルにLANDSCAPE句が含まれていることを確認する必要があります。
- レポートをMotif版のPCLプリンタに出力する場合、印刷方向の設定は無視されます。レポートは、デフォルトの設定を使用して処理されます。

BACKGROUND

説明

BACKGROUNDは、レポートをバックグラウンドで実行するかどうかを指定します。

構文

```
[BACKGROUND=] {YES|NO}
```

値

YESは、RWBLD60の有効なオプションではありません。

NO

デフォルト

NO

MODE

説明

MODEは、レポートをキャラクタ・モードまたはビットマップのどちらで実行するかを指定します。これによって、キャラクタ・モードのレポートをビットマップのReport Builderから、またはビットマップのレポートをキャラクタ・モードのReport Builderから実行できます。たとえば、端末 (vt220 など) からPostScriptプリンタにレポートを送る場合、キャラクタ・モードのRWRUN60を起動し、MODE=BITMAPを指定してレポートを実行できます。Windowsでは、MODE=CHARACTERを指定すると、Report Builder ASCIIドライバが編集可能なASCII出力の作成に使用されます。

構文

[MODE=] {BITMAP|CHARACTER|DEFAULT}

値

BITMAP

DEFAULTの場合、使用されている現行の実行プログラムのモードでレポートを実行します。

CHARACTER

デフォルト

DEFAULT

PRINTJOB

説明

PRINTJOBは、「印刷」ダイアログ・ボックスをレポートの実行前に表示するかどうかを指定します。

構文

[PRINTJOB=] {YES|NO}

値

YESまたはNO

デフォルト

YES

使用上の注意

- レポートが、発生したプロセスとして実行される（つまり、RWRUN60などの実行プログラムが、RWBLD60などの別の実行プログラムからコールされる）場合、PRINTJOBの指定にかかわらず、「印刷」ダイアログ・ボックスは表示されません。
- DESTYPE=MAILの場合、PRINTJOBの指定にかかわらず、「印刷」ダイアログ・ボックスは表示されません。

TRACEFILE

説明

TRACEFILEは、Report Builderがトレース情報のログをとるファイルの名前です。

構文

```
[TRACEFILE=] tracefile
```

値

任意の有効なファイル名です。

使用上の注意

- トレース情報は、.rdfファイルの実行時にのみ生成されます。.repファイルの実行時に、ロギングの指定はできません。
- TRACEFILEのみでなくLOGFILEまたはERRFILEも指定する場合、すべてのトレース情報は、最後に指定したファイルに保存されます。たとえば、次の場合、err.logがRWRUN60コマンドで指定した最後のファイルなので、指定したトレース情報はすべてこのファイルに保存されます。

```
RWRUN60 MODULE=order_entry  
USERID=scott/tiger  
TRACEFILE=trace.log LOGFILE=mylog.log  
ERRFILE=err.log
```

TRACEMODE

説明

TRACEMODEは、Report Builderでトレース情報をファイルに追加するかまたはファイル全体を上書きするかを示します。

構文

```
[TRACEMODE=] {TRACE_APPEND|TRACE_REPLACE}
```

値

TRACE_APPENDは、新しい情報をファイルの終わりに追加します。

TRACE_REPLACEは、ファイルを上書きします。

デフォルト

TRACE_APPEND

使用上の注意

- トレース情報は、.rdfファイルの実行時にのみ生成されます。.repファイルの実行時に、ログインの指定はできません。

TRACEOPTS

説明

TRACEOPTSは、レポートの実行時にトレース・ファイルにログをとるトレース情報を示します。

構文

```
[TRACEOPTS=] {TRACE_ERR|TRACE_PRF|TRACE_APP|TRACE_PLS|  
TRACE_SQL|TRACE_TMS|TRACE_DST|TRACE_ALL|(opt1, opt2, ...)}
```

値

カッコ内のオプションのリストは、カッコ内のオプションをすべて使用するということです。たとえば、TRACE_OPTS=(TRACE_APP, TRACE_PRF)は、TRACE_APPおよびTRACE_PRFを適用するということです。

TRACE_ALLの場合、トレース・ファイルに、考えられるすべてのトレース情報のログをとります。

TRACE_APPの場合、トレース・ファイルに、すべてのレポート・オブジェクトのトレース情報のログをとります。

TRACE_BRKの場合、トレース・ファイルに、ブレーク・ポイントをリストします。

TRACE_DSTの場合、トレース・ファイルに、配布リストをリストします。この情報を使用して、どのセクションがどの宛先に送られたかを調べることができます。トレース・ファイル・フォーマットは、.DSTファイル・フォーマットと非常に似ているため、カット・アンド・ペーストによってトレース・ファイルから.DSTファイルを生成できます。

TRACE_ERRの場合、トレース・ファイルに、エラー・メッセージと警告をリストします。

TRACE_PLSの場合、トレース・ファイルに、すべてのPL/SQLオブジェクトのトレース情報のログをとります。

TRACE_PRFの場合、トレース・ファイルに、パフォーマンス統計のログをとります。

TRACE_SQLの場合、トレース・ファイルに、トレース情報のログをとります。

TRACE_TMSの場合、トレース・ファイルに、各エントリのタイムスタンプを入力します。

デフォルト

TRACE_ALL

使用上の注意

- トレース情報は、.rdfファイルの実行時にのみ生成されます。.repファイルの実行時に、ロギングの指定はできません。

AUTOCOMMIT

説明

データベースの変更 (CREATEなど) をデータベースに自動的にコミットするかどうかを指定します。一部の非Oracleデータベース (SQL Serverなど) では、AUTOCOMMIT=YESを指定する必要がありますことに注意してください。

構文

```
[AUTOCOMMIT=] {YES|NO}
```

値

YESまたはNO

デフォルト

NO

NONBLOCKSQL

説明

Report Builderでデータベースからデータをフェッチしている間に、他のプログラムを実行できるかどうかを指定します。

構文

```
[NONBLOCKSQL=] {YES|NO}
```

値

YESの場合、データがフェッチされている間、他のプログラムを実行できます。

NOの場合、データがフェッチされている間、他のプログラムを実行できません。

デフォルト

YES

ROLE

説明

ROLEは、実行時にそのレポートについて確認されるデータベースのロールを指定します。RWBLD60では、ROLEは無視されます。

構文

```
[ROLE=] {rolename/[rolepassword]}
```

値

有効なロールおよびロールのパスワード（オプション）です。

BLANKPAGES

説明

BLANKPAGESは、レポートを印刷する際にブランク・ページを抑止するかどうかを指定します。このキーワードは、レポート出力時に、ブランク・ページを印刷しない場合に使用してください。

構文

```
[BLANKPAGES=] {YES|NO}
```

値

YESは、すべてのブランク・ページを印刷します。

NOはブランク・ページを印刷しません。

Default

YES

使用上の注意

BLANKPAGESは、論理ページが複数の物理ページ（またはパネル）にまたがっていて、すべての物理的なブランク・ページを印刷しない場合に特に便利です。

MAXIMIZE

説明

MAXIMIZEは、Reports Runtimeのウィンドウを最大化するかどうかを指定します。このキーワードによるReport Builderへの影響はなく、Reports Runtimeでのみ使用されます。

DISABLEPRINT

説明

DISABLEPRINTは、「ファイル」→「印刷」、「ファイル」→「ページ設定」、(Motif上での)「ファイル」→「プリンタ選択」およびランタイム・プレビューア内の同等のツールバー・ボタンを無効にするかどうかを指定します。

構文

[DISABLEPRINT=]{YES|NO}

値

YESまたはNO

Default

NO

DISABLEMAIL

説明

DISABLEMAILは、ランタイム・プレビューア内の「メール」メニューおよび同等のツールバー・ボタンを無効にするかどうかを指定します。

構文

[DISABLEMAIL=]{YES|NO}

値

YESまたはNO

Default

NO

DISABLEFILE

説明

DISABLEFILEは、ランタイム・プレビューア内の「ファイル」→「ファイル形式に出力」メニューを無効にするかどうかを指定します。

構文

[DISABLEFILE=]{YES|NO}

値

YESまたはNO

Default

NO

DISABLENEW

説明

DISABLENEWは、ランタイム・プレビューアの新規インスタンスを表示する機能を抑止するために「表示」→「新規プレビューア」メニューを無効にするかどうかを指定します。

構文

[DISABLENEW=]{YES|NO}

値

YESまたはNO

Default

NO

DELIMITER

説明

DELIMITERは、レポート出力のセルを区切るために使用する文字（列）を指定します。

構文

[DELIMITER=]value

値

次のような任意の英数字または英数字の文字列を使用します。

,	各セルをカンマで区切ります。
.	各セルをピリオドで区切ります。

次の4つの予約済みの値を使用することもできます。

tab	各セルをタブで区切ります。
space	各セルをスペースで区切ります。
return	各セルを改行で区切ります。
none	デリミタを使用しません。

ASCIIキャラクタ・セットに基づくエスケープ・シーケンスを使用することもできます。

¥t	各セルをタブで区切ります。
¥n	各セルを改行で区切ります。

デフォルト

[Tab]

使用上の注意

- この引数はDESFORMAT=DELIMITEDを指定した場合にのみ使用できます。

CELLWRAPPER

説明

CELLWRAPPERは、レポート出力の区切りセルの周囲に表示される文字（列）を指定します。

構文

[CELLWRAPPER=]value

Value

任意の英数字または英数字の文字列を使用します。

"	セルの両側に二重引用符が表示されます。
'	セルの両側に一重引用符が表示されます。

次の4つの予約済みの値を使用することもできます。

tab	セルの両側にタブが表示されます。
space	セルの両側にシングル・スペースが表示されます。
return	セルの両側に改行が表示されます。
none	セルの囲みを使用しません。

ASCIIキャラクタ・セットに基づくエスケープ・シーケンスを使用することもできます。

¥t	セルの両側にタブが表示されます。
¥n	セルの両側に改行が表示されます。

デフォルト

[none]

使用上の注意

- この引数はDESFORMAT=DELIMITEDを指定した場合にのみ使用できます。
- セルの囲みは、実際のデリミタとは異なります。

DATEFORMATMASK

説明

DATEFORMATMASKは、デリミタ付きレポート出力の日付値の表示方法を指定します。

構文

[DATEFORMATMASK=] mask

値

任意の有効な日付書式マスクです。

使用上の注意

- この引数は、DESFORMAT=DELIMITEDを指定した場合にのみ使用できます。

NUMBERFORMATMASK

説明

NUMBERFORMATMASKは、デリミタ付きレポート出力の数値の表示方法を指定します。

構文

[NUMBERFORMATMASK=] mask

値

任意の有効な数値書式マスクです。

使用上の注意

- この引数は、DESFORMAT=DELIMITEDを指定した場合にのみ使用できます。

DESTINATION

説明

DESTINATIONキーワードを使用すると、現行の実行レポートの配布を定義する.DSTファイルの名前を指定できます。

構文

[DESTINATION=] filename.DST

値

.DSTファイルの名前です。

使用上の注意

- DESTINATIONキーワードを使用可能にするには、コマンド・ラインからDISTRIBUTE=YESを指定する必要があります。

DISTRIBUTE

説明

DISTRIBUTEは、「配布」ダイアログ・ボックスまたは.DSTファイルで定義された配布リストに従って、レポート出力を複数の宛先に配布することを許可または禁止します。

構文

[DISTRIBUTE=] {YES|NO}

値

YESの場合、配布リストにレポートを配布します。

NOの場合、配布リストを無視して、DESNAMEおよびDESFORMATパラメータの指定に従ってレポートを出力します。これは基本的に、配布用のレポート・セットアップを実行するデバッグ・モードで、配布を実際には実行しません。

Default

NO

使用上の注意

- DESTINATIONキーワードを使用可能にするには、DISTRIBUTE=YESを指定する必要があります。

PAGESTREAM

説明

PAGESTREAMは、レポートをHTML出力またはHTMLCSS出力として書式設定するときに、レポートのページ・ストリームを許可または禁止します。レポートをHTML出力またはHTMLCSS出力として書式設定する場合、次のいずれかによって設定されるナビゲーション・コントロールを使用します。

- レポート・プロパティ・パレットの「ページ・ナビゲーション・コントロール・タイプ」プロパティおよび「ページ・ナビゲーション・コントロール値」プロパティ
- Before Reportトリガー(SRW.SET_PAGE_NAVIGATION_HTML)内のPL/SQL

構文

```
[PAGESTREAM=] { YES | NO }
```

値

YESの場合、ページはストリーム処理されます。

NOの場合、ページ・ストリームなしでレポートが出力されます。

デフォルト

NO

EXPRESS_SERVER

説明

EXPRESS_SERVERは、接続するOLAP Serverを指定します。

構文

```
EXPRESS_SERVER="server=[server]/domain=[domain]/
user=[userid]/password=[passwd]"
```

RAMでの構文

```
EXPRESS_SERVER="server=[server]/domain=[domain]/
user=[userid]/password=[passwd]/ramuser=[ramuserid]/
rampassword=[rampasswd]/ramexpressid=[ramexpid]/
ramserverscript=[ramsscript]/rammasterdb=[ramdb]/
ramconnecttype=[ramconn]"
```

値

次の有効な接続文字列を二重引用符 (") で囲みます。

server	OLAP Server文字列 (ncacn_ip_tcp:olap2-pc/si=x/st=x/ct=x/sv=x/ など)。サーバー文字列の詳細は次を参照してください。
domain	OLAP Serverのドメイン。
user	OLAP ServerにログインするためのユーザーID。
password	ユーザーIDに対応するパスワード。
ramuser	RDBMSにログインするためのユーザーID。
rampassword	RDBMSに対応するパスワード。
ramexpressid	Oracle Sales AnalyzerデータベースのユーザーID。これはOracle Sales Analyzerデータベースに対してのみ必要となります。
ramserverscript	サーバー上のリモート・データベース構成ファイル (RDC) の完全なファイル名 (フル・パスを含む)。このファイルではコードおよびデータベースの位置などの情報を指定します。UNC (Universal Naming Convention) 構文を使用することにより、複数のユーザーが同一の接続を使用してデータにアクセスすることができ、データの位置に対して同じドライブ文字をマップする必要がありません。UNC 構文のパスは、 <code>¥¥ServerName¥ShareName¥</code> の後に任意のサブフォルダまたはファイル、あるいはその両方が続きます。
rammasterdb	最初に接続するRelational Access Managerデータベースの名前。データベースファイル名のみを指定する必要があります。このデータベースは、OLAP Serverに対するServerDBPathのパス・リストにインクルードされるディレクトリに常駐している必要があります。ServerDBPathは、「Express Configuration Manager」ダイアログ・ボックスのファイルI/Oタブでチェックできます。
ramconnecttype	Expressの接続のタイプです。ダイレクト接続に対しては常に0を指定します

パラメータ

server値には、Oracle Express Connection Editorで作成され、接続ファイル (.xcf) に格納されている設定に対応する4つのパラメータが含まれます。これら4つのパラメータはすべて必須で、どのような順序でも指定できます。次の表は、パラメータとその設定を示しています。

パラメータ	説明	設定
sl	サーバー・ログイン	-2: ホスト(ドメイン・ログイン) -1: ホスト(サーバー・ログイン) 0: 認証不要 1: ホスト(ドメイン・ログイン)および接続セキュリティ 2: ホスト(ドメイン・ログイン)およびコール・セキュリティ 3: ホスト(ドメイン・ログイン)およびパケット・セキュリティ 4: ホスト(ドメイン・ログイン)および整合性セキュリティ 5: ホスト(ドメイン・ログイン)およびプライバシー・セキュリティ 注意: Windows NTでは、すべての設定を使用します。UNIXシステムでは、0、-1、-2の設定のみを使用します。 これらの設定の詳細は、Express Connection Editorのヘルプ・システムを参照してください。
st	サーバー・タイプ	1: OLAP Server
ct	接続タイプ	0: Express接続
sv	サーバー・バージョン	1: Express 6.2以降

使用上の注意

- 文字列全体が引用符で囲まれているので、必要に応じて文字列にスペースを入れることができます(ユーザーIDがJohn Smithの場合など)。
- 文字列の中でスラッシュ(/)が必要な場合は、追加のスラッシュをエスケープ文字として使用してください。たとえば、ドメインがtools/reportsの場合、コマンド・ラインは次のようになります。


```
EXPRESS_SERVER="server=ncacn_ip_tcp:sburns2-pc/sl=0/  
st=1/ct=0/sv=1/ domain=tools//reports"
```
- 文字列の中で一重引用符を使用することができます。二重引用符で囲まれているので、特別に扱われることはありません。

ユーザーID、パスワードおよびドメイン(RWBLD60)を指定したEXPRESS_SERVERの例

```
rwbl60 userid=scott/tiger@nt805  
express_server="server=ncacn_ip_tcp:sburns2-pc/sl=0/st=1/ct=0/sv=1/  
user=orbuild/domain=tools/password=buildme/"  
report=exp.rdf destype=file desname=exp1.html  
desformat=html batch=yes
```

ユーザーID、パスワードおよびドメイン(RWBLD60)を指定していないEXPRESS_SERVERの例

```
rwbl60 userid=scott/tiger@nt805  
express_server="server=ncacn_ip_tcp:sburns2-pc/sl=0/st=1/ct=0/sv=1/"  
report=exp.rdf destype=file desname=exp1.html
```

```
desformat=html batch=yes
```

RAM (RWBLD60) を指定したEXPRESS_SERVERの例

```
rwrn60 userid=scott/tiger@nt805  
express_server="server=ncacn_ip_tcp:olap2-pc/sl=0/st=1/  
ct=0/sv=1/ramuser=appqa/rampassword=appqa1/ramexpressid=appqa/  
ramserverscript=¥¥olap3-pc¥¥orant¥¥olap¥¥ram6202¥¥ram¥¥sqstnt.rdc/  
rammasterdb=sqstnt.db/ramconnecttype=0"  
report=exp.rdf destype=file desname=exp1.html  
desformat=html batch=yes
```

CUSTOMIZE

説明

CUSTOMIZEは、MODULEまたはREPORTキーワードで選択したレポートに適用するXMLファイルまたはファイルのリストを指定します。XMLファイルには、レポート定義をなんらかの方法で変更するカスタマイズ（たとえば、フォントの変更またはカラーの変更）が含まれます。

詳細は、『パブリッシング・レポート』マニュアルを参照してください。

構文

```
[CUSTOMIZE=]filename.xml|(filename1.xml, filename2.xml, _)
```

値

有効なXMLレポート定義（必要な場合にはファイル名の前に付けられているパス情報が指定されたXMLレポート定義）を含むファイルの名前。

使用上の注意

- 通常、XMLレポート定義のファイル拡張子はXMLですが、CUSTOMIZEキーワードが指定されている場合には必ずしもそうである必要はありません。

SAVE_RDF

説明

SAVE_RDFは、CUSTOMIZEキーワードで選択したXMLカスタマイズ・ファイルとともにMODULEまたはREPORTキーワードで選択したレポートに保存するファイルを指定します。この引数は、CUSTOMIZEキーワードを介してXMLファイルに適用する.RDFファイルを所有しており、この2つの組み合わせを新規の.RDFファイルに保存する場合に最も有効です。

詳細は、『パブリッシング・レポート』マニュアルを参照してください。

構文

```
[SAVE_RDF=] filename.rdf
```

値

任意の有効なファイル名です。

<param>

説明

<param> は、レポートの定義の一部になるパラメータです。パラメータに指定する値は、そのパラメータの値として使用されます。

構文

```
<param>=value
```

値

そのパラメータの任意の有効な値です。

デフォルト

パラメータの「初期値」プロパティから取得されます。

使用上の注意

- ランタイム・パラメータ・フォームのデフォルト値を上書きできます。
- 引数の値を一重または二重引用符で囲むことができます。一重または二重引用符の動きは、オペレーティング・システムによって異なります。
- パラメータは未定義の値を参照する場合もあるため、PARAM値の解析または妥当性チェックは行われません。

RWCGI60

説明

RWCGI60は、WebサーバーとReports Server (RWMTS60)間のインタフェースを提供するReports Web CGI実行可能プログラムです。 使用しているのWebブラウザからレポートを動的に実行することができます。RWCGI60の詳細は、『パブリッシング・レポート』を参照してください。

RWCLI60

説明

RWCLI60はコマンド・ラインを解析し、指定されたReports Server (RWMTS60)に渡します。RWCLI60は、RWRUN60とよく似たコマンド・ラインを使用します。

構文

```
RWCLI60 MODULE|REPORT=runfile USERID=userid  
[ [keyword=]value | (value1, value2, ...) ] SERVER=tnsname
```

ここで、keyword=valueは、有効なコマンド・ライン引数です。

使用上の注意

- クライアント・コマンド・ラインで指定されたすべてのファイル名とパスは、コマンド・ファイルを除いて、サーバー・マシン上のファイルとディレクトリを指します。
- コマンド・ラインにCMDFILE=が含まれている場合、コマンド・ファイルは元のコマンド・ラインに読み込まれ、追加されてから、Reports Serverに送られます。ランタイム・エンジンによってコマンド・ファイルが再び読み込まれることはありません。

RWCLI60コマンド・ライン引数

[MODULE|REPORT=]runfile

[USERID=]userid

[PARAMFORM=]NO

[CMDFILE=]cmdfile

[TERM=]termtype

[ARRAYSIZE=]n

```
[DESTYPE=] { CACHE | LOCALFILE | FILE | PRINTER | SYSOUT | MAIL | XML }
[DESNAME=] desname
[DESFORMAT=] desformat
[CACHELOB=] { YES | NO }
[COPIES=] n
[CURRENCY=] currency_symbol
[THOUSANDS=] thousands_symbol
[DECIMAL=] decimal_symbol
[READONLY=] { YES | NO }
[LOGFILE=] logfile
[BUFFERS=] n
[BATCH=] YES
[PAGESIZE=] width x height
[PROFILE=] profiler_file
[RUNDEBUB=] { YES | NO }
[ONSUCCESS=] { COMMIT | ROLLBACK | NOACTION }
[ONFAILURE=] { COMMIT | ROLLBACK | NOACTION }
[KEYIN=] keyin_file
[KEYOUT=] keyout_file
[ERRFILE=] error_file
[LONGCHUNK=] n
[ORIENTATION=] { DEFAULT | LANDSCAPE | PORTRAIT }
[BACKGROUND=] { YES | NO }
[MODE=] { BITMAP | CHARACTER | DEFAULT }
[PRINTJOB] { YES | NO }
[TRACEFILE=] tracefile
[TRACEMODE=] { TRACE_APPEND | TRACE_REPLACE }
```

```
[TRACEOPTS=] {TRACE_ERR|TRACE_PRF|TRACE_APP|TRACE_PLS|
TRACE_SQL|TRACE_TMS|TRACE_DST|TRACE_ALL| (opt1, opt2, ...)}

[AUTOCOMMIT] {YES|NO}

[NONBLOCKSQL=] {YES|NO}

[ROLE=] rolename/[rolepassword] | (rolename/[rolepassword], . . .)

[BLANKPAGES=] {YES|NO}

[DISABLEPRINT=] {YES|NO}

[DISABLEMAIL=] {YES|NO}

[DISABLEFILE=] {YES|NO}

[DISABLENEW=] {YES|NO}

[DESTINATION=] filename.DST

[DISTRIBUTE=] {YES|NO}

[DELIMITER=] value

[CELLWRAPPER=] value

[DATEFORMATMASK=] mask

[NUMBERFORMATMASK=] mask

[PAGESTREAM=] {YES|NO}

[SERVER=] tnsname

[JOBNAME=] string

[SCHEDULE=] string

[TOLERANCE=] number

[DELIMITER=] value

[CELLWRAPPER=] value

EXPRESS_SERVER="server=[server]/domain=[domain]/user=[userid]/password=
[passwd]"

[AUTHID]=username/password

[CUSTOMIZE]=filename.xml | (filename1.xml, filename2.xml, . . .)

[SAVE_RDF]=filename.rdf
```

```
<param>=value
```

PARAMFORM

説明

PARAMFORMを指定する場合は、NOとする必要があります。

構文

```
[PARAMFORM=] NO
```

DESTYPE

説明

DESTYPEは、レポート出力を受け取るデバイスのタイプです。

構文

```
[DESTYPE=] { CACHE | LOCALFILE | FILE | PRINTER | SYSOUT | MAIL }
```

値

CACHE	出力はReports Serverのキャッシュへ直接送られます。DESTYPE=CACHEには、DISTRIBUTEキーワードとの互換性がないことに注意してください。サーバーは、コマンド・ラインにDISTRIBUTEを見つけると、DESTYPE=CACHEコマンド・ライン引数を無視します。
LOCALFILE	出力はクライアント・マシン上のファイルへ送られ、BACKGROUNDの値にかかわらず、強制的に同期コールになります。
FILE	出力はDESNAMEで指定したサーバー・マシン上のファイルに送られます。
PRINTER	出力はDESNAMEで指定したサーバー・マシンのプリンタに送られます。
MAIL	出力はDESNAMEで指定されたメール・ユーザーに送られます。MAPI準拠またはサービス・プロバイダ・ドライバがインストールされているメール・システムであれば、メールを送ることができます。レポートは、添付ファイルとして送られます。
SYSOUT	出力はクライアント・マシンのデフォルトの出力デバイスに送られ、強制的に同期コールになります。

デフォルト

DESTYPEパラメータの初期値プロパティの設定値です。

使用上の注意

- R60CLIでは、DESTYPEに画面およびプレビュー画面を指定できません。

BATCH

説明

BATCHを指定する場合は、YESとする必要があります。

構文

```
[BATCH=] YES
```

BACKGROUND

説明

BACKGROUNDは、コールが同期 (BACKGROUND=NO) または非同期 (BACKGROUND=YES) のいずれであるかを指定します。同期コールの場合、クライアントはレポートがキューに入り、ランタイム・エンジンに割り当てられ、実行、終了するのを待ちます。非同期コールの場合、クライアントはただコールを送るのみで、コールの完了を待ちません。同期コール中にクライアント・プロセスが強制終了されると、ジョブは取り消されます。

構文

```
[BACKGROUND=] {YES|NO}
```

値

YESまたはNO

デフォルト

NO

SERVER

説明

SERVERは、Reports ServerのTNSアドレスです。

構文

```
[SERVER=] tnsname
```

値

任意の有効なTNSNAMEです。

使用上の注意

- Webサーバー・コンピュータで環境変数REPORTS60_REPORTS_SERVERを設定すると、デフォルト・サーバーに対して処理を要求する場合にSERVERコマンド・ライン引数を省略できます。あるいは、SERVERコマンド・ライン引数を組み込んでデフォルトを上書きすることもできます。

JOBNAME

説明

JOBNAMEは、Reports Queue Managerに表示されるジョブの名前です。コメントとして扱われ、ジョブの実行とは関係ありません。指定しない場合、レポート名がジョブ名として表示されます。

構文

```
[JOBNAME=] string
```

SCHEDULE

説明

SCHEDULEはスケジュール・コマンドです。デフォルトはnowです。スケジュール・コマンドを引用しなくてすむように、スペースのかわりにアンダースコア()を使用します。次に例を示します。

```
schedule=every_first_fri_of_month_from_15:53_Oct_23,_1999_retry_3_after_1_hour
```

```
schedule=last_weekday_before_15_from_15:53_Oct_23,_1999_retry_after_1_hour
```

注意: SCHEDULE構文の以前のフォームもサポートされますが、ここでは現行のSCHEDULE構文についてのみ説明します。

構文

```
[SCHEDULE=] string
```

ここで、stringは次のように指定します。

```
[FREQ from] TIME [retry {n} + after LEN]
```

FREQ	hourly daily weekly monthly {every LEN DAYREPEAT} {last {WEEKDAYS weekday weekend} before {n}+}
LEN	{n}+ {minute[s] hour[s] day[s] week[s] month[s]}

DAYREPEAT	{first second third fourth fifth} WEEKDAYS of month
WEEKDAYS	mon tue wed thu fri sat sun
TIME	now CLOCK [DATE]
CLOCK	h:m h:mm hh:m hh:mm
DATE	today tomorrow {MONTHS {d dd} [,year]}
MONTHS	jan feb mar apr may jun jul aug sep oct nov dec

TOLERANCE

説明

TOLERANCEは、重複ジョブ検出時間の許容範囲を分単位で示したものです。

構文

[TOLERANCE=] number

AUTHID

説明

AUTHIDは、保護されたReports Serverに対するユーザを認証するのに使用されるユーザ名およびパスワードです。セキュリティ検査の実行にはOracle WebDBが使用されます。WebDBにより、レポートを実行しているユーザは要求されたレポートを実行するのに必要なアクセス権限を持つことができるようになります。セキュリティの詳細は、Oracle WebDBのオンライン・ヘルプを参照してください。

ユーザが正常にログインすると、ブラウザは暗号化されたクッキーに送られ、これにより保護されたReports Serverに対するユーザの認証が行われます。デフォルトにより、クッキーは30分後には有効期限が切れます。クッキーが終了した場合、後続の要求(すでに保護されたサーバーへ送られているものなど)は、再び認証を受ける必要があります。

REPORTS60_COOKIE_EXPIRE環境変数を使用して認証クッキーの満了時間を変更できます。

ユーザーに対して認証を行い、そしてクッキーが終了するまでその認証を保持させる場合は、レポート要求からAUTHIDコマンドを省略します。ユーザーがWeb要求を出すたびに認証を行う場合は、Web CGIまたはWeb CartridgeコマンドのSHOWAUTHおよびAUTHYPE=SをレポートURLに組み込むか、%S引数をowscmd.dat (Web Cartridge)またはcgicmd.dat (Web CGI)ファイルのキー・マップ・エントリに組み込みます。

構文

[AUTHID=] username/password

値

Oracle WebDBで作成された任意の有効なユーザー名およびパスワード。WebDBに新規のユーザー・プロファイルを作成する場合はDBAに問い合わせてください。

ユーザーID

説明

ユーザーIDは、オプションのデータベース名またはリモート・データベースにアクセスするためのNet8サービス名、ODBCデータ・ソース名（非Oracleデータ・ソースにアクセスする場合）を持つOracleのユーザー名またはプレースホルダ・ユーザー名（たとえば、\$username）およびパスワードです。

オプションのNet8サービス名の詳細は、「リモート・ログイン」を参照してください。自動ログインの詳細は、「自動ログイン」を参照してください。

ユーザーがデータベースへログインできるようにする場合は、レポート要求からUSERIDコマンドを省略します。ユーザーがWeb要求を出すたびにログオンできるようにする場合は、Web CGIまたはWeb CartridgeコマンドのSHOWAUTHおよび AUTHYPE=DをレポートURLに組み込むか、%D引数をowscmd.dat（Web Cartridge）またはcgicmd.dat（Web CGI）ファイルのキー・マップ・エントリに組み込みます。

値

データベース接続の文字列は次の形式のいずれかで、長さは512バイトを超えてはいけません。

```
username [/password]
username [/password] [@database]
[user [/password]]@ODBC:datasource[:database] or
[user [/password]]@ODBC:*
<$username> [/password]
<$username> [/password] [@database]
```

RWCON60

説明

RWCON60によって、1つ以上のレポート定義またはPL/SQLライブラリをある保存形式から別の保存形式に変換できます。

RWCON60を使用して、次のレポート変換を実行できます。

- データベースに格納されたレポートを.rep、.rex、.rdfファイルまたはテンプレート(.tdfファイル)に変換
- rdfファイルに格納されたレポートをrepファイル、rexファイル、またはデータベース・レポート、テンプレート(tdfファイル)に変換
- rexファイルに格納されたレポートをデータベース・レポート、rdfファイル、repファイルまたはテンプレート(tdfファイル)に変換

RWCON60を使用して、次のPL/SQLライブラリ変換を実行できます。

- データベースに格納されたライブラリをpldまたはpllファイルに変換
- pldファイルに格納されたライブラリをデータベース・ライブラリまたはpllファイルに変換
- pllファイルに格納されたライブラリをデータベース・ライブラリまたはpldファイルに変換

構文

```
rwcon60 [ [keyword=]value ] ...
```

keyword=valueは、有効なコマンド・ライン引数です。

使用上の注意

- 変換を行うと、変換処理の一部として、レポートのPL/SQLが自動的にコンパイルされる場合もあります。変換先がrexファイルでないと、PL/SQLは次の場合に、自動的にコンパイルされます。その他の状況では、自分でレポートのPL/SQLをコンパイルする必要があります（「プログラム」→「コンパイル」→「すべて」など）。
- 「変換」を使用してrepファイルを作成します。コンパイルでエラーが発生すると、エラー・メッセージが表示され、repファイルは作成されません。
- ソース・レポートとしてrexファイルを使用します。コンパイルでエラーが発生すると、警告が表示されますが、変換は続行します。
- ソース・レポートとして別のプラットフォーム上に作成されたレポートを使用します。コンパイルでエラーが発生すると、警告が表示されますが、変換は続行します。

- フォントは、Report BuilderやReports Runtimeによってレポートをオープンしたときにマップされます。フォントは変換中にはマップされません。
- レポートをテンプレートに変換する場合、レポートのヘッダー・ページとトレーラ・ページおよびマージン領域にあるオブジェクトのみがテンプレートに使用されます。ボディにあるオブジェクトは無視されます。
- 連結ライブラリ付きのレポートを変換する場合、レポートに連結されたplファイルを変換してから、rdfファイルを変換する必要があります。

RWCON60コマンド・ライン引数

```
[USERID=]userid
[STYPE=] {DATABASE | PLLDB | PLDFILE | PLLFILE |
RDFFILE | REXFILE | XML}
[SOURCE=] {sname | (sname1, sname2, ...)}
[DTYPE=] {DATABASE | PLLDB | PLDFILE | PLLFILE |
RDFFILE | REPFIL | REXFILE | TDFFILE | XML}
[DEST=] {dname | (dname1, dname2, ...) | pathname}
[CMDFILE=] cmdfile
[LOGFILE=] logfile
[OVERWRITE=] {YES | NO | PROMPT}
[BATCH=] {YES | NO}
[DUNIT=] {CENTIMETER | CHARACTER | INCH | POINT}
[PAGESIZE=]width x height
[FORMSIZE=]width x height
[CUSTOMIZE=]filename.xml | (filename1.xml, filename2.xml, . . .)
```

STYPE

説明

STYPEは、変換するレポートまたはライブラリの形式です。

構文

```
[STYPE=] {DATABASE | PLLDB | PLDFILE | PLLFILE | RDFFILE | REXFILE | XML}
```

値

DATABASEの場合、ソース・レポートはOracleデータベースに格納されています。

PLLDDBの場合、ソースPL/SQLライブラリはデータベースに格納されています。

PLDFILEの場合、ソースPL/SQLライブラリはASCII形式のファイルに格納されています。

PLLFILFの場合、ソースPL/SQLライブラリはソース・コードおよびPコードを含むファイルに格納されています。

RDFFILEの場合、ソース・レポートは1つ以上のレポート定義ファイル（拡張子rdfを持つファイル）に格納されています。

REXFILEの場合、ソース・レポートは1つ以上のテキスト・ファイル（拡張子rexを持つファイル）に格納されています。

XMLの場合、ソース・レポートは1つ以上のXMLファイル（拡張子xmlを持つファイル）に格納されています。

デフォルト

database

SOURCE

説明

SOURCEは、変換する1つのレポートまたはライブラリ、あるいはレポートまたはライブラリのリストです。RWCON60を実行するには、ソース・レポートまたはソース・ライブラリを指定する必要があります。

構文

```
[SOURCE=] {sname | (sname1, sname2, ... ) }
```

値

任意の有効なレポートまたはライブラリ名、あるいはファイル名（qandaなど）。

名前をカンマで区切り、カッコで囲んだ有効なレポートまたはライブラリ名、あるいはファイル名のリスト（(qanda,test,dmast)など）。

使用上の注意

- データベースに格納されているレポートまたはライブラリには、SQLのワイルドカード文字（%および_）を使用できます。たとえばR%を指定すると、データベースに格納されているレポートのうちRで始まるものがすべてフェッチされます。該当するすべてのレポートが変換されます。
- レポートまたはライブラリ名、あるいはファイル名のリストは、名前をカンマで区切り、カッコで囲む必要があります。次に例を示します。

```
(qanda, test, dmast)
```

(qanda, test, dmast)

- ワイルドカード文字は、ファイルに格納されているレポートまたはライブラリ（拡張子rdf、rep、rex、pld、pllを持つもの）には無効です。
- このキーワードの引数は、オペレーティング・システム固有の場合があります。
- Oracleデータベースに格納されているレポートまたはライブラリを変換し、システム所有のReport Builder表を使用する場合、次の例のように、別のユーザーのレポートまたはライブラリをリストすることができます。

```
SOURCE=(sday.qanda,dsmith.test,dmast.sal)
```

- ユーザー所有のReport Builder表を使用する場合、複数のユーザーからのレポートまたはライブラリは、各ユーザー用に個別に変換する必要があります。
- レポートまたはライブラリを変換するには、これらを自分で作成済みであるかまたは自分が作成したものでない場合は、アクセス権限が付与されている必要があります。レポートまたはライブラリ名の前にユーザーIDが付いていない場合、ユーザーIDは現行のユーザーになります。
- データベースからファイルへの変換で、データベースのレポートまたはライブラリ名がファイル名として長すぎると、Report Builderでは、使用しているオペレーティング・システムに適した長さのファイル名を指定するように要求します。

DTYPE

説明

DTYPEは、レポートまたはライブラリの変換後の形式です。

構文

```
[DTYPE=] {DATABASE | PLLDB | PLDFILE | PLLFILE | RDIFFILE | REPPFILE | REXFILE | TDDFILE | XML}
```

値

DATABASEの場合、変換されたレポートは、Oracleデータベースに格納されます。

PLLDBの場合、変換されたPL/SQLライブラリは、データベースに格納されます。

PLDFILEの場合、変換されたPL/SQLライブラリは、ASCII形式のファイルに格納されます。

PLLFILEの場合、変換されたPL/SQLライブラリは、ソース・コードおよびPコードを含むファイルに格納されます。

RDIFFILEの場合、変換されたレポートは、1つ以上のレポート定義ファイル(拡張子rdfを持つファイル)に格納されます。

REPROFILEの場合、変換されたレポートは、1つ以上のバイナリ実行ファイル(拡張子repを持つファイル)に格納されます。

REXFILEの場合、変換されたレポートは、1つ以上のテキスト・ファイル(拡張子rexを持つファイル)に格納されます。

TDFFILEの場合、レポートはテンプレート(tdfファイル)に変換されます。

XMLの場合、宛先レポートは1つ以上のXMLファイル(拡張子xmlを持つファイル)に格納されています。

デフォルト

REXFILE

使用上の注意

- RWCON60を使用してrepファイルを作成する場合、ソース・レポートのPL/SQLは、自動的にコンパイルされます。コンパイルでエラーが発生すると、エラー・メッセージが表示され、repファイルは作成されません。こうならないように、repファイルを作成する前に、必ず「**プログラム**」→「**コンパイル**」を使用して、ソース・レポートのPL/SQLをコンパイルしてください。
- レポートをテンプレートに変換する場合、レポートのヘッダー・ページとトレーラ・ページおよびマージン領域にあるオブジェクトのみがテンプレートに使用されます。ボディにあるオブジェクトは無視されます。

DEST

説明

DESTは、データベースに格納される場合の変換済みのレポートまたはライブラリの名前またはファイル名です。

構文

```
[DEST=] {dname | (dname1, dname2, ...) | pathname}
```

値

任意の有効なレポートまたはライブラリ名、あるいはファイル名(qandaなど)。

名前をカンマで区切り、カッコで囲んだ有効なレポートまたはライブラリ名、あるいはファイル名のリスト((qanda,test,dmast)など)。

デフォルト

DESTキーワードを指定しない場合、RWCON60は次のデフォルト名を使用します。

- DTYPEがデータベースまたはPLLDDBならば、DESTはSOURCEです。

- DTYPEがPLDFILEならば、DESTはpld拡張子を持つSOURCEです。
- DTYPEがPLLFILEならば、DESTはpllファイル拡張子を持つSOURCEです。
- DTYPEがRDFFILEならば、DESTはrdf拡張子を持つSOURCEです。
- DTYPEがREPFILeならば、DESTはrep拡張子を持つSOURCEです。
- DTYPEがREXFILEならば、DESTはexpdat.rexです。

使用上の注意

- レポートまたはライブラリ名、あるいはファイル名のリストは、名前をカンマで区切り、カッコで囲む必要があります。次に例を示します。

```
(qanda, test, dmast)
```

```
(qanda, test, dmast)
```

- 存在するソース名よりも指定した宛先名が多い場合、余分な宛先名は無視されます。存在するソース名よりも指定した宛先名が少ない場合、宛先名がなくなるとデフォルト名が使用されます。
- このキーワードの引数は、オペレーティング・システム固有の場合があります。

CMDFILE

説明

CMDFILEは、RWCON60コマンドの引数を含むファイルです。このオプションを使用すると、レポートまたはライブラリを変換するのに、RWCON60を起動するたびに多くの引数を指定する必要がありません。

構文

```
[CMDFILE=] cmdfile
```

値

任意の有効なコマンド・ファイルです。

使用上の注意

- コマンド・ファイルは、別のコマンド・ファイルを参照できます。
- RWCON60引数のコマンド・ファイル構文は、コマンド・ライン上で使用される構文と同じです。
- コマンド・ラインに入力した値は、コマンド・ファイルで指定した値を上書きします。たとえば、コマンド・ラインからRWRUN60をCOPIES=1およびCMDFILE=RUNONE (コマンド・

ファイル)として指定するとします。RUNONEでは、COPIESは2に設定されています。この場合、生成されるレポートのコピーは1つのみです。詳細は、「コマンド・ライン構文ルール」を参照してください。

- このキーワードの引数は、オペレーティング・システム固有の場合があります。

LOGFILE

説明

LOGFILEは状態およびエラー出力が送られるファイル名。

構文

```
[LOGFILE=] logfile
```

値

任意の有効なファイル名です。

デフォルト

現行のディレクトリのdfltrep.logです。

OVERWRITE

説明

OVERWRITEは、既存のファイルまたはデータベース・オブジェクトを、変換済みのファイルまたはオブジェクトで上書きするかどうかを指定します。

構文

```
[OVERWRITE=] {YES|NO|PROMPT}
```

値

YESの場合、同名の既存のファイルまたはデータベース・オブジェクトを、自動的にすべて上書きします。

NOの場合、同名の既存のファイルまたはデータベース・オブジェクトがあっても、レポートを変換せず、警告メッセージを表示します。

PROMPTの場合、既存のファイルまたはデータベース・オブジェクトを上書きする前に、するかどうかの指示を要求します。

デフォルト

NO

BATCH

説明

BATCHは、レポートまたはライブラリの変換にユーザーが介入しないように、すべての端末入力および出力を非表示にします。

構文

```
[BATCH=] {YES|NO}
```

値

YESの場合、すべての端末入力および出力を抑止します。

NOの場合、特定の端末入力および出力を可能にします。「変換」ダイアログ・ボックスが表示され、ダイアログ・ボックスを確定すると、変換が実行されます。

デフォルト

NO

DUNIT

説明

DUNITは、レポートを変換する宛先のサイズの単位です。これを指定すると、DUNITは、ソース・レポートのサイズの単位と異なっている必要があります。これをブランクのままにしておくと、ソース・レポートのサイズの単位がDESTレポートのサイズの単位として使用されます。

構文

```
[DUNIT=] {CENTIMETER|CHARACTER|INCH|POINT}
```

値

CENTIMETERの場合、変換されたレポートは、サイズの単位として最初にセンチメートルを使用します。

CHARACTERの場合、変換されたレポートは、サイズの単位として最初に文字を使用します。

INCHの場合、変換されたレポートは、サイズの単位として最初にインチを使用します。

POINTの場合、変換されたレポートは、サイズの単位として最初にポイントを使用します。

デフォルト

ブランク

PAGESIZE

説明

PAGESIZEは、変換するレポートの論理ページのサイズを、宛先のサイズの単位(DUNITキーワードを使用して指定)で指定します。

構文

[PAGESIZE=]width x height

値

指定のサイズの単位でフォーム幅x高さの形をとる任意の有効値。

デフォルト

ビットマップでは、8.5 x 11インチです。キャラクタ・モードでは、80 x 66文字です。レポートがキャラクタ・モードに設計されており、ビットマップ上で実行または変換中の場合、何も指定されていないと、ページ・サイズを判断するのに次の計算式が使用されます。(デフォルトのページ・サイズ * キャラクタ・ページ・サイズ) / デフォルトのキャラクタ・ページ・サイズ。たとえば、キャラクタ・ページ・サイズが80 x 20の場合、ビットマップ・ページ・サイズは、 $(8.5 * 80) / 80 * (11 * 20) / 66 = 8.5 * 3.33$ となります。

使用上の注意

- キャラクタ以外のDUNITでは、端数を指定するのに小数を使用できます(8.5 x 11など)。

FORMSIZE

説明

FORMSIZEは、変換するレポートのランタイム・パラメータ・フォームを、宛先のサイズの単位(DUNITキーワードを使用して指定)で指定します。

構文

[FORMSIZE=]width x height

値

指定のサイズの単位でフォーム幅x高さの形をとる任意の有効値。

使用上の注意

- キャラクタ以外のDUNITでは、端数を指定するのに小数を使用できます(8.5 x 11など)。

CUSTOMIZE

説明

CUSTOMIZEは、SOURCEキーワードにリストされているレポートに適用するXMLファイルまたはファイルのリストを指定します。XMLファイルには、レポート定義を何らかの方法で変更するカスタマイズ（たとえば、フォントの変更またはカラーの変更）が含まれます。

詳細は、『Oracle Reports Developerパブリッシング・レポート』マニュアルを参照してください。

構文

```
[CUSTOMIZE=]filename.xml | (filename1.xml, filename2.xml, . . .)
```

値

有効なXMLレポート定義（必要な場合にはファイル名の前にパス情報を指定）を含むファイル名またはファイル名のリスト。

RWMTS60

説明

RWMTS60はOracle Reports Serverの実行可能プログラムです。レポートを実行する複数階層アーキテクチャを実現するために、Reports Serverを使用することで、リモート・アプリケーション・サーバー上でレポートを実行できるようになります。RWMTS60をコマンド・ラインから起動するには、RWCLI60を使用します。

Reports Web Cartridge、Reports Web CGIまたはReports Servletと一緒に使用する場合、Reports Serverでは標準URL構文を使用して、Webブラウザからレポートを実行することも可能です。サーバーは、Windows NT、Windows 95またはUNIX上にインストールできます。サーバーは、すべてのリクエストをジョブ・キューに入れることによって、レポートの実行というクライアントの要求を処理します。サーバーのランタイム・エンジンの1つが有効になると、キュー内の次のジョブが実行されます。キューのジョブ数が増えるにつれて、サーバーは、サーバー・プロセス開始時に指定した最大制限数に達するまで、ランタイム・エンジンを追加して起動することができます。同様に、アイドル状態が指定時間を超えると、アイドル・エンジンは停止します。

レポートを自動実行させるのに、サーバーを利用することができます。スケジュール・コマンドを使用することで、レポートに対して時間と頻度を指定することが可能になります。サーバーは、事前に定義された過去のジョブの最大数を記録しています。ジョブがいつキューに並んだか、起動したか、そして終了したかの情報は、レポートの最終状態と一緒に保存されます。この情報は、Reports Queue Manager(RWRQM60)から取り出されるか、あるいはAPIを介してWindows上で表示することができます。Reports Queue Managerは、Reports Serverと同じマシン上またはクライアント・マシン上に存在します。Unix上では、Reports Queue Viewer(RWRQV60)を使用してReports Serverのキューを表示できます。

RWRQM60

説明

RWRQM60は、(Windowsプラットフォームでの) Reports Queue Managerの実行可能プログラムです。リモート・サーバー上でのレポートの実行をスケジュールするのに、Reports Server実行可能プログラム(RWMTS60)を使用する場合、Reports Queue Managerを使用すると、レポートの最終状態とともに、いつジョブがキューに並んだか、起動したか、そして終了したかを表示することができます。また、ジョブとサーバーを操作することもできます。Reports Queue Managerは、Reports Serverと同じマシン上またはクライアント・マシン上に存在します。

UNIX上では、Reports Queue Viewer(RWRQV60)を使用してReports Serverのキューを表示できます。

RWRQV60

説明

RWRQV60は、(Unixプラットフォームでの) Reports Queue Viewer実行可能プログラムです。リモート・サーバ上でのレポートの実行をスケジュールするのにReports Server実行可能プログラム(RWMTS60)を使用する場合、Reports Queue Viewerを使用してReports Serverのキューを表示することができます。キーワード・ヘルプを表示するにはコマンド・ラインで、RWRQV60 "?"と入力します。

Windows上では、Reports Queue Manager(RWRQM60)を使用してReports Serverのキューを表示および操作できます。

RWOWS60

説明

RWOWS60は、Oracle Web ServerとReports Server(RWMTS60)間の接続を容易にするReports Webカートリッジです。この接続で、WebクライアントがReports Server上でレポートを実行したり、クライアントWebブラウザ上でレポート出力を見ることが可能になります。他のWebカートリッジと同様、RWOWS60はWeb Request Broker対応の動的リンクライブラリ、または共有ライブラリです。このライブラリは特にOracle Web Application Serverのリスナーと対応しています。RWOWS60はReports Web CGI (R30CGI) に代わるものであり、同じ機能を持っています。そしてOracle Web Application Serverとの統合にも対応しています。

RWRBE60

説明 RWRBE60はReportsバックグラウンド・エンジンです。RWRBE60はReportsバックグラウンド・エンジンは、コマンド・ラインがレポートを実行するのを待機します。コマンド・ラインを受け取ると、そのレポートをそのキューに入れます。一般に、Reportsバックグラウンド・エンジンでは、RUN_PRODUCTパッケージ・プロシージャからまたはSRW.RUN_REPORTを使用してレポートからコマンド・ラインを受け取ります。

構文

RWRBE60

使用上の注意

- RWRBE60は、以前のバージョンではR25SRVでコールされました。
- RWRBE60実行可能プログラムは、ダブル・クリックして起動することもできます。

- Windowsでは、RWISV60を使用してレポートをReportsバックグラウンド・エンジンに送り、実行することができます。これは、パフォーマンスのオーバーヘッドがReports Runtimeよりも少ないので、より効果的な方法です。RWISV60のコマンド・ライン引数は、RWRUN60のコマンド・ライン引数と同じです。
- RWRBE60は、レポートをローカルに実行するときのみ使用されます。Reports Server (RWMTS60)にレポートを送るには、RWCLI60を使用します。
- RWRBE60はセッションを閉じる際、別のデータベースに対するレポートを実行するまでデータベースとのセッションを閉じることはしません。これはデータベースとの接続時間を減らすためです。

RWRUN60

説明

RWRUN60は、レポート（つまり、rdfファイルまたはrepファイル）を対話形式またはバッチで実行します。

構文

```
RWRUN60 [ [keyword=]value|(value1, value2, ...) ]
```

ここで、keyword=valueは、有効なコマンド・ライン引数です。

使用上の注意

- repファイルを実行する場合、PL/SQLはすでにコンパイルされているので、再コンパイルはされません。rdfファイルを実行する場合は、必要に応じて、PL/SQLは自動的に再コンパイルされます（つまり、Report Builderのレポートをコンパイルまたは保存しなかったか、あるいは、レポートを実行しているプラットフォームまたはバージョンと、レポートを最後にコンパイルまたは保存したプラットフォームとの互換性がないとき）。
- RWRUN60には2つのバージョンがあります。いずれのバージョンでもキャラクタ・モードまたはビットマップ・レポートを実行できることに注意してください。唯一の制約は出力デバイスです（ビットマップ・レポートをvt220の画面で実行できないなど）。
 - キャラクタ・モードの実行プログラムは、キャラクタ・モード・マシン（vt220など）からレポートを実行するユーザー用です。
 - GUI実行プログラムは、ビットマップ・マシン（Windows 95/NT稼動PCなど）からレポートを実行するユーザー用です。

RWRUN60コマンド・ライン引数

[MODULE|REPORT=] runfile
[USERID=] userid
[PARAMFORM=] {yes|no}
[CMDFILE=] cmdfile
[TERM=] termttype
[ARRAYSIZE=] n
[DESTYPE=] {SCREEN|FILE|PRINTER|PREVIEW|SYSOUT|MAIL}
[DESNAME=] desname
[DESFORMAT=] desformat
[CACHELOB=] {YES|NO}
[COPIES=] n
[CURRENCY=] currency_symbol
[THOUSANDS=] thousands_symbol
[DECIMAL=] decimal_symbol
[READONLY=] {yes|no}
[LOGFILE=] logfile
[BUFFERS=] n
[BATCH=] {yes|no}
[PAGESIZE=] width x height
[PROFILE=] profiler_file
[RUNDEBUG=] {yes|no}
[ONSUCCESS=] {COMMIT|ROLLBACK|NOACTION}
[ONFAILURE=] {COMMIT|ROLLBACK|NOACTION}
[KEYIN=] keyin_file
[KEYOUT=] keyout_file
[ERRFILE=] error_file

```
[LONGCHUNK=] n

[ORIENTATION=] {DEFAULT | LANDSCAPE | PORTRAIT}

[BACKGROUND=] {yes | no}

[MODE=] {BITMAP | CHARACTER | DEFAULT}

[PRINTJOB] {YES | NO}

[TRACEFILE=] tracefile

[TRACEMODE=] {TRACE_APPEND | TRACE_REPLACE}

[TRACEOPTS=] {TRACE_ERR | TRACE_PRF | TRACE_APP | TRACE_PLS |
TRACE_SQL | TRACE_TMS | TRACE_DST | TRACE_ALL | (opt1, opt2, ...)}

[AUTOCOMMIT=] {YES | NO}

[NONBLOCKSQL=] {yes | no}

[ROLE=] rolename / [rolepassword]

[BLANKPAGES=] {YES | NO}

[DISABLEPRINT=] {YES | NO}

[DISABLEMAIL=] {YES | NO}

[DISABLEFILE=] {YES | NO}

[DISABLENEW=] {YES | NO}

[DELIMITER=] value

[CELLWRAPPER=] value

[DATEFORMATMASK=] mask

[NUMBERFORMATMASK=] mask

[DESTINATION=] filename.DST

[DISTRIBUTE=] {YES | NO}

[PAGESTREAM=] {YES | NO}

EXPRESS_SERVER="server=[server] /domain=[domain] /user=[userid] /password=
[passwd]"

[CUSTOMIZE]=filename.xml | (filename1.xml, filename2.xml, . . .)

[SAVE_RDF]=filename.rdf
```

<param>=value

MODULEIREPORT

説明

MODULEIREPORTは、実行するレポートの名前です。（REPORTは、下位互換性のためのものです。）

構文

[MODULE|REPORT=]runfile

値

任意の有効な実行ファイル（つまり、.rdfまたは.repの拡張子が付いたファイル）またはXMLレポート定義（.xmlファイル）です。ファイル拡張子を入力しないと、Reports Runtimeでは、最初に拡張子.repの付いたファイル、次に拡張子.rdfの付いたファイル、そして拡張子のないファイルの順に検索されます。Reports Runtimeでは、ファイルの検索にファイル・パス検索順序が使用されます。

使用上の注意

- レポートを実行するには、レポート定義は完全に行う必要があります（それ自身のデータ定義およびレイアウト定義が含まれていることなど）。レポート定義の一部のみを実行することはできません。
- MODULEIREPORTキーワードを指定した場合、XMLレポート定義ではXMLファイルの拡張子が指定されている必要があります。

MODULEIREPORTの制限

- MODULEIREPORTを使用して実行ファイルを指定すると、「RWRUN60」ウィンドウは呼び出されません（この動作により、別のアプリケーションからRWRUN60へ連続的に入ることができます）。MODULEIREPORTを使って実行ファイルを指定せず、BATCH=NOを指定すると、「RWRUN60」ウィンドウがオープンし、対話形式でレポートを実行できます。
- ビットマップのRWRUN60を介してキャラクタ・モードのレポートを実行する場合、Reports Runtimeでは警告を表示してから、8.5 x 11インチのページ・サイズと7 x 6インチのフォーム・サイズを使ってレポートを実行します。

PARAMFORM

説明

PARAMFORMは、レポートの実行時にランタイム・パラメータ・フォームを表示するかどうかを指定します。

構文

```
[PARAMFORM=] {YES|NO}
```

値

YESの場合、フォームを表示します。

NOの場合、フォームを非表示にします。

デフォルト

YES

使用上の注意

バッチ・モードでランタイム・パラメータ・フォームを表示しても無意味なので、PARAMFORM=YESでBATCH=YESは指定できません。

CMDFILE

説明

CMDFILEは、RWRUN60コマンドの引数を含むファイルです。このオプションを使用すると、レポートを実行するたびに多くの引数を指定しなくても、レポートを実行できます。

構文

```
[CMDFILE=] cmdfile
```

値

任意の有効なコマンド・ファイルです。

CMDFILEの制限

- コマンド・ファイルは、別のコマンド・ファイルを参照できます。
- RWRUN60引数のコマンド・ファイル構文は、コマンド・ラインで使用される構文と同じです。

- コマンド・ラインに入力した値は、コマンド・ファイルで指定した値を上書きします。たとえば、コマンド・ラインからRWRUN60をCOPIES=1およびCMDFILE=RUNONE（コマンド・ファイル）として指定するとします。RUNONEでは、COPIESは2に設定されています。この場合、生成されるレポートのコピーは1つのみです。
- このキーワードの引数は、オペレーティング・システム固有の場合があります。

TERM

説明

TERMは、RWRUN60を使っている端末のタイプです。TERMは、ランタイム・パラメータ・フォームおよびランタイム・プレビューアにのみ有効です。このキーワードが使用されるのは、キャラクタ・モードの場合のみです。

構文

```
[TERM=] termtyp
```

値

任意の有効な端末のタイプです。

デフォルト

インストール状態によって異なります（システム管理者に問い合せてください）。

使用上の注意

- このキーワードの引数は、使用しているオペレーティング・システムによっては、大/小文字の区別が必要な場合があります。

ARRAYSIZE

説明

ARRAYSIZEは、Oracle配列処理用の配列サイズ（キロバイト）です。一般に、配列サイズが大きいほど、レポートの実行速度は速まります。

構文

```
[ARRAYSIZE=] n
```

値

1～9,999までの数値です。つまり、Reports Runtimeは、レポートの問合せ1つに対してこのキロバイト数のメモリーを使用します。

デフォルト

デフォルトの配列サイズは10Kです。Oracleの配列処理の詳細は、『Oracle8 Server管理者ガイド』を参照してください。

DESTYPE

説明

DESTYPEは、レポート出力を受け取るデバイスのタイプです。

構文

```
[DESTYPE=] { SCREEN | FILE | PRINTER | PREVIEW | SYSOUT | MAIL }
```

値

SCREEN	出力は対話形式で表示するためにプレビューアに送られます。この値は、BATCH=NOの場合にのみ有効です。Screenでは、フォント・エイリアシングは実行されません。
FILE	出力はDESNAME(次に説明するキーワード)で指定したファイルに保存されます。
PRINTER	出力はDESNAME(次に説明するキーワード)で指定したプリンタに送られます。
PREVIEW	出力は対話形式で表示するためにプレビューアに送られます。ただし、Previewを指定すると、出力はPostScript出力としてフォーマットされます。プレビューアではDESNAMEを使って、出力の表示にどのプリンタのフォントを使用するかを判断します。Previewでは、フォント・エイリアシングが実行されます。
SYSOUT	出力はオペレーティング・システムのデフォルトの出力デバイスに送られます。この値は、BATCH=YESの場合にのみ有効です。
MAIL	出力はDESNAMEで指定されたメール・ユーザーに送られます。MAPI準拠またはサービス・プロバイダ・ドライバがインストールされているメール・システムであれば、メールを送ることができます。レポートは、添付ファイルとして送られます。

デフォルト

DESTYPEパラメータの初期値プロパティの設定値です。

使用上の注意

- DESTYPEをPRINTER、FILEまたはMAILに設定すると、非表示に指定していなければ、ラン

タイム・パラメータ・フォームを確定した後に、「印刷ジョブ」ダイアログ・ボックスが表示されます。「印刷ジョブ」ダイアログ・ボックスを確定すると、レポートが実行され、出力がプリンタ（「ファイル」→「ページ設定」の「プリンタの設定」で指定されている）またはファイル、メールIDに送られます。オペレーティング・システムによって、ダイアログ・ボックスの外観は異なります。

- このパラメータは、使用しているオペレーティング・システムによって上書きされる場合があります。
- オペレーティング・システムによっては、一部のDESTYPE値がサポートされないことがあります。

DESNAME

説明

DESNAMEは、レポート出力の送り先のファイル、プリンタ、あるいは電子メールID（または配布リスト）の名前です。電子メールでレポート出力を送るには、電子メール・アプリケーション（Windows上のMAPI準拠のアプリケーションや、UNIX上の固有のメール・アプリケーションなど）の場合と同様に電子メールIDを指定します。名前をカンマで区切ってカッコで囲むことによって、複数のユーザーを指定できます（(name, name, ...name)）。

構文

```
[DESNAME=] desname
```

値

長さが1Kを超えない任意の有効なファイル名またはプリンタ名、電子メールIDです。プリンタ名には、任意にポートを指定できます。次に例を示します。

```
DESNAME=printer, LPT1:  
DESNAME=printer, FILE:
```

デフォルト

DESNAMEパラメータの初期値プロパティの設定値です。DESTYPE=FILEとDESNAMEが指定されていない場合、実行時にデフォルトで<reportname>.lisになります。

使用上の注意

- DESTYPEがScreenまたはSysoutの場合、このキーワードは無視されます。
- DESTYPEがPreviewの場合、Reports RuntimeではDESNAMEを使って、出力の表示にどのプリンタのフォントを使用するかを判断します。

- このキーワードの引数は、使用しているオペレーティング・システムによっては、大/小文字の区別が必要な場合があります。
- このパラメータは、使用しているオペレーティング・システムによって上書きされる場合があります。

DESFORMAT

説明

DESFORMATは、ビットマップ環境では、DESTYPEがFileの場合に使用するプリンタ・ドライバを指定します。キャラクタ・モード環境では、DESNAMEで指定したプリンタの特性を指定します。

構文

[DESFORMAT=] desformat

値

長さが1Kを超えない任意の有効な宛先フォーマットです。このキーワードの有効値の例として、hpl、hplwide、dec、decwide、decland、dec180、dflt、wideなどがあります。有効な宛先フォーマットのリストについては、システム管理者に問い合わせてください。さらに、Report Builderでは次の宛先フォーマットがサポートされます。

PDF	レポート出力は、PDFビューアで読み込むことのできるファイルに送られる。PDF出力は、ご使用のシステム用に現在構成されているプリンタをベースにしています。現在選択されているプリンタのドライバが、出力の作成に使用されます。レポートを実行しているマシン用に、プリンタを構成する必要があります。
HTML	レポート出力は、HTML3.0準拠のブラウザ (Netscape 2.2など) で読み込むことのできるファイルに送られる。
HTMLCSS	ファイルに送られるレポート出力には、カスケード・スタイル・シートをサポートするHTML3.0準拠のブラウザで読み込むことのできるスタイル・シート拡張が含まれる。
HTMLCSSIE	ファイルに送られるレポート出力には、Microsoft Internet Explorer 3.xで読み込むことのできるスタイル・シート拡張が含まれる。
RTF	レポート出力は、標準のワード・プロセッサ (Microsoft Wordなど) で読み込むことのできるファイルに送られる。MS-WORDでファイルをオープンする場合、レポート内のすべての図形およびオブジェクトを表示するには、「表示」→「ページ・レイアウト」を選択する必要があります。
XML	レポート出力はXMLドキュメントとなり、XML拡張子のついた別個のファイルとして保存されます。このレポートは、XMLをサポートするブラウザで開いたり、読み込んだりできます。または選択したXML表示のアプリケーションに依存します。
DELIMITED	レポート出力は、標準のスプレッドシート・ユーティリティ (Microsoft Excelなど) で読み込むことのできるファイルに送られる。DELIMITERも指定する必要があります。

デフォルト

DESFORMATパラメータの初期値プロパティの設定値です。ビットマップのReports Runtimeでは、DESFORMATがblankまたはdfltの場合、現行のプリンタ・ドライバ（「ファイル」→「ページ設定」の「プリンタの設定」で選択）が使用されます。「プリンタの選択」で何も選択していない場合、デフォルトでPostScriptが使用されます。

使用上の注意

- DESTYPEがScreenまたはSysoutの場合、このキーワードは無視されます。
- このキーワードの値は、ご使用のオペレーティング・システムによっては、大文字と小文字の区別が必要な場合があります。

COPIES

説明

COPIESは、印刷するレポート出力の部数を指定します。

構文

[COPIES=] n

値

1～9,999までの任意の有効な整数です。

デフォルト

COPIESパラメータの初期値プロパティの設定値です。

使用上の注意

- DESTYPEがPrinterでない場合、このキーワードは無視されます。
- COPIESがランタイム・パラメータ・フォームで指定されていない場合、デフォルトで1に設定されます。

CACHELOB

説明

CACHELOBは、取り出されたOracle8ラージ・オブジェクトを（REPORTS60_TMPで指定された）一時ファイル・ディレクトリにキャッシュするかどうかを指定します。

値

YESの場合、LOBを一時ファイル・ディレクトリにキャッシュします。

NOの場合、LOBを一時ファイル・ディレクトリにキャッシュしません。

デフォルト

YES

使用上の注意

- このオプションは、コマンド・ラインからのみ設定できます。
- 一時ファイル・ディレクトリの場所に十分な空きディスク領域がない場合は、この値をNOに設定してください。ただし、値をNOに設定すると、サーバーから何度もLOBをフェッチすることになるため、パフォーマンスが低下する場合があります。

CURRENCY

説明

CURRENCYは、数値書式に使用される通貨記号です。

構文

```
[CURRENCY=] currency_symbol
```

値

長さが1Kを超えない任意の有効な英数字文字列です。

デフォルト

Oracleのデフォルトは、Oracle各国語サポート機能によって決まります。CURRENCYパラメータの初期値プロパティに、4文字までのデフォルトを設定することもできます。

使用上の注意

- ランタイム・パラメータ・フォームに入力されたCURRENCY値は、コマンド・ラインに入力されたCURRENCY値を上書きします。

THOUSANDS

説明

THOUSANDSは、数値書式に使用される3桁区切り記号です。

構文

```
[THOUSANDS=] thousands_symbol
```

値

任意の有効な英数字です。

デフォルト

Oracleのデフォルトは、Oracle各国語サポート機能によって決まります。THOUSANDSパラメータの初期値プロパティに、デフォルトを設定することもできます。

使用上の注意

- ランタイム・パラメータ・フォームに入力されたTHOUSANDS値は、コマンド・ラインに入力されたTHOUSANDS値を上書きします。
- THOUSANDS値として定義した英数字が、実際に返される値になります。たとえば、THOUSANDS値として "," を定義した場合は、"," が返されます。

DECIMAL

説明

DECIMALは、数値書式に使用される10進記号（インジケータ）です。

構文

```
[DECIMAL=] decimal_symbol
```

値

任意の有効な英数字です。

デフォルト

Oracleのデフォルトは、Oracle各国語サポート機能によって決まります。DECIMALパラメータの初期値プロパティに、デフォルトを設定することもできます。

使用上の注意

- ランタイム・パラメータ・フォームに入力されたDECIMAL値は、コマンド・ラインに入力されたDECIMAL値を上書きします。
- DECIMAL値として定義した英数字が、実際に返される値になります。たとえば、DECIMAL値として "." を定義した場合は、"." が返されます。

READONLY

説明

READONLYは、レポートの複数の問合せにおける読み一貫性を要求します。Oracleからデータにアクセスする場合、読み一貫性はSET TRANSACTION READ ONLY文によって達成されます (SET TRANSACTION READ ONLYの詳細は、『Oracle8 Server SQLリファレンス』を参照)。

構文

```
[READONLY=] {YES|NO}
```

値

YESの場合、読みの一貫性を要求します。

NOの場合、読みの一貫性は要求されません。

デフォルト

NO

使用上の注意

- Oracleは、1つの問合せを持つレポートでは、ロックせずに、読み一貫性を自動的に提供するので、READONLYが役に立つのは、複数の問合せを使用するレポートのみです。

LOGFILE

説明

- LOGFILEは、「ファイル」→「印刷」の出力の送り先のファイルの名前です。指定ファイルがすでにある場合、出力はそれに追加されます。このキーワードが使用されるのは、キャラクタ・モードの場合のみです。

構文

```
[LOGFILE=] logfile
```

値

任意の有効なファイル名です。

デフォルト

現行のディレクトリのdfltrep.logです。

BUFFERS

説明

BUFFERSは、仮想メモリー・キャッシュのサイズで、キロバイトで指定します。レポートの実行に十分な領域を確保する一方で、システムのリソースを使い過ぎないように調整する必要があります。

構文

```
[BUFFERS=]n
```

値

1から9,999までの数値です。オペレーティング・システムによっては、上限がこれよりも低い場合があります。

デフォルト

640K

使用上の注意

- この設定がセッションの途中で変更されても、変更は次のレポートが実行されるまで有効になりません。

BATCH

説明

BATCHは、レポートの実行時にユーザーが介入しないように、すべての端末入力および出力を抑止します。BATCH=YESにすると、RWRUN60インタフェースを使用しなくても、レポートを実行できます。バックグラウンドでレポートを実行する方法は、「BACKGROUND」を参照してください。

構文

[BATCH=] {YES | NO}

値

YESの場合、すべての端末入力および出力を抑止します。

NOの場合、特定の端末入力および出力を可能にします。

デフォルト

NO

使用上の注意

- BATCH=YESの場合、エラー・メッセージはSYSOUTに送られます。Sysoutの詳細は、「DESNAME」を参照してください。
- BATCH=YESの場合、バッチ・モードでランタイム・パラメータ・フォームを表示しても無意味なので、PARAMFORM=YESは無効です。
- BATCH=NOで、キャラクタ・モードのRWRUN60を使っている場合、ビットマップのレポートは実行できません（ビットマップのレポートはキャラクタ・モード・デバイスの画面で実行できません）。
- BATCH=YES、MODE=BITMAPで、キャラクタ・モードのRWRUN60を使っている場合、キャラクタ・モードのレポートは実行できません。

PAGESIZE

説明

PAGESIZEは、物理ページのサイズ（つまり、プリンタが出力するページのサイズ）です。ページには、レポートの内容が収まるだけの大きさが必要であることに注意してください。たとえば、レポートの枠サイズがページのサイズよりも大きくなると、レポートは実行されません。

構文

[PAGESIZE=]width x height

値

「ページ幅×ページの高さ」の形式で表される任意の有効なページ・サイズで、ページ幅およびページの高さは0（ゼロ）以上です。最大幅および高さは、サイズの単位によって異なります。インチでは、最大幅および高さは512インチです。センチメートルでは、1,312センチメートルです。パイカの場合、36,864パイカです。

デフォルト

ビットマップでは、8.5 x 11インチです。キャラクタ・モードでは、80 x 66文字です。レポートがキャラクタ・モードに設計されており、ビットマップ上で実行または変換中の場合、何も指定されていないと、ページ・サイズを判断するのに次の計算式が使用されます。(デフォルトのページ・サイズ * キャラクタ・ページ・サイズ) / デフォルトのキャラクタ・ページ・サイズ。たとえば、キャラクタ・ページ・サイズが80 x 20の場合、ビットマップ・ページ・サイズは、 $(8.5 * 80) / 80 * (11 * 20) / 66 = 8.5 * 3.33$ となります。

使用上の注意

- 物理ページの印刷可能領域が制限されているプリンタもあります。たとえば、プリンタ用紙は8.5 x 11インチでも、プリンタで印刷できる範囲が8 x 10.5インチしかない場合があります。Reports Runtimeでプリンタの印刷可能領域よりも大きいページ幅 x ページの高さを定義すると、レポート出力でクリッピングが発生する可能性があります。クリッピングを防ぐには、プリンタの印刷可能領域を拡大するか（オペレーティング・システムで可能な場合）、ページ幅 x ページの高さをページの印刷可能な領域のサイズに設定します。
- このキーワードを使用すると、その値はレポート定義のページのサイズを上書きします。
- ランタイム・パラメータ・フォームに入力されたPAGESIZE値は、コマンド・ラインに入力されたPAGESIZE値を上書きします。

PROFILE

説明

PROFILEは、レポートの実行に関するパフォーマンスの統計を格納するファイルの名前です。ファイル名を指定すると、Reports Runtimeは、レポートの実行にかかった経過時間およびCPU時間の統計を計算します。PROFILEは、次の統計を計算します。

- **合計経過時間**は、RWRUN60を発行してからレポートの実行を終えるまでに経過した時間です。合計経過時間は、Reports Runtime時間とOracle時間の合計です。
- **時間**は、Reports Runtimeでかかった時間です。
- **Oracle時間**は、データベースでかかった時間で、次のものから成り立っています。
 - **UPI**は、データベースへの接続およびSQLの解析、データのフェッチなどにかかった時間。
 - **SQL**は、SRW.DO_SQLの実行にかかった時間。
- **処理に使用した合計CPU時間**は、レポートの実行にかかったCPU時間。

注意: オペレーティング・システムによっては、データベースがReports Runtimeのプロセスに含まれているために、Reports Runtime時間にデータベース時間が含まれるものもあります。

構文

```
[PROFILE=]profiler_file
```

値

現行のディレクトリにある任意の有効なファイル名です。

RUNDEBUG

説明

RUNDEBUGは、レポートの論理エラーのチェックを行うかどうかを指定します。RUNDEBUGは、エラーではないが、望ましくない出力結果になる可能性があるものをチェックします。RUNDEBUGは、次のものをチェックします。

- 別のオブジェクトとオーバーラップしているが、囲んではいない枠または繰返し枠。これにより、オブジェクトが出力で他のオブジェクトを上書きする可能性があります。
- 固定のサイズ設定ではない、ページ依存の参照元を持つレイアウト・オブジェクト。Reports Runtimeでは、そのようなオブジェクトを「垂直拡張度」および「水平拡張度」プロパティの設定にかかわらず、固定サイズにします。
- 誤った頻度で参照されるPL/SQLのバインド変数。
- レポートはキャラクタ・モードまたはビットマップで、環境はその反対。

構文

```
[RUNDEBUG=] {YES|NO}
```

値

YESの場合、追加のランタイム・エラーのチェックを実行します。

NOの場合、追加のランタイム・エラーのチェックを実行しません。

デフォルト

NO

ONSUCCESS

説明

ONSUCCESSは、レポートの実行終了時に、COMMITまたはROLLBACKのどちらを実行するかを指定します。

構文

```
[ONSUCCESS=] { COMMIT | ROLLBACK | NOACTION }
```

値

COMMITの場合、レポートが終了するとCOMMITを実行します。

ROLLBACKの場合、レポートが終了するとROLLBACKを実行します。

NOACTIONの場合、レポートが終了しても何も行われません。

デフォルト

ユーザーIDが指定されている場合はCOMMITです。

ユーザーIDの指定がなく、外部ソース（Oracle Formsなど）からコールされる場合はNOACTIONです。

使用上の注意

- ONSUCCESSのCOMMITまたはROLLBACKは、「After Reportトリガー」の起動後に実行されます。その他のCOMMITおよびROLLBACKは、これより前に発生する場合があります。詳細は、「READONLY」を参照してください。

ONFAILURE

説明

ONFAILUREは、エラーが発生してレポートが終了できない場合に、COMMITまたはROLLBACKのどちらを実行するかを指定します。

構文

```
[ONFAILURE=] { COMMIT | ROLLBACK | NOACTION }
```

値

COMMITの場合、レポートが失敗するとCOMMITを実行します。

ROLLBACKの場合、レポートが失敗するとROLLBACKを実行します。

NOACTIONでは、レポートが失敗しても何も行いません。

デフォルト

ユーザーIDが指定されている場合は、ROLLBACKです。

ユーザーIDの指定がなく、外部ソース（Form Builderなど）からコールされる場合はNOACTIONです。

使用上の注意

- ONFAILUREのCOMMITまたはROLLBACKは、レポートが失敗した後に実行されます。その他のCOMMITおよびROLLBACKは、これより前に発生する場合があります。詳細は、「READONLY」を参照してください。

KEYIN

説明

KEYINは、実行時に実行するキーストローク・ファイルの名前です。KEYINは、KEYOUTで作成されたキーストローク・ファイルの実行に使用されます。KEYINは、キーストローク・ファイルの実行に使用されるので、キャラクタ・モード環境で実行している場合にのみ有効です。

構文

```
[KEYIN=]keyin_file
```

値

現行のディレクトリにある任意の有効なキー・ファイル名です。

KEYOUT

説明

KEYOUTは、Reports Runtimeによって、すべてのキーストロークを記録するキーストローク・ファイルの名前です。これを指定すると、キーストローク・ファイルを実行するのにKEYINを使用することができます。KEYOUTおよびKEYINは、レポートを実行するたびに実行する特定のキーストロークがある場合に便利です。これらはデバッグを行うのにも便利です。KEYOUTは、キーストローク・ファイルを作成するのに使用されるので、キャラクタ・モード環境でレポートを実行している場合にのみ有効です。

構文

```
[KEYOUT=]keyout_file
```

値

任意の有効なファイル名です。

ERRFILE

説明

ERRFILEは、Reports Runtimeによって、レポートの実行中に発行されたすべてのエラー・メッセージを格納するファイルの名前です。

構文

```
[ERRFILE=]error_file
```

値

任意の有効なファイル名です。

LONGCHUNK

説明

LONGCHUNKは、Reports RuntimeでLONG列値を取り出す変更分のサイズ（キロバイト）です。LONG値を取り出す場合、メモリーのサイズに制限があるので、すべてを一度に取り出さずに、変更分のみ取り出せます。LONGCHUNKは、Oracle7およびOracle8にのみ適用されます。

構文

```
[LONGCHUNK=]n
```

値

1から9,999までの数値です。オペレーティング・システムによっては、上限がこれよりも低い場合があります。

デフォルト

10K

ORIENTATION

説明

ORIENTATIONは、レポートのページを印刷する方向を制御します。

構文

```
[ORIENTATION=] {DEFAULT | LANDSCAPE | PORTRAIT }
```

値

DEFAULTの場合、現行のプリンタの方向設定を使用します。

LANDSCAPEの場合、用紙を横に使用します。

PORTRAITの場合、用紙を縦に使用します。

デフォルト

DEFAULT

使用上の注意

- キャラクタ・モードのレポートに、ORIENTATION=LANDSCAPEを設定する場合、プリンタ定義ファイルにLANDSCAPE句が含まれていることを確認する必要があります。
- レポートをMotif版のPCLプリンタに出力する場合、印刷方向の設定は無視されます。レポートは、デフォルトの設定を使用して処理されます。

BACKGROUND

説明

BACKGROUNDは、レポートをバックグラウンドで実行するかどうかを指定します。BACKGROUNDがYESの場合、別のプロセスが発生します。このオプションは、レポートの実行中に、フォアグラウンドで別の作業を行いたい場合に便利です。Windows 95/NTでは、BACKGROUND=YESを指定すると、レポートはReportsバックグラウンド・エンジンのキューに入れられます。

構文

```
[BACKGROUND=] {YES | NO }
```

値

YESまたはNO

デフォルト

NO

使用上の注意

- Windowsでは、RWRBE60を使ってレポートをReportsバックグラウンド・エンジンに送り、実行できます。これは、パフォーマンスのオーバーヘッドがReports Runtimeよりも少ないので、より効果的な方法です。RWRBE60のコマンド・ライン引数は、RWRUN60のコマンド・ライン引数と同じです。
- コマンド・ラインでは、BATCH=YES、DESTYPE=SCREENまたはPREVIEWあるいはRWBLD60の場合には、BACKGROUND=YESは使用できません。ただし、SRW.RUN_REPORTを使用している場合は、BATCH=YESでもBACKGROUND=YESが使用できます。
- コマンド・ラインやSRW.RUN_REPORTからではなく、Reports Runtimeからレポートを実行する場合、レポートを実行する前に、Before FormトリガーおよびAfter Formトリガー、妥当性チェック・トリガーで行ったデータベースの変更をコミットする必要があります。この方法で実行している場合、これらのトリガーは、親プロセスのデータベース接続を共有します。ただし、レポートが実際に実行されると、個々のデータベース接続が確立されます。

MODE

説明

MODEは、レポートをキャラクタ・モードまたはビットマップのどちらで実行するかを指定します。これによって、キャラクタ・モードのレポートをビットマップのReports Runtimeからまたはビットマップのレポートをキャラクタ・モードのReports Runtimeから実行できます。たとえば、端末(vt220など)からPostScriptプリンタにレポートを送る場合、キャラクタ・モードのRWRUN60を起動し、MODE=BITMAPを指定してレポートを実行できます。Windows 95/NTでは、MODE=CHARACTERを指定すると、Reports RuntimeASCIIドライバが編集可能なASCII出力の作成に使用されます。

構文

```
[MODE=] { BITMAP | CHARACTER | DEFAULT }
```

値

BITMAP

DEFAULTの場合、使用されている現行の実行プログラムのモードでレポートを実行します。

CHARACTER

デフォルト

DEFAULT

PRINTJOB

説明

PRINTJOBは、「印刷」ダイアログ・ボックスをレポートの実行前に表示するかどうかを指定します。

構文

```
[PRINTJOB=] {YES|NO}
```

値

YESまたはNO

デフォルト

YES

使用上の注意

- レポートが、発生したプロセスとして実行される（つまり、RWRUN60などの実行プログラムが、RWBLD60などの別の実行プログラムからコールされる）場合、PRINTJOBの指定にかかわらず、「印刷」ダイアログ・ボックスは表示されません。
- DESTYPE=MAILの場合、PRINTJOBの指定にかかわらず、「印刷」ダイアログ・ボックスは表示されません。

TRACEFILE

説明

TRACEFILEは、Reports Runtimeがトレース情報のログをとるファイルの名前です。

構文

```
[TRACEFILE=] tracefile
```

値

任意の有効なファイル名です。

デフォルト

trace.dat

使用上の注意

- トレース情報は、rdfファイルの実行時にのみ生成されます。repファイルの実行時に、ログインの指定はできません。
- TRACEFILEのみでなくLOGFILEまたはERRFILEも指定する場合、すべてのトレース情報は、最後に指定したファイルに保存されます。たとえば、次の場合、err.logがRWRUN60コマンドで指定した最後のファイルなので、指定したトレース情報はすべてこのファイルに保存されます。

```
rwrund60 MODULE=order_entry USERID=scott/tiger  
TRACEFILE=trace.log LOGFILE=mylog.log  
ERRFILE=err.log
```

TRACEMODE

説明

TRACEMODEは、Reports Runtimeでトレース情報をファイルに追加するかまたはファイル全体を上書きするかを示します。

構文

```
[TRACEMODE=] {TRACE_APPEND|TRACE_REPLACE}
```

値

TRACE_APPENDは、新しい情報をファイルの終わりに追加します。

TRACE_REPLACEは、ファイルを上書きします。

デフォルト

TRACE_APPEND

使用上の注意

- トレース情報は、rdfファイルの実行時にのみ生成されます。repファイルの実行時に、ログインの指定はできません。

TRACEOPTS

説明

TRACEOPTSは、 トレース・ファイルにログをとるトレース情報を示します。

構文

```
[TRACEOPTS=] {TRACE_ERR|TRACE_PRF|TRACE_APP|TRACE_PLS|  
TRACE_SQL|TRACE_TMS|TRACE_DST|TRACE_ALL| (opt1, opt2, ...)}
```

値

カッコ内のオプションのリストは、カッコ内のオプションをすべて使用するという事です。たとえば、TRACE_OPTS=(TRACE_APP, TRACE_PRF)は、TRACE_APPおよびTRACE_PRFを適用するという事です。

TRACE_ALLの場合、トレース・ファイルに、考えられるすべてのトレース情報のログをとります。

TRACE_APPの場合、トレース・ファイルに、すべてのレポート・オブジェクトのトレース情報のログをとります。

TRACE_DSTの場合、トレース・ファイルに、配布リストをリストします。この情報を使って、どのセクションがどの宛先に送られたかを調べることができます。

TRACE_ERRの場合、トレース・ファイルに、エラー・メッセージと警告をリストします。

TRACE_PLSの場合、トレース・ファイルに、すべてのPL/SQLオブジェクトのトレース情報のログをとります。

TRACE_PRFの場合、トレース・ファイルに、パフォーマンス統計のログをとります。

TRACE_SQLの場合、トレース・ファイルに、トレース情報のログをとります。

TRACE_TMSの場合、トレース・ファイルに、各エントリのタイムスタンプを入力します。

デフォルト

TRACE_ALL

使用上の注意

- トレース情報は、rdfファイルの実行時にのみ生成されます。repファイルの実行時に、ロギングの指定はできません。

AUTOCOMMIT

説明

AUTOCOMMITは、データベースの変更（CREATEなど）をデータベースに自動的にコミットするかどうかを指定します。一部の非Oracleデータベース（SQL Serverなど）では、AUTOCOMMIT=YESを指定する必要があることに注意してください。

構文

```
[AUTOCOMMIT=] {YES | NO}
```

値

YESまたはNO

デフォルト

NO

NONBLOCKSQL

説明

NONBLOCKSQLは、Reports Runtimeでデータベースからデータをフェッチしている間に、他のプログラムを実行できるかどうかを指定します。

構文

```
[NONBLOCKSQL=] {YES | NO}
```

値

YESの場合、データがフェッチされている間、他のプログラムを実行できます。

NOの場合、データがフェッチされている間、他のプログラムを実行できません。

デフォルト

YES

ROLE

説明

ROLEは、実行時にそのレポートについて確認されるデータベースのロールを指定します。ROLEを使用すると、通常はアクセス権限がないデータベースの表に問合せを行うレポートを実行できるので便利です。

構文

```
[ROLE=] rolename/ [rolepassword]
```

値

有効なロールおよびロールのパスワード（オプション）です。

BLANKPAGES

説明

BLANKPAGESは、レポートを印刷する際にブランク・ページを抑止するかどうかを指定します。このキーワードは、レポート出力時に、ブランク・ページを印刷しない場合に使用してください。

構文

```
[BLANKPAGES=] {YES|NO}
```

値

YESは、すべてのブランク・ページを印刷します。

NOはブランク・ページを印刷しません。

デフォルト

YES

使用上の注意

BLANKPAGESは、論理ページが複数の物理ページ（またはパネル）にまたがっていて、すべての物理的なブランク・ページを印刷しないようにする場合に特に便利です。

DISABLEPRINT

説明

DISABLEPRINTは、「ファイル」→「印刷」、「ファイル」→「ページ設定」、(Motif上での)「ファイル」→「プリンタ選択」およびランタイム・プレビューア内の同等のツールバー・ボタンを無効にするかどうかを指定します。

構文

```
[DISABLEPRINT=] {YES|NO}
```

値

YESまたはNO

デフォルト

NO

DISABLEMAIL

説明

DISABLEMAILは、ランタイム・プレビューア内の「メール」メニューおよび同等のツールバー・ボタンを無効にするかどうかを指定します。

構文

```
[DISABLEMAIL=] {YES|NO}
```

値

YESまたはNO

デフォルト

NO

DISABLEFILE

説明

DISABLEFILEは、ランタイム・プレビューア内の「ファイル」→「ファイル出力」メニューを無効にするかどうかを指定します。

構文

[DISABLEFILE=] {YES|NO}

値

YESまたはNO

デフォルト

NO

DISABLENEW

説明

DISABLENEWは、ランタイム・プレビューAの新規インスタンスを表示する機能を抑止するために「表示」→「新規プレビューA」メニューを無効にするかどうかを指定します。

構文

[DISABLENEW=] {YES|NO}

値

YESまたはNO

デフォルト

NO

DESTINATION

説明

DESTINATIONキーワードを使用すると、現在実行中のレポートの配布を定義する.DSTファイルの名前を指定できます。

構文

[DESTINATION=] filename.DST

値

レポートまたはレポート・セクションの配布を定義する.DSTファイルの名前です。

使用上の注意

- DESTINATIONキーワードを使用可能にするには、コマンド・ラインからDISTRIBUTE=YESを指定する必要があります。

DISTRIBUTE

説明

DISTRIBUTEは、複数の宛先へのレポート出力の配布を使用可能または使用不可にします。宛先は、レポート配布定義または.DSTファイルに定義された配布リストで指定します。

構文

```
[DISTRIBUTE=] {YES | NO}
```

値

YESの場合、配布リストにレポートを配布します。

NOの場合、配布リストを無視して、DESNAMEおよびDESFORMATパラメータの指定に従ってレポートを出力します。これは基本的に、配布用のレポート・セットアップを実行するデバッグ・モードで、配布を実際には実行しません。

デフォルト

NO

使用上の注意

- DESTINATIONキーワードを使用可能にするには、DISTRIBUTE=YESを指定する必要があります。

DELIMITER

説明

DELIMITERは、レポート出力のセルを区切るために使用する文字（列）を指定します。

構文

```
[DELIMITER=] value
```

値

次のような任意の英数字または英数字の文字列を使用します。

,	各セルをカンマで区切ります。
.	各セルをピリオドで区切ります。

次の4つの予約済みの値を使用することもできます。

tab	各セルをタブで区切ります。
space	各セルをスペースで区切ります。
return	各セルを改行で区切ります。
none	デリミタを使用しません。

ASCIIキャラクタ・セットに基づくエスケープ・シーケンスを使用することもできます。

¥t	各セルをタブで区切ります。
¥n	各セルを改行で区切ります。

デフォルト

Tab

使用上の注意

この引数は、DESFORMAT=DELIMITEDを指定した場合にのみ使用できます。

CELLWRAPPER

説明

CELLWRAPPERは、レポート出力の区切りセルの周囲に表示される文字（列）を指定します。

構文

[CELLWRAPPER=]value

値

任意の英数字または英数字の文字列を使用します。

"	セルの両側に二重引用符が表示されます。
'	セルの両側に一重引用符が表示されます。

次の4つの予約済みの値を使用することもできます。

tab	セルの両側にタブが表示されます。
space	セルの両側にシングル・スペースが表示されます。
return	セルの両側に改行が表示されます。

none	セルの囲みを使用しません。
ASCIIキャラクタ・セットに基づくエスケープ・シーケンスを使用することもできます。	
¥t	セルの両側にタブが表示されます。
¥n	セルの両側に改行が表示されます。

デフォルト

none

使用上の注意

- この引数は、DESFORMAT=DELIMITEDを指定した場合にのみ使用できます。
- セルの囲みは、実際のデリミタとは異なります。

DATEFORMATMASK

説明

DATEFORMATMASKは、デリミタ付きレポート出力の日付値の表示方法を指定します。

構文

[DATEFORMATMASK=] mask

値

任意の有効な日付書式マスクです。

使用上の注意

- この引数は、DESFORMAT=DELIMITEDを指定した場合にのみ使用できます。

NUMBERFORMATMASK

説明

NUMBERFORMATMASKは、デリミタ付きレポート出力の数値の表示方法を指定します。

構文

[NUMBERFORMATMASK=] mask

値

任意の有効な数値書式マスクです。

使用上の注意

- この引数は、DESFORMAT=DELIMITEDを指定した場合にのみ使用できます。

PAGESTREAM

説明

PAGESTREAMは、レポートの開発者が次のいずれかを使って設定したナビゲーション・コントロールを使用して、レポートをHTMLまたはHTMLCSS出力としてフォーマットするときにページ・ストリームを使用可能または使用不可にします。

- レポート・プロパティ・パレットの「ページ・ナビゲーション・コントロール・タイプ」および「ページ・ナビゲーション・コントロール値」プロパティ。
- Before ReportトリガーのPL/SQL(SRW.SET_PAGE_NAVIGATION_HTML)。

構文

```
[PAGESTREAM=] {YES|NO}
```

値

YESの場合、ページはストリーム処理されます。

NOの場合、ページ・ストリームなしでレポートが出力されます。

デフォルト

NO

EXPRESS_SERVER

説明 EXPRESS_SERVERは、接続するOLAP Serverを指定します。

構文

```
EXPRESS_SERVER="server=[server]/domain=[domain]/  
user=[userid]/password=[passwd]"
```

RAMでの構文

```
EXPRESS_SERVER="server=[server]/domain=[domain]/
```

```

user=[userid]/password=[passwd]/ramuser=[ramuserid]/
rampassword=[rampasswd]/ramexpressid=[ramexpid]/
ramserverscript=[ramsscript]/rammasterdb=[ramdb]/
ramconnecttype=[ramconn] "

```

値

次の有効な接続文字列を二重引用符 (") で囲みます。

server	OLAP Server文字列 (ncacn_ip_tcp:olap2-pc/sl=x/st=x/ct=x/sv=x/ など)。
Domain	OLAP Serverのドメイン。
User	OLAP Serverにログインするためのユーザー ID。
Password	ユーザー IDに対応するパスワード。
ramuser	RDBMSにログインするためのユーザー ID。
rampassword	RDBMSに対応するパスワード。
ramexpressid	Oracle Sales Analyzerデータベースのユーザー ID。これはOracle Sales Analyzerデータベースに対してのみ必要となります。
ramserverscript	サーバー上のリモート・データベース構成ファイル (RDC) の完全なファイル名 (フル・パスを含む)。このファイルではコードおよびデータベースの位置などの情報を指定します。UNC (Universal Naming Convention) 構文を使用することにより、複数のユーザーが同一の接続を使用してデータにアクセスすることができ、データの位置に対して同じドライブ文字をマップする必要がありません。UNC構文のパスは、 ¥¥ServerName¥ShareName¥ となり、その後任意のサブフォルダまたはファイル、あるいはその両方が続きます。
rammasterdb	最初に接続するRelational Access Managerデータベースの名前データベース・ファイル名のみを指定する必要があります。このデータベースは、OLAP Serverに対するServerDBPathのパス・リストに含まれるディレクトリに常駐している必要があります。ダイレクト接続に対しては常に0を指定します。
ramconnecttype	Express接続のタイプ。ダイレクト接続に対しては常に0を指定します。

server値には、Oracle Express Connection Editorで作成され、接続ファイル (.xcf) に格納されている設定に対応する4つのパラメータが含まれます。これら4つのパラメータはすべて必須で、どのような順序でも指定できます。次の表は、パラメータとその設定を示しています。

パラメータ	説明	設定
-------	----	----

sl	サーバー・ログイン	-2: ホスト (ドメイン・ログイン) -1: ホスト (サーバー・ログイン) 0: 認証不要 1: ホスト (ドメイン・ログイン) および接続セキュリティ 2: ホスト (ドメイン・ログイン) およびコール・セキュリティ 3: ホスト (ドメイン・ログイン) およびパケット・セキュリティ 4: ホスト (ドメイン・ログイン) および整合性セキュリティ 5: ホスト (ドメイン・ログイン) およびプライバシー・セキュリティ 注意: Windows NTでは、すべての設定を使用します。UNIXシステムでは、0、-1、-2の設定のみを使用します。 これらの設定の詳細は、Express Connection Editorのヘルプ・システムを参照してください。
st	サーバー・タイプ	1: OLAP Server
ct	接続タイプ	0: Express接続
sv	サーバー・バージョン	1: Express 6.2以降

使用上の注意

- 文字列全体が引用符で囲まれているので、必要に応じて文字列にスペースを入れることができます (ユーザーIDがJohn Smithの場合など)。
- 文字列の中でスラッシュ (/) が必要な場合は、追加のスラッシュをエスケープ文字として使用する必要があります。たとえば、ドメインがtools/reportsの場合、コマンド・ラインは次のようになります。


```
EXPRESS_SERVER="server=ncacn_ip_tcp:olap2-pc/sl=0/st=1/ct=0/sv=1/ domain=tools//reports"
```
- 文字列の中で一重引用符を使用することができます。二重引用符で囲まれているので、特別に扱われることはありません。

RAM (RWRUN60) を指定したEXPRESS_SERVERの例

```
rwrn60 userid=scott/tiger@nt805
express_server="server=ncacn_ip_tcp:olap2-pc/sl=0/st=1/ct=0/sv=1/ramuser=appqa/rampassword=appqa1/ramexpressid=appqa/ramserverscript=¥¥olap3-pc¥orant¥olap¥ram6202¥ram¥sqstnt.rdc/rammasterdb=sqstnt.db/ramconnecttype=0"
report=exp.rdf destype=file desname=exp1.html
desformat=html batch=yes
```

ユーザーID、パスワードおよびドメインを指定したEXPRESS_SERVER (RWRUN60) の例

```
rwrn60 userid=scott/tiger@nt805
express_server="server=ncacn_ip_tcp:olap2-pc/sl=1/st=1/ct=0/sv=1/ user=orbuild/domain=tools/password=buildme/"
report=exp.rdf destype=file desname=exp1.html
```

```
desformat=html batch=yes
```

ユーザーID、パスワードおよびドメインを指定していないEXPRESS_SERVER (RWRUN60) の例

```
rwrun60 userid=scott/tiger@nt805  
express_server="server=ncacn_ip_tcp:olap2-pc/sl=0/st=1/ct=0/  
sv=1/" report=exp.rdf destype=file desname=exp1.html  
desformat=html batch=yes
```

CUSTOMIZE

説明

CUSTOMIZEは、MODULEまたはREPORTキーワードで選択したレポートに適用するXMLファイルまたはファイルのリストを指定します。XMLファイルには、レポート定義を何らかの方法で変更するカスタマイズ（たとえば、フォントの変更またはカラーの変更）が含まれます。

詳細は、『パブリッシング・レポート』マニュアルを参照してください。

構文

```
[CUSTOMIZE=]filename.xml|(filename1.xml, filename2.xml, ...)
```

値

有効なXMLレポート定義（必要な場合にはファイル名の前に付けられているパス情報が指定されたXMLレポート定義）を含むファイルの名前。

使用上の注意

- 通常、XMLレポート定義のファイル拡張子はXMLですが、CUSTOMIZEキーワードが指定されている場合には必ずしもそうではありません。

SAVE_RDF

説明

SAVE_RDFは、CUSTOMIZEキーワードで選択したXMLカスタマイズ・ファイルとともにMODULEまたはREPORTキーワードで選択したレポートに保存するファイルを指定します。この引数は、CUSTOMIZEキーワードを介してXMLファイルに適用するRDFファイルを所有しており、この2つの組み合わせを新規のRDFファイルに保存する場合にもっとも有効です。

詳細は、『パブリッシング・レポート』マニュアルを参照してください。

構文

```
[SAVE_RDF=] filename.rdf
```

値

任意の有効なファイル名です。

<param>

説明

<param> は、レポートの定義の一部になるパラメータです。指定した値は、そのパラメータの値として使用されます。

構文

```
<param>=value
```

値

そのパラメータの任意の有効な値です。

デフォルト

パラメータの「初期値」プロパティから取得されます。

使用上の注意

- ランタイム・パラメータ・フォームのデフォルト値を上書きできます。
- 引数の値を一重または二重引用符で囲むことができます。一重または二重引用符の働きは、オペレーティング・システムによって異なります。
- REPORTキーワードを使って実行ファイルを指定した場合、レポートの各パラメータ(入力マスク、妥当性チェック・トリガーなど)に指定された設定を使って、PARAM値の妥当性チェックが行われます。
- REPORTキーワードを使って実行ファイルを指定しない場合、パラメータが未定義のパラメータを参照している場合があるので、PARAM値の解析や妥当性チェックは行われません。

A

After Parameter Formトリガー, 109
After Reportトリガー, 110

B

Before Parameter Formトリガー, 110
Before Reportトリガー, 111
Between Pagesトリガー, 111

H

HTML出力, 206
 イメージのポップアップの表示, 206
HTMLタグを含む, 138

N

NULL時の値, 172

O

ObjectHeight変数, 138
ObjectWidth変数, 138
OLE2オブジェクト, 245
 プロパティ, 245
OLE2プロパティ, 245

Oracle8, 114
 使用上の注意, 114
 制限, 114

P

PL/SQL, 97
 グループ・フィルタ, 97
PL/SQL式, 260
PRINTER_INTRAY属性(SRW.SET_ATTR), 25
PL/SQLトリガー, 153

R

Reports Queue Manager (RWRQM60), 434
Reports Queue Viewer (rwrqv60), 435
Reports Server (RWMTS60), 434
 概要, 434
Reports Web Cartridge (RWCGI60), 416
Reports Web Cartridge (RWOWS60), 435
Reportsバックグラウンド・エンジン(RWRBE60), 435
RWBLD60, 381, 383, 385, 386, 387, 388, 389, 390, 391,
 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402,
 403, 404, 405, 406, 407, 408, 409, 410, 411, 414, 415,
 445
 概要, 383
 キーワード, 381, 385, 386, 387, 389, 390, 391,
 392, 393, 394, 395, 396, 397, 398, 399, 400, 401,
 402, 403, 404, 405, 406, 407, 408, 409, 410, 411,
 414, 415, 445
RWCGI60, 416
RWCLI60, 391, 405, 416, 419, 420, 421, 422, 423, 439,
 440, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452,

453, 455, 456, 457, 458, 459, 460, 461, 462, 464, 465, 472

概要, 416

キーワード, 391, 405, 419, 420, 421, 422, 423, 439, 440, 443, 444, 445, 446, 447, 448, 449, 451, 452, 453, 455, 456, 457, 458, 459, 460, 461, 462, 464, 465, 472

RWCON60, 381, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433

概要, 424

キーワード, 381, 425, 427, 428, 429, 430, 431, 432, 433

RWMTS60, 434

概要, 434

RWOWS60, 435

RWRBE60, 435

RWRQM60 (Reports Queue Manager), 434

rwrqv60 (Reports Queue Viewer), 435

RWRUN60, 381, 391, 436, 437, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 471, 472

概要, 436

キーワード, 381, 391, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 471, 472

コマンド・ライン引数, 437

S

SELECT文/スタティック値, 252

SQL句, 239

SQL問合せ文, 265

SRW ビルトイン・パッケージ

プロシージャ, 2

SRWビルトイン・パッケージ, 2

W

Webレポート, 416, 434, 435

キュー管理 (Windows上で), 434

キューの表示 (Unix上で), 435

動的に実行, 416, 435

X

XML, 357

プロパティ, 357

あ

アイコン名, 149

アクション・トリガー, 105, 106

説明, 105

値, 253

値リスト, 254

後で改ページ, 184

後のプリンタ・コード, 196

アプリケーション・コマンド・ライン (PDF), 205

アンカー, 129, 146

プロパティ, 129

アンカー・オブジェクトと連動, 181

アンカー・プロパティ, 129

い

位置, 353

位置揃え, 351

イメージ, 206, 207, 348

HTMLポップアップの表示, 206

印刷, 277, 278

プロパティ・パレットの例, 277

印刷オブジェクト・オン, 192

印刷方向, 276

お

親オブジェクト名, 136
 親グループ, 239
 親の枠の種類, 135
 親の枠の割合(%) , 135
 親列, 238

か

外部問合せソース・ファイル, 263
 可視, 217
 型, 255
 型(ボイラプレート), 143
 型(ボタン), 149
 「型」プロパティ(問合せ), 268
 画面分割キーを使用不可にする, 307

き

基本印刷オン, 191
 キャラクタ単位で設計, 303
 キューイング, 434, 435
 Reports Queue Manager(RWRQM60)をWindows
 上で使用, 434
 Reports Queue Viewer(rwrqv60)をUnix上で使用,
 435
 行当りのフィールド数, 345
 境界線, 340
 境界線を含む, 306

く

繰返し枠, 269, 272, 275, 276, 277, 278
 プロパティ, 269
 プロパティ・パレットの設定, 272
 プロパティ・パレットの例, 275, 276, 277
 繰返し枠プロパティ, 269

グループ, 227
 プロパティ, 227
 グループ・フィルタ, 97
 トリガー, 97
 グループ・プロパティ, 227
 グループ上レポート, 190, 318
 同じページにディテールを表示, 190
 合計の割合(%)の例, 318
 グループ左レポート, 190
 同じページにディテールを表示, 190
 クロス積グループ, 241

け

計算位置, 317

こ

子オブジェクト名, 131
 子ディメンション, 227, 228
 子問合せ, 237
 子の枠の種類, 131
 子の枠の割合(%) , 131
 コマンド・ライン, 378, 383, 416, 424, 425, 436, 437
 RWBLD60, 383
 RWCLI60, 416
 RWCON60, 424, 425
 RWRUN60, 436, 437
 実行プログラム・オプションのヘルプ, 378
 コメント, 178
 「コメント」プロパティ, 262
 問合せ, 262
 子列, 236

さ

最小孤立行(ボイラプレート), 139
 最小孤立レコード数, 275

サイズの単位, 283
最大垂直ボディ・ページ, 284
最大水平ボディ・ページ, 284
最大取出し行, 264
削除, 251
サマリー列, 317, 319, 320
 合計の割合(%)の例, 318, 319
 プロパティ, 317
 プロパティ・パレットの例, 318, 319, 322, 324
サマリーをフィールドの位置に合わす, 333
参照カーソル, 106, 309
 問合せ, 106
 プロパティ, 309

し

式, 98
 説明, 98
式列, 225
 プロパティ, 225
式列プロパティ, 225
システム・パラメータ, 246
 ~について, 246
事前定義済み値に対するリストを制限, 252
実行プログラム, 378, 380
 引数, 378
 キーワードの使用方法, 380
 起動, 378
 構文ルール, 378
 命名規則, 378
集合列, 317
条件, 237
条件付き書式, 206
初期値, 250
書式マスク, 212, 213

す

垂直拡張度, 197
垂直間隔の使用, 355

垂直繰返し枠, 244
垂直方向に縮小, 132
水平拡張度, 178
水平繰返し枠, 243
水平方向に縮小, 132
数値揃え, 352
ズーム・キーを使用不可にする, 307
ズームで起動, 307
スタイル, 353

せ

整列, 331
積順序, 322
セクション, 310
 プロパティ, 310
セクション・プロパティ, 310

そ

ソース, 257, 329
ソース (フィールド), 223
ソース・ファイルの書式, 141
ソース・ファイル名, 142
「ソース」プロパティ (繰返し枠), 280

た

第1列を隠す, 249
タイトル, 287
高さ, 310, 311
高さ (パラメータ・フォーム・ウィンドウ), 289
妥当性チェック・トリガー, 99
 ~について, 99

ち

チャート・ハイパーリンク, 157
チャート・パラメータ, 159
チャート・ファイル名, 156
チャート・プロパティ, 154
チャート問合せ, 160
チャート列, 162
中央揃え, 132, 133
 フィールド, 132, 133

つ

追加, 247
追加の属性 (HTML), 247
追加のハイパーリンク属性, 207
ツールバー・コマンド, 117
 プロパティ・パレット, 118

て

データ・リンク, 234
 プロパティ, 234
データ・リンク・プロパティ, 234
データ型, 211, 248, 256
データ型 (列), 168
データベース, 381
 ログイン, 381
データベース列, 209
 プロパティ, 209
データベース列プロパティ, 209
テキスト, 148
テキスト (テンプレート), 354
テンプレート, 330
 プロパティ, 330

と

問合せ, 106, 262
 コメント, 262
 参照カーソル, 106, 107
 プロパティ, 262
問合せプロパティ, 262
同等枠間 (垂直), 339
同等枠間 (水平), 339
トリガー, 94, 95, 97, 98, 99, 102, 105, 109, 110, 111
 アクション, 105, 106
 グループ・フィルタ, 97
 式, 98
 妥当性チェック, 99
 フォーマット, 102
 レポート, 94, 95, 109, 110, 111

な

名前, 183, 184, 255, 256

に

入力マスク, 250

ぬ

塗りパターン, 345

は

ハイパーリンク (HTML/PDF), 202
ハイパーリンクの宛先, 203
配布, 287, 310
破線, 341
バックグラウンド・カラー, 333

パネルの印刷順序, 285
幅, 253
幅 (パラメータ・フォーム・ウィンドウ), 288
幅 (物理ページ), 316
幅 (列), 173
パラメータ, 100, 246
 システム, 246
 妥当性チェック・トリガー, 99
 プロパティ, 246
パラメータ・フォーム, 255, 256
パラメータ・フォーム
 フィールド・プロパティ, 256
 ボイラープレート・プロパティ, 255
パラメータ・プロパティ, 246

ひ

日付揃え, 342
ビットマップ・オブジェクトを含む, 305
表示名, 206
ビルトイン・パッケージ, 2
 ~について, 2
 SRW, 2, 3
 すべて, 2

ふ

ファイルからの読み込み, 170
ファイル形式, 170
ファンクション, 320
フィールド, 132, 133, 210, 218
 中央揃え, 132, 133
 プロパティ, 210
 非表示の参照, 218
フィールド間 (垂直), 349
フィールド間 (水平), 348
フィールドとラベル間 (水平), 334
フィールドの上にラベルを置く, 352
「フィールド」プロパティ, 210
フィルタ, 97

 グループ, 97
フィルタ・タイプ, 230
フォアグラウンド・カラー, 347
フォーマット・トリガー, 102
 説明, 102
フォーム後値, 302
フォーム後型, 301
フォーム前値, 300
フォーム前型, 299
フォント, 346, 347
複数選択, 117
 プロパティ・パレット, 117
ブックマーク, 204
ブレイク順序, 164
ブレイク順序の設定, 174
ブレースホルダ列, 258
 プロパティ, 258
ブレースホルダ列プロパティ, 258
プレビューア・タイトルの抑止, 308
プレビューアのステータス行のテキスト, 305
プレビューアのステータス行を使用, 304
プレビューアのヒント行のテキスト, 304
プレビューアのヒント行を使用, 303
プロパティ, 121, 129, 137, 146, 154, 209, 210, 225, 226,
 227, 234, 241, 245, 246, 255, 256, 258, 261, 262, 269,
 282, 309, 310, 317, 330
 OLE2オブジェクト, 245
 XML, 357
アンカー, 129
繰返し枠, 269
グループ, 227
サマリー列, 317
参照カーソル問合せ, 309
式列, 225
セクション, 310
チャート, 154
データ・リンク, 234
データベース列, 209
テンプレート, 330
問合せ, 262
パラメータ, 246
パラメータ・フォーム・フィールド, 256
パラメータ・フォーム・ボイラープレート, 255

フィールド, 210
 プレースホルダ列, 258
 ボイラープレート, 137
 ボタン, 146
 マトリックス, 241
 リスト, 121
 列ソート, 261
 レポート, 282
 枠, 226
 プロパティ・パレット, 116, 117
 ~について, 116
 複数選択, 117

へ

ページ・ナビゲーション・コントロール・タイプ, 290
 ページ・ナビゲーション・コントロール値, 291
 ページ当りの最大レコード数, 274
 ページ当りの垂直パネル, 315
 ページ当りの水平パネル, 312
 ページ後値, 298
 ページ後型, 297
 ページ数, 290
 ページと枠間 (垂直), 338
 ページと枠間 (水平), 337
 ページ番号付け, 219
 ページ前値, 296
 ページ前型, 296

ほ

ボイラープレート, 137
 プロパティ, 137
 方向, 286, 313
 ホスト・メニュー項目を使用不可にする, 306
 ボタン
 プロパティ, 146
 ボタン・プロパティ, 146

ま

前で改ページ, 187
 前のプリンタ・コード, 195
 マスターとディテール間 (垂直), 337
 マスターとディテール間 (水平), 336
 マトリックス, 241
 プロパティ, 241
 マトリックス・グループ, 242
 プロパティの例, 242
 マトリックス・プロパティ, 241
 マトリックス・レポート, 319, 320, 324
 合計の割合(%)の例, 319
 サマリー列の例, 322, 324
 マルチメディア・ファイル, 150
 マルチメディア・ファイル・タイプ, 151
 マルチメディア列, 151
 マルチメディア列の型, 152

め

命名規則, 378
 実行プログラムの~, 378

も

文字色, 355
 文字揃え, 341

ゆ

有効化トリガー, 253

ら

ラベルの型, 147

り

- リセット位置, 327
- リンク, 234
 - 「データ・リンク」を参照, 234

れ

- レイアウト・セクションの終了, 208
- レコード数, 232
- 列, 209, 225, 258, 261, 317
 - 式列プロパティ, 225
 - 集計列, 317
 - データベース列プロパティ, 209
 - ブレースホルダ列プロパティ, 258
 - 「列ソート」プロパティ, 261
- 列ソート, 261
 - 「列ソート」プロパティ, 261
- 列モード, 270
- レポート, 282
 - プロパティ, 282
- レポート・グループ, 161
- レポート・トリガー, 94
 - 使用するレポート・トリガーの選択, 94
 - トリガーが起動する場合, 94
- レポート・プロパティ, 282
- レポート後値, 295
- レポート後型, 294

- レポート高, 314
- レポート幅, 314
- レポート前値, 293
- レポート前型, 292
- レポート列 (チャート・パラメータ対応), 160
- レポート列 (チャート列対応), 162

ろ

- ログイン, 381

わ

- 枠, 226
 - プロパティ, 226
- 枠間 (垂直), 350
- 枠間 (水平), 349
 - 「枠間の垂直間隔」プロパティ, 280
- 枠間の水平間隔, 273
- 枠付きの行伸長, 143
- 枠とフィールド間 (垂直), 335
- 枠とフィールド間 (水平), 335
- 枠のパターン, 344
- 枠のバックグラウンド・カラー, 343
- 枠のフォアグラウンド・カラー, 343
- 枠プロパティ, 226