

Oracle® GoldenGate

Windows and UNIX リファレンス・ガイド

11*g* リリース 1 (11.1.1)

B62271-01

2010 年 8 月

ORACLE®

Oracle GoldenGate Windows and UNIX リファレンス・ガイド, 11g リリース 1 (11.1.1)

B62271-01 (原本部品番号: E17791-01)

Copyright © 1995, 2010 Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントが、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供される場合は、次の Notice が適用されます。

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアを危険が伴うアプリケーションで使用する際、このソフトウェアを安全に使用するために、適切な安全装置、バックアップ、冗長性（**redundancy**）、その他の対策を講じることは使用者の責任となります。このソフトウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

Oracle は Oracle Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

このソフトウェアおよびドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても、一切の責任を負いかねます。

目次

.....

はじめに	Oracle GoldenGate のガイドについて	11
	このマニュアルの表記規則	12
	Oracle GoldenGate のヘルプの入手方法	13
第 1 章	GGSCI コマンド	14
	コマンドの概要	14
	Manager コマンド	15
	INFO MANAGER	15
	SEND MANAGER	15
	START MANAGER	16
	STATUS MANAGER	16
	STOP MANAGER	17
	Extract コマンド	17
	ADD EXTRACT	17
	ALTER EXTRACT	25
	CLEANUP EXTRACT	26
	DELETE EXTRACT	26
	INFO EXTRACT	27
	KILL EXTRACT	32
	LAG EXTRACT	32
	SEND EXTRACT	33
	START EXTRACT	43
	STATS EXTRACT	43
	STATUS EXTRACT	46
	STOP EXTRACT	46
	Replicat コマンド	47
	ADD REPLICAT	47
	ALTER REPLICAT	50
	CLEANUP REPLICAT	50
	DELETE REPLICAT	51
	INFO REPLICAT	51

.....

KILL REPLICAT	55
LAG REPLICAT	55
SEND REPLICAT	56
START REPLICAT	58
STATS REPLICAT	63
STATUS REPLICAT	64
STOP REPLICAT	65
ER コマンド	65
トレイル・コマンド	66
ADD EXTTRAIL	66
ADD RMTTRAIL	67
ALTER EXTTRAIL	68
ALTER RMTTRAIL	68
DELETE EXTTRAIL	69
DELETE RMTTRAIL	69
INFO EXTTRAIL	70
INFO RMTTRAIL	70
パラメータ・コマンド	71
EDIT PARAMS	71
SET EDITOR	71
VIEW PARAMS	72
データベース・コマンド	72
DBLOGIN	72
ENCRYPT PASSWORD	73
LIST TABLES	74
Trandata コマンド	74
ADD TRANDATA	74
DELETE TRANDATA	79
INFO TRANDATA	79
チェックポイント表コマンド	80
ADD CHECKPOINTTABLE	80
CLEANUP CHECKPOINTTABLE	81
DELETE CHECKPOINTTABLE	81
INFO CHECKPOINTTABLE	82
Oracle トレース表コマンド	83
ADD TRACETABLE	83
DELETE TRACETABLE	84
INFO TRACETABLE	84

DDL コマンド	85
DUMPDDL	85
その他のコマンド	86
!command	86
ALLOWNESTED	88
CREATE SUBDIRS	88
FC	88
HELP	90
HISTORY	90
INFO ALL	90
INFO MARKER	91
OBEY	92
SHELL	93
SHOW	93
VERSIONS	94
VIEW GGSEVT	94
VIEW REPORT	94
第 2 章 Oracle GoldenGate パラメータ	96
Oracle GoldenGate パラメータの使用	96
パラメータ・カテゴリ	99
GLOBALS パラメータの概要	99
Manager パラメータの概要	100
Extract パラメータの概要	102
Replicat パラメータの概要	107
DEFGEN パラメータの概要	114
アルファベット順リファレンス	114
ALLOCFILES	114
ALLOWDUPTARGETMAP NOALLOWDUPTARGETMAP	115
ASCIITOEBCDIC	115
ASSUMETARGETDEFS	115
AUTORESTART	116
AUTOSTART	117
BATCHSQL	118
BEGIN	122
BLOBMEMORY	122
BOOTDELAYMINUTES	122
BR	123

BULKLOAD.....	129
CACHEMGR.....	129
CHECKMINUTES.....	136
CHECKPARAMS.....	136
CHECKPOINTSECS.....	137
CHECKPOINTTABLE	137
CHECKSEQUENCEVALUE NOCHECKSEQUENCEVALUE.....	138
CMDTRACE.....	139
COLMATCH	139
COMMENT --	140
COMPRESSDELETES NOCOMPRESSDELETES.....	141
COMPRESSUPDATES NOCOMPRESSUPDATES.....	141
CUSEREXIT.....	142
DBOPTIONS	143
DDL.....	149
DDLERROR	154
DDLOPTIONS.....	159
DDLSUBST.....	166
DDLTABLE	171
DECRYPTTRAIL	172
DEFERAPPLYINTERVAL	172
DEFSFILE.....	173
DISCARDFILE	173
DISCARDROLLOVER.....	174
DOWNCRITICAL	175
DOWNREPORT	176
DSOPTIONS.....	176
DYNAMICPORTLIST	177
DYNAMICPORTREASSIGNDELAY.....	178
DYNAMICRESOLUTION NODYNAMICRESOLUTION	178
DYNSQL NODYNSQL.....	178
ENCRYPTTRAIL NOENCRYPTTRAIL	179
END	180
EOFDELAY EOFDELAYCSECS.....	181
ETOLDFORMAT.....	181
EXTFILE.....	182
EXTRACT.....	184
EXTTRAIL	185

FETCHOPTIONS	187
FILTERDUPS NOFILTERDUPS	188
FLUSHSECS FLUSHCSECS	189
FORMATASCII	190
FORMATSQL	192
FORMATXML	193
FUNCTIONSTACKSIZE	194
GENLOADFILES	195
GETAPPLOPS IGNOREAPPLOPS	197
GETDELETES IGNOREDELETES	198
GETENV	198
GETINSERTS IGNOREINSERTS	199
GETREPLICATES IGNOREREPLICATES	199
GETTRUNCATES IGNORETRUNCATES	201
GETUPDATEAFTERS IGNOREUPDATEAFTERS	202
GETUPDATEBEFORES IGNOREUPDATEBEFORES	202
GETUPDATES IGNOREUPDATES	203
GGSCHEMA	203
GROUPTRANSOPS	203
HANDLECOLLISIONS NOHANDLECOLLISIONS	204
HANDLETPKUPDATE	206
INCLUDE	207
INSERTAPPEND NOINSERTAPPEND	207
INSERTALLRECORDS	208
INSERTDELETES NOINSERTDELETES	208
INSERTMISSINGUPDATES NOINSERTMISSINGUPDATES	208
INSERTUPDATES NOINSERTUPDATES	209
LAGCRITICAL	209
LAGINFO	210
LAGREPORT	210
LIST NOLIST	211
LOBMEMORY	211
MACRO	213
MACROCHAR	215
Extract 用 MAP	216
Replicat 用 MAP	217
MAPEXCLUDE	258
MARKERTABLE	259

MAXDISCARDRECS	259
MAXFETCHSTATEMENTS	260
MAXSQLSTATEMENTS	260
MAXTRANSOPS	261
MGRSERVNAME	262
NOHEADERS	262
NUMFILES	263
OBEY	263
OUTPUTFILEUMASK	264
OVERRIDEDUPS NOOVERRIDEDUPS	264
PASSTHRU NOPASSTHRU	265
PASSTHRUMESSAGES NOPASSTHRUMESSAGES	266
PORT	266
PURGEDDLHISTORY	266
PURGEMARKERHISTORY	268
PURGEOLDEXTRACTS	269
PURGEOLDTASKS	272
RECOVERYOPTIONS	273
REPERROR	274
REFETCHEDCOLOPTIONS	277
REPLACEBADCHAR	279
REPLACEBADNUM	280
REPLICAT	280
REPORT	281
REPORTCOUNT	282
REPORTROLLOVER	283
RESTARTCOLLISIONS NORESTARTCOLLISIONS	284
RETRYDELAY	284
RMTFILE	284
RMTHOST	287
RMTHOSTOPTIONS	290
RMTTASK	292
RMTTRAIL	293
ROLLOVER	294
SEQUENCE	295
SETENV	301
SHOWSYNTAX	302
SOURCEDB	303

SOURCEDEFS	304
SOURCEISTABLE	304
SPACESTONULL NOSPACESTONULL	305
SPECIALRUN	305
SQLDUPERR	307
SQLEXEC	307
STARTUPVALIDATIONDELAY[CSECS]	310
STATOPTIONS	311
SYSLOG	312
DEFGEN 用 TABLE	313
Extract 用 TABLE	313
Replicat 用 TABLE	352
TABLEEXCLUDE	353
TARGETDB	354
TARGETDEFS	355
TCPSSOURCETIMER NOTCPSSOURCETIMER	356
THREDOPTIONS	356
TRACE/TRACE2	358
TRACETABLE NOTRACETABLE	359
TRANLOGOPTIONS	360
TRANSACTIONTIMEOUT	374
TRANSMEMORY	375
TRIMSPACES NOTRIMSPACES	378
UNLOCKEDTRAILFILES	378
UPDATEDELETES NOUPDATEDELETES	379
UPREPORT	379
USEDATEPREFIX	379
USERID	380
USETIMEPREFIX	383
USETIMESTAMPMPREFIX	383
VAM	384
VARWIDTHNCHAR NOVARWIDTHNCHAR	385
WARNLONGTRANS	386
WARNRATE	387
WILDCARDRESOLVE	388

第 3 章	Collector パラメータ	389
第 4 章	列変換ファンクション	393
	列変換ファンクションでの Unicode およびネイティブ・ エンコーディングの使用	393
	列変換ファンクションの概要	395
	BINARY	397
	BINTOHEX	398
	CASE	398
	COLSTAT	399
	COLTEST	399
	COMPUTE	400
	DATE	401
	DATEDIFF	404
	DATENOW	404
	EVAL	405
	GETENV	405
	GETVAL	419
	HEXTOBIN	420
	HIGHVAL LOWVAL	421
	IF	421
	NUMBIN	422
	NUMSTR	422
	RANGE	423
	STRCAT	424
	STRCMP	425
	STREQ	425
	STREXT	426
	STRFIND	427
	STRLEN	427
	STRLTRIM	428
	STRNCAT	428
	STRNCMP	429
	STRNUM	430
	STRRTRIM	431
	STRSUB	431
	STRTRIM	432
	STRUP	432
	TOKEN	433

	VALONEOF.....	434
第5章	ユーザー・イグジット・ファンクション.....	435
	ユーザー・イグジットのコール.....	435
	インストールされるユーザー・イグジット・ファイルについて	435
	ファンクションの概要	436
	EXIT_CALL_TYPE の使用	437
	EXIT_CALL_RESULT の使用.....	438
	EXIT_PARAMS の使用.....	439
	ERCALLBACK の使用.....	440
	ファンクション・コード	442
	COMPRESS_RECORD.....	445
	DECOMPRESS_RECORD.....	446
	FETCH_CURRENT_RECORD	448
	FETCH_CURRENT_RECORD_WITH_LOCK.....	449
	GET_BEFORE_AFTER_IND.....	450
	GET_COL_METADATA_FROM_INDEX.....	451
	GET_COL_METADATA_FROM_NAME.....	454
	GET_COLUMN_INDEX_FROM_NAME	456
	GET_COLUMN_NAME_FROM_INDEX	457
	GET_COLUMN_VALUE_FROM_INDEX	458
	GET_COLUMN_VALUE_FROM_NAME	461
	GET_DDL_RECORD_PROPERTIES.....	464
	GET_ENV_VALUE	466
	GET_ERROR_INFO.....	467
	GET_GMT_TIMESTAMP.....	468
	GET_MARKER_INFO	469
	GET_OPERATION_TYPE	470
	GET_POSITION	472
	GET_RECORD_BUFFER.....	473
	GET_RECORD_LENGTH	476
	GET_RECORD_TYPE.....	477
	GET_STATISTICS	478
	GET_TABLE_COLUMN_COUNT	480
	GET_TABLE_METADATA.....	481
	GET_TABLE_NAME	482
	GET_TIMESTAMP	483
	GET_TRANSACTION_IND	484

GET_USER_TOKEN_VALUE	486
OUTPUT_MESSAGE_TO_REPORT	487
RESET_USEREXIT_STATS	487
SET_COLUMN_VALUE_BY_INDEX	487
SET_COLUMN_VALUE_BY_NAME	489
SET_OPERATION_TYPE	491
SET_RECORD_BUFFER	492
SET_TABLE_NAME	494
付録 1 Oracle GoldenGate コミット順序番号について	495
用語集	498
索引	510

はじめに

Oracle GoldenGate のガイドについて

.....

Oracle GoldenGate のドキュメント一式には、次のコンポーネントが含まれています。

HP NonStop プラットフォーム

- *Oracle GoldenGate の HP NonStop 管理者ガイド*: NonStop プラットフォーム上で Oracle GoldenGate のレプリケーション・ソリューションの計画、構成および実装する方法について説明されています。
- *Oracle GoldenGate の HP NonStop リファレンス・ガイド*: NonStop プラットフォーム用の Oracle GoldenGate のパラメータ、コマンドおよび関数の詳細が記載されています。

Windows、UNIX、Linux の各プラットフォーム

- *インストールおよびセットアップ・ガイド*: Oracle GoldenGate によってサポートされているデータベースごとにそのようなガイドが 1 つあります。それには、Oracle GoldenGate のレプリケーション・ソリューションをインストールするためのシステム要件、インストール前およびインストール後の手順、インストールの指示、システム固有のその他の情報が記載されています。
- *『Oracle GoldenGate Windows and UNIX 管理者ガイド』*: Windows および UNIX プラットフォーム上で Oracle GoldenGate のレプリケーション・ソリューションの計画、構成および実装する方法について説明されています。
- *『Oracle GoldenGate Windows and UNIX リファレンス・ガイド』*: Windows および UNIX プラットフォーム用の Oracle GoldenGate のパラメータ、コマンドおよび関数の詳細が記載されています。
- *『Oracle GoldenGate Windows and UNIX トラブルシューティングおよびチューニング・ガイド』*: Oracle GoldenGate のレプリケーション・ソリューションのパフォーマンスを改善するための推奨事項とよくある問題の解決策が記載されています。

その他の Oracle GoldenGate 製品

- *『Oracle GoldenGate Director 管理者ガイド』*: Oracle GoldenGate のレプリケーション・コンポーネントで構成、管理、監視およびレポートするために Oracle GoldenGate Director をインストール、実行および管理する方法が説明されています。
- *『Oracle GoldenGate Veridata 管理者ガイド』*: Oracle GoldenGate Veridata のデータ比較ソリューションをインストール、実行および管理する方法が説明されています。
- *Oracle GoldenGate for Java の管理者ガイド*: Oracle GoldenGate の証跡への JMS メッセージをキャプチャし、キャプチャされたデータをメッセージング・システムまたはカスタム API に配布するために Oracle GoldenGate for Java をインストール、構成および実行する方法が説明されています。
- *Oracle GoldenGate for Flat File の管理者ガイド*: バッチ入力として Oracle GoldenGate によってキャプチャされるデータの形式を ETL、固有またはレガシー・アプリケーションに合わせるために Oracle GoldenGate for Flat File をインストール、構成および実行する方法が説明されています。

.....

このマニュアルの表記規則

このマニュアルでは、次の表記規則を使用します。

- パラメータとコマンド引数は、次の例のように大文字で表記されます。
`CHECKPARAMS`
- ファイル名、表名およびその他の名前は、関連するオペレーティング・システムまたはソフトウェア・アプリケーションで大/小文字が区別される場合を除き、次の例のように小文字で表記されます。
`account_tab`
`GLOBALS`
- 変数は、次の例のように山カッコ (< >) 内に表記されます。
`<group name>`
- 複数の相互排他的な引数の 1 つをを選択する必要がある場合は、次の例のように、選択肢が中カッコで囲まれ、パイプ記号で区切られています。
`VIEW PARAMS {MGR | <group> | <file name>}`
- オプションの引数は、次の例のように大カッコで囲まれています。
`CLEANUP EXTRACT <group name> [, SAVE <count>]`
- オプションの引数が多数ある場合は、[<option>] のようなプレースホルダが使用され、次の例のようにオプションが個別にリストおよび説明されています。
`TRANLOGOPTIONS [<option>]`
- 1 つの引数が複数回渡される場合は、次の例のように省略記号 (...) が使用されています。
`PARAMS ([<requirement rule>] <param spec> [, <param spec>] [, ...])`
- アンパサド (&) は、Oracle GoldenGate パラメータ・ファイルで継続文字として使用されています。複数行にわたるパラメータ文の各行の末尾にアンパサドを配置する必要があります。このドキュメントの大半の例では、アンパサドは適切な位置に配置されていますが、発行フォーマットのスペースの制約により、複数行にわたる文の一部では省略されていることがあります。

Oracle GoldenGate のヘルプの入手方法

Oracle GoldenGate ドキュメントに加え、次の方法で Oracle GoldenGate のヘルプを入手できます。

Oracle GoldenGate インタフェースからのヘルプの入手

GGSCI と Oracle GoldenGate Director アプリケーションの両方でオンライン・ヘルプが提供されます。

GGSCI コマンド

Oracle GoldenGate コマンドに関するヘルプを入手するには、GGSCI で **HELP** コマンドを使用します。コマンド・カテゴリのサマリーを入手するには、**HELP** コマンドをオプションなしで実行します。特定のコマンドに関するヘルプを入手するには、入力としてのコマンド名を指定して、**HELP** コマンドを実行します。

```
HELP <command name>
```

例：

```
HELP ADD EXTRACT
```

ヘルプ・ファイルに、そのコマンドの構文および説明が表示されます。

Oracle GoldenGate Director

Oracle GoldenGate Director クライアントまたは Oracle GoldenGate DirectorWeb のいずれかに関するヘルプを入手するには、そのアプリケーション内で「**Help**」メニューを使用します。

質問および問題に対するヘルプの入手

トラブルシューティングに関するヘルプは、『*Oracle GoldenGate Windows and UNIX* トラブルシューティングおよびチューニング・ガイド』を参照してください。また、<http://support.oracle.com> の「Knowledge Base」からその他の情報を入手できます。答えが得られない場合は、サポート・サイトからサービス・リクエストをオープンしてください。

第 1 章

GGSCI コマンド

.....

Oracle GoldenGate ソフトウェア・コマンド・インタフェース (GGSCI) は、ユーザーと Oracle GoldenGate 機能コンポーネント間のコマンド・インタフェースです。

コマンドの概要

コマンド・グループ	目的
Manager コマンド	Manager プロセスを起動および管理します。
Extract コマンド	Extract グループを作成および管理します。
Replicat コマンド	Replicat グループを作成および管理します。
ER コマンド	複数の Extract および Replicat グループを 1 つのユニットとして制御します。
トレイル・コマンド	トレイルを Extract グループに関連付け、ファイル管理パラメータを提供します。
パラメータ・コマンド	エディタを実行し、パラメータを定義または変更します。
データベース・コマンド	データベース関連コマンドを発行します。
Trandata コマンド	Oracle GoldenGate が UPDATE 操作をレプリケートするために必要な補足情報をロギングするようにデータベースを構成します。
チェックポイント表コマンド	Oracle GoldenGate チェックポイント表を作成および管理します。
Oracle トレース表コマンド	双方向構成でのデータのループを防止するためのトレース表を作成および管理します。
DDL コマンド	DDL 同期に関連するコマンドです。
その他のコマンド	その他の機能を制御します。

.....

Manager コマンド

Manager コマンドでは、Manager プロセスを制御します。Manager は Oracle GoldenGate の親プロセスで、自身のプロセスおよびファイル、リソース、ユーザー・インタフェース、しきい値およびエラーのレポートを管理します。

コマンドの概要

[INFO MANAGER](#)
[SEND MANAGER](#)
[START MANAGER](#)
[STATUS MANAGER](#)
[STOP MANAGER](#)

INFO MANAGER

INFO MANAGER では、Manager プロセスが実行中かどうかを確認します。Manager が実行中の場合は、ポート番号が表示されます。このコマンドは STATUS MANAGER の別名です。

構文 INFO MANAGER

SEND MANAGER

SEND MANAGER では、アクティブな Manager プロセスのステータスを取得するか、Manager パラメータ・ファイルに構成されている動的ポート情報を取得します。

構文 SEND MANAGER
 [CHILDSTATUS [DEBUG]]
 [GETPORTINFO [DETAIL]]
 [GETPURGEOLDEXTRACTS]

引数	説明
CHILDSTATUS [DEBUG]	Manager によって起動されたプロセスに関するステータス情報を取得します。DEBUG は、Manager によって割り当てられたポート番号を返します（DYNAMICPORTLIST パラメータを使用してポートを指定しておく必要があります）。
GETPORTINFO [DETAIL]	デフォルトでは、Manager によって割り当てられたポート、および対応するプロセス ID の現在のリストを取得します（DYNAMICPORTLIST パラメータを使用してポートを指定しておく必要があります）。 DETAIL は、DYNAMICPORTLIST で指定されているすべてのポート、およびプロセスに割り当てられているポートを含む補足情報を提供します。
GETPURGEOLDEXTRACTS	Manager パラメータ・ファイルの PURGEOLDEXTRACTS パラメータで設定されているトレイル・メンテナンス・ルールに関する情報を表示します。PURGEOLDEXTRACTS の詳細は、269 ページを参照してください。

例 1 SEND MANAGER CHILDSTATUS DEBUG は、次のような子プロセス・ステータスを返します。基本の CHILDSTATUS オプションも同様の表示を返しますが、Port 列は含まれません。

ID	Group	Process	Retry	Retry Time	Start Time	Port
1	ORAEEXT	2400	0	None	2006/01/23 21:08:32	7840
2	ORAEEXT2	2450	0	None	2006/01/23 21:08:33	7842

例 2 SEND MANAGER GETPORTINFO DETAIL は、次のような動的ポート・リストを返します。

Entry	Port	Error	Process	Assigned	Program
0	8000	0	2387	2006-01-01	
1	8001	0		10:30:23	
2	8002	0			

例 3 SEND MANAGER GETPURGEOLDEXTRACTS は、次のような情報を出力します。

```
PurgeOldExtracts Rules
Fileset                               MinHours MaxHours MinFiles MaxFiles UseCP
S:\GGS\DIRDAT\EXTTRAIL\P4\*          0         0         1         0     Y
S:\GGS\DIRDAT\EXTTRAIL\P2\*          0         0         1         0     Y
S:\GGS\DIRDAT\EXTTRAIL\P1\*          0         0         1         0     Y
S:\GGS\DIRDAT\REPTRAIL\P4\*          0         0         1         0     Y
S:\GGS\DIRDAT\REPTRAIL\P2\*          0         0         1         0     Y
S:\GGS\DIRDAT\REPTRAIL\P1\*          0         0         1         0     Y
OK
Extract Trails
Filename                               Oldest_Chkpt_Seqno IsTable IsVamTwoPhaseCommit
S:\GGS\8020\DIRDAT\RT                  3         0         0
S:\GGS\8020\DIRDAT\REPTRAIL\P1\RT      13         0         0
S:\GGS\8020\DIRDAT\REPTRAIL\P2\RT      13         0         0
S:\GGS\8020\DIRDAT\REPTRAIL\P4\RT      13         0         0
S:\GGS\8020\GGSLOG                     735275     1         0
S:\GGS\8020\DIRDAT\EXTTRAIL\P1\ET      14         0         0
S:\GGS\8020\DIRDAT\EXTTRAIL\P2\ET      14         0         0
S:\GGS\8020\DIRDAT\EXTTRAIL\P4\ET      14         0         0
```

START MANAGER

START MANAGER では、Manager プロセスを起動します。このコマンドは、非クラスタ環境で使します。Windows クラスタ環境では、Cluster Administrator から Manager を停止する必要があります。

構文 START MANAGER

STATUS MANAGER

STATUS MANAGER では、Manager プロセスが実行中かどうかを確認します。Manager が実行中の場合は、ポート番号が表示されます。

構文 STATUS MANAGER

STOP MANAGER

STOP MANAGER では、Manager を停止します。このコマンドは、非クラスタ環境で使します。Windows クラスタ環境では、Cluster Administrator から Manager を停止する必要があります。

構文

STOP MANAGER [!]

引数	説明
!	(感嘆符) Manager をシャットダウンするかどうかを確認するプロンプトをバイパスします。

Extract コマンド

Extract コマンドでは、Extract グループを作成および管理します。Extract プロセスは、構成パラメータに応じて完全なデータ・レコードまたはトランザクション・データの変更を取得した後、ターゲット表に適用されるかロード・ユーティリティなどの別のプロセスによって処理されるデータをターゲット・システムに送信します。

コマンドの概要

ADD EXTRACT	LAG EXTRACT
ALTER EXTRACT	SEND EXTRACT
CLEANUP EXTRACT	START EXTRACT
DELETE EXTRACT	STATS EXTRACT
INFO EXTRACT	STATUS EXTRACT
KILL EXTRACT	STOP EXTRACT

ADD EXTRACT

ADD EXTRACT では、Extract グループを作成します。SOURCEISTABLE タスクまたは別名 Extract が指定されている場合を除き、ADD EXTRACT は実行間の処理の継続性を維持するためにチェックポイントを作成します。Extract グループを作成する前に『Oracle GoldenGate Windows and UNIX 管理者ガイド』を確認してください。

Oracle GoldenGateGGSCI コマンド・インタフェースでは、Oracle GoldenGate Manager のインスタンスごとに最大 300 の Extract および Replicat グループの同時処理が完全にサポートされています。サポートされているレベルでは、すべてのグループが INFO および STATUS コマンドなどの GGSCI コマンドで完全に制御および表示可能です。サポートされているレベル以外では、グループ情報は表示されず、エラーが発生することがあります。Oracle GoldenGate では、お使いの環境を効率よく管理するため、Extract および Replicat グループ(合せて)の数が 300 以下に保持されるようにすることをお勧めします。

構文 標準、パンプまたはデータ・ポンプ Extract の場合

```
ADD EXTRACT <group name>
{, SOURCEISTABLE |
  , TRANLOG [<bsds name>] |
  , VAM |
  , EXTFILESOURCE <file name> |
  , EXTTRAILSOURCE <trail name> |
  , VAMTRAILSOURCE <VAM trail name>}
{, BEGIN {NOW | yyyy-mm-dd [hh:mi:[ss[.cccccc]]]} |
  , EXTSEQNO <seqno>, EXTRBA <relative byte address> |
  , LOGNUM <log number>, LOGPOS <byte offset> |
  , EOF |
  , LSN <value> |
  , EXTRBA <relative byte address> |
  , EOF | LSN <value> |
  , PAGE <data page>, ROW <row> |
  }
[, THREADS <n>]
[, PASSIVE]
[, PARAMS <parameter file>]
[, REPORT <report file>]
[, DESC "<description>"]
```

構文 別名 Extract の場合

```
ADD EXTRACT <group name>
, RMTHOST {<host name> | <IP address>}
, {MGRPORT <port>} | {PORT <port>}
[, RMTNAME <name>]
[, DESC "<description>"]
```

引数	説明
<group name>	<p>Extract グループ名。次の命名規則を使用します。</p> <ul style="list-style-type: none"> ◆ 下線 (_) などの英数字以外の文字を含む、最大 8 つの ASCII 文字を使用できます。オペレーティング・システムでファイル名に許可されているかぎり、任意の ASCII 文字を使用できます。この理由は、グループが関連するチェックポイント・ファイルで識別されるからです。 ◆ 次の ASCII 文字はファイル名には使用できません。 { \ / : * ? " < > } ◆ HP UX、Linux および Solaris では、ファイル名にコロン (:) またはアスタリスク (*) を使用できますが、推奨はされません。 ◆ 一般に、Oracle GoldenGate 内でグループ名の大 / 小文字は区別されません。たとえば、finance、Finance および FINANCE は、すべて同じであるとみなされます。ただし Linux 上では、グループ名 (ADD コマンドで明示的に定義されている場合は、そのパラメータ・ファイル名も含む) は、すべて大文字か、小文字である必要があります。グループ名およびパラメータ・ファイル名に大 / 小文字が混ざっている場合、プロセスの開始時にエラーが発生します。 ◆ 空白は使用しないでください。 ◆ グループ名に「port」という言葉は使用しないでください。ただし、グループ名の一部として「port」の文字列を使用することはできます。 ◆ fin1、fin10 など、グループ名の末尾に数値を使用しないでください。数値は、1_fin、1fin など、グループ名の冒頭には使用できます。
ADD EXTRACT データ・ソース・オプション	
SOURCEISTABLE	<p>Oracle GoldenGate ダイレクト・ロード方法または SQL*Loader へのダイレクト・バルク・ロード方法を使用して、データベースから初期ロードのためにレコード全体を抽出する Extract タスクを作成します。SOURCEISTABLE を指定しない場合、ADD EXTRACT はオンライン変更同期プロセスを作成します。この場合は、他のデータ・ソース・オプションを 1 つ指定する必要があります。</p> <p>SOURCEISTABLE を使用するときは、サービス・オプションを指定しないでください。タスク・パラメータは、パラメータ・ファイルに指定する必要があります。</p> <p>初期ロード方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>
TRANLOG [<bsds name>]	<p>データ・ソースとしてトランザクション・ログを指定します。このオプションは、Teradata 以外のすべてのデータベースで使用します。TRANLOG を指定するときは、BEGIN オプションを指定する必要があります。</p> <p>z/OS システム上の DB2 では、<bsds name> オプションを使用してトランザクション・ログの BSDS (ブートストラップ・データ・セット) ファイル名を指定します。指定する BSDS 名が、Extract プロセスの接続先の DB2 インスタンスのものであることを確認してください。Oracle GoldenGate は、BSDS 指定の検証を実行しません。</p>

引数	説明
VAM	(Teradata) <i>Vendor Access Module</i> (VAM) と呼ばれる Extract API を Teradata Access Module (TAM) とのインタフェースとして指定します。このオプションは、Teradata データベースで使します。
EXTFILESOURCE <file name>	データ・ソースとして抽出ファイルを指定します。このオプションは、プライマリ Extract グループとターゲット・システム間の仲介の役割を果たすセカンダリ Extract グループ (データ・ポンプ) とともに使します。 <file name> には、ファイルの完全修飾パス名を指定します (例: c:\ggs\dirdat\extfile)。
EXTTRAILSOURCE <trail name>	データ・ソースとしてトレイルを指定します。このオプションは、プライマリ Extract グループとターゲット・システム間の仲介の役割を果たすセカンダリ Extract グループ (データ・ポンプ) とともに使します。 <trail name> には、トレイルの完全修飾パス名を指定します (例: c:\ggs\dirdat\aa)。
VAMTRAILSOURCE <VAM trail name>	(Teradata) VAM トレイルを指定します。このオプションは、Teradata 最大保護モードを使用しているときに使します。プライマリ Extract グループの書き込み先 VAM トレイルの完全修飾パス名を指定します。VAM トレイルの読取り、およびターゲット・システムへのデータの送信には、VAM ソート Extract グループを使します。

ADD EXTRACT サービス・オプション

```
BEGIN
{NOW | yyyy-mm-dd
[hh:mi:[ss[.cccccc]]]}
```

データ・ソース内の処理開始位置のタイムスタンプを指定します。

有効な値:

- ◆ NOW
- ◆ 次のフォーマットの日付と時刻:
yyyy-mm-dd [hh:mi:[ss[.cccccc]]]

NOW の意味:

DB2 LUW 以外のすべてのデータベースの場合、NOW では ADD EXTRACT コマンドの発行時刻を指定します。

DB2 LUW の場合 NOW では START EXTRACT を有効にする時刻を指定します。この日時にほぼ一致する最初のレコードが開始位置になります。これは、DB2 LUW ではトランザクションのコミットおよびアポート・レコードがタイムスタンプを含む唯一のログ・レコードで、これらのタイムスタンプとの比較しか開始位置を計算する方法がないためです。これは、Oracle GoldenGate が使用する API の制限によるものです。

データ・ポンプ Extract では、ADD EXTRACT 文より前にトレイルに取得されたデータをバイパスするときを除き、NOW を使しないでください。

引数	説明
	<p>SQL Server のトランザクション・ログのタイムスタンプによる開始位置の決定</p> <p>時刻で開始位置を決定する場合、SQL Server の次の制約によって影響を受けます。</p> <ul style="list-style-type: none"> ◆ SQL Server トランザクション・ログに記録されているタイムスタンプは、3.3333 マイクロ秒 (ms) の粒度を使用します。このレベルの粒度では、2 つのトランザクションが同じ 3.333 ミリ秒の時間間隔で開始する場合、2 つのトランザクションの時間差に基づいて開始位置を決定できないことがあります。 ◆ タイムスタンプは、すべての SQL Server ログ・レコードではなく、トランザクションを開始してコミットしたレコード、およびデータが含まれないその他の一部のレコードのみに記録されます。 ◆ SQL Server のタイムスタンプは、システム・クロックではなく、使用中の各プロセッサ固有の内部クロックに基づいています。このクロックは 1 秒間に数回更新されますが、更新から更新までの間にシステム・クロックとの同期が外れることがあります。その場合、時刻で決定する開始位置の正確さがさらに低下します。 ◆ ログ・バックアップ・ファイルに記録されるタイムスタンプは、バックアップ内に記録された時刻と正確に一致しない可能性があります（ただし、不正確さは 1 秒未満です）。 <p>LSN を使用した位置決めの方が正確です。<LSN <value> を参照してください。</p>
EXTSEQNO <seqno>, EXTRBA <relative byte address>	<p>Oracle と NonStop SQL/MX のプライマリ Extract、およびデータ・ポンプ Extract に有効です。次のいずれかを指定します。</p> <ul style="list-style-type: none"> ◆ Oracle REDO ログの順序番号と、そのログ内のデータ取得開始位置の RBA。 ◆ NonStop SQL/MX TMF オーディット・トレイルの順序番号と、そのファイル内のデータ取得開始位置の相対バイト・アドレス。この 2 つの要素によって、TMF マスター・オーディット・トレイル (MAT) の位置を指定します。 ◆ (データ・ポンプの場合) トレイル内のデータ取得開始ファイル。順序番号を指定しますが、埋め込みのために 0 を使用しないでください。たとえば、トレイル・ファイルが c:\ggs\dirdat\aa000026 の場合は、EXTSEQNO 26 と指定します。このオプションを使用しない場合、処理はデフォルトでトレイルの最初から開始されます。 <p>このオプションを使用する前に Oracle サポートに連絡してください。詳細は、http://support.oracle.com を参照してください。</p>
EXTRBA <relative byte address>	<p>z/OS 上の DB2 に有効です。トランザクション・ログ内のデータ取得開始位置の相対バイト・アドレスを指定します。</p>
LOGNUM <log number>, LOGPOS <byte offset>	<p>c-tree に有効です。c-tree トランザクション・ログ内のデータ取得開始位置の場所を指定します。</p> <ul style="list-style-type: none"> ◆ <log number> は、c-tree ログ・ファイルの番号です。 ◆ <byte offset> は、ファイルの初めからの相対位置です（基準は 0 です）。

引数	説明
EOF	SQL Server に有効です。ログ・ファイルの末尾（次のレコードの書き込み位置）から処理を開始するように構成します。アクティブなトランザクションは取得されません。
LSN <value>	<p>SQL Server に有効です。SQL Server トランザクション・ログ内のデータ取得開始位置の LSN を指定します。ログ・バックアップまたはオンライン・ログ内に存在する LSN を指定する必要があります。このオプションの別名は EXTLN です。</p> <p>SQL Server では、データベースが返す方法に応じて、LSN は次のいずれかで構成されます。</p> <ol style="list-style-type: none"> 1. 0X00000d7e:0000036b:01bd のように、接頭辞 0X と先行ゼロを付けたコロン区切り 16 進数文字列（8:8:4） 2. 000003454:0000000875:00445 のように、先行ゼロを付けたコロン区切り 10 進数文字列（10:10:5） 3. 0Xd7e:36b:1bd のように、接頭辞 0X を付けた先行ゼロなしのコロン区切り 16 進数文字列 4. 3454:875:445 のように、先行ゼロなしのコロン区切り 10 進数文字列 5. 3454000000087500445 のような 10 進数文字列 <p>上記で、最初の値は仮想ログ・ファイル番号、2 番目の値は仮想ログ内のセグメント番号、3 番目の値はエントリ番号です。</p> <p>次のような問合せを使用すると、名前付きトランザクションの LSN を検索できます。</p> <pre>select [Current LSN], [Transaction Name], [Begin Time] from fn_dblog(null, null) where Operation = 'LOP_BEGIN_XACT' and [Begin Time] >= <time></pre> <p>特定のトランザクションの開始時刻を確認した後、関連する LSN を検索し、開始時刻が同じ 2 つのトランザクションの間に開始位置を指定できます。</p>
EOF LSN <value>	<p>DB2 LUW に有効です。Extract 起動時のトランザクション・ログ内のデータ取得開始位置を指定します。</p> <ul style="list-style-type: none"> ◆ EOF では、ログ・ファイル内のアクティブ LSN から処理を開始するように構成します。アクティブ LSN は、ログ・ファイルの末尾位置（次のレコードの書き込み位置）です。アクティブなトランザクションは取得されません。 ◆ LSN <value> では、指定した LSN に有効なログ・レコードが存在する場合に、そこから処理を開始するように構成します。指定した LSN が存在しない場合、Extract は異常終了します。Extract は指定した LSN に配置されますが、Extract がその LSN を最初に処理するとはかぎらないことに注意してください。これは、ログ・ファイル内に、DB2 内部ログ・レコードなど Extract が無視する多数のタイプのレコードが存在するためです。Extract は、実際に最初に処理した LSN を Extract レポート・ファイルにレポートします。

引数	説明
PAGE <data page>, ROW <row>	<p>Sybase に有効です。Sybase トランザクション・ログ内の開始位置を定義するデータ・ページと行を指定します。開始位置は、指定した PAGE および ROW に最も近い位置、またはその位置から開始するトランザクションの最初のレコードである必要があります。Extract レポートには、次の位置が表示されます。</p> <ul style="list-style-type: none"> ◆ Positioning To: PAGE および ROW で指定されたレコードの位置。 ◆ Positioned To: Positioning To の位置またはそれ以降の位置で見つかった最初の BEGIN レコードの位置。 ◆ First Record Position: Positioning To の位置またはそれ以降の位置で見つかった最初の有効なレコードの位置。
PARAMS <parameter file>	<p>Extract パラメータ・ファイルの保管場所を、Oracle GoldenGate ディレクトリ内のデフォルトの <code>dirprm</code> 以外に指定します。完全修飾パス名を使用します。</p>
REPORT <report file>	<p>Extract レポート・ファイルの保管場所を、Oracle GoldenGate ディレクトリ内のデフォルトの <code>dirrpt</code> 以外に指定します。完全修飾パス名を使用します。</p>
THREADS <n>	<p>Oracle RAC クラスター・データベース構成からデータを抽出する場合に、REDO スレッド数を指定します。</p>
PASSIVE	<p>この Extract グループをパッシブ・モードで実行し、ターゲット・システム上の別名 Extract グループを起動または停止することによってのみ起動および停止できるように指定します。ソースおよびターゲット間の接続は、このグループではなく、ターゲットの別名 Extract グループによって確立されます。</p> <p>このオプションは、標準 Extract グループまたはデータ・ポンプ Extract グループに対して使用できます。これは、ネットワークを介してターゲット上のリモート・トレイルにデータを送信するソース・システム上の Extract に対してのみ使用する必要があります。</p> <p>パッシブおよび別名 Extract グループの構成方法は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>
DESC "<description>"	<p>グループの説明 ("Extracts account_tab on Serv1" など) を指定します。この説明は引用符で囲む必要があります。短縮キーワード DESC または完全なキーワード DESCRIPTION を使用できます。</p>
RMTHOST {<host name> <IP address>}	<p>このグループを別名 Extract として識別し、リモート・ホストの DSN 名またはその IP アドレスを指定します。</p>
{MGRPORT <port>} {PORT <port>}	<p>別名 Extract に対して使用します。次のいずれかを指定します。</p> <ul style="list-style-type: none"> ◆ MGRPORT では、Manager を実行しているリモート・システム上のポートを指定します。このオプションは、動的 Collector を使用しているときに使用します。 ◆ PORT では、静的 Collector ポートを指定します。このオプションは、静的 Collector を実行している場合にのみ、MGRPORT のかわりに使用します。
RMTNAME <name>	<p>別名 Extract に対して使用します。別名 Extract の名前と異なる場合に、パッシブ Extract の名前を指定します。</p>

- 例 1** 次の例では、トランザクション・ログからデータベースを抽出する、"finance" という名前の Extract グループを作成します。抽出は、ADD EXTRACT によってこのグループが作成された時刻に生成されたレコードから開始されます。

```
ADD EXTRACT finance, TRANLOG, BEGIN NOW
```

- 例 2** 次の例では、Oracle RAC ログからデータベースの変更を抽出する、"finance" という名前の Extract グループを作成します。抽出は、このグループが作成された時刻に生成されたレコードから開始されます。RAC インスタンスが 4 つあるので、Extract スレッド数は 4 になります。

```
ADD EXTRACT finance, TRANLOG, BEGIN NOW, THREADS 4
```

- 例 3** 次の例では、トランザクション・ログからデータベースの変更を抽出する、"finance" という名前の Extract グループを作成します。抽出は、2006 年 1 月 31 日 8 時に生成されたレコードから開始されます。

```
ADD EXTRACT finance, TRANLOG, BEGIN 2006-01-31 08:00
```

- 例 4** 次の例では、最大パフォーマンス・モードまたは最大保護モードの Teradata TAM とインタフェースを取る、"finance" という名前の Extract グループを作成します。Teradata ソースには、BEGIN ポイントを使用しません。

```
ADD EXTRACT finance, VAM
```

- 例 5** 次の例では、"finance." という名前の VAM ソート Extract グループを作成します。このプロセスは、VAM トレイル /ggs/dirdat/vt から読み取ります。

```
ADD EXTRACT finance, VAMTRAILSOURCE /ggs/dirdat/vt
```

- 例 6** この例では、"finance." という名前のデータ・ポンプ Extract を作成し、Oracle GoldenGate トレイル c:\ggs\dirdat\lt から読み取ります。

```
ADD EXTRACT finance, EXTTRAILSOURCE c:\ggs\dirdat\lt
```

- 例 7** 次の例では、"load" という名前の初期ロード Extract を作成します。

```
ADD EXTRACT load, SOURCEISTABLE
```

- 例 8** 次の例では、トランザクション・ログからデータベースの変更を抽出する、"finance" という名前のパッシブ Extract グループを作成します。

```
ADD EXTRACT finance, TRANLOG, BEGIN NOW, PASSIVE
```

- 例 9** 次の例では、"financeA" という名前の別名 Extract グループを作成します。この別名 Extract は、ソースシステム sysA 上の "finance" という名前のパッシブ Extract に関連付けられます。このシステム上の Manager は、ポート 7800 を使用しています。

```
ADD EXTRACT financeA, RMTHOST sysA, MGRPORT 7800, RMTNAME finance
```

ALTER EXTRACT

ALTER EXTRACT は、次の目的で使用します。

- ADD EXTRACT コマンドで作成した Extract グループの属性を変更する
- 次の順序のトレイル・ファイルに切り替える

このコマンドを使用する前に、STOP EXTRACT <group name> コマンドで Extract を停止してください。

構文

```
ALTER EXTRACT <group name>
[, <ADD EXTRACT attribute>]
[, THREAD <number>]
[, ETROLLOVER]
```

引数	説明
<group name>	変更する Extract グループの名前。
<ADD EXTRACT attribute>	次の例外を除き、ADD EXTRACT コマンドで指定した任意の属性を変更できます。 <ul style="list-style-type: none">◆ EXTRACTSOURCE オプションで指定した Extract の変更。◆ THREADS オプションで指定した RAC スレッド数の変更。 このような例外に該当する場合は、対象の Extract グループを削除し、このグループをもう一度追加してください。 BEGIN オプションを使用している場合は、同じ文に他のオプションを指定しないでください。次の例のように、別の文を発行してください。 ALTER EXTRACT finance, BEGIN 2006-01-01 ALTER EXTRACT finance, ETROLLOVER
THREAD <number>	Oracle RAC 構成で、指定した REDO スレッドに対してのみ Extract を変更します。指定できるスレッド番号は1つのみです。
ETROLLOVER	Extract の再起動時に、次の順序のトレイル・ファイルに切り替えさせます。たとえば、現在のファイルが ET000002 の場合、Extract を再起動すると現在のファイルは ET000003 になります。

- 例 1** 次の例では、2006 年 1 月 1 日に処理を開始するように Extract の設定を変更します。
ALTER EXTRACT finance, BEGIN 2006-01-01
- 例 2** 次の例では、トレイル内の特定の場所から処理を開始するように Extract の設定を変更します。
ALTER EXTRACT finance, EXTSEQNO 26, EXTRBA 338
- 例 3** 次の例では、Oracle RAC環境のExtractを変更し、REDOスレッド4のみに新しい開始位置を適用します。
ALTER EXTRACT accounts, THREAD 4, BEGIN YYYY-MM-DD
- 例 4** 次の例では、特定の LSN から処理を開始するように SQL Server 環境の Extract の設定を変更します。
ALTER EXTRACT sales, LSN 3454:875:445
- 例 5** 次の例では、Extract の設定を変更して、順序の次のトレイル・ファイルに切り替えさせます。
ALTER EXTRACT finance, ETROLLOVER

CLEANUP EXTRACT

CLEANUP EXTRACT では、特定の Extract グループの実行履歴を削除します。このコマンドでは、最後の実行レコードがそのまま保持されるので、Extract は停止した場所から処理を再開できます。このコマンドを使用する前に、STOP EXTRACT コマンドを発行して Extract を停止してください。

構文 CLEANUP EXTRACT <group name> [, SAVE <count>]

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループの実行履歴が削除されます。
SAVE <count>	指定した数の最新のレコードを削除から除外します。

例 1 次の例では、最新レコード以外のすべてのレコードを削除します。

```
CLEANUP EXTRACT finance
```

例 2 次の例では、最新の 5 レコード以外のすべてのレコードを削除します。

```
CLEANUP EXTRACT *, SAVE 5
```

DELETE EXTRACT

DELETE EXTRACT では、Extract グループを削除します。このコマンドを実行すると、チェックポイント・ファイルは削除されますが、パラメータ・ファイルはそのまま残ります。グループの削除後、必要に応じてグループをもう一度作成することも、パラメータ・ファイルを削除することもできます。

関連するトレイル・ファイルを削除する場合は、オペレーティング・システムを通じて手動で行ってください。

このコマンドを使用する前に、STOP EXTRACT <group name> コマンドを発行して Extract を停止してください。

構文 DELETE EXTRACT <group name> [!]

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループが削除されます。
!	(感嘆符) プロンプトを表示せずに、ワイルドカードに一致するすべての Extract グループを削除します。

INFO EXTRACT

INFO EXTRACT では、次の情報を表示します。

- Extract のステータス (STARTING、RUNNING、STOPPED または ABENDED)
- おおよその Extract ラグ
- チェックポイント情報
- プロセス実行履歴
- Extract の書き込み先トレイル

TASKS または ALLPROCESSES 引数なしの基本コマンドでは、オンライン (継続的な) Extract プロセスの情報のみを表示します。タスクは除外されます。次に例を示します。

図 1

INFO EXTRACT

```
EXTRACT                                EXTCUST Last Started 2006-01-05 16:09 Status RUNNING
Checkpoint Lag                        00:01:30 (updated 97:16:45 ago)
Log Read Checkpoint File             /rdbsms/data/oradata/redo03a.log
                                      2006-01-05 16:05:17 Seqno 2952, RBA 7598080
```

タイム・ラグについて

タイム・ラグは、最新のチェックポイントがトレイルに書き込まれた時点での遅延 (秒) を表します。次に例を示します。

- 現在の時刻 = 15:00:00
- 最新のチェックポイント = 14:59:00
- 最新の処理レコードのタイムスタンプ = 14:58:00

この場合、ラグは 00:01:00 (1 分、14:58 と 14:59 の差異) とレポートされます。

UNKNOWN というラグ値は、プロセスは実行しているがレコードをまだ処理していないか、(タイム・ゾーンの差ではなくクロックが正確でないために) ソース・システムのクロックがターゲット・システムのクロックよりも進んでいることを示します。

より正確なラグ情報を取得するには、LAG EXTRACT を使用します (32 ページを参照してください)。

詳細情報の表示

次に、DETAIL オプションの出力の例を示します。

図 2

INFO EXTRACT と DETAIL

```
EXTRACT    ORAEXT    Last Started 2006-03-15 16:16    Status STOPPED
Checkpoint Lag      00:00:00 (updated 114:24:48 ago)
Log Read Checkpoint File C:\ORACLE\ORADATA\ORA920\REDO03.LOG
                  2006-03-15 16:17:53 Seqno 46, RBA 3757568
```

Target Extract Trails:

```
Remote Trail Name                               Seqno      RBA      Max MB
```

```

c:\goldengate802\dir\dat\xx          0  57465      10
c:\goldengate802\dir\dat\jm          0  19155      10

Extract Source                        Begin                End

C:\ORACLE\ORADATA\ORA920\REDO03.LOG  2006-03-15 16:07    2006-03-15 16:17
C:\ORACLE\ORADATA\ORA920\REDO03.LOG  2006-03-15 15:55    2006-03-15 16:07
C:\ORACLE\ORADATA\ORA920\REDO03.LOG  2006-03-15 15:42    2006-03-15 15:55
C:\ORACLE\ORADATA\ORA920\REDO03.LOG  2006-03-15 15:42    2006-03-15 15:42
Not Available                        * Initialized *     2006-03-15 15:42

Current directory    C:\GoldenGate802

Report file          C:\GoldenGate802\dir\rpt\ORAEXT.rpt
Parameter file       C:\GoldenGate802\dir\prm\ORAEXT.prm
Checkpoint file      C:\GoldenGate802\dirchk\ORAEXT.cpe
Process file         C:\GoldenGate802\dir\pcs\ORAEXT.pce
Error log            C:\GoldenGate802\gg\serr.log

```

チェックポイントの表示

正確なチェックポイントの位置は、データ・ソースの読取りチェックポイントとトレイルの書込みチェックポイントで構成されます。次に、SHOWCH オプションで表示されるチェックポイント情報の例を示します。ここでは、データ・ソースは **Oracle RAC** データベース・クラスタなので、出力にはスレッド情報も含まれています。過去のチェックポイントを表示するには、表示するチェックポイント番号を SHOWCH エントリに続いてを指定します。

図 3 INFO EXTRACT と SHOWCH

```

EXTRACT    JC108XT Last Started 2006-06-09 14:15      Status ABENDED
Checkpoint Lag      00:00:00 (updated 00:00:01 ago)
Log Read Checkpoint File /orarc/oradata/racq/redo01.log
                  2006-06-09 14:16:45 Thread 1, Seqno 47, RBA 68748800
Log Read Checkpoint File /orarc/oradata/racq/redo04.log
                  2006-06-09 14:16:19 Thread 2, Seqno 24, RBA 65657408

Current Checkpoint Detail:

Read Checkpoint #1

Oracle RAC Redo Log

Startup Checkpoint (starting position in data source):
  Thread #: 1
  Sequence #: 47
  RBA: 68548112
  Timestamp: 2006-06-09 13:37:51.000000
  SCN: 0.8439720
  Redo File: /orarc/oradata/racq/redo01.log

Recovery Checkpoint (position of oldest unprocessed transaction in data
source):
  Thread #: 1

```

Sequence #: 47
RBA: 68748304
Timestamp: 2006-06-09 14:16:45.000000
SCN: 0.8440969
Redo File: /orarc/oradata/racq/redo01.log

Current Checkpoint (position of last record read in the data source):

Thread #: 1
Sequence #: 47
RBA: 68748800
Timestamp: 2006-06-09 14:16:45.000000
SCN: 0.8440969
Redo File: /orarc/oradata/racq/redo01.log

Read Checkpoint #2

Oracle RAC Redo Log

Startup Checkpoint(starting position in data source):

Sequence #: 24
RBA: 60607504
Timestamp: 2006-06-09 13:37:50.000000
SCN: 0.8439719
Redo File: /orarc/oradata/racq/redo04.log

Recovery Checkpoint (position of oldest unprocessed transaction in data source):

Thread #: 2
Sequence #: 24
RBA: 65657408
Timestamp: 2006-06-09 14:16:19.000000
SCN: 0.8440613
Redo File: /orarc/oradata/racq/redo04.log

Current Checkpoint (position of last record read in the data source):

Thread #: 2
Sequence #: 24
RBA: 65657408
Timestamp: 2006-06-09 14:16:19.000000
SCN: 0.8440613
Redo File: /orarc/oradata/racq/redo04.log

Write Checkpoint #1

GGs Log Trail

Current Checkpoint (current write position):

Sequence #: 2
RBA: 2142224
Timestamp: 2006-06-09 14:16:50.567638
Extract Trail: ./dirdat/eh

Header:

```

Version = 2
Record Source = A
Type = 6
# Input Checkpoints = 2
# Output Checkpoints = 1

File Information:
Block Size = 2048
Max Blocks = 100
Record Length = 2048
Current Offset = 0

Configuration:
Data Source = 3
Transaction Integrity = 1
Task Type = 0

Status:
Start Time = 2006-06-09 14:15:14
Last Update Time = 2006-06-09 14:16:50
Stop Status = A
Last Result = 400

```

表 1 Extract チェックポイントについて

チェックポイント	説明
Read Checkpoint (読取りチェックポイント)	次に Extract の読取りチェックポイントの説明を示します。Extract は、データ・ソースに読取りチェックポイントを作成します。
Startup Checkpoint (開始チェックポイント)	プロセス起動時にデータ・ソースに最初に作成されたチェックポイント。この統計は次の情報で構成されます。
Thread #	Oracle GoldenGate が Oracle RAC 環境で実行されている場合は、このチェックポイントを作成した Extract スレッドの番号。それ以外の環境で実行されている場合、この統計は表示されません。
Sequence #	チェックポイントが作成されたトランザクション・ログの順序番号。
RBA	チェックポイントが作成されたレコードの相対バイト・アドレス。
Timestamp	チェックポイントの作成時のレコードのタイムスタンプ。
SCN	チェックポイントが作成されたレコードのシステム変更番号。
Redo File	チェックポイントが作成されたレコードを含むトランザクション・ログのパス名。

表 1 Extract チェックポイントについて (続き)

チェックポイント	説明								
	<p>Recovery Checkpoint (リカバリ・チェックポイント) Extract によってまだ処理されていない最も古いトランザクションを含むレコードのデータ・ソース内の位置。この統計のフィールドは、他のタイプの読取りチェックポイントに含まれるフィールドと同じです。</p> <p>Current Checkpoint (現在のチェックポイント) Extract が読み取った最新レコードのデータ・ソース内の位置。この値は、サマリー、およびオプションなしの基本の INFO EXTRACT コマンドで表示される Log Read Checkpoint 統計と同じになるはず。この統計のフィールドは、他のタイプの読取りチェックポイントに含まれるフィールドと同じです。</p>								
Write Checkpoint (書き込みチェックポイント)	<p>次に Extract の書き込みチェックポイントの説明を示します。Extract は、トレイルに書き込みチェックポイントを作成します。</p> <p>Current Checkpoint (現在のチェックポイント) Extract が現在書き込んでいるトレイル内の位置。この統計は次の情報で構成されます。</p> <table> <tr> <td>Sequence #</td><td>チェックポイントが書き込まれたトレイル・ファイルの順序番号。</td></tr> <tr> <td>RBA</td><td>チェックポイントが作成されたトレイル・ファイル内レコードの相対バイト・アドレス。</td></tr> <tr> <td>Timestamp</td><td>チェックポイントの作成時のレコードのタイムスタンプ。</td></tr> <tr> <td>Extract Trail</td><td>トレイルの相対パス名。</td></tr> </table>	Sequence #	チェックポイントが書き込まれたトレイル・ファイルの順序番号。	RBA	チェックポイントが作成されたトレイル・ファイル内レコードの相対バイト・アドレス。	Timestamp	チェックポイントの作成時のレコードのタイムスタンプ。	Extract Trail	トレイルの相対パス名。
Sequence #	チェックポイントが書き込まれたトレイル・ファイルの順序番号。								
RBA	チェックポイントが作成されたトレイル・ファイル内レコードの相対バイト・アドレス。								
Timestamp	チェックポイントの作成時のレコードのタイムスタンプ。								
Extract Trail	トレイルの相対パス名。								
その他の SHOWCH 出力	SHOWCH 出力の最後に表示される Header、File Information、Configuration および Status 統計は、Oracle サポートのアナリストが使用する情報です。これらの統計には、サポート・ケースの解決に有益な内部情報が含まれています。								
構文	<pre>ALTER EXTRACT <group name> [, SHOWCH [<n>]] [, DETAIL] [, TASKS ALLPROCESSES]</pre>								
引数	説明								
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループの情報が表示されます。								
SHOWCH [<n>]	<p>基本コマンドでは、現在の Extract チェックポイントの情報を表示します。</p> <p><n> には、現在のチェックポイントに加えて表示する過去のチェックポイントの番号を指定します。</p> <p>注意: 出力に不規則なインデントおよび空白が表示されることがあります。これは正常な出力で、情報の正確性に影響はありません。</p>								

引数	説明
DETAIL	次の情報を表示します。 ◆ データ・ソース内の開始位置および停止位置（時刻で表示）を含む Extract 実行履歴。 ◆ Extract の書き込み先トレイル。
TASKS	Extract タスクのみを表示します。ワイルドカード引数で指定したタスクは、INFO EXTRACT では表示されません。
ALLPROCESSES	タスクを含むすべての Extract グループを表示します。

例 1 INFO EXTRACT fin*, SHOWCH

例 2 INFO EXTRACT *, TASKS

KILL EXTRACT

KILL EXTRACT では、標準またはパッシブモードで実行している Extract プロセスを中断します。このコマンドは、STOP EXTRACT コマンドでプロセスを正常に停止できない場合にのみ使用してください。中断された Extract プロセスは、Manager プロセスによって再起動されません。

構文 KILL EXTRACT <group name>

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*). たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループが中断されます。

例 KILL EXTRACT finance

LAG EXTRACT

LAG EXTRACT では、Extract とデータ・ソース間の正確なラグ・タイムを確認します。LAG EXTRACT は、トレイル内のチェックポイントの位置を読み取るのではなく、Extract と直接通信するので、INFO EXTRACT よりも正確にラグ・タイムを計算できます。

Extract のラグについて

Extract では、ラグとは、Extract によって処理されたレコードの時間（システム・クロックに基づく）とデータ・ソースにおけるそのレコードのタイプスタンプ間の秒単位での差異です。

次に LAG EXTRACT の出力の例を示します。

図 4 LAG EXTRACT 出力

```
Sending GETLAG request to EXTRACT CAPTPCC...
Last record lag: 2 seconds.
At EOF, no more records to process.
```

構文 LAG EXTRACT <group name>

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループのラグ・タイムを確認できます。

例 1 LAG EXTRACT *

例 2 LAG EXTRACT *fin*

SEND EXTRACT

SEND EXTRACT では、実行中の Extract プロセスと通信します。このリクエストは、Extract がユーザーからのコマンドを受け入れる準備ができるとすぐに処理されます。

構文

```

SEND EXTRACT <group name>, {
  CACHEMGR {CACHESTATS | CACHEQUEUES | CACHEPOOL} |
  FORCESTOP |
  FORCETRANS <ID> [THREAD <n>] [FORCE] |
  GETLAG |
  GETTCPSTATS |
  LOGEND |
  REPORT |
  ROLLOVER |
  SHOWTRANS [<ID>] [THREAD <n>] [COUNT <n>]
    [DURATION <duration><unit>] [TABULAR]
    [FILE <name> [DETAIL]] |
  SKIPTRANS <ID> [THREAD <n>] [FORCE] |
  STATUS |
  STOP |
  TRACE[2] <tracefile> |
  TRACE[2] OFF |
  TRACEINIT |
  TRANLOGOPTIONS {PURGEORPHANEDTRANSACTIONS | NOPURGEORPHANEDTRANSACTIONS} |
  TRANLOGOPTIONS TRANCLEANUPFREQUENCY <minutes> |
  VAMMESSAGE "<Teradata command>" |
  VAMMESSAGE {"ARSTATS" | "INCLUDELIST [filter]" | "EXCLUDELIST [filter]"} |

```

```
VAMMESSAGE "OPENTRANS"
}
```

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループにこのコマンドが送信されます。コマンド送信先の Extract が実行中でない場合はエラーが返されます。
CACHEMGR {CACHESTATS CACHEQUEUES CACHEPOOL}	<p>Oracle GoldenGate メモリー・キャッシュ・マネージャについての統計を取得します。</p> <ul style="list-style-type: none"> ◆ CACHESTATS は、使用中の仮想メモリー、再割当てのために再利用されている仮想メモリー、匿名メモリー内の仮想メモリー、ページングされた仮想メモリーなど、仮想メモリーの総合的な使用状況についての統計を返します。また、キャッシュ・サイズ、バッファ割当てとページングに関連するしきい値、およびディスク上のキャッシュ・ファイルとその関連 I/O に関する統計も提供します。 ◆ CACHEQUEUES は、キュー（今後の使用のために再利用されるメモリー・バッファ）についての統計を返します。リクエストされたサイズのバッファが検出された回数、（仮想メモリーの新しいリクエストにつながった）適切なサイズのバッファが検出されなかった回数などの有益な統計を取得できます。 ◆ CACHEPOOL は、大半のトランザクション行データ用としてログ・リーダー・スレッド（Oracle RAC の場合は 1 つ以上、それ以外の場合は 1 つ）ごとに 1 つ、および BLOB とその他の大規模アイテム用に 1 つで構成されるキャッシュ・サブプールに関する統計を返します。各タイプのプールに実際に割り当てられるデータは、保持されているタイプのデータと一致しないことがあります。 <p>これらの統計は、CACHEMGR パラメータを調整する基準として使用できます。ただし、キャッシュ・マネージャは自動調整を行うため、通常はこれらの統計を表示したり、CACHEMGR を変更したりする必要はありません。サポート・ケースをオープンしていないかぎり、メモリー・キャッシュ・マネージャの調整は行わないでください。詳細は、http://support.oracle.com を参照してください。</p>
FORCESTOP	すべての通知をバイパスし、Extract を強制的に停止します。このコマンドを発行すると、プロセスは即座に停止します。
FORCETRANS <ID> [THREAD <n>] [FORCE]	<p>（オラクルのみ）Extract に、ID 番号で指定したトランザクションをコミット済トランザクションとしてトレイルに書き込ませます。ID 番号は、SHOWTRANS コマンドまたは Extract ランタイム・メッセージから取得してください。Extract は、このコマンドの発行後にトランザクションに追加されたデータはすべて無視します。</p>

引数	説明
	<p>オプション:</p> <ul style="list-style-type: none"> ◆ THREAD <n> では、異なるスレッドに重複するトランザクション ID がある場合に、RAC 環境でこのトランザクションを生成したスレッドを指定します。 ◆ FORCE では、確認プロンプトをバイパスします。 <p>FORCETRANS では、ソース・データベースに対するトランザクションのコミットは行いません。Replicat が（暗黙的なコミットで）処理できるように、既存データをトレイルのみに書き込ませます。</p> <p>FORCETRANS を使用するには、SHOWTRANS コマンドで表示されるトランザクション・リスト内の最も古いトランザクションを指定する必要があります。このコマンドは、古い順に他のトランザクションに対して繰り返し実行できます。</p> <p>FORCETRANS を使用した後、SEND EXTRACT コマンドを FORCESTOP とともに発行する場合は、少なくとも 5 分待機してください。待機しないと、トランザクションはそのまま残ります。</p> <p>Extract の起動直後に FORCETRANS を使用する場合、待機してからコマンドを再度実行するように求めるエラー・メッセージを受信することがあります。これは、Extract がまだ他のトランザクションを処理していないことを意味しています。Extract が他のトランザクションの処理を開始すると、トランザクションをトレイルに書き込ませることができます。</p>
GETLAG	<p>Extract とデータ・ソース間の正確なラグ・タイムを確認します。LAG EXTRACT (32 ページを参照してください) と同じ結果が返されます。</p>
GETTCPSTATS	<p>Extract とターゲット・システム間のネットワーク・アクティビティについての統計を表示します。次の統計が含まれます。</p> <ul style="list-style-type: none"> ◆ ローカルおよびリモート IP アドレス ◆ インバウンドおよびアウトバウンド・メッセージ (バイトおよびバイト/秒) ◆ 受信 (インバウンド) および送信 (アウトバウンド) 数。受信は、インバウンド・メッセージ当たり少なくとも 2 つ (長さについて 1 つ、データについて 1 つ以上) あります。 ◆ 送信および受信当たりの平均バイト数 ◆ 送信および受信の待機時間: 送信の待機時間とは、TCP への書き込み完了にかかる時間です。送信の待機時間が短いほど、ネットワークのパフォーマンスが優れています。受信の待機時間とは、読取り完了にかかる時間です。送信および受信の待機時間によって、おおよそのネットワークのラウンド・トリップ時間を把握できます。これらの値は、マイクロ秒で表されます。 ◆ データ圧縮のステータス (有効または無効)

引数	説明
	<p>圧縮が有効の場合、次の統計が表示されます。</p> <ul style="list-style-type: none"> ◆ 圧縮 CPU 時間 : 圧縮の実行に費やされた CPU 時間。 ◆ 圧縮時間 : CPU リソースの待機時間などを含む、圧縮に費やした合計時間。 ◆ 圧縮前バイト数と圧縮後バイト数 : 圧縮前と圧縮後と比較すると圧縮率（圧縮前と圧縮後のバイトの差）を把握できます。この圧縮率と、1 秒当たりの圧縮バイト数を比較することで、リソースおよびネットワーク消費の観点から圧縮率がコストに見合うかどうか判断できます。 <p>RMTHOST および RMTHOSTOPTIONS の TCPBUFSIZE オプションでは、圧縮前データの TCP バッファ・サイズを制御します。圧縮が有効の場合、ネットワークに送信されるデータはこのサイズよりも小さくなります。GETTCPSTATS では、圧縮後のスループットを表示します。</p>
LOGEND	Extract がデータ・ソース内のすべてのレコードを処理したかどうかを確認します。
REPORT	Extract レポート・ファイルに一時的な統計レポートを生成します。RESETREPORTSTATS NORESETREPORTSTATS とともに使用する場合、表示される統計は、STATOPTIONS パラメータの構成によって異なります。311 ページを参照してください。
ROLLOVER	Extract の再起動時に、順序の次のトレイル・ファイルに切り替えさせます。たとえば、現在のファイルが ET000002 の場合、このコマンドの実行後、現在のファイルは ET000003 になります。
SHOWTRANS [<ID>] [THREAD <n>] [COUNT <n>] [DURATION <duration><unit>] [TABULAR] [FILE <name> [DETAIL]]	<p>(Oracle のみ) オープンしているトランザクションについての情報を表示します。SHOWTRANS が表示する情報：</p> <ul style="list-style-type: none"> ◆ プロセス・チェックポイント (Extract が再起動する場合に、トランザクション処理を実行する必要がある最も古いログを示します) ◆ トランザクション ID ◆ Extract グループ名 ◆ REDO スレッド番号 ◆ (トランザクションの実際の開始時刻でなく) Oracle GoldenGate がトランザクションから抽出した最初の操作のタイムスタンプ ◆ システム変更番号 (SCN) ◆ REDO ログ番号と RBA ◆ ステータス (Pending COMMIT または Running)。Pending COMMIT は、FORCETRANS 発行後のトランザクション書込み中に表示されます。 <p>オプションを指定しない場合 SHOWTRANS では、25K バッファに収まるオープンしているすべてのトランザクションが表示されます。出力をさらに制御するには、次のオプションを参照してください。</p>

引数	説明
	<p>SHOWTRANS オプション:</p> <ul style="list-style-type: none"> ◆ <ID> では、コマンド出力を特定の 1 つのトランザクションに制限します。 ◆ THREAD <n> では、コマンド出力を特定の Oracle RAC スレッドに対してオープンしているトランザクションに制限します。<n> には、Extract に認識されているスレッド番号を指定します。 ◆ COUNT <n> では、コマンド出力を指定した数のオープンしているトランザクションに制限します（最も古いトランザクションから開始します）。有効な値は 1 ～ 100,000 です。 ◆ DURATION <duration><unit> では、コマンド出力を指定した時間より長くオープンしているトランザクションに制限します。 <duration> には、時間の長さを整数で指定します。 <unit> には、second、minutes、hours または days を完全なスペリングまたは短縮形式で指定します。 S SEC SECS SECOND SECONDS M MIN MINS MINUTE MINUTES H HOUR HOURS D DAY DAYS ◆ TABULAR では、SQL*Plus のデフォルトの表出力のような表フォーマットで出力を生成します。デフォルトは field-per-row です。 ◆ FILE <name> では、Extract に、指定したファイルにトランザクション情報を書き込ませます。コンソールへの出力はありません。 <p>16 進数およびプレーン文字のデータ・ダンプを書き込むには、FILE と DETAIL を使用します。これにより、メモリーからファイルにトランザクション全体がダンプされます。このデータを確認することにより、トランザクションをスキップするかトレイルに強制的に書き込むかを判別できます。</p> <p>注意: 基本の詳細情報は、WARNLONGTRANS CHECKINTERVAL パラメータに指定されている間隔でレポート・ファイルに自動的に書き込まれます。</p> <p>SHOWTRANS の詳細情報は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p> <p>SKIPTRANS <ID> [THREAD <n>] [FORCE]</p> <p>(オラクルのみ) Extract に、指定したトランザクションをスキップさせます。これにより、メモリーから現在のすべてのデータが削除され、後続のすべてのデータが無視されます。ID 番号は、SHOWTRANS コマンドまたは Extract ランタイム・メッセージから取得してください。</p> <p>THREAD <n> では、重複するトランザクション ID がある場合に、このトランザクションを生成した RAC 環境のスレッドを指定します。</p> <p>FORCE では、確認プロンプトをバイパスします。</p>

引数	説明
	<p>注意: SKIPTRANS を使用するには、SHOWTRANS で表示されるトランザクション・リストの中で最も古いトランザクションを指定する必要があります。このコマンドは、古い順に他のトランザクションに対して繰り返し実行できます。</p> <p>SKIPTRANS を使用した後、SEND EXTRACT コマンドを FORCESTOP とともに発行する場合は、少なくとも 5 分待機してください。待機しないと、トランザクションはそのまま残ります。</p>
STATUS	<p>処理状況の詳細なステータス（現在の位置およびアクティビティを含む）を返します。</p> <p>Current status 行には、次のような処理ステータス・メッセージが表示されます。</p> <ul style="list-style-type: none"> ◆ Delaying – 追加のデータを待機中 ◆ Processing data – データを処理中 ◆ Starting initial load – 初期ロード・タスクを開始中 ◆ Processing source tables – 初期ロード・タスクのデータを処理中 ◆ Reading from data source – データ・ソース（ソース表、トランザクション・ログなど）から読取り中 ◆ Adding record to transaction list – ファイル・メモリー・トランザクション・リストにレコードを追加中 ◆ At EOF (end of file) – 処理するレコードなし <p>異常終了イベント後の Extract リカバリ中は、上記のステータスに加え、次のようなステータスの注釈が表示されます。Extract はリカバリ中にログ読取り位置を継続的に変更するので、リカバリの進捗状況を追跡できます。</p> <ul style="list-style-type: none"> ◆ In recovery[1] – Extract は、トランザクション・ログのチェックポイントにリカバリ中です。 ◆ In recovery[2] – Extract は、チェックポイントからトレイルの末尾にリカバリ中です。 ◆ Recovery complete – リカバリが終了し、正常な処理が再開されます。 <p>次にデータベース・タイプごとの出力を示します。</p> <p>c-tree ログベースの抽出</p> <ul style="list-style-type: none"> ◆ グループ名およびプロセス ID ◆ 処理ステータス ◆ タイムスタンプ <p>DB2 LUW</p> <ul style="list-style-type: none"> ◆ グループ名およびプロセス ID ◆ 処理ステータス ◆ LSN ◆ タイムスタンプ

引数	説明
	z/OS 上の DB2 ログベースの抽出 <ul style="list-style-type: none">◆ グループ名およびプロセス ID◆ 処理ステータス◆ ログ RBA◆ タイムスタンプ◆ BSDS
	Oracle および Oracle RAC ログベースの抽出 <ul style="list-style-type: none">◆ グループ名およびプロセス ID◆ 処理ステータス◆ REDO スレッド番号 (RAC のみ)◆ REDO ログ順序番号◆ REDO ログの RBA◆ タイムスタンプ◆ SCN (RAC のみ)◆ REDO ログ名
	SQL Server ログベースの抽出 <ul style="list-style-type: none">◆ グループ名およびプロセス ID◆ 処理ステータス◆ タイムスタンプ
	Teradata、プライマリ Extract <ul style="list-style-type: none">◆ グループ名およびプロセス ID◆ 処理ステータス◆ タイムスタンプ
	Teradata VAM ソート Extract <ul style="list-style-type: none">◆ 処理ステータス◆ VAM トレイル順序番号◆ VAM トレイルの RBA◆ タイムスタンプ◆ VAM トレイル名
	すべてのデータベース、SOURCEISTABLE Extract タスク <ul style="list-style-type: none">◆ Extract 名およびプロセス ID◆ RMTTASK◆ レコード番号◆ タイムスタンプ◆ 表名
STOP	Extract を停止します。 (WARNLONGTRANS パラメータに基づいて) 長時間におよぶトランザクションがある場合は、次のメッセージが表示されます。

引数	説明
	<p>Sending GETLAG request to EXTRACT CAPTPCC...</p> <p>There are open, long-running transactions. Before you stop Extract, make the archives containing data for those transactions available for when Extract restarts. To force Extract to stop, use the SEND EXTRACT <group>, FORCESTOP command.</p> <p>Oldest redo log file necessary to restart Extract is: Redo Thread 1, Redo Log Sequence Number 150, SCN 31248005, RBA 2912272.</p>
TRACE[2] {<tracefile> OFF}	<p>トレースの有効化と無効化を切り替えます。トレースは、処理のボトルネックを明らかにする情報を、指定したファイルに取得します。</p> <ul style="list-style-type: none"> ◆ TRACE では、ステップバイステップの処理情報を取得します。 ◆ TRACE2 では、具体的なステップでなく、コード・セグメントを特定します。 ◆ OFF ではトレースを無効化します。 <p>トレースがすでに実行中の場合は、既存のトレース・ファイルが閉じ、<tracefile> で指定した新しいファイルにトレースが再開されます。</p> <p>トレースによって重大な処理のボトルネックが明らかになった場合は、Oracle サポートに連絡してください。詳細は、http://support.oracle.com を参照してください。</p>
TRACEINIT	<p>トレースの統計を 0 にリセットし、統計の累積をもう一度開始します。このオプションは、履歴でなく現在の処理動作をトレースするときに使用します。</p>
TRANLOGOPTIONS {PURGEORPHANEDTRANSACTIONS NOPURGEORPHANEDTRANSACTIONS}	<p>Oracle RAC に有効です。ノードに障害が発生し、Extract がロールバックを取得できないときに発生する孤立トランザクションのパージを有効化または無効化します。360 ページの「TRANLOGOPTIONS」も参照してください。</p>
TRANLOGOPTIONS TRANCLEANUPFREQUENCY <minutes>	<p>Oracle RAC に有効です。Oracle GoldenGate が、孤立トランザクションのスキャン、再スキャンによる確認、および削除の一連の動作を実行する間隔（分）を指定します。有効な値は 1 ～ 43200 分です。デフォルトは 10 分です。360 ページの「TRANLOGOPTIONS」も参照してください。</p>

引数	説明
<pre>VAMMESSAGE "<Teradata command>" VAMMESSAGE { "ARSTATS" "INCLUDELIST [filter]" "EXCLUDELIST [filter]" } VAMMESSAGE "OPENTRANS"</pre>	<p>Extract が使用している取得 API にコマンドを送信します。</p> <p><Teradata command> に指定可能な値：</p> <ul style="list-style-type: none"> ◆ "control:terminate" レプリケーション・グループを停止します。Teradata のレプリケーション・グループを削除または変更する前に実行する必要があります。 ◆ "control:suspend" レプリケーション・グループを一時停止します。Oracle GoldenGate をアップグレードするときに使用できます。 ◆ "control:resume" 一時停止したレプリケーション・グループを再開します。 ◆ "control:copy <database>.<table>" ソース・データベースからターゲット・データベースに表をコピーします。 <p>指定可能な SQL/MX コマンド：</p> <ul style="list-style-type: none"> ◆ "ARSTATS" TMF オーディット読取り統計を表示します。 ◆ "INCLUDELIST [filter]" Extract がオーディット・トレイル内でデータ・レコードを検出した、TABLE パラメータの選択基準に一致する表のリストを表示します。[filter] オプションでは、ワイルドカード・パターンを使用して返される表のリストをフィルタできます。 ◆ "EXCLUDELIST [filter]" Extract がオーディット・トレイル内でデータ・レコードを検出した、TABLE パラメータの選択基準に一致しない表のリストを表示します。[filter] オプションでは、ワイルドカード・パターンを使用して返される表のリストをフィルタできます。暗黙的に除外されている特定のシステム表は、常に除外表のリストに含まれます。 <p>このモジュールは、GGSCI にレスポンスを返します。レスポンスは ERROR または OK で、レスポンス・メッセージとともに返されます。</p> <p>指定可能な SQL Server コマンド： "OPENTRANS" オープンしているトランザクションのリスト（トランザクション ID、開始時刻、最初の LSN、含まれている操作数を含む）を出力します。</p>

例 1	SEND EXTRACT finance, ROLLOVER
例 2	SEND EXTRACT finance, STOP
例 3	SEND EXTRACT finance, VAMMESSAGE "control:suspend"
例 4	SEND EXTRACT finance, TRANLOGOPTIONS TRANSCLEANUPFREQUENCY 20
例 5	SEND EXTRACT finance, SKIPTRANS 5.17.27634 THREAD 2
例 6	SEND EXTRACT finance, SHOWTRANS COUNT 2

例 7 次に SHOWTRANS の出力を示します。

SHOWTRANS のデフォルト出力

Oldest redo log file necessary to restart Extract is:
Redo Thread 1, Redo Log Sequence Number 148, SCN 30816254, RBA 17319664

```
-----
XID                : 5.15.52582
Items              : 30000
Extract            : JC108XT
Redo Thread        : 1
Start Time         : 2006-05-18:12:51:27
SCN                : 20634955
Redo Seq           : 103
Redo RBA           : 18616848
Status             : Running
-----
```

```
-----
XID                : 7.14.48657
Items              : 30000
Extract            : JC108XT
Redo Thread        : 1
Start Time         : 2006-05-18:12:52:14
SCN                : 20635145
Redo Seq           : 103
Redo RBA           : 26499088
Status             : Running
-----
```

TABULAR が有効なときの SHOWTRAN 出力 (右側を切り捨てた表示)

XID	Items	Extract	Redo Thread	Start Time
5.15.52582	30000	JC108XT1		2006-05-18:12:52:14

SHOWTRANS FILE <name> DETAIL

Dumping transaction memory at 2006-07-21 13:36:54.

Record #1:

Header (140 bytes):

0:	0000 0A4A 0000 FFFF 0000 0000 0057 6C10	...J.....Wl.
16:	02FF 3F50 FF38 7C40 0303 4141 414E 5A77	..?P.8 @..AAANZw
32:	4141 4641 4141 4B6F 4941 4144 0041 4141	AAFAAAKoIAAD.AAA
48:	4E5A 7741 4146 4141 414B 6F49 4141 4400	NZwAAFAAAKoIAAD.
64:	4141 414E 5A77 414A 2F41 4142 7A31 7741	AAANZwAJ/AABz1wA
80:	4141 0041 4141 4141 4141 4141 4141 4141	AA.AAAAAAAAAAAAA
96:	4141 4141 4100 0000 0140 FF08 0003 0000	AAAAA....@.....
112:	0000 0000 0000 70FF 0108 FFFF 0001 4A53p.....JS
128:	554E 2E54 4355 5354 4D45 5200	UN.TCUSTMER.

Data (93 bytes):

0:	2C00 0400 0400 0000 0100 0200 0300 0000	,.....
16:	0000 0000 0800 0000 1800 0000 2000 0400
32:	1000 0600 0200 0000 284A 414E 456C 6C6F(JANElllo
48:	6352 4F43 4B59 2046 4C59 4552 2049 4E43	cROCKY FLYER INC
64:	2E44 454E 5645 5220 6E43 4F20 7365 7400	.DENVER nCO set.
80:	0000 0000 0000 0C00 0000 0000 00

SHOWTRANS の出力のサマリーを分析するときは、データベース上で現在実行中の（事前定義済バッファに収まる）すべてのトランザクションが表示されることを理解してください。Oracle GoldenGate の構成に含まれる表の操作が、今後別のトランザクションに追加されかどうか分からないため、Extract は構成されている表の操作を含むトランザクションだけでなく、オープンしているすべてのトランザクションを追跡する必要があります。

SHOWTRANS の Items フィールドの出力は、トランザクション内の操作数の合計ではなく、これまでに Oracle GoldenGate によって取得されたトランザクション内の操作数です。取得された操作に、構成に含まれる表への操作が含まれていない、またはその一部しか含まれていない場合は、Items の値は 0、または操作数の合計よりも少ない値になります。

Start Time フィールドには、トランザクション自体の実際の開始時刻ではなく、Oracle GoldenGate がトランザクションから抽出した最初の操作のタイムスタンプが表示されます。

注意 Oracle GoldenGate ソフトウェアは継続的に拡張されるため、実際のコマンド出力はこの例とは若干異なる場合があります。

START EXTRACT

START EXTRACT では、Extract プロセスを起動します。Extract が起動したことを確認するには、INFO EXTRACT または STATUS EXTRACT コマンドを使用します。

Extract は、特定の同期構成ではオペレーティング・システムのコマンドラインからも起動できます。目的に適した構成および起動方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

構文 START EXTRACT <group name>

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループが起動されます。

例 START EXTRACT finance

STATS EXTRACT

STATS EXTRACT では、1 つ以上の Extract グループの統計を表示します。出力には、Oracle GoldenGate 構成に含まれる DML および DDL 操作が含まれます。

最も正確な 1 秒当たりの処理操作数を取得するには、次のことを行います。

1. STATS EXTRACT コマンドを RESET オプションとともに発行します。
2. STATS EXTRACT REPORTRATE コマンドを発行します。LATEST STATISTICS フィールドに 1 秒当たりの操作数が表示されます。

図 5 LATEST および REPORTFETCH オプションを使用した出力の例

```

Sending STATS request to EXTRACT GGSEXT...

Start of Statistics at 2006-06-08 11:45:05.

DDL replication statistics (for all trails):
*** Total statistics since extract started ***
      Operations                      3.00
      Mapped operations                3.00
      Unmapped operations              0.00
      Default operations               0.00
      Excluded operations              0.00

Output to ./dirdat/aa:

Extracting from JDADD.EMPLOYEES to JDADD.EMPLOYEES:
*** Latest statistics since 2006-06-08 11:36:55 ***
      Total inserts                    176.00
      Total updates                    0.00
      Total deletes                    40.00
      Total discards                   0.00
      Total operations                 216.00

Extracting from JDADD.DEPARTMENTS to JDADD.DEPARTMENTS:
*** Latest statistics since 2006-06-08 11:36:55 ***
No database operations have been performed.
End of Statistics.

```

注意 DB2 データベース上で実行された実際の DML 操作数は、Oracle GoldenGate がレポートした抽出 DML 操作数と一致しないことがあります。DB2 は、行を物理的に変更しない UPDATE 文のログを取らないので、Oracle GoldenGate はこのような文を検出することや、統計に反映させることができません。

構文

```

STATS EXTRACT <group name>
[, <statistic>]
[, TABLE <table>]
[, TOTALONLY <table spec>]
[, REPORTFETCH | NOREPORTFETCH]
[, REPORTRATE <time units>]
[, ... ]

```

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が「T」から始まるすべての Extract グループの統計が返されます。
<statistic>	表示する統計。それぞれをコンマで区切ることによって複数の統計を指定できます (例: STATS EXTRACT finance, TOTAL, DAILY)。 有効な値: TOTAL プロセス起動からの合計を表示します。 DAILY 現在の日付の開始からの合計を表示します。 HOURLY 現在の時間の開始からの合計を表示します。 LATEST 最後の RESET コマンド発行後の合計を表示します。 RESET LATEST 統計フィールドのカウンタをリセットします。
TABLE <table>	指定した表、またはワイルドカード (*) で指定した表グループのみの統計を表示します。
TOTALONLY <table spec>	指定した表、またはワイルドカード (*) で指定した表グループの統計サマリーを表示します。
REPORTFETCH NOREPORTFETCH	フェッチ操作についての統計を出力に含めるかどうかを制御します。デフォルトは NOREPORTFETCH です。311 ページの「 STATOPTIONS 」も参照してください。
REPORTRATE <time units>	統計を絶対値ではなく処理レートとして表示します。 有効な値: ◆ HR ◆ MIN ◆ SEC

例 次の例では、特定の表について、合計と 1 分当たりの統計を表示し、最新の統計をリセットし、フェッチ統計を出力します。

```
STATS EXTRACT finance, TOTAL, HOURLY, TABLE acct,
REPORTRATE MIN, RESET, REPORTFETCH
```

STATUS EXTRACT

STATUS EXTRACT では、Extract が実行中かどうかを確認します。

構文 STATUS EXTRACT <group name> [, TASKS | ALLPROCESSES]

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が「T」から始まるすべての Extract グループのステータスが返されます
TASKS	Extract タスクのステータスのみを表示します。デフォルトでは、(ワイルドカードを使用せずに) 単一の Extract グループを指定している場合を除き、タスクは表示されません。
ALLPROCESSES	すべての Extract グループのステータス (タスクを含む) を表示します。

例 1 STATUS EXTRACT finance

例 2 STATUS EXTRACT fin*

STOP EXTRACT

STOP EXTRACT では、Extract を正常に停止します。このコマンドを実行すると、Extract の次の起動に備えて同期のステータスが保持され、Extract は Manager に自動的に起動されません。

STOP EXTRACT の発行時に長時間オープンしているトランザクションがある場合、Extract を再起動すると、このトランザクションに必要な最も古いトランザクション・ログ・ファイルを通知されることがあります。SEND EXTRACT の SHOWTRANS オプションを使用すると、このようなトランザクションの詳細とデータを表示できます。必要な場合には、SKIPTRANS または FORCETRANS オプションを使用し、トランザクションをスキップするか、コミット済トランザクションとしてトレイルに強制的に書き込みます。33 ページを参照してください。

構文 STOP EXTRACT <group name>

引数	説明
<group name>	1 つの Extract グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Extract グループが停止されます。

例 STOP EXTRACT finance

Replicat コマンド

Replicat コマンドでは、Replicat グループを作成および管理します。Replicat プロセスは、Extract プロセスによって抽出されたデータを読み取り、このデータをターゲット表に適用するか、ロード・アプリケーションなどの他のアプリケーションがこのデータを使用できるように準備を整えます。

コマンドの概要

ADD REPLICAT	LAG REPLICAT
ALTER REPLICAT	SEND REPLICAT
CLEANUP REPLICAT	START REPLICAT
DELETE REPLICAT	STATS REPLICAT
INFO REPLICAT	STATUS REPLICAT
KILL REPLICAT	STOP REPLICAT

ADD REPLICAT

ADD REPLICATでは、Replicatグループを作成します。SPECIALRUNが指定されている場合を除き、ADD REPLICATは実行間の処理の継続性を維持するためにチェックポイントを作成します。Replicatグループを作成する前に、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を確認してください。

Oracle GoldenGateGGSCI コマンド・インタフェースでは、Oracle GoldenGate Manager のインスタンスごとに最大 300 の Extract および Replicat グループの同時処理が完全にサポートされています。サポートされているレベルでは、すべてのグループが INFO および STATUS コマンドなどの GGSCI コマンドで完全に制御および表示可能です。サポートされているレベル以外では、グループ情報は表示されず、エラーが発生することがあります。Oracle GoldenGate では、お使いの環境を効率よく管理するため、Extract および Replicat グループ(合せて)の数が 300 以下に保持されるようにすることをお勧めします。

構文

```
ADD REPLICAT <group name>
{, SPECIALRUN |
  , EXTFILE <full path name> |
  , EXTTRAIL <full path name>}
[, BEGIN {NOW | YYYY-MM-DD HH:MM[:SS[.CCCCC]]} |
  , EXTSEQNO <seqno>, EXTRBA <rba>]
[, CHECKPOINTTABLE <owner.table> | NODBCHECKPOINT]
[, PARAMS <parameter file>]
[, REPORT <report file>]
[, DESC "<description>"]
```

引数	説明
<group name>	<p>Replicat グループ名。次の命名規則を使用します。</p> <ul style="list-style-type: none"> ◆ 下線 (_) などの英数字以外の文字を含む、最大 8 つの ASCII 文字を使用できます。オペレーティング・システムでファイル名に許可されているかぎり、任意の ASCII 文字を使用できます。この理由は、グループが関連するチェックポイント・ファイルで識別されるからです。 ◆ 次の ASCII 文字はファイル名には使用できません。 { \ / : * ? " < > } ◆ HP UX、Linux および Solaris では、ファイル名にコロン (:) またはアスタリスク (*) を使用できますが、推奨はされません。 ◆ 一般に、Oracle GoldenGate 内でグループ名の大 / 小文字は区別されません。たとえば、finance、Finance および FINANCE は、すべて同じであるとみなされます。ただし Linux 上では、グループ名 (ADD コマンドで明示的に定義されている場合は、そのパラメータ・ファイル名も含む) は、すべて大文字か、小文字である必要があります。グループ名およびパラメータ・ファイル名に大 / 小文字が混ざっている場合、プロセスの開始時にエラーが発生します。 ◆ 空白は使用しないでください。 ◆ グループ名に「port」という言葉は使用しないでください。ただし、グループ名の一部として「port」の文字列を使用することはできます。 ◆ fin1、fin10 など、グループ名の末尾に数値を使用しないでください。数値は、1_fin、1fin など、グループ名の冒頭には使用できます。
SPECIALRUN	<p>Replicat 特別実行をタスクとして作成します。SPECIALRUN、EXTFILE または EXTTRAIL を指定する必要があります。Extract が SPECIALRUN モードの場合は、GGSCI の START REPLICAT コマンドでは起動しないでください。特別実行の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>
ADD REPLICAT データ・ソース・オプション	
EXTFILE <full path name>	<p>Extract パラメータ・ファイルの RMTFILE で指定されている抽出ファイルの完全修飾名を指定します。</p>
EXTTRAIL <full path name>	<p>ADD RMTTRAIL または ADD EXTTRAIL コマンドで作成されたトレイルの完全修飾名を指定します。</p>
ADD REPLICAT サービス・オプション	
BEGIN <start point>	<p>トレイルの初期チェックポイントを定義します。</p> <ul style="list-style-type: none"> ◆ ADD REPLICAT でグループが作成されたときに変更のレプリケートを開始するには、NOW 引数を使用します。 ◆ 特定の時刻から変更の抽出を開始するには、日付および時刻フォーマットとして YYYY-MM-DD HH:MM[:SS[.CCCCC]] を使用します。

引数	説明
EXTSEQNO <seqno>	トレイル内のデータ処理開始ファイルの順序番号を指定します。順序番号を指定しますが、埋め込みのために 0 を使用しないでください。たとえば、トレイル・ファイルが c:\ggs\dir\dat\aa000026 の場合は、EXTSEQNO 26 と指定します。 このオプションを使用しない場合、処理はデフォルトでトレイルの最初から開始されます。EXTSEQNO を使用するには、EXTRBA も使用する必要があります。このオプションを使用する前に Oracle サポートに連絡してください。詳細は、 http://support.oracle.com を参照してください。
EXTRBA <rba>	EXTSEQNO で指定したトレイル・ファイル内の相対バイト・アドレスを指定します。このオプションを使用する前に Oracle サポートに連絡してください。詳細は、 http://support.oracle.com を参照してください。
CHECKPOINTTABLE <owner.table>	この Replicat グループが、データベース内の指定した表にチェックポイントを書き込むように指定します。所有者と表名を指定します（例：hr.hr_checkpoint）。この引数は、GLOBALS ファイルのすべてのデフォルトの CHECKPOINTTABLE 指定よりも優先されます。ADD CHECKPOINTTABLE コマンドを使用して、この表を追加する必要があります。
NODBCHECKPOINT	この Replicat グループがチェックポイント表にチェックポイントを書き込まないように指定します。この引数は、GLOBALS ファイルのすべてのデフォルトの CHECKPOINTTABLE 指定よりも優先されます。この引数は、作成中の Replicat グループとともにチェックポイントを使用しないときに指定する必要があります。
PARAMS <parameter file>	パラメータ・ファイルの保管場所を、Oracle GoldenGate ディレクトリ内のデフォルトの dirprm 以外に指定します。完全修飾名を使用します。
REPORT <report file>	処理レポート・ファイルの保管場所を、Oracle GoldenGate ディレクトリ内のデフォルトの dirrpt 以外に指定します。完全修飾名を使用します。
DESC "<description>"	グループの説明（"Loads account_tab on Serv2" など）を指定します。この説明は引用符で囲む必要があります。短縮キーワード DESC または完全なキーワード DESCRIPTION を使用できます。

例 ADD REPLICAT sales, EXTTRAIL d:\ggs\dir\dat\rt

ALTER REPLICAT

ALTER REPLICAT では、ADD REPLICAT コマンドで作成した Replicat グループの属性を変更します。このコマンドを使用する前に、STOP REPLICAT <group name> コマンドを発行して Replicat を停止してください。

構文 ALTER REPLICAT <group name> , <option> [, ...]

引数	説明
<group name>	変更する Replicat グループの名前。
<option>	ADD REPLICAT のオプションの 1 つを指定します。CHECKPOINT および NODBCHECKPOINT オプションを除き、ADD REPLICAT コマンドで構成した説明または任意のサービス・オプションを変更できます。

例 1 ALTER REPLICAT finance, EXTSEQNO 53
例 2 ALTER REPLICAT finance, EXTRBA 0
例 3 ALTER REPLICAT finance, BEGIN 2006-06-07 08:00:00

CLEANUP REPLICAT

CLEANUP REPLICAT では、特定の Replicat グループの実行履歴を削除します。このコマンドでは、最後の実行レコードがそのまま保持されるので、Replicat は停止した場所から処理を再開できます。

このコマンドを使用する前に、STOP REPLICAT <group name> コマンドを発行して Replicat を停止してください。

構文 CLEANUP REPLICAT <group name> [, SAVE <count>]

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループの実行履歴が削除されます。
SAVE <count>	指定した数の最新のレコードを削除から除外します。

例 1 次の例では、最新レコード以外のすべてのレコードを削除します。
 CLEANUP REPLICAT finance
例 2 次の例では、最新の 5 レコード以外のすべてのレコードを削除します。
 CLEANUP EXTRACT *, SAVE 5

DELETE REPLICAT

DELETE REPLICAT では、Replicat グループを削除します。このコマンドを実行すると、チェックポイント・ファイルは削除されますが、パラメータ・ファイルはそのまま残ります。グループの削除後、必要に応じてグループをもう一度作成することも、パラメータ・ファイルを削除することもできます。このコマンドを実行すると、(他のプロセスがファイルを読み取っていない場合) 削除されたグループが使用していたチェックポイントが削除されるので、Manager によってトレイル・ファイルをパージできます。

DELETE REPLICAT を使用する前に、次のことを行います。

1. Replicat を停止します。

```
STOP REPLICAT <group name>
```

2. このグループがデータベースのチェックポイント表を使用している場合、チェックポイント表からチェックポイントを削除できるように、DBLOGIN コマンドを使用してデータベースにログインします。

構文 DELETE REPLICAT <group name> [!]

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループが削除されます。
!	このオプションは、データベースのチェックポイント表からではなく、ディスク上のチェックポイント・ファイルからこのグループのチェックポイントを削除するときに使用します。このオプションを使用すると、ワイルドカードで複数のグループを指定するときに発生するプロンプトを無視できます。

例 DELETE REPLICAT finance

INFO REPLICAT

INFO REPLICAT では、Replicat グループの処理履歴を取得します。このコマンド出力には次の情報が含まれます。

- Replicat のステータス (STARTING、RUNNING、STOPPED または ABENDED)
- おおよそのレプリケーション・ラグ
- Replicat が読み取るトレイル
- トレイル内のチェックポイントを含む Replicat 実行履歴
- Replicat 環境に関する情報

TASKS または ALLPROCESSES 引数なしの基本コマンドでは、オンライン (継続的な) Replicat プロセスの情報のみを表示します。タスクは除外されます。

ラグの値について

ここでは、INFO REPLICAT によって表示されるラグの値について説明します。

Byte lag は、最新のチェックポイントの時点におけるトレイル・ファイル内の **Replicat** の読取り位置と、現在のファイル終了インジケータとの差（バイト）です。

Time lag は、最新のチェックポイントがトレイルに書き込まれた時点での遅延（秒）です。次に例を示します。

- 現在の時刻 = 15:00:00
- 最新のチェックポイント = 14:59:00
- 最新の処理レコードのタイムスタンプ = 14:58:00

この場合、ラグは 00:01:00（1 分、14:58 と 14:59 の差異）とレポートされます。

ラグの値が **UNKNOWN** の場合は、**Replicat** は実行しているがレコードをまだ処理していないか、（タイム・ゾーンの差ではなくクロックが正確でないために）ソース・システムのクロックがターゲット・システムのクロックよりも進んでいることを示します。より正確なラグ情報を表示するには、**LAG REPLICAT** を使用します（55 ページを参照してください）。

詳細情報の表示

詳細情報を表示するには、**DETAIL** オプションを使用します。次に出力の例を示します。

図 6 詳細な INFO REPLICAT 出力

```

REPLICAT   DELTPCC                Last Started 2006-01-21 11:40 Status RUNNING
Checkpoint Lag              00:00:00 (updated 232:39:41 ago)
Log Read Checkpoint File    C:\GGS\DIRDAT\RT000000
                               2006-01-11 18:54:33.000000 RBA 4735245

Extract Source              Begin                               End
C:\GGS\DIRDAT\RT000000      2006-01-11 18:54              2006-01-11 18:54
C:\GGS\DIRDAT\RT000000      * Initialized *              2006-01-11 18:54

Current directory           C:\GGS
Report file                  C:\GGS\dirrpt\DELTPCC.rpt
Parameter file               dirprm\DELTPCC.prm
Checkpoint file              C:\GGS\dirchk\DELTPCC.cpr
Checkpoint table             GG.CHECKPT
Process file                 C:\GGS\dirpcs\DELTPCC.pcr
Error log                    C:\GGS\ggserr.log
  
```

チェックポイントの表示

Replicat は、最新の読取り位置をマーク付けするために、トレイル・ファイルにチェックポイントを作成します。プロセス・チェックポイントを表示するには、**SHOWCH** オプションを使用します。基本コマンドでは、現在のチェックポイントを表示します。特定の番号の過去のチェックポイントを表示するには、番号の値を **SHOWCH** エントリに続いて入力します。

図 7 INFO REPLICAT, SHOWCH

```
REPLICAT   JC108RP   Last Started 2006-06-12 13:10   Status RUNNING
Checkpoint Lag      00:00:00 (updated 111:46:54 ago)
Log Read Checkpoint File ./dirdat/eh000000
                  First Record   RBA 3702915

Current Checkpoint Detail:

Read Checkpoint #1

GGS Log Trail

Startup Checkpoint(starting position in data source):
Sequence #: 0
RBA: 3702915
Timestamp: Not Available
Extract Trail: ./dirdat/eh

Current Checkpoint (position of last record read in the data source):
Sequence #: 0
RBA: 3702915
Timestamp: Not Available
Extract Trail: ./dirdat/eh

Header:
Version = 2
Record Source = A
Type = 1
# Input Checkpoints = 1
# Output Checkpoints = 0

File Information:
Block Size = 2048
Max Blocks = 100
Record Length = 2048
Current Offset = 0

Configuration:
Data Source = 0
Transaction Integrity = -1
Task Type = 0

Status:
Start Time = 2006-06-12 13:10:13
Last Update Time = 2006-06-07 21:23:31
Stop Status = A
Last Result = 400
```

表 2 Replicat チェックポイントについて

チェックポイント	説明
Startup Checkpoint (開始チェックポイント)	プロセス起動時にトレイルに最初に作成されたチェックポイント。この統計は次の情報で構成されます。
	Sequence # チェックポイントが書き込まれたトレイル・ファイルの順序番号。
	RBA チェックポイントが作成されたレコードの相対バイト・アドレス。
	Timestamp チェックポイントの作成時のレコードのタイムスタンプ。
	Extract Trail トレイルの相対パス名。
Current Checkpoint (現在のチェックポイント)	Replicat が読み取った最新レコードのトレイル内の位置。この値は、サマリー、およびオプションなしの基本の INFO REPLICAT コマンドに表示される Log Read Checkpoint 統計と同じになるはず。この統計のフィールドは、Startup Checkpoint に含まれるフィールドと同じです。
その他の SHOWCH 出力	SHOWCH 出力の最後に表示される Header、File Information、Configuration および Status 統計は、Oracle サポートのアナリストが使用する情報です。これらの統計には、サポート・ケースの解決に有益な内部情報が含まれています。

構文

```
INFO REPLICAT <group name>
[ , DETAIL]
[ , SHOWCH [<n>]]
[ , TASKS | ALLPROCESSES]
```

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループが表示されます。
DETAIL	詳細情報を表示します。
SHOWCH	チェックポイント・ファイルに記録済のチェックポイント、チェックポイント表に記録済のチェックポイント、使用中のチェックポイントなど、現在のチェックポイントの詳細を表示します。データベース・チェックポイントの表示には、表名、ハッシュ・キー（一意の識別子）、作成時のタイムスタンプが含まれます。 <n> には、現在のチェックポイントに加えて表示する過去のチェックポイントの数を指定します。
TASKS	Replicat タスクのみを表示します。ワイルドカード引数で指定したタスクは、INFO REPLICAT では表示されません。
ALLPROCESSES	タスクを含むすべての Replicat グループを表示します。

例 1 INFO REPLICAT *, DETAIL, ALLPROCESSES
 例 2 INFO REPLICAT *, TASKS
 例 3 INFO REPLICAT finance, SHOWCH

KILL REPLICAT

KILL REPLICAT では、Replicat グループを中断します。プロセスを中断すると、最新のチェックポイントが記録され、現在のトランザクションはデータベースによってロールバックされます。したがって、データを失うことなくプロセスを再開できます。中断された Replicat プロセスは、Manager プロセスによって再起動されません。このコマンドは、STOP REPLICAT コマンドで Replicat を正常に停止できない場合にのみ使用してください。

構文 KILL REPLICAT <group name>

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループが中断されます。

例 KILL REPLICAT finance

LAG REPLICAT

LAG REPLICAT では、Replicat とトレイル間の正確なラグ・タイムを確認します。LAG REPLICAT は、チェックポイントの位置を読み取るのではなく、Replicat と直接通信するので、INFO REPLICAT よりも正確にラグ・タイムを計算できます。

Replicat のラグについて

Replicat では、ラグとは、Replicat によって処理された最後のレコードの時間（システム・クロックに基づく）と証跡におけるそのレコードのタイプスタンプ間の秒単位での差異です。

構文 LAG REPLICAT <group name>

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループのラグが表示されます。

例 1 LAG REPLICAT *
 例 2 LAG REPLICAT *fin*

SEND REPLICAT

SEND REPLICAT では、起動中または実行中の Replicat プロセスと通信します。このリクエストは、Replicat がユーザーからのコマンドを受け入れる準備ができるとすぐに処理されます。

構文

```
SEND REPLICAT <group name>, {  
FORCESTOP |  
GETLAG |  
HANDLECOLLISIONS [<table spec>] |  
NOHANDLECOLLISIONS [<table spec>] |  
REPORT [HANDLECOLLISIONS [<table spec>]] |  
STATUS |  
STOP |  
TRACE[2] [DDLINCLUDE | DDL[ONLY]] [FILE] <file name> |  
TRACE[2] OFF |  
TRACEINIT  
}
```

引数	説明
<group name>	Replicat グループ名。この Replicat が実行中でない場合はエラーが返されます。
FORCESTOP	すべての通知をバイパスし、Replicat を強制的に停止します。このコマンドは、アクティブなトランザクションをロールバックし、このプロセスを即座に停止します。
GETLAG	Replicat とトレイル間の正確なラグ・タイムを表示します。ラグ・タイムは、Replicat によって最後のレコードが処理された時刻と、トレイル内のそのレコードのタイムスタンプとの差（秒）です。この結果は、LAG REPLICAT と同じです。
HANDLECOLLISIONS [<table spec>]	<p>HANDLECOLLISIONS パラメータを有効にします。このオプションを使用するかわりに、Replicat パラメータ・ファイルに HANDLECOLLISIONS パラメータを指定することもできます。HANDLECOLLISIONS は、ソース・データベースがアクティブな間に初期データ・ロードを実行するときに発生するエラーを自動処理するために使用します。初期ロードを完了し、ターゲット表にオンライン・データ変更を適用したら、(SEND REPLICAT またはパラメータ・ファイルからこのパラメータを削除することによって) 必ず HANDLECOLLISIONS を無効にしてください。詳細は、204 ページを参照してください。</p> <p><table spec> では、HANDLECOLLISIONS の適用を特定のターゲット表、または標準ワイルドカード (*) で指定したターゲット表グループに制限します。</p>
NOHANDLECOLLISIONS [<table spec>]	<p>HANDLECOLLISIONS パラメータを無効にします。ただしパラメータ・ファイルからは削除しません。次に Replicat を起動するときに HANDLECOLLISIONS を有効化しないようにするには、パラメータ・ファイルからこのパラメータを削除する必要があります。</p> <p><table spec> では、NOHANDLECOLLISIONS の適用を特定のターゲット表、または標準ワイルドカード (*) で指定したターゲット表グループに制限します。</p>

引数	説明
REPORT [HANDLECOLLISIONS [<table spec>]]	<p>Extract レポート・ファイルに一時的な統計レポートを生成します。</p> <p>RESETREPORTSTATS NORESETREPORTSTATS とともに使用する場合、表示される統計は、STATOPTIONS パラメータの構成によって異なります。368 ページを参照してください。</p> <p>HANDLECOLLISIONS では、HANDLECOLLISIONS が有効な表を表示します。<table spec> では、出力を特定のターゲット表、または標準ワイルドカード (*) で指定したターゲット表グループに制限します。</p>
STATUS	<p>トレイル内の現在の位置、および現在のトランザクションに関する情報を返します。次のフィールドが出力されます。</p> <ul style="list-style-type: none"> ◆ 処理ステータス ◆ トレイル・ファイル内の位置 ◆ トレイル連続番号 ◆ トレイルの RBA ◆ トレイル名 <p>次のような処理ステータス・メッセージが表示されます。</p> <ul style="list-style-type: none"> ◆ Delaying – 追加のデータを待機中 ◆ Waiting on deferred apply – DEFERAPPLYINTERVAL パラメータに基づいて処理待機中 ◆ Processing data – データを処理中 ◆ Skipping current transaction – START REPLICAT と SKIPTRANSACTION を使用している場合 ◆ Searching for START ATCSN <csn> – START REPLICAT と ATCSN を使用している場合 ◆ Searching for START AFTERCSN <csn> – START REPLICAT と AFTERCSN を使用している場合 ◆ Performing transaction timeout recovery – 現在の未完了のトランザクションを中止し、新しいトランザクション開始のために再配置中 (TRANSACTIONTIMEOUT パラメータを参照してください) ◆ Waiting for data at logical EOF after transaction timeout recovery – TRANSACTIONTIMEOUT 終了後に未完了のソース・トランザクションの残りの部分の受信を待機中 ◆ At EOF (end of file) – 処理するレコードなし
STOP	Replicat を正常に停止します。
TRACE[2] [DDLINCLUDE DDL[ONLY]] [FILE] <tracefile>	<p>トレースの有効化と無効化を切り替えます。トレースは、処理のボトルネックを明らかにする情報を、指定したファイルに取得します。</p> <ul style="list-style-type: none"> ◆ TRACE では、ステップバイステップの処理情報を取得します。 ◆ TRACE2 では、具体的なステップでなく、コード・セグメントを特定します。 <p>トレースがすでに実行中の場合は、既存のトレース・ファイルが閉じ、<tracefile> で指定したファイルにトレースが再開されます。</p> <p>トレースによって重大な処理のボトルネックが明らかになった場合は、Oracle サポートに連絡してください。詳細は、http://support.oracle.com を参照してください。</p> <p>トレースは、Replicat パラメータの TRACE および TRACE2 でも有効化できます。</p>

引数	説明
	DDLINCLUDE DDLONLY (Replicat のみ) DDL トレースを有効化し、トレース・ファイルに DDL トレースを取得する方法を指定します。 ◆ DDLINCLUDE では、トランザクション・データ処理の通常のトレースに加えて DDL トレースを取得します。 ◆ DDL[ONLY] では、トランザクション・データ処理の通常のトレースを除外し、DDL のみをトレースします。このオプションは、DDL に短縮できます。 [FILE] <file name> Oracle GoldenGate のトレース情報の記録先ファイルの完全修飾名。FILE キーワードはオプションですが、次の例のように、ファイル名の後に他のオプションを指定する場合は使用する必要があります。 SEND REPLICAT <group> TRACE FILE <file name> DDLINCLUDE 次の例のように、ファイル名の後に他のオプションを指定しない場合は、FILE キーワードを省略できます。 SEND REPLICAT <group> TRACE DDLINCLUDE <file name>
TRACE[2] OFF	トレースを無効にします。
TRACEINIT	トレースの統計を 0 にリセットし、統計の累積をもう一度開始します。このオプションは、履歴でなく現在の処理動作をトレースするときに使用します。
例 1	SEND REPLICAT finance, HANDLECOLLISIONS
例 2	SEND REPLICAT finance, REPORT HANDLECOLLISIONS fin_*
例 3	SEND REPLICAT finance, GETLAG

START REPLICAT

START REPLICAT では、Replicat を起動します。Replicat が起動したことを確認するには、INFO REPLICAT または STATUS REPLICAT コマンドを使用します。

Replicat の処理開始オプションについて

通常の開始位置

START REPLICAT (オプションなし) では、Replicat はデータ整合性を維持するために次のいずれかのポイントから処理を開始します。

- 正常または異常終了後：前回の実行で処理されなかった最後のトランザクション (現在のチェックポイントが示す位置)
- グループ作成後の最初の起動時：アクティブなトレイル・ファイルの初め (順序番号 0、RBA0)

別の開始位置

START REPLICAT の SKIPTRANSACTION、ATCSN および AFTERCSN オプションを使用すると、Replicat は通常の開始位置以外のトレイル内のトランザクションから処理を開始します。これらのオプションは、次の目的で使用します。

- Replicat のトレイル内の処理進行を妨げるエラー以降の論理リカバリ位置を指定する。問題のあるトランザクションをスキップする位置に Replicat を配置できますが、このようなトランザクションのデータがターゲットに適用されないことを認識して使用してください。
- 初期ロード実行中にレプリケートされたトランザクション変更の適用を開始する位置を指定する。トランザクションによってデータベース内のデータが変更されるたびに、データベース・エンジンはその時点でのデータの状態を表す変更識別子を割り当てます。Oracle GoldenGate の用語でコミット順序番号 (CSN) と呼ばれるこの識別子により、データベースは様々なトランザクションにわたるデータの状態の変化を追跡できます。バックアップ完了に対応する CNS を知っていれば、Replicat を起動し、レプリケートされたトランザクションをその位置以降から適用できます。したがって Replicat は、バックアップ以前の状態でレプリケートされた変更をバイパスできます。古いデータ変更をスキップすることで、重複レコードおよび行方不明レコードのエラーを回避できます。

注意 トランザクションをスキップするか、特定の CSN またはそれ以降から開始する場合は、適切なトランザクション・レコードに到達するまでに読み取る必要があるデータ量に応じて、Replicat の処理開始が通常より遅くなることがあります。処理開始の進捗状況を表示するには、SEND REPLICAT コマンドと STATUS オプションを使用します。

コマンドラインからの Replicat の起動

Replicat は、特定の同期構成ではオペレーティング・システムのコマンドラインからも起動できます。目的に適した構成および起動方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

構文

```
START REPLICAT <group name>  
[SKIPTRANSACTION | ATCSN <csn> | AFTERCSN <csn>]
```

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*). たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループが起動されます。
SKIPTRANSACTION	<p>Replicat に、起動時の位置以降の 1 番目のトランザクションをスキップさせます。この 1 番目のトランザクション内の操作は、すべて除外されます。</p> <p>この Replicat に対して MAXTRANSOPS パラメータも使用している場合は、トランザクションの途中からトレイル・ファイルの読み取りが開始されることがあります。この場合、この不完全なトランザクションの残りの部分はスキップされ、Replicat はファイル内の次の開始トランザクション・レコードから通常の処理を再開します。DISCARDFILE パラメータを使用している場合は、スキップされたレコードは破棄ファイルに書き込まれます。このパラメータを使用していない場合は、次のようなメッセージがレポート・ファイルに書き込まれます。</p> <pre>User requested START SKIPTRANSACTION. The current transaction will be skipped. Transaction ID <txid>, position Seqno <seqno>, RBA <rba></pre> <p>制約事項:</p> <ul style="list-style-type: none">◆ Replicat の読み取り先トレイルが、(チェックポイントを使用した) オンライン変更同期構成に含まれている場合にのみ有効です。初期ロードやバッチ変更同期実行 (Replicat パラメータ・ファイルまたは ADD REPLICAT で SPECIALRUN を使用する場合) では有効ではありません。

引数	説明
ATCSN <csn> AFTERCSN <csn>	<ul style="list-style-type: none"> ◆ ATCSN <csn> では、指定したコミット順序番号（CSN）を持つ開始トランザクション・インジケータを見つけるまで、Replicat にトレイル内のトランザクションをスキップさせます。このトランザクションおよび後続のトランザクションがターゲットに適用されます。指定した CSN より小さい番号を持つトランザクションはすべてスキップされます。 ◆ AFTERCSN <csn> では、指定した CSN を持つトランザクション以降の最初のトランザクションを見つけるまで、Replicat にトレイル内のトランザクションをスキップさせます。開始トランザクション・レコードに、指定した CSN 以下の番号が含まれているトランザクションは、すべてスキップされます。 <p><csn> の詳細は、この説明に続く CSN についての表を参照してください。CSN は、データベースにネイティブなフォーマットで指定する必要があり、それ以外のフォーマットの場合、Replicat は異常終了し、レポート・ファイルにメッセージを書き込みます。</p> <p>使用に適した CSN を確認するには、GGSCI の VIEW REPORT <group> コマンドで Replicat レポート・ファイルを表示します。正しい CSN の確認にさらに詳しい調査が必要な場合、経験が豊富な Oracle GoldenGate ユーザーは Logdump ユーティリティを使用できます。Logdump 使用の詳細は、『Oracle GoldenGate Windows and UNIX トラブルシューティングおよびチューニング・ガイド』を参照してください。</p> <p>ATCSN または AFTERCSN を使用する場合は、次のようなメッセージがレポート・ファイルに書き込まれます。</p> <pre>User requested start at commit sequence number (CSN) <csn-string></pre> <p>または</p> <pre>User requested start at commit sequence number (CSN) <csn-string></pre> <p>制約事項:</p> <ul style="list-style-type: none"> ◆ Replicat の読取り先トレイルが、(チェックポイントを使用した) オンライン変更同期構成に含まれている場合にのみ有効です。初期ロードやバッチ変更同期実行 (Replicat パラメータ・ファイルまたは ADD REPLICAT で SPECIALRUN を使用する場合) では有効ではありません。 ◆ 特定の CSN またはその直後からの開始をサポートするには、トレイルは、CSN がファイル・ヘッダーに格納される Oracle GoldenGate リリース 10.0.0 以降のものである必要があります。それ以前のリリースのトレイルの場合に、AFTERCSN を指定して Replicat を起動すると、Replicat は異常終了し、トレイル・フォーマットがサポートされていないことを通知するエラーをレポートに書き込みます。

注意 CSN で指定されたレコードが見つからない場合、Replicat はチェックポイントを発行するので、次のチェックポイントより前に発生する再開プロセスは、リクエストされた CSN よりも前の地点ではなく、リクエストされた場所から処理を開始できます。

表 3 Oracle GoldenGate データベースごとの CSN 値¹

データベース	CSN 値
c-tree	<p><log number>.<byte offset></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <log number> は、10 進数の c-tree ログ・ファイルの番号で、先行 0(ゼロ) が埋め込まれます。 ◆ <byte offset> は、ファイル (0 ベース) の先頭部分からの 10 進数の相対バイト位置で、先行 0(ゼロ) が埋め込まれます。 <p>例:</p> <p>0000000068.0000004682</p>
DB2 for i	<p>DB2 for i について、Oracle GoldenGate による抽出 (取得) はサポートされていないので、このデータベースの CSN はありません。</p>
DB2 LUW	<p><LSN></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <LSN> は、可変長、10 進数ベースの DB2 ログ・シーケンス番号です。 <p>例:</p> <p>1234567890</p>
DB2 z/OS	<p><RBA></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <RBA> は、トランザクション・ログ内にあるコミット・レコードの 6 バイトの相対バイト・アドレスです。 <p>例:</p> <p>1274565892</p>
MySQL	<p><LogNum>:<LogPosition></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <RBA> は、識別されるトランザクションの START TRANSACTION レコードが含まれているログ・ファイルの名前です。 ◆ <LogPosition> は、そのレコードのイベント・オフセット値です。イベント・オフセット値は、ログ・レコードのレコード・ヘッダー・セクションに格納されています。 <p>たとえば、ログ番号が 12 で、ログの位置が 121 の場合、CSN は次のようになります。</p> <p>000012:000000000000121</p>
Oracle	<p><system change number></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <system change number> は、Oracle SCN 値です。 <p>例:</p> <p>6488359</p>

表 3 Oracle GoldenGate データベースごとの CSN 値¹ (続き)

データベース	CSN 値
SQL/MX	<p><sequence number>.<RBA></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <sequence number> は、6 進数の NonStop TMF 監査証跡シーケンス番号で、先行 0(ゼロ) が埋め込まれます。 ◆ <RBA> は、そのファイル内における 10 進数の相対バイト・アドレスで、先行 0(ゼロ) が埋め込まれます。 <p>これら 2 つで、TMF マスター監査証跡 (MAT) 内の場所が指定されます。</p> <p>例:</p> <p>000042.0000068242</p>
SQL Server	<p>データベースの返し方に従い、次のいずれかになります。</p> <ul style="list-style-type: none"> ◆ コロンで区切られた 16 進数の文字列で、先行 0(ゼロ) および 0x 接頭辞が埋め込まれます (8:8:4)。 ◆ コロンで区切られた 10 進数の文字列で、先行 0(ゼロ) が埋め込まれます (10:10:5)。 ◆ 0x 接頭辞付き、先行 0(ゼロ) なしのコロンで区切られた 16 進数の文字列 ◆ 先行 0(ゼロ) なしのコロンで区切られた 10 進数の文字列 ◆ 10 進数の文字列 <p>条件:</p> <ul style="list-style-type: none"> ◆ 最初の値は仮想ログ・ファイルの番号で、2 番目は仮想ログ内のセグメント番号、3 番目はエントリ番号です。 <p>例:</p> <p>0X00000d7e:0000036b:01bd 0000003454:0000000875:00445 0Xd7e:36b:1bd 3454:875:445 3454000000087500445</p>
Sybase	<p><time_high>.<time_low>.<page>.<row></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <time_high> および <time_low> は、ログ・ページのインスタンス ID を示します。各データベースのログ・ページのヘッダーに格納されます。<time_high> は 2 バイトで、<time_low> は 4 バイトで、それぞれ先行 0(ゼロ) が埋め込まれています。 ◆ <page> は、データベースの論理ページ番号で、0(ゼロ) が埋め込まれます。 ◆ <row> は行番号で、0(ゼロ) が埋め込まれます。 <p>これらのコンポーネントをあわせて、ログ・ストリームの一意の場所が示されます。timestamp-high の 2 バイトの整数に有効な範囲は 0 ～ 65535 です。timestamp-low の 4 バイトの整数の場合は 0 ～ 4294967295 です。</p> <p>例:</p> <p>00001.0000067330.0000013478.00026</p>

表 3 Oracle GoldenGate データベースごとの CSN 値¹ (続き)

データベース	CSN 値
Teradata	<sequence ID> 条件: ◆ <sequence ID> は、汎用固定長の印刷可能シーケンス ID です。 例: 0x0800000000000000D700000021
TimesTen	TimesTen について、Oracle GoldenGate による抽出 (取得) はサポートされていないので、このデータベースの CSN はありません。

¹ Oracle、DB2 LUW および DB2 z/OS 以外のすべてのデータベース・プラットフォームには固定長の CSN があります。CSN は、必要に応じて先行 0 (ゼロ) が埋め込まれます。Sybase CSN など、複数のフィールドが含まれている CSN を各フィールド内に埋め込むことができます。

CSN の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

例 1 START REPLICAT finance

例 2 次の例では、Replicat を起動し、Oracle 固有の CSN から処理を開始します。

START REPLICAT finance, ATCSN 6488359

例 3 次の例では、Replicat を起動し、SQL Server 固有の CSN から処理を開始します。

START REPLICAT finance, AFTERCSN 0X000004D2:0000162E:0009

STATS REPLICAT

STATS REPLICAT では、1 つ以上の Replicat グループの統計を表示します。

構文

```
STATS REPLICAT <group name>
[, <statistic>]
[, TABLE <table>]
[, TOTALONLY <table spec>]
[, REPORTDETAIL | NOREPORTDETAIL]
[, REPORTRATE <time units>]
[, ... ]
```

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループの統計が表示されます。
<statistic>	表示する統計。コンマで区切ることによって複数の統計を指定できます (例: STATS REPLICAT finance, TOTAL, DAILY)。
	有効な値:
TOTAL	プロセス起動からの合計を表示します。
DAILY	現在の日付の開始からの合計を表示します。

引数	説明
	HOURLY 現在の時間の開始からの合計を表示します。
	LATEST 最後の RESET コマンド発行後の合計を表示します。
	RESET LATEST 統計フィールドのカウンタをリセットします。
TABLE <table>	指定した表、またはワイルドカード (*) で指定した表グループのみの統計を表示します。
TOTALONLY <table spec>	指定した表、またはワイルドカード (*) で指定した表グループの統計サマリーを表示します。
REPORTDETAIL NOREPORTDETAIL	衝突エラーが原因でレプリケートされなかった操作を出力に含めるかどうかを制御します。出力を有効にすると、このような操作が通常の統計（実行された INSERT、UPDATE および DELETE 操作）、および詳細な表示の統計でレポートされます。たとえば、10 レコードが INSERT 操作で、キーの重複が原因でこれらすべてのレコードが無視された場合、レポートには INSERT 操作数 10、衝突が原因で破棄された操作数 10 と出力されます。デフォルトは REPORTDETAIL です。311 ページの「STATOPTIONS」も参照してください。
REPORTRATE <time units>	統計を絶対値ではなく処理レートとして表示します。 有効な値： ◆ HR ◆ MIN ◆ SEC

例 次の例では、特定の表について、プロセス起動後および現在の時間の開始後の 1 分当たりの統計を表示し、最新の統計をリセットします。破棄された操作の統計はレポートされません。

```
STATS REPLICAT finance, TOTAL, HOURLY, TABLE acct,  
REPORTRATE MIN, RESET, NOREPORTDETAIL
```

STATUS REPLICAT

STATUS REPLICAT では、Replicat が実行中かどうかを確認します。

構文 STATUS REPLICAT <group name>
 [, TASKS]
 [, ALLPROCESSES]

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループのステータスが表示されます。
TASKS	Replicat タスクのステータスのみを表示します。デフォルトでは、(ワイルドカードを使用せずに) 単一の Replicat グループを指定している場合を除き、タスクは表示されません。

引数	説明
ALLPROCESSES	すべての Replicat グループのステータス（タスクを含む）を表示します。

例 1 STATUS REPLICAT finance

例 2 STATUS REPLICAT fin*

STOP REPLICAT

STOP REPLICAT では、Replicat を正常に停止します。このコマンドを実行する場合は、Replicat の次の起動に備えて同期のステータスが保持され、Replicat は Manager に自動的に起動されません。

構文 STOP REPLICAT <group name> [!]

引数	説明
<group name>	1 つの Replicat グループ名、または複数のグループを指定するワイルドカード (*)。たとえば、「T*」と指定すると、名前が T から始まるすべての Replicat グループが停止されます。
!	(感嘆符) Replicat を即座に停止します。トランザクションが中止され、プロセスは終了します。

例 STOP REPLICAT finance

ER コマンド

ER コマンドでは、複数の Extract および Replicat グループを 1 つのユニットとして制御します。ワイルドカードとともに使用して、その条件を満たすすべての Extract および Replicat グループにコマンドを適用します。

構文 <command> ER <group wildcard specification>

引数	説明
<command>	次のいずれかを指定できます。 INFO KILL LAG SEND START STATS STATUS STOP これらのコマンドの説明およびオプションのパラメータの詳細は、この章に記載されている同等の Extract または Replicat コマンドを参照してください。

引数	説明
<group wildcard specification>	このコマンドを適用するグループをワイルドカードで指定します。Oracle GoldenGate は、自動的に内部記憶域を拡大し、最大で 100,000 のワイルドカード・エントリを追跡します。

例 次の例では、名前に X が含まれる Extract および Replicat グループを起動した後、このグループを停止します。

```
GGSCI (ggs3) > START ER *X*
GGSCI (ggs3) > STOP ER *X*
```

トレイル・コマンド

トレイル・コマンドでは、Oracle GoldenGate トレイルを作成および管理します。トレイルは、ターゲットの場所に適用されるまで、Oracle GoldenGate が抽出データを一時的に保管する、ディスク上の一連のファイルです。

コマンドの概要

ADD EXTTRAIL	DELETE EXTTRAIL
ADD RMTTRAIL	DELETE RMTTRAIL
ALTER EXTTRAIL	INFO EXTTRAIL
ALTER RMTTRAIL	INFO RMTTRAIL

ADD EXTTRAIL

ADD EXTTRAIL では、ローカル・システム上にオンライン処理用のトレイルを作成し、次のことも行います。

- このトレイルに Extract グループを関連付ける
- 最大ファイル・サイズを指定する

構文

```
ADD EXTTRAIL <trail name>, EXTRACT <group name>
[, MEGABYTES <n>]
[SEQNO <n>]
```

引数	説明
<trail name>	トレイルの完全修飾パス名。トレイル名に含められる文字数は 2 文字のみです。Oracle GoldenGate は、新しいファイルが作成されるたびに、この名前に 6 桁の順序番号を追加します。たとえば、名前が /ggs/dirdat/tr のトレイルのファイル名は、/ggs/dirdat/tr000001、/ggs/dirdat/tr000002 のようになります。
<group name>	トレイルに関連付ける Extract グループ名。1 つのトレイルにデータを書き込めるのは、1 つの Extract プロセスのみです。
MEGABYTES <n>	トレイル内のファイルの最大サイズ (MB)。デフォルトは 10 です。

引数	説明
SEQNO <n>	トレイル内の最初のファイルのトレイル順序番号を、指定した数字から開始させます。先行ゼロは付けないようにしてください。たとえば、"tr" という名前のトレイルを 3 番から開始するには、SEQNO 3 と指定します。実際のファイル名は、/ggs/dirdat/tr000003 になります。このオプションは、トラブルシューティングのために、Replicat の読取り位置を特定の順序番号のトレイルに変更する必要があるときに使用できます。これにより、所定の順序番号を読み取るように Replicat の設定を変更する必要がなくなります。

例 ADD EXTTRAIL c:\ggs\dirdat\aa, EXTRACT finance, MEGABYTES 20

例 ADD EXTTRAIL /ggs/dirdat/tr000003

ADD RMTTRAIL

ADD RMTTRAIL では、リモート・システム上にオンライン処理用のトレイルを作成し、次のことも行います。

- 最大ファイル・サイズを指定する
- このトレイルに Extract グループを関連付ける

パラメータ・ファイルでは、リモート・システムおよび Manager プロセス用の TCP/IP ポートを識別するために、RMTTRAIL エントリの前に RMTHOST エントリを指定します。

構文 ADD RMTTRAIL <trail name>, EXTRACT <group name>
 [, MEGABYTES <n>]
 [SEQNO <n>]

引数	説明
<trail name>	トレイルの完全修飾パス名。実際のトレイル名に含められる文字数は 2 文字のみです。Oracle GoldenGate は、新しいファイルが作成されるたびに、この名前に 6 桁の順序番号を追加します。たとえば、名前が /ggs/dirdat/tr のトレイルのファイル名は、/ggs/dirdat/tr000001、/ggs/dirdat/tr000002 のようになります。
<group name>	トレイルに関連付ける Extract グループ名。1 つのトレイルにデータを書き込めるのは、1 つの Extract プロセスのみです。
MEGABYTES <n>	トレイル内のファイルの最大サイズ (MB)。デフォルトは 10 です。
SEQNO <n>	トレイル内の最初のファイルのトレイル順序番号を、指定した数字から開始させます。先行ゼロは付けないようにしてください。たとえば、"tr" という名前のトレイルを 3 番から開始するには、SEQNO 3 と指定します。実際のファイル名は、/ggs/dirdat/tr000003 になります。このオプションは、トラブルシューティングのために、Replicat の読取り位置を特定の順序番号のトレイルに変更する必要があるときに使用できます。これにより、所定の順序番号を読み取るように Replicat の設定を変更する必要がなくなります。

例 ADD RMTTRAIL c:\ggs\dirdat\aa, EXTRACT finance, MEGABYTES 20

例 ADD RMTTRAIL /ggs/dirdat/tr000003

ALTER EXTTRAIL

ALTER EXTTRAIL では、ADD EXTTRAIL コマンドで作成した（ローカル・システム上の）トレイルの属性を変更します。この変更は、次に **Extract** を起動するときに有効になります。

MEGABYTES オプションでは、トレイル・ファイルのサイズを変更できます。ファイル・サイズを変更するには、次の手順を実行します。

1. 次のコマンドを発行し、変更するトレイルのパス名と関連する **Extract** グループ名を表示します。ワイルドカードを使用して、すべてのトレイルを表示します。

```
INFO EXTTRAIL *
```

2. 次のコマンドを発行し、ファイル・サイズを変更します。

```
ALTER EXTTRAIL <trail name>, EXTRACT <group name>, MEGABYTES <n>
```

3. 次のコマンドを発行し、**Extract** にトレイル内の次のファイルに切り替えさせます。

```
SEND EXTRACT <group name>, ROLLOVER
```

構文

```
ALTER EXTTRAIL <trail name>, EXTRACT < group name>
[, MEGABYTES <n>]
```

引数	説明
<trail name>	トレイルの完全修飾パス名（例：c:\ggs\dirdat\aa）。
<group name>	トレイルに関連付ける Extract グループ名。
MEGABYTES <n>	ファイルの最大サイズ（MB）。デフォルトは 10 です。

例

```
ALTER EXTTRAIL c:\ggs\dirdat\aa, EXTRACT finance,
MEGABYTES 20
```

ALTER RMTTRAIL

ALTER RMTTRAIL では、ADD RMTTRAIL コマンドで作成した（リモート・システム上の）トレイルの属性を変更します。この変更は、次に **Extract** を起動するときに有効になります。

MEGABYTES オプションでは、トレイル・ファイルのサイズを変更できます。ファイル・サイズを変更するには、次の手順を実行します。

1. 次のコマンドを発行し、変更するトレイルのパス名と関連する **Extract** グループ名を表示します。ワイルドカードを使用して、すべてのトレイルを表示します。

```
INFO RMTTRAIL *
```

2. 次のコマンドを発行し、ファイル・サイズを変更します。

```
ALTER RMTTRAIL <trail name>, EXTRACT <group name>, MEGABYTES <n>
```

3. 次のコマンドを発行し、**Extract** にトレイル内の次のファイルに切り替えさせます。

```
SEND EXTRACT <group name>, ROLLOVER
```

構文 ALTER RMTTRAIL <trail name>, EXTRACT <group name>
 [, MEGABYTES <n>]

引数	説明
<trail name>	トレイルの完全修飾パス名（例：c:\ggs\dir dat\aa）。
<group name>	トレイルに関連付ける Extract グループ名。
MEGABYTES <n>	ファイルの最大サイズ（MB）。デフォルトは 10 です。

例 ALTER RMTTRAIL c:\ggs\dir dat\aa, EXTRACT finance,
 MEGABYTES 20

DELETE EXTTRAIL

DELETE EXTTRAIL では、ローカル・システム上のトレイルに関連付けられているチェックポイント・レコードを削除します。チェックポイントは、Oracle GoldenGate ディレクトリの dirchk サブディレクトリにある、グループと同じ名前のファイルに保持されています。

このコマンドでは、指定したトレイルに対するチェックポイント・ファイルからの参照のみが削除されます。トレイル・ファイル自体は削除されません。トレイル・ファイルを削除するには、オペレーティング・システムの標準のファイル削除コマンドを使用してください。

構文 DELETE EXTTRAIL <trail name>

引数	説明
<trail name>	2 文字の接頭辞を含むトレイルの完全修飾パス名。

例 DELETE EXTTRAIL /home/ggs/dir dat/et

DELETE RMTTRAIL

DELETE RMTTRAIL では、リモート・システム上のトレイルに関連付けられているチェックポイントのレコードを削除します。チェックポイントは、Oracle GoldenGate ディレクトリの dirchk サブディレクトリにある、Extract グループと同じ名前のファイルに保持されています。

このコマンドでは、指定したトレイルに対するチェックポイント・ファイルからの参照のみが削除されます。トレイル・ファイル自体は削除されません。トレイル・ファイルを削除するには、オペレーティング・システムの標準のファイル削除コマンドを使用してください。

構文 DELETE RMTTRAIL <trail name>

引数	説明
<trail name>	2 文字の接頭辞を含むトレイルの完全修飾パス名。

例 DELETE RMTTRAIL /home/ggs/dir dat/et

INFO EXTTRAIL

INFO EXTTRAIL では、ローカル・トレイルの構成情報を取得します。トレイル名、書き込み元の Extract、最新のデータ処理位置、割り当てられた最大ファイル・サイズが表示されます。

図 8 INFO EXTTRAIL の出力例

```
Extract Trail: c:\gg_81\dir.dat\md
Extract: GGSEXT8
Seqno: 2
RBA: 51080
File Size: 10M
```

構文 INFO EXTTRAIL <trail name>

引数	説明
<trail name>	トレイルの完全修飾パス名、または複数のトレイルを指定するワイルドカード。

例 1 INFO EXTTRAIL c:\ggs\dir.dat\aa

例 2 INFO EXTTRAIL *

INFO RMTTRAIL

INFO RMTTRAIL では、リモート・トレイルの構成情報を取得します。トレイル名、書き込み元の Extract、最新のデータ処理位置、割り当てられた最大ファイル・サイズが表示されます。

図 9 INFO RMTTRAIL の出力例

```
Extract Trail: /gg_81/dir.dat/rt
Extract: GGSEXT
Seqno: 4
RBA: 78066
File Size: 10M
```

構文 INFO RMTTRAIL <trail name>

引数	説明
<trail name>	トレイルの完全修飾パス名、または複数のトレイルを指定するワイルドカード。

例 1 INFO RMTTRAIL c:\ggs\dir.dat\aa

例 2 INFO RMTTRAIL *

パラメータ・コマンド

パラメータ・コマンドでは、Oracle GoldenGate パラメータ・ファイルの表示および管理を行います。

コマンドの概要

[EDIT PARAMS](#)

[SET EDITOR](#)

[VIEW PARAMS](#)

EDIT PARAMS

EDIT PARAMS では、パラメータ・ファイルを作成または変更します。デフォルトでは、このコマンドによって Windows システムのメモ帳または UNIX システムの vi エディタが起動されます。エディタを変更するには、SET EDITOR コマンドを使用します。

構文 EDIT PARAMS {MGR | <group> | <file name>}

引数	説明
MGR	Manager プロセスのパラメータ・ファイルを開きます。
<group>	指定した Extract または Replicat グループのパラメータ・ファイルを開きます。
<file name>	指定したファイルを開きます。完全パス名を使用します。

例 1 EDIT PARAMS finance

例 2 EDIT PARAMS c:\lpparms\replp.prm

SET EDITOR

SET EDITOR では、現在の GGSCI セッションのデフォルト・テキスト・エディタを変更します。Windows のデフォルト・エディタはメモ帳、UNIX は vi です。

構文 SET EDITOR <program name>

引数	説明
<program name>	テキストを ASCII フォーマットで保存できる任意のエディタ。

例 次の例では、デフォルト・エディタをワードパッドに変更します。

SET EDITOR wordpad

VIEW PARAMS

VIEW PARAMS では、パラメータ・ファイルの内容を表示します。

構文 VIEW PARAMS {MGR | <group> | <file name>}

引数	説明
MGR	Manager パラメータ・ファイルを表示します。
<group>	指定した Extract または Replicat グループのパラメータ・ファイルを表示します。
<file name>	指定したファイルを表示します。完全パス名を使用します。

例 1 VIEW PARAMS finance

例 2 VIEW PARAMS c:\lpparms\replp.prm

データベース・コマンド

データベース・コマンドでは、データベースと通信します。

コマンドの概要

[DBLOGIN](#)

[ENCRYPT PASSWORD](#)

[LIST TABLES](#)

DBLOGIN

DBLOGIN では、データベースを操作する他の Oracle GoldenGate コマンドの発行準備として、データベースへの接続を確立します。

構文

```
DBLOGIN
{SOURCEDB <dsn> |
USERID <user>[, PASSWORD <password>] [SYSDBA] |
SOURCEDB <dsn>, USERID <user>[, PASSWORD <password>][SQLID <sqlid>]}
```

引数	説明
SOURCEDB <dsn>	データソース名。ODBC を使用するすべてのデータベースで必要があります。

引数	説明
USERID <user> [, PASSWORD <password>]	データベース・ユーザーおよびこのユーザーのパスワード。データベースの資格証明が必要な場合に使用します。PASSWORD はオプションです。このオプションを使用すると、パスワードはエコーされます。PASSWORD を省略すると、Oracle GoldenGate はパスワードを求めるプロンプトを表示し、パスワードはエコーされません。パスワードに大文字と小文字の区別がある場合は、そのとおりに入力します。
SYSDBA	(Oracle) ユーザーを sysdba としてログインするように指定します。
SQLID <sqlid>	(z/OS 上の DB2) USERID のログインが (適切な場合は PASSWORD を使用して) 完了した後に、SQL コマンド SET CURRENT SQLID = 'sqlid' を発行します。SET コマンドが失敗すると、DBLOGIN コマンド全体がユニットとして失敗します。

例 1 DBLOGIN USERID ggs, PASSWORD ggs123

例 2 DBLOGIN SOURCEDB ctdb@host1, USERID ggs, PASSWORD ggs SYSDBA

ENCRYPT PASSWORD

ENCRYPT PASSWORD では、USERID および ASMUSERID パラメータで指定したデータベース・ログイン・パスワードを暗号化します。オプションを指定しない場合、ENCRYPT PASSWORD はランダムな暗号化鍵を生成しますが、ENCRYPTKEY オプションを使用すると、参照ファイルから鍵を指定できます。

ENCRYPT PASSWORD では、暗号化されたパスワードを画面に出力します。この出力をコピーして、USERID および ASMUSERID パラメータの PASSWORD 引数に貼り付けます。

パスワード暗号化は、SQL/MX データベースではサポートされていません。

構文 ENCRYPT PASSWORD <password>
[ENCRYPTKEY <keyname>]

引数	説明
<password>	ログイン・パスワード。パスワードは引用符で囲みません。パスワードに大文字と小文字の区別がある場合は、そのとおりに入力します。
ENCRYPTKEY <keyname>	(オプション) ENCKEYS 参照ファイルに含まれている暗号化鍵の論理名を指定します。Oracle GoldenGate は、この鍵名を使用してファイル内の実際の鍵を参照します。 <keyname> オプションを使用するには、この鍵を生成し、ローカル・システムの ENCKEYS 参照ファイルに保管します。詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

例 ENCRYPT PASSWORD ny14072 ENCRYPTKEY superkey2

LIST TABLES

LIST TABLES では、コマンド引数の指定に一致するデータベース内のすべての表を一覧表示します。このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

構文

LIST TABLES <table>

引数	説明
<table>	表名、またはワイルドカード (*) で指定した表グループ。

例

次に LIST TABLES コマンドおよび出力例を示します。

```
GGSCI (sysa) 3> list tables tcust*
TCUSTMER
TCUSTORD
```

Trandata コマンド

Trandata コマンドとログベースの抽出では、次のいずれかを行います。

- c-tree ファイルの REPLICATE 属性を有効化する
- Oracle 表のキー値（および必要に応じて他の列）のロギングを制御する
- Sybase 表の表レプリケーションを制御する
- SQL Server 表のサプリメンタル・ログ・データを追加する

コマンドの概要

[ADD TRANDATA](#)
[DELETE TRANDATA](#)
[INFO TRANDATA](#)

ADD TRANDATA

ADD TRANDATA では、Oracle GoldenGate が必要とするトランザクション情報をトランザクション・レコードから取得します。このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

ADD TRANDATA は、次に示すデータベースでのみ必要です。サポートされている他のデータベースでは、この機能がすでに存在しているか、またはデータベース・インタフェースを使用して構成する必要があります。トランザクション情報を使用可能にするための特別な要件の詳細は、ご使用のデータベース用の Oracle GoldenGate インストレーション・ガイドを参照してください。

c-tree データベース

ADD TRANDATA では、c-tree サーバーの停止と起動を伴わずに c-tree ファイルの REPLICATE 属性を有効化できます。REPLICATE は、サーバーを起動する前に c-tree サーバー構成ファイル ctsrvr.cfg の 1 つ以上の REPLICATE パラメータを設定する方法でも有効化できます。

DB2 データベース

ADD TRANDATA では、指定した表の DATA CAPTURE CHANGES を有効化します。このコマンドは、DB2 LUW および DB2 z/OS をサポートしています。デフォルトでは、ADD TRANDATA はデータベースに次のいずれかのコマンドを発行します。

DB2 z/OS:

```
ALTER TABLE <name> DATA CAPTURE CHANGES;
```

DB2 LUW:

```
ALTER TABLE <name> DATA CAPTURE CHANGES INCLUDE LONGVAR COLUMNS;
```

DB2 LUW では、ADD TRANDATA と EXCLUDELONG オプションを使用することで、LONGVAR 句を除外できます。

SQL Server データベース

ADD TRANDATA では、Oracle GoldenGate が SQL 操作を再構成するために必要とする拡張ロギング情報を取得します。SQL Server トランザクション・ログは、デフォルトでは十分な情報を提供しません。

Sybase データベース

ADD TRANDATA では、Sybase の sp_setreptable および sp_setrepcol システム・プロシージャを実行して、Sybase 表をレプリケーション対象としてマーク付けします。ADD TRANDATA オプションでは、データベース機能を使用して、データベースが指定した表の LOB データを伝播する方法を制御します。ADD TRANDATA オプションのリストを参照してください。

注意 ADD TRANDATA コマンドは、表に対して現在設定されている LOB 設定を上書きします。

Oracle データベース

ADD TRANDATA は、デフォルトで表レベルのサプリメンタル・ロギングを有効化します。サプリメンタル・ログ・グループには、次に示す優先順位で、次の列セットのいずれかが含まれます。

1. 主キー。
2. 仮想列、UDT、関数ベースの列および NULL 値が許可されている列を含まない、英数字で構成された最初の一意キー。
3. 仮想列、UDT および関数ベースの列を含まないが、NULL 値が許可されている列を含むことが可能な、英数字で構成された最初の一意キー。
4. 先行するキー・タイプがいずれも存在しない場合（表内で他のタイプのキーが定義されている場合でも）、仮想列、UDT、関数ベースの列、および Oracle GoldenGate 構成から明示的に除外されているタイプを除く、データベースによって一意キーで使用が許可されているすべての列の疑似キーが Oracle GoldenGate により構築されます。

このコマンドは、表に対して定義されている一意制約のタイプ（または一意制約のない場合）に適した、ADD SUPPLEMENTAL LOG DATA 句を含む ALTER TABLE コマンドを発行します。

ADD TRANDATA を使用する場合は、指定したロギング方法に応じて、次のいずれかの追加手順を実行する必要があります。

- **サプリメンタル・ロギング:** Oracle GoldenGate が主キーおよび連鎖行に対する更新を処理するためには、表レベルのロギングに加えて、データベース・レベルで最小サプリメンタル・ロギングを有効にする必要があります。これは Oracle の既知の問題で、Oracle GoldenGate ではなく、データベース・インタフェースを通じてこの操作を行う必要があります。データベース・レベルでサプリメンタル・ロギングが有効化されていることを検証するために、次の文を発行してください。

```
SELECT SUPPLEMENTAL_LOG_DATA_MIN FROM V$DATABASE;
```

- Oracle 9i では、この問合せの出力が YES になる必要があります。他の LOG_DATA オプションを有効化している場合、LOG_DATA_MIN は自動的に有効化されますが、確認のために検証を行ってください。
- Oracle 10g では、この問合せの出力が YES または IMPLICIT になる必要があります。Oracle 10g 以降は、他の LOG_DATA オプションを有効化していても自動的に有効化されないため、LOG_DATA_MIN を明示的に設定する必要があります。

Oracle に関する補足情報

Oracle のサプリメンタル・ロギングには、次の追加のオプションがあります。

- COLS オプションでは、必要に応じて非キー列をロギングします。
- NOKEY オプションでは、必要に応じてキー列のロギングを抑止します。

Oracle データベースの ADD TRANDATA を使用する際、次の点に注意してください。

- Oracle GoldenGate でのデータ抽出の開始後、ロギング詳細のいずれかを変更する場合、データを変更する前に、影響を受ける表から読み取られる Extract プロセスを停止して再起動する必要があります。
- ADD TRANDATA を使用してサプリメンタル・ログ・グループを作成する際、Oracle GoldenGate により、GG\$ の接頭辞の後に表の名前、アンダースコアおよびオブジェクト ID が付けられます。Oracle ではオブジェクト名は 30 文字に制限されているため、Oracle GoldenGate により、接尾辞およびオブジェクト ID が含まれるように必要に応じて長い表名が切り捨てられます。

8.0.2 以前のリリースの Oracle GoldenGate では、オブジェクト ID はログ・グループ名に使用されません。これらのリリースのいずれかからアップグレードする場合は、のいずれかを実行して DELETE TRANDATA または INFO TRANDATA コマンドが使用される際に正しい名前が検出されるようにします。

- 推奨される方法は、既存のログ・グループを新規の名前形式に変換することです。まず、DELETE TRANDATA を使用し、次に ADD TRANDATA を使用します。
- 現在の名前が維持されるようにするには、INFO TRANDATA および DELETE TRANDATA を使用する際に OLDFORMAT オプションを使用します。

構文

```
ADD TRANDATA <owner.table>
[, COLS (<column list>)]
[, INCLUDELONG | EXCLUDELONG]
[, LOBSNEVER | LOBSALWAYS | LOBSIFCHANGED | LOBSALWAYSNOINDEX]
[, NOKEY]
[, OLDFORMAT]
```

引数	説明
<owner.table>	トランザクション・データをロギングする表またはファイルの所有者および名前。表名にはワイルドカードを使用できますが、所有者名には使用できません。
COLS (<column list>)	サプリメンタル・ロギングに特定の非キー列を追加します。KEYCOLS 句に指定した列、およびフィルタ処理や操作のために必要となる列のロギングに使用でき、TABLE 文の FETCHCOLS 句を使用するときよりも効率的に列の値をフェッチできる場合があります。カンマを使用して複数の列を区切ります (例: NAME, ID, DOB)。
INCLUDELONG EXCLUDELONG	<p>(DB2 LUW) ADD TRANDATA によって発行される ALTER TABLE に "INCLUDE LONGVAR COLUMNS" 属性を含めるかどうかを制御します。デフォルトは INCLUDELONG です。ADD TRANDATA をこのオプションとともに発行すると、Oracle GoldenGate は次の文を発行します。</p> <pre>ALTER TABLE <name> DATA CAPTURE CHANGES INCLUDE LONGVAR COLUMNS;</pre> <p>EXCLUDELONG を使用すると、次のコマンドが発行されます。</p> <pre>ALTER TABLE <name> DATA CAPTURE CHANGES;</pre> <p>EXCLUDELONG を使用する場合、Oracle GoldenGate では LONGVAR 列を含む表のビフォア・イメージを必要とする機能がサポートされません。この機能の例として、GETUPDATEBEFORES、NOCOMPRESSUPDATES および NOCOMPRESSDELETES パラメータがあります。この機能をサポートするには、トランザクション・ログの LONGVAR 列の設定を変更し、列値のビフォアおよびアフター・イメージ両方を含めるようにする必要があります。</p>
LOBSNEVER LOBSALWAYS LOBSIFCHANGED LOBSALWAYSNOINDEX	<p>(Sybase) データベースが、指定した表の LOB データを伝播する方法を制御します。</p> <p>注意： ADD TRANDATA コマンドは、表に対して現在設定されている LOB 設定を上書きします。その後に設定を変更するには、sp_setrepcol スクリプトを使用する必要があります。</p> <ul style="list-style-type: none"> ◆ LOBSNEVER を使用すると、LOB データは伝播されません。ただし例外があり、LOB 列に NULL 値が挿入されるか、LOB 列が INSERT 操作でスキップされた場合、Extract はトレイル内のこの列に NULL データを書き込みます。 ◆ LOBSALWAYS では 2 つのことを行います: sp_setrepcol を使用して LOB レプリケーションを ALWAYS_REPLICATE (トランザクション内で変更されたかどうかにかかわらず、常に LOB データをレプリケートする) に設定し、(sp_setreptable の USE_INDEX オプションを使用して) レプリケーションで索引を使用するように表をマーク付けします。単一のトランザクション内で LOB がレプリケーション対象としてマーク付けされると長時間かかる可能性があるため、USE_INDEX を使用して各 LOB にグローバル・ノンクラスタード索引を作成してその時間を短縮します。グローバル・ノンクラスタード・インデックスの作成中、共有表ロックは保持されます。

引数	説明
	<ul style="list-style-type: none"> ◆ LOBSIFCHANGED では、トランザクション中に LOB データが変更された場合に LOB データのみをレプリケートします。これにより、レプリケーションのオーバーヘッドは低減されますが、レプリケーション環境外のターゲットで発生する可能性がある不整合は防止できません。これはデフォルトです。 ◆ LOBSALWAYSNOINDEX では、LOB レプリケーションを ALWAYS_REPLICATE (トランザクション内で変更されたどうかにかかわらず、常に LOB データをレプリケートする) に設定します。これによりオーバーヘッドは追加されますが、レプリケーション環境外のターゲットで発生する可能性がある不整合は防止できます。LOBSALWAYSNOINDEX では、レプリケーションの索引を使用するように表をマーク付けしません。この設定のメリットは、ADD TRANDATA の実行中にロックが保持されないことです。LOBSALWAYSNOINDEX は、バージョン 15 より前の Sybase データベースのデフォルトです。 <p>注意： ALWAYS_REPLICATE オプションを使用しているときに、特定の LOB 列に NULL 値が含まれている場合、表内の (LOB でない) 他の列が更新されると、ALWAYS_REPLICATE を有効化していても、この LOB は取得されません。</p> <p>表の LOB 設定を確認するには、その表に対して ADD TRANDATA を使用した後に、INFO TRANDATA コマンドを使用します。すべての LOB 列の LOB 設定が表示されます。必要に応じて、Sybase システム・プロシージャを使用して特定の列の LOB 設定を変更できます。</p>
NOKEY	主キー列のサブリメンタル・ロギングを抑止します。NOKEY を使用する場合は、COLS オプションを使用してキーとして機能できる別の列をロギングし、TABLE または MAP パラメータの KEYCOLS オプションを使用してこれらの列を代替キーとして指定します。
OLDFORMAT	<p>(Oracle) Oracle GoldenGate リリース 8.0.2 より前に使用されていたログ・グループのネーミング・フォーマットを使用します。このフォーマットは、ADD TRANDATA で作成されたログ・グループ名に GGS_ を追加しますが、オブジェクト ID は含まれません。</p> <p>Oracle の最大長の 30 文字を越えると、Oracle GoldenGate は表名を切り捨てます。(複数の表が類似する名前を持ち) 切り捨てられた表名が同じ文字列になった場合、識別用のオブジェクト ID がいないため、重複するログ・グループが存在することになります。このような場合は、INFO TRANDATA および DELETE TRANDATA が誤ったオブジェクトに適用される可能性があります。このオプションを使用するときは、必ず Oracle サポート・アナリストの指示を受けるようにしてください。詳細は、http://support.oracle.com を参照してください。</p>
例 1	<p>次の例では、Oracle 表の主キーのロギング、SQL Server 表の補足データのロギング、レプリケーション対象の Sybase 表のマーク付け、または c-tree ファイルの REPLICATE 属性の有効化のいずれかを行います。</p> <pre>ADD TRANDATA finance.acct</pre>
例 2	<p>次の Oracle の例では、主キー、および非キー列 name および address をロギングします。</p> <pre>ADD TRANDATA finance.acct, COLS (name, address)</pre>
例 3	<p>次の Oracle の例では、主キーはロギングせず、かわりに非キー列の name と pid ををロギングします。</p> <pre>ADD TRANDATA finance.acct, NOKEY, COLS (name, pid)</pre>

例 4 次の Sybase の例では、レプリケーション対象の acct 表にマーク付けし、トランザクション中に変更された場合にのみ LOB データをロギングするように指定します。

```
ADD TRANDATA finance.acct, LOBSIFCHANGED
```

DELETE TRANDATA

DELETE TRANDATA では、次のいずれかの操作を行います。

- c-tree: 指定したファイルのレプリケーションの無効化
- DB2 LUW および z/OS 上の DB2: 表を DATA CAPTURE NONE に変更
- Oracle: サプリメンタル・ロギングの無効化
- Sybase: レプリケーションの無効化
- SQL Server: 拡張ロギングの無効化

このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。このコマンドで指定するユーザーは、ADD TRANDATA の実行に必要な権限と同じ権限を持っている必要があります。

構文

```
DELETE TRANDATA <owner.table>
[, OLDFORMAT]
```

引数	説明
<owner.table>	表またはファイルの所有者および名前。ワイルドカードは、表名には使用できますが所有者名には使用できません。
OLDFORMAT	(Oracle) コマンドに、オブジェクト ID を考慮せず、表名のみを使用して検索させます。このオプションは、ログ・グループ名が 8.0.2 より前のリリースの Oracle GoldenGate によってサポートされているフォーマットの場合、または ADD TRANDATA を OLDFORMAT オプションとともに発行している場合に使用します。

例 1 DELETE TRANDATA finance.acct

例 2 DELETE TRANDATA finance.ac*

INFO TRANDATA

INFO TRANDATA では、次の情報を取得します。

- c-tree: レプリケーションが有効かどうかを確認します。
- DB2 LUW および z/OS 上の DB2: DATA CAPTURE が有効かどうかを確認します。
- Oracle: サプリメンタル・ロギングが有効かどうかを確認します。
- Sybase: レプリケーションが有効かどうか、およびすべての LOB 列が (ADD TRANDATA LOB オプションの指定と) 同じ設定を持つかどうかを確認します。
- SQL Server: 拡張ロギングが有効かどうかを確認します。

このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

構文 INFO TRANDATA <owner.table>
 [, OLDFORMAT]

引数	説明
<owner.table>	表示するトランザクション情報が含まれる表またはファイルの所有者および名前。 DBLOGIN コマンドで指定したユーザーと同じである場合、所有者の指定は不要です。表名にはワイルドカードを使用できますが、所有者名には使用できません。
OLDFORMAT	(Oracle) コマンドに、オブジェクト ID を考慮せず、表名のみを使用して検索させます。このオプションは、ログ・グループ名が 8.0.2 より前のリリースの Oracle GoldenGate によってサポートされているフォーマットの場合、または ADD TRANDATA を OLDFORMAT オプションとともに発行している場合に使用します。

例 1 INFO TRANDATA finance.acct

例 2 INFO TRANDATA finance.ac*

チェックポイント表コマンド

チェックポイント表コマンドでは、Oracle GoldenGate がトレイル内の Replicat の現在位置の追跡に使用するチェックポイント表を管理します。チェックポイント表の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

コマンドの概要

[ADD CHECKPOINTTABLE](#)
[CLEANUP CHECKPOINTTABLE](#)
[DELETE CHECKPOINTTABLE](#)
[INFO CHECKPOINTTABLE](#)

ADD CHECKPOINTTABLE

ADD CHECKPOINTTABLE では、ターゲット・データベースにチェックポイント表を作成します。Replicat は、リカバリで活用するために、この表を使用してトレイル内の読取り位置のレコードを保持します。

チェックポイント表はオプションです。チェックポイントは、ディスク内のファイルにも保持されます。チェックポイント表を使用すると、チェックポイントは Replicat トランザクションの一部になります。これにより Replicat は、特定の状況において、チェックポイントのみを使用するときよりも正確にリカバリできます。

GLOBALS の CHECKPOINTTABLE パラメータで 1 つの表を指定する場合、この表を 1 つの Oracle GoldenGate インスタンス内のすべての Replicat グループのデフォルト・チェックポイント表として使用できます。複数の Oracle GoldenGate インスタンス (複数のインストール) が同一のチェックポイント表を使用できます。Oracle GoldenGate は、別々のインスタンスに同一の名前の Replicat グループが存在する場合にもチェックポイントを追跡します。詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。この表の列の名前または属性は変更しないでください。ただし、表記憶域の属性は変更できます。

構文 ADD CHECKPOINTTABLE [<owner.table>]

引数	説明
<owner.table>	<p>作成するチェックポイント表の所有者および名前。この表をデフォルトのチェックポイント表として使用し、この表を GLOBALS ファイルの CHECKPOINTTABLE に指定している場合は、所有者と名前を省略できます。</p> <p>必須ではありませんが、Oracle GoldenGate 専用のスキーマでこの表を作成することをお薦めします。所有者と名前を指定しない場合、デフォルト表は GLOBALS パラメータ・ファイルの CHECKPOINTTABLE パラメータに基づいて作成されます。</p> <p>統計の表示やこの表の削除で必要になるため、この表の名前を記録しておいてください。</p>

例 1 次の例では、GLOBALS ファイルに指定されているデフォルト名でチェックポイント表を追加します。

```
ADD CHECKPOINTTABLE
```

例 2 次の例では、ユーザー定義名でチェックポイント表を追加します。

```
ADD CHECKPOINTTABLE ggs.fin_check
```

CLEANUP CHECKPOINTTABLE

CLEANUP CHECKPOINTTABLE では、(GGSCI が起動された) ワーキング Oracle GoldenGate ディレクトリ内に、関連するチェックポイント・ファイルが存在しないチェックポイント・レコードをチェックポイント表から削除します。このコマンドの目的は、グループの変更やファイルの移動によって不要になったチェックポイント・レコードを削除することです。

このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

構文 CLEANUP CHECKPOINTTABLE [<owner.table>]

引数	説明
<owner.table>	<p>クリーンアップするチェックポイント表の所有者および名前。所有者と名前を指定しない場合は、GLOBALS パラメータ・ファイルの CHECKPOINTTABLE パラメータで指定されている表が影響を受けます。</p>

例 CLEANUP CHECKPOINTTABLE ggs.fin_check

DELETE CHECKPOINTTABLE

DELETE CHECKPOINTTABLE では、データベースからチェックポイント表を削除します。このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

関連付けられている Replicat グループがアクティブな状態のときにチェックポイント表の使用を停止するには、次の手順を実行します。

1. GGSCI を実行します。

2. Replicat を停止します。

```
STOP REPLICAT <group>
```

3. Replicat グループを削除した後、次のコマンドを使用してこのグループをもう一度追加します。

```
DELETE REPLICAT <group>
ADD REPLICAT <group>, EXTTRAIL <trail>, NODBCHECKPOINT
```

4. GGSCI を終了し、もう一度起動します。

5. Replicat をもう一度起動します。

```
START REPLICAT <group>
```

6. DBLOGIN コマンドと適切な認証オプションを使用してデータベースにログインします。

```
DBLOGIN
{SOURCEDB <dsn> |
USERID <user>[, PASSWORD <password>] [SYSDBA] |
SOURCEDB <dsn>, USERID <user>[, PASSWORD <password>] [SQLID <sqlid>]
```

7. DELETE CHECKPOINTTABLE を使用してチェックポイント表を削除します。

Replicat が実行中でトランザクションが発生している間にチェックポイント表を削除すると、Replicat は異常終了し、チェックポイント表が見つからないことを示すエラーを返します。ただし、チェックポイントはディスク内のチェックポイント・ファイルに保持されています。処理を再開するには、チェックポイント表を同じ名前でもう一度追加します。トレイル内データのレプリケーションが再開されます。それでもチェックポイント表を削除する必要がある場合は、上記の推奨手順に従ってください。

構文

```
DELETE CHECKPOINTTABLE [<owner.table>] [!]
```

引数	説明
<owner.table>	削除するチェックポイント表の所有者および名前。GLOBALS ファイルの CHECKPOINTTABLE パラメータの指定と同じ場合、所有者と名前は省略できます。
!	表を削除するかどうかを確認するプロンプトをバイパスします。

例

```
DELETE CHECKPOINTTABLE ggs.fin_check
```

INFO CHECKPOINTTABLE

INFO CHECKPOINTTABLE では、チェックポイント表の存在を確認し、作成日時を表示します。次のようなメッセージが返されます。

```
Checkpoint table HR.CHKPT_TBLE created 2006-01-06 11:51:53.
```

このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

構文 INFO CHECKPOINTTABLE [<owner.table>]

引数	説明
<owner.table>	チェックポイント表の所有者および名前。GLOBALS ファイルの CHECKPOINTTABLE パラメータの指定と同じ場合、所有者と名前は省略できます。

例 INFO CHECKPOINTTABLE ggs.fin_check

Oracle トレース表コマンド

トレース表コマンドでは、Oracle データベースの双方向同期で使用する Oracle GoldenGate トレース表を管理します。Replicat は、各トランザクションの開始時にトレース表に操作を生成します。Extract は、トレース表上の操作で開始されるすべてのトランザクションを無視します。Replicat の操作を無視することで、ソース表とターゲット表間のデータのループバックを防止できます。

双方向同期の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

コマンドの概要

[ADD TRACETABLE](#)
[DELETE TRACETABLE](#)
[INFO TRACETABLE](#)

ADD TRACETABLE

ADD TRACETABLE では、Oracle データベースにトレース表を作成します。トレース表は、USERID パラメータで構成した Oracle GoldenGate Extract ユーザーのスキーマにある必要があります。

トレース表により、双方向同期構成で Replicat トランザクションがもう一度抽出されることを防止できます。

このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

証跡表には、次の記述が含まれます。

表 4 証跡表の記述

名前	NULL?	タイプ	説明
GROUP_ID	NOT NULL	VARCHAR2(8)	Replicat グループまたはバッチ・プロセスの名前。
DB_USER		VARCHAR2(30)	Replicat グループまたはバッチ・プロセスのユーザー ID。
LAST_UPDATE		DATE	トランザクションのタイムスタンプ。

構文 ADD TRACETABLE [<owner>.<table>]

引数	説明
<owner>.<table>	<p>(オプション) トレース表にデフォルトの GGS_TRACE と異なる名前を指定する場合にのみ必要です。所有者は、Extract パラメータ・ファイルの USERID パラメータで指定した所有者と同一である必要があります。</p> <p>デフォルト名を使用するには、この引数を省略します。可能なときは常にデフォルトの表名を使用してください。</p> <p>デフォルトの GGS_TRACE 以外のトレース表名を指定する場合は、Extract および Replicat パラメータ・ファイルの TRACETABLE パラメータを使用して指定します。パラメータ・ファイルで使用するときや、統計の表示およびこの表の削除で必要になるため、この表の名前を記録しておくってください。詳細は、359 ページの「TRACETABLE NOTRACETABLE」を参照してください。</p>

例 1 次の例では、デフォルト名 GGS_TRACE のトレース表を追加します。

```
ADD TRACETABLE
```

例 2 次の例では、ユーザー定義名 ora_trace を持つトレース表を追加します。

```
ADD TRACETABLE ora_trace
```

DELETE TRACETABLE

DELETE TRACETABLE では、トレース表を削除します。このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

構文 DELETE TRACETABLE [<owner.table>]

引数	説明
<owner.table>	<p>削除するトレース表の所有者および名前。所有者が USERID パラメータの指定と同じで、トレース表がデフォルト名の GGS_TRACE を持つ場合は、所有者と名前を省略できます。</p>

例 DELETE TRACETABLE ora_trace

INFO TRACETABLE

INFO TRACETABLE コマンドでは、指定したトレース表がデータベースのローカル・インスタンスに存在しているかどうかを検証します。表が存在する場合、Oracle GoldenGate は表の名前と作成日時を表示します。表が存在しない場合、Oracle GoldenGate はこの表が存在しないことを示すメッセージを表示します。このコマンドを使用する前に、DBLOGIN コマンドを使用してデータベース接続を確立してください。

構文 INFO TRACETABLE [<owner.table>]

引数	説明
<owner.table>	検証するトレース表の所有者および名前。所有者が USERID パラメータの指定と同じで、トレース表がデフォルト名の GG\$TRACE を持つ場合は、所有者と名前を省略できます。

例 INFO TRACETABLE ora_trace

DDL コマンド

次のコマンドでは、DDL レプリケーションを制御します。

DUMPDDL

Oracle GoldenGate DDL 履歴表にデータを表示するには、**DUMPDDL** コマンドを使用します。この情報は、**Extract** プロセスで使用される情報と同じです。固有形式で格納されますが、人間が読むことができる形式での画面へのエクスポート、または通常の **SQL** を使用して問合せ可能な一連の **SQL** 表にエクスポートできます。

この情報は、DDL 変更前トリガーによって提供されるため、DDL 変更前のオブジェクトの状態を表します。したがって、**CREATE** 操作のデータはありません。

注意 このビフォア・トリガーのデフォルト名は、**GG\$DDL_TRIGGER_BEFORE** です。

DUMPDDL を使用する前に、**DBLOGIN** コマンドを使用して、履歴表の所有者としてデータベースにログインしてください。

基本 DUMPDDL

基本 **DUMPDDL** コマンドでは、次の表にメタデータを出力します。

表 5 DUMPDDL 表

表	説明
GG\$DDL_OBJECTS	同期されている DDL 操作のオブジェクトに関する情報。SEQNO は主キーです。ここで説明する他のすべての表には、GG\$DDL_OBJECTS に対する外部キーとなる SEQNO 列が含まれます。
GG\$DDL_COLUMNS	DDL 同期に関連するオブジェクトの列に関する情報。
GG\$DDL_LOG_GROUPS	DDL 同期に関連するサブリメンタル・ログ・グループに関する情報。
GG\$DDL_PARTITIONS	DDL 同期に関連するオブジェクトのパーティションに関する情報。
GG\$DDL_PRIMARY_KEYS	DDL 同期に関連するオブジェクトの主キーに関する情報。

SEQNO 列は、Extract および Replicat レポート・ファイルに表示される DDL 順序番号です。これは、DDL 履歴表（デフォルト名 GGS_DDL_HIST）に問合せで取得することもできます。

これらの表はすべて、DDL オブジェクトのインストール中に Oracle GoldenGate DDL スキーマとして指定されたスキーマに所有されています（『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください）。これらの表の構造を表示するには、SQL*Plus の DESC コマンドを使用します。

DUMPDDL と SHOW

DUMPDDL と SHOW オプションでは、履歴表に含まれる情報を標準出力フォーマットで画面にダンプします。出力表は作成されません。DDL 履歴表のすべてのレコードが表示されます。

DUMPDDL に関する補足ガイドライン

DUMPDDL は常に DDL 履歴表のすべてのレコードをダンプします。特定のオブジェクトおよび操作に関する情報を表示するには、SQL 問合せまたは検索を使用して標準出力にリダイレクトします。履歴表には膨大な量のデータが含まれているため、効率的なパフォーマンスを維持するために、各 DDL 文の最初の約 4,000 バイトのみが表示されます。

メタデータのフォーマットは文字列ベースです。完全にエスケープされており、表または列名で標準以外の文字（=、?、* など）がサポートされています。また、マルチバイト・システム（中国語やドイツ語のように、UTF-8 キャラクタ・セットおよびマルチバイト・エンコーディングを使用）もサポートされています。

構文

DUMPDDL [SHOW]

引数	説明
SHOW	DDL 情報を標準出力フォーマットで画面にダンプします。

その他のコマンド

次のコマンドでは、Oracle GoldenGate のその他の様々な機能を制御します。

コマンドの概要

!command	OBEY
ALLOWNESTED	SHELL
CREATE SUBDIRS	SHOW
FC	VERSIONS
HELP	VIEW GGSEVT
HISTORY	VIEW REPORT
INFO ALL	

!command

! コマンドでは、以前に実行した GGSCI コマンドを、変更せずに実行します。再実行する前にコマンド

に変更を加えるには、FC コマンドを使用します（88 ページを参照してください）。以前に実行したコマンドのリストを表示するには、HISTORY コマンドを使用します（90 ページを参照してください）。

引数なしの ! では、最後に実行したコマンドを実行します。オプションを使用して行番号またはテキスト部分文字列を指定すると、以前に実行した任意のコマンドを実行できます。コマンド履歴はセッション間で保持されないため、以前に実行したコマンドは、現在の GGSCI セッション中に発行されたもののみ再実行できます。

構文 ! [<n> | -<n> | <string>]

引数	説明
<n>	指定した GGSCI 行からコマンドを実行します。各 GGSCI コマンド行には、セッションの最初を 1 として順序番号が付与されています。
-<n>	現在の行から <n> 行前に発行されたコマンドを実行します。
<string>	指定した文字列から始まる最新のコマンドを実行します。

例 1 ! 9
例 2 ! -3
例 3 ! sta

ALLOWNESTED

ALLOWNESTED では、ネストされた OBEY ファイルの使用を有効にします。ネストされた OBEY ファイルとは、他の OBEY ファイルが含まれる OBEY ファイルです。ネスト・レベルの最大数は 16 です。デフォルト・モードの NOALLOWNESTED で、ネストされた OBEY ファイルの実行を試みると、次のようなエラーが発生します。

```
ERROR: Nested OBEY scripts not allowed. Use ALLOWNESTED to allow nested scripts.
```

GGSCI セッションを終了すると、次の GGSCI セッションでは設定が NOALLOWNESTED に戻ります。

詳細は、92 ページの「OBEY」を参照してください。

構文

ALLOWNESTED | NOALLOWNESTED

CREATE SUBDIRS

CREATE SUBDIRS は、Oracle GoldenGate のインストール時に使用します。このコマンドでは、Oracle GoldenGate ホーム・ディレクトリ内にデフォルト・ディレクトリを作成します。CREATE SUBDIRS は、他の構成タスクを実行する前に使用してください。

構文

CREATE SUBDIRS

FC

FC では、以前に発行した GGSCI コマンドを表示して編集した後、このコマンドを再実行します。以前に実行したコマンドはメモリー・バッファに保持されているので、HISTORY コマンドを発行することによって表示できます（90 ページを参照してください）。

以前に実行したコマンドの表示

引数なしで FC を発行すると、最後に実行したコマンドが表示されます。オプションの行番号またはテキスト部分文字列を指定すると、以前に実行した任意のコマンドを実行できます。コマンド履歴はセッション間で保持されないため、以前に実行したコマンドは、現在の GGSI セッション中に発行されたもののみ編集できます。

コマンドの編集

FC を実行すると、指定したコマンドが表示された後、エディタが開き、2 つのドットで始まる空白行が表示されます。コマンドを編集するには、[Space] キーを使用して表示されているコマンド内の編集開始文字の下にカーソルを動かし、次のいずれかの引数を使用します。引数は大 / 小文字が区別されません。また引数は、組み合わせて使用できます。

表 6 FC エディタ・コマンド

引数	説明
i <text>	<p>テキストを挿入します。次に例を示します。</p> <pre>GGSCI (SysA) 24> fc 9 GGSCI (SysA) 24> send mgr GGSCI (SysA) 24.. i childstatus GGSCI (SysA) 24> send mgr childstatus</pre>
r <text>	<p>テキストを置き換えます。次に例を示します。</p> <pre>GGSCI (SysA) 25> fc 9 GGSCI (SysA) 25> info mgr GGSCI (SysA) 25.. reextract extjd GGSCI (SysA) 25> info extract extjd</pre>
d	<p>文字を削除します。複数の文字を削除するには、1 文字につき d を 1 つ入力します。次に例を示します。</p> <pre>GGSCI (SysA) 26> fc 10 GGSCI (SysA) 26> info extract extjd, detail GGSCI (SysA) 26.. dddddddd GGSCI (SysA) 26> info extract extjd</pre>
<replacement text>	<p>表示されているコマンドを、入力するテキストと 1 対 1 で置き換えます。次に例を示します。</p> <pre>GGSCI (SysA) 26> fc 10 GGSCI (SysA) 26> info mgr GGSCI (SysA) 26.. extract extjd GGSCI (SysA) 26> info extract extjd</pre>

このコマンドを実行するには、**[Enter]** を 2 回押します。1 回目でエディタを終了し、2 回目でコマンドを発行します。編集をキャンセルするには、フォワード・スラッシュ (/) を 2 回入力します。

構文 FC [<n> | -<n> | <string>]

引数	説明
<n>	指定した行からコマンドを表示します。各 GGSCI コマンド行には、セッションの最初を 1 として順序番号が付与されています。
-<n>	現在の行から <n> 行前に発行されたコマンドを表示します。
<string>	指定した文字列から始まる最新のコマンドを表示します。

- 例 1** FC 9
例 2 FC -3
例 3 FC sta

HELP

HELP では、Oracle GoldenGate コマンドに関する情報を取得します。基本コマンドは、コマンド・カテゴリおよび関連コマンドのリストを返します。<command> オプションでは、特定のコマンドに出力を制限します。

構文 HELP [<command>]

引数	説明
<command>	ヘルプを表示するコマンド。

例 HELP add replicat

HISTORY

HISTORY では、GGSCI セッションの開始以降で最近発行された GGSCI コマンドのリストを表示します。!コマンド(86 ページ)またはFC コマンド(88 ページ)を使用して、リスト内のコマンドを再実行できます。

構文 HISTORY [<n>]

例 HISTORY 7

引数	説明
<n>	特定の最近のコマンドを返します。<n> には、任意の正の数を指定します。

このコマンドの結果は次のようになります。

```
1: start manager
2: status manager
3: info manager
4: send manager childstatus
5: start extract extjd
6: info extract extjd
7: history
```

INFO ALL

INFO ALL では、システム上のすべての Manager、Extract および Replicat プロセスのステータスおよび（関連する場合は）ラグを表示します。オプションなしの基本コマンドでは、オンライン（継続的な）プロセスのみを表示します。タスクを表示するには、INFO ALL TASKS または INFO ALL ALLPROCESSES を使用します。

図 10 INFO ALL 出力の例

Program	Status	Group	Lag	Time Since Chkpt
MANAGER	RUNNING			
EXTRACT	ABENDED	EXTCUST	00:00:00	96:56:14
EXTRACT	STOPPED	INITDL		
EXTRACT	STOPPED	INITDBL		

構文 INFO ALL [TASKS | ALLPROCESSES]

引数	説明
TASKS	タスクの情報のみ表示します。
ALLPROCESSES	オンライン・プロセスおよびタスクの情報を表示します。

例 1 INFO ALL TASKS

例 2 INFO ALL ALLPROCESSES

INFO MARKER

INFO MARKER では、NonStop システムの最近処理されたマーカを確認します。GGSCI、Logger、Extract または Replicat によるマーカ処理ごとに 1 つレコードが表示されます。

マーカは、HP NonStop ソフトウェア用の Oracle GoldenGate for NonStop を使用してのみ追加できます。

次に出力の例を示します。

Processed	Added	Diff	Prog	Group	Node
2006-02-16:14:41:15	2006-02-16:14:41:08	00:00:07	EXTRACT	PQACMD	\QAMD
	GROUPCMD REPLICAT RQACMD CLOSEFILES				
2006-02-16:14:41:13	2006-02-16:14:41:08	00:00:05	EXTRACT	PQACMD	\QAMD
	TACLMD REPLICAT RQACMD FUP PURGEDATA #QA16.QAETAR				
2006-02-16:14:41:13	2006-02-16:14:41:08	00:00:05	EXTRACT	EQACMD	\QAMD
	GROUPCMD REPLICAT RQACMD CLOSEFILES				
2006-02-16:14:41:12	2006-02-16:14:41:07	00:00:05	EXTRACT	PQACMD	\QAMD
	TACLMD EXTRACT EQACMD FUP PURGEDATA #DATA07.QASRC				
2006-02-16:14:41:11	2006-02-16:14:41:08	00:00:03	EXTRACT	EQACMD	\QAMD
	TACLMD REPLICAT RQACMD FUP PURGEDATA #QA16.QAETAR				
2006-02-16:14:41:10	2006-02-16:14:41:06	00:00:04	EXTRACT	PQACMD	\QAMD
	TESTING FREE-FORM TEXT TO ALL GROUPS				
2006-02-16:14:41:09	2006-02-16:14:41:08	00:00:01	GGSCI	N/A	\QAMD
	GROUPCMD REPLICAT RQACMD CLOSEFILES				
2006-02-16:14:41:08	2006-02-16:14:41:07	00:00:01	EXTRACT	EQACMD	\QAMD
	TACLMD EXTRACT EQACMD FUP PURGEDATA #DATA07.QASRC				
2006-02-16:14:41:08	2006-02-16:14:41:08	00:00:00	GGSCI	N/A	\QAMD

条件:

- **Processed** は、マーカが処理されたローカル時間です。
- **Added** は、マーカが NonStop オーディット・トレイルまたはログ・トレイルに挿入されたローカル時間です。
- **Diff** は、Processed 値と Added 値の時間差です。Diff は、ユーザー・アプリケーションと Extract および Replicat のアクティビティのラグを示すインジケータとして活用できます。

- **Prog** は、GGSCI、Logger、Extract または Replicat など、マーカーを処理したプロセスです。
- **Group** は、マーカーを処理した Extract または Replicat グループ、または Logger プロセスです。GGSCI がマーカーを処理した場合は、N/A が表示されます。
- **Node** は、オーディット・トレイル内のマーカーの挿入先ノードです。
- ADD MARKER 文にユーザー定義テキストが含まれていた場合は、追加の列が表示されることがあります。

構文 INFO MARKER [COUNT <num items>]

引数	説明
COUNT <num items>	リストへの出力を特定の数の最新のマーカーに制限します。

OBEY

OBEY では、Oracle GoldenGate コマンドのリストを含むファイル进行处理します。OBEY は、特定の順序で頻繁に使用されるコマンドを実行するときに便利です。

OBEY ファイルは、他の OBEY ファイルから呼出しできます。こうしたファイルをネストされた OBEY ファイルと呼びます。最大で 16 の OBEY ファイルをネストできます。ネストされた OBEY ファイルを使用するには、まず ALLOWNESTED コマンドを発行してこの機能を有効化する必要があります。88 ページを参照してください。

構文 OBEY <file name>

引数	説明
<file name>	コマンドのリストを含むファイルの完全修飾名。

例 1 obey ./mycommands.txt

上記のコマンドでは、次のようなファイルを実行します。

```
add extract fin, tranlog, begin now
add exttrail ggs/dirdat/aa, extract fin
add extract hr, tranlog, begin now
add exttrail ggs/dirdat/bb, extract hr
start extract *
info extract *, detail
```

例 2 次に、ネストされた OBEY ファイルの例を示します。OBEY ファイル名を addcmds.txt とします。このファイル内には、startcmds.txt という名前の OBEY ファイルを呼び出して別のコマンド・セットを実行する、別の OBEY コマンドも含まれています。

```
obey ./addcmds.txt
```

(このコマンドは次のことを実行します)

```
add extract fin, tranlog, begin now
add exttrail ggs/dirdat/aa, extract fin
add extract hr, tranlog, begin now
add exttrail ggs/dirdat/bb, extract hr
add replicat fin2, exttrail ggs/dirdat/aa, begin now
add replicat hr2, exttrail ggs/dirdat/bb, begin now
obey ./startcmds.txt
```

(ネストされた startcmds.txt ファイルは次のことを実行します)

```
start extract *
info extract *, detail
start replicat *
info replicat *, detail
```

SHELL

SHELL では、GGSCI インタフェース内からシェル・コマンドを実行します。

構文

SHELL <command>

引数	説明
<command>	実行するシステム・コマンド。

例 1 SHELL dir dirprm*

例 2 SHELL rm ./dat*

SHOW

SHOW では、Oracle GoldenGate 環境を表示します。

図 11

SHOW の表示例

```
Parameter settings:
SET SUBDIRS      ON
SET DEBUG        OFF
Current directory: C:\GG_81
Using subdirectories for all process files
Editor:  notepad
Reports (.rpt)           C:\GG_81\dirrpt
Parameters (.prm)        C:\GG_81\dirprm
Replicat Checkpoints (.cpr) C:\GG_81\dirchk
Extract Checkpoints (.cpe) C:\GG_81\dirchk
Process Status (.pcs)     C:\GG_81\dirpcs
SQL Scripts (.sql)        C:\GG_81\dirsqli
Database Definitions (.def) C:\GG_81\dirdef
```

構文

SHOW

VERSIONS

VERSIONS では、オペレーティング・システムおよびデータベースのバージョン情報を表示します。ODBC 接続の場合は、ドライバ・バージョン情報も表示されます。データベース情報を表示するために、まず DBLOGIN コマンドを発行してデータベース接続を確立してください。

構文

VERSIONS

VIEW GGSEVT

VIEW GGSEVT では、Oracle GoldenGate エラー・ログ (ggserr.log ファイル) を表示します。このファイルには、プロセス起動、シャットダウン、例外条件などの Oracle GoldenGate イベントに関する情報が含まれています。この情報はシステム・エラー・ログにも記録されますが、Oracle GoldenGate エラー・ログのほうが簡単に表示でき、またシステム・エラー・ログよりも過去に遡った情報が保持されていることもあります。

表示される情報は長くなることがあります。最後に到達する前に表示を終了するには、オペレーティング・システムの標準の画面出力終了方法を使用してください。

図 12

VIEW GGSEVT の出力例

```
2006-01-08 11:20:56 GGS INFO      301 GoldenGate Manager for Oracle, mgr.prm:
Command received from GUI (START GGSCI ).
2006-01-08 11:20:56 GGS INFO      302 GoldenGate Manager for Oracle, mgr.prm:
Manager started GGSCI process on port 7840.

2006-01-08 11:21:31 GGS INFO      301 GoldenGate Manager for Oracle, mgr.prm:
Command received from GUI (START GGSCI ).
2006-01-08 11:21:31 GGS INFO      302 GoldenGate Manager for Oracle, mgr.prm:
Manager started GGSCI process on port 7841.
2006-01-08 11:24:15 GGS INFO      301 GoldenGate Manager for Oracle, mgr.prm:
Command received from GUI (START GGSCI ).
2006-01-08 11:24:15 GGS INFO      302 GoldenGate Manager for Oracle, mgr.prm:
Manager started GGSCI process on port 7842.
```

構文

VIEW GGSEVT

VIEW REPORT

VIEW REPORT では、Extract または Replicat が生成したプロセス・レポートを表示します。このレポートには、プロセス・パラメータ、実行統計、エラー・メッセージ、その他の診断情報がリストされます。

このコマンドでは、現在のレポートのみ表示します。プロセスが起動されるたびにレポートはエージングされます。過去のレポートには、順序番号が付けられます (例: finance0.rpt、finance1.rpt)。過去のレポートを表示するには、[<n>] オプションを使用します。

図 13

出力例

```
*****
** Running with the following parameters **
*****
```



```

sourceisfile
userid ggs, password *****
rmthost sys1, mgrport 8040
rmtfile /home/jdad/ggsora/dirdat/tcustord.dat, purge
table tcustord;
rmtfile /home/jdad/ggsora/dirdat/tcustmer.dat, purge
table tcustmer;

Processing table TCUSTORD
Processing table TCUSTMER

*****
* ** Run Time Statistics ** *
*****

Report at 2006-01-13 11:07:36 (activity since 2006-01-13 11:07:31)

Output to /home/jdad/ggsora/dirdat/tcustord.dat:

From Table TCUSTORD:
#          inserts:          2
#          updates:          0
#          deletes:          0
#          discards:         0

Output to /home/jdad/ggsora/dirdat/tcustmer.dat:

From Table TCUSTMER:
#          inserts:          2
#          updates:          0
#          deletes:          0
#          discards:         0

```

構文 VIEW REPORT {<group name>[<n>] | <file name>}

引数	説明
<group name>	グループ名。このコマンドは、Oracle GoldenGate dirrpt サブディレクトリの <group>.rpt という名前のファイルをレポート・ファイルとみなします。
<n>	過去のレポートの番号。レポート・ファイルには、0（最新）～9（最も古い）までの番号が付けられています。
<file name>	完全修飾ファイル名（例：c:\ggs\dirrpt\orders.rpt）。

例 1 次の例では、orders グループの過去のレポート・ファイル（番号 3）を表示します。

```
VIEW REPORT orders3
```

例 2 次の例では、ファイル名で指定した特定のレポートを表示します。

```
VIEW REPORT c:\ggs\dirrpt\orders.rpt
```

第 2 章

Oracle GoldenGate パラメータ

.....

この章では、Oracle GoldenGate プロセスの構成、実行および管理にユーザーが使用可能なすべての Oracle GoldenGate パラメータについて説明します。この章の前半では、各プロセス用のパラメータの概要を説明します。概要に続いて、すべてのパラメータのリファレンスをアルファベット順に記載しています。

Oracle GoldenGate パラメータの使用

ここでは、Oracle GoldenGate パラメータ・ファイルの使用に関する基本的なガイドラインを説明します。詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

パラメータ・ファイルの作成

パラメータ・ファイルを作成するには、GGSCI ユーザー・インタフェース内で EDIT PARAMS コマンドを使用するか（推奨）またはテキスト・エディタを直接使用します。GGSCI を使用する場合、標準のテキスト・エディタを使用しますが、パラメータ・ファイルは自動的に正しいファイル名で正しいディレクトリ内に保存されます。

EDIT PARAMS コマンドにより、GGSCI インタフェース内に次のテキスト・エディタが起動されます。

- Microsoft Windows システムの場合はメモ帳
- UNIX および Linux システムの場合は vi エディタ

注意 GGSCI インタフェースから SET EDITOR コマンドを使用して、デフォルトのエディタを変更できます。

GGSCI でのパラメータ・ファイルの作成手順

1. Oracle GoldenGate がインストールされているディレクトリで、GGSCI を実行します。
2. GGSCI で、次のコマンドを実行して、デフォルトのテキスト・エディタを開きます。

```
EDIT PARAMS <group name>
```

条件: <group name> は、mgr (Manager プロセスの場合) か、作成される Extract または Replicat グループの名前のいずれかになります。Extract または Replicat パラメータ・ファイルの名前は、プロセス・グループの名前と一致する必要があります。

extora という名前の Extract グループのパラメータ・ファイルを作成または編集する場合、次のようにします。

```
EDIT PARAMS extora
```

Manager プロセスのパラメータ・ファイルを作成または編集する場合、次のようにします。

EDIT PARAMS MGR

注意 Linux 上では、グループおよびパラメータ・ファイルの名前はすべて大文字か、小文字である必要があります。ファイル名に大 / 小文字が混ざっている場合、プロセスの開始時にエラーが発生します。

3. エディタの編集関数を使用して、このファイルを記述するのに必要な数のコメント行を入力します。コメント行の冒頭にはそれぞれ 2 つのハイフン (--) があることを確認します。
4. コメント化されていない行で、Oracle GoldenGate パラメータを、それぞれパラメータ文ごとに新しい行に入力します。

Oracle GoldenGate パラメータは、次の構文で構成されます。

<PARAMETER> <argument> [, <option>] [&]

条件:

- <PARAMETER> は、パラメータの名前です。
- <argument> は、パラメータに必要な引数です。パラメータには引数が使用可能なものと、そうでないものがあります。次の例のように、すべての引数をコンマで区切ります。

```
USERID ggs, PASSWORD AACAAAAAAAAAAAAIALCKDZIRHOJBHOJUH, &
ENCRYPTKEY superx128
RMTHOST sysb, MGRPORT 8040
RMTTRAIL /home/ggs/dirdat/cl, PURGE
```
- [, <option>] は、オプションの引数です。
- 前述の例の USERID パラメータ文にあるように、複数行に渡るパラメータ文の場合は各行の末尾に [&] が必要です。

5. ファイルを保存し、閉じます。

テキスト・エディタを使用したパラメータ・ファイルの作成手順

1. 新しいファイルをテキスト・エディタで作成します。
2. エディタの編集関数を使用して、このファイルを記述するのに必要な数のコメント行を入力します。コメント行の冒頭にはそれぞれ 2 つのハイフン (--) があることを確認します。
3. コメント化されていない行で、Oracle GoldenGate パラメータを、それぞれパラメータ文ごとに新しい行に入力します。

Oracle GoldenGate パラメータは、次の構文で構成されます。

<PARAMETER> <argument> [, <option>] [&]

条件:

- <PARAMETER> は、パラメータの名前です。
- <argument> は、パラメータに必要な引数です。パラメータには引数が使用可能なものと、そうでないものがあります。次の例のように、すべての引数をコンマで区切ります。

```
USERID ggs, PASSWORD AACAAAAAAAAAAAAIALCKDZIRHOJBHOJUH, &
ENCRYPTKEY superx128
RMTHOST sysb, MGRPORT 8040
RMTTRAIL /home/ggs/dirdat/cl, PURGE
```

- [`<option>`] は、オプションの引数です。
 - 前述の例の `USERID` パラメータ文にあるように、複数行に渡るパラメータ文の場合は各行の末尾に [`&`] が必要です。
4. ファイルを保存し、閉じます。GGSCI 外でパラメータ・ファイルを作成する場合、次の点を確認してください。
- 所有される `Extract` または `Replicat` グループの名前でパラメータ・ファイルを保存します。`Manager` プロセスによって所有される場合は、"`mgr`" の名前で保存します。
 - Oracle GoldenGate インストール・ディレクトリの `dirprm` ディレクトリ内にパラメータ・ファイルを保存します。

パラメータ・ファイルの変更

Oracle GoldenGate プロセスは、パラメータ・ファイルを編集する前に停止し、パラメータ・ファイルが保存されてから再開される必要があります。プロセスの実行中にパラメータ設定を変更する場合、特に表の追加やマッピングまたはフィルタ・ルールを変更する場合、予期しない、悪い結果となることがあります。

パラメータの変更手順

1. GGSCI で次のコマンドを使用してプロセスを停止します。ただし、Windows で `Manager` を停止する場合は、クラスタ・アドミニストレータを使用して `Manager` を停止する必要があります。

```
STOP {EXTRACT | REPLICAT | MANAGER} <group name>
```

2. テキスト・エディタまたは GGSCI で `EDIT PARAMS` コマンドを使用して、パラメータ・ファイルを開きます。

```
EDIT PARAMS mgr
```

3. 編集を行い、ファイルを保存します。

4. プロセスを開始します (Windows クラスタで `Manager` を起動する場合はクラスタ・アドミニストレータを使用します)。

```
START {EXTRACT | REPLICAT | MANAGER} <group name>
```

パラメータ・カテゴリ

Oracle GoldenGate パラメータは、次のカテゴリに分類されます。

概要	開始ページ
GLOBALS パラメータの概要	99 ページ
Manager パラメータの概要	100 ページ
Extract パラメータの概要	102 ページ
Replicat パラメータの概要	107 ページ
DEFGEN パラメータの概要	114 ページ

GLOBALS パラメータの概要

GLOBALS ファイルは、特定のプロセス実行時パラメータではなく、Oracle GoldenGate インスタンス全体に関連するパラメータを保持します。

表 7 すべての GLOBALS パラメータ

パラメータ	説明
MGRSERVNAME	Windows サービスとしてインストールされている場合に Manager プロセス名を指定します。
CHECKPOINTTABLE	デフォルトのチェックポイント表を指定します。
GGSCHEMA	Oracle の DDL 同期をサポートするデータベース・オブジェクトを含むスキーマ名を指定します。
DDLTABLE	Oracle の DDL 同期をサポートする DDL 履歴表の非デフォルト名を指定します。
MARKERTABLE	Oracle の DDL 同期をサポートする DDL マーカー表の非デフォルト名を指定します。
OUTPUTFILEUMASK	Oracle GoldenGate プロセスがトレイル・ファイルおよび破棄ファイルの作成に使用できる umask を指定します。
SYSLOG	システム・ログに書き込まれる Oracle GoldenGate メッセージのタイプをフィルタします。
UNLOCKEDTRAILFILES	ターゲット・システムに影響を与えるネットワーク障害が発生している場合に、Collector によってロックされているトレイル・ファイルの問題を回避します。

Manager パラメータの概要

Manager は Oracle GoldenGate の親プロセスで、自身のプロセス、リソース、ユーザー・インタフェース、しきい値およびエラーのレポートを管理します。ほとんどの場合、Manager はデフォルト設定で問題なく機能します。

表 8 Manager パラメータ : 一般

パラメータ	説明
COMMENT --	パラメータ・ファイルにコメントを挿入します。
SYSLOG	システム・ログに書き込まれる Oracle GoldenGate メッセージのタイプをフィルタします。

表 9 Manager パラメータ : ポート管理

パラメータ	説明
DYNAMICPORTLIST	Manager が動的に割り当て可能なポートを指定します。
DYNAMICPORTREASSIGNDELAY	以前に割り当て済のポート番号を再割り当てするまでの待機時間を指定します。
PORT	Manager がリクエストをリスニングする TCP/IP ポート番号を指定します。

表 10 Manager パラメータ : プロセス管理

パラメータ	説明
AUTORESTART	障害発生後に Manager によって再起動されるプロセスを指定します。
AUTOSTART	Manager の起動時に起動されるプロセスを指定します。
BOOTDELAYMINUTES	システムのブート後、Manager がメイン処理アクティビティを実行するまでの遅延時間を決定します。このパラメータは、Windows をサポートします。
UPREPORT	プロセス・ハートビート・メッセージをレポートする間隔を決定します。

表 11 Manager パラメータ : イベント管理

パラメータ	説明
DOWNCRITICAL	正常または異常終了したプロセスをレポートします。
DOWNREPORT	停止したプロセスをレポートする間隔を制御します。

表 11 Manager パラメータ : イベント管理 (続き)

パラメータ	説明
LAGCRITICAL	クリティカルとみなすラグしきい値を指定し、このしきい値に到達したときにとエラー・ログに警告を生成します。
LAGINFO	情報メッセージをエラー・ログにレポートするラグしきい値を指定します。
LAGREPORT	ラグ・タイムをエラー・ログにレポートする間隔を設定します。

表 12 Manager パラメータ : データベース・ログイン

パラメータ	説明
SOURCEDB	ログイン情報の一部としてデータ・ソース名を指定します。
USERID	データベースにアクセスするときに必要なログイン情報を Manager に提供します。

表 13 Manager パラメータ : メンテナンス

パラメータ	説明
CHECKMINUTES	Manager がメンテナンス・アクティビティを実行する間隔を決定します。
PURGEDDLHISTORY	Oracle DDL 履歴表から不要になった行をページします。
PURGEMARKERHISTORY	不要になった Oracle マーカー表の行をページします。
PURGEOLDEXTRACTS	不要になったトレイル・データをページします。
PURGEOLDTASKS	指定した時間の経過後に Extract および Replicat タスクをページします。
STARTUPVALIDATIONDELAY[CSECS]	Manager が起動後のプロセスが実行中かどうかを確認するまでの遅延時間を設定します。

Extract パラメータの概要

Extract プロセスは、構成パラメータに応じて完全なデータ・レコードまたはトランザクション・データの変更を取得した後、ターゲット表に適用されるかロード・ユーティリティなどの別のプロセスによって処理されるデータをターゲット・システムに送信します。

表 14 Extract パラメータ : 一般

パラメータ	説明
CHECKPARAMS	パラメータ・ファイルの構文を検証します。
COMMENT --	パラメータ・ファイルのコメントを示します。
ETOLDFORMAT	Oracle GoldenGate リリース 6.0 より前の Replicat バージョンと互換性のあるフォーマットでトレイルを生成します。
GETENV	SETENV パラメータで設定された変数を取得します。
OBEY	異なるパラメータ・ファイルに含まれているパラメータ文を処理します。
RECOVERYOPTIONS	Extract プロセスのリカバリ・モードを制御します。
SETENV	GGSCI インタフェース内から UNIX 環境変数の値を指定します。
TCPSOURCE TIMER NOTCPSOURCE TIMER	送信元および送信先システムの時間が異なるときに、送信されるレコードを送信先システムのタイムスタンプに合わせて調整します。
TRACETABLE NOTRACETABLE	双方向同期中に Replicat によって生成されたデータベース変更を Extract に無視させます。Oracle をサポートします。

表 15 Extract パラメータ : 処理方法

パラメータ	説明
BEGIN	処理実行をいつ開始するかを指定します。
DSOPTIONS	Teradata Access Module (TAM) を使用している場合に Extract 処理オプションを指定します。
END	処理実行をいつ終了するかを指定します。
EXTRACT	Extract グループをオンライン・プロセスとして定義します。
GETAPPLOPS IGNOREAPPLOPS	Replicat 以外のすべてのプロセスの操作をトレイルまたはファイルに書き込むかどうかを制御します。
GETREPLICATES IGNOREREPLICATES	レプリケートされた操作を、同じシステム上で動作している Extract に取得させるかどうかを制御します。

表 15 Extract パラメータ : 処理方法 (続き)

パラメータ	説明
PASSTHRU NOPASSTHRU	パススルー・モードのデータポンプ Extract が表をパススルー・モードで処理するか、またはデータ定義を使用するかを制御します。
RMTTASK	リモート・システムに処理タスクを作成します。
SOURCEISTABLE	ソース表からレコード全体を抽出します。
SPECIALRUN	実行間でチェックポイントを作成しないワнтаイトム処理タスクを指定します。
VAM	Teradata Access Module (TAM) を使用して Extract プロセスにトランザクション・データを提供することを指定します。

表 16 Extract パラメータ : データベース・ログイン

パラメータ	説明
SOURCEDB	ログイン情報の一部としてデータ・ソースを指定します。
USERID	データベース接続情報を指定します。

表 17 Extract パラメータ : データの選択とマッピング

パラメータ	説明
ASCIITOEBCDIC	UNIX System Services を実行する z/OS 上の DB2 用に ASCII テキストを EBCDIC に変換します。
COLMATCH	グローバル列マッピング・ルールを作成します。
COMPRESSDELETES NOCOMPRESSDELETES	Oracle GoldenGate が削除操作のときにトレイルにキーのみを書き込むかすべての列を書き込むかを制御します。
COMPRESSUPDATES NOCOMPRESSUPDATES	更新操作のときに主キー列および変更された列のみを書き込みます。
DDL	DDL 操作の取得を有効化およびフィルタします。
DDLSUBST	DDL 処理で文字列を置き換えます。
FETCHOPTIONS	Oracle GoldenGate のデータ・フェッチ方法を制御します。
GETDELETES IGNOREDELETES	削除操作の抽出を制御します。
GETINSERTS IGNOREINSERTS	挿入操作の抽出を制御します。

表 17 Extract パラメータ : データの選択とマッピング (続き)

パラメータ	説明
GETTRUNCATES IGNORETRUNCATES	切捨て文の抽出を制御します。
GETUPDATEAFTERS IGNOREUPDATEAFTERS	アフター・イメージの抽出を制御します。
GETUPDATEBEFORES IGNOREUPDATEBEFORES	ビフォア・イメージの抽出を制御します。
GETUPDATES IGNOREUPDATES	更新操作の抽出を制御します。
REPLACEBADCHAR	無効な文字値を別の値に置き換えます。
SEQUENCE	同期する順序を指定します。
Extract 用 TABLE	抽出する表を指定し、列マッピングおよび変換を制御します。
TABLEEXCLUDE	抽出プロセスから表を除外します。
TARGETDEFS	NonStop プラットフォームに存在するターゲット・データベースのターゲット表定義を含むファイルを指定します。
TRIMSPACES NOTRIMSPACES	CHAR 列を VARCHAR 列にマッピングする際に、末尾の空白を切り捨てるかどうかを制御します。
WILDCARDRESOLVE	TABLE 文でワイルドカードで指定されている表を処理するルールを定義します。

表 18 Extract パラメータ : データ・ルーティング

パラメータ	説明
EXTFILE	ローカル・システム上に、抽出されたデータを書き込む抽出ファイルを指定します。
EXTTRAIL	ローカル・システム上に、抽出されたデータを書き込むトレイルを指定します。
RMTFILE	リモート・システム上に、抽出されたデータを書き込む抽出ファイルを指定します。
RMTHOST	ターゲット・システムおよび Manager ポート番号を指定します。
RMTTRAIL	リモート・システム上に、抽出されたデータを書き込むトレイルを指定します。

表 19 Extract パラメータ : データ・フォーマット

パラメータ	説明
FORMATASCII	抽出データを外部 ASCII フォーマットでフォーマットします。
FORMATSQL	抽出データを SQL 文にフォーマットします。
FORMATXML	抽出データを XML 構文にフォーマットします。
NOHEADERS	レコード・ヘッダーのトレイルへの書き込みを防ぎます。

表 20 Extract パラメータ : カスタム処理

パラメータ	説明
CUSEREXIT	処理中にユーザー・イグジット・ルーチンを起動します。
INCLUDE	マクロ・ライブラリを起動します。
MACRO	Oracle GoldenGate マクロを定義します。
MACROCHAR	デフォルトの # 以外のマクロ文字を定義します。
SQLEXEC	Extract 処理中にストアド・プロシージャまたは問合せを実行します。
VARWIDTHNCHAR NOVARWIDTHNCHAR	NCHAR 列の長さ情報をトレイルに書き込むかどうかを制御します。

表 21 Extract パラメータ : レポート

パラメータ	説明
CMDTRACE	Extract レポート・ファイルにマクロ展開手順を表示します。
LIST NOLIST	レポート・ファイルにマクロのリストを表示するかどうかを制御します。
REPORT	統計レポートをスケジュールします。
STATOPTIONS	統計表示に含める情報を指定します。
REPORTCOUNT	処理されたレコード数をレポートします。
TRACE/TRACE2	プロセスのボトルネックの把握に役立つ Extract 処理情報を表示します。
WARNLONGTRANS	長時間におよぶトランザクションを定義し、これらのトランザクションを確認およびレポートする間隔を制御します。

表 22 Extract パラメータ : エラー処理

パラメータ	説明
DDLERROR	DDL 抽出のエラー処理を制御します。
DISCARDFILE	処理できなかったレコードを記録します。

表 23 Extract パラメータ : チューニング

パラメータ	説明
ALLOCFILES	NUMFILES の値に到達した後に割り当てる増分メモリ構造数を制御します。
BR	Extract の Bounded Recovery 機能を制御します。
CACHEMGR	仮想メモリ・キャッシュ・マネージャを制御します。
CHECKPOINTSECS	Extract のチェックポイント書き込み頻度を制御します。
DBOPTIONS	データベース・オプションを指定します。
DDLOPTIONS	DDL 処理オプションを指定します。
DYNAMICRESOLUTION NODYNAMICRESOLUTION	Extract が表のトランザクション・データを検出するまで、その表のメタデータ参照を抑止します。多数の表を同期するときに、Extract の処理開始を高速化します。
EOFDELAY EOFDELAYCSECS	Extract が処理する新しいデータを探すまでの待機時間を指定します。
FLUSHSECS FLUSHCSECS	レコード・データがトレイルに書き込まれるまでバッファに保持される時間を指定します。
FUNCTIONSTACKSIZE	Oracle GoldenGate ファンクションの処理に使用されるメモリ・スタックのサイズを制御します。
GROUPTRANSOPS	1 回のバッチでトレイルに送信されるレコード数を制御します。
LOBMEMORY	LOB を含むトランザクションのキャッシングに使用できるメモリおよび一時ディスク領域の量を制御します。
MAXFETCHSTATEMENTS	Extract がデータベースからのデータのフェッチにできる、準備済問合せの最大数を制御します。
NUMFILES	Oracle GoldenGate が処理する表に関する情報を格納する専用メモリの初期割当てを制御します。
RMTHOSTOPTIONS	パッシブ Extract グループが使用する TCP/IP 接続のホスト情報以外の接続属性を指定します。

表 23 Extract パラメータ : チューニング (続き)

パラメータ	説明
THREADOPTIONS	Extract の Oracle Real Application Cluster 環境での動作方法を制御します。Oracle をサポートします。
TRANLOGOPTIONS	ログ処理オプションを指定します。
TRANSMEMORY	コミットされていないトランザクション・データのキャッシングに使用できるメモリーおよび一時ディスク領域の量を制御します。

表 24 Extract パラメータ : メンテナンス

パラメータ	説明
DISCARDROLLOVER	新しい破棄ファイルを作成する頻度を制御します。
PURGEOLDEXTRACTS	古いトレイル・ファイルをパージします。
REPORTROLLOVER	新しいレポート・ファイルをいつ作成するか指定します。
ROLLOVER	トレイル・ファイルをエーijingする方法を指定します。

表 25 Extract パラメータ : セキュリティ

パラメータ	説明
DECRYPTTRAIL	トレイルまたは抽出ファイルのデータを復号化します。
ENCRYPTTRAIL NOENCRYPTTRAIL	トレイルまたは抽出ファイルのデータの暗号化を制御化します。

Replicat パラメータの概要

Replicat プロセスは、Extract プロセスによって抽出されたデータを読み取り、このデータをターゲット表に適用するか、ロード・ユーティリティなどの他のアプリケーションがこのデータを使用できるように準備を整えます。

表 26 Replicat パラメータ : 一般

パラメータ	説明
CHECKPARAMS	パラメータ・ファイルの構文を検証します。
COMMENT --	パラメータ・ファイルのコメントを示します。
GETENV	SETENV パラメータで設定された変数を取得します。

表 26 Replicat パラメータ : 一般 (続き)

パラメータ	説明
OBEY	異なるパラメータ・ファイルに含まれているパラメータ文を処理します。
SETENV	GGSCI インタフェース内から UNIX 環境変数の値を指定します。
TRACETABLE NOTRACETABLE	Replicat がターゲット・データベースを更新するたびにレコードを追加するトレース表を指定します。Oracle をサポートします。

表 27 Replicat パラメータ : 処理方法

パラメータ	説明
BEGIN	Replicat 処理の開始位置を指定します。SPECIALRUN を指定するときに指定する必要があります。
BULKLOAD	Oracle SQL*Loader ユーティリティのインタフェースに直接データをロードします。
END	Replicat 処理の停止位置を指定します。SPECIALRUN を使用するとき指定する必要があります。
GENLOADFILES	データベース・ロード・ユーティリティと互換性のある実行ファイルと制御ファイルを生成します。
REPLICAT	オンライン変更同期を行う Replicat グループを指定します。
SPECIALRUN	実行間でチェックポイントを必要としないワнтаイム処理タスクで使用します。

表 28 Replicat パラメータ : データベース・ログイン

パラメータ	説明
TARGETDB	ログイン情報の一部としてデータ・ソースを指定します。
USERID	データベース接続情報を指定します。

表 29 Replicat パラメータ : データの選択、変換およびマッピング

パラメータ	説明
ALLOWDUPTARGETMAP NOALLOWDUPTARGETMAP	パラメータ・ファイル内でソースとターゲットが同一の MAP 文を複数回使用することを許可します。
ASCHTOEBCDIC	UNIX System Services を実行する z/OS 上の DB2 用に、受信 ASCII テキストを EBCDIC に変換します。
ASSUMETARGETDEFS	ソース表とターゲット表が同一の列構造を持つとみなします。
COLMATCH	グローバル列マッピングルールを作成します。
DDL	DDL 操作の取得を有効化およびフィルタします。
DDLSUBST	DDL 処理で文字列を置き換えます。
GETDELETES IGNOREDELETES	削除操作のレプリケーションを制御します。
GETINSERTS IGNOREINSERTS	挿入操作のレプリケーションを制御します。
GETUPDATEAFTERS IGNOREUPDATEAFTERS	アフター・イメージのレプリケーションを制御します。
GETUPDATEBEFORES IGNOREUPDATEBEFORES	ビフォア・イメージのレプリケーションを制御します。
GETUPDATES IGNOREUPDATES	更新操作のレプリケーションを制御します。
GETTRUNCATES IGNORETRUNCATES	TRUNCATE 文のレプリケーションを包含または除外します。
INSERTALLRECORDS	レコードに対する各変更操作を、新しいレコードとしてターゲット表に挿入します。
INSERTDELETES NOINSERTDELETES	削除を挿入に変換します。
INSERTMISSINGUPDATES NOINSERTMISSINGUPDATES	ターゲット行が存在しない場合に、更新を挿入に変換します。
INSERTUPDATES NOINSERTUPDATES	更新を挿入に変換します。
Replicat 用 MAP	1 つ以上のソース表およびターゲット表の関係を指定し、列マッピングおよび変換を制御します。
MAPEXCLUDE	MAP 文でワイルドカードで指定されている表を処理から除外します。
REPLACEBADCHAR	無効な文字値を別の値に置き換えます。
REPLACEBADNUM	無効な数値を別の値に置き換えます。

表 29 Replicat パラメータ : データの選択、変換およびマッピング (続き)

パラメータ	説明
SOURCEDEFS	DEFGEN ユーティリティによって作成されたソース・データ定義を含むファイルを指定します。
SPACESTONULL NOSPACESTONULL	空白のみを含むターゲット列を NULL に変換するかどうかを制御します。
Replicat 用 TABLE	行がフィルタ基準を満たしたときにイベント・アクションの対象にする 1 つまたは複数の表を指定します。
TRIMSPACES NOTRIMSPACES	CHAR 列を VARCHAR 列にマッピングする際に、末尾の空白を切り捨てるかどうかを制御します。
UPDATEDELETES NOUPDATEDELETES	削除を更新に変更します。
USEDATEPREFIX	DATE データ型のデータ値に、Teradata データベースに必要な DATE リテラル接頭辞を追加します
USETIMEPREFIX	TIME データ型のデータ値に、Teradata データベースに必要な TIME リテラル接頭辞を追加します。
USETIMESTAMPPREFIX	TIMESTAMP データ型のデータ値に、Teradata データベースに必要な TIMESTAMP リテラル接頭辞を追加します。

表 30 Replicat パラメータ : データ・ルーティング

パラメータ	説明
EXTFILE	レプリケートするデータを含むローカル・システム上の抽出ファイル名を定義します。ワнтаイム処理で使します。
EXTTRAIL	レプリケートするデータを含むトレイルを定義します。ワнтаイム処理で使します。

表 31 Replicat パラメータ : カスタム処理

パラメータ	説明
CUSEREXIT	処理中にユーザー・イグジット・ルーチンを起動します。
DEFERAPPLYINTERVAL	レプリケートされた操作をターゲット・データベースに適用するまで Replicat が待機する時間を指定します。
INCLUDE	パラメータ・ファイルのマクロ・ライブラリを参照します。
MACRO	Oracle GoldenGate マクロを定義します。

表 31 Replicat パラメータ : カスタム処理 (続き)

パラメータ	説明
SQLEXEC	Replicat 処理中にストアド・プロシージャまたは問合せを実行します。

表 32 Replicat パラメータ : レポート

パラメータ	説明
CMDTRACE	レポートにマクロ展開手順を表示します。
LIST NOLIST	レポート・ファイルにマクロのリストを表示するかどうかを制御します。
REPORT	特定の日または時刻に統計レポートをスケジュールします。
REPORTCOUNT	処理されたレコード数をレポートします。
SHOWSYNTAX	Replicat の SQL 文をレポート・ファイルに出力します。
STATOPTIONS	統計表示に含める情報を指定します。
TRACE/TRACE2	ボトルネックの把握に役立つ Replicat 処理情報を表示します。

表 33 Replicat パラメータ : エラー処理

パラメータ	説明
CHECKSEQUENCEVALUE NOCHECKSEQUENCEVALUE	Replicat がターゲットの順序値がソースより高いかを検証し、検出した不整合を修正するかどうかを制御します。
DDLERROR	DDL レプリケーションのエラー処理を制御します。
DISCARDFILE	処理できなかったレコードを記録します。
HANDLECOLLISIONS NOHANDLECOLLISIONS	重複レコードおよび不明レコードのエラーを処理します。
HANDLETPKUPDATE	一時主キー更新のレプリケートに関連する制約エラーを防止します。
OVERRIDEDUPS NOOVERRIDEDUPS	重複レコード・エラーが発生するたびに、既存のターゲット・レコードをレプリケートされた挿入レコードで上書きします。
RESTARTCOLLISIONS NORESTARTCOLLISIONS	競合のために Oracle GoldenGate が異常終了した後に、Replicat が HANDLECOLLISIONS ロジックを適用するかどうかを制御します。
REPERROR	Replicat がデータベース・エラーに対応する方法を決定します。

表 33 Replicat パラメータ : エラー処理 (続き)

パラメータ	説明
REFFETCHEDCOLOPTIONS	データベースからのフェッチを必要とする操作に対する Replicat のレスポンス方法を決定します。
SQLDUPERR	重複レコードを示すデータベース・エラー番号を指定します。OVERRIDEDUPS とともに使用します。
WARNRATE	データベース・エラーをレポートする頻度を指定します。

表 34 Replicat パラメータ : チューニング

パラメータ	説明
ALLOCFILES	NUMFILES で指定されているメモリー量に到達したときに割り当てる増分メモリー量を制御します。
BATCHSQL	同様の SQL 文を配列にまとめ、より高速に適用することによって、Replicat 処理のスループットを向上させます。
CHECKPOINTSECS	トランザクション・コミットの結果チェックポイントが生成されないときに、Replicat がチェックポイントを書き込む頻度を制御します。
DBOPTIONS	データベース・オプションを指定します。
DDLOPTIONS	DDL 処理オプションを指定します。
DYNAMICRESOLUTION NODYNAMICRESOLUTION	多数の表を同期するときに Replicat の処理開始を高速化します。
DYNSQL NODYNSQL	1 回のコンパイルで何回も同じ文を使用する方法のかわりに、Replicat にリテラル SQL 文を使用させます。
EOFDELAY EOFDELAYCSECS	Replicat が処理する次のデータを探すまでの待機時間 (秒) を指定します。
FUNCTIONSTACKSIZE	Oracle GoldenGate ファンクションの処理に使用するメモリー・スタックのサイズを制御します。
GROUPTRANSOPS	1 つの Replicat トランザクションにグループ化されるレコード数を制御します。
INSERTAPPEND NOINSERTAPPEND	INSERT 操作を Oracle ターゲット表に適用するときに、Replicat が APPEND ヒントを使用するかどうかを制御します。
MAXDISCARDRECS	破棄ファイルにレポートする破棄レコード数を制限します。
MAXSQLSTATEMENTS	Replicat が使用できる準備済 SQL 文の数を制御します。

表 34 Replicat パラメータ : チューニング (続き)

パラメータ	説明
MAXTRANSOPS	大きなソース・トランザクションをターゲット・システム上で小さなトランザクションに分割します。
NUMFILES	Oracle GoldenGate が処理する表に関する情報を格納する専用メモリーの初期割当てを制御します。
RETRYDELAY	失敗した SQL 操作を再試行するまでの遅延を指定します。
TRANSACTIONTIMEOUT	Replicat が、オープンしているターゲット・トランザクションのコミット、およびソース・トランザクション全体の適用準備ができるまで保持しているターゲット・トランザクション内の未完了のソース・トランザクションのロールバックを実行するまで待機する時間を指定します。
TRANSMEMORY	コミットされていないトランザクション・データのキャッシングに使用できるメモリーおよび一時ディスク領域の量を制御します。
WILDCARDRESOLVE	MAP 文に指定されたワイルドカードを解決するルールを変更します。

表 35 Replicat パラメータ : メンテナンス

パラメータ	説明
DISCARDROLLOVER	新しい破棄ファイルをいつ作成するか指定します。
PURGEOLDEXTRACTS	古いトレイル・ファイルをページします。
REPORTROLLOVER	新しいレポート・ファイルをいつ作成するか指定します。

表 36 Replicat パラメータ : セキュリティ

パラメータ	説明
DECRYPTTRAIL	トレイルまたは抽出ファイルのデータを復号化します。

DEFGEN パラメータの概要

DEFGEN では、ソース表またはターゲット表のデータ定義を含むファイルを作成します。データ定義は、ソース表とターゲット表が異なる定義を持つ場合、またはデータベースのタイプが異なる場合に必要です。

表 37 すべての DEFGEN パラメータ

パラメータ	説明
DEFSFILE	DEFGEN が定義を書き込むファイルの名前を指定します。
SOURCEDB	ログイン情報の一部としてデータ・ソースを指定します。
DEFGEN 用 TABLE	定義を取得する表を指定します。
USERID	データベース接続情報を指定します。

アルファベット順リファレンス

ここからは、Windows および UNIX プラットフォームで使用する Oracle GoldenGate パラメータをアルファベット順に説明します。

ALLOCFILES

適用対象 Extract および Replicat

ALLOCFILES では、NUMFILES パラメータで指定されている初期メモリ割当てに到達した後に割り当てる追加のメモリ構造数を制御します (263 ページを参照してください)。この 2 つのパラメータによって、処理中のソース表およびターゲット表に関する情報を保持するプロセス・メモリの割当て方法を制御します。

システム・リソースが許可するメモリーが、プロセスによって必要に応じて割り当てられるので、NUMFILES、ALLOCFILES ともデフォルト値で問題なく機能するはずです。

ALLOCFILES は、すべての TABLE または MAP エントリが有効になる前に指定する必要があります。

デフォルト 500

構文 ALLOCFILES <number of structures>

引数	説明
<number of structures>	割り当てる追加のメモリ構造数。メモリーの無駄な消費を防ぐため、ALLOCFILES を不必要に高い数に設定しないでください。Oracle GoldenGate のメモリ構造は、200 万表までサポートします。

例 ALLOCFILES 1000

ALLOWDUPTARGETMAP | NOALLOWDUPTARGETMAP

適用対象 Replicat

ALLOWDUPTARGETMAP および NOALLOWDUPTARGETMAP パラメータでは、1つのパラメータ・ファイル内で、同一のソースおよびターゲット・オブジェクトに対する重複する MAP 文を受け入れるかどうかを制御します。たとえば、ALLOWDUPTARGETMAP が有効な場合、次のパラメータ・ファイルは許可されます。

```
REPLICAT repcust
USERID GoldenUser, PASSWORD ****
SOURCEDEFS /ggs/dirdef/source.def
ALLOWDUPTARGETMAP

GETINSERTS
GETUPDATES
IGNOREDELETES
MAP ggs.tcustmer, TARGET ggs.tcustmer, COLMAP (USEDEFAULTS, deleted_row = "N");

IGNOREINSERTS
IGNOREUPDATES
GETDELETES
UPDATEDELETES
MAP ggs.tcustmer, TARGET ggs.tcustmer, COLMAP (USEDEFAULTS,
deleted_row = "Y");
```

同じ目的でワイルドカードを使用できるため、デフォルトでは ALLOWDUPTARGETMAP は無効化されています。

デフォルト NOALLOWDUPTARGETMAP

構文 ALLOWDUPTARGETMAP | NOALLOWDUPTARGETMAP

ASCIITOEBCDIC

適用対象 Extract および Replicat

ASCIITOEBCDIC パラメータでは、ASCII から EBCDIC フォーマットへのデータの変換を制御します。Unicode データは、いずれのフォーマットにも変換されません。

このパラメータは、Oracle GoldenGate v9.5.0 ビルド 004 より前のリリースの Extract を使用している場合にのみ必要です。

デフォルト なし

構文 ASCIITOEBCDIC

ASSUMETARGETDEFS

適用対象 Replicat

ASSUMETARGETDEFS パラメータは、ホット・サイトを同期するときなど、MAP 文で指定したソースおよびターゲット表が同一の列構造を持つ場合に使用します。Oracle GoldenGate にソース定義ファイルからソース構造を参照しないよう指示します。

ソース列およびターゲット列の構造を同一なものにする場合、次の条件を満たす必要があります。

- 列の名前が同一である（該当する場合は大 / 小文字の区別も含める）。
- データ・タイプが同一である
- 列のサイズが同一である
- 文字列の長セマンティクスが同じである（バイトと文字の違い）
- 各表の順序が同じである

たとえば、ソースの **Oracle** データベースのセマンティクスがバイトとして構成されているのに対し、ターゲット・セマンティクスが文字として構成されている場合（またはその逆）、表の構造が同一であってもソース定義ファイルが必要です。また別の例として次に示す、ソースおよびターゲット表の組合せでは名前列の順序以外は同一ですが、ソース定義ファイルが必要です。

ソース	ターゲット
CREATE TABLE emp (employee_id NUMBER(6) , first_name VARCHAR2(20) , last_name VARCHAR2(25) , phone_number VARCHAR2(20) , hire_date DATE DEFAULT SYSDATE	CREATE TABLE emp (employee_id NUMBER(6) , last_name VARCHAR2(25) , first_name VARCHAR2(20) , phone_number VARCHAR2(20) , hire_date DATE DEFAULT SYSDATE

ソース表およびターゲット表が同一の構造を持たない場合は、**ASSUMETARGETDEFS** のかわりに **SOURCEDEFS** パラメータを使用します。304 ページの「**SOURCEDEFS**」を参照してください。

デフォルト	なし
構文	ASSUMETARGETDEFS

AUTORESTART

適用対象	Manager
-------------	---------

1 つ以上の **Extract** および **Replicat** プロセスが失敗した後自動的に開始されるようにするには、**AUTORESTART** パラメータを使用します。**AUTORESTART** により、トランザクション・ログへのアクセスが中断される断続的なネットワーク障害またはプログラムなど、何か一時的な障害がプロセスで発生した場合のフォルト・トレランスが提供されます。

AUTORESTART 文は、同一のパラメータ・ファイル内で複数回使用できます。

パッシブ・モードで作成された **Extract** グループにこのパラメータを適用するには、関連付けられている別名 **Extract** グループが存在するターゲット・システム上の **Manager** を使用します。**Oracle GoldenGate** は、ソース・システムに起動コマンドを送信します。パッシブ **Extract** グループに対してローカルで使用すると、**AUTORESTART** は無視されます。

Manager は、再起動時に順序が正しくないトランザクションを検出すると、**Extract** を再起動しません。かわりに警告を出し、**SEND EXTRACT** の **ETROLLOVER** オプションを使用してトレイルを次のファイルに切り替え、エラーの原因のトランザクションをスキップするよう通知します。

デフォルト	自動的に再起動しない
--------------	------------

構文

```
AUTORESTART <process type> <group name>
[, RETRIES <max retries>]
[, WAITMINUTES <wait minutes>]
[, RESETMINUTES <reset minutes>]
```

引数	説明
<process type>	次のいずれかを指定します。 EXTRACT REPLICAT ER (Extract および Replicat)
<group name>	1 つのグループ名、または複数のグループを指定するワイルドカード。ワイルドカードを使用する場合、Oracle GoldenGate は、パッシブ・モードのものを除き、ワイルドカードの条件を満たす、指定された <process type> のローカル・システム上のすべてのグループを起動します。
RETRIES <max retries>	再試行を中止するまで Manager がプロセスの再起動を試行する最大回数。デフォルトの試行回数は 2 です。
WAITMINUTES <wait minutes>	異常終了したプロセスの検出からプロセスの再起動までの待機時間 (分)。このオプションは、必要なリソースが利用可能になるまで、または他のイベントが発生するまで再起動を遅延させるために使用します。デフォルトの遅延は 2 分です。
RESETMINUTES <reset minutes>	再試行回数をカウントする時間枠。デフォルトは 120 分 (2 時間) です。この時間が過ぎると、再試行回数は 0 に戻されます。

例

次の例では、Manager は、1 時間の時間枠内に処理に失敗したすべての Extract プロセスの再起動を 3 回試行します。各再試行の前に 5 分待機します。

```
AUTORESTART EXTRACT *, RETRIES 3, WAITMINUTES 5, &
RESETMINUTES 60
```

AUTOSTART

適用対象 Manager

Manager の起動時に自動的に開始されるようにするには、AUTOSTART パラメータを使用します。AUTOSTART により、すべてのプロセス・グループが認識され、すぐに同期化アクティビティが開始されます。

AUTOSTART 文は、同一のパラメータ・ファイル内で複数回使用できます。

パッシブ・モードで作成された Extract グループにこのパラメータを適用するには、関連付けられている別名 Extract グループが存在するターゲット・システム上の Manager に対してこのパラメータを使用します。Oracle GoldenGate は、ソース・システムに起動コマンドを送信します。パッシブ Extract グループに対してローカルで使用すると、AUTOSTART は無視されます。

Manager は、再起動時に順序が正しくないトランザクションを検出すると、Extract を再起動しません。かわりに警告を出し、SEND EXTRACT の ETROLLOVER オプションを使用してトレイルを次のファイルに切り替え、エラーの原因のトランザクションをスキップするよう通知します。

デフォルト 自動的に起動しない

構文 AUTOSTART <process type> <group name>

引数	説明
<process type>	次のいずれかを指定します。 EXTRACT REPLICAT ER (Extract および Replicat)
<group name>	1 つのグループ名、または複数のグループを指定するワイルドカード。ワイルドカードを使用する場合、Oracle GoldenGate は、パッシブ・モードのものを除き、ワイルドカードの条件を満たす、指定された <process type> のローカル・システム上のすべてのグループを起動します。

例 AUTOSTART ER *

BATCHSQL

適用対象 Replicat

Replicat のパフォーマンスを向上するには、BATCHSQL パラメータを使用します。BATCHSQL により、Replicat で、同様の SQL 文が配列に整理され、より速い速度で適用されます。通常モードでは、Replicat で一度に 1 つの SQL 文が適用されます。

BATCHSQL は次のデータベースに有効です。

- Oracle
- DB2 LUW
- z/OS 上の DB2
- Teradata
- SQL Server

BATCHSQL の動作

BATCHSQL モードでは、Replicat は同様の SQL 文をメモリー・キュー内でバッチにまとめ、1 回のデータベース操作で各バッチを適用します。1 つのバッチに含まれる SQL 文は、同一の表、操作タイプ（挿入、更新または削除）および列リストを処理します。たとえば、次の各グループは 1 つのバッチです。

- 表 A への挿入
- 表 B への挿入
- 表 A への更新
- 表 B への更新
- 表 A からの削除
- 表 B からの削除

注意 Oracle GoldenGate は、バッチの実行の前に、バッチの外部キー参照依存関係を分析します。異なるバッチ内の文の間に依存関係がある場合、各バッチの複数の SQL 文で参照整合性の維持が必要になることがあります。

キャッシュされる文の数の制御

MAXSQLSTATEMENTS パラメータは、キャッシュされる文の数を制御します（260 ページを参照してください）。古い SQL 文は、最低使用頻度アルゴリズムを使用して再利用されます。バッチは、指定されているしきい値（「メモリーの管理」を参照してください）に基づいて実行されます。

使用制限

例外として処理される SQL 文には、次のものが含まれます。

- LOB または LONG データを含む文。
- 25K よりも長い行を含む文。
- ターゲット表に、主キー以外に 1 つ以上の一意キーが含まれている文。BATCHSQL では、主キー以外のキーの値が変更されたときにそれらのキーの正しい順序付けが保証されないため、バッチではこのような文を処理できません。
- (SQL Server) ターゲット表にトリガーが含まれている文。
- エラーを発生させる文。

Replicat は、バッチ・モードで例外が発生すると、バッチ操作をロールバックし、次の方法で例外の処理を試行し、常にトランザクションの整合性を維持します。

- まず、Replicat は標準モードの使用を試み、GROUPTRANSOPS パラメータ（203 ページを参照してください）で設定されているトランザクション境界内で 1 回に 1 つの SQL 文を適用します。
- 標準モードで失敗すると、Replicat はソース・モードの使用を試み、ソースと同じトランザクション境界内で SQL を適用します。

例外処理が終了すると、Replicat は BATCHSQL モードを再開します。

図 14 Replicat のモード

Source transactions*	Replicat transaction (normal mode)	Replicat transaction (BATCHSQL mode)	Replicat transactions (source mode)
Transaction 1 INSERT DELETE	INSERT DELETE	INSERT (x3) DELETE (x3)	INSERT DELETE
Transaction 2 INSERT DELETE	INSERT DELETE INSERT DELETE		INSERT DELETE
Transaction 3 INSERT DELETE	INSERT DELETE		INSERT DELETE

*assumes same table and column list

BATCHSQL を使用するケースの決定

Replicat を BATCHSQL モードで使用する場合、行の変更が小さく、行が大きく変更される場合に比べパフォーマンスが向上します。行変更ごとのデータが 100 バイトの場合、BATCHSQL は、Replicat のパフォーマンスを 300% 向上するという結果が報告されていますが、実際のパフォーマンスの利点は操作の組合せにより異なります。行変更ごとのデータが約 5,000 バイトの場合、BATCHSQL を使用する利点は減少します。

メモリーの管理

SQL 文をバッチにまとめると、効率性は向上しますが、メモリーも消費します。最適なパフォーマンスを維持するために、次の BATCHSQL オプションを使用してください。

BATCHESPERQUEUE

BYTESPERQUEUE

OPSPERBATCH

OPSPERQUEUE

値を設定するベンチマークとして、それぞれ 500 バイトの 1,000 の SQL 文をまとめた 1 つのバッチが必要とするメモリーは 10MB 未満であると想定してください。

デフォルト 無効（標準 Replicat モードで処理）

構文

```
BATCHSQL
[BATCHERRORMODE | NOBATCHERRORMODE]
[BATCHESPERQUEUE <n>]
[BATCHTRANSOPS <n>]
[BYTESPERQUEUE <n>]
[OPSPERBATCH <n>]
[OPSPERQUEUE <n>]
[TRACE]
```

オプション	説明
BATCHERRORMODE NOBATCHERRORMODE	<p>BATCHSQL 処理モード中に発生するエラーに対する Replicat のレスポンスを設定します。</p> <ul style="list-style-type: none"> BATCHERRORMODE では、BATCHSQL モードのまま Replicat にエラーの解決を試行させます。重複レコード・エラーが原因で失敗した挿入を更新に変換し、削除に対する行方不明レコード・エラーを無視します。BATCHERRORMODE を使用する場合は、HANDLECOLLISIONS パラメータを使用して Replicat の異常終了を防止します。 NOBATCHERRORMODE（デフォルト）では、エラーが発生すると、Replicat は一時的に BATCHSQL 処理を無効にし、まず標準モードでトランザクションを再試行し、標準モードで失敗すると、ソース・モード（ソースと同じトランザクション境界）で実行します。

オプション	説明
BATCHESPERQUEUE <n>	<p>1 つのメモリー・キューに維持できる最大バッチ数を制御します。 BATCHESPERQUEUE の値に到達すると、ターゲット・トランザクションが実行されます。</p> <ul style="list-style-type: none"> ◆ 最小値は 1 です。 ◆ 最大値は 1000 です。 ◆ デフォルトは 50 です。
BATCHTRANSOPS <n>	<p>コミットが必要になる前に 1 つのトランザクションにグループ化可能な最大バッチ操作数を制御します。BATCHTRANSOPS の値に到達すると、操作がターゲットに適用されます。BATCHTRANSOPS は、デフォルトの 1000 以上に設定します。</p> <ul style="list-style-type: none"> ◆ 最小値は 1 です。 ◆ 最大値は 100000 です。 ◆ デフォルトは 1000 です。
BYTESPERQUEUE <n>	<p>1 つのキューに維持できる最大バイト数を設定します。BYTESPERQUEUE の値に到達すると、ターゲット・トランザクションが実行されます。</p> <ul style="list-style-type: none"> ◆ 最小値は 1000000 バイト (1MB) です。 ◆ 最大値は 1000000000 バイト (1GB) です。 ◆ デフォルトは 2000000 バイト (20MB) です。
OPSPERBATCH <n>	<p>1 つのバッチに含められる最大行操作数を設定します。OPSPERBATCH の値に到達すると、ターゲット・トランザクションが実行されます。</p> <ul style="list-style-type: none"> ◆ 最小値は 1 です。 ◆ 最大値は 100000 です。 ◆ デフォルトは 1200 です。
OPSPERQUEUE <n>	<p>1 つのキューに維持できるすべてのバッチの最大行操作数を設定します。 OPSPERQUEUE の値に到達すると、ターゲット・トランザクションが実行されます。</p> <ul style="list-style-type: none"> ◆ 最小値は 1 です。 ◆ 最大値は 100000 です。 ◆ デフォルトは 1200 です。
TRACE	<p>コンソールおよびレポート・ファイルへの BATCHSQL アクティビティの詳細なトレースを有効化します。トレースを設定するときは、必ず Oracle サポート・アナリストの指示を受けるようにしてください。</p>

例 BATCHSQL BATCHESPERQUEUE 100, OPSPERBATCH 2000

BEGIN

適用対象 Extract および Replicat

BEGIN パラメータでは、BEGIN で指定した時刻と同じまたはそれより大きいタイムスタンプを持つ、データベース・トランザクション・ログ内または Oracle GoldenGate トレイル内の最初のレコードから処理を開始します。それ以降のレコードは、タイムスタンプが指定した時刻よりも小さいものを含めすべて処理されます。BEGIN は、SPECIALRUN を指定してワンタイム・バッチ処理実行を作成するときに使用します。

バッチ実行の構成方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

デフォルト なし

構文 BEGIN <date> [<time>]

引数	説明
<date> [<time>]	いつ処理を開始するかを指定します。有効な値： <ul style="list-style-type: none"> ◆ yyyy-mm-dd フォーマットの日付。 ◆ hh:mi[:ss[.cccccc]] (24 時間表記) フォーマットの時刻。秒およびセンチ秒はオプションです。

例 BEGIN 1995-08-12 09:22:55

BLOBBMEMORY

このパラメータは、LOBMEMORY の別名です (211 ページを参照してください)。

BOOTDELAYMINUTES

適用対象 Manager

パラメータの実行など、Manager の起動時に Manager により実行されるアクティビティを遅延するには、Windows システムで BOOTDELAYMINUTES を使用します。たとえば、BOOTDELAYMINUTES は、データベース・サービスが開始されるまで AUTOSTART パラメータを遅延するのに使用できます。

BOOTDELAYMINUTES は、他のパラメータ・エントリより先に指定します。このパラメータは、Windows のみサポートしています。

デフォルト なし (遅延なし)

構文 BOOTDELAYMINUTES <minutes>

引数	説明
<minutes>	システムの起動後、Oracle GoldenGate の処理開始までの遅延時間 (分)。

例 BOOTDELAYMINUTES 5

BR

適用対象 Extract (Oracle のみ)

BR パラメータでは、Bounded Recovery (BR) 機能を制御します。この機能は、現在 Oracle データベースをサポートしています。

Bounded Recovery は、一般的な Extract チェックポイント機能のコンポーネントです。計画的または計画外の何らかの理由で Extract が停止した場合、その時点でオープンしている（コミットされていない）トランザクションの数、およびトランザクションの古さにかかわらず、Extract の停止後の効率的なリカバリを保証します。Bounded Recovery により、Extract が停止した位置までリカバリし、正常な処理を再開するまでにかかる最大時間の上限を設定できます。

警告 このパラメータをデフォルト設定から変更する前に、Oracle サポートに連絡してください。ほとんどの本番環境では、このパラメータの変更は必要ありません。ただし、Bounded Recovery チェックポイント・ファイル用のディレクトリは、Oracle サポートの支援なしで指定できます。

Extract によるオープンしていたトランザクションのリカバリ方法

Extract は、REDO ログ内でトランザクションの開始位置（Oracle の場合は最初の実行可能 SQL 文）を検出すると、このトランザクション内の取得が指定されているすべてのデータをメモリーにキャッシングし始めます。取得が必要なデータを含まないトランザクションにも、今後の操作によって取得が必要なデータが含まれる可能性があるため、Extract はこのようなトランザクションもキャッシュする必要があります。

Extract は、トランザクションのコミット・データを検出すると、キャッシュしているこのトランザクション全体をトレイルに書込み、メモリーからクリアします。Extract は、トランザクションのロールバック・レコードを検出すると、このトランザクション全体をメモリーから破棄します。Extract がコミットまたはロールバックを処理するまで、トランザクションはオープンしているとみなされ、その情報収集が継続されます。

トランザクションのコミットまたはロールバック・レコードを検出する前に Extract が停止した場合には、Extract の再起動の際に、キャッシュされていた情報をすべてリカバリする必要があります。Extract が停止したときにオープンしていたすべてのトランザクションがリカバリの対象になります。

Extract は、次のようにこのリカバリを実行します。

- Extract が停止したときにオープンしているトランザクションがなかった場合は、現在の Extract 読み取りチェックポイントからリカバリを開始します。これは標準リカバリです。
- ログ内での開始位置の時刻が Extract の停止時刻と非常に近いオープンしていたトランザクションがある場合、Extract はその中で最も古いトランザクションの開始位置からログを再度読み取ってリカバリを開始します。この場合 Extract は、停止前にトレイルに書き込み済または破棄したトランザクションについては重複作業を行う必要がありますが、この作業は処理するデータ量が比較的少ないため許容できます。これも標準リカバリとみなされます。
- Extract が長時間オープンしているトランザクションとみなすトランザクションが 1 つ以上あった場合は、Extract は *Bounded Recovery* を利用してリカバリを開始します。

Bounded Recovery の動作

トランザクションは、BR パラメータの BRINTERVAL オプションで指定されている 1 つの *Bounded Recovery* 間隔より長くオープンしている場合に長時間におよぶトランザクションとみなされます。たとえば、Bounded Recovery 間隔が 4 時間の場合は、4 時間よりも前に開始されたすべてのトランザクションが長時間オープンしているトランザクションになります。

Extract は、Bounded Recovery 間隔ごとに *Bounded Recovery* チェックポイントを作成し、(存在する場合) 長時間におよぶトランザクションの状態とデータを含め、Extract の現在の状態とデータをディスクに永続化します。Extract は、Bounded Recovery チェックポイント作成後に停止した場合、オープンしている長時間におよぶ最も古いトランザクションが最初に登場するログ内の位置からではなく、直前の Bounded Recovery 間隔内の位置、または最新の Bounded Recovery チェックポイントからリカバリを開始します。

最大 *Bounded Recovery* 時間(Extract が停止位置までのリカバリにかかる最大時間)は、現在の Bounded Recovery チェックポイント間隔の 2 倍を上回ることはありません。実際のリカバリ時間は、次の要因によって決まります。

- 最新の有効な Bounded Recovery 間隔から Extract 停止までの時間。
- その期間の Extract 使用状況。

以前にトレイルに書き込まれたトランザクションの使用率。Bounded Recovery は、これらのトランザクションをディスクに書き込まなければならなかった最初の Extract 処理よりも、(破棄することによって) はるかに早くこれらのトランザクションを処理します。トランザクション・データに対して発生する再処理のほとんどは破棄です。

Extract は、リカバリの際、最新の Bounded Recovery チェックポイントで保存された永続データおよび状態 (長時間におよぶトランザクションのものを含む) をリストアします。

たとえば、あるトランザクションが 24 時間オープンしていて、Bounded Recovery 間隔が 4 時間であるとしします。このときの最大リカバリ時間は、Extract 処理時間の 8 時間を超えず、たいていこの時間を下回ります。最新の有効な Bounded Recovery チェックポイントに対する停止位置、およびその時点における Extract のアクティビティによって、実際の時間は異なります。

Bounded Recovery によって解決される問題

ディスクの永続性を使用して長時間におよぶトランザクションを保管し、リカバリすることによって、Extract は発生頻度は少ないながら、発生した場合にパフォーマンスに重大な (悪い) 影響を及ぼす状況に対応できます。多くの場合、長時間におよぶトランザクションの開始点は、Extract が停止したときのログ内の処理位置よりもはるか前に遡ります。長時間におよぶトランザクションは、多数の古いログにまたがっている可能性があり、古いログの一部はアクセス可能な記憶域に存在しないか、削除されていることもあります。長時間におよぶトランザクションの開始からもう一度ログを読み取るためには、許容できないほど長時間要するだけではなく、長時間におよぶトランザクションはまれにしか発生しないため、大半はすでにトレイルに書き込み済み、または破棄された他のトランザクションをもう一度取得する作業が必要になります。永続化された長時間におよぶトランザクションの状態とデータをリストアできれば、この作業は不要になります。Bounded Recovery は、通常の状況ではシステム・パフォーマンスに影響を与えません。

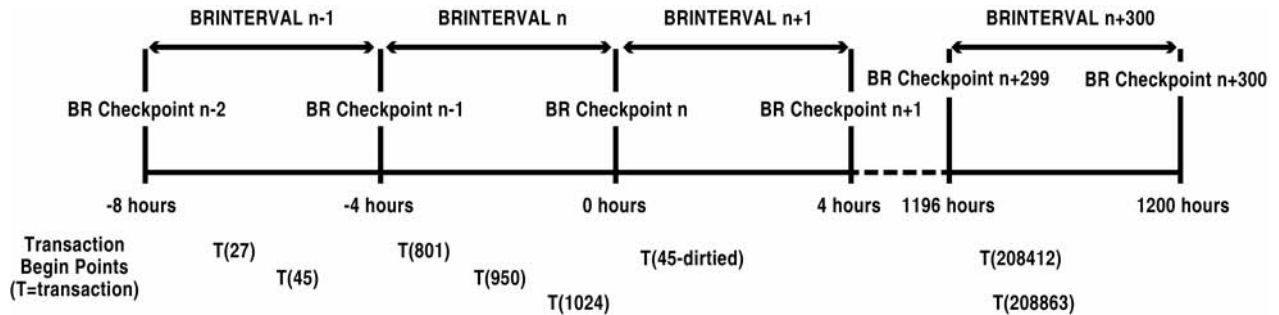
注意 永続化された長時間におよぶトランザクションの数とサイズは、ログから再処理が必要な非永続化データのリカバリには影響しません。

Bounded Recovery の例

次の図は、一連のトランザクションが開始されたタイムラインを表しています。これは、長時間オープンしているトランザクションが特定の間隔の後にディスクにどのように永続化され、処理失敗の後にリカバリされるかを示しています。これは、この例で使用されている用語の理解に役立ちます。

- 永続オブジェクトとは、Bounded Recovery チェックポイントで永続化されたキャッシュ内の任意のオブジェクトです。通常はトランザクションの状態またはデータですが、キャッシュは Extract の内部のオブジェクトにも使用されます。これらすべてを集合的にオブジェクトと呼びます。

- 最も古い非永続オブジェクトとは、現在の Bounded Recovery チェックポイント直前の間隔内のキャッシュにある、オープンしている最も古いオブジェクトです。通常は、この間隔内にあるオープンしている最も古いトランザクションです。Bounded Recovery を再起動すると、実行時プロセスは最も古い非永続オブジェクトの位置から再開し、通常のトランザクションの場合は、これが REDO ログ内のこのトランザクションの位置になります。



この例では、Bounded Recovery 間隔は 4 時間です。オープンしているトランザクションは、現在の Bounded Recovery チェックポイントから 1 つの Bounded Recovery 間隔より長い間オープンしている場合、現在のチェックポイントで永続化されます。

BR チェックポイント n:

- オープンしているトランザクションは、T (27)、T (45)、T (801)、T (950) および T (1024) の 5 つです。これ以外のトランザクションはすべて、コミットされてトレイルに送信されたか、ロールバックされました。トランザクションは、タイムラインに沿って開始位置が示されています。
- 1 つの Bounded Recovery 間隔より長くオープンしているトランザクションは、T(27) および T(45) です。BR チェックポイント n でこれらのトランザクションはディスクに永続化されます。
- 最も古い非永続オブジェクトは T (801) です。これは、オープンしている時間が 1 つの Bounded Recovery 間隔に満たないので、ディスクに永続化される資格がありません。このトランザクションは最も古い非永続化オブジェクトなので、ログ内のこのトランザクションの開始位置が BR チェックポイント n チェックポイント・ファイルに保管されます。Extract は BR チェックポイント n 以降に予期せず停止すると、このログ位置までリカバリし、そこからログの再読取りを開始します。直前の Bounded Recovery 間隔に最も古い非永続オブジェクトが存在しない場合、Extract は現在の Bounded Recovery チェックポイントのログ位置からログの再読取りを開始します。

BR チェックポイント n+1:

- T (45) は直前の Bounded Recovery 間隔で使用済（更新済）なので、新しい永続オブジェクト・ファイルに書き込まれます。古いファイルは、BR チェックポイント n+1 完了後に削除されます。
- Extract は、BR チェックポイント n+1 の書き込み中、または BR チェックポイント n と BR チェックポイント n+1 間の Bounded Recovery チェックポイント間隔内のある時点で処理に失敗すると、最後の有効なチェックポイントである BR チェックポイント n からリカバリを開始します。BR チェックポイント n の再開位置は、最も古い非永続トランザクションである T (801) の開始点です。したがって、最も長いリカバリ時間は常に、Extract が停止したポイントから 2 つの Bounded Recovery 間隔未満、このケースでは 8 時間未満になります。

BR チェックポイント n+3000

- このシステムは長い時間実行されています。T (27) および T (45) のみが永続トランザクションです。T (801) および T (950) は、BR チェックポイント n+2999 より前のある時点でコミットされ、トレイルに書き込まれました。現在オープンしているトランザクションは、T (208412) および T (208863) のみです。

- **BR チェックポイント n+3000** が書き込まれます。
- **BR チェックポイント n+3000** の後の間隔で電源障害が発生します。
- 新しい Extract が **BR チェックポイント n+3000** までリカバリします。T(27)およびT(45)は、**BR チェックポイント n** 時点の状態を含む、それぞれの永続ファイルからリストアされます。ログ読取りは、T(208412) の開始から再開します。

長時間におよぶトランザクションの管理

Oracle GoldenGate は、長時間におよぶトランザクションを管理するために、次のパラメータとコマンドを提供します。

- **WARNLONGTRANS** パラメータでは、トランザクションのオープン後、**Extract** によってこのトランザクションが長時間実行されていることの警告メッセージが生成されるまでの時間を指定します。また **WARNLONGTRANS** も使用して、**Oracle GoldenGate** が長時間におよぶトランザクションをチェックする頻度を制御します。この設定は **Bounded Recovery** 間隔とは無関係で、**Bounded Recovery** 間隔に影響しません。
- **SEND EXTRACT** コマンドと **SKIPTRANS** オプションでは、指定したトランザクションを **Extract** にスキップさせます。
- **SEND EXTRACT** コマンドと **FORCETRANS** オプションでは、指定したトランザクションを **Extract** にコミット済トランザクションとしてトレイルに書き込ませます。
- **TRANLOGOPTIONS** パラメータと **PURGEORPHANEDTRANSACTIONS** オプションでは、ノートに障害が発生し、**Extract** がロールバックを取得できないときに発生する孤立トランザクションのページを有効化します。

ディスクに書き込まれるファイルについて

Bounded Recovery 間隔が終了すると、**Extract** は常に **Bounded Recovery** チェックポイント・ファイルを作成します。永続化が必要な長時間におよぶトランザクションが存在する場合、これらのトランザクションは各自の永続オブジェクト・ファイルに書き込まれます。永続オブジェクト・ファイルには、ディスクに永続化される単一のトランザクションの状態およびデータが格納されます。

これまでの経験では、永続化が必要な長時間におよぶトランザクションが発生することはほとんどなく、このようなケースのほとんどでトランザクションは空です。少量のキャッシュされたデータが含まれていることもあります。

以前に永続化されたオブジェクトがまだオープンしていて、完了したばかりの **Bounded Recovery** 間隔中にその状態やデータが変更された場合、このオブジェクトは新しい永続オブジェクト・ファイルに再度永続化されます。このような変更が発生しない場合、オープンしているトランザクションの以前に作成された永続化オブジェクト・ファイルは変更されません。

理論的には、1 つの長時間におよぶトランザクションを永続化するために、複数の永続ファイルが必要になることも可能性があります。

注意 **Bounded Recovery** ファイルは、**Extract** が別のシステムに移動された場合、同一のデータベースが使用されているとしても、新しいシステムのすべての関連要素が元のシステムと同一でなければ、**Extract** の状態のリカバリに使用できません。たとえば、Oracle 10g Solaris プラットフォームに書き込まれたチェックポイント・ファイルは、Oracle 10g Linux プラットフォーム上の **Extract** のリカバリに使用できません。

Bounded Recovery を標準リカバリに変更する状況

Extract は、永続オブジェクトが存在するまれな状況を除き、ほとんどのケースで Bounded Recovery ではなく標準リカバリを使用します。特定の異常な状況では、Extract は Bounded Recovery から標準リカバリ・モードに切り替えられない場合があります。たとえば、(長時間におよぶトランザクションの永続データが保管されている) ディスクの物理的な破損、Bounded Recovery チェックポイント・ファイルの不注意な削除、環境の継続性に影響を及ぼすその他のアクションやイベントなどが発生する状況があげられます。もっと容易に修正可能な障害が原因の場合もあります。

一部のケースを除き、Extract は、リカバリ中に Bounded Recovery 処理に失敗すると、標準モードに切り替えます。標準リカバリの完了後、Bounded Recovery はもう一度有効化されます。

Bounded Recovery は、次の状況では起動されません。

- Extract の開始位置が CSN または時間によって変更された。
- Extract I/O チェックポイントが変更された。
- Extract パラメータ・ファイルがリカバリ中に変更された (TABLE 指定の変更など)。

リカバリの完了後、Bounded Recovery は次の実行に向けてもう一度有効化されます。

Bounded Recovery 中に Extract が異常終了した場合の措置

Extract が Bounded Recover 中に異常終了した場合は、エラー・ログを確認して理由を特定してください。無効なパラメータ・ファイル、Bounded Recovery ファイルが含まれるディレクトリの不適切な権限など、迅速に解決できる問題が原因の場合もあります。こうしたケースでは、問題の修正後、Extract を再起動して Bounded Recovery を有効化できます。

問題が修正できない場合は、Extract を GGSCI の BRRESET コマンドで再起動してください。Extract は標準モードでリカバリを実行し、その後 Bounded Recovery を再び有効化します。

BR パラメータの変更

Bounded Recover は、デフォルトで有効化されており、デフォルトの Bounded Recovery 間隔は 4 時間です (BRINTERVAL オプションで制御します)。この間隔は、ほとんどの環境に適切なはずですが、BR パラメータを変更するときは、先に必ず Oracle サポート・アナリストの指示を受けるようにしてください。Oracle GoldenGate アナリストは、使用可能な Bounded Recovery 実行時統計に基づいて Bounded Recovery 使用状況を分析し、デフォルトでは不十分な不測のイベント発生時に適切な BRINTERVAL の設定を特定できます。

BR の変更を要求された場合は、Bounded Recovery が標準 Extract チェックポイント間隔の倍数であることに注意してください。Extract チェックポイントは、CHECKPOINTSECS パラメータで制御されます。したがって、BR パラメータは、標準 Extract チェックポイントに対する Bounded Recovery チェックポイントの割合を制御します。Oracle 担当者から指示された場合は、両方のパラメータを変更する必要があります。

サポートされているデータベース

このパラメータは Oracle データベースに適用されます。他のデータベースでは、Extract は障害時の最も古いトランザクションの開始位置から古いログを読取ってリカバリを行い、長時間におよぶトランザクションは永続化されません。

デフォルト BR BRINTERVAL 4, BRDIR BR

構文

```
BR
[, BRDIR <directory>]
[, BRINTERVAL <interval><unit>]
[, BRKEEPSTALEFILES]
[, BROFF]
[, BROFFONFAILURE]
[, BRRESET]
```

引数	説明
BRDIR <directory>	<p>BR ディレクトリを含める親ディレクトリの完全パス名を指定します。BR ディレクトリには、Bounded Recovery チェックポイント・ファイルが保管されます。このディレクトリ名は変更できません。BR ディレクトリのデフォルトの親ディレクトリは、Oracle GoldenGate インストール・ファイルを含むルート・ディレクトリにある BR という名前のディレクトリです。</p> <p>特定の Oracle GoldenGate インストール内の各 Extract グループは、BRDIR で指定されたディレクトリの下に各自のサブディレクトリを持ちます。各サブディレクトリには、関連する Extract グループの名前が付けられます。</p>
BRINTERVAL <interval><unit>	<p>Bounded Recovery チェックポイント間の時間を指定します。この時間を <i>Bounded Recovery 間隔</i> と呼びます。この間隔は、CHECKPOINTSECS パラメータで制御される標準 Extract チェックポイント間隔の整数の倍数です。ただし、正確な値を設定する必要はありません。Bounded Recovery は、必要に応じて内部で有効な BRINTERVAL パラメータを調整します。</p> <p><interval> の最小値は 20 分です。最大値は 96 時間です。</p> <p><unit> は次のように指定できます。</p> <ul style="list-style-type: none"> ◆ M (分の場合) ◆ H (時間の場合) <p>デフォルト間隔は 4 時間です。</p>
BRKEEPSTALEFILES	<p>古い Bounded Recovery チェックポイント・ファイルを保持します。デフォルトでは、現在のチェックポイント・ファイルのみが保持されます。Extract は、古い Bounded Recovery チェックポイント・ファイルからはリカバリできません。古いファイルは、Oracle サポート・アナリストに指示された場合にのみ保持するようにしてください。</p>
BROFF	<p>Bounded Recovery の実行およびリカバリを無効にします。このオプションを使用する前に Oracle に連絡してください。通常の場合、Bounded Recovery に問題があると、Bounded Recovery は自身を無効化します。</p>
BROFFONFAILURE	<p>エラー発生後、Bounded Recovery を無効化します。デフォルトでは、Extract は Bounded Recovery 処理中にエラーを検出すると、標準リカバリに切り替えますが、リカバリの完了後 Bounded Recovery をもう一度有効化します。BROFFONFAILURE では、Bounded Recovery の実行時処理を無効化します。</p>

引数	説明
BRRESET	<p>注意: このオプションを使用するには、Extract をコマンドラインから起動する必要があります。『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p> <p>BRRESET では、現在の実行の間、Bounded Recovery に標準リカバリののみを使用させます。このパラメータは、エラーを検出しても Bounded Recovery が標準モードに戻らない、まれなケースで使用します。Bounded Recovery は、実行時に有効化されます。このオプションを使用する前に Oracle に連絡してください。</p>

例 BR BRDIR /user/checkpt/br では、Bounded Recovery チェックポイント・ファイルの作成先を /user/checkpt/br ディレクトリに指定します。

BULKLOAD

適用対象 Replicat

初期ロード Replicat 用の BULKLOAD パラメータは、SQL*Loader へのダイレクト・バルク・ロード方法を使用するときに使用します。この方法では、ダイレクト・ロードを実行するために、初期ロード・データを Oracle の SQL*Loader ユーティリティのインタフェースに直接渡します。Collector プロセスとトレイルは使用されません。

BULKLOAD パラメータを使用するときは、Extract パラメータ・ファイルの NOBINARYCHARS パラメータを使用します。 NOBINARYCHARS を使用する前に、Oracle サポートに連絡してください。 詳細は、<http://support.oracle.com> を参照してください。

Oracle GoldenGate でのデータ・ロード方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

デフォルト なし

構文 BULKLOAD

CACHEMGR

適用対象 Extract (z/OS 上の DB2 および NonStop SQL/MX を除くすべてのデータベース)

CACHEMGR パラメータは、コミットされていないトランザクション・データのキャッシュに使用可能な仮想メモリーの量および一時ディスク領域の制御に使用します。

警告

このパラメータをデフォルト・キャッシュ設定から変更する前に、Oracle サポートに連絡してください。Oracle GoldenGate のキャッシュ・マネージャは、内部で自動構成および自動調整を行うため、ほとんどの本番環境ではこのパラメータを変更する必要はありません。ただし、ページ・ファイルのディレクトリは、Oracle サポートの支援なしで指定できます。

メモリー管理について

注意

ここにはできるかぎり正確に説明していますが、メモリー管理コンポーネントの基本設計は、製品の継続的な改善のために変更されることがあります。

Oracle GoldenGate はコミットされたトランザクションのみをレプリケートするため、トランザクションのコミットまたはロールバックを受信するまで、キャッシュと呼ばれる管理仮想メモリー・プールに各トランザクションの操作を保持します。1 つのグローバル・キャッシュが 1 つの Extract プロセスの共有リソースとして機能します。このグローバル・キャッシュから、次の仮想メモリー・サブプールが割り当てられます。

- 大半のトランザクション行データ用としてログ・リーダー・スレッド当たり 1 つのサブプール
- BLOB データおよびその他の大きなアイテム用として 1 つのサブプール

各サブプール内に、グローバル・キャッシュから個別のバッファが割り当てられます。各バッファには、Oracle GoldenGate が処理中の 1 つのトランザクションに関連する情報が保持されます。初期バッファおよび増分バッファのサイズは、CACHEMGR の CACHEBUFFERSIZE オプションで制御します。

Oracle GoldenGate キャッシュ・マネージャでは、Oracle GoldenGate による持続的および効率的な作業の処理が可能になるように、オペレーティング・システムのメモリー管理機能が利用されます。キャッシュ内では、次のような最新の仮想メモリー方法が利用されます。

- アクティブ・バッファの効率的な割当ておよび管理を行う。
- 可能であれば、ディスクへのページングのかわりに古いバッファを再利用する。
- 必要に応じて、あまり使用されない情報をディスクにページングする。

Oracle GoldenGate プロセスで使用する物理メモリーの実際の量はいずれも、Oracle GoldenGate プログラムではなくオペレーティング・システムで制御されます。

キャッシュ・マネージャでは、グローバル・キャッシュ・サイズのソフト・リミット内で機能する Oracle GoldenGate プロセスが保持され、必要に応じて仮想メモリー（物理メモリーではなく）のみが割り当てられます。キャッシュ・サイズ増加のためのシステム・コールは最後の手段としてのみ行われ、使用された場合は常にシステムへの仮想メモリーの解放が続けて行われます。

システムには、実行される各 Oracle GoldenGate Extract および Replicat プロセスに十分なスワップ領域が必要です。必要なスワップ領域を決定するには、次のようにします。

1. Extract または Replicat をいずれか 1 つ起動します。
2. GGSCI を実行します。
3. レポート・ファイルを確認し、PROCESS VM AVAIL FROM OS (min) 行を見つけます。
4. 必要に応じて、値に小数点が含まれないように、ギガバイトに繰り上げます。たとえば、1.76GB は 2GB に繰り上げます。
5. 実行している Extract および Replicat プロセスの数でその値を乗算します。その結果が、必要なスワップ領域の最大量となります。必要なプロセス数を決定するには、『Oracle GoldenGate Windows and UNIX 管理者ガイド』の構成に関する各章を参照してください。

すべての Oracle GoldenGate プロセスが使用する実際の物理メモリー量は、Oracle GoldenGate プロセスではなく、オペレーティング・システムによって制御されます。グローバル・キャッシュ・サイズは、CACHEMGR の CACHESIZE オプションで制御します。

注意 キャッシュ・マネージャは、Oracle GoldenGate によって、Extract の BLOB 用サブプールやその他のトランザクション・データ用のサブプールの制御以外の目的でも内部で使用されます。統計を表示するときに、このような追加メモリー・プールの情報が表示されることがあります。

CACHEMGR を調整する時期

メモリー・マネージャが生成する統計は、SEND EXTRACT または SEND REPLICAT コマンドを CACHEMANAGER オプションとともに使用すると表示できます。この統計は、メモリー・プールのサイズ、ページングの頻度、トランザクションのサイズ、システム・プロファイルを作成するその他の情報を表示します。

このプロファイルを確認し、パフォーマンスの問題がファイル・キャッシングに関連していると思われるときは、メモリー・キャッシュの調整が必要になることがあります。その場合は、まず CACHESIZE および CACHEPAGEOUTSIZE パラメータを変更します。生成されているトランザクションのサイズとタイプに応じて、より大きなまたは小さなキャッシュ・サイズ、より大きなまたは小さなページ・サイズ、またはこれら 2 つを組み合わせる必要があります。CACHEBUFFERSIZE オプションを使用して、初期メモリー割当ての調整が必要になる場合もあります。

ただし、オペレーティング・システムの制約によって、CACHEMGR パラメータでのコンポーネント変更の効果が制限されることがあります。特に、オペレーティング・システムでプロセス当たりの仮想メモリーの制限が低い値に設定されている場合は、CACHEMGR の構成にかかわらず、多くのファイル・キャッシングを強制されます。

32 ビット Windows マシンで、頻繁なファイル・キャッシングによってパフォーマンスが低下している場合は、追加のメモリーを割り当てるために、Boot.ini ファイルで /3GB スイッチの使用を試みることができます。/3GB スイッチは、デフォルトの 2GB 制限のかわりに、3GB の仮想アドレス領域を割り当てます。Oracle GoldenGate で /3GB スイッチを使用する前に、Oracle テクニカル・サポートに連絡してください。

キャッシュ・マネージャ統計の使用の詳細は、33 ページの「SEND EXTRACT」を参照してください。

レポート・ファイルの基本統計の表示

キャッシュ・マネージャは、自身の初期化を完了すると、Extract レポート・ファイルに次のような統計を書き込みます。

```
CACHEMGR virtual memory values (may have been adjusted)
CACHEBUFFERSIZE: 64K
CACHEBUFFERSIZE (soft max): 4M
CACHESIZE: 1G
CACHEPAGEOUTSIZE (normal): 4M
PROCESS VM AVAIL FROM OS (min): 1.79G
CACHESIZEMAX (strict force to disk): 1.58G
```

条件:

- CACHEBUFFERSIZE は、デフォルトの初期バッファ割当て単位を示します。この値はオペレーティング・システムによって決定されます。このバッファ・サイズは、オペレーティング・システムに許可されている最小初期割当て単位より小さい値には設定できません。
- CACHEBUFFERSIZE (soft max) は、CACHEBUFFERSIZE オプションで指定されているバッファ割当ての最大デフォルト・サイズを示します。
- CACHESIZE は、Extract がトランザクション・データのキャッシュに使用できる仮想メモリー量を示します。これは、PROCESS VM AVAIL FROM OS (min) の値に基づいて動的に決定されます。CACHEMGR の CACHESIZE オプションを使用して制御できます。
- CACHEPAGEOUTSIZE (normal) はしきい値で、この値を超えると必要に応じてトランザクション・データをディスクにページングできます。CACHEMGR の CACHEPAGEOUTSIZE オプションを使用して制御できます。
- PROCESS VM AVAIL FROM OS (min) は、このプロセスが使用可能と判断したおおよその仮想メモリー量を示します。内部的な理由で、この量はオペレーティング・システムによって使用可能と表示される量より少ない場合があります。

- CACHESIZEMAX (strict force to disk) は、PROCESS VM AVAIL FROM OS および CACHESIZE から導出されます。これは、キャッシュ・マネージャがディスクへのページング候補のトランザクションを決定する手段と解釈できます。通常は、現在仮想メモリ・バッファが CACHEPAGEOUTSIZE を超えているトランザクションのみがページングの候補です。メモリ・リクエストの合計が CACHESIZE の値を超えると、キャッシュ・マネージャはディスクに書き込むトランザクションを探し、ページング候補のリストからトランザクションを選択します。ページング候補のトランザクションがすでにディスクにページングされていて、使用中の仮想メモリが CACHESIZEMAX (strict force to disk) を超えている場合は、追加のバッファを必要とするすべてのトランザクションをページング候補にできます。このような方法で、常に仮想メモリの可用性が確保されています。使用中のメモリが CACHESIZEMAX を下回ると、CACHEPAGEOUTSIZE ルールが再度適用されます。

ページング・ディレクトリの特定

デフォルトでは、Oracle GoldenGate で、Oracle GoldenGate インストール・ディレクトリの dirtmp サブディレクトリ内のディスクにスワップされるデータが保持されます。キャッシュ・マネージャは、ファイル・システム上のすべての空き領域が使用可能であると想定されます。CACHEMGR パラメータの CACHEDIRECTORY オプションを使用して、ディレクトリを割り当てることができます。

CACHEMGR 使用のガイドライン

- このパラメータは、z/OS 上の DB2 および NonStop SQL/MX を除くすべてのデータベースに有効です。
- 少なくとも 1 つの引数を指定する必要があります。CACHEMGR 単独では無効です。
- パラメータ・オプションは、任意の順番で指定できます。
- 1 つのパラメータ・ファイルでは、1 つの CACHEMGR パラメータのみの使用が許可されています。
- (ページ・ファイルのディレクトリを指定する以外に) このパラメータを正しく使用するためには、システムのプロファイル、およびアプリケーションから伝播されるトランザクションの種類を理解する必要があります。キャッシュ・マネージャは自己調整を行うため、標準の環境ではこのパラメータを変更する必要はありません。調整が必要と思われる場合は、Oracle サービス・リクエストをオープンしてください。詳細は、<http://support.oracle.com> を参照してください。

デフォルト なし

構文

```
CACHEMGR {
    [, CACHESIZE <size>]
    [, CACHEBUFFERSIZE <size>]
    [, CACHEDIRECTORY <path> [<size>] [, ...]]
    [, CACHEPAGEOUTSIZE <size>]
}
```

引数	説明
CACHESIZE <size>	<p>トランザクション・データのキャッシングにできる仮想メモリー量（キャッシュ・サイズ）の弱い制限を指定します。デフォルトは、64 ビットシステムで 8GB、32 ビットシステムで 2GB です。</p> <p>メモリーはオンデマンドで割り当てられます。デフォルトでは、キャッシュ・マネージャは、オペレーティング・システムから提供される使用可能な仮想メモリー量を動的に判断し、適切なキャッシュ・サイズを決定します。使用可能な仮想メモリーは、レポート・ファイルの PROCESS VM AVAIL FROM OS 値にレポートされます。CACHESIZE 値は、プロセスが使用可能な仮想メモリー量よりも大きい、またはこの量に非常に近い場合、拒否されるか低減されます。ただし、大規模なアドレス領域を持つシステムでは、内部制限に到達した後にキャッシュ・マネージャが使用可能なメモリーを判断することはありません。</p> <p>CACHESIZE 値は、PROCESS VM AVAIL FROM OS 自体が 2 の 2 乗値の場合を除き（そのケースではその値の半分になり）、その値から切り捨てられて常に 2 の累乗値になります。指定されたサイズがデータによって消費されると、メモリー・マネージャはシステムに新しいメモリーをリクエストする前に、ディスクにデータをページングするか、古いバッファを再利用してメモリーの解放を試みます。</p>
CACHEBUFFERSIZE <size>	<p>キャッシュ・マネージャが割り当てる初期バッファ・サイズを制御します。明示的に指定する場合でも、これは 2 の累乗値にする必要があり、内部で調整されることがあります。オペレーティング・システムの最小仮想メモリー割当て単位よりも小さい値に設定しないでください。オペレーティング・システムは、CACHEBUFFERSIZE 設定にかかわらず、オペレーティング・システムの最小サイズを割り当てます。CACHEBUFFERSIZE のデフォルトはオペレーティング・システムのデフォルト・サイズで、ほとんどのプラットフォームでは 64KB、64 ビット Solaris では 1MB です。</p> <p>あるトランザクションが必要とするキャッシュ仮想メモリーが CACHEBUFFERSIZE を上回ると、このトランザクションの現在のキャッシュ・データのサイズ、新しいデータに必要なサイズ、すべてのトランザクションで使用されている仮想メモリー量などの要因に基づいて、キャッシュ・マネージャによって割り当てられる追加のキャッシュ・メモリー量が動的に決定されます。</p> <p>キャッシュ・マネージャの統計のアプリケーションの使用プロファイルにおいて、大半のトランザクションがほぼ同じ量のデータをキャッシュしていることが判明した場合、ファイル・キャッシングの増加につながらないかぎり、CACHEBUFFER サイズを増やすことによって、効率性が少し高まる可能性があります。</p>

引数	説明
CACHEPAGEOUTSIZE <size>	<p>必要な場合、この値を超えたときにトランザクション・データをディスクにページングするしきい値を設定します。システム・オーバーヘッドの不必要な使用を回避するため、キャッシュ・マネージャは次の状況に当てはまる場合にのみトランザクションをファイル・システムにページングします。</p> <ul style="list-style-type: none"> ◆ トランザクションがページングの候補である。 ◆ CACHESIZE に基づくと、新しいメモリー・リクエストを満たすための十分な空き仮想メモリーがない。 <p>例外的に、他のトランザクションがファイル・キャッシングの候補になることがあります。</p> <p>デフォルト・サイズはおおよそ 4MB です。1MB より小さい値を使用しないでください。</p> <p>この値は、バイト、または次のいずれかの形式の GB、MB または KB で指定できます。</p> <p>GB MB KB G M K gb mb kb g m k</p> <p>システムにパフォーマンスの問題があると考えられる場合、このパラメータを変更することでパフォーマンスを改善できることがあります。それ以外の場合は、この値はデフォルトのままにしてください。多数の大規模な同時トランザクションや、1 つまたは 2 つの非常に大規模な長時間におよぶトランザクションが発生している環境では、ページングのしきい値がパフォーマンスの問題の原因になることがあります。たとえば、メモリーに制約があるシステム上で、多数の LOB データを含む大規模なトランザクションが行われている場合です。こうした状況では、キャッシュ・マネージャの統計を活用して、次の内容を確認します。</p> <ul style="list-style-type: none"> ◆ BLBO メモリー・プールにキャッシュされているトランザクションのサイズの分布。 ◆ ログ・リーダーのメモリー・プール。 ◆ ファイル・キャッシングに関する統計。 <p>このパラメータを変更する前に、Oracle サポートに連絡してください。詳細は、http://support.oracle.com を参照してください。</p>
CACHEDIRECTORY <path> [<size>]	<p>Oracle GoldenGate が必要なときに一時的にディスクに書き込むトランザクション・データを保持するディレクトリの名前を指定します。このパラメータを指定しない場合のデフォルトは、Oracle GoldenGate インストール・ディレクトリの dirtmp サブディレクトリです。一時ファイル用のディレクトリは、AIX などの mmap() システム・コールをサポートしていないダイレクト I/O またはコンカレント I/O が有効なマウントされたファイル・システムには配置できません。</p>

引数	説明
	<ul style="list-style-type: none"> ◆ <path> は、完全修飾ディレクトリ名です。 ◆ <size> には、指定したディレクトリに割当て可能な最大ディスク領域量を指定します。上限は、最大ファイル・サイズやファイル数など、ファイル・システムによって決められている値です。最小サイズは、強制的に 2GB になります。デフォルト値はありません。リソースの制限によって、Oracle GoldenGate が使用するスワップ領域を制限する必要がないかぎり、このオプションは使用しないでください。 <p>ディレクトリごとに CACHEDIRECTORY 句を使用すると、複数のディレクトリを指定できます。</p> <p>この値は、バイト、または次のいずれかの形式の GB、MB または KB で指定できます。</p> <p>GB MB KB G M K gb mb kb g m k</p>

例 CACHEMGR CACHEBUFFERSIZE 64KB, CACHESIZE 500MB, CACHEDIRECTORY /ggs/temp, CACHEDIRECTORY /ggs2/temp

CHECKMINUTES

適用対象 Manager

CHECKMINUTES パラメータでは、Manager がメンテナンス・アクティビティを行う間隔を制御します。このパラメータの値を低くすると、トレイル・ファイルのロールオーバーを頻繁に行う場合にはパフォーマンスに大きな影響が及ぶことがあります。プロセスの異常終了などの他のイベントが発生した場合にも、メンテナンス・サイクルがトリガーされます。

デフォルト 10 分間隔

構文 CHECKMINUTES <minutes>

引数	説明
<minutes>	メンテナンス・アクティビティを実行する間隔（分）。

例 CHECKMINUTES 15

CHECKPARAMS

適用対象 Extract および Replicat

CHECKPARAMS パラメータでは、パラメータ・ファイルの構文をテストします。テストを開始するには、次の手順を実行します。

1. パラメータ・ファイルを編集して CHECKPARAMS を追加します。
2. (オプション) 表を検証する場合には、NODYNAMICRESOLUTION パラメータを追加します。
3. プロセスを起動します。Oracle GoldenGate は、データ処理は行わず、構文を検証します。NODYNAMICRESOLUTION が存在する場合、Oracle GoldenGate はデータベースに接続して、TABLE または MAP で指定されている表が存在するかどうか検証します。構文エラーがある場合、このプロセスはエラー 190 とともに異常終了します。構文にエラーがない場合、プロセスは停止し、パラメータを正常に処理したことを示すメッセージをレポート・ファイルに書き込みます。
4. 次のいずれかを実行します。
 - テストが成功した場合は、パラメータ・ファイルを編集して CHECKPARAMS パラメータ（および使用している場合は NODYNAMICRESOLUTION パラメータ）を削除した後、プロセスを起動して処理を再開します。
 - テストが失敗した場合は、パラメータ・ファイルを編集し、レポートの結果に基づいて構文を修正した後、NODYNAMICRESOLUTION を削除してプロセスをもう一度起動します。

CHECKPARAMS は、パラメータ・ファイルに任意の位置に配置できます。

デフォルト なし

構文 CHECKPARAMS

CHECKPOINTSECS

適用対象 Extract および Replicat

Extract および Replicat でのルーチン・チェックポイントの回数を制御するには、CHECKPOINTSECS パラメータを使用します。

- この値を低くすると、チェックポイントの作成頻度が増えます。この場合、プロセスが失敗したときに再処理が必要なデータ量が減りますが、データのディスクへの書き込み頻度が増えるため、パフォーマンスが低下することがあります。
- この値を高くすると、チェックポイントの作成頻度が減ります。この場合、パフォーマンスは向上しますが、プロセスが失敗したときに再処理が必要なデータ量は増加します。Extract チェックポイントの作成頻度を少なくするときは、データの再処理が必要になるときに備えて、常にトランザクション・ログを利用できる状態にしてください。

注意 Replicat は、ルーチン・チェックポイントの他に、トランザクションをコミットするときにもチェックポイントを作成します。

CHECKPOINTSECS を変更するときは、最初に必ず Oracle サービス・リクエストをオープンしてください。詳細は、<http://support.oracle.com> を参照してください。

デフォルト 10 秒

構文 CHECKPOINTSECS <seconds>

引数	説明
<seconds>	チェックポイントを発行するまでの待機する時間（秒）。

例 CHECKPOINTSECS 20

CHECKPOINTTABLE

適用対象 GLOBALS

GLOBALS パラメータ・ファイルの CHECKPOINTTABLE パラメータでは、1 つ以上の Oracle GoldenGate インスタンス内のすべての Replicat グループが使用できる、デフォルトのチェックポイント表の名前を指定します。ADD REPLICAT コマンドで作成されたすべての Replicat グループは、このコマンドの CHECKPOINTTABLE オプションを使用して上書きされないかぎり、デフォルトでこの表を使用します。

チェックポイント表を作成するには、GGSCI で ADD CHECKPOINTTABLE コマンドを使用します。

デフォルト なし

構文 CHECKPOINTTABLE <owner.table>

引数	説明
owner.table	チェックポイント表の所有者および名前。

例 CHECKPOINTTABLE ggs.chkpt

CHECKSEQUENCEVALUE | NOCHECKSEQUENCEVALUE

適用対象 Replicat

CHECKSEQUENCEVALUE および NOCHECKSEQUENCEVALUE パラメータでは、Replicat が、ターゲットの順序値がソースの順序値より高いことを検証するかどうかを制御します。ターゲットの値が低すぎるか高すぎる場合、Replicat はソースおよびターゲットの値を適切な差異の状態に戻します。

Oracle GoldenGate では、ターゲット・シーケンスの値が次のように設定されます。

- 増分間隔が正の値である場合に、ソース値よりも大きい。
- 増分間隔が負の値である場合に、ソース値よりも小さい。

増分方向に基づき、Replicat では、挿入を実行する際のテストとして次の式のいずれかが適用されます。

```
source_highwater_value + (source_cache_size * source_increment_size * source_RAC_nodes) <=
target_highwater_value
```

または

```
source_highwater_value + (source_cache_size * source_increment_size * source_RAC_nodes) >=
target_highwater_value
```

式の結果が FALSE である場合、ターゲット・シーケンスはソース値よりも大きい値（シーケンスが増分される場合）またはソース値よりも小さな値（シーケンスが減分される場合）に更新されます。ターゲットは常に、式のカッコ内の式の前にある、またはそれと等しいものである必要があります。たとえば、ソースの水位標値が 40、CACHE が 20、ソースの INCREMENTBY 値が 1、および 2 つのソース RAC ノードがある場合、ターゲットの水位標値は最低 80 である必要があります。

```
40 + (20*1*2) < 80
```

ターゲットの水位標値が 80 未満である場合、Oracle GoldenGate は、ターゲットがソースの前にあり続けるように、シーケンスを更新して水位標値を増分します。現在の水位標値を取得するには、次の問合せを実行します。

```
SELECT last_number FROM all_sequences WHERE sequence_owner=upper('SEQUENCEOWNER')
AND sequence_name=upper('SEQUENCENAME');
```

（トレイルの破損やプロセスの失敗などによって）順序の更新で差異が生じないことを把握していて、Oracle GoldenGate のパフォーマンスを高める必要があるときを除き、デフォルトの CHECKSEQUENCEVALUE を使用してください。

デフォルト CHECKSEQUENCEVALUE

構文 CHECKSEQUENCEVALUE | NOCHECKSEQUENCEVALUE

CMDTRACE

適用対象 Extract および Replicat

CMDTRACE パラメータでは、レポート・ファイルにマクロ展開手順を表示します。このパラメータは、異なるマクロに異なるオプションを設定するために、パラメータ・ファイルで複数回使用できます。

デフォルト OFF

構文 CMDTRACE [ON | OFF | DETAIL]

引数	説明
ON	マクロ展開の表示を有効化します。
OFF	マクロ展開の表示を無効化します。
DETAIL	マクロ展開の詳細な表示を生成します。

例 次の例では、**#testmac** を起動する前にトレースを有効化し、マクロの実行後に無効化します。

```
MACRO #testmac
BEGIN
col1 = col2,
col3 = col4
END;
...
CMDTRACE ON
MAP test.table2 , TARGET test.table2,
COLMAP (#testmac);
CMDTRACE OFF
```

COLMATCH

適用対象 Extract および Replicat

COLMATCH パラメータでは、列マッピングのグローバル・ルールを作成します。COLMATCH ルールは、COLMATCH 文に続くすべての TABLE または MAP 文に適用されます。グローバル・ルールを以降の TABLE または MAP エントリに対して無効にするには、RESET オプションを使用します。

COLMATCH を使用すると、同様の構造を持っていて、同じデータ・セットを異なる列名に保持している表同士をマップできます。COLMATCH では、個別の TABLE または MAP 文で COLMAP 句を使用するよりも、このタイプの列を効率的にマップできます。

COLMATCH では、次のことを実行できます。

- 列名に基づいて明示的にマップする
- 名前の接頭辞またはサフィックスを無視する

名前が異なるソース列とターゲット列をマッピングする場合は、COLMATCH か、TABLE または MAP 文の COLMAP 句を使用する必要があります。

列マッピングの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

デフォルト なし

構文

```
COLMATCH
{NAMES <target column> = <source column> |
PREFIX <prefix> |
SUFFIX <suffix> |
RESET}
```

引数	説明
NAMES <target column> = <source column>	ターゲットおよびソースの列名を指定します（例：CUSTOMER_CODE およ び CUST_CODE）。
<prefix>	無視する列名接頭辞を指定します。たとえば、ORDER_ID という名前の ターゲット列を P_ORDER_ID という名前のソース列にマップする場合は、 次のように指定します。 COLMATCH PREFIX P_
<suffix>	無視する列名サフィックスを指定します。たとえば、CUST_CODE_K という 名前のターゲット列を CUST_CODE という名前のソース列にマップする場 合は、次のように指定します。 COLMATCH SUFFIX _K
RESET	前に定義済の COLMATCH ルールを以降の TABLE または MAP 文で無効化し ます。

例 1 COLMATCH NAMES CUSTOMER_CODE = CUST_CODE
例 2 COLMATCH PREFIX P_
例 3 COLMATCH SUFFIX _K
例 4 COLMATCH RESET

COMMENT | --

適用対象 Manager、Extract、Replicat

COMMENT パラメータまたは 2 連ハイフン (--) では、パラメータ・ファイル内のコメントを示します。
COMMENT または 2 連ハイフン以降の同一行内の記述は、処理中に無視されます。

COMMENT または 2 連ハイフンを使用して、パラメータ・ファイル内のどの位置にもコメントを記述で
きます。次の行にコメントが続く場合は、次の行頭に COMMENT キーワードまたは 2 連ハイフンをもう一度
記述する必要があります。

注意 同期されている表の列に "comment" という単語が含まれている場合は、COMMENT パラメー
タと競合する可能性があります。その場合には 2 連ハイフンを使用してください。

デフォルト なし

構文 {COMMENT <comment text>} | {-- <comment text>}

例 1 COMMENT GoldenGate param file for fin Extract group.

例 2 -- GoldenGate param file for fin Extract group.

COMPRESSDELETES | NOCOMPRESSDELETES

適用対象 Extract

ログベース Extract グループ用の COMPRESSDELETES および NOCOMPRESSDELETES パラメータでは、削除操作の際にトレイル・レコードを列に書き込む方法を制御します。

デフォルトの COMPRESSDELETES では、削除操作の場合、Extract は主キーのみをトレイルに書き込みます。主キーのみを書き込むと、処理が必要なデータ量を制限しながら、適切なターゲット・レコードの削除に十分な情報を提供できます。

NOCOMPRESSDELETES では、すべての列をトレイルに送信します。表定義に主キーまたは一意索引が含まれていない場合は、これがデフォルトになります。TABLE の KEYCOLS オプションで代替キーが定義されている場合は、実際のキーが定義されているかどうかにかかわらず、これらの列がトレイルに書き込まれます。

COMPRESSDELETES および NOCOMPRESSDELETES は、パラメータ・ファイル内のすべての TABLE 文にグローバルに適用することも、各 TABLE 文に個別スイッチとして使用することもできます。

これらのパラメータは、データ・ポンプには影響しません。

デフォルト COMPRESSDELETES

構文 COMPRESSDELETES | NOCOMPRESSDELETES

COMPRESSUPDATES | NOCOMPRESSUPDATES

適用対象 Extract

Extract 用の COMPRESSUPDATES および NOCOMPRESSUPDATES パラメータでは、更新操作の際にトレイル・レコードを列に書き込む方法を制御します。

デフォルトの COMPRESSUPDATES では、更新操作の場合、Extract は主キーおよび行の変更された列のみをトレイルに書き込みます。これらを書き込むと、処理が必要なデータ量を制限しながら、適切なターゲット・レコードの更新に十分な情報を提供できます。

NOCOMPRESSUPDATES では、すべての列をトレイルに送信します。表定義に主キーまたは一意索引が含まれていない場合は、これがデフォルトになります。TABLE パラメータの KEYCOLS オプションで代替キーが定義されている場合は、実際のキーが定義されているかどうかにかかわらず、これらの列がトレイルに書き込まれます。

COMPRESSUPDATES および NOCOMPRESSUPDATES は、パラメータ・ファイル内のすべての TABLE 文にグローバルに適用されます。

このパラメータは、次のデータベースのみをサポートしています。

- DB2 LUW および z/OS 上の DB2
- Teradata バージョン 12 以上
- SQL Server
- Sybase

COMPRESSUPDATES および NOCOMPRESSUPDATES は、データ・ポンプには影響しません。

デフォルト COMPRESSUPDATES

構文 COMPRESSUPDATES | NOCOMPRESSUPDATES

CUSEREXIT

適用対象 Extract および Replicat

Oracle GoldenGate 処理内の定義済 exit ポイントで、Windows DLL または UNIX 共有オブジェクトから C プログラミング・コードで記述されたカスタム exit ルーチンと呼び出すには、CUSEREXIT パラメータを使用します。ユーザーの exit ルーチンは、Extract および Replicat プロセスからの様々なイベントや情報の受信、必要に応じて情報の処理、およびコール元（それを呼び出した Oracle GoldenGate プロセス）へのレスポンスや情報の返信が可能である必要があります。

ユーザー・イグジットは、Oracle GoldenGate ソリューション内で使用可能なデータ変換ファンクションのかわりとしても、こうしたファンクションと一緒に使用できます。

ユーザー・イグジットの作成および実装の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

デフォルト なし

構文 CUSEREXIT <DLL or shared object name> <routine name>
[, PASSTHRU]
[, INCLUDEUPDATEBEFORES]
[, PARAMS "<startup string>"]

引数	説明
<DLL or shared object name>	ユーザー・イグジット・ファンクションを含む Windows DLL または UNIX 共有オブジェクトの名前。
<routine name>	実行するイグジット・ルーチン名。
PASSTHRU	<p>Extract データ・ポンプにのみ有効です。データベースが不要で、出力トレイルが許可されていないとみなされます。ユーザー・イグジットがすべての処理を実行し、Extract がレコードをスキップするとみなされます。Extract は、ユーザー・イグジットにレコードを渡す前にすべての必要なデータマッピングを行います。</p> <p>レスポンスのステータスは、EXIT_OK_VAL ではなく EXIT_PROCESSED_REC_VAL になります。すべての処理統計は、レコードが Oracle GoldenGate によって処理されたように更新されます。</p>

引数	説明
INCLUDEUPDATEBEFORES	<p>列値のビフォア・イメージをユーザー・イグジットに渡します。このパラメータを使用するときは、<code>requesting_before_after_ind</code> フラグをサポートするコールバック・ファンクション内でこのフラグを <code>BEFORE_IMAGE_VAL</code> に設定し、ビフォア・イメージを明示的にリクエストする必要があります。この設定を行わないと、アフター・イメージのみがユーザー・イグジットに渡されます。デフォルトでは、Oracle GoldenGate はアフター・イメージのみを使用します。</p> <p>データ・ポンプまたは Replicat からコールされるユーザー・イグジットに対して INCLUDEUPDATEBEFORES を使用する場合は、常にプライマリ Extract プロセスで GETUPDATEBEFORES パラメータを使用することにより、ビフォア・イメージを取得してトレイルに書き込み、ユーザー・イグジットで <code>process_record</code> イベントを発生させます。プライマリ Extract にもユーザー・イグジットが含まれている場合は、GETUPDATEBEFORES によってビフォアおよびアフター・イメージ両方が別々の <code>EXIT_CALL_PROCESS_RECORD</code> イベントとしてユーザー・イグジットに送信されます。</p> <p>ユーザー・イグジットが（トランザクション・ログを読み込む）プライマリ Extract からコールされる場合、この Extract に必要なのは INCLUDEUPDATEBEFORES のみです。このケースでは、他のダウンストリームの Oracle GoldenGate プロセスがトレイルへのビフォア・イメージ書き込みを必要としないかぎり、GETUPDATEBEFORES は必要ありません。INCLUDEUPDATEBEFORES を指定しても、ビフォア・イメージはトレイルに書き込まれません。</p>
PARAMS "<startup string>"	<p>指定した文字列を起動時に渡します。プロパティ・ファイル、起動パラメータ、その他の文字列を渡すために使用できます。文字列は、二重引用符で囲む必要があります。</p> <p>この文字列のデータは、<code>EXIT_CALL_START exit_params_def.function_param</code> のユーザー・イグジットに渡されます。PARAMS で、引用符で囲まれた文字列が指定されない場合、<code>exit_params_def.function_param</code> は NULL です。</p>

例 1	CUSEREXIT userexit.dll MyUserExit
例 2	CUSEREXIT userexit.dll MyUserExit, PARAMS "init.properties"
例 3	CUSEREXIT userexit.dll MyUserExit, INCLUDEUPDATEBEFORES, & PARAMS "init.properties"
例 4	CUSEREXIT userexit.dll MyUserExit, INCLUDEUPDATEBEFORES, PASSTHRU, & PARAMS "init.properties"
例 5	CUSEREXIT cuserexit.dll MyUserExit, & PASSTHRU, & INCLUDEUPDATEBEFORES, & PARAMS "Some text to start with during startup"

DBOPTIONS

適用対象 Extract および Replicat

DBOPTIONS パラメータでは、データベース・オプションを指定します。これはグローバル・パラメータで、パラメータ・ファイル内のすべての TABLE または MAP 文に適用されます。DBOPTIONS は、TARGETDB か SOURCEDB パラメータ文、または USERID 文（あるいはその両方）よりも先に指定する必要があります。一部のオプションは、Extract または Replicat のみに適用されます。

デフォルト なし

構文

```

DBOPTIONS
[ALLOWLOBDATATRUNCATE | NOALLOWLOBDATATRUNCATE]
[ALLOWUNUSEDCOLUMN]
[CATALOGCONNECT | NOCATALOGCONNECT]
[CONNECTIONPORT <port>]
[DEFERREFCONST]
[DISABLELOBCACHING]
[EMPTYLOBSTRING '<string>']
[FETCHBATCHSIZE <num_recs>]
[FETCHLOBS | NOFETCHLOBS]
[HOST <host ID>]
[LIMITROWS | NOLIMITROWS]
[LOBBUFSIZE]
[LOBWRITESIZE <size>]
[SHOWINFOMESSAGES]
[SHOWWARNINGS]
[SPTHREAD | NOSPTHREAD]
[SUPPRESSTRIGGERS | NOSUPPRESSTRIGGERS]
[TDSPACKETSIZE <bytes>]
[TRUSTEDCONNECTION]
[USEODBC]
[USEREPLICATIONUSER]
[XMLBUFSIZE <buffer size>]

```

引数	説明
ALLOWUNUSEDCOLUMN	<p>Oracle の Extract に有効です。Replicat が未使用の列を持つ表を検出したときに異常終了することを防ぎます。Replicat は、異常終了せずに処理を継続し、警告を生成します。</p> <p>このパラメータを使用する場合は、ターゲットにも同じ未使用の列が存在している必要があります。この列が存在しない場合には、正しいメタデータ・マッピングを実行できるように、Replicat にこの表のソース定義ファイルを指定する必要があります。デフォルトでは、Extract は未使用の列を検出すると異常終了します。</p>
ALLOWLOBDATATRUNCATE NOALLOWLOBDATATRUNCATE	<p>Sybase および MySQL の Replicat に有効です。</p> <p>ALLOWLOBDATATRUNCATE では、レプリケートする LOB データがターゲットの CHAR、VARCHAR、BINARY または VARBINARY 列に対して大きすぎる場合に Replicat が異常終了することを防ぎます。この LOB データは、ターゲット列の最大サイズに切り捨てられ、エラー・メッセージや警告は生成されません。</p> <p>デフォルトは NOALLOWLOBDATATRUNCATE で、レプリケートする LOB が大きすぎる場合に、Replicat を異常終了させ、エラー・メッセージを生成します。</p>

引数	説明
CATALOGCONNECT NOCATALOGCONNECT	ODBC データベースの Extract および Replicat に有効です。デフォルトでは、Oracle GoldenGate はカタログ問合せのために新しい接続を作成しますが、NOCATALOGCONNECT を使用すると新しい接続の作成を防止できます。z/OS 上の DB2 の場合、NOCATALOGCONNECT では、MVS DB2 初期化パラメータ mvsattachtype が CAF に設定されているときに、Oracle GoldenGate による複数の接続の試みを防ぎます。CAF モードでは複数の接続がサポートされていないため、Oracle GoldenGate はオープンしている接続のコミットを受信するまで、システム・カタログ表領域にコミット・ロックを発行する可能性があります。コミット・ロックを防ぐために、Oracle GoldenGate では、複数の接続をサポートする RRSF (mvsattachtype=RRSF) を使用することをお勧めします。
CONNECTIONPORT <port>	マルチデーモン MySQL の Replicat に有効です。TCP/IP ポートを指定します。
DEFERREFCONST	Oracle の Replicat に有効です。Replicat トランザクションがコミットされるまで、データベースによる参照整合性制約チェックおよび強制を延期します。ご使用のデータベースが Oracle リリース 9.2.0.7 以上の場合、ターゲット表の制約を無効化するかわりにこのパラメータを使用できます。この機能をサポートしていない Oracle リリースとともに使用する場合、DEFERREFCONST は無視され、GoldenGate ログに通知は返されません。
DISABLELOBCACHING	Oracle の Replicat に有効です。Oracle の LOB キャッシング・メカニズムを無効にします。デフォルトでは、Replicat は Oracle の LOB キャッシング・メカニズムを有効にします。
EMPTYLOBSTRING '<string>'	Replicat に有効です。ターゲットにレプリケートされる、Sybase IMAGE または TEXT 値などの空の（ゼロ長）LOB 列を、文字列値に置換します。デフォルトでは、Oracle GoldenGate は、空の列をターゲットで NULL に設定します。ターゲット・データベースが LOB 列に NULL を許可していない場合、Replicat は異常終了します。このオプションは、Replicat の異常終了を防ぐために使用します。 '<string>' には、列が受け入れる任意の文字列を指定し、一重引用符で囲みます。デフォルトは NULL です。 例： DBOPTIONS EMPTYLOBSTRING 'empty'

引数	説明
FETCHBATCHSIZE <num_recs>	<p>Oracle、DB2、Ingres、SQL/MX、Sybase、SQL Server、Sybase および Teradata の Extract に有効です。初期ロードのパフォーマンスを高めるために、1 行ごとのフェッチのかわりに配列フェッチを有効にします。有効な値は、フェッチ当たり 0 ～ 1000000 レコードです。デフォルトは 1000 です。バッチ・サイズが非常に小さくなるか非常に大きくなると、パフォーマンスは低下します。表に LOB データが含まれている場合、Extract は 1 行フェッチ・モードに戻り、その後バッチ・フェッチモードを再開します。</p>
HOST <host id>	<p>マルチデーモン MySQL の Replicat に有効です。ホストの DNS 名または IP アドレスを指定します。</p>
FETCHLOBS NOFETCHLOBS	<p>z/OS 上の DB2 および DB2 LUW に有効です。表の LOB オプションが NOT LOGGED に設定されている場合に、データベース表からの LOB の直接のフェッチを抑止します。NOT LOGGED が設定されている場合、この列の値はトランザクション・ログからは入手できず、表自体からのみ取得できます。デフォルトでは、Oracle GoldenGate は LOB への変更をトランザクション・ログから取得します。デフォルトは、FETCHLOBS です。</p>
LIMITROWS NOLIMITROWS	<p>Oracle、SQL Server および Sybase の Replicat に有効です。LIMITROWS では、ターゲット表に主キーまたは一意キーが含まれていない場合に、同一の Replicat SQL 文によって複数の行を更新または削除することを防ぎます。WHERE 句がすでに存在するかどうかに応じて、次のいずれかの句を追加することにより、Replicat が使用する WHERE 句を変更します。</p> <p>WHERE ROWNUM = 1</p> <p>または</p> <p>AND ROWNUM = 1</p> <p>この変更は、表に一意キーが含まれていない場合にのみ適用されます。</p> <p>NOLIMITROWS では、同一の Replicat SQL 文によって複数の行を更新または削除することを許可します。このオプションは、Oracle の OCI を使用しているときは機能しません。</p> <p>デフォルトは LIMITROWS です。LIMITROWS および NOLIMITROWS は、パラメータ・ファイル内のすべての MAP 文にグローバルに適用されます。</p>
LOBBUFSIZE <bytes>	<p>Oracle の Extract に有効です。Oracle オブジェクト・タイプの各埋込み LOB 属性に割り当てるメモリー・バッファ・サイズを決定します。有効な値は、1024 ～ 10485760 バイトです。デフォルトは 1048576 バイトです。</p> <p>埋込み LOB の長さが指定の LOBBUFSIZE サイズを超えると、次のようなエラー・メッセージが生成されます。</p> <p>GG ERROR ZZ-0L3 Buffer overflow, needed: 2048, allocated: 1024.</p>

引数	説明
LOBWRITESIZE <size>	<p>Oracle の Replicat に有効です。Replicat がターゲット・データベースに書き込む各 LOB のフラグメント・サイズを指定します。LOB データは、このサイズに達するまでバッファに保持されます。LOB はフラグメントでデータベースに書き込む必要があるため、より大きなブロックで書き込むことで過度の I/O を防止できます。この値が大きいくほど、LOB 全体をデータベースに書き込むための Replicat からデータベース・サーバーへの I/O コールは少なくなります。</p> <p>Oracle LOB フラグメント・サイズの倍数を指定します。指定した値は、必要に応じて Oracle LOB フラグメント・サイズの倍数に切り上げられます。デフォルトの LOB 書き込みサイズは 32K です。有効な値は、2,048 ～ 1,048,576 バイト (1MB) です。</p> <p>デフォルトでは、Replicat は Oracle の LOB キャッシング・メカニズムを有効にします。Oracle の LOB キャッシングを無効にするには、DBOPTIONS の DISABLELOBCACHING オプションを使用します。</p>
SHOWINFOMESSAGES	<p>Sybase の Extract および Replicat に有効です。エラー・ログに次の Sybase サーバー・メッセージを出力します。</p> <pre>0: /* General informational message */ 5701: /* Changed Database Context */ 5703: /* Changed language setting */ 5704: /* Changed client character set */ 7326: /* Non ANSI Escaping */</pre> <p>Oracle GoldenGate 処理に影響がないため、通常はこれらのメッセージは出力されません。</p>
SHOWWARNINGS	<p>Sybase の Extract および Replicat に有効です。重大度レベル 10 を上回る Sybase サーバー・メッセージのログギングを有効化します。これらのメッセージは、Sybase がデータ変更を伴う修正処理を実行するときに、デバッグに役立ちます。</p>
SPTHREAD NOSPTHREAD	<p>Extract および Replicat に有効です。ストアド・プロシージャに別のデータベース接続スレッドを作成します。デフォルトは NOSPTHREAD です。</p>
SUPPRESSTRIGGERS NOSUPPRESSTRIGGERS	<p>Oracle の Replicat に有効です。Oracle GoldenGate でレプリケーションするように構成されているトリガーがターゲット・オブジェクト上で起動することを防ぎます。このパラメータは、Oracle 10.2.0.5 以上のパッチ、および Oracle 11.2.0.2 以上をご使用の場合に、トリガーを手動で無効化するかわりに使用できます。このオプションを使用するには、Replicat ユーザーは Oracle Streams 管理者である必要があります (dbms_goldengate_auth.grant_admin_privilege を起動することによってこの権限を付与できます)。</p> <p>デフォルトは NOSUPPRESSTRIGGERS です。</p>

引数	説明
TDS_PACKETSIZE <bytes>	<p>Replicat に有効です。Sybase ターゲットへのレプリケーションで使用する TDS パケット・サイズを設定します。512 の倍数に設定する必要があります。有効な値は、512 (デフォルト) ～ 8192 です。</p>
TRUSTEDCONNECTION	<p>SQL Server の Extract および Replicat に有効です。Oracle GoldenGate に、trusted connection = yes を使用して接続させます。このオプションを使用する前に Oracle サポートに連絡してください。詳細は、http://support.oracle.com を参照してください。</p>
USEODBC	<p>SQL Server の Replicat に有効です。Replicat が ODBC を使用して DML 操作を実行するように構成します。デフォルトでは、OLE DB を使用します。USEREPLICATIONUSER を有効化しているときはこの設定は無効になり、Replicat は異常終了します。</p> <p>注意: Replicat は、データベース・カタログに接続してメタデータを取得する場合は、常に ODBC を使用します。</p>
USEREPLICATIONUSER	<p>SQL Server の Replicat に有効です。SQL Server レプリケーション・ユーザーとしてターゲット DML 操作を実行するように Replicat を構成します。レプリケーション・ユーザーは、SQL Server ユーザーまたはアカウントではなく、データベース接続のプロパティです。USEREPLICATIONUSER は、SQL Server の NOT FOR REPLICATION フラグを有効化します。</p> <p>レプリケーション・ユーザーを使用する場合は、データの整合性に対する次の懸念を解消する必要があります。</p> <ul style="list-style-type: none"> ◆ ターゲットの IDENTITY シードが更新されません。ターゲットが読取り専用の場合を除き、主キー違反の回避のためにパーティショニング・スキームが必要です。 ◆ 外部キー制約が強制されません。 ◆ ON UPDATE CASCADE、ON DELETE CASCADE およびトリガーが無効化されます。重複する操作が回避されるのでこの無効化は Replicat には有益ですが、ターゲット・アプリケーションにとっては適切でなく、データ整合性の確保のために制約またはトリガーのコードの変更が必要になることがあります。 ◆ CHECK 制約が強制されないため、ターゲット上でデータ整合性を確保できません。 <p>これらの考慮事項の詳細は、『Oracle GoldenGate SQL Server インストールおよびセットアップ・ガイド』を参照してください。</p> <p>デフォルトでは、USEREPLICATIONUSER は無効化されています。USEREPLICATIONUSER は、送信パフォーマンスを高める必要がある場合にのみ使用することをお勧めします。USEODBC を有効化しているときはこの設定は無効になり、Replicat は異常終了します。</p>

引数	説明
XMLBUFSIZE <bytes>	Oracle の Extract に有効です。SDO_GEORASTER オブジェクト・タイプの <code>sys.xmltype</code> 属性から抽出された XML データを保持するメモリ・バッファのサイズを設定します。デフォルトは 1048576 バイト（1MB）です。データがデフォルトのバッファ・サイズを超えると、Extract は異常終了します。その場合は、バッファ・サイズを増やしてから、Extract を再起動してください。有効な値の範囲は、1024 ～ 10485760 バイトです。
例 1	DBOPTIONS HOST 127.0.0.1, CONNECTIONPORT 3307
例 2	DBOPTIONS TDSPACKETSIZE 2048
例 3	DBOPTIONS FETCHBATCHSIZE 2000
例 4	DBOPTION XMLBUFSIZE 2097152

DDL

適用対象 Extract および Replicat

DDL パラメータを使用して、DDL サポートの有効化および DDL 操作のフィルタ処理ができます。オプションなしで使用する場合、DDL パラメータにより、すべての DDL 操作は次のように伝播されます。

- Extract パラメータとして、すべてのサポートされるデータ・オブジェクトで生成されるすべてのサポートされる DDL 操作を取得し、証跡に送信します。
- Replicat パラメータとして、Oracle GoldenGate 証跡からのすべての DDL 操作を複製し、ターゲットに適用します。これは、このパラメータがない場合のデフォルトの動作と同じです。

オプションとともに使用される場合、DDL はフィルタ処理エージェントとして機能し、以下に基づいて DDL 操作を包含または除外します。

- スコープ
- オブジェクト・タイプ
- 操作タイプ
- オブジェクト名
- DDL コマンド構文またはコメント、あるいはその両方に含まれる文字列

1 つの DDL パラメータのみをパラメータ・ファイルに使用できますが、複数の inclusion および exclusion オプションを組み合わせる必要なレベルに DDL をフィルタ処理できます。

- 組み合わせられると、複数のオプションの指定は AND 文として論理的に連結されます。
- 複数のオプションで指定されるすべての基準は、複製される DDL 文について満たされる必要があります。
- 複雑な DDL フィルタ基準を使用する場合、本番環境で使用する前にテスト環境で構成をテストすることをお勧めします。

警告

Oracle GoldenGate によってインストールされた DDL オブジェクトを DDL パラメータ、TABLE パラメータ、MAP パラメータ、TABLEEXCLUDE パラメータまたは MAPEXCLUDE パラメータに含めないでください。これらのパラメータにワイルドカードを指定する場合、Oracle GoldenGate にインストールされた DDL オブジェクトが含まれないよう

にしてください。これらのオブジェクトは、Oracle GoldenGate 構成に含まれないようにする必要がありますが、Extract プロセスでこれらのオブジェクトに行う操作が認識される必要があるため、EXCLUDE、TABLEEXCLUDE または MAPEXCLUDE パラメータ文から明示的に除外しないでください。

次のプロセスでは、DDL を使用しないでください。

- Extract データ・ポンプ
- VAM ソート Extract (Teradata ソース・データベース)

これらのプロセス・タイプでは、DDL のマッピングまたは変換が許可されず、PASSTHRU モードで自動的に DDL レコードが伝播されます（265 ページを参照してください）。特定の名前のソース表で実行される DDL（ALTER TABLE TableA... など）は、Replicat によって同一の名前の表に適用されます（ALTER TABLE TableA）。ALTER TABLE TableB としてマッピングすることはできません。

Oracle GoldenGate DDL サポートの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

構文

```
DDL [
{INCLUDE | EXCLUDE}
    [, MAPPED | UNMAPPED | OTHER | ALL]
    [, OPTYPE <type>]
    [, OBJTYPE '<type>']
    [, OBJNAME "<name>"]
    [, INSTR '<string>']
    [, INSTRCOMMENTS '<comment_string>']
]
[...]
```

表 38 DDL の包含および除外オプション

オプション	説明
INCLUDE EXCLUDE	<p>inclusion または exclusion 句の開始部分を識別するには、INCLUDE および EXCLUDE を使用します。</p> <ul style="list-style-type: none"> ◆ inclusion 句には、このパラメータによって影響される DDL を識別するフィルタ基準が含まれます。 ◆ exclusion 句には、このパラメータから特定の DDL を除外するフィルタ基準が含まれます。 <p>inclusion または exclusion 句は、INCLUDE または EXCLUDE キーワードに続き、適用されるパラメータの他のオプションの有効な組合せで構成される必要があります。</p> <p>EXCLUDE を使用する場合、対応する INCLUDE 句を作成する必要があります。たとえば、次は無効です。</p> <pre>DDL EXCLUDE OBJNAME "hr.*"</pre> <p>ただし、次のいずれかは使用できます。</p> <pre>DDL INCLUDE ALL, EXCLUDE OBJNAME "hr.*" DDL INCLUDE OBJNAME "fin.*" EXCLUDE "fin.ss"</pre> <p>EXCLUDE は、同じ基準が含まれているいずれの INCLUDE よりも優先されます。複数の inclusion および exclusion 句を使用できます。</p>

表 38 DDL の包含および除外オプション（続き）

オプション	説明
MAPPED UNMAPPED OTHER ALL	<p>DDL 操作スコープに基づいて INCLUDE または EXCLUDE を適用するには、MAPPED、UNMAPPED、OTHER および ALL を使用します。</p> <ul style="list-style-type: none"> ◆ MAPPED は、INCLUDE または EXCLUDE を MAPPED スコープの DDL 操作に適用します。MAPPED フィルタ処理は、他の DDL パラメータ・オプションで指定されたフィルタ処理の前に実行されます。 ◆ UNMAPPED は、INCLUDE または EXCLUDE を UNMAPPED スコープの DDL 操作に適用します。 ◆ OTHER は、INCLUDE または EXCLUDE を OTHER スコープの DDL 操作に適用します。 ◆ ALL は、INCLUDE または EXCLUDE をすべてのスコープの DDL 操作に適用します。
OPTYPE <type>	<p>INCLUDE または EXCLUDE を、CREATE、ALTER、RENAME など、特定のタイプの DDL 操作に適用するには、OPTYPE を使用します。<type> には、データベースに有効な任意の DDL コマンドを使用します。たとえば、ALTER 操作を含める場合の正しい構文は次のようになります。</p> <pre>DDL INCLUDE OPTYPE ALTER</pre>
OBJTYPE '<type>'	<p>INCLUDE または EXCLUDE を特定のデータベース・オブジェクトに適用するには OBJTYPE を使用します。<type> には、TABLE、INDEX、TRIGGER など、データベースに有効な任意のオブジェクト・タイプを使用します。Oracle マテリアライズド・ビューおよびマテリアライズド・ビュー・ログの場合、正しいタイプはそれぞれ snapshot と snapshot log です。オブジェクト・タイプの名前を二重引用符で囲みます。</p> <p>例：</p> <pre>DDL INCLUDE OBJTYPE 'INDEX'</pre> <pre>DDL INCLUDE OBJTYPE 'SNAPSHOT'</pre> <p>Oracle オブジェクト・タイプ USER の場合、USER のみがスキーマを持つのにに対し、OBJNAME で "owner.object" が予期されるので、OBJNAME オプションは使用しないでください。</p>
OBJNAME "<name>"	<p>INCLUDE または EXCLUDE を、owner.table_name など、オブジェクトの完全修飾名に適用するには、OBJNAME を使用します。このオプションでは、二重引用符で囲まれた文字列を入力として使用できます。</p> <p>ワイルドカードは、オブジェクト名にのみ使用できます。</p> <p>例：</p> <pre>DDL INCLUDE OBJNAME "accounts.*"</pre> <p>OracleUSER オブジェクトの場合、USER のみがスキーマを持つのにに対し、OBJNAME で "owner.object" が予期されるので、OBJNAME は使用しないでください。</p> <p>Replicat パラメータ・ファイルで OBJNAME を MAPPED とともに使用する場合、OBJNAME の値が、MAP 文の TARGET 句で指定されている名前を参照する必要があります。たとえば、次の MAP 文では、正しい値は OBJNAME "fin2.*" となります。</p> <pre>MAP fin.exp_*, TARGET fin2.*;</pre>

表 38 DDL の包含および除外オプション（続き）

オプション	説明
	<p>次の例では、CREATE TABLE 文によりソースで実行されるものは次のようになります。</p> <pre>CREATE TABLE fin.exp_phone;</pre> <p>ターゲットについては次のようになります。</p> <pre>CREATE TABLE fin2.exp_phone;</pre> <p>ターゲット・オーナーが MAP で指定されていない場合、Replicat により USERID パラメータで指定されているデータベース・ユーザーにマッピングされます。</p> <p>トリガー、シノニム、および索引を作成する DDL の場合、OBJNAME の値はトリガー、シノニム、または索引の名前ではなく、ベース・オブジェクトの名前である必要があります。</p> <p>たとえば、次の DDL 文を含めるための正しい値は、"hr.insert_trig" ではなく、"hr.accounts" になります。</p> <pre>CREATE TRIGGER hr.insert_trig ON hr.accounts;</pre> <p>RENAME 操作の場合、OBJNAME の値は新しい表の名前である必要があります。</p> <p>たとえば、次の DDL 文を含めるための正しい値は、"hr.acct" になります。</p> <pre>ALTER TABLE hr.accounts RENAME TO acct;</pre>
INSTR '<string>'	<p>INCLUDE または EXCLUDE を、コマンド構文（自体、ただしコメント内ではなく）内の特定の文字列を含んだ DDL 文に適用するには、INSTR を使用します。たとえば、次の場合は、索引を作成する DDL は除外されます。</p> <pre>DDL INCLUDE ALL EXCLUDE INSTR 'CREATE INDEX'</pre> <p>文字列を一重引用符で囲みます。文字列の検索では大 / 小文字は区別されません。</p>
INSTRCOMMENTS '<comment_string>'	<p>INCLUDE または EXCLUDE を、コマンド構文内自体ではなく、コメント内の特定の文字列を含んだ DDL 文に適用するには、INSTRCOMMENTS を使用します。INSTRCOMMENTS を使用することで、コメントをフィルタ処理用エージェントとして使用できます。</p> <p>たとえば、次の場合、コメントに "source" が含まれている DDL 文は除外されます。</p> <pre>DDL INCLUDE ALL EXCLUDE INSTRCOMMENTS 'SOURCE ONLY'</pre> <p>この例では、次のような DDL 文は複製されません。</p> <pre>CREATE USER john IDENTIFIED BY john /*source only*/;</pre> <p>文字列を一重引用符で囲みます。文字列の検索では大 / 小文字は区別されません。INSTR と INSTRCOMMENTS を組み合わせて、同じ DDL 文のコマンド構文内およびコメント内の文字列をフィルタ処理できます。</p>

表 38 DDL の包含および除外オプション（続き）

オプション	説明
INSTRWORDS '<word list>'	<p>INCLUDE または EXCLUDE を特定の単語を含んだ DDL 文に適用するには、INSTRWORDS を使用します。</p> <p><word list> には、複数の単語を任意の順序で指定し、これらを一重引用符で囲みます。空白を含める場合は、二重引用符内に空白（該当する場合は単語も）を入れます。二重引用符は、文を囲む場合にも使用できます。</p> <p>INSTRWORDS が有効になるには、指定した単語がすべて DDL に存在する必要があります。</p> <p>例：</p> <pre>ALTER TABLE INCLUDE INSTRWORDS 'ALTER CONSTRAINT " xyz"'</pre> <p>この例は以下と一致します。</p> <pre>ALTER TABLE ADD CONSTRAINT xyz CHECK</pre> <p>および</p> <pre>ALTER TABLE DROP CONSTRAINT xyz</pre>
INSTRCOMMENTSWORDS '<word list>'	<p>INSTRWORDS と同様に機能しますが、DDL 構文自体ではなく DDL 文内のコメントにのみ適用されます。INSTRCOMMENTS を使用することで、コメントをフィルタ処理用エージェントとして使用できます。</p> <p>INSTRWORD と INSTRCOMMENTSWORDS を組み合わせて、同じ DDL 文のコマンド構文内およびコメント内の文字列をフィルタ処理できます。</p>

例 次に、DDL パラメータ・オプションを組み合わせ使用する例を示します。

```
DDL &
INCLUDE UNMAPPED &
  OPTYPE alter &
  OBJTYPE 'table' &
  OBJNAME "users.tab*" &
INCLUDE MAPPED OBJNAME "*" &
EXCLUDE MAPPED OBJNAME "temporary.tab*"
```

この文では、次のようにフィルタ基準の組合せを指定します。

- INCLUDE は、TABLE または MAP 文にマッピングされていない (UNMAPPED スコープ) 表のすべての ALTER TABLE 文が含まれます。
 - ただし、表が "users" によって所有されておらず、その名前が "tab" で始まる場合のみです。
- さらに INCLUDE では、TABLE または MAP 文にマッピングされている (MAPPED スコープ) すべての表のすべての DDL 操作タイプが含まれます。
- さらに、EXCLUDE では、MAPPED スコープのすべての表のすべての DDL 操作タイプが除外されます。
 - ただし、これらの表が "temporary" によって所有される場合のみです。
 - また、その名前が "tab" で始まる場合のみです。

DDLERROR

適用対象 Extract および Replicat

ソースおよびターゲット・システム上の DDL エラーを処理するには、DDLERROR パラメータを使用します。Extract および Replicat 用の各種オプションが用意されています。

Extract 用 DDLERROR オプション

Extract によって認識される、メタデータが検出不可能なオブジェクトに関するエラーを処理するには、DDLERROR パラメータの Extract オプションを使用します。

デフォルト 異常終了

構文 DDLERROR [RESTARTSKIP <num skips>] [SKIPTRIGGERERROR <num errors>]

引数	説明
RESTARTSKIP <num skips>	Extract の起動時に、特定の数の DDL 操作をスキップおよび無視させ、エラーによる Extract の異常終了を防ぎます。デフォルトでは、DDL エラーが発生すると Extract は異常終了するため、操作はスキップされません。このパラメータで有効な値は、1 ～ 100000 です。 スキップした操作を Extract レポート・ファイルに書き込むには、DDLOPTIONS パラメータと REPORT オプションを使用します。
SKIPTRIGGERERROR <num errors>	(Oracle) Extract の起動時に、DDL トリガーが原因で発生する特定の数の DDL エラーをスキップおよび無視させます。<num errors> の有効な値は、1 ～ 100000 です。 SKIPTRIGGERERROR は、RESTARTSKIP オプションの前にチェックされます。Extract がトリガー・エラーのために DDL 操作をスキップする場合、その操作は RESTARTSKIP 指定の対象としてカウントされません。

Replicat 用 DDLERROR オプション

DDL がターゲット・データベースに適用される際に発生するエラーを処理するには、DDLERROR パラメータの Replicat オプションを使用します。DDLERROR オプションを使用して、処理の停止などのたいいていのエラーをデフォルトの方法で処理でき、また他のエラーも特定の方法で処理できます。同じパラメータ・ファイルで複数のインスタンスの DDLERROR を使用して、予期されるすべてのエラーを処理できます。

デフォルト 異常終了

構文 DDLERROR
{<error> | DEFAULT} {<response>}
[RETRYOP MAXRETRIES <n> [RETRYDELAY <delay>]]
{INCLUDE <inclusion clause> | EXCLUDE <exclusion clause>}
[IGNOREMISSINGTABLES | ABENDONMISSINGTABLES]

引数	説明
'<search_string>'	置換するソース DDL 文内の文字列。文字列を一重引用符で囲みます。文字列内で一重引用符を表す場合は、二重引用符を使用します。
WITH	必要なキーワード。
'<replace_string>'	ターゲット DDL 内で置換として使用する文字列。文字列を一重引用符で囲みます。文字列内で一重引用符を表す場合は、二重引用符を使用します。
INCLUDE <inclusion clause> EXCLUDE <exclusion clause>	文字列の置換ルールが適用される DDL 操作をフィルタ処理するには、1 つ以上の INCLUDE および EXCLUDE 文を使用します。次の表を参照してください。

表 39 DDL の包含および除外オプション

オプション	説明
INCLUDE EXCLUDE	<p>inclusion または exclusion 句の開始部分を識別するには、INCLUDE および EXCLUDE を使用します。</p> <ul style="list-style-type: none"> ◆ inclusion 句には、このパラメータによって影響される DDL を識別するフィルタ基準が含まれます。 ◆ exclusion 句には、このパラメータから特定の DDL を除外するフィルタ基準が含まれます。 <p>inclusion または exclusion 句は、INCLUDE または EXCLUDE キーワードに続き、適用されるパラメータの他のオプションの有効な組合せで構成される必要があります。</p> <p>EXCLUDE を使用する場合、対応する INCLUDE 句を作成する必要があります。たとえば、次は無効です。</p> <pre>DDL EXCLUDE OBJNAME "hr.*"</pre> <p>ただし、次のいずれかは使用できます。</p> <pre>DDL INCLUDE ALL, EXCLUDE OBJNAME "hr.*"</pre> <pre>DDL INCLUDE OBJNAME "fin.*" EXCLUDE "fin.ss"</pre> <p>EXCLUDE は、同じ基準が含まれているいずれの INCLUDE よりも優先されます。複数の inclusion および exclusion 句を使用できます。</p>

表 39 DDL の包含および除外オプション（続き）

オプション	説明
MAPPED UNMAPPED OTHER ALL	<p>DDL 操作スコープに基づいて INCLUDE または EXCLUDE を適用するには、MAPPED、UNMAPPED、OTHER および ALL を使用します。</p> <ul style="list-style-type: none"> ◆ MAPPED は、INCLUDE または EXCLUDE を MAPPED スコープの DDL 操作に適用します。MAPPED フィルタ処理は、他の DDL パラメータ・オプションで指定されたフィルタ処理の前に実行されます。 ◆ UNMAPPED は、INCLUDE または EXCLUDE を UNMAPPED スコープの DDL 操作に適用します。 ◆ OTHER は、INCLUDE または EXCLUDE を OTHER スコープの DDL 操作に適用します。 ◆ ALL は、INCLUDE または EXCLUDE をすべてのスコープの DDL 操作に適用します。
OPTYPE <type>	<p>INCLUDE または EXCLUDE を、CREATE、ALTER、RENAME など、特定のタイプの DDL 操作に適用するには、OPTYPE を使用します。<type> には、データベースに有効な任意の DDL コマンドを使用します。たとえば、ALTER 操作を含める場合の正しい構文は次のようになります。</p> <pre>DDL INCLUDE OPTYPE ALTER</pre>
OBJTYPE '<type>'	<p>INCLUDE または EXCLUDE を特定のデータベース・オブジェクトに適用するには OBJTYPE を使用します。<type> には、TABLE、INDEX、TRIGGER など、データベースに有効な任意のオブジェクト・タイプを使用します。Oracle マテリアライズド・ビューおよびマテリアライズド・ビュー・ログの場合、正しいタイプはそれぞれ snapshot と snapshot log です。オブジェクト・タイプの名前を二重引用符で囲みます。</p> <p>例：</p> <pre>DDL INCLUDE OBJTYPE 'INDEX'</pre> <pre>DDL INCLUDE OBJTYPE 'SNAPSHOT'</pre> <p>Oracle オブジェクト・タイプ USER の場合、USER のみがスキーマを持つのにに対し、OBJNAME で "owner.object" が予期されるので、OBJNAME オプションは使用しないでください。</p>
OBJNAME "<name>"	<p>INCLUDE または EXCLUDE を、owner.table_name など、オブジェクトの完全修飾名に適用するには、OBJNAME を使用します。このオプションでは、二重引用符で囲まれた文字列を入力として使用できます。</p> <p>ワイルドカードは、オブジェクト名にのみ使用できます。</p> <p>例：</p> <pre>DDL INCLUDE OBJNAME "accounts.*"</pre> <p>OracleUSER オブジェクトの場合、USER のみがスキーマを持つのにに対し、OBJNAME で "owner.object" が予期されるので、OBJNAME は使用しないでください。</p> <p>Replicat パラメータ・ファイルで OBJNAME を MAPPED とともに使用する場合、OBJNAME の値が、MAP 文の TARGET 句で指定されている名前を参照する必要があります。たとえば、次の MAP 文では、正しい値は OBJNAME "fin2.*" となります。</p> <pre>MAP fin.exp_*, TARGET fin2.*;</pre>

表 39 DDL の包含および除外オプション（続き）

オプション	説明
	<p>次の例では、CREATE TABLE 文によりソースで実行されるものは次のようになります。</p> <pre>CREATE TABLE fin.exp_phone;</pre> <p>ターゲットについては次のようになります。</p> <pre>CREATE TABLE fin2.exp_phone;</pre> <p>ターゲット・オーナーが MAP で指定されていない場合、Replicat により USERID パラメータで指定されているデータベース・ユーザーにマッピングされます。</p> <p>トリガー、シノニム、および索引を作成する DDL の場合、OBJNAME の値はトリガー、シノニム、または索引の名前ではなく、ベース・オブジェクトの名前である必要があります。</p> <p>たとえば、次の DDL 文を含めるための正しい値は、"hr.insert_trig" ではなく、"hr.accounts" になります。</p> <pre>CREATE TRIGGER hr.insert_trig ON hr.accounts;</pre> <p>RENAME 操作の場合、OBJNAME の値は新しい表の名前である必要があります。</p> <p>たとえば、次の DDL 文を含めるための正しい値は、"hr.acct" になります。</p> <pre>ALTER TABLE hr.accounts RENAME TO acct;</pre>
INSTR '<string>'	<p>INCLUDE または EXCLUDE を、コマンド構文（自体、ただしコメント内ではなく）内の特定の文字列を含んだ DDL 文に適用するには、INSTR を使用します。たとえば、次の場合は、索引を作成する DDL は除外されます。</p> <pre>DDL INCLUDE ALL EXCLUDE INSTR 'CREATE INDEX'</pre> <p>文字列を一重引用符で囲みます。文字列の検索では大 / 小文字は区別されません。</p>
INSTRCOMMENTS '<comment_string>'	<p>INCLUDE または EXCLUDE を、コマンド構文内自体ではなく、コメント内の特定の文字列を含んだ DDL 文に適用するには、INSTRCOMMENTS を使用します。INSTRCOMMENTS を使用することで、コメントをフィルタ処理用エージェントとして使用できます。</p> <p>たとえば、次の場合、コメントに "source" が含まれている DDL 文は除外されます。</p> <pre>DDL INCLUDE ALL EXCLUDE INSTRCOMMENTS 'SOURCE ONLY'</pre> <p>この例では、次のような DDL 文は複製されません。</p> <pre>CREATE USER john IDENTIFIED BY john /*source only*/;</pre> <p>文字列を一重引用符で囲みます。文字列の検索では大 / 小文字は区別されません。INSTR と INSTRCOMMENTS を組み合わせて、同じ DDL 文のコマンド構文内およびコメント内の文字列をフィルタ処理できます。</p>

表 39 DDL の包含および除外オプション（続き）

オプション	説明
INSTRWORDS '<word list>'	<p>INCLUDE または EXCLUDE を特定の単語を含んだ DDL 文に適用するには、INSTRWORDS を使用します。</p> <p><word list> には、複数の単語を任意の順序で指定し、これらを一重引用符で囲みます。空白を含める場合は、二重引用符内に空白（該当する場合は単語も）を入れます。二重引用符は、文を囲む場合にも使用できます。</p> <p>INSTRWORDS が有効になるには、指定した単語がすべて DDL に存在する必要があります。</p> <p>例：</p> <pre>ALTER TABLE INCLUDE INSTRWORDS 'ALTER CONSTRAINT " xyz"'</pre> <p>この例は以下と一致します。</p> <pre>ALTER TABLE ADD CONSTRAINT xyz CHECK</pre> <p>および</p> <pre>ALTER TABLE DROP CONSTRAINT xyz</pre>
INSTRCOMMENTSWORDS '<word list>'	<p>INSTRWORDS と同様に機能しますが、DDL 構文自体ではなく DDL 文内のコメントにのみ適用されます。INSTRCOMMENTS を使用することで、コメントをフィルタ処理用エージェントとして使用できます。</p> <p>INSTRWORD と INSTRCOMMENTSWORDS を組み合わせて、同じ DDL 文のコマンド構文内およびコメント内の文字列をフィルタ処理できます。</p>

例 1 DDLERROR の基本的な例

次の例では、DDLERROR 文により、Replicat で 10 秒間隔で 3 回操作が試行されたあと、特定のエラーが無視されます。Replicat により、"tab1*" を満たすもの以外の、"tab*" のワイルドカードを満たす名前のオブジェクト（任意のユーザー、操作）に実行された DDL 操作にエラー処理が適用されます。

```
DDLERROR <error> IGNORE RETRYOP MAXRETRIES 3 RETRYDELAY 10 &
INCLUDE ALL OBJTYPE TABLE OBJNAME "tab*" EXCLUDE OBJNAME "tab1*"
```

そのエラー以外のすべてのエラーを処理するには、次の DDLERROR 文を追加します。

```
DDLERROR DEFAULT ABENDS
```

この場合、DDL エラーにより Replicat が異常終了します。

例 2 複数の DDLERROR 文の使用

パラメータ・ファイルの DDLERROR 文にリストされている順序は、複数の DDLERROR 文で、追加の修飾子なしで同じエラーが指定されていないかぎり、有効性には影響しません。その場合は、Replicat では最初にリストされているもののみが使用されます。たとえば、次の文の場合、エラーにより Replicat が異常終了します。

```
DDLERROR <error1> ABEND
DDLERROR <error1> IGNORE
```


ただし、適切な修飾子が指定される場合は、前述の構成が有効になります。例：

```
DDLERROR <error1> ABEND INCLUDE OBJNAME "tab*"
DDLERROR <error1> IGNORE
```

この場合、INCLUDE 文があるので、誤った DDL 文内のオブジェクト名がワイルドカード "tab*" と一致する場合にのみ、Replicat は異常終了します。Replicat で、その他のオブジェクト名が含まれている誤った操作は無視されます。

DDLOPTIONS

適用対象 Extract および Replicat

フィルタ処理および文字列の置換以外の DDL 処理を構成するには、DDLOPTIONS パラメータを使用します。複数の DDLOPTIONS 文を使用できますが、1 つのみを使用することをお勧めします。複数の DDLOPTIONS 文を使用する場合、それぞれ上書きされないように、一意であることを確認してください。複数の DDLOPTIONS 文は、リストされている順序で実行されます。

Oracle GoldenGateのDDLサポートの範囲でDDLOPTIONSを使用する方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください、

デフォルト 引数の説明を参照してください。

構文

```
DDLOPTIONS
[, ADDTRANDATA [ABEND | RETRYOP <RETRYDELAY <seconds> MAXRETRIES <retries>]
[, DECRYPTPASSWORD <shared secret> ENCRYPTKEY {DEFAULT | <key name>}]
[, DEFAULTUSERPASSWORDPASSWORD <password>
  [ENCRYPTKEY DEFAULT | ENCRYPTKEY <keyname>]]
[, GETAPPLOPS | IGNOREAPPLOPS]
[, GETREPLICATES | IGNOREREPLICATES]
[, IGNOREMAPPING]
[, MAPDERIVED | NOMAPDERIVED]
[, MAPSCHEMAS]
[, MAPSESSIONSCHEMA] <source_schema> TARGET <target_schema>
[, NOCROSSRENAME]
[, PASSWORD ENCRYPTKEY [DEFAULT | ENCRYPTKEY <keyname>]
[, REMOVECOMMENTS {BEFORE | AFTER}]
[, REPLICATEPASSWORD | NOREPLICATEPASSWORD]
[, REPORT | NOREPORT]
[, UPDATEMETADATA]
[, USEOWNERFORSESSION]
```

引数	説明
ADDTRANDATA [ABEND RETRYOP <RETRYDELAY <seconds> MAXRETRIES <retries>]	<p>Extract (Oracle および Teradata) に有効です。</p> <p>ADDTRANDATA は次の目的で使用します。</p> <ul style="list-style-type: none"> ◆ CREATE TABLE で作成された新しい表にオラクルのサブリメンタル・ロギングを自動的に有効化する。 ◆ 列を追加または削除する ALTER TABLE の影響を受けた表のサブリメンタル・ロギングを更新する。 ◆ 名前が変更された表のサブリメンタル・ロギングを更新する。 ◆ 一意キーまたは主キーが追加または削除された表のサブリメンタル・ロギングを更新する。 <p>ABEND RETRYOP <RETRYDELAY <seconds> MAXRETRIES <retries></p> <p>表がロックされているために ADD TRANDATA コマンドが失敗した場合に、Extract が異常終了するか再試行するかを制御します。</p> <ul style="list-style-type: none"> ◆ ABEND を指定すると、Extract は異常終了します。 ◆ RETRYOP を指定すると、Extract は RETRYDELAY および MAXRETRIES に基づいてコマンドを再試行します。RETRYOP を使用するとき、次を両方とも指定する必要があります。 ◆ RETRYDELAY では、Extract の再試行までの待機時間を設定します。最大待機時間は 10,000 秒です。 ◆ MAXRETRIES では、Extract が異常終了するまでに実行する再試行回数を設定します。最大再試行回数は 10,000 です。 <p>デフォルトは次のとおりです。</p> <pre>DDOPTIONS ADDTRANDATA RETRYOP RETRYDELAY 10 MAXRETRIES 10</pre> <p>この機能を使用するには、『Oracle GoldenGate Windows and UNIX リファレンス・ガイド』の指示に基づいて、DDL 取得のために Oracle GoldenGate、データベース、および適切な表を構成する必要があります。Oracle の場合は、『Oracle GoldenGate Oracle インストールおよびセットアップ・ガイド』の指示に基づいて、Oracle GoldenGate DDL オブジェクトがインストールおよび構成されている必要があります。</p> <p>CREATE TABLE で作成された新しい表には、ADDTRANDATA は Oracle ALTER TABLE コマンドと ADD SUPPLEMENTAL LOG GROUP オプションを発行するため、GGSCI のデフォルトの ADD TRANDATA コマンドと同じ結果になります。Oracle GoldenGate は、ソースで CREATE TABLE または ALTER TABLE が取得されると、このコマンドを実行します。サブリメンタル・ロギングに関して特別な要件がある場合は、DDOPTIONS ADDTRANDATA ではなく、ADD TRANDATA を使用してください。</p> <p>デフォルトでは、サブリメンタル・ロギングを追加する ALTER TABLE は、GETREPLICATES オプションを使用していなければ、ターゲットにレプリケートされません。</p>

引数	説明
	<p>名前が変更された表には、ADDTRANDATA は古い名前の表のサプリメンタル・ロギングを削除し、新しい表のサプリメンタル・ロギングを作成します。ADDTRANDATA を使用していない場合に表の名前を変更するときは、次の手順を実行して、ログ・グループを作成してから名前を変更してください。</p> <ol style="list-style-type: none"> 1. データベース・インタフェースまたは GGSCI の DELETE TRANDATA <table> を使用して、サプリメンタル・ログ・グループを削除します。 2. 表の名前を変更します。 3. データベース・インタフェースまたは GGSCI の ADD TRANDATA <table> を使用して、新しいサプリメンタル・ログ・グループを作成します。 <p>新しい DDL 操作の実行から ADD TRANDATA の発効までの間に、ラグがあることがあります。影響を受ける表のデータをレプリケートする場合は、この間にこの表での DML 操作 (INSERT、UPDATE、DELETE) を許可しないでください。許可すると、データは取得されません。ADDTRANDATA の発効後にいつ DML を再開できるかを特定するには、次の操作を実行します。</p> <ol style="list-style-type: none"> 1. GGSCI で Extract パラメータ・ファイルを編集します。 EDIT PARAMS <group> 2. DDLOPTIONS に REPORT オプションを追加し、ファイルを保存して閉じます。 DDLOPTIONS [, other DDLOPTIONS options], REPORT 3. Extract を停止および起動してパラメータへの変更を有効化します。 STOP EXTRACT <group> START EXTRACT <group> 4. Extract プロセス・レポートを表示します。 VIEW REPORT <group name> 5. 表にログ・グループを追加した ALTER TABLE を探し、このコマンドが発効になった時刻をメモします。次のようなエントリです。 Successfully added TRAN DATA for table with the key, table [QATEST1.MYTABLE], operation [ALTER TABLE "QATEST1"."MYTABLE" ADD SUPPLEMENTAL LOG GROUP "GGS_MYTABLE_53475" (MYID) ALWAYS /* GOLDENGATE_DDL_REPLICATION */]. 6. 新しい表での DML 操作を許可します。

引数	説明
DEFAULTUSERPASSWORD <password> [ENCRYPTKEY DEFAULT ENCRYPTKEY <keyname>]	<p>Replicat に有効です (Oracle のみ)。</p> <p>レプリケートされる {CREATE ALTER} USER <name> IDENTIFIED BY <password> 文に、ソース文で使用されているパスワードとは異なるパスワードを指定します。Replicat は、Extract がトレイルに書き込むブレースホルダを、指定されたパスワードに置き換えます。</p> <ul style="list-style-type: none"> ◆ DEFAULTUSERPASSWORD <password> では、クリアテキストのパスワードを指定します。パスワードに大 / 小文字区別がある場合は、そのとおりに入力します。 ◆ DEFAULTUSERPASSWORD <encrypted password> ENCRYPTKEY DEFAULT では、Oracle GoldenGate によって生成されたランダム鍵を使用して暗号化されたパスワードを指定します。ターゲットでは、このパスワードは ENCKEYS ファイルを必要とせずに、自動的に復号化されます。 ◆ DDOPTIONS DEFAULTUSERPASSWORD <encrypted password> ENCRYPTKEY <keyname> では、ユーザー定義の方法で暗号化されたパスワードを指定し、ターゲット・システム上の ENCKEYS ファイル内の参照鍵を指定します。 <p>DEFAULTUSERPASSWORD を使用するときは、Extract 用 DDOPTIONS の NOREPLICATEPASSWORD オプションを使用します。</p> <p>暗号化オプションを使用するには、まず GGSCI の ENCRYPT PASSWORD コマンドで、暗号化されたパスワードを生成する必要があります。Oracle GoldenGate の暗号化オプションの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>
GETAPPLOPS IGNOREAPPLOPS	<p>Extract に有効です (Oracle のみ)。</p> <p>Replicat 以外のビジネス・アプリケーションが生成した DDL 操作を、Extract がトレイルまたはファイルに書き込むコンテンツに含めるかどうかを制御します。GETAPPLOPS および IGNOREAPPLOPS は、GETREPLICATES および IGNOREREPLICATES オプションとともに使用して、双方向またはカスケード構成で伝播する DDL を制御できます。</p> <ul style="list-style-type: none"> ◆ 双方向構成では、GETAPPLOPS と GETREPLICATES を使用します。UPDATEMETADATA オプションも使用する必要があります。 ◆ カスケード構成では、DDL 操作をターゲットにカスケードするシステム上で、IGNOREAPPLOPS と GETREPLICATES を使用します。 <p>デフォルトは GETAPPLOPS です。</p>
GETREPLICATES IGNOREREPLICATES	<p>Extract に有効です (Oracle のみ)。Replicat が生成した DDL 操作を Extract がトレイルまたはファイルに書き込むコンテンツに含めるかどうかを制御します。デフォルトは IGNOREREPLICATES です。詳細は、DDOPTIONS の GETAPPLOPS IGNOREAPPLOPS オプションを参照してください。</p>

引数	説明
MAPDERIVED NOMAPDERIVED	<p>Replicat (Oracle および Teradata) に有効です。導出オブジェクト名のマッピング方法を制御します。</p> <ul style="list-style-type: none"> ◆ MAPDERIVED: 導出オブジェクトの MAP 文が存在する場合、名前はその文の TARGET 句に指定されている名前にマッピングされます。それ以外の場合、名前はベース・オブジェクトを含む MAP 文の TARGET 句に指定されている名前にマッピングされます。MAPDERIVED はデフォルトです。 ◆ NOMAPDERIVED: 名前のマッピングを行わないようにします。NOMAPDERIVED は、導出オブジェクトの名前を含むすべての明示的な MAP 文より優先されます。 <p>DDL レプリケーション中の導出オブジェクト処理方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>
MAPSESSIONSCHEMA <source_schema> TARGET <target_schema>	<p>Replicat に有効です (Oracle のみ)。ソースのセッション・スキーマを、ターゲット上の異なるセッション・スキーマにマッピング (変換) します。</p> <ul style="list-style-type: none"> ◆ <source_schema> は、ソース上で ALTER SESSION set CURRENT_SCHEMA で設定されているセッション・スキーマです。 ◆ <target_schema> は、ターゲット上で ALTER SESSION set CURRENT_SCHEMA で設定されているセッション・スキーマです。 <p>ワイルドカードはサポートされていません。複数の MAPSESSIONSCHEMA パラメータを使用して、異なるスキーマをマップできます。</p> <p>MAPSESSIONSCHEMA は、マスターまたは導出オブジェクト名に基づくすべてのスキーマ名マッピングより優先されます。</p> <p>このセクション最後の使用例を参照してください。</p>
NOCROSSRENAME	<p>Extract (Oracle および Teradata) に有効です。Oracle GoldenGate 構成から除外されているオブジェクトを、構成に含まれる名前に変更することを許可しないルールを施行します。オブジェクトが構成に含まれる名前に変更されると、Extract が警告を生成するので、適切な措置を取ることができます。たとえば、名前が変更されたオブジェクトが Oracle GoldenGate にサポートされていない構造を持つ場合は、この通知を活用してエラーを防止できます。</p> <p>NOCROSSRENAME によって、除外されている表が含まれる名前に変更された場合に、この表の追跡のために必要なプロセスを排除できるので、Oracle RAC のパフォーマンスが向上します。</p> <p>NOCROSSRENAME は、パラメータ・ファイルのすべての TABLE および TABLEEXCLUDE 文、および除外されているものの TABLE または TABLEEXCLUDE に指定されていないすべてのオブジェクトにグローバルに適用されます。</p> <p>NOCROSSRENAME 機能は、TABLEEXCLUDE パラメータの NORENAME オプション (353 ページを参照してください) に似ていますが、NORENAME では範囲をより限定できます。</p>

引数	説明
PASSWORD ENCRYPTKEY [DEFAULT ENCRYPTKEY <keyname>]	<p>Extract に有効です (Oracle のみ)。</p> <p>DDL をトレイルに書き込む前に、Extract にソース DDL のすべてのパスワードを暗号化させます。</p> <ul style="list-style-type: none"> ◆ DEFAULT では、Oracle GoldenGate によって生成されたランダム鍵を使用してパスワードが暗号化されていることを指定します。ターゲットでは、このパスワードは ENCKEYS ファイルを必要とせずに、自動的に復号化されます。 ◆ ENCRYPTKEY <keyname> では、ユーザー定義の方法で暗号化パスワードが暗号化されていることを指定し、ターゲット・システム上の ENCKEYS ファイル内の参照鍵を指定します。 <p>暗号化オプションを使用するには、まず GGSCI の ENCRYPT PASSWORD コマンドで、暗号化されたパスワードを生成する必要があります。Oracle GoldenGate の暗号化オプションの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>
REMOVECOMMENTS {BEFORE AFTER}	<p>Extract および Replicat に有効です (Oracle のみ)。DDL 操作からコメントを削除するかどうかを制御します。デフォルトでは、DDLSUBST パラメータを使用して文字列の置換でできるように、コメントは削除されません (166 ページを参照してください)。</p> <ul style="list-style-type: none"> ◆ REMOVECOMMENTS BEFORE では、Extract または Replicat によって DDL 操作が処理される前にコメントを削除します。コメントは、文字列置換で使用できません。 ◆ REMOVECOMMENTS AFTER では、文字列置換で使用された後にコメントを削除します。
REPLICATEPASSWORD NOREPLICATEPASSWORD	<p>Extract に有効です (Oracle のみ)。{CREATE ALTER} USER <user> IDENTIFIED BY <password> コマンドのパスワードに適用します。</p> <ul style="list-style-type: none"> ◆ デフォルト (REPLICATEPASSWORD) では、Oracle GoldenGate はターゲットの CREATE または ALTER 文でソース・パスワードを使用します。 ◆ ソース・パスワードのターゲットへの送信を防ぐには、NOREPLICATEPASSWORD を使用します。 <p>NOREPLICATEPASSWORD を使用する場合は、Replicat パラメータ・ファイルの DDLOPTIONS 文と DEFAULTUSERPASSWORD オプションを使用して、ターゲット DDL 文用のパスワードを指定します。</p>
REPORT NOREPORT	<p>Extract および Replicat (Oracle および Teradata) に有効です。レポート・ファイルに詳しい DDL 処理情報を書き込むかどうかを制御します。デフォルトの NOREPORT では、基本的な DDL 統計をレポートします。REPORT では、使用しているパラメータ、および処理された操作のステップバイステップの履歴をレポートに追加します。</p>

引数	説明
UPDATEMETADATA	Replicat に有効です (Oracle のみ)。アクティブアクティブの双方向構成で使用します。このパラメータは、DDL が開始されたシステム上の Replicat に、もう一方のシステムにこの DDL が伝播されたため、新しいメタデータに合わせてすぐにオブジェクト・メタデータ・キャッシュを更新するように通知します。これにより、Replicat のメタデータ・キャッシュと、ローカル・データベースの現在のメタデータの同期性を維持します。
USEOWNERFORSESSION	Replicat に有効です (Oracle のみ)。Replicat DDL 文の未修飾のオブジェクトのスキーマを、ALTER SESSION SET CURRENT_SCHEMA 文のスキーマ (デフォルト動作) ではなく、Replicat のセッション・スキーマのスキーマにします。

例 1 次に、DEFAULTUSERPASSWORD を使用して、レプリケートされる {CREATE | ALTER} USER <name> IDENTIFIED BY <password> 文にソース文と異なるパスワードを割り当てる様々な方法を示します。

```
DDLOPTIONS DEFAULTUSERPASSWORD ocean
```

```
DDLOPTIONS DEFAULTUSERPASSWORD AACAAAAAAAAAAAAADESGTFTATAOEIKB ENCRYPTKEY  
superkey1
```

```
DDLOPTIONS DEFAULTUSERPASSWORD AACAAAAAAAAAAAAADALHSFYDIEWDEIEIHC ENCRYPTKEY  
DEFAULT
```

例 2 次に、MAPSESSIONSCHEMA を使用して、ソース・セッション・スキーマをターゲット上の別のスキーマにマッピングする方法を示します。

Extract および Replicat で、次のように DDL 取得およびマッピングが構成されているとします。

Extract

```
DDL INCLUDE OBJNAME "SRC.*" &  
INCLUDE OBJNAME "SRC1.*"  
TABLE SRC.*;  
TABLE SRC1.*;
```

Replicat

```
DDLOPTIONS MAPSESSIONSCHEMA SRC TARGET DST  
DDLOPTIONS MAPSESSIONSCHEMA SRC1 TARGET DST1  
MAP SRC.*, TARGET DST.*;  
MAP SRC1.*, TARGET DST1.*;  
DDL INCLUDE OBJNAME "DST.*" &  
INCLUDE "OBJNAME "DST1.*"
```

ソースのログイン・ユーザー OTH によって、次の DDL 文が発行されるとします。

```
ALTER SESSION SET CURRENT_SCHEMA=SRC;  
CREATE TABLE tab (X NUMBER);  
CREATE TABLE SRC1.tab (X NUMBER) AS SELECT * FROM tab;
```

Replicat は、次のような DDL を実行します (各コード・セグメントの前に説明を記載しています)。

```
-- Set session to DST, because SRC.* is mapped to DST.* in MAP statement.  
ALTER SESSION SET CURRENT_SCHEMA=DST;  
-- Create the first TAB table in the DST schema, using the DST session schema.  
CREATE TABLE DST.tab (X NUMBER);  
-- Restore Replicat schema.  
ALTER SESSION SET CURRENT_SCHEMA=REPUSER  
-- Set session schema to DST, per MAPSESSIONSCHEMA, so that AS SELECT succeeds.
```

```
ALTER SESSION SET CURRENT_SCHEMA=DST;
-- Create the DST1.TAB table AS SELECT * FROM the first table (DST.TAB).
CREATE TABLE DST1.tab (X NUMBER) AS SELECT * FROM tab;
-- Restore Replicat schema.
ALTER SESSION SET CURRENT_SCHEMA=REPUSER
```

MAPSESSIONSCHEMA を指定しない場合、SELECT * FROM TAB は存在しない SRC.TAB 表からの選択を試み、失敗します。デフォルトでは、ソース・スキーマを、ターゲット DDL 文の未修飾のオブジェクトに適用します。その場合、DDL 文は次のようになり、失敗します。

```
-- Set session to DST, because SRC.* is mapped to DST.* in MAP statement.
ALTER SESSION SET CURRENT_SCHEMA=DST;
-- Create the first TAB table in the DST schema, using the DST session schema.
CREATE TABLE DST.tab (X NUMBER);
-- Restore Replicat schema.
ALTER SESSION SET CURRENT_SCHEMA=REPUSER
-- Set session schema to SRC, because TAB in the AS SELECT is unqualified and
SRC is the source session schema.
ALTER SESSION SET CURRENT_SCHEMA=SRC;
-- Create DST1.TAB AS SELECT * from SRC.TAB (SRC=current session schema).
CREATE TABLE DST1.tab (X NUMBER) AS SELECT * FROM tab;
-- SRC.TAB does not exist.
-- Abend with an error unless the error is handled by a DDLERROR statement.
```

DDLSUBST

適用対象 Extract および Replicat

DDLSUBST パラメータでは、DDL 操作内の文字列を置換します。たとえば、ある表の名前を別の名前に置換したり、コメント内の文字列を置換したりできます。検索では、大 / 小文字は区別されません。文字列の中で引用符を表すには、二重引用符を使用します。

DDLSUBST の使用に関するガイドライン

- DDLSUBST を使用して、列の名前およびデータ・タイプをターゲット上の異なる名前およびデータ・タイプに変換しないでください。この方法でターゲット・オブジェクトの構造を変更すると、データが複製される際にエラーが発生します。同様に、DDLSUBST を使用して、ターゲットの DDL 文のオーナーおよび表の名前を変更しないでください。複製された DDL 操作を異なるターゲット・オブジェクトにマッピングする場合は、常に MAP 文を使用してください。
- DDLSUBST は、パラメータ・ファイル内の相対順序にかかわらず、常に DDL パラメータの後に実行されます。フィルタ処理が最初に行われるので、文字列の置換に使用する基準と互換性のあるフィルタ基準を使用します。たとえば、次のパラメータ文を考慮します。

```
DDL INCLUDE OBJNAME "fin.*"
DDLSUBST 'cust' WITH 'customers' INCLUDE OBJNAME "sales.*"
```

この例では、INCLUDE および DDLSUBST 文のオブジェクトは異なるので、置換は行われません。fin により所有されるオブジェクトは、Oracle GoldenGateDDL 構成に含まれますが、sales により所有されるオブジェクトは含まれません。

- 複数の DDLSUBST パラメータを使用できます。これらは、パラメータ・ファイルにリストされている順序で実行されます。

- コメントが含まれている Oracle DDL について、これらのコメントに対して文字列の置換を行う場合、DDLOPTIONS パラメータを REMOVECOMMENTS BEFORE オプションとともに使用しないでください。REMOVECOMMENTS BEFORE により、文字列の置換が発生する前にコメントが削除されます。コメントを削除するが、文字列の置換は可能にする場合は REMOVECOMMENTS AFTER オプションを使用します。
- 置換には、データベースの制限以外、最大文字列サイズの制限はありません。文字列サイズがデータベースの制限を超える場合、操作を実行中の Extract または Replicat 処理は異常終了します。

デフォルト 置換なし

構文 DDLSUBST '<search_string>' WITH '<replace_string>'
[INCLUDE <inclusion clause> | EXCLUDE <exclusion clause>]

引数	説明
'<search_string>'	置換するソース DDL 文内の文字列。文字列を一重引用符で囲みます。文字列内で一重引用符を表す場合は、二重引用符を使用します。
WITH	必要なキーワード。
'<replace_string>'	ターゲット DDL 内で置換として使用する文字列。文字列を一重引用符で囲みます。文字列内で一重引用符を表す場合は、二重引用符を使用します。
INCLUDE <inclusion clause> EXCLUDE <exclusion clause>	文字列の置換ルールが適用される DDL 操作をフィルタ処理するには、1 つ以上の INCLUDE および EXCLUDE 文を使用します。次の表を参照してください。

表 40 DDL の包含および除外オプション

オプション	説明
INCLUDE EXCLUDE	<p>inclusion または exclusion 句の開始部分を識別するには、INCLUDE および EXCLUDE を使用します。</p> <ul style="list-style-type: none"> ◆ inclusion 句には、このパラメータによって影響される DDL を識別するフィルタ基準が含まれます。 ◆ exclusion 句には、このパラメータから特定の DDL を除外するフィルタ基準が含まれます。 <p>inclusion または exclusion 句は、INCLUDE または EXCLUDE キーワードに続き、適用されるパラメータの他のオプションの有効な組合せで構成される必要があります。</p> <p>EXCLUDE を使用する場合、対応する INCLUDE 句を作成する必要があります。たとえば、次は無効です。</p> <pre>DDL EXCLUDE OBJNAME "hr.*"</pre> <p>ただし、次のいずれかは使用できます。</p> <pre>DDL INCLUDE ALL, EXCLUDE OBJNAME "hr.*"</pre> <pre>DDL INCLUDE OBJNAME "fin.*" EXCLUDE "fin.ss"</pre> <p>EXCLUDE は、同じ基準が含まれているいずれの INCLUDE よりも優先されます。複数の inclusion および exclusion 句を使用できます。</p>
MAPPED UNMAPPED OTHER ALL	<p>DDL 操作スコープに基づいて INCLUDE または EXCLUDE を適用するには、MAPPED、UNMAPPED、OTHER および ALL を使用します。</p> <ul style="list-style-type: none"> ◆ MAPPED は、INCLUDE または EXCLUDE を MAPPED スコープの DDL 操作に適用します。MAPPED フィルタ処理は、他の DDL パラメータ・オプションで指定されたフィルタ処理の前に実行されます。 ◆ UNMAPPED は、INCLUDE または EXCLUDE を UNMAPPED スコープの DDL 操作に適用します。 ◆ OTHER は、INCLUDE または EXCLUDE を OTHER スコープの DDL 操作に適用します。 ◆ ALL は、INCLUDE または EXCLUDE をすべてのスコープの DDL 操作に適用します。
OPTYPE <type>	<p>INCLUDE または EXCLUDE を、CREATE、ALTER、RENAME など、特定のタイプの DDL 操作に適用するには、OPTYPE を使用します。<type> には、データベースに有効な任意の DDL コマンドを使用します。たとえば、ALTER 操作を含める場合の正しい構文は次のようになります。</p> <pre>DDL INCLUDE OPTYPE ALTER</pre>

表 40 DDL の包含および除外オプション（続き）

オプション	説明
OBJTYPE ' <code><type></code> '	<p>INCLUDE または EXCLUDE を特定のデータベース・オブジェクトに適用するには OBJTYPE を使用します。<type> には、TABLE、INDEX、TRIGGER など、データベースに有効な任意のオブジェクト・タイプを使用します。Oracle マテリアライズド・ビューおよびマテリアライズド・ビュー・ログの場合、正しいタイプはそれぞれ snapshot と snapshot log です。オブジェクト・タイプの名前を二重引用符で囲みます。</p> <p>例：</p> <pre>DDL INCLUDE OBJTYPE 'INDEX'</pre> <pre>DDL INCLUDE OBJTYPE 'SNAPSHOT'</pre> <p>Oracle オブジェクト・タイプ USER の場合、USER のみがスキーマを持つのに対し、OBJNAME で "owner.object" が予期されるので、OBJNAME オプションは使用しないでください。</p>
OBJNAME " <code><name></code> "	<p>INCLUDE または EXCLUDE を、owner.table_name など、オブジェクトの完全修飾名に適用するには、OBJNAME を使用します。このオプションでは、二重引用符で囲まれた文字列を入力として使用できます。</p> <p>ワイルドカードは、オブジェクト名にのみ使用できます。</p> <p>例：</p> <pre>DDL INCLUDE OBJNAME "accounts.*"</pre> <p>OracleUSER オブジェクトの場合、USER のみがスキーマを持つのに対し、OBJNAME で "owner.object" が予期されるので、OBJNAME は使用しないでください。</p> <p>Replicat パラメータ・ファイルで OBJNAME を MAPPED とともに使用する場合、OBJNAME の値が、MAP 文の TARGET 句で指定されている名前を参照する必要があります。たとえば、次の MAP 文では、正しい値は OBJNAME "fin2.*" となります。</p> <pre>MAP fin.exp_*, TARGET fin2.*;</pre> <p>次の例では、CREATE TABLE 文によりソースで実行されるものは次のようになります。</p> <pre>CREATE TABLE fin.exp_phone;</pre> <p>ターゲットについては次のようになります。</p> <pre>CREATE TABLE fin2.exp_phone;</pre> <p>ターゲット・オーナーが MAP で指定されていない場合、Replicat により USERID パラメータで指定されているデータベース・ユーザーにマッピングされます。</p>

表 40 DDL の包含および除外オプション（続き）

オプション	説明
	<p>トリガー、シノニム、および索引を作成する DDL の場合、OBJNAME の値はトリガー、シノニム、または索引の名前ではなく、ベース・オブジェクトの名前である必要があります。</p> <p>たとえば、次の DDL 文を含めるための正しい値は、"hr.insert_trig" ではなく、"hr.accounts" になります。</p> <pre>CREATE TRIGGER hr.insert_trig ON hr.accounts;</pre> <p>RENAME 操作の場合、OBJNAME の値は新しい表の名前である必要があります。</p> <p>たとえば、次の DDL 文を含めるための正しい値は、"hr.acct" になります。</p> <pre>ALTER TABLE hr.accounts RENAME TO acct;</pre>
INSTR '<string>'	<p>INCLUDE または EXCLUDE を、コマンド構文（自体、ただしコメント内ではなく）内の特定の文字列を含んだ DDL 文に適用するには、INSTR を使用します。たとえば、次の場合は、索引を作成する DDL は除外されます。</p> <pre>DDL INCLUDE ALL EXCLUDE INSTR 'CREATE INDEX'</pre> <p>文字列を一重引用符で囲みます。文字列の検索では大 / 小文字は区別されません。</p>
INSTRCOMMENTS '<comment_string>'	<p>INCLUDE または EXCLUDE を、コマンド構文内自体ではなく、コメント内の特定の文字列を含んだ DDL 文に適用するには、INSTRCOMMENTS を使用します。INSTRCOMMENTS を使用することで、コメントをフィルタ処理用エージェントとして使用できます。</p> <p>たとえば、次の場合、コメントに "source" が含まれている DDL 文は除外されます。</p> <pre>DDL INCLUDE ALL EXCLUDE INSTRCOMMENTS 'SOURCE ONLY'</pre> <p>この例では、次のような DDL 文は複製されません。</p> <pre>CREATE USER john IDENTIFIED BY john /*source only*/;</pre> <p>文字列を一重引用符で囲みます。文字列の検索では大 / 小文字は区別されません。INSTR と INSTRCOMMENTS を組み合わせて、同じ DDL 文のコマンド構文内およびコメント内の文字列をフィルタ処理できます。</p>

表 40 DDL の包含および除外オプション（続き）

オプション	説明
INSTRWORDS '<word list>'	<p>INCLUDE または EXCLUDE を特定の単語を含んだ DDL 文に適用するには、INSTRWORDS を使用します。</p> <p><word list> には、複数の単語を任意の順序で指定し、これらを一重引用符で囲みます。空白を含める場合は、二重引用符内に空白（該当する場合は単語も）を入れます。二重引用符は、文を囲む場合にも使用できます。</p> <p>INSTRWORDS が有効になるには、指定した単語がすべて DDL に存在する必要があります。</p> <p>例：</p> <pre>ALTER TABLE INCLUDE INSTRWORDS 'ALTER CONSTRAINT " xyz"'</pre> <p>この例は以下と一致します。</p> <pre>ALTER TABLE ADD CONSTRAINT xyz CHECK</pre> <p>および</p> <pre>ALTER TABLE DROP CONSTRAINT xyz</pre>
INSTRCOMMENTSWORDS '<word list>'	<p>INSTRWORDS と同様に機能しますが、DDL 構文自体ではなく DDL 文内のコメントにのみ適用されます。INSTRCOMMENTS を使用することで、コメントをフィルタ処理用エージェントとして使用できます。</p> <p>INSTRWORD と INSTRCOMMENTSWORDS を組み合わせて、同じ DDL 文のコマンド構文内およびコメント内の文字列をフィルタ処理できます。</p>

- 例 1** 次の例では、"fin" スキーマ内の表の文字列 'cust' を文字列 'customers' に置換します。
- ```
DDLSUBST 'cust' WITH 'customers'
INCLUDE ALL OBJTYPE 'table' OBJNAME "fin.*"
```
- 例** 次の例では、DDL コマンドに単語 "logfile" が含まれている場合にのみ新しいディレクトリに置換します。検索文字列が複数回検出された場合、置換文字列は複数回挿入されます。
- ```
DDLSUBST '/file1/location1' WITH '/file2/location2' INCLUDE INSTR'logfile'
```
- 例 2** 次の例では、複数の DDLSUBST 文を使用します。文はリストされている順番に実行されます。
- ```
DDLSUBST 'a' WITH 'b' INCLUDE ALL
DDLSUBST 'b' WITH 'c' INCLUDE ALL
```
- 上記の最終結果では、すべての "a" および "b" 文字列が "c" に置換されます。

## DDLTABLE

### 適用対象

GLOBALS

DDLTABLE パラメータでは、Oracle DDL 同期をサポートする DDL 履歴表の名前をデフォルトの GGS\_DDL\_HIST 以外にする場合に、その名前を指定します。DDL 履歴表には、Oracle GoldenGate に処理された DDL 操作の履歴が保持されます。

履歴表の名前は、params.sql スクリプトの ddl\_hist\_table パラメータにも指定する必要があります。このスクリプトは、ルート Oracle GoldenGate インストール・ディレクトリにあります。

このパラメータは、Oracle にのみ有効です。DDL 履歴表および `params.sql` の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** GGS\_DDL\_HIST

**構文** DDLTABLE <table\_name>

| 引数           | 説明          |
|--------------|-------------|
| <table_name> | DDL 履歴表の名前。 |

**例** DDLTABLE GG\_DDL\_HISTORY

## DECRYPTTRAIL

**適用対象** Extract および Replicat

DECRYPTTRAIL パラメータでは、トレイルまたは抽出ファイルのデータを復号化します。DECRYPTTRAIL は、トレイルを暗号化するためにアップストリーム Extract プロセスに対して ENCRYPTTRAIL パラメータを使用している場合にのみ使用してください。

**デフォルト** なし

**構文** DECRYPTTRAIL

## DEFERAPPLYINTERVAL

**適用対象** Replicat

DEFERAPPLYINTERVAL パラメータでは、Replicat が取得したトランザクションをターゲット・データベースに適用するまでの待機時間を設定します。Replicat は、トランザクションをいつ適用するかを決定するために、ソース・システムのローカル GMT 時間で記録されているソース・ソーストランザクションのコミット・タイムスタンプにこの遅延値を追加します。

DEFERAPPLYINTERVAL は、ソース・データに対して行われた誤った変更の伝播の防止や、異なるタイムゾーン間でのデータ到着の制御、またターゲットにデータを適用する前に他の計画イベントを実行する時間を確保するなどの目的に使用できます。DEFERAPPLYINTERVAL を使用すると、ターゲット・データに意図的に遅延を発生させることになるため、ターゲット・アプリケーションで時間が重視される場合には、慎重に使用する必要があります。

Replicat が操作を延期しているかどうかを判断するには、SEND REPLICAT コマンドと STATUS オプションを使用して、Waiting on deferred apply のステータスを確認します。

**注意** TCPSOURCETIMER を使用している場合は、ソース・トランザクションとターゲット・トランザクションのタイムスタンプが数秒異なり、Replicat がトランザクションを数秒間オープンにしておく（したがって行ロックしている）可能性があります。この小さな差異によって、パフォーマンスに大きな影響がおよぶことはありません。

**デフォルト** 0（遅延なし）

**構文** DEFERAPPLYINTERVAL <n><unit>

| 引数     | 説明                                                                                                                    |
|--------|-----------------------------------------------------------------------------------------------------------------------|
| <n>    | 遅延時間を示す数値。最小遅延時間は、EOFDELAY パラメータで設定されている値です。最大日数は 7 日間です。                                                             |
| <unit> | 遅延時間の単位。次の単位を使用できます。<br>S   SEC   SECS   SECOND   SECONDS   MIN   MINS   MINUTE   MINUTES   HOUR   HOURS   DAY   DAYS |

**例** この例では、Replicat にトランザクションの適用まで 10 時間待機させます。

```
DEFERAPPLYINTERVAL 10
```

トランザクションがソースの GTM 時間 08:00:00 に完了している場合、このトランザクションは同日のターゲットの GMT 時間 18:00:00 に適用されます。

## DEFSFILE

**適用対象** DEFGEN

DEFSFILE パラメータでは、DEFGEN がデータ定義を書き込むファイルの名前を指定します。

**デフォルト** なし

**構文** DEFSFILE <filename> [APPEND | PURGE]

| 引数         | 説明                                                           |
|------------|--------------------------------------------------------------|
| <filename> | 完全修飾ファイル名。このファイルは DEFGEN を実行するときに作成されます。                     |
| APPEND     | 指定したファイルがすでに存在している場合に、DEFGEN に既存のコンテンツの末尾から新しいコンテンツを書き込ませます。 |
| PURGE      | 現在の実行で新しいコンテンツを書き込む前に、DEFGEN に指定したファイルをパージさせます。これはデフォルトです。   |

**例** DEFSFILE /home/ggs/dirdef/orcldef

**構文**

## DISCARDFILE

**適用対象** Extract および Replicat

DISCARDFILE パラメータでは、Oracle GoldenGate が処理できないレコードを記録できる破棄ファイルを生成します。レコードは、複数の理由で破棄されることがあります。たとえば、レコードがトレイルに書き込まれた後に、基盤の表構造が変更されると、レコードは破棄されます。破棄ファイルは、処理エラーの原因の特定にも使用できます。

破棄ファイルの各エントリには、破棄レコード・バッファと、理由を示すエラーコードが含まれます。Oracle GoldenGate は、指定された破棄ファイルを Oracle GoldenGate インストール・ディレクトリの

dirrpt サブディレクトリに作成します。破棄ファイルは、テキスト・エディタか、GGSCI で次のコマンドを使用して表示できます。

```
VIEW REPORT <file name>
```

**条件:** <file name> は、破棄ファイルの完全修飾名です。

破棄ファイルの手動でのメンテナンスを不要にするには、PURGE または APPEND オプションを使用します。Oracle GoldenGate は既存の破棄ファイルに書き込みを行わないため、これらのオプションを使用しない場合は、各プロセスの実行前に異なる破棄ファイルを指定する必要があります。

ファイル・サイズの上限を設定するには、MAXBYTES または MEGABYTES オプションを使用します。指定したサイズを超えると、プロセスは異常終了します。

**デフォルト** デフォルトでは、Oracle GoldenGate は破棄ファイルを生成しません。

**構文**

```
DISCARDFILE <file name>
[, APPEND | PURGE]
[, MAXBYTES <n> | MEGABYTES <n>]
```

| 引数            | 説明                                                                                                                                                            |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <file name>   | 破棄ファイルの完全修飾名。このファイルが Oracle GoldenGate ディレクトリにある場合は、完全名は不要です。Oracle GoldenGate は、Oracle GoldenGate インストール・ディレクトリでこの名前を修飾します。可能な場合は常に、プロセス・グループと同じ名前を使用してください。 |
| APPEND        | ファイルがすでに存在する場合に、既存のコンテンツに新しいコンテンツを追加します。                                                                                                                      |
| PURGE         | 新しいコンテンツを書き込む前にファイルをページします。                                                                                                                                   |
| MAXBYTES <n>  | ファイルの最大サイズを設定します（バイト）。有効な値は 1 ～ 2147483646 です。デフォルトは 1000000 です。                                                                                              |
| MEGABYTES <n> | ファイルの最大サイズを設定します（MB）。有効な値は 1 ～ 2147483646 です。デフォルトは 1MB です。                                                                                                   |

**例** DISCARDFILE discard.txt, PURGE, MEGABYTES 2

## DISCARDROLLOVER

**適用対象** Extract および Replicat

DISCARDROLLOVER パラメータを使用して、破棄ファイルのエージング・スケジュールを設定できます。長期または連続的な実行の場合、エージング・スケジュールを設定することにより、破棄ファイルがすべて使用され、プロセスが異常終了されるのを防ぐことができ、アーカイブ・ルーチンに含める予期可能なアーカイブ・セットを得ることができます。

DISCARDROLLOVER のエージング・ポイントに到達すると、新しい破棄ファイルが作成され、古いファイルは次に説明する <group name><n>.<extension> フォーマットの名前に変更されます。

- <group name> は、Extract または Replicat グループ名です。



- <n> は、新しいファイル作成のたびに 1 ずつ増分される数です。たとえば、myext0.dsc、myext1.dsc、myext2.dsc などのようになります。

時刻、曜日、またはその両方を指定できます。時刻（AT オプション）のみで曜日（ON オプション）を指定しない場合、指定した時刻に毎日破棄ファイルが生成されます。

**デフォルト** 無効。ルールは指定されていません。

**構文**

```
DISCARDROLLOVER
{AT <hh:mi> |
ON <day of week> |
AT <hh:mi> ON <day of week>}
```

| 引数               | 説明                                                                                                                                                                    |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AT <hh:mi>       | <p>ファイルをエージングする時刻（24 時間表記）。</p> <p>有効な値：</p> <ul style="list-style-type: none"> <li>◆ &lt;hh&gt; は、1 ～ 23 までの時間です。</li> <li>◆ &lt;mi&gt; は 00 ～ 59 までの分です。</li> </ul> |
| ON <day of week> | <p>ファイルをエージングする曜日。</p> <p>有効な値：</p> <p>SUNDAY<br/>MONDAY<br/>TUESDAY<br/>WEDNESDAY<br/>THURSDAY<br/>FRIDAY<br/>SATURDAY</p> <p>大 / 小文字は区別されません。</p>                   |

**例 1** DISCARDROLLOVER AT 05:30  
**例 2** DISCARDROLLOVER ON friday  
**例 3** DISCARDROLLOVER AT 05:30 ON friday

## DOWNCRITICAL

**適用対象** Manager

DOWNCRITICAL パラメータでは、DOWNREPORT パラメータによって生成されるレポートに、異常終了または正常終了したプロセスを含めます。

**デフォルト** なし

**構文** DOWNCRITICAL

## DOWNREPORT

**適用対象** Manager

DOWNREPORTMINUTES または DOWNREPORTEHOURS パラメータでは、Manager が実行中でない Extract および Replicat プロセスをレポートする間隔を指定します。エラー・ログには、プロセスが開始または終了するたびにイベントが生成されますが、ログが大きい場合には、こうしたメッセージは見逃されてしまいがちです。DOWNREPORTMINUTES および DOWNREPORTEHOURS は、停止したプロセスが見逃されないように、定期的にレポートします。

実行中のプロセスをレポートするには、UPREPORT パラメータを使用します。

**デフォルト** 停止したプロセスをレポートしない。

**構文** DOWNREPORTMINUTES <minutes> | DOWNREPORTEHOURS <hours>

| 引数        | 説明                       |
|-----------|--------------------------|
| <minutes> | 実行中でないプロセスをレポートする間隔（分）。  |
| <hours>   | 実行中でないプロセスをレポートする間隔（時間）。 |

**例** 次の例では、30 分間隔でレポートが生成されます。

DOWNREPORTMINUTES 30

## DSOPTIONS

**適用対象** Extract

DSOPTIONS パラメータでは、Teradata Access Module (TAM) を使用する Extract の抽出処理オプションを指定します。Oracle GoldenGate の Teradata 抽出の構成の詳細は、『Oracle GoldenGate Teradata インストレーションおよびセットアップ・ガイド』を参照してください。

**デフォルト** なし

**構文** DSOPTIONS  
[COMMITTEDTRANLOG]  
[CREATETRANLOG]  
[SORTTRANLOG]  
[RESTARTAPPEND]

| 引数               | 説明                                                                                                                                                                                                                 |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMMITTEDTRANLOG | (最大パフォーマンス・モード) トランザクション・データをディスクに永続化しないように指定します。TAM が Extract プロセスにトランザクション・データの変更を送信すると、Extract プロセスはトランザクションのコミット順に標準の Oracle GoldenGate トレイルにこれらのデータを保存します。トレイルは、Replicat またはデータ・ポンプ Extract グループが読み取ることができます。 |

| 引数            | 説明                                                                                                                                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CREATETRANLOG | (最大保護モード) Extract に VAM トレイル (ローカル・トレイル) を作成させます。トランザクション・データの変更は、VAM ソート Extract による次の処理のために、VAM トレイルに永続化されます。データは、様々なトランザクションからの変更レコードをインタリーブするログスタイル・フォーマットで VAM トレイルに書き込まれます。このオプションは、Teradata Access Module (TAM) とインタフェースを取り、VAM トレイルへの書き込みを行うプライマリ Extract プロセスに対して使用します。VAM トレイルは、GGSCI の ADD EXTTRAIL コマンドで指定します。 |
| SORTTRANLOG   | (最大保護モード) Extract にトランザクションのソート・ファンクションを実行させます。このオプションは、プライマリ Extract プロセスによって移入された VAM トレイルを読み取る VAM ソート Extract グループに対して使用します。VAM ソート Extract は、Oracle GoldenGate による次の処理の前に、インタリーブされた操作を正しい準備 / コミット / ロールバック・トランザクション単位にソートします。                                                                                        |
| RESTARTAPPEND | (最大パフォーマンス・モード) Extract の再起動時に、前回の実行で書き込んだデータを書き換えるかわりに、Oracle GoldenGate トレイルの末尾にデータを追加させます。このオプションは、COMMITTEDTRANLOG 引数とともに使用します。                                                                                                                                                                                        |

## DYNAMICPORTLIST

|       |                                                                                                                                                              |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 適用対象  | Manager                                                                                                                                                      |
|       | DYNAMICPORTLIST パラメータでは、ソース・システムとターゲット・システム (Collector プロセス、Replicat プロセス、GGSCI プロセスなど) の動的な通信のために Manager が動的に割当て可能なポートを指定します。個別のポート、ポートの範囲、またはその両方を指定できます。 |
|       | Manager のポート再割当て試行までの待機時間を制御するには、DYNAMICPORTREASSIGNDelay パラメータを使用します。                                                                                       |
| デフォルト | なし                                                                                                                                                           |
| 構文    | DYNAMICPORTLIST {<port>   <port>-<port>} [ , ...]                                                                                                            |

| 引数     | 説明                                                                                                                                                                                                                                                                                                                                    |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <port> | 割当て可能なポート番号。ポート・エントリの最大数は 256 です。 <ul style="list-style-type: none"> <li>複数のポートを指定するには、コンマ区切りリストを使用します。次に例を示します。<br/>7830, 7833</li> <li>ポート範囲を指定するには、ダッシュ (-) を使用して範囲の最初のポートと最後のポートを区切ります。ダッシュの前後に空白を入力しないでください。次に例を示します。<br/>7830-7835</li> <li>ポート範囲および個別ポートを指定するには、範囲と個別ポート番号の間にコンマを置きます。次に例を示します。<br/>7830-7835, 7839</li> </ul> |

例 DYNAMICPORTLIST 7820-7830, 7833, 7835

## DYNAMICPORTREASSIGNDELAY

**適用対象** Manager

DYNAMICPORTREASSIGNDELAY パラメータでは、Manager がすでに割り当てたポートの再割当てを試行するまでの待機時間を指定します。このパラメータは、DYNAMICPORTLIST パラメータとともに使用します。

**デフォルト** 3 秒

**構文** DYNAMICPORTREASSIGNDELAY <seconds>

| 引数        | 説明      |
|-----------|---------|
| <seconds> | 待機する秒数。 |

**例** DYNAMICPORTREASSIGNDELAY 5

## DYNAMICRESOLUTION | NODYNAMICRESOLUTION

**適用対象** Extract および Replicat

DYNAMICRESOLUTION および NODYNAMICRESOLUTION パラメータでは、表名の解決方法を制御します。

DYNAMICRESOLUTION を使用すると、TABLE または MAP 文で多数の表が指定されているときに、より早く処理を開始できます。デフォルトでは、Oracle GoldenGate はプロセスが起動するたびにデータベースにすべての表の属性を問合せ、すべての表のオブジェクト・レコードを構築します。このレコードはメモリーおよびディスク上に保持されます。データベースが大規模な場合、このレコードの構築には多くの時間がかかることがあります。

DYNAMICRESOLUTION を指定すると、一度にすべての表でなく、一度に 1 つの表のオブジェクト・レコードが構築されます。特定の表の属性は、そのオブジェクト ID がトランザクション・ログに最初に記録される時（この表の最初の抽出トランザクションと同時）に追加されます。他の表のレコード構築は、アクティビティが発生するまで延期されます。DYNAMICRESOLUTION は、WILDCARDRESOLVE DYNAMIC と同様に動作します。

NODYNAMICRESOLUTION を指定すると、プロセスの起動時にオブジェクト・レコードが構築されます。このオプションは、Teradata ではサポートされていません。NODYNAMICRESOLUTION は、WILDCARDRESOLVE IMMEDIATE と同様に動作します。

WILDCARDRESOLVE の詳細は、388 ページを参照してください。

**デフォルト** DYNAMICRESOLUTION

**構文** DYNAMICRESOLUTION

## DYNSQL | NODYNSQL

**適用対象** Replicat

DYNSQL および NODYNSQL パラメータでは、SQL 文の生成方法を制御します。NODYNSQL を指定すると、Replicat はリテラル SQL 文と解決されたバインド変数を使用します。デフォルトの DYNSQL を使用すると、Replicat は動的 SQL を使用して 1 つの文を一度だけ生成した後、異なるバインド変数を使用してこの文を何回も実行します。

- NODYNSQL を指定しない文：  
UPDATE <table> ... WHERE ID = :B
- NODYNSQL を指定した文：  
UPDATE <table> ... WHERE ID = '1234'

DYNSQL を使用すると、ほとんどの環境で最高の効率化と最大限のスループットを達成できます。ただし、分離されたインスタンスでは、NODYNSQL を使用した場合に、さらに高速化および効率化できることが実証されています。NODYNSQL は、Replicat のスループットが十分でない場合にのみ試行してください。

動的 SQL をサポートしていないターゲット・データベースにレプリケートする場合は、DYNSQL を使用しないでください。

NODYNSQL を使用する場合は、NOBINARYCHARS パラメータも使用する必要があります。これらのいずれかのパラメータを使用する前に、Oracle サポートに連絡してください。詳細は、<http://support.oracle.com> を参照してください。

Oracle GoldenGate for MySQL は、NODYNSQL モードでの LOB レプリケーションをサポートしていません。

|       |                   |
|-------|-------------------|
| デフォルト | DYNSQL            |
| 構文    | DYNSQL   NODYNSQL |

## ENCRYPTTRAIL | NOENCRYPTTRAIL

適用対象 Extract

ENCRYPTTRAIL および NOENCRYPTTRAIL パラメータでは、Oracle GoldenGate がトレイル・ファイルに書き込むデータを暗号化するかどうかを制御します。ENCRYPTTRAIL を使用すると、すべてのデータ・リンク、およびファイル内両方のすべてのレコードが暗号化されます。NOENCRYPTTRAIL を使用すると、暗号化は行われません。

暗号化は、Extract パラメータ・ファイル内の次のパラメータで指定しているトレイルまたは抽出ファイルに対して使用できます。

RMTRAIL  
EXTTRAIL  
RMFILE  
EXTFILE

ENCRYPTTRAIL および NOENCRYPTTRAIL は、トレイルまたはファイルに固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、パラメータ・ファイル内でそれ以降のすべてのトレイルまたは抽出ファイルに影響します。このパラメータは、適用するトレイルのパラメータ・エントリの前に配置する必要があります。

データを復号化するには、暗号化されたファイルを読み取るすべての Replicat プロセスのパラメータ・ファイルで DECRYPTTRAIL パラメータを使用します (172 ページを参照してください)。また DECRYPTTRAIL は、列マッピング、フィルタリング、変換などのデータを復号化するために、Extract データ・ポンプでも使用できます。その後は、ダウンストリーム・トレイルまたはファイルのために復号化したままにしておくことも、ENCRYPTTRAIL を使用してこれらのファイルに書き込む前にもう一度暗号化することもできます。

ENCRYPTTRAIL および NOENCRYPTTRAIL は、ASCII フォーマットでファイルにデータを書き込むために FORMATASCII を使用しているときは使用できません。トレイルまたはファイルは、デフォルトの Oracle

GoldenGate 正規フォーマットで書き込まれる必要があります。

ENCRYPTTRAIL は、データ・ブロックのみを暗号化します。ユーザー・トークンは暗号化されません。

**デフォルト** NOENCRYPTTRAIL (暗号化なし)

**構文** ENCRYPTTRAIL | NOENCRYPTTRAIL

**例** 次の例では、emp 表のデータが暗号化され、stores 表のデータは暗号化されません。

```
ENCRYPTTRAIL
RMTTRAIL /home/ggsora/dirdat/em
TABLE hr.emp;
NOENCRYPTTRAIL
RMTTRAIL /home/ggsora/dirdat/st
TABLE ops.stores;
```

## END

**適用対象** Extract および Replicat

バッチ実行用の END パラメータでは、タイムスタンプが指定した時点と一致する、データ・ソース内の最初のレコードを検出したときに、プロセスを終了します。そのタイムスタンプまでのレコードがバッチで処理されます。

END を使用しない場合、プロセスは、次の時点まで継続的に実行されます。

- トランザクション・ログまたはトレイルの末尾に到達したとき。この時点で、プロセスは正常に終了します。
- コマンド・シェルから手動で終了したとき。

END は、ターゲット表を継続的に更新するのではなく、ポイントインタイム・スナップショットとしてデータを送信するために、バッチ・ジョブまたはオンライン変更同期構成の一部として SPECIALRUN パラメータとともに使用します。

バッチ実行の構成方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** 継続処理

**構文** END {<date> [<time>] | RUNTIME}

| 引数              | 説明                                                                                                                                                                                                                                                   |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <date> [<time>] | タイムスタンプがこのパラメータの指定を超えるデータ・ソース内のレコードに到達したときに、Extract または Replicat を終了させます。<br>有効な値:<br><ul style="list-style-type: none"> <li>◆ &lt;date&gt; は、yyyy-mm-dd フォーマットの日付です。</li> <li>◆ &lt;time&gt; は、24 時間表記の hh:mi[:ss[.cccccc]] フォーマットの時刻です。</li> </ul> |
| RUNTIME         | タイムスタンプが現在の日時を超えるデータ・ソース内のレコードに到達したときに、Extract または Replicat を終了させます。この時点までのタイムスタンプを持つ未処理レコードは、すべて処理されます。RUNTIME を使用するメリットの 1 つは、実行のたびにパラメータ・ファイルで日時を変更する必要がない点です。かわりに、バッチ・プログラミング内でプロセスの開始時刻を制御できます。                                                 |

**例 1**           SPECIALRUN  
                  END 2009-01-12 17:00:00

**例 2**           SPECIALRUN  
                  END RUNTIME

## EOFDELAY | EOFDELAYCSECS

**適用対象**     Extract および Replicat

データ・ソース内で現在のデータが終わりに達した後で、Extract、データ・ポンプまたは Replicat が新しいデータを確認する回数を制御するには、EOFDELAY または EOFDELAYCSECS パラメータを使用します。このパラメータの値を増やすことにより、これらの読取りのシステム I/O オーバーヘッドを減らすことができます。

**注意**        この値を大幅に増やすと、特にソース・データベース上のアクティビティが少ない場合には、ターゲット・データの遅延が増加する可能性があります。

このパラメータは、SOURCEISTABLE を使用している場合は無効です。

**デフォルト**   1 秒

**構文**         EOFDELAY <seconds> | EOFDELAYCSECS <centiseconds>

| 引数             | 説明                    |
|----------------|-----------------------|
| <seconds>      | 処理するデータを探すまでの遅延（秒）。   |
| <centiseconds> | 処理するデータを探すまでの遅延（ミリ秒）。 |

**例**            EOFDELAY 3

## ETOLDFORMAT

**適用対象**     Extract

ETOLDFORMAT パラメータでは、Oracle GoldenGate リリース 6.0 以前の Replicat バージョンと互換性のあるフォーマットでトレイルを生成します。

Oracle GoldenGate リリース 10 以上では、ETOLDFORMAT は互換性レベルが 0 であることを意味します。ETOLDFORMAT は、すべての出力ファイルにグローバルに適用されます。ETOLDFORMAT の他に、EXTFILE、EXTTRAIL、RMFILE または RMTTRAIL で互換性レベルが指定されている場合、Extract プロセスは異常終了します。詳細は、これらのパラメータの説明を参照してください。

**デフォルト**   なし

**構文**         ETOLDFORMAT

## EXTFILE

**適用対象** Extract および Replicat

EXTFILE パラメータでは、ローカル・システム上の抽出ファイルを指定します。このパラメータの実装は、プロセスによって若干異なります。

- Extract の場合、このパラメータでは、データ・ポンプ Extract グループまたはローカル・システム上の Replicat グループによって読み取られるローカル・ファイルを指定します。
- Replicat の場合、このパラメータでは、SPECIALRUN を使用してバッチ実行を生成するときに、ローカル抽出ファイルを指定します。

Solaris システムでは、Extract または Replicat によって処理される場合、抽出ファイルのサイズの上限は 2GB です。初期ロードで使用するネイティブ・ロード・ユーティリティなどの別のアプリケーションによって処理される場合は、抽出ファイルのサイズをさらに大きくできます。

EXTFILE は、関連するすべての TABLE または MAP 文よりも先に指定する必要があります。異なるファイルを定義するために、複数の EXTFILE 文を使用できます。

### ファイル・バージョンングについて

すべての Oracle GoldenGate プロセスは分離されるので、異なる Oracle GoldenGate リリースのものである可能性があることから、各証跡ファイルまたは抽出ファイルでは、バージョンがファイル・ヘッダー内に格納されています。デフォルトでは、証跡のバージョンは、ファイルを作成したプロセスの現在のバージョンになります。証跡のバージョンを設定するには、EXTTRAIL、EXTTRAIL、RMTTRAIL または RMTFILE パラメータの FORMAT オプションを使用します。

異なる Oracle GoldenGate プロセス・バージョンの上位および下位互換性が保証されるようにするには、ファイルのヘッダー・フィールドは標準のトークン形式で記述される必要があります。プロセスの新しいバージョンで作成される新しいトークンは、古いバージョンでは無視されるようにできるので、下位互換性が保証されます。同様に新しい Oracle GoldenGate バージョンでは古いトークンがサポートされます。さらに、トークンが新しいプロセス・バージョンで推奨されない場合、デフォルト値がトークンに割り当てられるので、古いバージョンは引き続き適切に機能します。ファイル・バージョンを指定するトークンは COMPATIBILITY で、Logdump ユーティリティで表示可能です。また、@GETENV 関数の GGFILHEADER オプションで取得することもできます。

証跡または抽出ファイルには、それらが読み込まれるプロセスのバージョン以下のバージョンが必要です。それ以外の場合はプロセスが異常終了します。さらに、Oracle GoldenGate では、データ・ポンプの出力証跡またはファイルのバージョンが、入力証跡またはファイルのバージョンと同じであることが強制されます。再起動の際、Extract では新しいファイルが証跡にロールされるので、各ファイルが 1 つのバージョンのみ（ファイルが空の場合を除く）であることが保証されます。

### パラメータの依存関係

RECOVERYOPTIONS パラメータと EXTTRAIL、RMTTRAIL、EXTFILE および RMTFILE の FORMAT オプション間には依存性が存在します。RECOVERYOPTIONS が APPENDMODE に設定される場合、FORMAT オプションは RELEASE 10.0 以上に設定される必要があります。RECOVERYOPTIONS が OVERWRITEMODE に設定される場合、FORMAT オプションは RELEASE 9.5 以下に設定される必要があります。

**デフォルト** なし



**構文**

```
EXTFILE <file name>
[, MAXFILES <number>]
[, MEGABYTES <megabytes>]
[, FORMAT RELEASE <major>.<minor>]
```

| 引数                    | 説明                                                                                                                                                                                                                                                                       |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <file name>           | Extract および Replicat に有効です。抽出ファイルの完全修飾名を指定します。                                                                                                                                                                                                                           |
| MAXFILES <number>     | Extract に有効です。単一のファイルでなく、連続するファイルを作成させます。1 つのファイルのサイズがオペレーティング・システムに許可される制限を越える可能性があるときに使用します。MAXFILES を使用して、必要な数のファイルを作成できます。エンディングされたファイルには、6 桁の順序番号が付けられます（例：datafile000002）。チェックポイントは、これらのファイルには保持されません。MAXFILES を使用するときは、MEGABYTES も使用して、連続ファイル内の各ファイルの最大サイズを設定します。 |
| MEGABYTES <megabytes> | Extract に有効です。ファイル（MAXFILES を使用している場合は各ファイル）の最大サイズを定義します。                                                                                                                                                                                                                |

| 引数                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FORMAT<br>RELEASE <major>.<minor> | <p>Extract により証跡、ファイルまたは（リモート・タスクの場合）別のプロセスに送信されるデータのメタデータ形式を指定します。形式は Oracle GoldenGate プロセスのリリースに依存します。古い Oracle GoldenGate リリースには新しいものよりも含まれるメタデータの数が少ない、または異なります。メタデータによって、データ・レコードがサポートされるリリースのものであるかどうかプロセスに伝達されます。</p> <ul style="list-style-type: none"> <li>◆ FORMAT は、必須のキーワードです。</li> <li>◆ RELEASE では Oracle GoldenGate リリースが指定されます。&lt;major&gt; はメジャー・リリース番号で、&lt;minor&gt; はマイナー・リリース番号です。有効な値は 9.0 から現在の Oracle GoldenGate リリース番号の間です。（9.0 より前の Oracle GoldenGate リリースを使用している場合は、9.0 か 9.5 のいずれかを指定します。）プログラムとして、リリースは適切な証跡形式の互換性レベルにマッピングされます。デフォルト値は、この証跡に書き込まれるプロセスの現在のリリースです。</li> </ul> <p>追加モードが Extract で有効でない場合、デフォルトのファイルのバージョン番号は自動的に 9（Oracle GoldenGate リリース 9.x）になります。Extract モードは、RECOVERYOPTIONS パラメータで制御されます。</p> <p>証跡または抽出ファイルには、それらが読み込まれるプロセスのバージョン以下のバージョンが必要です。それ以外の場合はプロセスが異常終了します。さらに、Oracle GoldenGate では、データ・ポンプの出力証跡またはファイルのバージョンが、入力証跡またはファイルのバージョンと同じであることが強制されます。再起動の際、Extract では新しいファイルが証跡にロールされるので、各ファイルが 1 つのバージョンのみ（ファイルが空の場合を除く）であることが保証されます。</p> |

- 例 1      EXTFILE /ggs/dirdat/datafile
- 例 2      EXTFILE /ggs/dirdat/extdat, MAXFILES 3, MEGABYTES 5
- 例 3      EXTFILE /ggs/dirdat/extdat, FORMAT RELEASE 10.4

## EXTRACT

**適用対象**      Extract

EXTRACT パラメータでは、オンライン変更同期用の Extract グループを指定します。このパラメータによって、現在の実行と前回の実行が関連付けられるので、データ変更を継続的に処理し、ソース表とターゲット表の同期性を維持できます。Extract は継続的に実行し、データ・ソースおよびトレイルにチェックポイントを保持することにより、計画的または計画外のプロセス終了、システム停止、ネットワーク障害が発生した場合にもデータ整合性とフォルト・トレランスを確保します。EXTRACT は、パラメータ・ファイルの最初のエントリにする必要があります。

変更同期の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト**      なし

**構文**           EXTRACT <group name>

| 引数           | 説明                          |
|--------------|-----------------------------|
| <group name> | ADD EXTRACT コマンドで定義したグループ名。 |

**例**           次の例では、"finance" という名前の Extract グループを指定します。

EXTRACT finance

## EXTTRAIL

**適用対象**   Extract および Replicat

EXTTRAIL パラメータでは、ローカル・システム上のトレイルを指定します。このパラメータの実装は、プロセスによって若干異なります。

- Extract の場合は、このパラメータを使用して、ADD EXTTRAIL コマンドで作成されたローカル・トレイルを指定します。通常このトレイルは、データ・ポンプ Extract グループ、またはローカル・システム上の Replicat グループによって読み取られます。
- Replicat の場合、このパラメータは、バッチ実行を生成する SPECIALRUN パラメータを指定しているときにのみ使用します。連続変更同期の場合、EXTTRAILは、GGCSIのADD REPLICAT コマンドでReplicat に対して指定します。

EXTTRAIL は、関連するすべての TABLE 文よりも先に指定する必要があります。異なるトレイルを定義するために、複数の EXTTRAIL 文を使用できます。

パッシブ・モードで構成されている Extract には、EXTTRAIL を使用しないでください。詳細は、17 ページの「ADD EXTRACT」を参照してください。

## ファイル・バージョンングについて

すべての Oracle GoldenGate プロセスは分離されるので、異なる Oracle GoldenGate リリースのものである可能性があることから、各証跡ファイルまたは抽出ファイルでは、バージョンがファイル・ヘッダー内に格納されています。デフォルトでは、証跡のバージョンは、ファイルを作成したプロセスの現在のバージョンになります。証跡のバージョンを設定するには、EXTTRAIL、EXTTRAIL、RMTTRAIL または RMTFILE パラメータの FORMAT オプションを使用します。

異なる Oracle GoldenGate プロセス・バージョンの上位および下位互換性が保証されるようにするには、ファイルのヘッダー・フィールドは標準のトークン形式で記述される必要があります。プロセスの新しいバージョンで作成される新しいトークンは、古いバージョンでは無視されるようにできるので、下位互換性が保証されます。同様に新しい Oracle GoldenGate バージョンでは古いトークンがサポートされます。さらに、トークンが新しいプロセス・バージョンで推奨されない場合、デフォルト値がトークンに割り当てられるので、古いバージョンは引き続き適切に機能します。ファイル・バージョンを指定するトークンは COMPATIBILITY で、Logdump ユーティリティで表示可能です。また、@GETENV 関数の GGFILEHEADER オプションで取得することもできます。

証跡または抽出ファイルには、それらが読み込まれるプロセスのバージョン以下のバージョンが必要です。それ以外の場合はプロセスが異常終了します。さらに、Oracle GoldenGate では、データ・ポンプの出力証跡またはファイルのバージョンが、入力証跡またはファイルのバージョンと同じであることが強制されます。再起動の際、Extract では新しいファイルが証跡にロールされるので、各ファイルが 1 つのバージョンのみ（ファイルが空の場合を除く）であることが保証されます。

## パラメータの依存関係

RECOVERYOPTIONS パラメータと EXTTRAIL、RMTTRAIL、EXTFILE および RMTFILE の FORMAT オプション間には依存性が存在します。RECOVERYOPTIONS が APPENDMODE に設定される場合、FORMAT オプションは RELEASE 10.0 以上に設定される必要があります。RECOVERYOPTIONS が OVERWRITEMODE に設定される場合、FORMAT オプションは RELEASE 9.5 以下に設定される必要があります。

**デフォルト** なし

**構文** EXTTRAIL <file name>  
[, FORMAT RELEASE <major>.<minor>]

| 引数                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <file name>                       | トレイルの完全修飾名。最大文字数（2 文字）を名前に使用します。エージングされたトレイル・ファイルは、この名前に 6 桁の順序番号が追加されます（例： /ggs/dirdat/rt000001）。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| FORMAT<br>RELEASE <major>.<minor> | <p>Extract により証跡、ファイルまたは（リモート・タスクの場合）別のプロセスに送信されるデータのメタデータ形式を指定します。形式は Oracle GoldenGate プロセスのリリースに依存します。古い Oracle GoldenGate リリースには新しいものよりも含まれるメタデータの数が少ない、または異なります。メタデータによって、データ・レコードがサポートされるリリースのものであるかどうかプロセスに伝達されます。</p> <ul style="list-style-type: none"> <li>◆ FORMAT は、必須のキーワードです。</li> <li>◆ RELEASE では Oracle GoldenGate リリースが指定されます。&lt;major&gt; はメジャー・リリース番号で、&lt;minor&gt; はマイナー・リリース番号です。有効な値は 9.0 から現在の Oracle GoldenGate リリース番号の間です。（9.0 より前の Oracle GoldenGate リリースを使用している場合は、9.0 か 9.5 のいずれかを指定します。）プログラムとして、リリースは適切な証跡形式の互換性レベルにマッピングされます。デフォルト値は、この証跡に書き込まれるプロセスの現在のリリースです。</li> </ul> <p>追加モードが Extract で有効でない場合、デフォルトのファイルのバージョン番号は自動的に 9（Oracle GoldenGate リリース 9.x）になります。Extract モードは、RECOVERYOPTIONS パラメータで制御されます。</p> <p>証跡または抽出ファイルには、それらが読み込まれるプロセスのバージョン以下のバージョンが必要です。それ以外の場合はプロセスが異常終了します。さらに、Oracle GoldenGate では、データ・ポンプの出力証跡またはファイルのバージョンが、入力証跡またはファイルのバージョンと同じであることが強制されます。再起動の際、Extract では新しいファイルが証跡にロールされるので、各ファイルが 1 つのバージョンのみ（ファイルが空の場合を除く）であることが保証されます。</p> |

**例 1** EXTTRAIL /ggs/dirdat/ny

**例 2** EXTTRAIL /ggs/dirdat/ex, FORMAT RELEASE 10.4

# FETCHOPTIONS

**適用対象** Extract

FETCHOPTIONS パラメータでは、Oracle GoldenGate が次の状況でデータをフェッチする方法を制御します。

- Extractが更新操作を再構築するために十分な情報がトランザクション・レコードに含まれていない
- TABLE 文の FETCHCOLS 句の結果として、Oracle GoldenGate が列値をフェッチする必要がある

FETCHOPTIONS は表に固有です。1 つの FETCHOPTIONS 文は、別の FETCHOPTIONS 文が見つかるまで、後続のすべての TABLE 文に適用されます。

ほとんどのインストールでは、デフォルトのフェッチ・プロパティで十分に機能します。

**デフォルト** 行方不明の行を無視し、処理を継続する

**構文**

```

FETCHOPTIONS
[, MISSINGROW <action>]
[, NOFETCH]
[, USEKEY | NOUSEKEY]
[, USELATESTVERSION | NOUSELATESTVERSION]
[, USESNAPSHOT | NOUSESNAPSHOT]
[, USEROWID | NOUSEROWID]

```

| 引数                  | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                             |        |                                                                                    |         |                                                               |       |                                                           |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----------------------------|--------|------------------------------------------------------------------------------------|---------|---------------------------------------------------------------|-------|-----------------------------------------------------------|
| MISSINGROW <action> | <p>Oracle GoldenGate がフェッチする行を特定できない場合のレスポンスを指定し、行（変更された値）の一部のみでも処理できるようにします。通常、行が特定できない理由は、変更レコードが作成されてからフェッチがトリガーされるまでの間に行が削除されてしまっているか、必要な行イメージが指定されている UNDO 保存期間以前のものであったためです。</p> <p>&lt;action&gt; には、次のいずれかを指定できます。</p> <table> <tr> <td>IGNORE</td><td>状況を無視して処理を継続します。これはデフォルトです。</td></tr> <tr> <td>REPORT</td><td>状況と行のコンテンツを破棄ファイルにレポートしますが、部分的な行の処理を継続します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</td></tr> <tr> <td>DISCARD</td><td>データを破棄し、部分的な行は処理しません。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</td></tr> <tr> <td>ABEND</td><td>データを破棄し、処理を中止します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</td></tr> </table> | IGNORE | 状況を無視して処理を継続します。これはデフォルトです。 | REPORT | 状況と行のコンテンツを破棄ファイルにレポートしますが、部分的な行の処理を継続します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。 | DISCARD | データを破棄し、部分的な行は処理しません。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。 | ABEND | データを破棄し、処理を中止します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。 |
| IGNORE              | 状況を無視して処理を継続します。これはデフォルトです。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |        |                             |        |                                                                                    |         |                                                               |       |                                                           |
| REPORT              | 状況と行のコンテンツを破棄ファイルにレポートしますが、部分的な行の処理を継続します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                             |        |                                                                                    |         |                                                               |       |                                                           |
| DISCARD             | データを破棄し、部分的な行は処理しません。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |                             |        |                                                                                    |         |                                                               |       |                                                           |
| ABEND               | データを破棄し、処理を中止します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |        |                             |        |                                                                                    |         |                                                               |       |                                                           |
| NOFETCH             | <p>Extract によるデータベースからの列のフェッチを防ぎます。Extract はトレイルにレコードを書き込みますが、列が行方不明であることを示すトークンを挿入します。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |        |                             |        |                                                                                    |         |                                                               |       |                                                           |

| 引数                                       | 説明                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| USEKEY  <br>NOUSEKEY                     | <p>Oracle に有効です。フェッチする行を見つけるために Oracle GoldenGate が主キーを使用するかどうかを決定します。</p> <p>USEKEY および USEROWID 両方を指定する場合、レコードにより高速にアクセスできる ROWID の方が優先されます。デフォルトは USEROWID です。</p>                                                                                                                                                                  |
| USELATESTVERSION  <br>NOUSELATESTVERSION | <p>Oracle 9i 以上に有効です。USESAPNSHOT とともに使用します。デフォルトの USELATESTVERSION では、UNDO 表領域からフェッチできない場合に、Extract にソース表からデータをフェッチさせます。NOUSELATESTVERSION は、スナップショット・フェッチが失敗した場合に、Extract にその状況を見無視して処理を継続させます。</p> <p>スナップショット・フェッチが失敗した場合の代替アクションを指定するには、MISSINGROW オプションを使用します。</p>                                                                |
| USESAPNSHOT   NOUSESAPNSHOT              | <p>Oracle 9i 以上に有効です。デフォルトの USESAPNSHOT では、Extract に、Flashback Query メカニズムを使用して、LOB データ (Oracle 9i のみ)、ユーザー定義データ型、ネストされた表、および XMLType レコードを含む操作の再構築に必要なデータを UNDO 表領域からフェッチさせます。NOUSESAPNSHOT では、Extract に、必要なデータをソース表からフェッチさせます。Oracle GoldenGate の Oracle からのデータ・フェッチ方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p> |
| USEROWID  <br>NOUSEROWID                 | <p>Oracle に有効です。フェッチする行を見つけるために Oracle GoldenGate が行 ID を使用するかどうかを決定します。</p> <p>USEKEY および USEROWID 両方を指定する場合、レコードにより高速にアクセスできる ROWID の方が優先されます。デフォルトは USEROWID です。</p>                                                                                                                                                                |

- 例 1** 次の例では、Extract に Flashback Query を使用してデータをフェッチさせ、フェッチが失敗した場合にはその状況を見無視してレコードの処理を継続させます。
- ```
FETCHOPTIONS USESAPNSHOT, NOUSELATESTVERSION
```
- 例 2** 次の例では、Extract に Flashback Query を使用してデータをフェッチさせ、データを使用できない場合には Extract を異常終了させます。
- ```
FETCHOPTIONS USESAPNSHOT, NOUSELATESTVERSION, MISSINGROW ABEND
```

## FILTERDUPS | NOFILTERDUPS

**適用対象** Replicat

FILTERDUPS および NOFILTERDUPS パラメータでは、アプリケーションが同一トランザクション内で同一のレコードに複数の操作を実行するときに NonStop システムで発生する可能性がある異常を処理します。このタイプのトランザクションによって、TMF オーディット・トレイルのレコードの順番が不正になり、Replicat が異常終了することがあります。次に例を示します。

- ソース・アプリケーションが削除を実行してから挿入を実行した場合でも、オーディット・トレイルでは、同一の主キーの削除の前に挿入が発生することがあります (結果として、Replicat が挿入を実行するときに重複レコード・エラーが発生します)。

- オーディット・トレイルでは、同一の主キーの挿入の前に更新が発生することがあります（結果として、Replicat が更新を実行するときに行方不明レコード・エラーが発生します）。

FILTERDUPS は、Replicat の異常終了を防止するために、次のようにしてこの状況を解決します。

- 挿入するレコードが重複する場合、Replicat はこの挿入操作をトランザクションの最後に保存します。その後、同一主キーの削除を検出すると、Replicat は削除を実行し、その後に挿入を実行します。
- 更新するレコードが行方不明の場合、Replicat はこの更新操作をトランザクションの最後に保存します。その後、同一の主キーの挿入を検出すると、Replicat は挿入を実行し、その後に更新を実行します。

このような異常は、一般に IDX 医療アプリケーションや一部の BASE24 バンク・アプリケーションなどで見られます（ただし、これらのアプリケーションに限定されません）。FILTERDUPS は、重複するレコードや行方不明のレコードのために Replicat が異常終了し、その原因が NonStop システムで開始された不正な順序のトランザクションであることが判明している場合にのみ使用してください。この状況の診断には、Logdump ユーティリティを使用できます。『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

FILTERDUPS および NOFILTERDUPS は、必要に応じて異なる MAP 文グループの例外処理を有効化または無効化する個別スイッチとして使用できます。

**デフォルト** NOFILTERDUPS

**構文** FILTERDUPS | NOFILTERDUPS

**例** この例は、FILTERDUPS を Orders に対して有効にしますが、同一のパラメータ・ファイルのそれ以降に定義されているすべての MAP 文に対しては無効にします。

```
FILTERDUPS
MAP $DATA1.SQDAT.ORDERS, TARGET MASTER.ORDERS;
NOFILTERDUPS
```

## FLUSHSECS | FLUSHCSECS

**適用対象** Extract

FLUSHSECS および FLUSHCSECS パラメータでは、Oracle GoldenGate がいつ Extract メモリー・バッファをフラッシュするかを制御します。リモート・システムにデータを送信する際、Extract はデータをバッファしてネットワーク・パフォーマンスを最適化します。バッファは、一杯になるか、FLUSHSECS または FLUSHCSECS で指定された時間が経過した後にフラッシュされます。バッファがフラッシュされてデータが送信されるまで、ターゲットのユーザーは変更データを使用できません。バッファのサイズを制御するには、RMTHOST の TCPBUFSIZE オプションを使用します（287 ページを参照してください）。

FLUSHSECS または FLUSHCSECS の値を増やすと、ネットワーク使用効率が若干高まる場合がありますが、ソース・システムのアクティビティが少なく、バッファが一杯にならない場合には、ターゲット・データの遅延が増える可能性があります。ソース表のアクティビティが多い場合は、FLUSHSECS および FLUSHCSECS はほとんど効果がありません。

**デフォルト** 1 秒

構文 FLUSHSECS <seconds> | FLUSHCSECS <centiseconds>

| 引数             | 説明                       |
|----------------|--------------------------|
| <seconds>      | バッファをフラッシュするまでの遅延（秒）。    |
| <centiseconds> | バッファをフラッシュするまでの遅延（センチ秒）。 |

例 FLUSHSECS 80

## FORMATASCII

適用対象 Extract

FORMATASCII パラメータでは、デフォルトの Oracle GoldenGate 正規フォーマットではなく、外部 ASCII フォーマットでデータを出力します。FORMATASCII を使用すると、大半のデータベース・ロード・ユーティリティおよびASCII入力を必要とするその他のプログラムと互換性を持つように出力をフォーマットできます。このパラメータは、*File-to-Database-Utility* による初期ロード方法を使用するときが必要です。

FORMATASCII 文は、パラメータ・ファイル内のこの文以降のすべての抽出ファイルまたはトレイルに影響します。パラメータ・ファイル内の文の相対的な順序は重要です。ファイルまたはトレイルの指定の後にリストされた場合、FORMATASCII は効力がありません。

**注意** データが Replicat プロセスによって処理される場合は、FORMATASCII を使用しないでください。Replicat はデフォルトの正規フォーマットを受け付けます。FORMATSQL または FORMATXML を使用している場合は、FORMATASCII を使用しないでください。

### デフォルト出力

オプションを使用しない場合、FORMATASCII は次のフォーマットでレコードを生成します。

行 1、次のタブ区切りリスト：

- 操作タイプ・インジケータ :I、D、U、V（挿入、削除、更新、圧縮更新）
- ビフォアまたはアフター・イメージ・インジケータ :B または A.
- 表名
- 列名、列値、列名、列値など
- 改行文字（新しい行を開始）

行 2、次のタブ区切り開始トランザクション・レコード：

- 開始トランザクション・インジケータ、B
- トランザクションのコミット時のタイムスタンプ
- コミットが検出されたトランザクション・ログの順序番号
- トランザクション・ログ内のコミット・レコードの相対バイト・アドレス（RBA）

行 3、次のタブ区切りコメント・レコード：

- コミット文字、C
- 改行文字



ソース・トランザクション内のすべてのレコードは、開始インジケータとコミット・インジケータの間に含まれます。コミット・タイムスタンプと RBA の各組み合わせは一意です。

出力フォーマットは、オプションの引数でカスタマイズできます。

**デフォルト** 「デフォルト出力」を参照してください。

**構文** FORMATASCII [, <option>] [, ...]

| オプション                      | 説明                                                                                                                                                                                                                                         |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BCP                        | SQL Server の BCP、DTS、または SQL Server Integration Services (SSIS) バルクロード・ユーティリティと互換性を持つように出力をフォーマットします。                                                                                                                                      |
| DATE   TIME   TS           | 次のいずれかを出力します。 <ul style="list-style-type: none"> <li>◆ DATE は日付を出力します (年～日)。</li> <li>◆ TIME は時間を出力します (年～秒)。</li> <li>◆ TS はトランザクションのタイムスタンプを出力します (年～秒の端数)。</li> </ul>                                                                   |
| DELIMITER <delimiter>      | 代替デリミタ文字 (デフォルトはタブ)。<br>有効な値: <ul style="list-style-type: none"> <li>◆ TAB (タブで区切り)</li> <li>◆ 一重引用符で囲んだ文字 (例: ' / ')</li> </ul>                                                                                                           |
| EXTRACOLS <number>         | 各レコードの末尾に追加列のためのプレースホルダを含めます。このオプションは、ターゲット表にソース表よりも多くの列が含まれる場合に使用してください。                                                                                                                                                                  |
| NAMES   NONAMES            | 出力の一部に列名を包含または除外します。圧縮更新では、PLACEHOLDERS オプションも指定している場合を除き、列名は含まれます。                                                                                                                                                                        |
| NOHDRFIELDS<br>[IND], [OP] | 次のように出力を抑止します。 <ul style="list-style-type: none"> <li>◆ オプションなしの NOHDRFIELDS は、データ値を除くすべてを抑止します。</li> <li>◆ IND は、ビフォアまたはアフター・インジケータ (B または A) およびデータ値を除くすべてを抑止します。</li> <li>◆ OP は、操作タイプ・インジケータ (I、D、U、V) およびデータ値を除くすべてを抑止します。</li> </ul> |
| NOQUOTE                    | 文字データから引用符を除外します。NOQUOTE を指定しない場合、文字は一重引用符で囲まれます。                                                                                                                                                                                          |
| NOTRANSTMTS                | トランザクション情報を除外します。                                                                                                                                                                                                                          |
| NULLISSPACE                | NULL 列を空の列として出力します。NULLISSPACE を指定しない場合、NULL 列は単語 "NULL" として出力されます。                                                                                                                                                                        |
| PLACEHOLDERS               | 行方不明の列に対してプレースホルダを出力します。たとえば、4 列の表で 2 番目および 4 番目の列が見つからない場合、データは次のように出力されます。<br>'ABC',,123,,                                                                                                                                               |

| オプション     | 説明                                                                                     |
|-----------|----------------------------------------------------------------------------------------|
| SQLLOADER | Oracle SQL*Loader ユーティリティまたは IBM Load Utility プログラムと互換性を持つ固定長 ASCII フォーマット・ファイルを生成します。 |

**例**

次の例は、**test.customer** という名前のソース表とサンプル・トランザクションに基づいています。この例では、様々な **FORMATASCII** オプションがどのように出力を構成するかを示します。

**表 test.customer**

|          |          |             |
|----------|----------|-------------|
| CUSTNAME | CHAR(10) | primary key |
| LOCATION | CHAR(10) |             |
| BALANCE  | INTEGER  |             |

**トランザクション**

```
INSERT INTO CUSTOMER VALUES ("Eric", "San Fran", 550);
UPDATE CUSTOMER SET BALANCE = 100 WHERE CUSTNAME = "Eric";
COMMIT;
```

**例 1** オプションなしの **FORMATASCII** では、次の出力が生成されます。

```
B,1997-02-17:14:09:46.421335,8,1873474,
I,A,TEST.CUSTOMER,CUSTNAME,'Eric',LOCATION,
'San Fran',BALANCE,550,
V,A,TEST.CUSTOMER,CUSTNAME,'Eric',BALANCE,100,
C,
```

**例 2** **FORMATASCII, NONAMES, DELIMITER'|'** では、次の出力が生成されます。

```
B|1997-02-17:14:09:46.421335|8|1873474|
I|A|CUSTOMER|'Eric'|'San Fran'|550|
V|A|CUSTOMER|CUSTNAME|'Eric'|BALANCE|100|
C|
```

レコードは圧縮更新され、**PLACEHOLDERS** が使用されなかったため、最後のレコードは、**CUSTNAME** および **BALANCE** 列の列名を返します。

**例 3** **FORMATASCII, NOHDRFIELDS, OP, TS, NONAMES, NOQUOTE** では、次の出力が生成されます。

```
I,CUSTOMER,1997-02-17:14:09:46.421335,Eric,San Fran,550,
V,CUSTOMER,1997-02-17:14:09:46.421335,Eric,,100,
```

圧縮更新レコードの 2 番目の列の不在の値は、2 つの連続するコンマで示されています。

## FORMATSQL

**適用対象** Extract

**FORMATSQL** パラメータでは、デフォルトの Oracle GoldenGate 正規フォーマットではなく、外部 SQL フォーマットでデータを出力します。**FORMATSQL** は、Oracle GoldenGate Replicat 以外のユーティリティで SQL および Enscribe 表に適用可能な、SQL 文 (INSERT、UPDATE および DELETE) を生成します。

**注意** データが Replicat プロセスによって処理される場合は、FORMATSQL を使用しないでください。Replicat はデフォルトの正規フォーマットを受け付けます。FORMATASCII または FORMATXML を使用しているときは、FORMATSQL を使用しないでください。

FORMATSQL 文は、それ以降に定義されているすべての抽出ファイルまたはトレイルに影響します。

## デフォルト出力

オプションなしの FORMATSQL トランザクションは、コンマ区切りフォーマットで次のように出力されます。

- 開始トランザクション・インジケータ、B
- トランザクションのコミット時のタイムスタンプ
- コミットが検出されたトランザクション・ログの順序番号
- トランザクション・ログ内のコミット・レコードの相対バイト・アドレス (RBA)
- SQL 文
- コミット・インジケータ、C
- 改行インジケータ

トランザクション内のすべてのレコードは、開始インジケータとコミット・インジケータの間に含まれます。コミット・タイムスタンプと RBA の各組み合わせは一意です。出力フォーマットは、オプションの引数でカスタマイズできます。

**デフォルト** 「デフォルト出力」を参照してください。

**構文** FORMATSQL [<option>] [, ...]

| オプション       | 説明                                                                                                                         |
|-------------|----------------------------------------------------------------------------------------------------------------------------|
| NONAMES     | 挿入操作の列名を省略します (挿入にはすべての列名が含まれているため)。このオプションにより、ファイル・サイズを低減できます。                                                            |
| NOPKUPDATES | ターゲットの主キーの列に影響する UPDATE 操作を、次に INSERT が続く DELETE に変換します。デフォルト (NOPKUPDATES なし) では、出力は標準の UPDATE 操作です。                      |
| ORACLE      | 次の例のように、日付および時刻列を SQL*Plus が受け付けるフォーマットに変換し、Oracle データベースと互換性を持つようにレコードをフォーマットします。<br>TO_DATE ('1996-05-01', 'YYYY-MM-DD') |

**例** FORMATSQL ORACLE, NONAMES

## FORMATXML

**適用対象** Extract

FORMATXML パラメータでは、デフォルトの Oracle GoldenGate 正規フォーマットではなく、XML フォーマットでデータを出力します。FORMATXML 文は、それ以降に定義されているすべての抽出ファイルまたはトレイルに影響します。

FORMATXML を使用するときは、NOBINARYCHARS パラメータを使用してください。NOBINARYCHARS は、Oracle GoldenGate にバイナリ・データを NULL で終わる文字列として処理させる、ドキュメントに記載され

ていないパラメータです。NOBINARYCHARS を使用する前に、Oracle サポートに連絡してください。詳細は、<http://support.oracle.com> を参照してください。

**注意** データが Replicat プロセスによって処理される場合は、FORMATXML を使用しないでください。Replicat はデフォルトの正規フォーマットを受け付けます。FORMATASCII または FORMATSQLE を使用しているときは、FORMATXML を使用しないでください。

**デフォルト** なし

**構文** FORMATXML [<option>] [, ...]

| オプション                                    | 説明                                                              |
|------------------------------------------|-----------------------------------------------------------------|
| INLINEPROPERTIES  <br>NOINLINEPROPERTIES | XML タグ内にプロパティを含めるか、別に書き込むかどうかを制御します。デフォルトは INLINEPROPERTIES です。 |
| TRANS   NOTRANS                          | トランザクション境界およびコミットのタイムスタンプを XML 出力に含めるかどうかを制御します。TRANS はデフォルトです。 |

**例** FORMATXML NOINLINEPROPERTIES, NOTRANS

## FUNCTIONSTACKSIZE

**適用対象** Extract および Replicat

FUNCTIONSTACKSIZE パラメータでは、Oracle GoldenGate ファンクションの処理に使用されるメモリー・スタックのサイズを制御します。メモリー・スタックには、Oracle GoldenGate ファンクションとやり取りする引数が保持されます。このパラメータは、Oracle GoldenGate からスタック・サイズを増やすように通知するメッセージが返されないかぎり、使用する必要はありません。このようなメッセージは、多数のファンクションまたは引数を使用している場合に返されることがあります。

FUNCTIONSTACKSIZE なしのデフォルトは、200 引数で、この値では Oracle GoldenGate のパフォーマンスおよびシステム・メモリー使用率が最適化されます。このパラメータの値を増やすと、パフォーマンスおよびシステム・メモリー使用率に悪影響が及ぶことがあります。

FUNCTIONSTACKSIZE は、パラメータ・ファイル内で、すべてのパラメータ句よりも前にリストする必要があります。FUNCTIONSTACKSIZE はグローバル・パラメータです。これはパラメータ・ファイル内のすべての句に影響します。

**デフォルト** 200 引数

**構文** FUNCTIONSTACKSIZE <stack size>

| 引数           | 説明                                     |
|--------------|----------------------------------------|
| <stack size> | パラメータ句で許可するファンクション引数の数を示す値 (0 ~ 5000)。 |

**例** FUNCTIONSTACKSIZE 300

## GENLOADFILES

**適用対象** Replicat

GENLOADFILES パラメータでは、*File-to-Database-utility* 初期ロード方法を使用するときに、次のユーティリティと互換性のある実行ファイルと制御ファイルを生成します。

- Oracle の SQL\*Loader ユーティリティ
- Microsoft の BCP、DTS または SQL Server Integration Services (SSIS) ユーティリティ
- IBM の Load Utility (LOADUTIL)

実行ファイルおよび制御ファイルは、Replicat パラメータ・ファイル内の各 MAP 文で生成されます。Replicat は、制御ファイルおよび実行ファイルを生成後に停止し、データを処理しません。

実行ファイルおよび制御ファイルは、ターゲットにロードするデータを含むデータ・ファイルとともに使用します。データ・ファイルを生成するには、Extract パラメータ・ファイル内で FORMATASCII パラメータを使用します。Oracle および z/OS 上の DB2 のユーティリティを使用する場合は、FORMATASCII の SQLLOADER オプションを使用し、Microsoft ユーティリティを使用する場合は BCP オプションを使用します。

FORMATASCII は、ロード・ユーティリティと互換性を持つ外部 ASCII フォーマットで Oracle GoldenGate トレイルまたはファイルに表データを出力します。複数のファイルを指定すると、複数のデータ・ファイルを生成できます。Oracle GoldenGate のロード・ファイル出力の構成方法および初期ロードの実行方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**注意** IBM の Load Utility を使用する場合は、RMTHOST パラメータの PARAMS オプションとともに、-E および -d <defs file> Collector パラメータを指定する必要があります。これらのパラメータは、ASCII から EBCDIC への変換、およびソース定義ファイルの指定のために必要です。

デフォルトでは、GENLOADFILES は次のファイル名を生成します。

- SQL\*Loader 実行ファイルは <source table>.run、制御ファイルは <source table>.ctl という名前が付けられます。<source table> は、MAP 文で指定されたソース表の名前です。
- BCP/DTS/SSIS 実行ファイルは <target table>.bat、制御ファイルは <target table>.fmt という名前が付けられます。<target table> は MAP 文で指定されたターゲット表の名前です。
- Load Utility 実行ファイルは <target table>.run、制御ファイルは <target table>.ctl という名前が付けられます。<target table> は MAP 文で指定されたターゲット表の名前です。

### 制御ファイル

制御ファイルには、テンプレートに基づいて生成されたロード・パラメータが含まれます。Oracle GoldenGate は、SQL\*Loader、BCP/DTS/SSIS、および Load Utility 用のデフォルト・テンプレートを提供します。必要に応じてテンプレートを変更してロード・ルールを変更することも、新しいテンプレートを作成することもできます。

図 15、図 16、および図 17 は、Oracle GoldenGate テンプレートを示しています。テンプレートには、ターゲット表、FORMATASCII によって生成されたデータ・ファイル、およびその他の実行パラメータ用のプレースホルダが含まれています。Oracle GoldenGate は、これらのプレースホルダを Replicat パラメータ・ファイルに指定されたパラメータに基づいた値に置き換えます。

図 15 SQL\*Loader テンプレート sqlldr.tpl

```
File Names
controlfile ?target.ctl
runfile ?target.run
#
Run File Template
sqlldr userid=?pw control=?target log=?target direct=true
#
Control File Template
unrecoverable
load data
infile ?source.dat
truncate
into table ?target
```

図 16 BCP/DTS/SSIS テンプレート bcpfmt.tpl

```
Run File Template
Substitute your database name for <db>
bcp <db>..?target in ?source.dat -U ?user -P ?pw -f ?target.fmt -e ?target.err
#
Control File Template
The value below must specify the BCP version, not the Sybase Adaptive
Server or Microsoft SQL Server version. "bcp -v" can be used to
determine the correct version number.
12.0
```

図 17 Load Utility テンプレート db2cntl.tpl

```
File Names
controlfile ?target.ctl
runfile ?target.run
#
Run File Template
odb2 load
#
Control File Template
LOAD REPLACE INTO TABLE ?target
```

## 実行ファイル

実行ファイルには、ロードを開始するための入力パラメータが含まれます。実行ファイルを実行するには、次のいずれかのコマンドを発行します。

- UNIX コマンド・シェルから SQL\*Loader 実行ファイルを実行します。  
% <table>.run
- DOS シェルから BCP 実行ファイルを実行します。  
> <table>.bat

- z/OS 上の DB2 表にデータをロードするための JCL スクリプトを使用して、Load Utility 実行ファイルを実行します。必要に応じて、ジョブ・スクリプトの他の環境関連パラメータを追加してください。

**注意** WILDCARDRESOLVE パラメータ用の DYNAMIC の設定は、GENLOADFILES パラメータとは互換性がありません。Oracle GoldenGate では、GENLOADFILES が指定される場合、デフォルトで IMMEDIATE が使用されます。

**デフォルト** なし

GENLOADFILES [<template file>]

| 引数              | 説明                                                                                                                                                                |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <template file> | テンプレート・ファイルの完全修飾名。デフォルトのテンプレート・ファイルは、sqlldr.tpl (SQL*Loader)、bcpfmt.tpl (BCP、DTS、または SSIS)、および db2cntl.tpl (z/OS 上の DB2) で、すべて Oracle GoldenGate ホーム・ディレクトリにあります。 |

## GETAPPLOPS | IGNOREAPPLOPS

**適用対象** Extract

GETAPPLOPS または IGNOREAPPLOPS パラメータでは、ローカル Replicat を除くすべてのアプリケーションによって生成された DML 操作を取得または無視します。デフォルトでは、アプリケーション・データは取得されます。

これらのパラメータは、次の目的で GETREPLICATES および IGNOREREPLICATES パラメータとともに有効に活用できます。

- ローカル Replicat によって実行されたデータ操作を、Oracle GoldenGate で抽出するように構成されているビジネス・アプリケーションによって実行されたデータ操作と分離する。IGNOREAPPLOPS および GETREPLICATES を、Replicat1 つのトレイルまたはファイルに対して使用して Replicat 操作のみを包含し、GETAPPLOPS および IGNOREREPLICATES を、別のトレイルまたはファイルに対して使用してビジネス・アプリケーションの操作のみを包含します。
- カスケード構成の一部として使用する。カスケード構成では、Replicat がローカルで適用した変更をローカル Extract が取得し、別のシステムに伝播する必要があります。このケースでは、IGNOREAPPLOPS および GETREPLICATES を使用します。
- 双方向レプリケーションを行う場合にループ検出スキームの一部として使用する。デフォルトの GETAPPLOPS および IGNOREREPLICATES の組み合わせでは、Extract はアプリケーション・データを取得し、同一のデータベース・オブジェクトに送信された Replicat 操作を無視します。このケースでは、これらのパラメータを使用することに加え、Replicat トランザクションを特定できるように Extract を構成する必要があります。

IGNOREREPLICATES の詳細は、199 ページを参照してください。

双方向同期の構成方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

GETAPPLOPS および IGNOREAPPLOPS は、グローバルに使用することも、特定の TABLE 文に関連付けて使用することもできます。同一のパラメータ・ファイル内で複数の文がグローバルに使用されている場合は、実行時に最後のエントリのみが発効します。

## Oracle の順序に対する GETAPPLOPS の使用

Replicat によってレプリケートされる順序を取得するには、GETAPPLOPS を有効にする必要があります。Replicat は、トレース表に反映されないように、自律型トランザクションで順序の更新を発行します。順序更新は、アプリケーション操作のように見えます。

## DDL 操作に対する GETAPPLEOPS の使用

GETAPPLOPS または IGNOREAPPLOPS 機能を DDL 操作に対して使用する場合は、159 ページの DDLOPTIONS パラメータの項を参照してください。

**デフォルト** GETAPPLOPS  
**構文** GETAPPLOPS | IGNOREAPPLOPS

## GETDELETES | IGNOREDELETES

**適用対象** Extract および Replicat

GETDELETES および IGNOREDELETES パラメータでは、Oracle GoldenGate が削除操作を処理するかどうかを制御します。これらのパラメータは表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての TABLE または MAP 文に有効です。

**デフォルト** GETDELETES  
**構文** GETDELETES | IGNOREDELETES

## GETENV

**適用対象** Extract および Replicat

GETENV パラメータでは、SETENV パラメータで設定された環境変数を表示します。結果は、画面およびレポート・ファイルに出力されます。取得する変数ごとに 1 つの GETENV 文を使用します。

**デフォルト** なし  
**構文** GETENV (<environment variable>)

| オプション                  | 説明     |
|------------------------|--------|
| <environment variable> | 環境変数名。 |

**例** 次に、GETENV 文と返される値の例を示します。

```
GETENV (ORACLE_HOME)
ORACLE_HOME = /home/oracle/ora9/product

GETENV (ORACLE_SID)
ORACLE_SID = ora9
```



## GETINSERTS | IGNOREINSERTS

**適用対象** Extract および Replicat

GETINSERTS および IGNOREINSERTS パラメータでは、Oracle GoldenGate が挿入操作を処理するかどうかを制御します。これらのパラメータは表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての TABLE または MAP 文に有効です。

**デフォルト** GETINSERTS

**構文** GETINSERTS | IGNOREINSERTS

## GETREPLICATES | IGNOREREPLICATES

**適用対象** Extract

GETREPLICATES および IGNOREREPLICATES パラメータでは、Replicat が発行した DML トランザクションを、同じシステム上で同じ表を処理している Extract プロセスに取得させるか無視させるかを制御します。

GETREPLICATES および IGNOREREPLICATES は、グローバルに使用することも、特定の TABLE 文に関連付けて使用することもできます。同一のパラメータ・ファイル内で複数の文がグローバルに使用されている場合は、実行時に最後のエントリのみが発効します。

これらのパラメータは、Teradata には無効です。

### Replicat トランザクションの無視

デフォルトでは、Extract は IGNOREREPLICATES および GETAPPLOPS の組み合わせを使用します。この構成では、Extract は Oracle GoldenGate によって同期するように構成されているすべてのアプリケーション・データを取得し、すべての Replicat 操作を無視します。双方向構成では、この設定によって、Replicat が適用するデータが元のシステムにループ・バックする（そのために重複レコード・エラーが発生する）ことを防止できます。

### Replicat トランザクションの取得

カスケード構成で GETREPLICATES と IGNOREAPPLOPS を使用すると、レプリケートされたデータをもう一度仲介システム上の Extract が取得し、最終的なターゲットにレプリケートできます。たとえば、データベース A をデータベース B にレプリケートし、データベース B をデータベース C にレプリケートする場合には、データベース B の Extract に対して GETREPLICATES を使用します。

**注意** ただし、GETREPLICATES が有効な場合でも、TABLE または MAP 文の WHERE または FILTER 句を使用して、特定のレプリート・データを取得から除外できます。

### Replicat トランザクションの識別

一部のデータベースでは、Extract に Replicat トランザクションを無視させる場合、IGNOREREPLICATES または GETREPLICATES を使用するだけでなく、Extract にこれらのトランザクションを識別させる必要があります。

使用するデータベースに応じて、Replicat に適用される DML トランザクションを識別できるように、Extract に明示的な指示を出す必要がある場合とない場合があります。

## z/OS および LUW 上の DB2

Extract パラメータ・ファイル内の次のパラメータ文を使用して、Replicat ユーザー名を識別します。

```
TRANLOGOPTIONS EXCLUDEUSER <user name>
```

このパラメータでは、このユーザーによって生成されたすべてのトランザクションが Replicat トランザクションとしてマークされます。ユーザー名は、Extract で読み出されるトランザクション・レコードに含まれます。

## NonStop SQL/MX

Extract パラメータ・ファイル内の次のパラメータ文を使用して、Replicat チェックポイント表の名前を識別します。

```
TRANLOGOPTIONS FILTERTABLE <table_name>
```

Replicat で、各トランザクションの末尾にあるチェックポイント表にチェックポイント・プロシージャの一部としてチェックポイントが書き込まれます。(この表は、ADD CHECKPOINTTABLE コマンドで作成されます。) この表は、すべての Replicat トランザクションに含まれるので、双方向構成内にある Replicat トランザクションを識別するのに使用できます。FILTERTABLE でチェックポイント表の名前が識別されるので、Extract ではそれに対するすべての操作が含まれているトランザクションが無視されます。

**注意** PURGEDATA では、双方向構成内の NonStop SQL/MX にはサポートされません。PURGEDATA/TRUNCATE 操作は DDL であることから暗黙的なトランザクションであるので、Oracle GoldenGate ではそのトランザクション内のチェックポイントの更新はできません。

## SQL Server

Extract パラメータ・ファイル内の次のパラメータ文を使用して、Replicat トランザクション名を識別します。

```
TRANLOGOPTIONS EXCLUDETRANS <transaction name>
```

このパラメータ文は、Replicat トランザクション名がデフォルトの ggs\_repl 以外に設定されている場合にのみ必要です。

## Sybase

次のいずれかを実行します。

- Extract パラメータ・ファイル内の次のパラメータ文を使用して、Replicat トランザクション名を識別します。

```
TRANLOGOPTIONS EXCLUDETRANS <transaction name>
```

- Extract パラメータ・ファイル内の次のパラメータ文を使用して、Replicat ユーザー名を識別します。

```
TRANLOGOPTIONS EXCLUDEUSER <user name>
```

EXCLUDEUSER では、このユーザーによって生成されたすべてのトランザクションが Replicat トランザクションとしてマークされます。ユーザー名は、Extract で読み出されるトランザクション・レコードに含まれます。

- いずれの操作も行わず、Replicat でデフォルトの名前の ggs\_repl が使用されるようにします。

## Teradata

Teradata データベースに適用される Replicat トランザクションを識別する必要はありません。

## c-tree

Extract では、c-tree データベースに適用される Replicat トランザクションは自動的に識別されます。

## Oracle

(Oracle 10g 以上) 次のいずれかを実行して、Replicat データベース・ユーザーを指定します。このユーザーで生成されるすべてのトランザクションが、取得から除外されます。この情報は、トランザクション・レコードに含められ、Extract による使用が可能になります。

- Extract パラメータ・ファイル内の次のパラメータ文を使用して、Replicat データベース・ユーザーを名前で識別します。

```
TRANLOGOPTIONS EXCLUDEUSER <user name>
```

- Extract パラメータ・ファイル内の次のパラメータ文を使用して、Replicat データベース・ユーザーを数値の Oracle ユーザー ID (uid) で識別します。

```
TRANLOGOPTIONS EXCLUDEUSERID <user-id>
```

(Oracle 9i より前) GGSCI で ADD TRACETABLE コマンドを実行して証跡表を作成します。

## これらのパラメータの補足情報

- チェックポイント表作成の詳細は、80 ページの ADD CHECKPOINTTABLE コマンドの項を参照してください。
- トレース表の作成と使用の詳細は、359 ページの TRACETABLE パラメータの項、および 83 ページの ADD TRACETABLE の項を参照してください。
- TRANLOGOPTIONS の詳細は、360 ページを参照してください。
- SQLEXEC の詳細は、307 ページを参照してください。
- カスケードまたは双方向構成の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。
- GETAPPLPLOS および IGNOREAPPLPLOS の詳細は、197 ページを参照してください。
- GETAPPLPLOS または IGNOREAPPLPLOS 機能を DDL 操作に対して使用する場合は、159 ページの DDLOPTIONS パラメータの項を参照してください。

デフォルト    IGNOREREPLICATES

構文            GETREPLICATES | IGNOREREPLICATES

# GETTRUNCATES | IGNORETRUNCATES

適用対象    Extract および Replicat

GETTRUNCATES および IGNORETRUNCATES パラメータでは、Oracle GoldenGate プロセスが表切捨て操作を処理するかどうかを制御します。デフォルトでは、切捨て操作はソースでの取得やターゲットへのレプリケーションが行われません。

切捨ては次のようにサポートされます。

- 切捨て操作の抽出は、Oracle、SQL Server 2000、(CU6 にアップグレードしている場合) SQL Server 2005、および Sybase でサポートされています。Sybase では、表名がデータベース内のすべてのスキーマで一意である必要があります。

- 切捨て操作の送信は、Oracle、SQL Server 2000、(CU6 にアップグレードしている場合) SQL Server 2005、Sybase、DB2 LUW、DB2 z/OS、MySQL、Ingres、および TRUNCATE コマンドをサポートするその他の ODBC ターゲットでサポートされています。

次の点に注意してください。

- DB2 LUW では TRUNCATE コマンドがサポートされていないため、Replicat は NULL (空白) ファイルから IMPORT REPLACE を実行することによって、切捨て操作をレプリケートします。
- Oracle 10gr2 以降、データベースは空の表の切捨て操作のログを取らないため、これらの操作は Oracle GoldenGate に取得されません。この目的には、Oracle GoldenGate の DDL サポートを使用できます。
- Oracle リリース 10gr2 以降、データベースは空のパーティションに対する切捨て操作のログを取らないため、表に空のパーティションが含まれている場合、Oracle GoldenGate は表レベルの切捨てを取得できません。そのため、これらの Oracle リリースでは、パーティション化された表に対して GETTRUNCATES を使用しないでください。空のパーティションを含む可能性がある表に対する切捨て操作の取得には、Oracle GoldenGate の DDL サポートを使用できます。

GETTRUNCATES および IGNORETRUNCATES は、表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての TABLE または MAP 文に有効です。

**デフォルト** IGNORETRUNCATES  
**構文** GETTRUNCATES | IGNORETRUNCATES

## GETUPDATEAFTERS | IGNOREUPDATEAFTERS

**適用対象** Extract および Replicat

GETUPDATEAFTERS および IGNOREUPDATEAFTERS パラメータでは、Oracle GoldenGate に処理されるレコードに、更新されるレコードのアフター・イメージを含めるかどうかを制御します。アフター・イメージには、更新の結果が含まれます。

これらのパラメータは表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての TABLE または MAP 文に有効です。

**デフォルト** GETUPDATEAFTERS  
**構文** GETUPDATEAFTERS | IGNOREUPDATEAFTERS

## GETUPDATEBEFORES | IGNOREUPDATEBEFORES

**適用対象** Extract および Replicat

GETUPDATEBEFORES および IGNOREUPDATEBEFORES パラメータでは、Oracle GoldenGate に処理されるレコードに、更新される列のビフォア・イメージを含めるかどうかを制御します。ビフォア・イメージには、レコードが更新される前に存在していた列の詳細情報が含まれます。ビフォア・イメージを抽出するには Extract パラメータ・ファイルで、ビフォア・イメージをレプリケートするには Replicat パラメータ・ファイルで、GETUPDATEBEFORES を使用します。

ビフォア・イメージとアフター・イメージを比較して、トランザクションの最終結果を確認したり、その他の差分計算を実行したりできます。たとえば、更新前の BALANCE フィールドが 100 ドルで更新後が 120 ドルの場合、これらを比較して 20 ドルの差異を確認できます。Oracle GoldenGate の列変換ファンクションを使用して、比較および計算を実行できます。

GETUPDATEBEFORES を使用して、トランザクション履歴表を保持することもできます。差分計算の実行とトランザクション履歴の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

GETUPDATEBEFORES および IGNOREUPDATEBEFORES パラメータは、表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての TABLE または MAP 文に有効です。

**デフォルト** IGNOREUPDATEBEFORES  
**構文** GETUPDATEBEFORES | IGNOREUPDATEBEFORES

## GETUPDATES | IGNOREUPDATES

**適用対象** Extract および Replicat

GETUPDATES および IGNOREUPDATES パラメータでは、Oracle GoldenGate プロセスが更新操作を処理するかどうかを制御します。これらのパラメータは表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての TABLE または MAP 文に有効です。

**デフォルト** GETUPDATES  
**構文** GETUPDATES | IGNOREUPDATES

## GGSCHEMA

**適用対象** GLOBALS

GGSCHEMA パラメータでは、Oracle GoldenGate による Oracle DDL の同期をサポートする DDL データベース・オブジェクトを含むスキーマ名を指定します。このパラメータは、Oracle GoldenGate for Oracle にのみ有効です。

**デフォルト** なし  
**構文** GGSCHEMA <schema\_name>

| 引数            | 説明         |
|---------------|------------|
| <schema_name> | DDL スキーマ名。 |

## GROUPTRANSOPS

**適用対象** Replicat

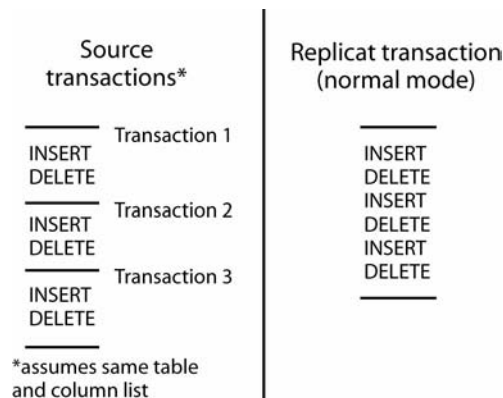
通常モードで操作された場合に Replicat トランザクションに含まれる SQL 操作の数を制御するには、GROUPTRANSOPS パラメータを使用します。Replicat トランザクションでの操作の回数を増やすことにより、次のように Oracle GoldenGate のパフォーマンスが向上します。

- Replicat で実行されるトランザクションの数の減少。
- チェックポイント・ファイルおよびチェックポイント表が使用される場合、I/O アクティビティの減少。Replicat では、スケジュールされたチェックポイントに加え、トランザクションがターゲットに適用されるたびにチェックポイントが実行されます。

Replicat は、ソース・トランザクションからトランザクションの順序で操作を蓄積し、ターゲット上で 1 つのトランザクション内のグループとして操作を適用します。GROUPTRANSOPS では、ソース・トランザクションの分割を回避するために、絶対値ではなく最小値を設定します。Replicat は、グループ内の最後のソース・トランザクションからすべての操作を受信するまで待機してから、ターゲット・トランザクションを適用します。

たとえば、トランザクション A に 500 の操作が含まれ、トランザクション B に 600 の操作が含まれている場合、GROUPTRANSOPS がデフォルトの 1,000 に設定されているとしても、Replicat トランザクションには 1,100 すべての操作が含まれることになります。逆に、トレイルに処理するデータが残っていないければ、Replicat は GROUPTRANSOPS に設定されている値に到達する前にトランザクションを適用することがあります。

図 18 Replicat 標準モード



ソース・トランザクションの境界とターゲット・トランザクションの境界の差によって、ターゲット・データの遅延が長びくことがあるため、GROUPTRANSOPS を不必要に高い値に設定しないでください。

**デフォルト** 1000 操作  
**構文** GROUPTRANSOPS <min transaction count>

| 引数                      | 説明                                                                          |
|-------------------------|-----------------------------------------------------------------------------|
| <min transaction count> | Replicat トランザクションで適用される最小操作数。値が 1 の場合、ソース・トランザクションと同一のトランザクション境界内の操作を実行します。 |

**例** GROUPTRANSOPS 2000

## HANDLECOLLISIONS | NOHANDLECOLLISIONS

**適用対象** Replicat

HANDLECOLLISIONS および NOHANDLECOLLISIONS パラメータでは、Replicat がターゲットに SQL を適用するときに、重複レコード・エラーおよび行方不明レコード・エラーの解決を試行させるかどうかを制御します。このようなエラーは、Oracle GoldenGate がソース表に対して行われたトランザクション変更をレプリケートしている間に、初期ロード（ソース表のデータをターゲット表にロード）が実行される場合に発生することがあります。HANDLECOLLISIONS を使用すると、初期ロードが終了して Oracle GoldenGate がレプリケートされた変更を適用するときに、Replicat はターゲット表の重複レコードを

上書きし、行方不明レコードのエラーには別のエラー処理方法で対処します。

HANDLECOLLISIONS および NOHANDLECOLLISIONS は、次の方法で使用できます。

- パラメータ・ファイルのルート・レベルで HANDLECOLLISIONS または NOHANDLECOLLISIONS のいずれかを使用して、すべての MAP 文に適用できます。
- 必要に応じて表グループのエラー処理を有効化または無効化する個別スイッチとして、HANDLECOLLISIONS および NOHANDLECOLLISIONS を使用できます。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての MAP 文に有効です。
- 1 つの MAP 文の中で、HANDLECOLLISIONS および NOHANDLECOLLISIONS を使用し、特定の表に対してこの機能を有効化または無効化できます。217 ページを参照してください。

上記のどの方法も組み合わせて使用できます。MAP 文内で使用する場合は、他の設定よりも優先されます。表グループの個別スイッチとして使用する場合は、グローバル設定よりも優先されます。たとえば、グローバルに NOHANDLECOLLISIONS を設定し、MAP 文の中で HANDLECOLLISIONS を使用して特定の表にのみ有効化できます。

## HANDLECOLLISIONS の動作

次の例で、HANDLECOLLISIONS の動作を説明します。

Replicat が重複レコード・エラーを検出すると、初期ロードで適用されたレコードは、変更レコードによって上書きされます。操作の観点からすると、重複レコード・エラーは、無視するよりも変更を適用するほうが安全です。

Replicat は、更新または削除操作中に行方不明レコード・エラーを検出すると、変更レコードを破棄します。このようなエラーは、表データが初期ロード・プロセスによって抽出される前に、ソース表でレコードが変更および削除されている場合に発生します。次に例を示します。

1. アプリケーションがソース表のレコード A1 を更新します。
2. Extract がこの更新を抽出します。
3. アプリケーションがソース表 1 のレコード A を削除します。
4. Extract がこの削除を抽出します。
5. Oracle GoldenGate が、レコード A が存在しないソース表 1 から初期ロード・データを抽出します。
6. Oracle GoldenGate がレコード A なしで初期ロードを適用します。
7. Replicat がレコード A の更新の適用を試みます。
8. データベースは "レコード行方不明" エラーを返します。
9. Replicat がレコード A の削除の適用を試みます。
10. データベースは "レコード行方不明" エラーを返します。

初期ロード中に取得されたトランザクション変更がターゲット表に適用されたら、HANDLECOLLISIONS を無効化し、その後のエラーが Oracle GoldenGate によって自動的に処理されることを防ぐ必要があります。初期同期以降に発生するエラーは異常な状態を意味するため、解決方法を特定できる担当者によって検証される必要があります。たとえば、行方不明エラーは、ソース表に存在するデータがターゲット・システムから誤って削除されたことが原因の可能性があります。

HANDLECOLLISIONS は、次の方法で無効化できます。

- Replicat を停止し、Replicat パラメータ・ファイルから HANDLECOLLISIONS を削除する（ターゲットに遅延が生じる可能性があります）か、パラメータ・ファイルを編集して該当の MAP 文の前に NOHANDLECOLLISIONS を追加します。
- Replicat の実行中に GGSCI を実行し、SEND REPLICAT コマンドと該当の表に対する NOHANDLECOLLISIONS オプションを使用します（56 ページを参照してください）。SEND REPLICAT を使用する場合は、HANDLECOLLISIONS が再度有効化されないように、Replicat の実行を次に開始する前に、パラメータ・ファイルから HANDLECOLLISIONS を削除するか、NOHANDLECOLLISIONS パラメータを追加してください。

**デフォルト** なし

**構文** HANDLECOLLISIONS | NOHANDLECOLLISIONS

**例 1** 次の例では、パラメータ・ファイル内のすべての MAP 文に対して HANDLECOLLISIONS を有効にします。

```
HANDLECOLLISIONS
MAP hr.emp, TARGET hr.emp;
MAP hr.job_hist, TARGET hr.job_hist;
MAP hr.dep, TARGET hr.dep;
MAP hr.country, TARGET hr.country;
```

**例 2** 次の例では、パラメータ・ファイル内の一部の MAP 文に対してのみ HANDLECOLLISIONS を有効にし、それ以外に対しては無効にします。

```
HANDLECOLLISIONS
MAP hr.emp, TARGET hr.emp;
MAP hr.job_hist, TARGET hr.job_hist;
NOHANDLECOLLISIONS
MAP hr.dep, TARGET hr.dep;
MAP hr.country, TARGET hr.country;
```

**例 3** 次に、グローバルおよび MAP レベルを組み合わせた使用例を示します。指定された表では、MAP レベルの指定がグローバル指定よりも優先されます。

```
HANDLECOLLISIONS
MAP hr.emp, TARGET hr.emp;
MAP hr.job_hist, TARGET hr.job_hist;
MAP hr.dep, TARGET hr.dep, NOHANDLECOLLISIONS;
MAP hr.country, TARGET hr.country, NOHANDLECOLLISIONS;
```

## HANDLETPKUPDATE

**適用対象** Replicat

HANDLETPKUPDATE パラメータでは、トランザクションに一時主キー更新が含まれているときの制約エラーを防止します。一時主キー更新とは、複数の行の主キーに影響する更新文です。通常このタイプの文は、x+n の式、または新しい値の 1 つを古い値の 1 つと同じにするように値をシフトする手法を使用します。

次に、この状況を発生させる可能性がある順序値の変更の例を示します。この例では、表を "ITEM"、主キー列名を "CODE"、この表の行の主キーの値を 1、2、3 とします。

```
update item set code = 2 where code = 1;
update item set code = 3 where code = 2;
update item set code = 4 where code = 3;
```



この例では、最初の更新がターゲットに適用されると、表にすでに主キー値 2 が存在するため、エラーが発生します。ターゲット・トランザクションは、制約違反エラーを返します。デフォルトでは、Replicat はこうした違反を検出または処理せずに異常終了します。

HANDLETPKUPDATE を使用すると、ターゲット表に DEFERRABLE が制約として作成されます。INITIALLY DEFERRED または INITIALLY IMMEDIATE を指定できます。Replicat は、必要に応じて状態を DEFERRED に変更します。この状態では、Replicat トランザクションがコミットされるときにのみ制約がチェックされます。

ターゲットの制約を DEFERRABLE にできない場合、Replicat は HANDLECOLLISIONS および REPERROR パラメータに指定された既存のルールに従ってエラーを処理するか、異常終了します。

**デフォルト** 一時主キー更新で異常終了する

**構文** HANDLETPKUPDATE

## INCLUDE

**適用対象** Extract および Replicat

INCLUDE パラメータでは、パラメータ・ファイルにマクロ・ライブラリを含めます。

**デフォルト** なし

**構文** INCLUDE <path name>

| 引数          | 説明               |
|-------------|------------------|
| <path name> | ライブラリ・ファイルの完全パス。 |

**例** 次の例では、マクロ・ライブラリ mdatelib.mac を含めます。

```
INCLUDE /ggs/dirprm/mdatelib.mac
```

## INSERTAPPEND | NOINSERTAPPEND

**適用対象** Replicat

Replicat で Oracle ターゲット表に INSERT が適用される際に APPEND ヒントが使用されるかどうかを制御するには、INSERTAPPEND および NOINSERTAPPEND パラメータを使用します。これらのパラメータは、Oracle データベースにのみ有効です。

これらのパラメータは、2 つの方法で使用できます。パラメータ・ファイルのルートでスタンドアロン・パラメータとして使用される場合、1 つはその後のすべての TABLE または MAP 文に対して、もう 1 つが発生するまでそのまま有効になります。MAP 文内で使用される場合、MAP 文の前にくるスタンドアロンの INSERTAPPEND または NOINSERTAPPEND エントリよりも優先されます。

INSERTAPPEND により、INSERT 操作が Oracle ターゲット表に適用される際に Replicat で APPEND\_VALUES ヒントが使用されるようになります。複製されたトランザクションが大きく、同じ表内に複数の挿入が含まれる際のパフォーマンス向上のために使用されるのに適しています。トランザクションが小さい場合、INSERTAPPEND を使用するとパフォーマンスが低下することがあります。APPEND ヒントを使用する時期の詳細は、Oracle ドキュメントを参照してください。

INSERTAPPEND を使用する際には、BATCHSQL パラメータを使用する必要があります。BATCHSQL が使用されない場合、Replicat は異常終了します。

MAP 構文の詳細は、217 ページを参照してください。

**デフォルト** NOINSERTAPPEND

**構文** INSERTAPPEND | NOINSERTAPPEND

**例** 次の例では、inventory 表を除き、fine スキーマのすべての表で INSERTAPPEND が使用されます。

```
INSERTAPPEND
MAP fin.*, TARGET fin.*;
MAPEXCLUDE fin.inventory;
NOINSERTAPPEND
MAP fin.inventory, TARGET fin.inventory;
```

## INSERTALLRECORDS

**適用対象** Replicat

現在のバージョンのみでなく、すべての操作のレコードがターゲット・レコードに保持されるようにするには、INSERTALLRECORDS パラメータを使用します。INSERTALLRECORDS により、Replicat であらゆる変更操作に関するレコードが新しいレコードとしてデータベースに挿入されるようになります。初期の挿入、およびその後の更新や削除は、ある時点のスナップショットとして保持されます。

履歴データと特別なトランザクション情報の組み合わせにより、さらに便利なターゲット・レポート・データベースを作成できます。トランザクション履歴表の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

このパラメータは、MAP 文内でも使用できます。217 ページを参照してください。

**デフォルト** なし

**構文** INSERTALLRECORDS

## INSERTDELETES | NOINSERTDELETES

**適用対象** Replicat

INSERTDELETES および NOINSERTDELETES パラメータでは、Oracle GoldenGate がソースの削除操作をターゲット・データベース上で挿入操作に変換するかどうかを制御します。これらのパラメータは表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての MAP 文に有効です。

**デフォルト** NOINSERTDELETES

**構文** INSERTDELETES | NOINSERTDELETES

## INSERTMISSINGUPDATES | NOINSERTMISSINGUPDATES

**適用対象** Replicat

INSERTMISSINGUPDATES および NOINSERTMISSINGUPDATES パラメータでは、ターゲット・レコードが存在しない場合に Oracle GoldenGate がソース・レコードに基づいてレコードを挿入するかどうかを制御します。

INSERTMISSINGUPDATES は、行方不明のレコードに対する更新を挿入しますが、ソース・データベースが非圧縮更新を使用している場合にのみ使用するようになっています。ターゲット・データベースが、行方

不明の列の値に NULL 値の使用を許可する場合には、圧縮更新を使用するデータベースとともに使用できます。

デフォルトの NOINSERTMISSINGUPDATES が有効な場合は、行方不明レコードによってエラーが発生し、REPERERROR の設定に応じて、トランザクションは異常終了することがあります。

INSERTMISSINGUPDATES および NOINSERTMISSINGUPDATES パラメータは、表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての MAP 文に有効です。

**デフォルト** NOINSERTMISSINGUPDATES  
**構文** INSERTMISSINGUPDATES | NOINSERTMISSINGUPDATES

## INSERTUPDATES | NOINSERTUPDATES

**適用対象** Replicat

INSERTUPDATES および NOINSERTUPDATES パラメータでは、Oracle GoldenGate プロセスが非圧縮更新操作を挿入操作に変換するかどうかを制御します。これらのパラメータは表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての MAP 文に有効です。

**デフォルト** NOINSERTUPDATES  
**構文** INSERTUPDATES | NOINSERTUPDATES

## LAGCRITICAL

**適用対象** Manager

重要とみなされるラグしきい値を指定し、しきい値に達したときに警告メッセージがエラー・ログに書き込まれるようにするには、LAGCRITICALSECONDS、LAGCRITICALMINUTES または LAGCRITICALHOURS パラメータを使用します。このパラメータは、ローカル・システム上の Extract および Replicat プロセスに影響します。

**デフォルト** ラグ情報をレポートしない  
**構文** LAGCRITICALSECONDS <seconds> |  
LAGCRITICALMINUTES <minutes> |  
LAGCRITICALHOURS <hours>

| 引数        | 説明          |
|-----------|-------------|
| <seconds> | ラグしきい値（秒）。  |
| <minutes> | ラグしきい値（分）。  |
| <hours>   | ラグしきい値（時間）。 |

**例** LAGCRITICALSECONDS 60

## LAGINFO

**適用対象**     Manager

エラー・ログへのラグ情報のレポート回数を指定するには、LAGINFOSECONDS、LAGINFOMINUTES または LAGINFOHOURS パラメータを使用します。ラグが、LAGCRITICAL パラメータで指定された値よりも大きい場合、Manager でラグが重要であるとレポートされます。大きくない場合は、ラグは情報メッセージとしてレポートされます。値がゼロ (0) の場合、LAGREPORTMINUTES または LAGREPORTHOURS パラメータで指定された頻度でメッセージがレポートされます。

**デフォルト**     ラグ情報をレポートしない

**構文**            LAGINFOSECONDS <seconds> |  
                  LAGINFOMINUTES <minutes> |  
                  LAGINFOHOURS <hours>

| 引数         | 説明                  |
|------------|---------------------|
| <seconds>  | ラグをレポートする間隔 (秒)。    |
| <minutes > | ラグをレポートする間隔 (分)。    |
| <hours>    | ラグ情報をレポートする間隔 (時間)。 |

**例**                LAGINFOHOURS 1

## LAGREPORT

**適用対象**     Manager

Manager による Extract および Replicat ラグのチェック間隔を指定するには、LAGREPORTMINUTES または LAGREPORTHOURS パラメータを使用します。

**デフォルト**     なし

**構文**            LAGREPORTMINUTES <minutes> | LAGREPORTHOURS <hours>

| 引数        | 説明                |
|-----------|-------------------|
| <minutes> | ラグ情報を確認する間隔 (分)。  |
| <hours>   | ラグ情報を確認する間隔 (時間)。 |

**例**                LAGREPORTHOURS 1

## LIST | NOLIST

|              |                                                                                                                                                                                       |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>適用対象</b>  | Extract および Replicat                                                                                                                                                                  |
|              | LIST および NOLIST パラメータでは、レポート・ファイルにマクロ・ライブラリのマクロをリストするかどうかを制御します。レポートへのリストを有効化または無効化するには、パラメータ・ファイル内かマクロ・ライブラリ・ファイル内に LIST または NOLIST パラメータを配置します。NOLIST を使用すると、レポート・ファイルのサイズが小さくなります。 |
| <b>デフォルト</b> | LIST                                                                                                                                                                                  |
| <b>構文</b>    | LIST   NOLIST                                                                                                                                                                         |
| <b>例</b>     | 次の例では、NOLIST によって、hugelib マクロ・ライブラリのマクロをレポートへのリストから除外します。INCLUDE 文の後に LIST を使用し、それ以降のマクロを通常のリスト設定に戻します。<br><br>NOLIST<br>INCLUDE /ggs/hugelib.mac<br>LIST                             |

## LOBMEMORY

|             |                                                                                                                                                                                                                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>適用対象</b> | z/OS 上の DB2 および NonStop SQL/MX の場合の Extract および Replicat                                                                                                                                                                                                                                        |
|             | LOB が含まれるトランザクションのキャッシュに使用可能なメモリーおよび一時ディスク領域を制御するには、LOBMEMORY パラメータを使用します。Oracle GoldenGate では、コミット済トランザクションのみがターゲット・データベースに適用されるので、コミットまたはロールバック・インジケータが受信されるまで LOB を格納するのに十分なシステム・メモリーが必要です。<br><br>このパラメータは、z/OS 上の DB2 データベースおよび NonStop SQL/M データベース用です。その他のデータベースには、CACHEMGR パラメータを使用します。 |

### LOBMEMORY を使用したメモリー管理について

LOBMEMORY は、LOB トランザクション用の Oracle GoldenGate のキャッシュ・サイズの調整、およびキャッシュのサイズを超えるデータを格納するためのディスク上における一時的な場所の定義を可能にします。キャッシュ・サイズの合計、トランザクションごとのメモリー・サイズ、初期と増分メモリーの割当て、およびディスク格納領域の定義のための各オプションがあります。

LOB トランザクションが RAM により指定されるメモリー・プールに追加され、それぞれ、TRANSRAM に達するとディスクにフラッシュされます。メモリーの初期量は、INITTRANSRAM に基づいて各トランザクションに割り当てられ、必要に応じて RAMINCREMENT で指定されている量ずつ、最大 TRANSRAM で設定されている量まで増加します。このため、TRANSRAM の値は、(INITTRANSRAM + RAMINCREMENT) の合計で割り切れる数値である必要があります。

|              |                                                                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>デフォルト</b> | オプションのデフォルト値を参照してください。                                                                                                                                                                          |
| <b>構文</b>    | LOBMEMORY<br>[RAM <size>]<br>[TRANSRAM <size>]<br>[TRANSALLSOURCES <size>]<br>[INITTRANSRAM <size>]<br>[RAMINCREMENT <size>]<br>[DIRECTORY (<directory name>, <max dir size>, <max file size>)] |

| オプション                                                         | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RAM <size>                                                    | <p>すべてのキャッシュ済 LOB トランザクションに使用されるメモリの総量を指定します。デフォルト値は <b>200MB</b> です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m k</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| TRANSRAM <size>                                               | <p>単一の LOB トランザクションに使用されるメモリの総量を指定します。デフォルト値は <b>50MB</b> です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m k</p> <p>最適な結果を得るには、TRANSRAM を INITTRANSRAM と RAMINCREMENT の両方で割り切れる値に設定する必要があります。</p>                                                                                                                                                                                                                                                                                                                                                                                             |
| TRANSALLSOURCES <size>                                        | <p>単一の LOB トランザクションに使用されるメモリの総量およびディスク領域を指定します。デフォルト値は、使用可能なメモリ（メモリおよびディスク）の総量の <b>50%</b> です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m k</p>                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| INITTRANSRAM <size>                                           | <p>1 つの LOB トランザクションに割り当てられる最初のメモリ量を指定します。デフォルト値は <b>50KB</b> です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m k</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| RAMINCREMENT <size>                                           | <p>LOB トランザクションに追加のメモリが必要な場合に、増分するメモリの量を指定します。デフォルト値は <b>50KB</b> です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m </p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| DIRECTORY (<directory name>, <max dir size>, <max file size>) | <p>LOB トランザクション・データ用のディスク領域が TRANSRAM で指定した最大値を超える場合に、一時的なディスク領域を指定します。複数の DIRECTORY を指定できます。</p> <ul style="list-style-type: none"> <li>◆ &lt;directory&gt; は、ディレクトリの完全修飾名です。デフォルト値は、Oracle GoldenGate ディレクトリの dirtmp サブディレクトリです。</li> <li>◆ &lt;max dir size&gt; は、このディレクトリ内のすべてのファイルの最大サイズです。デフォルト値は <b>2GB</b> です。指定した領域が使用できない場合は、使用可能なディスク領域の <b>75%</b> が使用されます。</li> <li>◆ &lt;max file size&gt; は、このディレクトリ内の各ファイルの最大サイズです。デフォルト値は <b>200MB</b> です。</li> </ul> <p>この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m k</p> <p>ディレクトリ・サイズおよびファイル・サイズは、RAM で指定したメモリのサイズより大きな値である必要があります。</p> |

| オプション | 説明                                                                                                                                                                                                                                                                                                                          |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>ファイル名には、次の形式を使用します。</p> <pre>&lt;group&gt;_blob_00001.mem</pre> <p>または</p> <pre>&lt;PID&gt;_blob_00001.mem</pre> <p>グループ名はオンライン・プロセスに使用されます。システム・プロセス ID 番号 (PID) は、SPECIALRUN パラメータで指定されるバッチ実行数に使用されます。</p> <p>スレッド化された Extract の場合の形式は、データベースごとに以下のような形式になります。</p> <pre>&lt;group&gt;_&lt;thread #&gt;_00001.mem</pre> |

**例 1** 次の例では、INITTRANSRAM で初期割当て値を指定した後、RAMINCREMENT で 9 回の増分を許可した場合に、データのディスクへのフラッシュ前に各トランザクションを 10 回増分できるようになります。

```
LOBMEMORY DIRECTORY(c:\test\dirtmp, 3000000000,
3000000000), RAM 8000K, TRANSRAM 1000K, INITTRANSRAM 100K,
RAMINCREMENT 100K
```

**例 2** 次の例は、前述の例と同じですが、2 番目のディレクトリを追加しています。

```
LOBMEMORY DIRECTORY(c:\test\dirtmp, 3000000000,
3000000000), DIRECTORY (c:\test\dirtmp2, 1000000000,
50000000), RAM 8000K, TRANSRAM 1000K, INITTRANSRAM 100K,
RAMINCREMENT 100K
```

**注意** これらの例では、スペースの制約のため、パラメータ指定文が複数行にわたって記述されています。実際のパラメータ・ファイルでは、複数行にわたってパラメータを指定する場合は、各行の末尾にアンパサンド文字 (&) を含める必要があります。

## MACRO

**適用対象** Extract および Replicat

MACRO パラメータでは、Oracle GoldenGate マクロを作成します。マクロ使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** なし

**構文** 次を示す順序で使用する必要があります。

```
MACRO <macrochar><macro name>
PARAMS (<macrochar><paramname> [, ...])
BEGIN
<macro body>
END;
```

| 引数           | 説明                                                                                                                                                                                                                                                                                                                                                            |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <macrochar>  | <p>マクロ文字。マクロおよびパラメータ名は、マクロ文字で開始する必要があります。パラメータ・ファイル内でマクロ文字で始まる要素はすべて、マクロまたはマクロ・パラメータとみなされます。</p> <p>デフォルトのマクロ文字は、次の例に示すようにシャープ (#) 文字です。</p> <pre>MACRO #macro1 PARAMS (#param1, #param2)</pre> <p>マクロ文字を変更するには、<b>MACROCHAR</b> パラメータを使用します。</p>                                                                                                             |
| <macro name> | <p>マクロ名。マクロ名は大 / 小文字が区別されません。引用符は使用しないでください (使用すると、マクロ名はテキストとして扱われ、無視されます)。</p>                                                                                                                                                                                                                                                                               |
| <paramname>  | <p>マクロに対するパラメータを記述します。パラメータ名は大 / 小文字が区別されません。引用符は使用しないでください (使用すると、パラメータ名はテキストとして扱われ、無視されます)。パラメータ句はオプションです。</p> <p>マクロで使用するすべてのパラメータは、<b>PARAMS</b> 文で宣言する必要があり、マクロを呼び出すときは、呼び出し文に各パラメータの値が含まれている必要があります。</p> <p>例 1 に、パラメータを取るマクロを示します。</p> <p>パラメータなしのマクロ (頻繁に使用するコマンドを含むマクロなど) も作成できます。例 2 を参照してください。</p> <p>他のマクロをパラメータとして使用することもできます。例 3 を参照してください。</p> |
| BEGIN        | <p>マクロ本文を開始します。マクロ本文の前に指定する必要があります。</p>                                                                                                                                                                                                                                                                                                                       |
| <macro body> | <p>マクロ名本文。マクロ本文には、次のいずれかのタイプの文を含めることができます。</p> <ul style="list-style-type: none"> <li>◆ 次のような単純なパラメータ文： <pre>COL1 = COL2</pre> </li> <li>◆ 次のような複雑な文： <pre>COL1 = #val2</pre> </li> <li>◆ 次のような他のマクロの呼び出し： <pre>#colmap(COL1, #sourcecol)</pre> </li> </ul>                                                                                                   |
| END          | <p>マクロ定義を終了します。</p>                                                                                                                                                                                                                                                                                                                                           |

**例 1** 次の例では、パラメータを取るマクロを定義します。

```
MACRO #make_date
PARAMS (#year, #month, #day)
BEGIN
@DATE("YYYY-MM-DD", "CC", @IF(#year < 50, 20, 19),
"YY", #year, "MM", #month, "DD", #day)
END;
```



**例 2** 次の例では、パラメータを必要としないマクロを定義します。

```
MACRO #option_defaults
BEGIN
GETINSERTS
GETUPDATES
GETDELETES
INSERTDELETES
END;
```

**例 3** 次の例では、`#make_date` という名前の他のマクロを呼び出す `#assign_date` という名前のマクロを定義します。

```
MACRO #assign_date
PARAMS (#target_col, #year, #month, #day)
BEGIN
#target_col = #make_date (#year, #month, #day)
END;
```

## MACROCHAR

**適用対象** Extract および Replicat

MACROCHAR パラメータでは、マクロ文字を `#` 以外の文字に変更します。指定したマクロ文字で始まるパラメータ・ファイル内の要素はすべて、マクロまたはマクロ・パラメータとみなされます。

たとえば、表名に `#` 文字が含まれている場合は、マクロ文字を変更する必要があります。

MACROCHAR は一度だけ指定でき、最初のマクロ文の前に指定する必要があります。

**デフォルト** `#` (シャープ記号)

**構文** MACROCHAR <character>

| 引数          | 説明                                                                            |
|-------------|-------------------------------------------------------------------------------|
| <character> | マクロ文字として使用する文字。有効なマクロおよびパラメータ文字は英数字で、アンダースコア文字 ( <code>_</code> ) も含むことができます。 |

**例** 次の例では、`$` をマクロ文字として定義します。

```
MACROCHAR $
MACRO $mymac
PARAMS ($p1)
BEGIN
col = $p1
END;
```

## Extract 用 MAP

**適用対象** Extract

Extract 用の MAP 文は、ALTID コンポーネントを使用してオブジェクト ID をオブジェクト名にマップするときに使用します。ALTID では、Extract が接続先データベース以外のデータベースによって生成された Oracle トランザクション・ログを取得する場合に、正しいオブジェクト ID を指定します。この構成は、Oracle GoldenGate が本番環境の（ソース）データベースへの直接接続を許可されていない状態で、本番環境のトランザクションを取得する必要があるときに必要です。

こうしたケースでは、Extract はライブ・スタンバイ・データベースまたはその他の複製データベースに接続しますが、本番環境のデータベースから送信されたトランザクション・ログを読み取ります。代替データベースのカatalogへの問合せによって、Extract はトランザクション・データを有効な SQL 文に展開するために必要なメタデータを取得できますが、表のローカル・オブジェクト ID が本番環境のデータベース（およびトランザクション・ログ）のものとは異なるため、この問い合わせで取得したオブジェクト ID は使用できません。MAP 文と ALTID を使用して、各表名を本番環境（ソース）のオブジェクト ID に手動でマップする必要があります。

### MAP と ALTID の使用方法

- 取得する表ごとに1つのMAP文とALTIDを作成します。ALTIDが含まれているMAPパラメータでは、表名のワイルドカード指定は許可されていません。
- 表に対して、データのフィルタリングや操作などの他の処理を指定するときは、TABLE 文も作成する必要があります。他の処理を指定しない場合は TABLE は必要ありません。
- Replicat パラメータ・ファイルでは、通常どおり標準の Replicat MAP 文を使用します。Extract 用 MAP は、ソース表をターゲット表にマップするために必要な Replicat 用 MAP のかわりには使用できません。
- ALTID を使用するときは、DDL 取得およびレプリケーションはサポートされません。

**デフォルト** なし

**構文** MAP <table spec>, ALTID <object ID> [, object ID>]

| コンポーネント     | 説明                                                                                               |
|-------------|--------------------------------------------------------------------------------------------------|
| <object ID> | 本番環境（ソース）データベース内での表のオブジェクト ID。<br>表がパーティション化されている場合は、レプリケートするパーティションのオブジェクト ID をコンマで区切ってリストできます。 |

**例** map QASOURCE.T2, altid 75740;

**例** map QASOURCE.T\_P1, altid 75257,75258;

## Replicat 用 MAP

**適用対象** Replicat

Replicat 用の MAP パラメータでは、1 つ以上のソースおよびターゲット・オブジェクト間の関係を確立します。Oracle GoldenGate で同期を取るすべてのターゲット・オブジェクトは、このパラメータを使用してソースにマッピングする必要があります。

**注意** MAP でマッピングするソース・オブジェクトを取得するには、Extract パラメータ・ファイルで TABLE パラメータ文を使用します。

MAP では、次のオブジェクトを指定できます。

- 表
- 索引
- 順序
- トリガー
- マテリアライズド・ビュー

### サポートの制限

表に対しては、すべての MAP オプションを使用できます。次のことを実行できます。

- 表の行を選択およびフィルタする
- 表の列をマップする
- データを変換する
- キー列を指定する
- ストアド・プロシージャおよび問合せを実行する
- 例外およびエラー処理を指定する
- 表へのすべての操作を挿入として適用する
- ユーザー・イグジットにパラメータを渡す

索引、順序、トリガーおよびマテリアライズド・ビューに対しては、MAP ではソース・オブジェクトとターゲット・オブジェクトのマップのみを行い、EXCEPTIONONLY および REPERROR オプションを使用して処理エラーを処理します。これらのオブジェクトに対しては、これ以外のどの MAP オプションも使用しないでください。

**注意** Oracle GoldenGate は、Oracle 順序およびマテリアライズド・ビューの実際のデータ値のレプリケーションをサポートしています。Oracle GoldenGate は、Oracle と Teradata 両方の索引およびトリガーの DDL レプリケーションをサポートしていますが、これらのオブジェクトのコンテンツはサポートしていません。DDL サポートの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** なし

```
構文 MAP <table spec>, TARGET <table spec>
 [, DEF <definitions template>]
 [, TARGETDEF <definitions template>]
 [, COLMAP (<column mapping expression>)]
 [, EVENTACTIONS (<action>)]
 [, EXCEPTIONSONLY]
 [, EXITPARAM "<parameter string>"]
 [, FILTER (<filter specification>)]
 [, HANDLECOLLISIONS | NOHANDLECOLLISIONS]
 [, INSERTALLRECORDS]
 [, INSERTAPPEND | NOINSERTAPPEND]
 [, KEYCOLS (<column specification>)]
 [, REPERERROR (<error> , <response>)]
 [, SQLEXEC (<SQL specification>)]
 [, TRIMSPACES | NOTRIMSPACES]
 [, WHERE (<where clause>)]
 ;
```

表 41 MAP 構文コンポーネントの概要

| コンポーネント                               | 説明                                                       |
|---------------------------------------|----------------------------------------------------------|
| MAP <table spec>                      | ソース・オブジェクトを指定します。219 ページの「オブジェクト名およびオーナー」を参照してください。      |
| TARGET <table spec>                   | ターゲット・オブジェクトを指定します。この表に続いて記載されているガイドラインを参照してください。        |
| DEF <definitions template>            | ソース定義テンプレートを指定します。                                       |
| TARGETDEF <definitions template>      | ターゲット定義テンプレートを指定します。                                     |
| COLMAP                                | 異なるソース列およびターゲット列間でレコードをマップします。                           |
| EVENTACTIONS (<action>)               | 指定されているフィルタ・ルールを満たすレコードに基づいて、アクションをトリガーします。              |
| EXCEPTIONSONLY                        | 例外 MAP 文内でエラー処理を指定します。                                   |
| EXITPARAM                             | リテラル文字列形式でパラメータをユーザー・イグジットに渡します。                         |
| FILTER                                | 数値演算子に基づいてレコードを選択します。FILTER は、WHERE よりも柔軟に使用できます。        |
| HANDLECOLLISIONS   NOHANDLECOLLISIONS | 初期ロード・プロセスでターゲット表に適用された変更の結果と、変更同期グループによって適用された変更を調整します。 |
| INSERTALLRECORDS                      | すべての行変更を挿入として適用します。                                      |
| INSERTAPPEND   NOINSERTAPPEND         | Replicat が INSERT 文で Oracle APPEND ヒントを使用するかどうかを制御します。   |

表 41 MAP 構文コンポーネントの概要（続き）

| コンポーネント                      | 説明                                                 |
|------------------------------|----------------------------------------------------|
| KEYCOLS                      | 行を一意に特定する列を指定します。                                  |
| REPERERROR                   | MAP 文の実行時に Replicat がエラーに対応する方法を制御します。             |
| SQLEXEC                      | ストアド・プロシージャおよび問合せを実行します。                           |
| TRIMSPACES  <br>NOTRIMSPACES | CHAR 列を VARCHAR 列にマッピングする際に、末尾の空白を切り捨てるかどうかを制御します。 |
| WHERE                        | 条件演算子に基づいてレコードを選択します。                              |
| ;                            | MAP 文を終了します（必須）。                                   |

## オブジェクト名およびオーナー

MySQL では、データベース名がディレクトリ名、テーブル名がファイル名に、それぞれマッピングされます。ソースおよびターゲット・オブジェクト名は、fin.emp のように、Oracle GoldenGate パラメータ・ファイルで完全に修飾される必要があります。Oracle GoldenGate では、次のように大 / 小文字がサポートされます。

## 大 / 小文字の区別

次は、Oracle GoldenGate に関連するオブジェクト名の大 / 小文字の区別に関する一般的なガイドラインです。データベースまたは基盤となるオペレーティング・システムで大 / 小文字の区別がサポートされるかにより、これらのガイドラインがデータベースに適用される場合と、されない場合があります。MySQL では、大 / 小文字が区別される表の名前は MySQL が実行されるオペレーティング・システムによって異なります。MySQL については、Windows プラットフォームでは大 / 小文字が混じった表の名前は識別されず、また、たいていの UNIX プラットフォームでは大 / 小文字が区別されます。大 / 小文字の区別（またはその区別なし）は、ソース・データベースに適用されてもターゲット・データベースには適用されない（あるいはその逆）場合があります。

- システムまたはデータベースで大 / 小文字が区別される場合、Oracle GoldenGate では、データベースの名前、オーナーとスキーマ名、オブジェクト名、列の名前およびユーザー名について、大 / 小文字の区別がサポートされます。
- システムまたはデータベースで大 / 小文字が区別されない場合（または大 / 小文字が区別されないように設定されている場合）、Oracle GoldenGate ではすべての名前が大文字に変換されます。例外は Oracle 11g で、パスワードが必要とされる Oracle GoldenGate 入力で、パスワードの大 / 小文字が区別されます。

### Oracle GoldenGate 構成で大 / 小文字の区別を保持する方法

Oracle GoldenGate パラメータ・ファイルで、大 / 小文字が区別される名前をデータベースで表示されるとおりに指定します。他のデータベース（大 / 小文字が区別されるオブジェクトのソースまたはターゲット）で大 / 小文字が区別されない場合、TABLE および MAP パラメータで、大 / 小文字が区別される名前を二重引用符で囲みます。

大 / 小文字が区別されないソースから大 / 小文字が区別されるターゲットに複製する場合、Extract ではこれらは大文字で証跡に書き込まれるので、Replicat MAP 文にソース名を大文字で入力します。

例：

```
MAP SALES.CUSTOMER, TARGET "Sales.Account";
```

**注意** 引用符で囲まれた列名は、リテラルとして扱われます。Oracle GoldenGate による列名の大 / 小文字区別への対応の詳細は、224 ページの「COLMAP の使用」を参照してください。

## サポートされる文字

Oracle GoldenGate では、オブジェクト名およびキー列とキー以外の列の名前で英数字の文字がサポートされます。また、Oracle GoldenGate では、Oracle GoldenGate でキーとして使用されない列について、次の英数字以外の文字もサポートされます。

表 42 オブジェクト名およびキー以外の列の名前でサポートされる英数字以外の文字<sup>1</sup>

| 文字  | 説明            |
|-----|---------------|
| ~   | チルド           |
| <>  | 大なりおよび小なり記号   |
| /   | フォワード・スラッシュ   |
| \   | バックスラッシュ      |
| !   | 感嘆符           |
| @   | @記号           |
| #   | シャープ記号        |
| \$  | ドル記号          |
| %   | パーセント記号       |
| ^   | 脱字記号          |
| ()  | 丸カッコ（開きおよび閉じ） |
| _   | 下線            |
| -   | ダッシュ          |
| +   | プラス記号         |
| =   | 等号            |
|     | パイプ           |
| []  | 角カッコ（開きおよび閉じ） |
| { } | 中カッコ（開きおよび閉じ） |

<sup>1</sup> Oracle GoldenGate によって使用されるキーのタイプは、指定される表の定義、および KEYCOLS 句によるオーバーライドがあるかどうかにかかわらず、Oracle GoldenGate では、主キー（ある場合）または一意キー / 索引（データベースにより異なる）が使用されます。これらの定義がない場合、表のすべての列が使用されますが、KEYCOLS 句は存在するすべてのキー・タイプよりも優先されます。Oracle GoldenGate によってキーとして使用される列の場合、名前の文字が WHERE 句への包含に有効である必要があります。このリストにはすべてが含まれていますが、データベースのプラットフォームでこれらの文字がサポートされる場合とされない場合があります。

## サポートされない文字

Oracle GoldenGate では、次の文字はオブジェクトまたは列の名前でサポートされません。

表 43 オブジェクトおよび列の名前でサポートされない文字<sup>1</sup>

| 文字  | 説明             |
|-----|----------------|
| &   | アンパサンド         |
| *   | アスタリスク         |
| ?   | 疑問符            |
| :   | コロロン           |
| ;   | セミコロロン         |
| ,   | コンマ            |
| '   | 一重引用符          |
| “ ” | 二重引用符          |
| ‘ ’ | アクセント記号（発音区別符） |
| .   | ピリオド           |
|     | 空白             |

<sup>1</sup> このリストにはすべてが含まれていますが、データベースのプラットフォームでこれらの文字がサポートされる場合とされない場合があります。

## ワイルドカードの使用

TABLE、SEQUENCE および MAP パラメータでは、1 つの文内で複数のオブジェクトを指定する場合にアスタリスクのワイルドカードを使用できます。ワイルドカードはオブジェクト名には使用できますが、スキーマまたはデータベース名には使用できません。ターゲット・オブジェクトがターゲット・データベースに存在する必要があります。

### ワイルドカードの解決方法

ワイルドカードはデフォルトで次の方法で解決されます。

**ソース・オブジェクト :** ソース・オブジェクトの名前が明示的に記述されている場合、そのオブジェクトおよびターゲット・オブジェクトの解決はプロセスの開始時に行われます。ソース・オブジェクトの名前でワイルドカードが使用されている場合、このソース・オブジェクトおよびそのターゲット・オブ

ジェットの解決は、ソース・オブジェクトの最初の行が処理されるときに行われます。

**ターゲット・オブジェクト**：ターゲット・オブジェクトでワイルドカードが使用されている場合、Oracle GoldenGate は、このワイルドカードを対応するソース・オブジェクトの名前で置き換えます。（「ワイルドカードの使用に関するルール」を参照してください。）

このデフォルトの動作により、Oracle GoldenGate による、処理の開始後に作成されたソース・オブジェクトの取得が可能になります。ワイルドカードのルールを変更するには、WILDCARDRESOLVE パラメータを使用します。デフォルト値は、DYNAMIC です。

TABLEEXCLUDE および MAPEXCLUDE パラメータを使用して、ワイルドカード・オブジェクトの選択の使用と、明示的オブジェクトの除外を組み合わせることができます。

## ワイルドカードの使用に関するルール

ワイルドカードを使用する際、次のルールが適用されます。

- アスタリスク (\*) は、任意の文字数（ゼロ・シーケンスを含む）と一致します。
- ソース・オブジェクトの場合は、名前の一部にワイルドカードを付けた形式を使用できます。たとえば、次のソース指定は有効です。

```
hq.t_*;
```

- ターゲット・オブジェクトの場合、名前の一部にワイルドカードを付けた形式は、ワイルドカードが使用されたターゲット名のアスタリスクがソース・オブジェクトの名前と置換されるので、使用できません。たとえば、次の文は無効です。

```
TABLE hq.t_*, TARGET rpt.t_*;
MAP hq.t_*, TARGET rpt.t_*;
```

これらは、ターゲット指定内のワイルドカードがソース・オブジェクトの名前である T\_TEST と置換されるので、次のような解決結果になります。

```
TABLE HQ.T_TEST1, TARGET RPT.T_T_TEST1;
MAP HQ.T_TEST1, TARGET RPT.T_T_TEST1;
```

次の例は、正しいワイルドカードの使用方法を示すものです。

```
TABLE hq.t_*, TARGET rpt.*;
MAP hq.t_*, TARGET rpt.*;
```

これにより、次のような正しい結果が得られます。

```
TABLE HQ.T_TEST1, TARGET RPT.T_TEST1;
MAP HQ.T_TEST1, TARGET RPT.T_TEST1;
```

以下に示すスキーマでのワイルドカードの使用例では、データベースに、USER1、USER2、A132B、AB、AX および XB のスキーマがあります。また、USER1.SALES、USER2.SALES、A132B.SALES、AB.SALES、AX.SALES、XB.SALES の表があります。TABLE 文は次のようになります。

```
TABLE A*B.SALES;
```

この構成では、次の表が取得されます。

```
A132B.SALES
```

```
AB.SALES
```

次の表は取得されません。

```
AX.SALES
```



XB.SALES

## MAP 文での Unicode およびネイティブ・エンコーディングの使用

Oracle GoldenGate では、Unicode、または Windows、文字列の列内の文字を UNIX および Linux オペレーティング・システムのネイティブ・キャラクタ・エンコーディングで表示するためのエスケープ・シーケンスの使用がサポートされます。エスケープ・シーケンスは、TABLE または MAP 文内の次の要素で使用されます。

- WHERE 句
- Unicode キャラクタを Unicode 列に割り当てる、またはネイティブ・エンコーディングの文字を列に割り当てる COLMAP 句
- COLMAP 句内の Oracle GoldenGate 列変換関数

Oracle GoldenGate では、次のタイプのエスケープ・シーケンスがサポートされます。

- \uFFFF Unicode エスケープ・シーケンス
- \377 8 進数エスケープ・シーケンス
- \xFF 16 進数エスケープ・シーケンス

次の制限が適用されます。

- このサポートは、7ビットASCIIに相当するU+0000～U+007Fの間のUTF-16コード・ポイントに制限されます。
- ソース列およびターゲット列の両方が Unicode である必要があります。
- ソースおよびターゲットのデータ・タイプが同一である必要があります(NCHAR から NCHAR への変換など)。

### エスケープ・シーケンスの使用手順

各エスケープ・シーケンスの先頭にリバース・ソリダスを付け（コード・ポイント U+005C）、次にキャラクタ・コード・ポイントを指定します。（リバース・ソリダスは、一般にはバックスラッシュ記号として知られています。）パラメータ文内の入力文字列または列変換関数内で、実際の文字のかわりにエスケープ・シーケンスを使用します。

#### \uFFFF Unicode エスケープ・シーケンスの使用手順

- 小文字の u を先頭に付け、4 桁の 16 進数のみを指定します。
- サポートされる範囲は次のとおりです。
  - 0 ～ 9 (U+0030 ～ U+0039)
  - A ～ F (U+0041 ～ U+0046)
  - a ～ f (U+0061 ～ U+0066)
- NCHAR および NVARCHAR 列に使用可能なエスケープ・シーケンスは、これのみです。
- サロゲート・ペアはサポートされていません。

例

\u20ac は、ユーロ通貨記号用の Unicode エスケープ・シーケンスです。

**注意** クロスプラットフォームのサポートを確実なものにするには、Unicode エスケープ・シーケンスを使用します。8 進数および 16 進数のエスケープ・シーケンスは、異種のオペレーティング・システム間では標準化されていません。

### \\377 8 進数エスケープ・シーケンスの使用手順

- 3つの8進数が含まれている必要があります。
- サポートされる範囲は次のとおりです。
  - 最初の桁の範囲が0～3（U+0030～U+0033）です。
  - 2番目と3番目の桁の範囲は0～7（U+0030～U+0037）です。

**例** \\200 は、Microsoft Windows 環境でのユーロ通貨記号用の8進数エスケープ・シーケンスです。

### \\xFF 16 進数エスケープ・シーケンスの使用手順

- 小文字のxを先頭に付け、2桁の16進数のみを指定します。
- サポートされる範囲は次のとおりです。
  - 0～9（U+0030～U+0039）
  - A～F（U+0041～U+0046）
  - a～f（U+0061～U+0066）

**例** \\x80 は、Microsoft Windows 環境でのユーロ通貨記号用の16進数エスケープ・シーケンスです。

## COLMAP の使用

COLMAP では、ソース列を異なる名前のターゲット列に明示的にマップするか、ソース名とターゲット名が同一の場合にデフォルトの列マッピングを指定します。COLMAP は、列データの選択、変換、および移動方法を提供します。

**注意** 後続の MAP 文のすべての表に適用するグローバル・ルールを作成するには、COLMATCH パラメータを使用します。

### データ定義の生成

構造が同一でないソース表とターゲット表に対して COLMAP を使用する場合は、ソース表のデータ定義を生成し、ターゲットに転送し、SOURCEDEFS パラメータを使用して定義ファイルを特定する必要があります。

ソースおよびターゲットが同一の構造とみなされるためには、同一の列名（該当する場合は大 / 小文字区別を含む）とデータ型が含まれ、かつ各表の列の順序が同一である必要があります。また、両方の表は、文字用の列の列長セマンテック（バイトまたは文字）が同一である必要があります。

両方の表が同一の構造を持ち、変換などのその他の機能のために COLMAP を使用する場合は、ソース定義は必要ありません。かわりに ASSUMETARGETDEFS パラメータを使用できます。

詳細は、次を参照してください。

- SOURCEDEFS は 304 ページ
- ASSUMETARGETDEFS は 115 ページ
- 『Oracle GoldenGate Windows and UNIX 管理者ガイド』のデータ定義ファイルの作成に関する項

### キー列への値のマッピング

COLMAP を使用してキー列に値をマップする場合（この操作は主キーの更新になります）、Oracle GoldenGate がターゲット行の特定に使用する WHERE 句では、キー列の正しいビフォア・イメージは使用されません。かわりにアフター・イメージが使用されます。そのため、SQLEXEC など、このキー列に基づくファンクションを使用するときにエラーが発生します。

次にどのような状況が起きるかを説明します。

|                 |                                |
|-----------------|--------------------------------|
| ソース表: TCUSTMER1 | 列レイアウト、両方の表:                   |
| ターゲット表:         | ◆ Column 1 = Cust              |
| TCUSTMER2       | ◆ Column 2 = Name              |
|                 | ◆ Column 3 = City              |
|                 | ◆ Column 4 = State             |
|                 | ◆ 主キーは Cust、Name、および City 列です。 |

MAP 文の SQLEXEC 文です。

```
SQLEXEC (id mytest, query "select city from TCUSTMER1 WHERE state = 'CA'",
noparams, ERROR RAISE),
```

MAP 文の COLMAP 文です。

```
COLMAP (usedefaults, city = mytest.city);
```

次に一連のイベントを示します。

#### 1. INSERT 文で次を挿入します。

```
INSERT into TCUSTMER1 values (Cust = '1234', Name = 'Ace', City = 'SF', State =
'CA');
Commit;
```

この文は成功します (SQLEXEC 問合せが mytest.city = 'SF' を返すため、ターゲット表も City の値が SF、State の値が CA になるためです)。

#### 2. UPDATE 文で、ソースの City を SF から LA に変更します。これはターゲットでは成功しません。SQLEXEC は、TCUSTMER1 で City 列を探し、値として LA を返します。COLMAP 句に基づくと、City のビフォアおよびアフター・イメージは、ともに現在 LA です。ターゲットの City 列に LA 値が存在しないため、ターゲット WHERE 句を実行するときに、SQL エラー 1430 が発生します。

### デフォルトの列マッピングの使用

ソース列と対応するターゲット列の名前が同一の場合は、明示的なマッピング文ではなく、デフォルト・マッピングを使用できます。デフォルト・マッピングでは、Oracle GoldenGate は名前が同じ列を自動的にマップします。適切な場合は、データ変換も自動的に行います。

デフォルト・マッピングを使用するには、USEDEFAULTS オプションを使用します。デフォルト・マッピングは、明示的なマッピング文でマップされていない列にのみ有効です。

Sybase および SQL Server を除くすべてのデータベースでは、列名は名前比較のために大文字に変更されます。Sybase および SQL Server では、USEDEFAULTS は次の方法で大 / 小文字区別をサポートします。

- 名前がターゲット列と正確に一致するソース列が見つかった場合、2 つの列はマップされます。
- 大 / 小文字が一致する列が見つからない場合、大 / 小文字の一致にかかわらずターゲット列と名前が一致する最初の候補のソース列を使用してマップが作成されます。

たとえば、次は大 / 小文字を区別する列を含むソース表とターゲット表です。

| ソース表 USER1.SM01 | ターゲット表 USER3.SM01 |
|-----------------|-------------------|
| id              | ID                |
| owner           | owner             |
| created         | id                |
| changed         | Creator           |
| creator         | comment           |
| modifiedBy      | ModifiedBy        |
| comment         | creationDate      |
| COMMENT         | alterationDate    |
|                 | Comment           |
|                 | COMMENT           |

次に示すこれらの表に対する列マップには、明示的およびデフォルトの列マッピングが含まれています。

```
MAP USER1.SM01, TARGET USER3.SM01,
COLMAP (USEDEFAULTS,
 ID = id,
 creationDate = created,
 alterationDate = changed,
);
```

このマップの結果を次に示します。デフォルト・マッピングでは、該当する場合には大 / 小文字の区別が識別され、それ以外の場合には名前のみが照合されます。2 つのターゲット列は、明示的にマップされておらず、デフォルト・マッピングも確立できなかったため、マップされません。

| マッピングのタイプ       | マッピングの結果                                                                                                    |
|-----------------|-------------------------------------------------------------------------------------------------------------|
| 明示的マッピング        | ID = id,<br>creationDate = created,<br>alterationDate = changed                                             |
| デフォルト・マッピング     | owner = owner,<br>comment = comment,<br>COMMENT = COMMENT,<br>Creator = creator,<br>ModifiedBy = modifiedby |
| マッピングされないターゲット列 | id, Comment                                                                                                 |

列マッピングの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

#### 構文

```
MAP <table spec>, TARGET <table spec>,
COLMAP (
[USEDEFAULTS,]
<target column> = <source expression>
[, BINARYINPUT]
[, ...]
);
```

| コンポーネント                                  | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <table spec>                             | ソースまたはターゲット表。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <target column> =<br><source expression> | <p>ソースおよびターゲット列間のマップを明示的に定義します。</p> <p>&lt;target column&gt; はターゲット列名です。列でサポートされている文字の詳細は、220 ページの「サポートされる文字」を参照してください。Unicode またはネイティブ・エンコーディングの文字列または列を使用するときに適用されるルールの詳細は、223 ページを参照してください。</p> <p>&lt;source expression&gt; には、次のいずれかを指定できます。</p> <ul style="list-style-type: none"> <li>◆ ソース列名 (例: ORD_DATE)</li> <li>◆ 数値定数 (例: 123)</li> <li>◆ 引用符で囲まれた文字列定数 (例: "ABCD")</li> <li>◆ Oracle GoldenGate 列変換ファンクションを使用する式 (例: @STREXT(COL1,1,3))。列変換ファンクションの詳細は、第 4 章を参照してください。</li> </ul> |
| BINARYINPUT                              | BINARYINPUT は、ターゲット列が RAW または BLOB などのバイナリ・データ型として定義されているが、ソース入力 of データの中にバイナリ・ゼロが含まれている場合に使用します。BINARYINPUT は、単一の列として定義されている完全な Enscribe レコードをターゲット列にレプリケートするときに使用します。ソース入力はバイナリ入力として処理され、データ値の置換は抑止されます。                                                                                                                                                                                                                                                                                      |
| USEDEFAULTS                              | 明示的な列マップで指定されていない場合に、同一の名前を持つソース列とターゲット列を自動的にマップします。1 つの列セットには、明示的マップまたは USEDEFAULTS を使用し、両方を使用しないでください。詳細は、225 ページの「デフォルトの列マッピングの使用」を参照してください。USEDEFAULTS は、明示的な列マップの前に指定します。                                                                                                                                                                                                                                                                                                                  |

- 例 1**      MAP ggs.tran, TARGET ggs.tran2, COLMAP (loc2 = loc, type2 = type);
- 例 2**      MAP ggs.tran, TARGET ggs.tran2, COLMAP ( EUROVAL = "\u20ac0" );
- 例 3**      MAP ggs.tran, TARGET ggs.tran2, COLMAP (SECTION = @STRCAT("\u00a7", SECTION ));

## DEF の使用

DEF では、ソース定義テンプレートを指定します。定義テンプレートは、特定のソース表に対して DEFGEN が実行されるときに、このオブジェクトの定義に基づいて作成されます。テンプレートが作成されると、この表と同一の定義を持つ新しいソース表は、DEFGEN を実行せずに、かつ Replicat の停止と起動を伴わずに追加できます。DEF で指定されたテンプレートの定義は、定義の参照に使用されます。DEFGEN の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文** MAP <table spec>, TARGET <table spec>, DEF <definitions template>;

| 引数                     | 説明                                                                                               |
|------------------------|--------------------------------------------------------------------------------------------------|
| <definitions template> | DEFGEN パラメータ・ファイルの TABLE の DEF オプションで指定されている定義テンプレート名。テンプレートに含まれる定義は、この MAP 文の表の定義と同一である必要があります。 |

**例** MAP acct.cust\*, TARGET acct.cust\*, DEF custdef;

## EVENTACTIONS の使用

EVENTACTIONS では、特定のフィルタリング・ルールに適合する、イベント・レコードと呼ばれるトレイル内のレコードに基づいて、Replicat に定義済のアクションを実行させます。このシステムを使用すると、データベース・イベントに基づいて処理をカスタマイズできます。

**注意** ターゲット表へのデータ適用を必要としないアクションのトリガーを目的にイベント・マーカー・システムを使用するために、Replicat 用 TABLE パラメータと EVENTACTIONS をサポートしているフィルタリング・オプションを使用できます。352 ページを参照してください。

**警告** EVENTACTIONS は、ソース・データベースが Teradata で、Extract が最大パフォーマンス・モードで構成されている場合にはサポートされません。

このシステムの使用方法的例には、プロセスの開始または停止、変換の実行、統計のレポートなどがあります。イベント・マーカー・システムは、次のような目的で使用されます。

- SQLEXEC またはユーザー・イグジット関数が実行される同期化ポイントを確立する。
- データ検証スクリプトを実行するシェル・コマンドを実行する。
- 特定のアカウント番号が検出される際の証拠を有効化する。
- ラグ履歴を取得する。
- バッチ・プロセスまたは営業後レポート・プロシージャを開始する時点を確認する。

イベント・マーカー機能は、データ変更のレプリケーションにはサポートされていますが、初期ロードにはサポートされていません。

### イベント・マーカー・システムの使用方法：

システムには、次の入力コンポーネントが必要です。

1. 操作をトリガーするイベント・レコードを指定します。これを行うには、次のいずれか 1 つのパラメータ文内に FILTER か WHERE 句、または SQLEXEC 問合せかプロシージャを含めます。
  - Extract パラメータ・ファイル内の TABLE 文
  - Replicat パラメータ・ファイル内の MAP 文

- ソース表をターゲット表にマッピングすることなく **EVENTACTIONS** 操作の実行を可能にする、**Replicat** パラメータ・ファイル内の特別な **TABLE** 文
2. イベント・レコードを指定した同じ **TABLE** または **MAP** 文で、**EVENTACTIONS** パラメータを適切なオプションとともに含めて、プロセスに適用される操作を指定します。

**注意**      すべてではありませんが、多くの **EVENTACTIONS** オプションは（**Extract** 用）**TABLE** および（**Replicat** 用）**MAP** 両方に適用されるため、ここでは両方のプロセスのすべてのオプションを説明します。一方のみに適用されるオプションには、その旨記載しています。

#### 複数の操作の組合せ方法:

- すべてではありませんが、多くの **EVENTACTIONS** オプションは組み合わせて使用できます。たいていの場合、目的を達成するのに 2 つ以上の操作を組み合わせる必要があります。
- **EVENTACTIONS** 文全体を最初に解析し、優先順位に応じて実行するオプションを指定します。次のリストでは、**Process the record** の前にリストされている操作は、レコードが証跡に書き込まれる、またはターゲットに適用される（プロセスに基づく）前に発生します。**Process the record** の後にリストされる操作は、レコードが処理された後に実行されます。
  - **TRACE**
  - **LOG**
  - **CHECKPOINT BEFORE**
  - **IGNORE**
  - **DISCARD**
  - **SHELL**
  - **ROLLOVER**
  - **(Process the record)**
  - **REPORT**
  - **ABORT**
  - **CHECKPOINT AFTER**
  - **FORCESTOP**
  - **STOP**

#### イベント・レコード自体の処理の制御方法:

イベント・レコード自体が通常の方法で処理されないようにするには、**IGNORE** または **DISCARD** オプションを使用します。**IGNORE** および **DISCARD** は、レコード自体の前に評価されるので、レコードが処理されるのを防ぐことができます。これらのオプションがない場合、**Extract** によりレコードが証跡に書き込まれ、**Replicat**により、このレコード内に含まれている操作がターゲット・データベースに適用されます。

トランザクションに、イベント操作をトリガーする 2 つ以上のレコードが含まれている可能性がある点を考慮する必要があります。そのような場合、特定の **EVENTACTIONS** の指定が複数実行されることがあります。たとえば、2 つの連続的な **ROLLOVER** 操作をトリガーする 2 つの有効なレコードを実行すると、**Extract** で証跡が 2 回ロールオーバーされ、2 つのうち 1 つが実質、空のままになります。

```
構文
EVENTACTIONS (
[STOP | ABORT | FORCESTOP]
[IGNORE [TRANSACTION [INCLUDEEVENT]]]
[DISCARD]
[LOG [INFO | WARNING]]
[REPORT]
[ROLLOVER]
[SHELL <command>]
[TRACE <trace file> [TRANSACTION] [PURGE | APPEND]]
[CHECKPOINT [BEFORE | AFTER | BOTH]]
[, ...]
)
```

| 操作    | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STOP  | <p>指定されたイベント・レコードが発生した場合に、プロセスを正常に停止します。プロセスは、停止前にオープン・トランザクションが完了するのを待ちます。トランザクションが <b>Replicat</b> によってグループ化またはバッチ化されたトランザクションである場合、プロセスが正常に停止する前に現在のグループのトランザクションが適用されます。プロセスは、レコードがトランザクションの終わりであることも意味するかぎり、イベント・レコード後の次のレコードで再開されます。</p> <p>プロセスでは、トランザクションがまだオープンであるためにすぐに停止できない場合、メッセージをログに記録します。ただし、長い間実行されているオープン・トランザクション内でイベント・レコードが発生した場合は、トランザクションの未コミット状態を警告する警告メッセージは表示されません。したがって、<b>STOP</b> イベントにかかわらず、プロセスは長期間実行されたままになることがあります。</p> <p><b>STOP</b> は、<b>ABORT</b> および <b>FORCESTOP</b> 以外の他の <b>EVENTACTIONS</b> オプションと組み合わせることができます。</p> |
| ABORT | <p>指定されたイベント・レコードが発生した場合、オープン・トランザクションがある、ないにかかわらず、プロセスがただちに終了されます。イベント・レコードは処理されません。致命的なエラーは、ログに書き込まれ、<b>DISCARD</b> も指定されている場合は、イベント・レコードは破棄ファイルに書き込まれます。プロセスについては、起動時にリカバリが実行されます。</p> <p><b>ABORT</b> は、<b>CHECKPOINT BEFORE</b>、<b>DISCARD</b>、<b>SHELL</b> および <b>REPORT</b> とのみ、組み合わせることができます。</p>                                                                                                                                                                                                                                                      |



| 操作                                             | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>FORCESTOP</p>                               | <p>指定されたイベント・レコードが発生すると、イベント・レコードが、トランザクションで最後の操作である場合、またはトランザクション内の唯一のレコードである場合にのみ、プロセスが正常に停止されます。レコードは通常どおり書き込まれます。</p> <p>イベント・レコードが、長い間実行されているオープン・トランザクション内で発生した場合、ABORT で指定されるように、プロセスはログに警告メッセージを書き込み、すぐに終了します。この場合、起動時のリカバリが必要になる場合があります。FORCESTOP 操作が、長い間事項されているトランザクションの途中でトリガーされた場合、プロセスは警告メッセージなしに終了します。</p> <p>FORCESTOP は、ABORT、STOP、CHECKPOINT AFTER および CHECKPOINT BOTH 以外の他の EVENTACTIONS オプションと組み合わせることができます。ROLLOVER とともに使用される際、プロセスが正常に停止する場合にロールオーバーのみが発生します。</p>                                                            |
| <p>IGNORE<br/>[TRANSACTION [INCLUDEEVENT]]</p> | <p>デフォルトでは、プロセスで指定されたイベント・レコードが無視されます。いずれの警告またはメッセージもログに書き込まれませんが、Oracle GoldenGate 統計が更新され、レコードが無視されたことが示されます。</p> <ul style="list-style-type: none"> <li>◆ TRANSACTION を使用して、イベントをトリガーしたレコードが含まれるトランザクション全体を無視できます。TRANSACTION が使用される場合、イベント・レコードはトランザクション内で最初の者である必要があります。トランザクションを無視する際、デフォルトでイベント・レコードも無視されます。TRANSACTION は、TRANS に短縮できます。</li> <li>◆ INCLUDEEVENT を TRANSACTION とともに使用して、イベント・レコードを証跡またはターゲットに伝播することができますが、残りの関連するトランザクションは無視されます。</li> </ul> <p>IGNORE は、ABORT および DISCARD 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p> |
| <p>DISCARD</p>                                 | <p>プロセスで次の処理が行われます。</p> <ul style="list-style-type: none"> <li>◆ 指定されたイベント・レコードを破棄ファイルに書き込む。</li> <li>◆ Oracle GoldenGate 統計を更新して、レコードが破棄されたことを示す。</li> </ul> <p>プロセスは、証跡の次のレコードから処理を再開します。このオプションを使用した場合、DISCARDFILE パラメータを使用して破棄ファイルの名前を指定できます。デフォルトでは、破棄ファイルは作成されません。</p> <p>DISCARD は、IGNORE 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p>                                                                                                                                                                                                  |

| 操作                   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LOG [INFO   WARNING] | <p>指定されたイベント・レコードが発生した場合に、プロセスによってイベントがログに記録されます。メッセージはレポート・ファイル、Oracle GoldenGate エラー・ログおよびシステム・イベント・ログに書き込まれます。</p> <p>次のオプションを使用して、メッセージの重大度を指定します。</p> <ul style="list-style-type: none"> <li>◆ INFO では、重大度の低い情報メッセージを指定します。これはデフォルトです。</li> <li>◆ WARNING では、重大度の高い警告メッセージを指定します。</li> </ul> <p>LOG は、ABORT 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。ABORT を使用する場合、プロセスが終了される前に ABORT によってログに致命的エラーが記録されるので、LOG は必要ありません。</p>                                                                                                                                                       |
| REPORT               | <p>指定されたイベント・レコードが発生した場合に、プロセスでレポート・ファイルが生成されます。これは、GGSCI で SEND コマンドを REPORT オプションとともに使用した場合と同じです。</p> <p>REPORT メッセージは、イベント・レコードが処理された後に発生するので (DISCARD、IGNORE または ABORT が使用されていないかぎり)、レポート・データにイベント・レコードが含まれます。</p> <p>REPOR は、他のすべての EVENTACTIONS オプションと組み合わせることができます。</p>                                                                                                                                                                                                                                                                                                             |
| ROLLOVER             | <p>Extract にのみ有効です。指定されたイベント・レコードが発生した場合に、Extract により証跡ファイルが新しいファイルにロール・オーバーされます。ROLLOVER 操作は、Extract によるイベント・レコードの証跡ファイルへの書込み前に発生するので、DISCARD、IGNORE または ABORT も使用されていないかぎり、レコードは新しいファイル内で最初のものになります。</p> <p>ROLLOVE は、ABORT 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>注意：</p> <p>次の理由により、ROLLOVER を ABORT と組み合わせることはできません。</p> <ul style="list-style-type: none"> <li>◆ ROLLOVER では、プロセスによるチェックポイントへの書込みはできません。</li> <li>◆ ROLLOVER は、ABORT の前に発生する。</li> </ul> <p>ROLLOVER チェックポイントがない場合、ABORT により、Extract は再起動時に、以前の証跡ファイル内にある以前のチェックポイントを確認しようとします。この結果、ロールオーバーが取り消されます。</p> |

| 操作                                                                     | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SHELL <command>                                                        | <p>指定されたイベント・レコードが発生した場合に、プロセスによって指定されたシェル・コマンドが実行されます。</p> <ul style="list-style-type: none"> <li>◆ &lt;command&gt; には、実行されるシステムまたはシェル・コマンドを指定します。</li> <li>◆ シェル・コマンドが成功すると、プロセスでは情報メッセージがレポート・ファイルおよびイベント・ログに書き込まれます。成功したかどうかは、UNIX シェル言語に準拠するコマンドのイグジット状態に基づきます。この言語では、ゼロは成功を示します。</li> <li>◆ システム呼出しに失敗すると、プロセスは致命的エラーを出して異常終了します。UNIX シェル言語では、ゼロ以外は失敗です。</li> </ul> <p>SHELL は、他のすべての EVENTACTIONS オプションと組み合わせることができます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <p>TRACE &lt;trace file&gt;<br/>[TRANSACTION]<br/>[PURGE   APPEND]</p> | <p>指定されたイベント・レコードが発生した場合に、プロセスにより証跡情報が証跡ファイルに書き込まれます。</p> <p>デフォルトでは、証跡はプロセスが終了されるまで有効になります。証跡レベルを設定するには、Oracle GoldenGate TRACE または TRACE2 パラメータを使用します。</p> <ul style="list-style-type: none"> <li>◆ &lt;trace file&gt; には、証跡ファイルの名前を指定し、TRACE キーワード後すぐに表示される必要があります。一意の証跡ファイルを指定するか、またはスタンドアロンの TRACE または TRACE2 パラメータで指定されるデフォルトの証跡ファイルを使用できます。</li> </ul> <p>EVENTACTIONS TRACE が使用されている異なる TABLE または MAP 文で、同じ証跡ファイルを使用できます。複数の TABLE または MAP 文で同じ証跡ファイル名が指定されているが、TRACE オプションが統一して使用されていない場合、この証跡ファイルが含まれている最後に解決された TABLE または MAP 内のオプションが優先されます。</p> <ul style="list-style-type: none"> <li>◆ プロセスが終了されるときではなく、現在のトランザクションの終わりまで証跡を有効化するには、TRANSACTION を使用します。Replicat の場合、トランザクション境界は、通常の Replicat によるグループ化またはバッチ化されたターゲット・トランザクションではなく、ソース・トランザクションに基づきます。TRANSACTION は、TRANS に短縮できます。</li> <li>◆ PURGE を使用して、追加の証跡レコードの書き込み前に証跡ファイルの切捨てを行うか、または、APPEND を使用して、既存のレコードの終わりに新しい証跡レコードを書き込みます。APPEND がデフォルトです。</li> </ul> <p>TRACE は、ABORT 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>指定された証跡ファイルへの証跡を無効化するには、GGSCI の SEND &lt;process&gt; コマンドを TRACE OFF &lt;filename&gt; オプションとともに実行します。</p> |

| 操作                                    | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHECKPOINT<br>[BEFORE   AFTER   BOTH] | <p>指定されたイベント・レコードが発生した場合に、プロセスでチェックポイントが書き込まれます。チェックポイント操作により、TABLE または MAP 文で定義されている処理に関するコンテキストが提供されます。このコンテキストとは開始ポイントおよび終了ポイントであるので、SQLEXEC およびユーザー・イグジットで実行される関数のマッピング用の同期化ポイントが与えられます。</p> <p>◆ BEFORE</p> <p>Extract の場合、BEFORE では、Extract による証跡へのイベント・レコードの書き込み前にチェックポイントの書き込みがプロセスによって行われます。</p> <p>Replicat の場合、BEFORE では、Replicat によってレコード内に含まれている SQL 操作のターゲットへの適用前に、チェックポイントの書き込みがプロセスによって行われます。</p> <p>BEFORE では、イベント・レコードがトランザクション内の最初のレコードである必要があります。最初のレコードでない場合、プロセスは異常終了します。BEFORE を使用して、これより前のすべてのトランザクションが、イベント・レコードがコミットされた時点で開始されるように設定できます。</p> <p>CHECKPOINT BEFORE は、すべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>◆ AFTER</p> <p>Extract の場合、AFTER では、Extract による証跡へのイベント・レコードの書き込み後にチェックポイントの書き込みがプロセスによって行われます。</p> <p>Replicat の場合、AFTER では、Replicat によってレコード内に含まれている SQL 操作のターゲットへの適用後に、チェックポイントの書き込みがプロセスによって行われます。</p> <p>AFTER では、チェックポイント・リクエストが勧告としてフラグされます。つまり、プロセスで、次の実質的な機会があった場合にのみチェックポイントが実行されます。たとえば、イベント・レコードが複数のレコードのトランザクションの 1 つである場合、チェックポイントは、Oracle GoldenGate データ整合性モデルに従って、次のトランザクション境界で発生します。</p> <p>CHECKPOINT AFTER は、ABORT 以外のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>◆ BOTH</p> <p>BOTH は、BEFORE と AFTER を組み合わせたものです。Extract または Replicat プロセスでは、イベント・レコードが処理される前後にチェックポイントの書き込みが行われます。</p> <p>CHECKPOINT BOTH は、ABORT 以外のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>CHECKPOINT は、CP に短縮できます。</p> |

**例 1** 次に、特定のレコードを無視するようにプロセスを構成する方法の例を示します。Replicat は、`name=goldengate` を持つトレイル・レコードを処理するときに、このレコードを無視します。

```
MAP <owner.table>, TARGET <owner2.table2>, &
WHERE (name = "goldengate"), &
EVENTACTIONS (ignore);
```

**例 2** EVENTACTIONS オプションの互換性および優先度ルールでは、DISCARD は ABORT よりも優先されるため、この例では、イベント・レコードはプロセスが異常終了する前に破棄ファイルに書き込まれます。

```
MAP <owner.table>, TARGET <owner2.table2>, &
WHERE (name = "goldengate"), &
EVENTACTIONS (DISCARD, ABORT);
```

**例 3** 次に、トレースの異なる構成方法の例を示します。

```
MAP tab1, TARGET tab1 EVENTACTIONS (TRACE ./dirrpt/trace1.txt);
MAP tab2, TARGET tab2 EVENTACTIONS (TRACE ./dirrpt/trace2.txt TRANSACTION);
```

- 最初の MAP 文では、最初の tab1 イベント・レコードがターゲットに適用される直前に trace1.txt トレース・ファイルが生成されます。これには、この時点から Replicat が終了するまで、または GGSCI の SENDREPLICAT コマンドでトレースが無効化されないかぎり、すべてのトレース情報が含まれます。
- 2 つ目の MAP 文には、TRANSACTION オプションが含まれているため、trace2.txt ファイルは最初の tab2 イベント・レコードがターゲットに適用される直前に作成されますが、トレースは tab2 イベント・レコードを含むトランザクションの終了時に自動的に停止します。

イベント・マーカー・システムの他の使用例および詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

## EXCEPTIONONLY の使用

EXCEPTIONONLY は、エラーを処理するための例外 MAP 文で使います。例外 MAP 文は、エラーが発生する可能性がある MAP 文に続ける必要があります。例外 MAP 文は、先行する標準の MAP 文で処理された最後のレコードでエラーが発生した場合にのみ実行されます。

EXCEPTIONONLY を使用するには、標準の MAP 文内、またはパラメータ・ファイルのルートで、REPEROR 文と EXCEPTION オプションを使います。REPEROR の詳細は、274 ページを参照してください。

例外 MAP 文では、標準の MAP 文と同じ SOURCE 表を指定する必要がありますが、例外 MAP 文の TARGET 表は例外表である必要があります。例外 MAP 文を使用する場合は、標準の MAP 文でワイルドカード・オブジェクト名を使用しないでください。

**注意** オブジェクト名のワイルドカードをサポートするには、MAPEXCEPTION オプションを参照してください。

例外 MAP 文の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文** MAP <table spec>, TARGET <table spec>, EXCEPTIONONLY

## EXITPARAM の使用

EXITPARAM では、MAP 文からレコードを検出するたびにユーザー・イグジット・ルーチンにパラメータを渡します。ユーザー・イグジットの詳細は、第 5 章を参照してください。

**構文**

```
MAP <table spec>, TARGET <table spec>,
EXITPARAM "<parameter string>";
```

| コンポーネント              | 説明                                                            |
|----------------------|---------------------------------------------------------------|
| "<parameter string>" | リテラル文字列のパラメータ。パラメータは二重引用符で囲みます。パラメータ文字列では、最大で 100 文字まで指定できます。 |

## FILTER の使用

FILTER では、数値に基づいてレコードを選択または除外します。フィルタ式では、条件演算子、Oracle GoldenGate 列変換ファンクション、またはその両方を使用できます。

**注意** 文字列に基づいてフィルタするには、文字列ファンクションを使用するか、WHERE オプションを使用します。

FILTER コンポーネントはすべてコンマで区切ります。FILTER 句には、次を含めることができます。

- 数字
  - 数字を含む列
  - 数字を返すファンクション
  - 算術演算子：
    - + (加算)
    - (減算)
    - \* (乗算)
    - / (除算)
    - \ (余り)
  - 比較演算子：
    - > (より大きい)
    - >= (以上)
    - < (より少ない)
    - <= (以下)
    - = (等しい)
    - <> (等しくない)
- 比較から導出した結果はゼロ (FALSE を示す) またはゼロ以外 (TRUE を示す) になります。
- カッコ (式の結果をグループ化)
  - 結合演算子: AND、OR

**構文**

```
MAP <table spec>, TARGET <table spec> , FILTER (
[, ON INSERT | ON UPDATE | ON DELETE]
[, IGNORE INSERT | IGNORE UPDATE | IGNORE DELETE]
, <filter clause>
[, RAISEERROR <error>]
);
```

| コンポーネント                                          | 説明                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <filter clause>                                  | <p>次のように、式に基づいてレコードを選択します。</p> <pre>FILTER ((PRODUCT_PRICE*PRODUCT_AMOUNT)&gt;10000)</pre> <p>テキスト文字列は二重引用符で囲みます (例: <code>FILTER (@STRFIND(NAME, "JOE") &gt; 0)</code>)。</p> <p>フィルタ句では、次の例のように、Oracle GoldenGate の列変換ファンクションを使用できます。</p> <pre>FILTER (@COMPUTE (PRODUCT_PRICE*PRODUCT_AMOUNT)&gt;10000)</pre> |
| ON INSERT  <br>ON UPDATE  <br>ON DELETE          | <p>レコードのフィルタリングを、指定した操作に限定します。次のように、操作はコンマで区切ります。</p> <pre>FILTER (ON UPDATE, ON DELETE,<br/>@COMPUTE (PRODUCT_PRICE*PRODUCT_AMOUNT)&gt;10000)</pre> <p>この例では、更新および削除に対してフィルタを実行しますが、挿入に対しては実行しません。</p>                                                                                                         |
| IGNORE INSERT   IGNORE<br>UPDATE   IGNORE DELETE | <p>指定した操作にフィルタを適用しません。次のように、操作はコンマで区切ります。</p> <pre>FILTER (IGNORE INSERT, @COMPUTE<br/>(PRODUCT_PRICE*PRODUCT_AMOUNT)&gt;10000)</pre> <p>この例では、更新および削除に対してフィルタを実行しますが、挿入は無視します。</p>                                                                                                                             |
| RAISEERROR <error>                               | <p>フィルタが失敗したときにユーザー定義のエラー番号を発行します。REPEROR パラメータの入力として使用し、エラー処理を起動できます。&lt;error&gt; は、データベースまたは Oracle GoldenGate によって使用されているエラー番号範囲外の値にしてください。(例: <code>RAISEERROR 21000</code>)。</p>                                                                                                                          |

## HANDLECOLLISIONS | NOHANDLECOLLISIONS の使用

HANDLECOLLISIONS および NOHANDLECOLLISIONS では、ソース表のターゲット表への初期ロードが行われている間に、Oracle GoldenGate によってレプリケートされた同じソース表へのトランザクション変更を Oracle GoldenGate が調整するかどうかを制御します。HANDLECOLLISIONS は、初期ロード方法を使用し、ソース表がオンライン状態のままでユーザーによってデータが変更されている場合に必要です。HANDLECOLLISIONS を使用すると、初期ロードが終了して Oracle GoldenGate がレプリケートされた変更を適用するときに、Replicat はターゲット表の重複レコードを上書き、行方不明レコードのエラーには別のエラー処理方法で対処します。

HANDLECOLLISIONS および NOHANDLECOLLISIONS は、パラメータ・ファイルでグローバルに使用することも、表グループに対する個別の有効化 / 無効化スイッチとして使用することもできます。MAP 文で使用される場合は、他の場所の指定よりも優先されます。HANDLECOLLISIONS の詳細は、204 ページを参照してください。

|     |                                                                                                            |
|-----|------------------------------------------------------------------------------------------------------------|
| 構文  | <pre>MAP &lt;table spec&gt;, TARGET &lt;table spec&gt;,<br/>[HANDLECOLLISIONS   NOHANDLECOLLISIONS];</pre> |
| 例 1 | <p>次に、MAP 文内での基本的な使用例を示します。</p> <pre>MAP dbo.tcust, TARGET dbo.tcust, HANDLECOLLISIONS;</pre>              |

**例 2** 次に、グローバルな使用とパラメータ・ファイル内での MAP 固有の使用を組み合わせた例を示します。MAP の指定はグローバルな指定よりも優先されるため、tcust 表のペアに対しては衝突は処理されません。

```
REPLICAT fin
USERID ggs, PASSWORD ggs
HANDLECOLLISIONS
ASSUMETARGETDEFS
MAP dbo.torders, TARGET dbo.torders;
MAP dbo.tprod, TARGET dbo.tprod;
MAP dbo.tcust, TARGET dbo.tcust, NOHANDLECOLLISIONS;
```

## INSERTALLRECORDS の使用

現在のバージョンのみでなく、すべての操作のレコードがターゲット・レコードに保持されるようにするには、INSERTALLRECORDS パラメータを使用します。INSERTALLRECORDS により、Replicat であらゆる変更操作に関するレコードが新しいレコードとしてデータベースに挿入されるようになります。初期の挿入、およびその後の更新や削除は、ある時点のスナップショットとして保持されます。

履歴データと特別なトランザクション情報の組み合わせにより、さらに便利なターゲット・レポート・データベースを作成できます。トランザクション履歴表の詳細は、『*Oracle GoldenGate Windows and UNIX 管理者ガイド*』を参照してください。

INSERTALLRECORDS は、複数の MAP 文に同時に適用するために、パラメータ・ファイルのルート・レベルで単独パラメータとしても使用できます。208 ページを参照してください。

### 構文

```
MAP <table spec>, TARGET <table spec>, INSERTALLRECORDS;
```

## INSERTAPPEND | NOINSERTAPPEND の使用

Replicat で Oracle ターゲット表に INSERT が適用される際に APPEND ヒントが使用されるかどうかを制御するには、INSERTAPPEND および NOINSERTAPPEND パラメータを使用します。これらのパラメータは、Oracle データベースにのみ有効です。

これらのパラメータは、2 つの方法で使用できます。パラメータ・ファイルのルートでスタンドアロン・パラメータとして使用される場合、1 つはその後のすべての TABLE または MAP 文に対して、もう 1 つが発生するまでそのまま有効になります。MAP 文内で使用される場合、MAP 文の前になるスタンドアロンの INSERTAPPEND または NOINSERTAPPEND エントリよりも優先されます。

INSERTAPPEND により、INSERT 操作が Oracle ターゲット表に適用される際に Replicat で APPEND\_VALUES ヒントが使用されるようになります。複製されたトランザクションが大きく、同じ表内に複数の挿入が含まれる際のパフォーマンス向上のために使用されるのに適しています。トランザクションが小さい場合、INSERTAPPEND を使用するとパフォーマンスが低下することがあります。APPEND ヒントを使用する時期の詳細は、Oracle ドキュメントを参照してください。

INSERTAPPEND を使用する際には、BATCHSQL パラメータを使用する必要があります。BATCHSQL が使用されない場合、Replicat は異常終了します。

Replicat 用 INSERT 文のデフォルトは、NOINSERTAPPEND です。

単独で使用する場合の構文および例は、207 ページの INSERTAPPEND | NOINSERTAPPEND の項を参照してください。



**構文** MAP <table spec>, TARGET <table spec>, [INSERTAPPEND | NOINSERTAPPEND];

**例** 次では、inventory 表を除き、MAP 文のすべての表で INSERTAPPEND が使用されます。

```
INSERTAPPEND
MAP fin.orders, TARGET fin.orders;
MAP fin.inventory, TARGET fin.inventory, NOINSERTAPPEND;
MAP fin.customers, TARGET fin.customers;
```

## KEYCOLS の使用

KEYCOLS では、ターゲット表の 1 つ以上の列を一意列として定義します。主に KEYCOLS は、ターゲット表で主キーまたは一意索引が使用できないときに、代替主キーを定義するために使用します。

ソースおよびターゲットのキー列または一意索引列は、データベースで定義されている場合も、KEYCOLS によって代替キーが指定されている場合でも、一致する必要があります。ソース表には、少なくともターゲット表と同じ数のキー列または索引列が含まれている必要があります。そうでなければ、ソースのキー列または索引列を更新する際に、Replicat は余剰なターゲット列のピフォア・イメージを取得できません。

キーを定義する際は、次のガイドラインに従ってください。

- ソース表とターゲット表両方にキーまたは一意索引が含まれていない場合は、TABLE および MAP 文両方で KEYCOLS を使用し、一致する列セットを指定します。
- いずれか一方の表にキーまたは一意索引が含まれていない場合は、その表に対して KEYCOLS を使用し、もう一方の表の実際のキーまたは索引列に一致する列を指定します。一致する列セットを定義できない場合は、TABLE および MAP 文両方で KEYCOLS を使用し、一意の値が含まれる一致する列セットを指定します。KEYCOLS の指定は、既存のキーまたは索引よりも優先されます。
- ターゲット表にソース表よりも大きなキー（または多くの一意索引列）が含まれている場合は、TABLE 文で KEYCOLS を使用し、実際のソースのキーまたは索引列に加え、余剰なターゲット列と一致するソース列を指定する必要があります。表に主キーまたは一意索引が含まれている場合、KEYCOLS の指定はこれらよりも優先されるため、余剰な列のみを指定しないでください。このように KEYCOLS を使用すると、キーまたは索引列は更新のときにピフォア・イメージを利用できます。

KEYCOLS を使用するときは、指定した列をトランザクション・ログ に記録し、Replicat がトレイルで使用できるようにしてください。この設定は、データベース・インタフェースを使用するか、ADD TRANDATA コマンドの COLS オプションを使用して行えます（Oracle ログベース抽出のみ）。

ターゲット表では、KEYCOLS で定義したキー列に一意索引を作成します。索引によって、Oracle GoldenGate が処理する必要があるターゲット行をより高速に特定できます。

**構文** MAP <table spec>, TARGET <table spec>, KEYCOLS (<column> [, ... ]);

| コンポーネント    | 説明                                                                                                                                     |
|------------|----------------------------------------------------------------------------------------------------------------------------------------|
| (<column>) | 代替主キーとして使用する列を定義します。複数の列を指定するには、次のようにコンマ区切りリストを作成します。<br><br>KEYCOLS (id, name<br><br>主キーまたは一意索引が存在する場合は、これらの列を KEYCOLS 指定に含める必要があります。 |

## MAPEXCEPTION の使用

MAPEXCEPTION では、REPEROR パラメータによって例外としてフラグ付けされている操作に対するマッピングおよびその他のオプションを指定します。MAPEXCEPTION によって、Replicat が失敗した操作を書き込み可能な例外表を指定し、マッピングおよび処理オプションを許可します。

MAPEXCEPTION は、ソース・ターゲット表マッピングおよびその他の標準の MAP オプションを含む MAP 文内で使用できます。ソースおよびターゲット表名には、ワイルドカードを含めることができます。

MAPEXCEPTION を使用するには、同一の MAP 文内、または Replicat パラメータ・ファイルのルートで、REPEROR 文と EXCEPTION オプションを使用します。

### 構文

```
MAP <object spec>, TARGET <object spec>,
[<MAP_options>],
MAPEXCEPTION (TARGET <object spec> [, <exception_MAP_options>]);
```

| 引数                      | 説明                                                                                                  |
|-------------------------|-----------------------------------------------------------------------------------------------------|
| TARGET                  | 必須のキーワード。                                                                                           |
| <MAP_options>           | 必要な場合、(成功した) 非例外操作を処理する標準の MAP オプション。                                                               |
| <object spec>           | 例外表の完全修飾名。オブジェクト名には標準の Oracle GoldenGate ルールが適用されます。このドキュメントの MAP に関するガイドラインを参照してください。              |
| <exception_MAP_options> | MAP パラメータの有効な任意のオプション。エラー処理のためにこれらのオプションが適用されます。非例外データをマップする場合は、MAPEXCEPTION 句の外でも MAP オプションを使用します。 |

### 例

これは、例外のマッピング用の MAPEXCEPTION の使用例です。MAP および TARGET 句には、ワイルドカード付きのソースおよびターゲット表の名前が含まれています。TRX で始まる名前の表を処理すると発生する例外は、専用のマッピングを使用して fin.trxexceptions 表に取得されます。

```
MAP src.trx*, TARGET trg.*,
MAPEXCEPTION (TARGET fin.trxexceptions,
COLMAP (USEDEFAULTS,
ACCT_NO = ACCT_NO,
OPTYPE = @GETENV ("LASTERR", "OPTYPE"),
DBERR = @GETENV ("LASTERR", "DBERRNUM"),
DBERRMSG = @GETENV ("LASTERR", "DBERRMSG")
)
);
```

## REPERROR の使用

REPERROR では、エラーおよびレスポンスを指定し、MAP 文を実行したときに Replicat がエラーに対応する方法を制御します。REPERROR を MAP レベルで使用すると、REPERROR パラメータで設定されているグローバルなエラー処理ルール・セットよりも優先され、このルールを補完できます（274 ページを参照してください）。自動化された包括的なエラー管理、および無停止のレプリケーション処理を行うために、複数の REPERROR 文を同一の MAP 文に適用できます。

### 構文

```
MAP <object spec>, TARGET <object spec>,
REPERROR (
{DEFAULT | DEFAULT2 | <SQL error> | <user-defined error>},
{ABEND | DISCARD | EXCEPTION | IGNORE |
RETRYOP [MAXRETRIES <n>] |
TRANSABORT [, MAXRETRIES] [, DELAYSECS <n> | DELAYCSECS <n>]}
)
[, ...];
```

| 引数                             | 説明                                                                                                                                      |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>処理するエラーを指定するオプション：</b>      |                                                                                                                                         |
| DEFAULT                        | 明示的な REPERROR 文が指定されているエラーを除くすべてのエラーに対するグローバルなレスポンスを設定します。                                                                              |
| DEFAULT2                       | DEFAULT のレスポンスが EXCEPTION に設定されている場合に、バックアップのデフォルト・アクションを提供します。DEFAULT2 は、エラーの発生が予想される MAP 文に例外 MAP 文が指定されていないときに使用します。                 |
| <SQL error>                    | SQL エラー番号。                                                                                                                              |
| <user-defined error>           | MAP 文の FILTER 句の RAISEERROR オプションで指定されているユーザー定義エラー。                                                                                     |
| <b>エラーに対するレスポンスを指定するオプション：</b> |                                                                                                                                         |
| ABEND                          | トランザクションをロールバックし、処理を異常終了します。ABEND はデフォルトです。                                                                                             |
| DISCARD                        | エラーを破棄ファイルに記録しますが、このトランザクションおよび後続のトランザクションの処理を継続します。                                                                                    |
| EXCEPTION                      | データを例外として処理し、正常な処理を続けます。エラーの発生が予想される場合は、このオプションとともに例外 MAP 文を使用できます。エラー処理の構成方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。 |
| IGNORE                         | エラーを無視します。                                                                                                                              |

| 引数                                                                   | 説明                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RETRYOP [MAXRETRIES <n>]                                             | <p>操作を再試行します。MAXRETRIES オプションでは、再試行回数を制御します。たとえば、表がエクステンント不足の場合、RETRYOP と MAXRETRIES によって、トランザクションの失敗を防ぐためにエクステンントを追加する時間を確保できます。指定した MAXRETRIES 回数の後、Replicat は異常終了します。</p> <p>試行間隔を設定するには、284 ページで説明されている RETRYDELAY を設定します。</p> |
| TRANSABORT<br>[, MAXRETRIES]<br>[, DELAYSECS <n>  <br>DELAYSECS <n>] | <p>トランザクションを中止し、Replicat をトランザクションの開始位置に再配置します。この動作は、操作の処理が成功するか、MAXRETRIES が終了するまで継続されます。MAXRETRIES が設定されていない場合、TRANSABORT アクションは何度も繰り返されます。</p> <p>DELAY オプションでは、再試行までの待機時間を指定します。</p>                                           |

**例** 次に、グローバルな REPERROR 文とともに、MAP 文で REPERROR を使用する様々な例を示します。

**例 1:**

```
REPLICAT <group>
REPERROR (<error1> , <response1>)
MAP <src1>, TARGET <tgt1>, REPERROR (<error1>, <response2>);
MAP <src2>, TARGET <tgt2>, REPERROR (<error2>, <response3>);
```

上記の例では、最初の MAP 文で error1 が発生した場合、優先される文が指定されているため、アクションは response1 ではなく response2 になります。ただし、2 番目の MAP 文で error1 が発生した場合には、グローバル・レスポンスの response1 がレスポンスになります。error2 に対するレスポンスは、MAP 固有の response3 になります。

**例 2:**

```
REPLICAT <group>
REPERROR (<error1> , <response1>)
MAP <src1>, TARGET <tgt1>, REPERROR (<error2>, <response2>),
REPERROR (<error3>, <response3>);
```

上記の例では、すべての REPERROR 文が異なるエラーに対応するため (MAP 固有の優先なし)、src1 から src2 にレプリケートするときに、すべてのエラーおよびアクション (1 ~ 3) が適用されます。

**例 3:**

```
REPLICAT <group>
REPERROR (<error1> , <response1>)
MAP <src1>, TARGET <tgt1>, REPERROR (<error1>, <response2>);
MAP <src2>, TARGET <tgt2>, REPERROR (<error2>, <response3>);

REPERROR (<error1> , <response4>)
MAP <src2>, TARGET <tgt2>, REPERROR (<error3>, <response3>);
```

上記の例では、最初の MAP 文で error1 が発生した場合、アクションは response2 になります。2 番目の文ではアクションは response1 (グローバル・レスポンス) になり、3 番目の文では response4 になります (2 番目の REPERROR 文が適用されるため)。グローバル REPERROR 文は、パラメータ・ファイル内で別の REPERROR 文で新しいルールが開始されるまで、後続のすべての MAP 文に適用されます。

## SQLEXEC の使用

SQLEXEC では、Oracle GoldenGate 処理中に MAP 文内から SQL ストアド・プロシージャまたは問合せを実行します。SQLEXEC により、Oracle GoldenGate はデータベースと直接通信し、データベースによってサポートされているファンクションを実行できます。データベース・ファンクションは、列変換のための値取得などの同期プロセスの一部として使用することも、データの抽出またはレプリケーションと関係なく使用することもできます。

MAP 文内で使用する場合、実行されるプロシージャまたは問合せは、ソースまたはターゲット行から入力パラメータを受け付け、出力パラメータを渡すことができます。

**警告**                      問合せまたはプロシージャは、SQLEXEC 文の実行時に正しく構築されている必要があります。Replicat は、問合せまたはプロシージャの問題を検出すると、設定されているエラー処理ルールにかかわらず、即座に異常終了します。

## サポートされているデータベースとデータ型

以下は、SQLEXEC によってサポートされるデータベース、および入力と出力パラメータにサポートされるデータ・タイプです。

### DB2 LUW および z/OS

- CHAR
- VARCHAR
- DATE
- すべての数値データ・タイプ
- BLOB データ・タイプ

### Ingres

LOB 以外のすべてのデータ・タイプ

### MySQL

TEXT および BLOB 以外のすべてのデータ・タイプ

### Oracle

次のものを除く、すべての Oracle タイプがサポートされます。

- BFILE
- BLOB
- CFILE
- CLOB
- NCLOB
- NTY

### SQL Server

- CHAR
- VARCHAR
- DATETIME
- すべての数値データ・タイプ
- 長さが 200 バイト未満のイメージおよびテキスト・データ・タイプ
- TIMESTAMP    パラメータ・タイプはネイティブにサポートされませんが、他のデータ・タイプをパラメータとして使用し、ストアド・プロシージャ内の TIMESTAMP 形式に値を変換できます。

## Sybase

TEXTIMAGE および UDT 以外のすべてのデータ・タイプ

## Teradata

Oracle GoldenGate でサポートされる、すべての Teradata データ・タイプ

Oracle GoldenGate でのストアド・プロシージャおよび問合せの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

## SQLEXEC の依存関係および制約事項

- SQL は、Oracle GoldenGate プロセスを実行しているデータベース・ユーザーによって実行されます。このユーザーは、ストアド・プロシージャの実行、およびデータベース提供のプロシージャのコール権限を持っている必要があります。
- 問合せまたはプロシージャは、SQLEXEC 文の実行時に、データベースの有効な構文を使用して正しく構築されている必要があり、そうでない場合、Replicat は、設定されているエラー処理ルールにかかわらず異常終了します。許可されている SQL 構文の詳細は、データベース・ベンダーによって提供されている SQL リファレンス・ガイドを参照してください。
- SQLEXEC は、主キー列の値を変更するために使用しないでください。主キーの値は、Extract から Replicat に渡されます。主キーの値がない場合、Replicat 操作は完了できません。主キーの値を SQLEXEC で変更する必要がある場合は、元のキーの値を別の列にマッピングした後、KEYCOLS オプションでこの列を代替キーとして定義することにより、エラーを回避できます。239 ページの「KEYCOLS の使用」を参照してください。
- z/OS 上の DB2 に対して、Oracle GoldenGate は、ODBC SQLExecDirect フังก์ションを使用して SQL 文を動的に実行します。つまり、接続先のデータベース・サーバーは、SQL 文を動的に準備する必要があります。ODBC は、実行のたびに（リクエストされる間隔で）SQL 文を準備します。通常は、このことが Oracle GoldenGate ユーザーの問題になることはありません。詳細は、z/OS 上の DB2 のマニュアルを参照してください。

Oracle GoldenGate を使用して DDL をレプリケートする場合は、SQL の実行前に、ストアド・プロシージャまたは問合せに影響を受けるすべてのオブジェクトが、正しい構造で存在している必要があります。したがって、これらのオブジェクトの構造に影響する DDL (CREATE や ALTER など) は、SQLEXEC の実行前に実行される必要があります。

## SQLEXEC とストアド・プロシージャの使用

MAP 文内からでストアド・プロシージャを実行するには、SPNAME 句を使用します。

**構文**

```
SQLEXEC (
 SPNAME <sp name>
 [, ID <logical name>]
 {, PARAMS <param spec> | NOPARAMS}
 [, <option>] [, ...]
)
```

| コンポーネント                           | 説明                                                                                                                                                                                |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <sp name>                         | 実行するプロシージャ名を指定します。                                                                                                                                                                |
| ID <logical name>                 | プロシージャの論理名を定義します。このオプションは、MAP 文内でプロシージャを複数回実行するときに使用します。1 つの MAP 文で、最大 20 のストアド・プロシージャで実行できます。プロシージャの実行が 1 回の場合は必要ありません。                                                          |
| PARAMS <param spec>  <br>NOPARAMS | プロシージャがパラメータを受け付けるかどうかを定義します。<br>PARAMS <param spec> または NOPARAMS のいずれかを使用する必要があります。<br><param spec> では、入力パラメータおよび入力ソースを定義します。                                                    |
| <option>                          | ストアド・プロシージャの影響を制御するために、単独または他のオプションと組み合わせて使用できる、次のいずれかのオプションを指定します。<br>AFTERFILTER   BEFOREFILTER<br>ALLPARAMS<br>DBOP<br>ERROR<br>EXEC<br>MAXVARCHARLEN<br>PARAMBUFSIZE<br>TRACE |

SQLEXEC コンポーネントの説明は、アルファベット順に 249 ページから記載されています。

## SQLEXEC と問合せの使用

MAP 文内から問合せを実行するには、ID および QUERY 句を使用します。

### 構文

```
SQLEXEC (
 ID <logical name>
 , QUERY "<sql query>"
 {, PARAMS <param spec>| NOPARAMS}
 [, <option>] [, ...]
)
```

| コンポーネント                          | 説明                                                                                                                                                                                                                                                                                                                  |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID <logical name>                | 問合せの論理名を定義します。問合せの結果から値を抽出するには、論理名が必要です。ID <logical name> は、問合せから返された列値を参照します。                                                                                                                                                                                                                                      |
| QUERY<br>"<sql query>"           | データベースに対して実行する SQL 問合せの構文を指定します。問合せは、問合せを実行するデータベースの有効な標準問合せ言語を使用している必要があります。<br><br>問合せは、SELECT 文の結果を返すか、INSERT、UPDATE、または DELETE 文を実行できます。SELECT 文の出力を生成する問合せの場合は、SELECT によって最初に返される行のみが処理されます。SELECT 文には "INTO ..." 句を指定しないでください。<br><br>問合せは 1 行に収め、引用符で囲む必要があります。最良の結果を得るために、各開始引用符の後ろおよび各終了引用符の前に空白を入力してください。 |
| PARAMS<br><param spec>  NOPARAMS | 問合せがパラメータを受け付けるかどうかを定義します。これらのいずれかのオプションを使用する必要があります。<param spec> では、入力パラメータおよび入力ソースを定義します。                                                                                                                                                                                                                         |
| <option>                         | 問合せの影響を制御するために、単独または他のオプションと組み合わせて使用できる、次のいずれかのオプションを指定します。<br><br>AFTERFILTER   BEFOREFILTER<br>ALLPARAMS<br>DBOP<br>ERROR<br>EXEC<br>MAXVARCHARLEN<br>PARAMBUFSIZE<br>TRACE                                                                                                                                       |

SQLEXEC コンポーネントの説明は、アルファベット順に 249 ページから記載されています。



## 入力パラメータ用のプレースホルダの使用

ほとんどの問合せは、入力パラメータ用のプレースホルダを必要とします。問合せ内でのパラメータの指定方法は、データベースのタイプによって異なります。

- Oracle の場合は、次の例のように、入力パラメータはコロン (:) を使用して指定し、その後にパラメータ名を続けます。  

```
"SELECT NAME FROM ACCOUNT WHERE SSN = :SSN AND ACCOUNT = :ACCT"
```
- 他のデータベースの場合は、次の例のように、入力パラメータは疑問符を使用して指定します。  

```
"SELECT NAME FROM ACCOUNT WHERE SSN = ? AND ACCOUNT = ?"
```

どのデータベースでも、パラメータ名を引用符で囲む必要はありません。

## パラメータの値渡し

Oracle GoldenGate は、入力値および出力値をプロシージャまたは問合せとやり取りするためのオプションを提供しています。

- ストアド・プロシージャまたは問合せ内で入力パラメータにデータ値を渡すには、SQLEXEC の PARAMS オプションを使用します (254 ページを参照してください)。
- ストアド・プロシージャまたは問合せから入力として値を FILTER または COLMAP 句に渡すには、次の構文を使用します。  

```
{<procedure name> | <logical name>}.<parameter>
```

### 条件:

- <procedure name> は、ストアド・プロシージャの実際の名前で、SQLEXEC 文の SPNAME に指定している値と一致する必要があります。この引数は、Oracle GoldenGate 実行中にプロシージャを 1 回実行する場合にのみ指定します。
- <logical name> は、SQLEXEC 文の ID オプションで指定した論理名です。この引数は、MAP 文内でプロシージャを複数回実行する場合に、問合せまたはストアド・プロシージャのインスタンスから値を渡すときに使用します。
- <parameter> は、参照表の列などのパラメータ名か、返された値を抽出する場合の RETURN\_VALUE のいずれかです。

上記の構文のかわりとして、@GETVAL ファンクションを使用できます。詳細は、419 ページを参照してください。

入力パラメータのネーミングには、次のような異なる構成があります。

- Oracle は、次の例のように、入力パラメータに論理名を付けることを許可します。  

```
SQLEXEC (ID appphone, QUERY " select per_type from ps_personal_data "
" where emplid = :vemplid "
" and per_status = 'N' and per_type = 'A' ",
PARAMS (vemplid = emplid)),
TOKENS (applid = @GETVAL(appphone.per_type));
```

- 他のデータベースでは、次の例のように、入力パラメータに P1、P2 のような名前を付け、入力パラメータごとに番号を増分させる必要があります。

```
SQLEXEC (ID appphone, QUERY " select per_type from ps_personal_data "
 " where emplid = ? "
 " and per_status = 'N' and per_type = 'A' ",
 PARAMS (p1 = emplid)),
TOKENS (applid = @GETVAL(appphone.per_type));
```

#### 例

次に、Oracle のソース表、ターゲット表、および参照表と、これらの表のパラメータをストアド・プロシージャの単一のインスタンス、およびストアド・プロシージャの複数のインスタンスに渡す方法の例を示します。

#### ソース表 "cust"

| 列名                      | 説明      |
|-------------------------|---------|
| custid                  | Number  |
| current_residence_state | Char(2) |
| birth_state             | Char(2) |

#### ターゲット表 "cust\_extended"

| 列名                           | 説明          |
|------------------------------|-------------|
| custid                       | Number      |
| current_residence_state_long | Varchar(30) |
| birth_state_long             | Varchar(30) |

#### 参照表 "state\_lookup"

| 列名           | 説明          |
|--------------|-------------|
| abbreviation | Char(2)     |
| long_name    | Varchar(30) |

#### 例 1

次に、参照表から値を取得するために 1 回実行されるストアド・プロシージャの使用例を示します。この値は、COLMAP 文のターゲット列にマッピングされます。

```
MAP sales.cust, TARGET sales.cust_extended, &
SQLEXEC (SPNAME lookup, &
PARAMS (long_name = birth_state)), &
COLMAP (custid = custid, &
birth_state_long = lookup.long_name);
```

**例 2** 次に、参照表から値を取得するストアド・プロシージャを複数回実行する例を示します。値は、ターゲット列にマッピングされます。

```
MAP sales.cust, TARGET sales.cust_extended, &
 SLEXEC (SPNAME lookup, ID lookup1, &
 PARAMS (long_name = current_residence_state)), &
 SLEXEC (SPNAME lookup, ID lookup2, &
 PARAMS (long_name = birth_state)), &
 COLMAP (custid = custid, current_residence_state_long = lookup1.long_name, &
 birth_state_long = lookup2.long_name);
```

### AFTERFILTER および BEFOREFILTER の使用

AFTERFILTER および BEFOREFILTER では、MAP 文の FILTER 句との関連でストアド・プロシージャまたは問合せをいつ実行するかを指定します。

**構文** AFTERFILTER | BEFOREFILTER

| ルール          | 説明                                                                            |
|--------------|-------------------------------------------------------------------------------|
| AFTERFILTER  | FILTER 文の後に SQL 文を実行します。これにより、フィルタが成功しない場合に SQL 実行のオーバーヘッドを回避できます。これはデフォルトです。 |
| BEFOREFILTER | SQL の結果をフィルタで使用できるように、FILTER 文の前に SQL を実行します。                                 |

**例** SLEXEC (SPNAME check, NOPARAMS, BEFOREFILTER)

### ALLPARAMS の使用

ALLPARAMS は、ストアド・プロシージャまたは問合せの実行のために、指定されたすべてのパラメータが存在している必要があるかどうかを決定するグローバル・ルールとして使用します。ALLPARAMS で設定するグローバル・ルールよりも、PARAMS 句内で設定する個別のパラメータのルールのほうが優先されます。

**構文** ALLPARAMS {OPTIONAL | REQUIRED}

| ルール      | 説明                                                   |
|----------|------------------------------------------------------|
| OPTIONAL | すべてのパラメータが存在しているかどうかにかかわらず、SQL の実行を許可します。これはデフォルトです。 |
| REQUIRED | SQL を実行するために、すべてのパラメータが存在している必要があります。                |

**例** SLEXEC (SPNAME lookup,  
PARAMS (long\_name = birth\_state, short\_name = state),  
ALLPARAMS OPTIONAL)

### DBOP の使用

DBOP では、ストアド・プロシージャまたは問合せ内で実行された INSERT、UPDATE、DELETE、および SELECT 文をコミットします。コミットしない場合、これらの文はロールバックされる可能性があります。Oracle GoldenGate は、ソース・トランザクションと同じトランザクション境界内でコミットを発行します。

**警告** データベース、特に本番環境のデータベースに対して SQLEXEC プロシージャを実行する場合は、注意して使用してください。プロシージャによってコミットされた変更は、既存のデータを上書きすることがあります。

**構文** DBOP

**例** SQLEXEC (SPNAME check, NOPARAMS, DBOP)

### ERROR の使用

ERROR では、ストアド・プロシージャまたは問合せに関連するエラーに対するレスポンスを定義します。明示的なエラー処理の指定がない場合、Oracle GoldenGate プロセスはエラーを検出すると異常終了します。プロシージャからプロセスにエラーを返させ、ERROR でレスポンスを指定するようにしてください。

**構文** ERROR <action>

| アクション  | 説明                                                                                                                                                                                                                       |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IGNORE | Oracle GoldenGate に、ストアド・プロシージャまたは問合せに関連するすべてのエラーを無視させ、処理を継続させます。このパラメータ抽出結果は、" 列行方不明 " 状態になります。これはデフォルトです。                                                                                                              |
| REPORT | ストアド・プロシージャまたは問合せに、関連するすべてのエラーを破棄ファイルにレポートさせます。（エラーをレポートさせるには、DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。）このレポートは、エラーの原因の追跡に役立ちます。ここには、エラーの説明と、プロシージャまたは問合せとやり取りしたパラメータの値両方が含まれます。Oracle GoldenGate は、エラーをレポートした後、処理を継続します。 |
| RAISE  | REPERERROR パラメータで設定されたルールに従ってエラーを処理します。Oracle GoldenGate は、エラーを処理する前に、現在の MAP 文に関連する他のストアド・プロシージャまたは問合せの処理を継続します。                                                                                                        |
| FINAL  | RAISE と似ていますが、プロシージャまたは問合せに関連するエラーを検出したときに、残りのストアド・プロシージャおよび問合せがバイパスされます。エラー処理は、エラーの直後に起動されます。                                                                                                                           |
| FATAL  | プロシージャまたは問合せに関連するエラーを検出したときに、即座に Oracle GoldenGate を異常終了させます。                                                                                                                                                            |

**例** 次に、RAISE オプションを使用する Oracle の例外文の例を示します。Oracle のファンクション、RAISE\_APPLICATION\_ERROR を使用します。Replicat パラメータ・ファイルで REPERERROR (-20000, DISCARD) を使用してエラーが定義されている場合、Oracle GoldenGate はこのレコードを破棄し、処理を継続します。

```
EXCEPTION
WHEN no_match_rec THEN
RAISE_APPLICATION_ERROR(-20000, 'No Matching Update In Target');
```

### EXEC の使用

EXEC では、MAP 文のストアド・プロシージャまたは問合せを実行する頻度、および出力パラメータを抽出する場合に結果を有効とみなす期間を制御します。

構文 EXEC <frequency>

| 頻度          | 説明                                                                                                                                                                         |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAP         | プロシージャまたは問合せが指定されている各ソース・ターゲットのマップで、プロシージャまたは問合せを 1 回実行します。MAP を使用する場合、同一のソース表を持つそれ以降のマップでは結果は無効になります。たとえば、ソース表が複数のターゲット表と同期を取る場合、結果は最初のソース・ターゲットのマップでのみ有効です。MAP はデフォルトです。 |
| ONCE        | Oracle GoldenGate の実行中、関連する MAP 文の最初の呼び出し時にプロシージャまたは問合せを 1 回実行します。結果は、プロセスが実行しているかぎり有効です。                                                                                  |
| TRANSACTION | プロシージャまたは問合せをソース・トランザクションで 1 回実行します。結果は、トランザクションのすべての操作に有効です。                                                                                                              |
| SOURCEROW   | プロシージャまたは問合せをソース行操作で 1 回実行します。このオプションは、ソース表を複数のターゲット表と同期し、プロシージャまたは問合せの結果がソース・ターゲット・マッピングのたびに呼び出される場合に使用します。                                                               |

**例 1** 次に、ONCE の使用例を示します。

```
MAP sales.cust, TARGET sales.cust_extended, &
SQLEXEC (SPNAME lookup, PARAMS (long_name = birth_state), EXEC ONCE), &
COLMAP (custid = custid, &
birth_state_long = lookup.long_name);
```

**例 2** 次に、TRANSACTION の使用例を示します。

```
MAP sales.cust, TARGET sales.cust_extended, &
SQLEXEC (SPNAME lookup, PARAMS (long_name = birth_state), EXEC TRANSACTION), &
COLMAP (custid = custid, &
birth_state_long = lookup.long_name);
```

**例 3** 次に、デフォルト (MAP) の誤った使用例を示します。2 つの MAP 文で、同一のソース表と 2 つの異なるターゲット表を同期します。しかし、プロシージャ lookup の結果は、2 番目の MAP の実行までに無効になってしまい、2 番目の MAP は " 行列方不明 " の状態になります。この機能を正しく実装するには、SOURCEROW を使用する必要があります。

```
MAP sales.srctab, TARGET sales.targtab, &
SQLEXEC (SPNAME lookup, PARAMS (param1 = srccol)), &
COLMAP (targcol = lookup.param2);
```

```
MAP sales.srctab, TARGET sales.targtab2, &
COLMAP (targcol2 = lookup.param2);
```

**例 4** 次に、SOURCEROW の使用例を示します。このケースでは、プロシージャが各ソース行操作で実行されるため、2 番目の MAP は有効な値を返します。

```
MAP sales.srctab, TARGET sales.targettab, &
SQLEXEC (SPNAME lookup, PARAMS (param1 = srccol), EXEC SOURCEROW), &
COLMAP (targcol = lookup.param2);
```

```
MAP sales.srctab, TARGET sales.targettab2, &
COLMAP (targcol2 = lookup.param2);
```

### ID の使用

ID は、MAP 文内の問合せおよびストアド・プロシージャに対して、次のように使用します。

- 問合せでは、ID <logical name> を使用し、Oracle GoldenGate がこの名前を使用して問合せから返される列値を参照できるようにします。
- ストアド・プロシージャでは、ID <logical name> を使用して、たとえば 2 つの異なる列のマッピングなどのために、1 つの MAP 文内でプロシージャを複数回呼び出せるようにします。それ以外の場合、ID は必要ありません。1 つの MAP 文で、最大 20 のストアド・プロシージャで実行できます。これらは、パラメータ・ファイルにリストされた順に実行されます。

**構文** ID <logical name>

| コンポーネント        | 説明                                                                                      |
|----------------|-----------------------------------------------------------------------------------------|
| <logical name> | ストアド・プロシージャまたは問合せの論理名。たとえば、"lookup" という名前のプロシージャの論理名を "lookup1"、"lookup2" などのように指定できます。 |

**例 1** 次に、ID <logical name> の使用例を示します。各列マッピングが、lookup1 および lookup2 を使用して lookup という名前のストアド・プロシージャを個別にコールし、それぞれの結果を参照します。

```
MAP sales.srctab, TARGET sales.targettab, &
SQLEXEC (SPNAME lookup, ID lookup1, PARAMS (param1 = srccol)), &
COLMAP (targcol1 = lookup1.param2), &
SQLEXEC (SPNAME lookup, ID lookup2, PARAMS (param1 = srccol)), &
COLMAP (targcol = lookup2.param2);
```

**例 2** 次に、lookup という名前のストアド・プロシージャを 1 回実行する例を示します。このケースでは、実際のプロシージャ名を使用します。論理名は必要ありません。

```
MAP sales.tab1, TARGET sales.tab2, &
SQLEXEC (SPNAME lookup), PARAMS (param1 = srccol), &
COLMAP (targcol = lookup.param1);
```

**例 3** 次に、Oracle および SQL Server の問合せにそれぞれ ID <logical name> を使用する例を示します。この例では、このドキュメントのスペースの制約によって、SQLEXEC 文が複数の行にまたがっています。実際の SQLEXEC 文は、1 行内に含める必要があります。

```
MAP sales.account, TARGET sales.newacct, &
SQLEXEC (ID lookup, &
QUERY "select desc_col into desc_param from lookup_table &
where code_col = :code_param", &
PARAMS (code_param = account_code)), &
COLMAP (newacct_id = account_id, &
newacct_val = lookup.desc_param);
```

```
MAP sales.account, TARGET sales.newacct, &
SQLEXEC (ID lookup, &
QUERY "select desc_col into desc_param from lookup_table &
where code_col = ?", &
PARAMS (p1 = account_code)), &
COLMAP (newacct_id = account_id, &
newacct_val = lookup.desc_param);
```

### MAXVARCHARLEN の使用

MAXVARCHARLEN では、ストアド・プロシージャまたは問合せの出力パラメータに割り当てる最大長を指定します。この最大値を超える出力値は切り捨てられます。

**構文** MAXVARCHARLEN <num bytes>

| コンポーネント     | 説明                                                                      |
|-------------|-------------------------------------------------------------------------|
| <num bytes> | 出力パラメータに許可する最大バイト数を定義します。明示的に MAXVARCHARLEN 句を使用しない場合のデフォルトは、255 バイトです。 |

**例** MAXVARCHARLEN 100

### NOPARAMS の使用

NOPARAMS は、ストアド・プロシージャまたは問合せがパラメータを必要としない場合に、PARAMS のかわりに使用します。PARAMS 句または NOPARAMS のいずれかを指定する必要があります。

**構文** NOPARAMS

**例** SQLEXEC (SPNAME check, NOPARAMS)

### PARAMBUFSIZE の使用

PARAMBUFSIZE では、入力および出力パラメータを含むパラメータ情報を保持するメモリー・バッファの最大値を指定します。Oracle GoldenGate は、パラメータに割り当てられたメモリーと最大値との差が 500 バイト以内に達するたびに警告を発行します。

**構文** PARAMBUFSIZE <num bytes>

| コンポーネント       | 説明                                                                          |
|---------------|-----------------------------------------------------------------------------|
| (<num bytes>) | メモリー・バッファに許可する最大バイト数を定義します。明示的に PARAMBUFSIZE 句を使用しない場合のデフォルトは、10,000 バイトです。 |

**例** PARAMBUFSIZE 15000

## PARAMS の使用

PARAMS では、入力を受け付けるストアド・プロシージャまたは問合せのパラメータ名と、入力を提供するソース列名または Oracle GoldenGate 列変換ファンクション名を指定します。PARAMS 句または NOPARAMS のいずれかを指定する必要があります。

以下は、SQLEXEC によってサポートされるデータベース、および入力と出力パラメータにサポートされるデータ・タイプです。

### DB2 LUW および z/OS

- CHAR
- VARCHAR
- DATE
- すべての数値データ・タイプ
- BLOB データ・タイプ

### Ingres

LOB 以外のすべてのデータ・タイプ

### MySQL

TEXT および BLOB 以外のすべてのデータ・タイプ

### Oracle

次のものを除く、すべての Oracle タイプがサポートされます。

- BFILE
- BLOB
- CFILE
- CLOB
- NCLOB
- NTY

### SQL Server

- CHAR
- VARCHAR
- DATETIME
- すべての数値データ・タイプ
- 長さが 200 バイト未満のイメージおよびテキスト・データ・タイプ
- TIMESTAMP パラメータ・タイプはネイティブにサポートされませんが、他のデータ・タイプをパラメータとして使用し、ストアド・プロシージャ内の TIMESTAMP 形式に値を変換できます。

### Sybase

TEXTIMAGE および UDT 以外のすべてのデータ・タイプ

### Teradata

Oracle GoldenGate でサポートされる、すべての Teradata データ・タイプ

デフォルトでは、出力パラメータは、パラメータ当たり 255 バイトで切り捨てられます。プロシージャがこれより長いパラメータを必要とする場合は、MAXVARCHARLEN オプションを使用します。



```
構文 PARAMS (
 [OPTIONAL | REQUIRED]
 <param name> = {<source column> | <GG function>}
 [, ...]
)
```

| コンポーネント                                                                             | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OPTIONAL   REQUIRED                                                                 | <p>パラメータ値が見つからないときに、プロシージャまたは問合せを実行するかどうかを決定します。</p> <ul style="list-style-type: none"> <li>◆ <b>OPTIONAL</b> は、<b>SQL</b> の実行にパラメータ値が必要ないことを示します。必要なソース列がデータベース操作で見つからない場合、またはソース列が見つからないために列変換ファンクションを完了できない場合でも、<b>SQL</b> は実行されます。</li> </ul> <p><b>OPTIONAL</b> は、<b>Oracle</b> 以外のすべてのデータベースのデフォルトです。<b>Oracle</b> では、ストアド・プロシージャ定義を取得するときに、パラメータがオプションかどうか自動的に判別されます。</p> <ul style="list-style-type: none"> <li>◆ <b>REQUIRED</b> は、パラメータ値が存在している必要があることを示します。パラメータ値が存在しない場合、<b>SQL</b> は実行されません。</li> </ul>                                                                                                                                                                                                                                                                                                                                        |
| <pre>&lt;param name&gt; = {   &lt;source column&gt;     &lt;GG function&gt; }</pre> | <p>パラメータ名を、入力を提供する列またはファンクションにマップします。</p> <p>条件：</p> <ul style="list-style-type: none"> <li>◆ <b>&lt;param name&gt;</b> は、次のいずれかです。           <ul style="list-style-type: none"> <li>ストアド・プロシージャの場合、ストアド・プロシージャ内の入力を受け付ける任意のパラメータ名（参照表の列など）です。</li> <li><b>Oracle</b> 問合せの場合、先行するコロンを除いた、問合せの任意の入力パラメータ名です。たとえば、<code>:param1</code> の場合は、<b>PARAMS</b> 句では <code>param1</code> と指定します。</li> <li><b>Oracle</b> 以外の問合せの場合は、<b>P<math>n</math></b> と指定します（<math>n</math> は、1 から始まる文内のパラメータ番号です）。たとえば、2 つのパラメータを持つ問合せの場合、<b>&lt;param name&gt;</b> エントリは <b>P1 および P2</b> になります。</li> </ul> </li> <li>◆ <b>&lt;source column&gt;</b> は、ソース列名です。デフォルトでは、指定した列が（圧縮更新であるために）ログに存在しない場合、このパラメータはプロシージャまたは問合せでこのパラメータに指定されているデフォルト値を使用します。</li> <li>◆ <b>&lt;GG function&gt;</b> は、<b>Oracle GoldenGate</b> 列変換ファンクション名です。列変換ファンクションの詳細は、第 4 章を参照してください。</li> </ul> |

**例**      次に、**account** 表からターゲット表 **newacct** にデータをマップする例を示します。**account** 表からのレコードを処理するとき、**Oracle GoldenGate** は、列マップを実行する前に **lookup** ストアド・プロシージャを実行します。プロシージャの **code\_param** パラメータは、**account\_code** ソース列から入力を受け付けます。

```
MAP sales.account, TARGET sales.newacct, &
 SQLEXEC (SPNAME lookup, PARAMS (code_param = account_code)), &
 COLMAP (newacct_id = account_id, &
 newacct_val = lookup.desc_param);
```

## TRACE の使用

TRACE では、入力および出力パラメータをレポート・ファイルに記録します。

SQLEXEC 証跡が有効化されているサンプルの破棄ファイル:

```
Input parameter values...

LMS_TABLE: INTERACTION_ATTR_VALUES
KEY1: 2818249
KEY2: 1
Report File:

From Table MASTER.INTERACTION_ATTR_VALUES to MASTER.INTERACTION_ATTR_VALUES:
inserts: 0
updates: 0
deletes: 0
discards: 1

Stored procedure GGS_INTERACTION_ATTR_VALUES:
 attempts: 2
 successful: 0
```

**構文**      TRACE {ALL | ERROR}

| アクション | 説明                                                                |
|-------|-------------------------------------------------------------------|
| ALL   | 呼び出された各プロシージャまたは問合せの入力および出力パラメータをレポート・ファイルに書き込みます。これはデフォルトです。     |
| ERROR | SQL エラーの発生後にのみ、呼び出された各プロシージャまたは問合せの入力および出力パラメータをレポート・ファイルに書き込みます。 |

**例**      SQLEXEC (SPNAME lookup, PARAMS (long\_name = birth\_state,  
short\_name = state), TRACE ERROR)

## TARGETDEF の使用

TARGETDEF では、ターゲット定義テンプレートを指定します。定義テンプレートは、特定のターゲット表に対して DEFGEN が実行されるときに、このオブジェクトの定義に基づいて作成されます。テンプレートが作成されると、この表と同一の定義を持つ新しいターゲット表は、DEFGEN を実行せずに、かつ Replicat の停止と起動を伴わずに追加できます。TARGETDEF で指定されたテンプレートの定義は、定義の参照に使用されます。DEFGEN の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文**      MAP <table spec>, TARGET <table spec>, TARGETDEF <definitions template>;

| 引数                     | 説明                                                                                               |
|------------------------|--------------------------------------------------------------------------------------------------|
| <definitions template> | DEFGEN パラメータ・ファイルの TABLE の DEF オプションで指定されている定義テンプレート名。テンプレートに含まれる定義は、この MAP 文の表の定義と同一である必要があります。 |

**例**      MAP acct.cust\*, TARGET acc.cust\*, DEF custdef, TARGETDEF tcustdef;

## TRIMSPACES および NOTRIMSPACES の使用

TRIMSPACES および NOTRIMSPACES では、ターゲット表に送信するときに末尾の空白を切り捨てるかどうかを制御します。これらのパラメータは、CHAR から VARCHAR への列マッピングにのみ影響します。デフォルトは TRIMSPACES です。

TRIMSPACES および NOTRIMSPACES は、異なる MAP 文および文グループに対する切捨て機能を有効化または無効化するために、パラメータ・ファイルのルート・レベルでも使用できます。

**構文** MAP <table spec>, TARGET <table spec>, {TRIMSPACES | NOTRIMSPACES};

**例** 次に、最初の 2 つのターゲットに対してはデフォルトの末尾切捨てを行い、最後の 2 つのターゲットでは末尾切捨てを行わない例を示します。

```
MAP fin.src1, TARGET fin.tgt1;
MAP fin.src1, TARGET fin.tgt2;
MAP fin.src1, TARGET fin.tgt3, NOTRIMSPACES;
MAP fin.src1, TARGET fin.tgt4, NOTRIMSPACES;
```

## WHERE の使用

WHERE では、条件文に基づいてレコードを選択します。拡張 ASCII または出力不可能文字を含む Unicode 列または文字列を使用するには、223 ページを参照してください。WHERE 句の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文** MAP <table spec>, TARGET <table spec>,  
WHERE (<where clause>);

| コンポーネント        | 説明                                                                                                                                                                   |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <where clause> | <p>次の例のように、条件に基づいてレコードを選択します。</p> <pre>WHERE (branch = "NY")</pre> <p>次の表に、許可されている WHERE 演算子を示します。</p> <p>WHERE は、主キー更新操作の一部として条件文の主キー列のビフォア・イメージの評価をサポートしていません。</p> |

表 44 許可されている WHERE 演算子

| 演算子             | 例:                                                                                        |
|-----------------|-------------------------------------------------------------------------------------------|
| 列名              | PRODUCT_AMT                                                                               |
| 数値              | -123, 5500.123                                                                            |
| 引用符で囲まれたリテラル文字列 | "AUTO", "Ca"                                                                              |
| 列テスト            | @NULL, @PRESENT, @ABSENT (レコードの列が NULL、存在、不在かのテスト)。これらのテストは、Oracle GoldenGate に組み込まれています。 |
| 比較演算子           | =, <>, >, <, >=, <=                                                                       |
| 結合演算子           | AND, OR                                                                                   |
| グループ化用カッコ       | 複数の要素を論理的にグループ化するには、開きおよび閉じカッコを使用します。                                                     |

**例** 次の WHERE 句は、TABLE または MAP 文に含まれる場合の WHERE 句での Unicode エスケープ・シーケンスの使用を示したものです。それぞれ、「Élise」、「Zoé」および「Véronique」の入力文字内の別の場所でアキュート・アクセントが付いたラテン文字の小文字の e (コード・ポイント U+00E9) がエスケープされています。

```
WHERE (FIRSTNAME <> "\u00e9lise")
WHERE (FIRSTNAME <> "Zo\u00e9")
WHERE (FIRSTNAME <> "V\u00e9ronique")
```

## MAPEXCLUDE

**適用対象** Replicat

MAPEXCLUDE パラメータと MAP パラメータでは、ワイルドカード指定から表を明示的に除外します。MAPEXCLUDE は、除外する表を含むすべての MAP 文よりも先に指定する必要があります。

**デフォルト** なし

**構文** MAPEXCLUDE <exclude specification>

| 引数                      | 説明                                                                             |
|-------------------------|--------------------------------------------------------------------------------|
| <exclude specification> | 除外する表の名前またはワイルドカード指定。MAPEXCLUDE には、MAP 文で表をワイルドカード指定するときと同じワイルドカード・ルールが適用されます。 |

**例** 次の例では、MAP 文は TEST という名前の表を除くすべての表を取得します。

```
MAPEXCLUDE fin.TEST
MAP fin.*, TARGET fin.*;
```

## MARKERTABLE

**適用対象** GLOBALS

MARKERTABLE パラメータでは、Oracle DDL 同期をサポートする DDL マーカー表の名前をデフォルトの GGS\_DDL\_HIST 以外にする場合に、その名前を指定します。マーカー表には、DDL 操作に関する情報が保持されます。このパラメータは、Oracle にのみ有効です。

マーカー表の名前は、params.sql スクリプトの marker\_table\_name パラメータにも指定する必要があります。このスクリプトは、ルート Oracle GoldenGate インストール・ディレクトリにあります。

マーカー表および params.sql の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** GGS\_MARKER

**構文** MARKERTABLE <table\_name>

| 引数           | 説明      |
|--------------|---------|
| <table_name> | マーカー表名。 |

## MAXDISCARDRECS

**適用対象** Replicat

MAXDISCARDRECS パラメータでは、破棄ファイルにレポートされる MAP 文当たりのエラー数を制限します。

このパラメータは、次の理由で使用します。

- 多数のエラーの発生が予想されるが、レポートする必要がない。
- 破棄ファイルのサイズを管理する。

このパラメータは表に固有で、それ以降のすべての MAP 文に適用されます。1 つのパラメータ・ファイルで、MAXDISCARDRECS の複数のインスタンスを使用できます。

**デフォルト** なし

**構文** MAXDISCARDRECS <number>

| 引数       | 説明             |
|----------|----------------|
| <number> | レポートするエラーの最大数。 |

**例** MAXDISCARDRECS 1000

## MAXFETCHSTATEMENTS

**適用対象** Extract

MAXFETCHSTATEMENTS パラメータでは、Extract が Oracle ソース・データベースから行データをフェッチするために使用可能な、許可される準備済問合せの最大数を制御します。Extract は、表から直接、または UNDO 表領域（Oracle 9i 以上）からデータをフェッチする必要がある場合があります。フェッチされたデータは、トランザクション・ログ・レコードから論理 SQL 文を構築するために使用できる情報が不足しているときに使用されます。

問合せは必要に応じて準備およびキャッシュされます。MAXFETCHSTATEMENTS で指定した値に到達すると、最も古い問合せが最も新しい問合せに置き換えられます。このパラメータの値は、Extract がフェッチ問合せのみのために保持しているオープンされているカーソル数を制御します。Extract は、ストアド・プロシージャなど他の目的のために追加のカーソルを使用できます。このパラメータは、Oracle データベースにのみ有効です。

**デフォルト** 100

**構文** MAXFETCHSTATEMENTS <number>

| 引数       | 説明                                                                                                                  |
|----------|---------------------------------------------------------------------------------------------------------------------|
| <number> | Extract が準備済問合せに使用する最大カーソル数。データベースが、MAXFETCHSTATEMENTS で指定するカーソル数の他に、別のアプリケーションおよびプロセスが使用するカーソルもサポートできることを確認してください。 |

**例** MAXFETCHSTATEMENTS 150

## MAXSQLSTATEMENTS

**適用対象** Replicat

MAXSQLSTATEMENTS パラメータでは、Replicat が標準処理モードおよび BATCHSQL モード（118 ページを参照してください）両方で使用可能な準備済 SQL 文の数を制御します。MAXSQLSTATEMENTS の値によって、Replicat が保持するオープンされているカーソル数が決定されます。データベースが、ここで指定するカーソル数の他に、別のアプリケーションおよびプロセスが使用するカーソルもサポートできることを確認してください。MAXSQLSTATEMENTS を変更する前に、Oracle サポートに連絡してください。詳細は、<http://support.oracle.com> を参照してください。

MAXSQLSTATEMENTS を使用するには、DYNSQL パラメータが有効である必要があります。DYNSQL はデフォルトです。

**デフォルト** 250 カーソル

**構文** MAXSQLSTATEMENTS <number>

| 引数       | 説明                                           |
|----------|----------------------------------------------|
| <number> | Replicat が使用する最大カーソル数。最大値は 250 です。最小値は 1 です。 |

**例** MAXSQLSTATEMENTS 200

# MAXTRANSOPS

**適用対象** Replicat

MAXTRANSOPS パラメータでは、大きなソース・トランザクションを小さく分割してターゲット・システムに送信します。このパラメータは、ターゲット・データベースが大きなトランザクションに対応するように構成されていないときに使用できます。たとえば、ターゲットの Oracle ロールバック・セグメントが百万の削除を実行するソース・トランザクションを再作成できるほど大きくない場合に MAXTRANSOPS 10000 を指定すると、Replicat は各 10,000 削除グループの後にコミットを発行できます。

## 使用の制限

MAXTRANSOPS は、効率化だけではなく正当なビジネス上の目的を持って使用する必要があります。MAXTRANSOPS を使用する場合、Replicat は正しい順番で操作を適用しますが、ソース・アプリケーションで定義された境界に課せられているトランザクションの整合性を変更することになります。

デフォルトでは、Oracle GoldenGate リリース 10 以降は、失敗からのリカバリの際、Extract は失敗の発生時に処理中だったソース・トランザクション全体を再送し、古いデータを上書きするのではなく、トレイル・ファイルの末尾にこのトランザクションを追加します。新しいトランザクションには再開レコードによってフラグが立てられ、ロールバックしてこのトランザクションをもう一度開始する必要があることが Replicat に通知されます。ただし、MAXTRANSOPS によって Replicat にこのトランザクションを分割させた場合、Replicat がロールバックできるのは、ターゲット・データベースにコミットされていないトランザクションのみです。Replicat がコミット済操作をもう一度処理した場合、SQL 操作のタイプに応じて、行重複エラーまたは行方不明エラーが発生します。

MAXTRANSOPS を使用しながらこのような状況を回避するには、RECOVERYOPTIONS パラメータを OVERWRITEMODE に設定して Extract を構成し、新しいデータで新しいデータを上書きします。ただし、上書きによってトレイル・レコードが破損することがあるため、このモードでは特定の失敗の後のリカバリがより困難になる場合があります。このようなケースでは、Oracle サービス・リクエストをオープンすることをお勧めします。詳細は、<http://support.oracle.com> を参照してください。

**デフォルト** 100,000,000

**構文** MAXTRANSOPS <transaction count>

| 引数                  | 説明                        |
|---------------------|---------------------------|
| <transaction count> | 単一のトランザクション・グループに分割する操作数。 |

**例** MAXTRANSOPS 10000

## MGRSERVNAME

**適用対象** GLOBALS

GLOBALS パラメータ・ファイルの MGRSERVNAME パラメータでは、Manager プロセスを Windows サービスとしてインストールする場合にその名前を指定します。このパラメータは、Oracle GoldenGate の複数のインスタンスをインストールするときや、Manager プロセスを使用する Oracle GoldenGate Veridata もインストールするときなど、同一のシステムにサービスとして複数の Manager のインスタンスをインストールする場合にのみ必要です。

インストールする Manager サービスごとに、MGRSERVNAME を含む GLOBALS ファイルが必要です。システムにサービス名を登録するときに、インストーラが MGRSERVNAME を参照するため、これらのファイルはサービスのインストール前に作成する必要があります。

**デフォルト** なし

**構文** MGRSERVNAME <name>

| 引数   | 説明                    |
|------|-----------------------|
| <名前> | Manager サービス名 (1 単語)。 |

**例** MGRSERVNAME GoldenGate

## NOHEADERS

**適用対象** Extract

NOHEADERS パラメータでは、抽出ファイルにレコード・ヘッダーが含まれていないことを示します。この場合、Replicat は、抽出ファイルには単一の表タイプに関連する挿入レコードのみが含まれ、各レコードの長さとタイプが同一であるとみなします。

NOHEADERS を使用する場合は、Extract パラメータ・ファイルで FORMATASCII パラメータと NOHDRFIELDS オプションを使用する必要があります。NOHEADERS を使用する場合、すべての MAP 文で指定できるソース表は 1 つのみになります。

**デフォルト** なし

**構文** NOHEADERS



## NUMFILES

**適用対象** Extract および Replicat

NUMFILES パラメータでは、TABLE または MAP 文で指定されている表の情報を保持するために割り当てる初期メモリ構造数を制御します。NUMFILESS は、すべての TABLE または MAP エントリが有効になる前に指定する必要があります。

NUMFILES の値に到達後に動的に割り当てる追加のメモリ構造数を制御するには、ALLOCFILES パラメータを使用します（114 ページを参照してください）。システム・リソースが許可するメモリーが、プロセスによって必要に応じて割り当てられるので、NUMFILES と ALLOCFILES は、ともにデフォルト値で問題なく機能するはずです。

**デフォルト** 1000

**構文** NUMFILES <number of structures>

| 引数                     | 説明                                                                                                    |
|------------------------|-------------------------------------------------------------------------------------------------------|
| <number of structures> | 割り当てるメモリ構造数。メモリーの無駄な消費を防ぐため、NUMFILES を不必要に高い数に設定しないでください。Oracle GoldenGate のメモリーは、最大で 200 万表をサポートします。 |

**例** NUMFILES 4000

## OBEY

**適用対象** Extract および Replicat

OBEY パラメータでは、現在のパラメータ・ファイル以外のファイルからパラメータ設定を取得します。Oracle GoldenGate は、他のファイルのパラメータを処理した後に現在のパラメータ・ファイルの処理を再開します。

OBEY を使用するには、取得するパラメータを含むパラメータ・ファイルを作成して保存しておく必要があります。現在のパラメータ・ファイルで OBEY を使用すると、他のファイルが呼び出されます。OBEY 文は、他の OBEY 文内にネストできません。

頻繁に使用するパラメータを呼び出すために、OBEY のかわり、または OBEY の使用に加えて、Oracle GoldenGate マクロを使用できます。マクロ使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** なし

**構文** OBEY <file name>

| 引数          | 説明                          |
|-------------|-----------------------------|
| <file name> | パラメータまたはコマンドの取得元ファイルの完全修飾名。 |

**例** OBEY /home/ggs/myparams

## OUTPUTFILEUMASK

**適用対象** GLOBALS

OUTPUTFILEUMASK パラメータでは、Oracle GoldenGate プロセスがトレイル・ファイルおよび破棄ファイルの作成に使用する 8 進数 umask を指定します。OUTPUTFILEUMASK は、WIN32 システムには有効ではありません。

**デフォルト** umask 0 (すべての権限)

**構文** OUTPUTFILEUMASK <umask>

| 引数      | 説明                                                                                                      |
|---------|---------------------------------------------------------------------------------------------------------|
| <umask> | umask 値。0 ～ 0777 に設定する必要があります (それ以外の値にすると、エラー "Missing or invalid option for OUTPUTFILEUMASK." が発生します)。 |

**例** OUTPUTFILEUMASK 066

## OVERRIDEDUPS | NOOVERRIDEDUPS

**適用対象** Replicat

OVERRIDEDUPS および NOOVERRIDEDUPS パラメータでは、両方のレコードが同一のキーを持つときに、Replicat がターゲット・データベースの既存のレコードをレプリケートされたレコードで上書きするかどうかを制御します。

- OVERRIDEDUPS では、既存のレコードを上書きします。事前にターゲット表の切捨てを行わない初期ロードや、信頼できるソースとターゲット表の再同期のときに使用できます。
- NOOVERRIDEDUPS (デフォルト) では、既存のレコードを上書きせず、かわりに重複レコード・エラーを生成します。例外 MAP 文を使用して SQLEXEC 句で SQL プロシージャを実行し、エラーに対するレスポンスを開始できます。そうしない場合、トランザクションは異常終了することがあります。例外マップの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

例外マップが使用できない場合に Replicat を異常終了させずに重複レコードをバイパスするには、次のような REPERROR パラメータ文を指定します (<duplicate key error> は、主キー制約エラーに対するデータベース・エラー番号です)。

```
REPERROR (<duplicate key error>, IGNORE)
```

たとえば、Oracle データベースの場合、この文は次のようになります。

```
REPERROR (1, IGNORE)
```

重複レコードは、破棄ファイルに出力されます。

OVERRIDEDUPS および NOOVERRIDEDUPS は、TABLE または MAP 文に固有なので、各表または表グループに異なるルールを作成できます。SQLDUPERR パラメータ (307 ページを参照してください) を OVERRIDEDUPS とともに使用して、データベースに返される重複挿入の数値エラー・コードを指定します。

OVERRIDEDUPS は、HANDLECOLLISIONS が指定されたときに自動的に有効化されます。

OVERRIDEDUPS が有効な場合は、すべての Replicat プロセスにわたってレコードが時系列順に処理されない場合があります。

|       |                               |
|-------|-------------------------------|
| デフォルト | NOOVERRIDEDUPS                |
| 構文    | OVERRIDEDUPS   NOOVERRIDEDUPS |

## PASSTHRU | NOPASSTHRU

**適用対象** Extract

PASSTHRU および NOPASSTHRU パラメータでは、データ・ポンプ Extract がパススルー・モードで表を処理するか、標準モードで処理するかを制御します。パススルー・モードでは、Extract プロセスはデータベースからもデータ定義ファイルからも表定義を参照しません。通常は、Extract プロセスはデータベースにログインしてデータ定義を取得し、ターゲットが NonStop の場合には、データ定義ファイルを読み取ります。データ定義は、マッピングや変換ファンクションの実行に使用されます。

パススルー・モードを使用すると、取得したデータを、データベースがインストールされていない仲介システム上のデータ・ポンプにカスケードできます。フィルタリング、列マッピング、SQLEXEC ファンクション、変換、および操作や変換を必要とするその他のファンクションが使用できないため、ソースおよびターゲットの表名と構造は同一である必要があります。

PASSTHRU および NOPASSTHRU パラメータは、表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての MAP 文およびトレイルに有効です。したがって、特定の表セットにはパススルー動作を指定し、他の表に対してはデータ操作を含む標準処理を継続できます。操作が必要な表では、フィルタリングを実行する場合にソース定義ファイルが必要になり、列マッピングまたは変換を実行する場合にターゲット定義ファイルが必要になります。これらのファイルは、Oracle GoldenGate がこうしたアクションを実行するために必要なメタデータを提供します。

PASSTHRU モードでは、データ・ポンプは ASCII から EBCDIC、または EBCDIC から ASCII への自動変換を実行しません。

### DDL レプリケーションでの PASSTHRU

PASSTHRU モードでは、DDL はデータ・ポンプまたは VAM ソート Extract を通じて自動的に伝播されます。そのため、特定の名前のソース表で実行される DDL (ALTER TABLE TableA... など) は、データ・ポンプまたは VAM ソート Extract によって同一の名前の表に適用されます (ALTER TABLE TableA)。他の表を指定する TABLE 文の有無にかかわらず、このプロセスでは ALTER TABLE TableB としてマップすることはできません。

|       |                       |
|-------|-----------------------|
| デフォルト | NOPASSTHRU            |
| 構文    | PASSTHRU   NOPASSTHRU |

**例** 次に、fin.acct からのすべてのデータをパススルーし、fin.sales には通常の処理を許可するパラメータ・ファイルの例を示します。

```
EXTRACT fin
RMTHOST sysb, MGRPORT 7809
RMTTRAIL /ggs/dirdat/rt
PASSTHRU
TABLE fin.acct;
NOPASSTHRU
TABLE fin.sales, WHERE (ACCOUNT-CODE < 100);
```

## PASSTHRUMESSAGES | NOPASSTHRUMESSAGES

PASSTHRUMESSAGES および NOPASSTHRUMESSAGES パラメータでは、パススルー・モードで処理される表に関するメッセージを Extract レポート・ファイルに書き込むかどうかを制御します。有効にすると、次のようなメッセージが書き込まれます。

```
"PASSTHRU mapping resolved for source table <table name>"
```

|       |                                       |
|-------|---------------------------------------|
| デフォルト | PASSTHRUMESSAGES                      |
| 構文    | PASSTHRUMESSAGES   NOPASSTHRUMESSAGES |

## PORT

|       |                                                                                                                                              |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 適用対象  | Manager                                                                                                                                      |
|       | PORT では、動的サービスをリクエストするリモート・プロセス（通常は初期ロードの Replicat または Collector プロセス）とやり取りする Manager プロセス用の TCP/IP ポート番号を指定します。可能な場合にはデフォルト・ポート番号を使用してください。 |
| デフォルト | Port 7809                                                                                                                                    |
| 構文    | PORT <number>                                                                                                                                |

| 引数       | 説明          |
|----------|-------------|
| <number> | 使用可能なポート番号。 |

|   |           |
|---|-----------|
| 例 | PORT 7809 |
|---|-----------|

## PURGEDDLHISTORY

|      |                                                                                                                                                                                                                                            |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 適用対象 | Manager                                                                                                                                                                                                                                    |
|      | PURGEDDLHISTORY パラメータでは、行をパージすることによって Oracle データベースの DDL 履歴表のサイズを制御します。履歴表をパージするときは注意が必要です。これは DDL 同期プロセスの整合性にとって非常に重要な表であり、パージは Oracle GoldenGate によってリカバリ不能であるため、早期にパージを実行しないようにしてください。DDL データの損失の可能性をなくすために、定期的に履歴表のバックアップを取るようにしてください。 |
|      | 行を保持する最小時間および最大時間は、最終更新日に基づいて指定できます。最大時間および最小時間両方のルールを指定する必要があります。両方の指定がなければ、Manager が行をいつ削除するかの完全な基準がありません。たとえば、MINKEEPHOURS 3 を MAXKEEPHOURS 5 とともに使用した場合、過去 3 時間変更されなかった行を保持し、少なくとも 5 時間変更されなかったときに削除するよう指定できます。                        |
|      | このパラメータには、表名を指定する必要がありません。Oracle GoldenGate は、まず GLOBALS ファイルの DDLTABLE <table> パラメータに指定されている表を探し、この表が存在しない場合、Oracle GoldenGate はデフォルト名の GGS_DDL_HIST を使用します。                                                                              |

**注意** DDL 履歴表のページの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

PURGEDDLHISTORY を使用するには、USERID パラメータ、および必要なときは SOURCEDB パラメータも使用してログイン情報を指定する必要があります。

このパラメータは、Oracle にのみ有効です。

**デフォルト** 1 時間ごとにページ

**構文** PURGEDDLHISTORY  
{, <max rule>}  
[, <min rule>]  
[, <frequency>]

| 引数          | 説明                                                                                                                                                                                                                                                                                                                                                                            |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <max rule>  | <p>必須。次のいずれかを使用して行を保持する最大時間を設定します。</p> <p>MAXKEEPPHOURS &lt;n&gt;<br/>行が &lt;n&gt; 時間変更されなかった場合にページします。</p> <p>MAXKEEPDAYS &lt;n&gt;<br/>行が &lt;n&gt; 日間変更されなかった場合にページします。</p>                                                                                                                                                                                                |
| <min rule>  | <p>オプションですが、指定することをお勧めします。次のいずれかを使用して行を保持する最小時間を設定します。</p> <p>MINKEEPPHOURS &lt;n&gt;<br/>少なくとも指定した時間変更されていない行を保持します。</p> <p>MINKEEPDAYS &lt;n&gt;<br/>少なくとも指定した日数にわたって変更されていない行を保持します。</p>                                                                                                                                                                                   |
| <frequency> | <p>DDL 履歴をページする間隔を設定します。Manager のデフォルトのメンテナンス・タスクの実行間隔は、CHECKMINUTES パラメータで指定されているとおり 10 分です (136 ページを参照してください)。Manager は、10 分ごとに PURGEOLDEXTRACTS の間隔を確認し、指定された間隔の経過後にページを実行します。&lt;frequency&gt; には、次のいずれかを指定できます。</p> <p>FREQUENCYMINUTES &lt;n&gt;<br/>DDL 履歴をページする間隔 (分) を設定します。デフォルトのページ間隔は 60 分です。</p> <p>FREQUENCYHOURS &lt;n&gt;<br/>DDL 履歴をページする間隔 (時間) を設定します。</p> |

**例** 次の例では、過去 3 日間変更されなかったすべての行を保持し、少なくとも 5 日間変更されなかったときに削除します。ページ間隔は 30 分です。

PURGEDDLHISTORY MINKEEPDAYS 3, MAXKEEPDAYS 5, FREQUENCYMINUTES 30

# PURGEMARKERHISTORY

**適用対象**     Manager

PURGEMARKERHISTORY では、行をパージすることによって Oracle GoldenGate マーカー表のサイズを制御します。マーカー表はいつでもパージできます。

このパラメータには、表名を指定する必要がありません。Oracle GoldenGate は、まず GLOBALS ファイルの MARKERTABLE <table> パラメータに指定されている表を探し、この表が存在しない場合、Oracle GoldenGate はデフォルト名の GGS\_MARKER を使用します。

行を保持する最小時間および最大時間は、最終更新日に基づいて指定できます。最大時間および最小時間両方のルールを指定する必要があります。両方の指定がなければ、Manager がいつ行を削除するかの完全な基準がありません。たとえば、MINKEEPHOURS 3 を MAXKEEPHOURS 5 とともに使用した場合、過去 3 時間変更されなかった行を保持し、少なくとも 5 時間変更されなかったときに削除するよう指定できます。

**注意**     マーカー表のパージの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

PURGEMARKERHISTORY を使用するには、USERID パラメータ、および必要なときは SOURCEDB パラメータも使用してログイン情報を指定する必要があります。

**デフォルト**     1 時間ごとにパージ

**構文**     PURGEMARKERHISTORY  
{, <max rule>}  
[, <min rule>]  
[, <frequency>]

| 引数          | 説明                                                                                                                                                                                                                       |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <max rule>  | <p>必須。次のいずれかを使用して行を保持する最大時間を設定します。</p> <p>MAXKEEPHOURS &lt;n&gt;<br/>行が &lt;n&gt; 時間変更されなかった場合にパージします。</p> <p>MAXKEEPPDAYS &lt;n&gt;<br/>行が &lt;n&gt; 日間変更されなかった場合にパージします。</p>                                           |
| <min rule>  | <p>オプションですが、指定することをお勧めします。次のいずれかを使用して行を保持する最小時間を設定します。</p> <p>MINKEEPHOURS &lt;n&gt;<br/>少なくとも指定した時間変更されていない行を保持します。</p> <p>MINKEEPPDAYS &lt;n&gt;<br/>少なくとも指定した日数にわたって変更されていない行を保持します。</p>                              |
| <frequency> | <p>マーカー履歴をパージする間隔を設定します。Manager のデフォルトのメンテナンス・タスクの実行間隔は、CHECKMINUTES パラメータで指定されているとおり 10 分です (136 ページを参照してください)。Manager は、10 分ごとに PURGEOLDEXTRACTS の間隔を確認し、指定された間隔の経過後にパージを実行します。&lt;frequency&gt; には、次のいずれかを指定できます。</p> |

| 引数 | 説明                                                                   |
|----|----------------------------------------------------------------------|
|    | FREQUENCYMINUTES <n><br>マーカー履歴をパージする間隔（分）を設定します。デフォルトのパージ間隔は 60 分です。 |
|    | FREQUENCYHOURS <n><br>マーカー履歴をパージする間隔（時間）を設定します。                      |

**例** 次の例では、過去 3 日間変更されなかったすべての行を保持し、少なくとも 5 日間変更されなかったときに削除します。パージ間隔は 30 分です。

```
PURGEMARKERHISTORY MINKEEPDAYS 3, MAXKEEPDAYS 5, FREQUENCYMINUTES 30
```

## PURGEOLDEXTRACTS

**適用対象** Manager、Extract、Replicat

このパラメータの実装は、プロセスによって異なります。

### Extract および Replicat 用 PURGEOLDEXTRACTS

Extract または Replicat パラメータ・ファイルの PURGEOLDEXTRACTS パラメータでは、Oracle GoldenGate が新しいトレイル・ファイルから処理を開始するたびに、古いトレイル・ファイルを削除します。トレイル・ファイルを蓄積しないのでディスク領域を節約できます。パージは、プロセスがファイルの処理を終了したことがチェックポイントによって示された後に実行されます。

Extract によるパージの実行は、プロセスがデータ・ポンプの場合に適しています。データがターゲット・システムに送信された後にファイルをパージできます。それ以外の場合は、通常は Replicat によってパージを実行します。

Extract または Replicat パラメータ・ファイルでは、プロセスのインスタンスが 1 つの場合にのみ PURGEOLDEXTRACTS を使用するようにしてください。複数のグループが同一のトレイル・ファイル・セットを読み込む場合には、1 つのプロセスが読み込みを完了する前に、別のプロセスによってファイルがパージされてしまう可能性があります。かわりに、一元的にトレイル・ファイルを管理することができ、すべての Oracle GoldenGate 構成での使用が推奨される、Manager 用の PURGEOLDEXTRACTS を使用するようにしてください。

**デフォルト** 順序の次のファイルに切り替えるときにトレイル・ファイルをパージする。

**構文** PURGEOLDEXTRACTS

### Manager 用 PURGEOLDEXTRACTS

Oracle GoldenGate による証跡ファイルの処理の完了後、これらが消去されるようにするには、PURGEOLDEXTRACTS パラメータを使用します。PURGEOLDEXTRACTS を使用しない場合、消去は実行されず、証跡ファイルによって大量のディスク領域が消費される可能性があります。

Extract または Replicat の PURGEOLDEXTRACTS を使用するよりも、PURGEOLDEXTRACTS を Manager パラメータとして使用することをお勧めします。Manager パラメータとして PURGEOLDEXTRACTS を使用すると、証跡ファイルを一括管理でき、複数のプロセスを考慮できます。

パージを制御するには、次のルールに従います。

- **USECHECKPOINTS** では、すべてのプロセスがファイルの処理を完了したことがチェックポイントによって示されたときにページを行います。これはデフォルトですが、**NOUSECHECKPOINTS** オプションを使用すると無効化できます。チェックポイントを基準にページする場合、すべてのプロセスがデータの処理を終了するまでデータは削除されません。明示的な **NOUSECHECKPOINTS** エントリがある場合を除き、**USECHECKPOINTS** は **PURGEOLDEXTRACTS** とともに明示的に定義されているかにかかわらず使用されます。本番環境では、データの整合性を確保するために、チェックポイントに基づいてページを実行することが不可欠です。**USECHECKPOINTS** では、ページを実行する前に、**Extract** および **Replicat** 両方のチェックポイントが考慮されます。

- **MINKEEP** ルールでは、変更されていないデータを保持する最小時間を設定します。
  - **MINKEEPPHOURS** または **MINKEEPDAYS** では、<n> 時間または日間データを保持します。
  - **MINKEEPFILES** では、少なくとも <n> 個のトレイル・ファイル（アクティブなファイルを含む）を保持します。デフォルトは 1 です。

**MINKEEP** オプションは 1 つのみ使用します。このオプションが複数使用されている場合、Oracle GoldenGate は次に基づいてそのうち 1 つを選択します。

- **MINKEEPPHOURS** および **MINKEEPDAYS** 両方が指定されている場合、最後のオプションを受け付け、もう一方を無視します。
- **MINKEEPPHOURS** または **MINKEEPDAYS** が **MINKEEPFILES** とともに使用されている場合、**MINKEEPPHOURS** または **MINKEEPDAYS** を受け付け、**MINKEEPFILES** を無視します。

**Manager** は、**CHECKMINUTES** パラメータで設定されている値に基づいてページを実行します (136 ページを参照してください)。この値に到達すると、次のようにページ・ルールが評価されます。

1. **USECHECKPOINTS** のみ。**MINKEEP** ルールが指定されておらず、**USECHECKPOINTS** が有効化されている場合、保持するファイルの最小数は 1 になります。チェックポイントによって処理されたことが示されているファイルは、ファイルの最小数 1 を下回らないかぎりページされます。
2. **USECHECKPOINTS** と **MINKEEP** ルール。**USECHECKPOINTS** が有効化され、チェックポイントによってファイルが処理されたことが示されている場合、適切な **MINKEEP** ルールを違反しないかぎり、このファイルはページされます。
3. **NOUSECHECKPOINTS** のみ。**MINKEEP** ルールおよび **NOUSECHECKPOINTS** が指定されていない場合、チェックポイントは考慮されず、デフォルト・ルールの 1 つのファイルの保持ルールを違反しないかぎり、ファイルはページされます。
4. **NOUSECHECKPOINTS** と **MINKEEP** ルール。**MINKEEP** ルールおよび **NOUSECHECKPOINTS** が指定されている場合、**MINKEEP** ルールを違反しないかぎり、ファイルはページされます。

**Manager** は、ローカル・システムに構成されている **Extract** および **Replicat** プロセスに基づいてページするファイルを決定します。少なくとも 1 つのプロセスがトレイル・ファイルを読み取っている場合、**Manager** は指定されているルールを適用しますが、それ以外の場合はルールは有効になりません。

同一の **Manager** パラメータ・ファイル内では、500 を超える **PURGEOLDEXTRACTS** パラメータ文を使用しないでください。

このパラメータを使用する場合は、ユーザーおよび Oracle GoldenGate 以外のプログラムにトレイル・ファイルの削除を許可しないでください。それにより、**PURGEOLDEXTRACTS** が正常に機能しなくなります。

#### デフォルト **USECHECKPOINTS**



**構文**

```
PURGEOLDEXTRACTS <trail name>
[, USECHECKPOINTS | NOUSECHECKPOINTS]
[, <min rule>]
[, <frequency>]
```

| 引数               | 説明                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <trail name>     | ページするトレイル名。完全修飾名を使用します。                                                                                                                                                                                                                                                                                                                                                          |
| USECHECKPOINTS   | 指定されている MINKEEP ルールに従い、すべての Extract および Replicat プロセスが処理を終了したことがチェックポイントによって示された後、データのページを許可します。                                                                                                                                                                                                                                                                                 |
| NOUSECHECKPOINTS | 次のいずれかの最小値を維持しながら、チェックポイントを考慮せずにページを許可します。 <ul style="list-style-type: none"> <li>◆ MINKEEP ルールが使用されていない場合は 1 ファイル</li> <li>または</li> <li>◆ MINKEEP ルールで指定されたファイル数</li> </ul>                                                                                                                                                                                                     |
| <MINKEEP rule>   | データの最小保持時間を設定する次のいずれかのルールを使用できます。 <p>MINKEEPHOURS &lt;n&gt;</p> <p>少なくとも指定した時間変更されていないファイルを保持します。</p> <p>MINKEEPDAYS &lt;n&gt;</p> <p>少なくとも指定した日数にわたって変更されていないファイルを保持します。</p> <p>MINKEEPFILES &lt;n&gt;</p> <p>少なくとも &lt;n&gt; 個の変更されていないトレイル・ファイル（アクティブなファイルを含む）を保持します。</p>                                                                                                     |
| <frequency>      | 古いトレイル・ファイルをページする間隔を設定します。Manager のデフォルトのメンテナンス・タスクの実行間隔は、CHECKMINUTES パラメータで指定されているとおり 10 分です（136 ページを参照してください）。Manager は、10 分ごとに PURGEOLDEXTRACTS の間隔を確認し、指定された間隔の経過後にページを実行します。<frequency> には、次のいずれかを指定できます。 <p>FREQUENCYMINUTES &lt;n&gt;</p> <p>古いトレイル・ファイルをページする間隔（分）を設定します。デフォルトのページ間隔は 60 分です。</p> <p>FREQUENCYHOURS &lt;n&gt;</p> <p>古いトレイル・ファイルをページする間隔（時間）を設定します。</p> |

**例 1**      状況：トレイル・ファイルとして AA000000、AA000001、および AA000002 が存在します。Replicat は 4 時間にわたって停止しており、いずれのファイルの処理も完了していません。Manager パラメータには、次が含まれています。

```
PURGEOLDEXTRACTS /ggs/dirdat/AA*, USECHECKPOINTS, MINKEEPHOURS 2
```

結果：変更されていないファイルの保持時間が経過しました。しかし、チェックポイントによって、Replicat がこれらのファイルの処理を完了していないことが示されているため、どのファイルもパージされません。

**例 2** 状況：トレイル・ファイルとして AA000000、AA000001、および AA000002 が存在します。Replicat は 4 時間にわたって停止しており、いずれのファイルの処理も完了していません。Manager パラメータには、次が含まれています。

```
PURGEOLDEXTRACTS /ggs/dirdat/AA*, NOUSECHECKPOINTS, MINKEEPHOURS 2
```

結果：ファイルの最小保持時間のルールが満たされているため、すべてのトレイル・ファイルがパージされます。

**例 3** 状況：Replicat および Extract はデータの処理を完了しています。過去 5 時間にわたってトレイル・ファイルへのアクセスは発生していません。トレイル・ファイルとして AA000000、AA000001、および AA000002 が存在します。Manager パラメータには、次が含まれています。

```
PURGEOLDEXTRACTS /ggs/dirdat/AA*, USECHECKPOINTS, MINKEEPHOURS 4, &
MINKEEPFILES 4
```

結果：これは、MINKEEP オプションの使用を 1 つのみにする理由を示す例です。USECHECKPOINTS の要件は満たされているため、AA000002 をパージするかどうかを決定するときに最少数のルールが考慮されます。AA000002 がパージされると、残りのファイルが 2 つになり、MINKEEPFILES ルール違反になります。しかし、MINKEEPFILES および MINKEEPHOURS 両方が指定されているため、MINKEEPFILES は無視されます。5 時間変更されておらず、かつ MINKEEPHOURS の 4 時間の条件も満たすため、このファイルはパージされます。

## PURGEOLDTASKS

**適用対象** Manager

PURGEOLDTASKS パラメータでは、指定した時間の経過後または Extract および Replicat が正常に停止した後に、Extract および Replicat タスクをパージします。次のルールに従って、いつタスクを削除するかを指定できます。

- 特定の時間または日数前に最後に開始されたタスク。開始されたことがないタスクの場合は、このルールの適用基準として作成日時が使用されます。
- 正常に停止したか、開始されたことがないタスク。このルールは、最後に開始された時刻よりも優先されます。このルールを使用して、異常終了したタスクのパージを防止します。

同一の Manager パラメータ・ファイルで利用できる PURGEOLDTASKS パラメータ文は、300 未満です。

**デフォルト** なし

**構文** PURGEOLDTASKS <process> <group name>  
[, <purge option>]  
[USESTOPSTATUS]

| 引数        | 説明                                                                                                            |
|-----------|---------------------------------------------------------------------------------------------------------------|
| <process> | 有効な値： <ul style="list-style-type: none"> <li>◆ EXTRACT</li> <li>◆ REPLICAT</li> <li>◆ ER (両方のプロセス)</li> </ul> |

| 引数             | 説明                                                                                                                                                                       |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <group name>   | グループ名、または複数のグループを指定するワイルドカード。                                                                                                                                            |
| <purge option> | 指定した数の時間または日数にわたって更新されていない場合にタスクをパージします。<br>有効な値：<br><ul style="list-style-type: none"> <li>◆ AFTER &lt;number&gt; DAYS</li> <li>◆ AFTER &lt;number&gt; HOURS</li> </ul> |
| USESTOPSTATUS  | 正常に停止したか、開始されたことがない場合にタスクをパージします。                                                                                                                                        |

**例** 次の例では、少なくとも 3 日間更新されていないすべての **Extract** タスクを削除し、正常に停止したか少なくとも 2 時間更新されていない場合に **test\_rep Replicat** タスクを削除します。

```
PURGEOLDTASKS EXTRACT *, AFTER 3 DAYS
PURGEOLDTASKS REP test_rep, AFTER 2 HOURS, USESTOPSTATUS
```

## RECOVERYOPTIONS

**適用対象** Extract

RECOVERYOPTIONS パラメータでは、**Extract** が再起動時に既存のトレイル・ファイルのコンテンツを上書きするか、再起動後にファイルの既存のデータに新しいレコードを追加するかを制御します。

### 追加モード

デフォルトでは、**Extract** は追加モードで機能します。このモードでは、プロセスの失敗があった場合、リカバリ・マーカーが証跡に書き込まれ、以前のすべてのデータをリカバリの目的で保持するため、**Extract** によってリカバリ・データがファイルに追加されます。

追加モードの場合、**Extract** の初期化で、起動時に証跡に書き込まれた最後の完了済トランザクションのアイデンティティが判別されます。**Extract** はこの情報を使用して、そのトランザクションのコミット・レコードがデータ・ソースで発生した時点でリカバリを終了し、抽出に指定される次のコミット済トランザクションを使用して新しいデータの取得、および証跡への新しいデータの追加を開始します。データ・ポンプまたは **Replicat** は、そのリカバリ・ポイントから読取りを再開します。

### 上書きモード

上書きモードは、リリース 10.0 より前の Oracle GoldenGate のリリースで使用されていた **Extract** リカバリのもう 1 つのバージョンです。これらのバージョンでは、**Extract** は、新しいデータを追加するのではなく、証跡内の最後の書き込みチェックポイントの位置の後に既存のトランザクション・データを上書きします。書き込まれる最初のトランザクションは、データ・ソース内の最後の読取りチェックポイントの位置の後で抽出に指定される最初のトランザクションです。

上書きモードでは、上書き前と同じレコード・イメージが上書き後に精密に正確な順序で配置されない可能性があります。この変化は、大きなオブジェクトのフェッチの実行が必要な場合や、**Extract** の構成パラメータが変更されたときに発生する可能性があります。上書きアクティビティが開始されてからトレイル・ファイルの最後に到達するまでの間に、**Extract** 処理に変更がある場合、ファイルの上書き部分の先端にずれが生じ、結果として再び書き込まれた最後のレコードの末尾が、以前の **Extract** インスタンスに書き込まれたレコードの最初の部分と共有する境界に配置されなくなります。1 つのトレイ

ル・レコードから次のトレイル・レコードにかけての読取りを試行する Replicat またはデータ・ポンプは、破損したレコードの途中で到達すると、エラーとともに異常終了します。

## 推奨事項

Oracle Support のアナリストに指示されないかぎり、RECOVERYOPTIONS をデフォルトから変更しないでください。Extract がトレイルの既存データの上書きを許可されている場合、Oracle GoldenGate によって失敗後のリカバリは困難になり、ターゲットに送信する必要があるデータが失われる可能性があります。

一部のケースでは、ターゲットで使用されている Oracle GoldenGate のリリースが Oracle GoldenGate リリース 10 よりも古い場合、Extract は下位互換性をサポートするために自動的に上書きモードに戻ります。古いバージョンでは、追加モードはサポートされていません。

## パラメータの依存関係

RECOVERYOPTIONS パラメータと EXTTRAIL、RMTTRAIL、EXTFILE および RMTFILE の FORMAT オプション間には依存性が存在します。RECOVERYOPTIONS が APPENDMODE に設定される場合、FORMAT オプションは RELEASE 10.0 以上に設定される必要があります。RECOVERYOPTIONS が OVERWRITEMODE に設定される場合、FORMAT オプションは RELEASE 9.5 以下に設定される必要があります。

**デフォルト** APPENDMODE  
**構文** RECOVERYOPTIONS {APPENDMODE | OVERWRITEMODE}

| 引数            | 説明                                        |
|---------------|-------------------------------------------|
| APPENDMODE    | 新しいデータをトレイル・ファイルの既存のデータに追加します。これはデフォルトです。 |
| OVERWRITEMODE | 最新のチェックポイントの位置から、古いデータを新しいデータで上書きします。     |

**例** RECOVERYOPTIONS OVERWRITEMODE

# REPEROR

**適用対象** Replicat

REPEROR パラメータでは、Replicat のエラーへの対応方法を制御します。1 つの REPEROR 文を使用して大半のエラーをデフォルトの方法で処理しながら、1 つまたは複数の別の REPEROR 文を使用して、特定のエラーを別の方法で処理できます。たとえば、重複レコード・エラーは無視し、他のすべてのケースでは処理を異常終了させることができます。

次に示す構文では、<error> および <response> をカッコ内に指定する必要があることに注意してください。次に例を示します。

```
REPEROR (DEFAULT, ABEND)
REPEROR (-1, IGNORE)
```

ただし、RESET オプションはカッコ内に含めることができます。

```
REPEROR RESET
```

**デフォルト** TRANSABORT (デッドロックの場合)、ABEND (その他すべての場合)

```
構文
REPERORR { (
{DEFAULT | DEFAULT2 | <SQL error> | <user-defined error>},
{ABEND | DISCARD | EXCEPTION | IGNORE |
RETRYOP [MAXRETRIES <n>] |
TRANSABORT [, MAXRETRIES] [, DELAYSECS <n> | DELAYCSECS <n>]
}) |
RESET }
```

| 引数                             | 説明                                                                                                                                                                                                                                   |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>処理するエラーを指定するオプション:</b>      |                                                                                                                                                                                                                                      |
| DEFAULT                        | 明示的な REPERORR 文が指定されているエラーを除くすべてのエラーに対するグローバルなレスポンスを設定します。                                                                                                                                                                           |
| DEFAULT2                       | DEFAULT のレスポンスが EXCEPTION に設定されている場合に、バックアップのデフォルト・アクションを提供します。DEFAULT2 は、エラーの発生が予想される MAP 文に例外 MAP 文が指定されていないときに使用します。                                                                                                              |
| <SQL error>                    | SQL エラー番号。                                                                                                                                                                                                                           |
| <user-defined error>           | MAP 文の FILTER 句の RAISEERROR オプションで指定されているユーザー定義エラー。                                                                                                                                                                                  |
| <b>エラーに対するレスポンスを指定するオプション:</b> |                                                                                                                                                                                                                                      |
| ABEND                          | トランザクションをロールバックし、処理を異常終了します。ABEND はデフォルトです。                                                                                                                                                                                          |
| DISCARD                        | エラーを破棄ファイルに記録しますが、このトランザクションおよび後続のトランザクションの処理を継続します。                                                                                                                                                                                 |
| EXCEPTION                      | エラーを例外として処理します。エラーの発生が予想される場合は、エラー発生後にのみ実行される例外 MAP 文を作成できます。たとえば、この MAP 文を使用して、失敗した更新文の列を "行方不明の更新" 表にマップできます。パラメータ・ファイルでは、エラーの発生が予想される MAP 文の後ろに例外 MAP 文を指定します。<br>エラー処理の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。 |
| IGNORE                         | エラーを無視します。                                                                                                                                                                                                                           |
| RETRYOP<br>[MAXRETRIES <n>]    | 操作を再試行します。MAXRETRIES オプションでは、再試行回数を制御します。たとえば、表がエクステンツ不足の場合、RETRYOP と MAXRETRIES によって、トランザクションの失敗を防ぐためにエクステンツを追加する時間を確保できます。指定した MAXRETRIES 回数の後、Replicat は異常終了します。試行間隔を設定するには、284 ページで説明されている RETRYDELAY を設定します。                    |

| 引数                                                                        | 説明                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TRANSABORT<br>[, MAXRETRIES <n>]<br>[, DELAYSECS <n>   DELAYCSECS<br><n>] | <p>トランザクションを中止し、<b>Replicat</b> をトランザクションの開始位置に再配置します。この動作は、レコードの処理が成功するか、<b>MAXRETRIES</b> が終了するまで継続されます。<b>MAXRETRIES</b> が設定されていない場合、<b>TRANSABORT</b> アクションは何度も繰り返されます。</p> <p><b>DELAY</b> オプションでは、再試行を遅延させます。デフォルトの遅延は 60 秒です。</p> <p><b>TRANSABORT</b> オプションは、タイムアウトやデッドロックをサポートしているデータベースで、このような状況に対処するときに活用できます。</p> |
| RESET                                                                     | <p>パラメータ・ファイルのこれ以前の <b>REPERERROR</b> 文で指定されているすべてのエラー・ルールを削除します。すべてのエラーへのレスポンスをデフォルトの <b>ABEND</b> に戻します。<b>Replicat</b> は、トランザクションをロールバックし、処理を異常終了します。このオプションは、<b>MAP</b> 文の <b>REPERERROR</b> とともに使用することはできません。</p>                                                                                                        |

**例 1** 次に、大半のエラーでは処理を停止し、重複レコード・エラーは無視する方法の例を示します。

```
REPERERROR (DEFAULT, ABEND)
REPERERROR (-1, IGNORE)
```

**例 2** 次に、**account** 表のエラーを処理するために作成された例外 **MAP** 文を呼び出す例を示します。**product** 表には例外 **MAP** 文が定義されていないため、この表のエラーによって **Replicat** は異常終了します。

```
REPERERROR (DEFAULT, EXCEPTION)
REPERERROR (DEFAULT2, ABEND)
MAP sales.product, TARGET sales.product;
MAP sales.account, TARGET sales.account;
INSERTALLRECORDS
MAP sales.account, TARGET sales.account_exception,
EXCEPTIONSONLY,
COLMAP (account_no = account_no,
optype = @GETENV ("lasterr", "optype"),
dberr = @GETENV ("lasterr", "dberrnum"),
dberrmsg = @GETENV ("lasterr", "dberrmsg"));
```

**例 3** 次に、最初の **MAP** 文にエラー・ルールを適用した後、2 つ目の文ではデフォルトの **ABEND** に戻す例を示します。

```
REPERERROR (-1, IGNORE)
MAP sales.product, TARGET sales.product;
REPERERROR RESET
MAP sales.account, TARGET sales.account;
```

# REPFETCHEDCOLOPTIONS

**適用対象**      Replicat

REPFETCHEDCOLOPTIONS パラメータでは、Replicat によるソース・データベースからのフェッチが必要とされていた操作に対応する方法を制御します。Extract プロセスは、トランザクション・レコードに SQL 文を構築するための十分な情報が含まれていないとき、または FETCHCOLS 句が使用されている (332 ページを参照してください) ときに、表または UNDO 表領域 (Oracle 9i 以上) から列データをフェッチします。

**デフォルト**      なし

**構文**      REPFETCHEDCOLOPTIONS  
[ , INCONSISTENTROW]  
[ , LATESTROWVERSION {IGNORE | REPORT | DISCARD | ABEND}]  
[ , MISSINGROW {IGNORE | REPORT | DISCARD | ABEND}]  
[ , NOFETCH <action>]  
[ , REDUNDANTROW]  
[ , SETIFMISSING [<string>]]  
[ , SNAPSHOTROW]

| 引数                           | 説明                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INCONSISTENTROW              | 行 ID による列データのフェッチは成功したものの、キーが一致しなかったことを示します。行 ID が再利用されたか、この操作の後 (およびフェッチの前) に主キー更新が行われました。デフォルトでは、Replicat はこの行を破棄ファイルに記録し、以降のデータ処理を継続します。                                                                                                                                                                                                                                                                     |
| LATESTROWVERSION<br><action> | <p>表の現在の行から列データがフェッチされた場合のレスポンスを指定します。有効な値:</p> <p>IGNORE                      状況が無視して処理を継続します。</p> <p>REPORT                      状況と行のコンテンツを破棄ファイルにレポートしますが、行の処理を継続します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</p> <p>DISCARD                      データを破棄し、行は処理しません。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</p> <p>ABEND                      データを破棄し、処理を中止します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</p> |
| NOFETCH <action>             | <p>フェッチを防ぎます。このオプションの 1 つの用途は、データベースがスタンドバイで、Oracle GoldenGate がデータベースに接続していないときです。このケースでは、データベースからのフェッチが試行されるとエラーが発生します。他のシナリオでも、このパラメータの使用が必要になることがあります。</p> <p>Oracle GoldenGate が通常フェッチしているデータをフェッチできない場合、ターゲットでデータ整合性の問題が発生する可能性があります。</p>                                                                                                                                                                 |

| 引数                         | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------------------|-----------------|------------------------------------------------------------------------------------|---------------------|---------------------------------------------------------------|-----------|-----------------------------------------------------------|
|                            | <p>次に、NOFETCH が検出された場合に実行可能な有効なアクションを示します。</p> <ul style="list-style-type: none"> <li>◆ ABEND: 操作を破棄ファイルに書き込み、Replicat プロセスを異常終了させます。これはデフォルトです。</li> <li>◆ ALLOW: レコード長が 0 でないかぎり、操作を処理します。</li> <li>◆ IGNORE: 操作を無視します。（STATOPTIONS 設定に基づいて）フェッチ統計がプロセス・レポートにレポートされている場合、レポートはこの結果によって更新されます。</li> <li>◆ REPORT: レコードを破棄ファイルに書き込み、操作を処理します。</li> <li>◆ DISCARD: レコードを破棄ファイルに書き込み、操作を処理しません。（STATOPTIONS 設定に基づいて）フェッチ統計がプロセス・レポートにレポートされている場合、レポートはこの結果によって更新されます。</li> </ul>                                                                                             |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| MISSINGROW <action>        | <p>Replicat が処理で行の一部（変更された値）のみ使用できる場合のレスポンスを指定します。トレイルで行方不明の列データは、変更レコードが作成されてからフェッチがトリガーされるまでの間に行が削除されてしまっているか、必要な行イメージが指定されている UNDO 保存期間以前ののものであったため、通常はフェッチできません。</p> <p>有効な値：</p> <table> <tr> <td>IGNORE</td><td>状況は無視して処理を継続します。</td></tr> <tr> <td>REPORT</td><td>状況と行のコンテンツを破棄ファイルにレポートしますが、部分的な行の処理を継続します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</td></tr> <tr> <td>DISCARD</td><td>データを破棄し、部分的な行は処理しません。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</td></tr> <tr> <td>ABEND</td><td>データを破棄し、処理を中止します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。</td></tr> </table> | IGNORE | 状況は無視して処理を継続します。 | REPORT          | 状況と行のコンテンツを破棄ファイルにレポートしますが、部分的な行の処理を継続します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。 | DISCARD             | データを破棄し、部分的な行は処理しません。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。 | ABEND     | データを破棄し、処理を中止します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。 |
| IGNORE                     | 状況は無視して処理を継続します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| REPORT                     | 状況と行のコンテンツを破棄ファイルにレポートしますが、部分的な行の処理を継続します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| DISCARD                    | データを破棄し、部分的な行は処理しません。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| ABEND                      | データを破棄し、処理を中止します。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| REDUNDANTROW               | このレコードの列データがフェッチ済のため、列データがフェッチされなかったことを示します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| SETIFMISSING<br>[<string>] | <p>フェッチは成功しなかった（およびトレイル・レコードで値が行方不明である）ものの、ターゲット列が Not NULL 制約を持つ場合に、値を指定します。CHAR および BINARY データ型の値としてオプションの ASCII 文字列を受け付けるか、次のデフォルトを受け付けます。</p> <table> <tr> <th>列</th><th>デフォルト</th></tr> <tr> <td>CHAR<br/>VARCHAR</td><td>単一の空白</td></tr> <tr> <td>BINARY<br/>VARBINARY</td><td>NULL バイト</td></tr> <tr> <td>TIMESTAMP</td><td>現在の日付 / 時刻</td></tr> </table>                                                                                                                                                                                                           | 列      | デフォルト            | CHAR<br>VARCHAR | 単一の空白                                                                              | BINARY<br>VARBINARY | NULL バイト                                                      | TIMESTAMP | 現在の日付 / 時刻                                                |
| 列                          | デフォルト                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| CHAR<br>VARCHAR            | 単一の空白                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| BINARY<br>VARBINARY        | NULL バイト                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |
| TIMESTAMP                  | 現在の日付 / 時刻                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |        |                  |                 |                                                                                    |                     |                                                               |           |                                                           |



| 引数 | 説明                                                                                                                                                                                                                                    |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>FLOAT                      ゼロ</p> <p>INTEGER</p> <p>SETIFMISSING に加え、MAP 文の COLMAP 句を使用してターゲット列に値をマップできます。224 ページを参照してください。</p> <p>SNAPSHOTROW              列データがスナップショットからフェッチされたことを示します。通常、このオプションは操作のレポートまたは破棄のためにのみ使用されます。</p> |

## REPLACEBADCHAR

|              |                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>適用対象</b>  | Extract および Replicat                                                                                                                            |
|              | REPLACEBADCHAR パラメータでは、文字用の列をマップするときに検出される無効な文字データの代替値を指定します。代替値が指定されていない場合、出力不可能な文字が含まれる文字用の列は、16 進数文字列として出力されます。REPLACEBADCHAR はグローバルに適用されます。 |
| <b>デフォルト</b> | NONE                                                                                                                                            |
| <b>構文</b>    | REPLACEBADCHAR {<char>   SPACE   NULL   UNPRINTABLE   NONE}                                                                                     |

| 引数          | 説明                                                  |
|-------------|-----------------------------------------------------|
| <char>      | 指定した文字に置換します。                                       |
| SPACE       | 空白に置換します。                                           |
| NULL        | ターゲット列が NULL 値を受け付ける場合は NULL に置換し、それ以外の場合は空白に置換します。 |
| UNPRINTABLE | 無効なデータを含むすべての列を拒否します。                               |
| NONE        | ダブルバイト文字セット値からデフォルト文字への変換を抑制します。これはデフォルトです。         |

- 例 1**              次に、無効な文字を空白に置換する例を示します。
- REPLACEBADCHAR SPACE
- 例 2**              次に、出力不可能な文字をカレット記号に置換する例を示します。
- REPLACEBADCHAR ^

## REPLACEBADNUM

**適用対象** Replicat

REPLACEBADNUM パラメータでは、数字列をマップするときに検出される無効な数字データの代替値を指定します。REPLACEBADNUM はグローバルに適用されます。

**デフォルト** 無効な数字をゼロに置換する。

**構文** REPLACEBADNUM {<number> | NULL | UNPRINTABLE}

| 引数          | 説明                                                  |
|-------------|-----------------------------------------------------|
| <number>    | 指定した数字に置換します。                                       |
| NULL        | ターゲット列が NULL 値を受け付ける場合は NULL に置換し、それ以外の場合はゼロに置換します。 |
| UNPRINTABLE | 出力不可能なデータを含むすべての列を拒否します。プロセスは停止し、不正な値をレポートします。      |

**例 1** REPLACEBADNUM 1

**例 2** REPLACEBADNUM NULL

## REPLICAT

**適用対象** Replicat

REPLICAT パラメータでは、オンライン変更同期用の Replicat グループを指定します。このパラメータによって、現在の実行と前回の実行が関連付けられるので、データ変更を継続的に処理し、ソース表とターゲット表の同期性を維持できます。Replicat は継続的に実行し、データ・ソースおよびトレイルにチェックポイントを保持することにより、計画的または計画外のプロセス終了、システム停止、ネットワーク障害が発生した場合にもデータ整合性とフォルト・トレランスを確保します。

Replicat パラメータ・ファイルには REPLICAT または SPECIALRUN のいずれかの指定が必要で、パラメータ・ファイルの最初のエントリにする必要があります。SPECIALRUN では、バッチ実行を指定します。305 ページを参照してください。

**デフォルト** なし

**構文** REPLICAT <group name>

| 引数           | 説明                           |
|--------------|------------------------------|
| <group name> | ADD REPLICAT コマンドで定義したグループ名。 |

**例** REPLICAT finance

# REPORT

**適用対象** Extract および Replicat

REPORT パラメータでは、Extract または Replicat がプロセス・レポートにいつ一時的な実行時統計を生成するかを制御します。統計は既存のレポートに追加されます。デフォルトでは、実行時統計は、プロセスが意図的に中断されないかぎり実行終了時に表示されます。

REPORT のレポートは、前回のレポートから繰り越されます。たとえば、プロセスが初日に 1000 万の挿入、2 日目に 2000 万の挿入を実行し、レポートが毎日 3:00 に生成される場合、最初のレポートは最初の 1000 万の挿入をレポートし、次のレポートは前日の 1000 万件と当日の 2000 万件の合計、3000 万件をレポートします。新しいレポートが生成されたときに統計をリセットするには、STATOPTIONS パラメータと RESETREPORTSTATS オプションを使用します。311 ページを参照してください。

プロセス・レポート使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** 各実行の終了時に実行時統計を生成する。

**構文**

```
REPORT
{AT <hh:mi> |
ON <day> |
AT <hh:mi> ON <day>}
```

| 引数         | 説明                                                                                                                                                                            |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AT <hh:mi> | 指定の時刻にレポートを生成します。AT を ON なしで指定すると、指定した時刻に毎日レポートが生成されます。                                                                                                                       |
| ON <day>   | 指定の曜日にレポートを生成します。有効な値： <div>SUNDAY</div> <div>MONDAY</div> <div>TUESDAY</div> <div>WEDNESDAY</div> <div>THURSDAY</div> <div>FRIDAY</div> <div>SATURDAY</div> 大 / 小文字は区別されません。 |

**例 1** REPORT AT 17:00

**例 2** REPORT ON SUNDAY AT 1:00

# REPORTCOUNT

**適用対象** Extract および Replicat

REPORTCOUNT パラメータを使用して、起動後に Extract または Replicat で処理されたトランザクション・レコードの件数をレポートに表示できます。各トランザクション・レコードは、Oracle GoldenGate で取得されたトランザクション内で実行された論理データベース操作を示します。レコード件数は、レポート・ファイルに出力および画面上に表示されます。

**注意** この数は、Oracle GoldenGate トレイルに含まれているレコード数とは異なる場合があります。データが 4K を上回る操作は、複数のトレイル・レコードに保持する必要があります。したがって、レコード数は 1,000 レコード（データベース操作）を示していても、トレイル数ではそれより多くのレコード数を示す場合があります。トレイルのレコード数を取得するには、Logdump ユーティリティを使用します。

レコード数のレポートは、定期的な間隔、または特定のレコード数に到達後の生成をスケジュールできます。レコード数は、レポートからレポートに繰り越されます。

REPORTCOUNT は、パラメータ・ファイルで 1 回のみ使用できます。REPORTCOUNT の複数のインスタンスがある場合、Oracle GoldenGate は最後に指定されているインスタンスを使用します。

**デフォルト** なし

**構文** REPORTCOUNT [EVERY] <count>  
{RECORDS | SECONDS | MINUTES | HOURS} [, RATE]

| 引数                                     | 説明                                                                                                                                                                                                                   |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <count>                                | レコード数を出力する間隔。                                                                                                                                                                                                        |
| RECORDS   SECONDS   MINUTES<br>  HOURS | <count> の測定単位（レコード、秒、分、または時間）。                                                                                                                                                                                       |
| RATE                                   | 1 秒当たりの操作数、およびパフォーマンス指標として変更レートをレポートします。例 2 を参照してください。"rate" 統計は、レコード数合計をプロセス起動からの合計時間で割った値です。"delta" 統計は、最後のレポート以降のレコード数を最後のレポート出力時からの時間で割った値です。<br><br>注意：この計算では、マイクロ秒の粒度が使用されます。時間間隔では秒の端数なしで表示され、レート値は整数で表示されます。 |

**例 1** 次に、5,000 レコードごとにレコード数を生成する例を示します。

```
REPORTCOUNT EVERY 5000 RECORDS
```

**例 2** 次に、10 分間隔でレコード数のレポートを生成し、処理統計もレポートする例を示します。

```
REPORTCOUNT EVERY 10 MINUTES, RATE
```

処理統計は次のようになります。

```
12000 records processed as of 2006-08-31 12:27:40 (rate 203,delta 308)
```

# REPORTROLLOVER

**適用対象** Extract および Replicat

REPORTROLLOVER をパラメータを使用して、プロセスの開始時のかわりに定期的にレポート・ファイルを非アクティブにすることができます。長期または連続的な実行の場合、エージング・スケジュールを設定することにより、アクティブなレポート・ファイルのサイズを制御し、アーカイブ・ルーチンに含める予期可能なアーカイブ・セットを得ることができます。

**注意** レポート統計は、レポートからレポートに繰り越されます。新しいレポートで統計をリセットするには、STATOPTIONS パラメータと RESETREPORTSTATS オプションを使用します。

時刻、曜日、またはその両方を指定できます。時刻（AT オプション）のみで曜日（ON オプション）を指定しない場合、指定した時刻に毎日レポートが生成されます。

このパラメータによって発生するロールオーバーでは、プロセス・レポートに実行時統計が生成されません。

- レポート・ファイルに実行時統計をいつ生成するかを制御するには、REPORT パラメータを使用します。
- 新しい実行時統計をオンデマンドで生成するには、SEND EXTRACT または SEND REPLICAT コマンドとともに REPORT オプションを使用します。

**デフォルト** 起動時にレポートをロールオーバーする。

**構文**  
REPORTROLLOVER  
{AT <hh:mi> |  
ON <day> |  
AT <hh:mi> ON <day>}

| 引数         | 説明                                                                                                                                                                |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AT <hh:mi> | <p>ファイルをエージングする時刻。</p> <p>有効な値：</p> <ul style="list-style-type: none"> <li>◆ hh は 24 時間表記の時間で、1 ～ 23 までの値を受け付けます。</li> <li>◆ mi は 00 ～ 59 までの値を受け付けます。</li> </ul> |
| ON <day>   | <p>ファイルをエージングする曜日。有効な値：</p> <p>SUNDAY<br/>MONDAY<br/>TUESDAY<br/>WEDNESDAY<br/>THURSDAY<br/>FRIDAY<br/>SATURDAY</p> <p>大 / 小文字は区別されません。</p>                       |

**例 1** REPORTROLLOVER AT 05:30

**例 2** REPORTROLLOVER ON friday

**例 3** REPORTROLLOVER AT 05:30 ON friday

## RESTARTCOLLISIONS | NORESTARTCOLLISIONS

**適用対象** Replicat

RESTARTCOLLISIONS および NORESTARTCOLLISIONS パラメータでは、競合が原因で Oracle GoldenGate が停止した後に、Replicat が HANDLECOLLISIONS ロジックを適用するかどうかを制御します。デフォルトでは、NORESTARTCOLLISIONS が適用されます。ただし、Oracle GoldenGate 起動後の最初のトランザクションに HANDLECOLLISIONS ロジックを適用する必要がある場合もあります。たとえば、サーバーが強制的にシャット・ダウンされ、データベースが最後の Replicat トランザクションをコミットしたものの、Oracle GoldenGate が確認を受信していない場合です。このケースでは、Replicat は起動時にトランザクションを再試行します。HANDLECOLLISIONS は、結果として発生するエラーを自動的に処理します。

RESTARTCOLLISIONS は、最初の Replicat チェックポイント（トランザクション）が完了するまで HANDLECOLLISIONS 機能を有効化します。パラメータ・ファイルに HANDLECOLLISIONS パラメータを指定する必要はありません。最初のチェックポイントの完了後、HANDLECOLLISIONS は自動的に無効化されます。

HANDLECOLLISIONS の詳細は、204 ページを参照してください。

**デフォルト** NORESTARTCOLLISIONS

**構文** RESTARTCOLLISIONS | NORESTARTCOLLISIONS

## RETRYDELAY

**適用対象** Replicat

RETRYDELAY パラメータでは、失敗した操作を再試行するまでの間隔を指定します。このパラメータは、REPERERROR パラメータの RETRYOP オプションを使用しているときに使用します（274 ページを参照してください）。

**デフォルト** 60 秒

**構文** RETRYDELAY <seconds>

| 引数        | 説明           |
|-----------|--------------|
| <seconds> | 再試行までの間隔（秒）。 |

**例** REPERERROR (100, RETRYOP MAXRETRIES 3) RETRYDELAY 30

## RMTFILE

**適用対象** Extract

RMTFILE パラメータでは、抽出データを書き込むリモート・システム上の抽出ファイル名を定義します。このパラメータはバッチ処理で使用します。オンライン変更同期では、RMTTRAIL パラメータを使用します。

Solaris システムでは、Extract または Replicat によって処理される場合、抽出ファイルのサイズの上限は 2GB です。ネイティブ・ロード・ユーティリティなどの別のアプリケーションによって処理される場合は、抽出ファイルのサイズをさらに大きくできます。

RMTFILE は、RMTHOST 文の後に指定する必要があります、すべての TABLE 文より先に指定する必要があります。

## ファイル・バージョンングについて

すべての Oracle GoldenGate プロセスは分離されるので、異なる Oracle GoldenGate リリースのものである可能性があることから、各証跡ファイルまたは抽出ファイルでは、バージョンがファイル・ヘッダー内に格納されています。デフォルトでは、証跡のバージョンは、ファイルを作成したプロセスの現在のバージョンになります。証跡のバージョンを設定するには、EXTTRAIL、EXTTRAIL、RMTTRAIL または RMTFILE パラメータの FORMAT オプションを使用します。

異なる Oracle GoldenGate プロセス・バージョンの上位および下位互換性が保証されるようにするには、ファイルのヘッダー・フィールドは標準のトークン形式で記述される必要があります。プロセスの新しいバージョンで作成される新しいトークンは、古いバージョンでは無視されるようにできるので、下位互換性が保証されます。同様に新しい Oracle GoldenGate バージョンでは古いトークンがサポートされます。さらに、トークンが新しいプロセス・バージョンで推奨されない場合、デフォルト値がトークンに割り当てられるので、古いバージョンは引き続き適切に機能します。ファイル・バージョンを指定するトークンは COMPATIBILITY で、Logdump ユーティリティで表示可能です。また、@GETENV 関数の GGFILEHEADER オプションで取得することもできます。

証跡または抽出ファイルには、それらが読み込まれるプロセスのバージョン以下のバージョンが必要です。それ以外の場合はプロセスが異常終了します。さらに、Oracle GoldenGate では、データ・ポンプの出力証跡またはファイルのバージョンが、入力証跡またはファイルのバージョンと同じであることが強制されます。再起動の際、Extract では新しいファイルが証跡にロールされるので、各ファイルが 1 つのバージョンのみ（ファイルが空の場合を除く）であることが保証されます。

## パラメータの依存関係

RECOVERYOPTIONS パラメータと EXTTRAIL、RMTTRAIL、EXTFILE および RMTFILE の FORMAT オプション間には依存性が存在します。RECOVERYOPTIONS が APPENDMODE に設定される場合、FORMAT オプションは RELEASE 10.0 以上に設定される必要があります。RECOVERYOPTIONS が OVERWRITEMODE に設定される場合、FORMAT オプションは RELEASE 9.5 以下に設定される必要があります。

**デフォルト** なし

**構文**

```
RMTFILE <file name>
[, APPEND]
[, PURGE]
[, MAXFILES <number>]
[, MEGABYTES <megabytes>]
[, FORMAT RELEASE <major>.<minor>]
```

| 引数          | 説明                                                             |
|-------------|----------------------------------------------------------------|
| <file name> | ファイルの完全修飾名。                                                    |
| APPEND      | 現在のデータをファイルの既存のデータに追加します。APPEND を使用する場合、PURGE は使用しないでください。     |
| PURGE       | 新しいファイルを作成する前に、既存のファイルを削除します。PURGE を使用する場合、APPEND は使用しないでください。 |

| 引数                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAXFILES <number>                 | <p>単一のファイルでなく、連続するファイルを作成します。このオプションは、1つのファイルのサイズがオペレーティング・システムに許可される制限を越える可能性があるときに使用します。</p> <p>MAXFILES を使用して、必要な数のファイルを作成できます。エージングされたファイルには、6桁の順序番号が付けられます（例：datafile000002）。</p> <p>MAXFILES を使用するときは、MEGABYTES も使用して、連続ファイル内の各ファイルの最大サイズを設定します。</p> <p>チェックポイントは、これらのファイルには保持されません。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| MEGABYTES <megabytes>             | <p>ファイル（MAXFILES を使用している場合は各ファイル）の最大サイズを定義します。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| FORMAT RELEASE<br><major>.<minor> | <p>Extract により証跡、ファイルまたは（リモート・タスクの場合）別のプロセスに送信されるデータのメタデータ形式を指定します。形式は Oracle GoldenGate プロセスのリリースに依存します。古い Oracle GoldenGate リリースには新しいものよりも含まれるメタデータの数が少ない、または異なります。メタデータによって、データ・レコードがサポートされるリリースのものであるかどうかプロセスに伝達されます。</p> <ul style="list-style-type: none"> <li>◆ FORMAT は、必須のキーワードです。</li> <li>◆ RELEASE では Oracle GoldenGate リリースが指定されます。&lt;major&gt; はメジャー・リリース番号で、&lt;minor&gt; はマイナー・リリース番号です。有効な値は 9.0 から現在の Oracle GoldenGate リリース番号の間です。（9.0 より前の Oracle GoldenGate リリースを使用している場合は、9.0 か 9.5 のいずれかを指定します。）プログラムとして、リリースは適切な証跡形式の互換性レベルにマッピングされます。デフォルト値は、この証跡に書き込まれるプロセスの現在のリリースです。</li> </ul> <p>追加モードが Extract で有効でない場合、デフォルトのファイルのバージョン番号は自動的に 9（Oracle GoldenGate リリース 9.x）になります。</p> <p>Extract モードは、RECOVERYOPTIONS パラメータで制御されます。</p> <p>証跡または抽出ファイルには、それらが読み込まれるプロセスのバージョン以下のバージョンが必要です。それ以外の場合はプロセスが異常終了します。さらに、Oracle GoldenGate では、データ・ポンプの出力証跡またはファイルのバージョンが、入力証跡またはファイルのバージョンと同じであることが強制されます。再起動の際、Extract では新しいファイルが証跡にロールされるので、各ファイルが 1つのバージョンのみ（ファイルが空の場合を除く）であることが保証されます。</p> |

**例 1** RMTFILE /ggs/dirdat/salesny, MEGABYTES 3, PURGE

**例 2** RMTFILE /ggs/dirdat/salesny, MEGABYTES 3, FORMAT RELEASE 10.4



# RMTHOST

**適用対象** Extract

RMTHOST パラメータでは、Manager プロセスが実行されているリモート・システム、およびこのシステム上の TCP/IP ポート番号を特定し、Oracle GoldenGate によって確立されたソースおよびターゲット・システム間の TCP/IP 接続の様々な属性を制御します。このパラメータでは、圧縮、データ暗号化、バッファ属性、接続タイムアウトしきい値、および接続リクエストの待機時間を制御します。Collector パラメータの設定にも使用できます。

パラメータ・ファイルで複数のリモート・システムを特定するには、次に示す例のように、各システムに対して 1 つの RMTHOST 文を指定し、関連するトレイルおよび表マップの指定を続けます。

```
EXTRACT sales
USERID ggs, PASSWORD ggs123
RMTHOST ny, MGRPORT 7888
RMTTRAIL /ggs/dirdat/aa
TABLE ora.orders;
RMTHOST la, MGRPORT 7888
RMTTRAIL /ggs/dirdat/bb
TABLE ora.orders;
```

パッシブ・モードで作成されている Extract には、RMTHOST を使用しないでください。パッシブ Extract の詳細は、17 ページを参照してください。

## 最適なバッファ・サイズの決定

TCPBUFSIZE オプションでは、大きなパケット・サイズをターゲット・システムに送信できるように、Extract が保持を試みる TCP ソケット・バッファのサイズを制御します。お使いのネットワークに最適なバッファ・サイズの決定のガイドラインとして、次の式を使用できます。

1. コマンド・シェルから ping コマンドを実行し、次の例に示すような平均ラウンド・トリップ時間 (RTT) を取得します。

```
C:\home\ggs>ping ggsoftware.com
Pinging ggsoftware.com [192.168.116.171] with 32 bytes of data:
Reply from 192.168.116.171: bytes=32 time=31ms TTL=56
Reply from 192.168.116.171: bytes=32 time=61ms TTL=56
Reply from 192.168.116.171: bytes=32 time=32ms TTL=56
Reply from 192.168.116.171: bytes=32 time=34ms TTL=56
Ping statistics for 192.168.116.171:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 31ms, Maximum = 61ms, Average = 39ms
```

2. この値にネットワーク帯域幅を掛けます。たとえば、平均RTTが0.08秒で帯域幅が100MB/秒の場合の最適なバッファ・サイズは次のようになります。

$0.08 \text{ second} * 100 \text{ megabits per second} = 8 \text{ megabits}$

3. この結果を 8 で割り、バイト数を決定します (8 ビットから 1 バイトに変換します)。次に例を示します。

$8 \text{ megabits} / 8 = 1 \text{ megabyte per second}$

TCPBUFSIZE で必要な単位はバイトなので、この値を 1000000 に設定します。

Windows 以外のシステムの最大ソケット・バッファ・サイズは、通常デフォルトで制限されています。Oracle GoldenGate が TCPBUFSIZE で構成するバッファ・サイズを増やすことができるように、システム管理者にソースおよびターゲット・システムのデフォルト値を増やすように依頼してください。

**注意** パフォーマンスが改善するのは、ターゲットの Oracle GoldenGate リリースが 8.0.4 以上の場合のみです。

**デフォルト** なし

**構文**

```
RMTHOST
{<host name> | <IP address>}
[, MGRPORT <port> | PORT <port>]
[, COMPRESS]
[, COMPRESSTHRESHOLD]
[, ENCRYPT {NONE | BLOWFISH}]
[, KEYNAME <keyname>]
[, PARAMS <collector parameters>]
[, TCPBUFSIZE <bytes>]
[, TCPFLUSHBYTES <bytes>]
[, TIMEOUT <seconds>]
```

| 引数                           | 説明                                                                                                                                                                                                                                                                                                                             |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {<host name>   <IP address>} | ターゲット・システムの DNS 名または IP アドレス。いずれか 1 つを使用してホストを定義できます。                                                                                                                                                                                                                                                                          |
| COMPRESS                     | 送信レコードのブロックを圧縮して、必要な帯域幅を低減します。Oracle GoldenGate は、トレイルに書き込む前にデータを解凍します。COMPRESS での通常の圧縮率は少なくとも 4:1 で、それより高くなる場合もあります。ただし、データを圧縮すると CPU リソースを消費します。                                                                                                                                                                             |
| COMPRESSTHRESHOLD            | 圧縮を行うための最小ブロック・サイズを設定します。有効な値は、0 ～ 28000 です。デフォルトは 1,000 バイトです。                                                                                                                                                                                                                                                                |
| ENCRYPT<br>{NONE   BLOWFISH} | <p>ターゲット・システムに TCP/IP を介して送信されるデータ・ストリームを暗号化します。</p> <ul style="list-style-type: none"> <li>◆ NONE では暗号化なし（デフォルト）を指定します。</li> <li>◆ BLOWFISH では、Blowfish 暗号化を指定します。KEYNAME オプションを使用する必要があります。データはターゲット・システム上で自動的に復号化されるため、復号化パラメータは不要です。</li> </ul> <p>データ暗号化の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p> |
| KEYNAME <keyname>            | ENCKEYS ファイルの鍵名。Oracle GoldenGate は、この鍵を使用してデータを復号化します。ターゲット・システムの ENCKEYS ファイルに一致する鍵名が存在しない場合、Oracle GoldenGate は異常終了します。ENCKEYS ファイルの作成の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。                                                                                                                            |
| MGRPORT <port>               | Manager が実行されているリモート・システム上のポート番号。MGRPORT または PORT の指定が必要です。                                                                                                                                                                                                                                                                    |

| 引数                               | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PORT <port>                      | 静的 Collector プロセスのポート番号。Manager ポート（動的 Collector を使用している場合）または静的 Collector ポート番号の指定が必要です。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| PARAMS<br><collector parameters> | <p>NonStop ターゲット・システム上の Collector パラメータを指定します。</p> <p>注意：Manager が Collector を動的に起動する場合は、Collector ポート（-p 引数）を指定しないでください。</p> <p>NonStop プラットフォームの Collector パラメータの詳細は、Oracle GoldenGate <i>HP NonStop</i> リファレンス・ガイドを参照してください。</p>                                                                                                                                                                                                                                                                                                                                                                      |
| TCPBUFSIZE <bytes>               | <p>Extract が保持を試みる TCP ソケット・バッファのサイズ（バイト）を制御します。バッファ・サイズを増やすと、ターゲット・システムにより大きなパケットを送信できます。</p> <p>実際のバッファ・サイズは、TCP スタックの実装およびネットワークによって決定されます。デフォルトは 30,000 バイトですが、最新のネットワーク構成では、通常これより大きい値がサポートされています。有効な値は、1000 ～ 200000000（2 億）バイトです。ネットワーク管理者とともに最適な値を決定してください。287 ページの「最適なバッファ・サイズの決定」も参照してください。</p> <p>ターゲット・システムの Oracle GoldenGate インストールが 8.0.4 より前のリリースの場合は、TCPBUFSIZE の指定にかかわらず、30,000 バイトのバッファのみが使用されます。以前のリリースの Collector プロセスでは、これより大きなパケットはサポートされていません。</p> <p>初期ロード Extract では、このパラメータを使用しないでください。これは、オンライン Extract グループにのみ有効です。</p> <p>ターゲット・システムが NonStop の場合は、このパラメータを使用しないでください。</p> |
| TCPFLUSHBYTES <bytes>            | <p>ネットワークを介して送信されるデータを収集するバッファのサイズ（バイト）を制御します。この値または FLUSHSECS パラメータの値に到達したときに、データがターゲットにフラッシュされます。</p> <p>デフォルトは 30,000 バイトです。有効な値は 1000 ～ 200000000（2 億）バイトですが、少なくとも TCPBUFSIZE の値を設定する必要があります。</p> <p>初期ロード Extract では、このパラメータを使用しないでください。これは、オンライン Extract グループにのみ有効です。</p> <p>ターゲット・システムが NonStop の場合は、このパラメータを使用しないでください。</p>                                                                                                                                                                                                                                                                      |
| TIMEOUT <seconds>                | <p>Collector が Extract からの接続を待機する時間を定義します。最小値は 1 秒ですが、本番環境では、タイムアウト設定を非常に低くすることをお薦めしません。最大値は 1800 秒（30 分）です。デフォルト値は 300 秒（5 分）です。孤立した Collector プロセスは、定義済タイムアウト時間の経過後に自動的にクローズします。このパラメータは、静的な Collector には影響しません。</p>                                                                                                                                                                                                                                                                                                                                                                                   |

- 例 1** RMTHOST 20.20.20.17, MGRPORT 7809, ENCRYPT blowfish, KEYNAME newyork
- 例 2** RMTHOST newyork, MGRPORT 7809, COMPRESS, COMPRESSTHRESHOLD 750
- 例 3** RMTHOST newyork, MGRPORT 7809, TCPBUFSIZE 100000, TCPFLUSHBYTES 300000

# RMTHOSTOPTIONS

**適用対象**      パッシブ Extract

RMTHOSTOPTIONS パラメータでは、信頼性の低いソース上でパッシブ・モードで実行されている Extract グループと、より安全性の高いネットワーク・ゾーン内のターゲット・システム間の TCP/IP 属性を制御します。このパラメータでは、圧縮、データ暗号化、バッファ属性、および接続リクエストの待機時間を制御します。Collector パラメータの設定にも使用できます。

このパラメータは、リモート接続の確立に必要なホスト情報を提供しないため、RMTHOST パラメータとは異なります。Extract がパッシブ・モードで実行されている場合、ソースおよびターゲット間のすべての接続はターゲット上の別名 Extract グループによって確立されます。ゾーン化されたネットワークでの Oracle GoldenGate の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

すべてのパラメータ・オプションは、1 つの RMTHOSTOPTIONS 文内に指定する必要があります。複数の RMTHOSTOPTIONS 文が使用されている場合は、パラメータ・ファイルで最後に指定されている文が使用され、他の文は無視されます。RMTHOSTOPTIONS は、ファイル内のすべての RMTHOST 文よりも優先されます。

**デフォルト**      なし

**構文**

```
RMTHOSTOPTIONS
[, COMPRESS]
[, COMPRESSTHRESHOLD]
[, ENCRYPT {NONE | BLOWFISH}]
[, KEYNAME <keyname>]
[, PARAMS <collector parameters>]
[, TCPBUFSIZE <bytes>]
[, TCPFLUSHBYTES <bytes>]
[, TIMEOUT <seconds>]
```

| 引数                           | 説明                                                                                                                                                                                                                                                                                                                             |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMPRESS                     | 送信レコードのブロックを圧縮して、必要な帯域幅を低減します。Oracle GoldenGate は、トレイルに書き込む前にデータを解凍します。COMPRESS での通常の圧縮率は少なくとも 4:1 で、それより高くなる場合もあります。ただし、データを圧縮すると CPU リソースを消費します。                                                                                                                                                                             |
| COMPRESSTHRESHOLD            | 圧縮を行うための最小ブロック・サイズを設定します。有効な値は、0 ～ 28000 です。デフォルトは 1,000 バイトです。                                                                                                                                                                                                                                                                |
| ENCRYPT<br>{NONE   BLOWFISH} | <p>ターゲット・システムに TCP/IP を介して送信されるデータ・ストリームを暗号化します。</p> <ul style="list-style-type: none"> <li>◆ NONE では暗号化なし（デフォルト）を指定します。</li> <li>◆ BLOWFISH では、Blowfish 暗号化を指定します。KEYNAME オプションを使用する必要があります。データはターゲット・システム上で自動的に復号化されるため、復号化パラメータは不要です。</li> </ul> <p>データ暗号化の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p> |

| 引数                               | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KEYNAME <keyname>                | <p>ENCKEYS ファイルの鍵名。Oracle GoldenGate は、この鍵を使用してデータを復号化します。ターゲット・システムの ENCKEYS ファイルに一致する鍵名が存在しない場合、Oracle GoldenGate は異常終了します。ENCKEYS ファイルの作成の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>                                                                                                                                                                                                                                                                                                                                                                                                       |
| PARAMS<br><collector parameters> | <p>NonStop ターゲット・システム上の Collector パラメータを指定します。</p> <p>注意 : Manager が Collector を動的に起動する場合は、Collector ポート (-p 引数) を指定しないでください。</p> <p>NonStop プラットフォームの Collector パラメータの詳細は、Oracle GoldenGate HP NonStop リファレンス・ガイドを参照してください。</p>                                                                                                                                                                                                                                                                                                                                                                                 |
| TCPBUFSIZE <bytes>               | <p>Extract が保持を試みる TCP ソケット・バッファのサイズ (バイト) を制御します。バッファ・サイズを増やすと、ターゲット・システムにより大きなサイズのパケットを送信できます。</p> <p>実際のバッファ・サイズは、TCP スタックの実装およびネットワークによって決定されます。デフォルトは 30,000 バイトですが、最新のネットワーク構成では、通常これより大きい値がサポートされています。有効な値は、1000 ~ 200000000 (2 億) バイトです。ネットワーク管理者とともに最適な値を決定してください。287 ページの「最適なバッファ・サイズの決定」も参照してください。</p> <p>ターゲット・システムの Oracle GoldenGate インストールが 8.0.4 より前のリリースの場合は、TCPBUFSIZE の指定にかかわらず、30,000 バイトのバッファのみが使用されます。以前のリリースの Collector プロセスでは、これより大きなパケットはサポートされていません。</p> <p>初期ロード Extract では、このパラメータを使用しないでください。これは、オンライン Extract グループにのみ有効です。</p> <p>ターゲット・システムが NonStop の場合は、このパラメータを使用しないでください。</p> |
| TCPFLUSHBYTES <bytes>            | <p>ネットワークを介して送信されるデータを収集するバッファのサイズ (バイト) を制御します。この値または FLUSHSECS パラメータの値に到達したときに、データがターゲットにフラッシュされます。</p> <p>デフォルトは 30,000 バイトです。有効な値は 1000 ~ 200000000 (2 億) バイトですが、少なくとも TCPBUFSIZE の値を設定する必要があります。</p> <p>初期ロード Extract では、このパラメータを使用しないでください。これは、オンライン Extract グループにのみ有効です。</p> <p>ターゲット・システムが NonStop の場合は、このパラメータを使用しないでください。</p>                                                                                                                                                                                                                                                                          |
| TIMEOUT <seconds>                | <p>パッシブ・モードで実行している Extract が Collector からの接続をどれくらい待機するかを指定します。最小値は 1 秒ですが、本番環境では、タイムアウト設定を非常に低くすることをお薦めしません。最大値は 1800 秒 (30 分) です。デフォルト値は 300 秒 (5 分) です。孤立した Collector プロセスは、定義済タイムアウト時間の経過後に自動的にクローズします。このパラメータは、静的な Collector には影響しません。</p>                                                                                                                                                                                                                                                                                                                                                                   |

**例** RMTHOSTOPTIONS ENCRYPT blowfish, KEYNAME newyork, COMPRESS, COMPRESSTHRESHOLD 750, TCPBUFSIZE 100000, TCPFLUSHBYTES 300000

# RMTASK

**適用対象** Extract

初期ロード Extract 用の RMTASK パラメータでは、Oracle GoldenGate ダイレクト・ロードまたは SQL\*Loader へのダイレクト・バルク・ロード中に、Replicat 処理タスクを開始します。RMTASK では、Extract に TCP/IP を介して Replicat と通信させ、Collector プロセスまたはディスク・ストレージの使用をバイパスします。また RMTASK では、Extract から Manager に、Replicat を自動起動させ、実行の終了時に Replicat を停止するようにリクエストさせます。タスクはチェックポイントを作成しません。

次に依存関係にあるパラメータを示します。

- 初期ロード Extract ファイルの各 RMTASK 文の後に、RMTHOST 文を続ける必要があります。
- 初期ロード Replicat パラメータ・ファイルで、SPECIALRUN を使用する必要があります。

RMTASK は、どのような種類の暗号化もサポートしていません。暗号化を使用するには、ファイルにデータを書き込み、Replicat がこのデータを読み込んでロードする初期ロード方法を使用する必要があります。

RMTASK は、ユーザー定義型 (UDT) をサポートしていません。

RMTASK を使用するときは、Replicat を START REPLICAT コマンドで起動しないでください。Replicat は、このタスク中に自動的に起動されます。

初期データ・ロードの実行方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** なし

**構文** RMTASK REPLICAT, GROUP <group name>  
FORMAT RELEASE <major>.<minor>

| 引数                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GROUP <group name>                | ターゲット・システム上の初期ロード Replicat グループ名。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| FORMAT<br>RELEASE <major>.<minor> | <p>Extract により証跡、ファイルまたは (リモート・タスクの場合) 別のプロセスに送信されるデータのメタデータ形式を指定します。形式は Oracle GoldenGate プロセスのリリースに依存します。古い Oracle GoldenGate リリースには新しいものよりも含まれるメタデータの数が少ない、または異なります。メタデータによって、データ・レコードがサポートされるリリースのものであるかどうかプロセスに伝達されます。</p> <ul style="list-style-type: none"><li>◆ FORMAT は、必須のキーワードです。</li><li>◆ RELEASE では Oracle GoldenGate リリースが指定されます。&lt;major&gt; はメジャー・リリース番号で、&lt;minor&gt; はマイナー・リリース番号です。有効な値は 9.0 から現在の Oracle GoldenGate リリース番号の間です。(9.0 より前の Oracle GoldenGate リリースを使用している場合は、9.0 か 9.5 のいずれかを指定します。) プログラムとして、リリースは適切な証跡形式の互換性レベルにマッピングされます。デフォルト値は、この証跡に書き込まれるプロセスの現在のリリースです。</li></ul> <p>追加モードが Extract で有効でない場合、デフォルトのファイルのバージョン番号は自動的に 9 (Oracle GoldenGate リリース 9.x) になります。Extract モードは、RECOVERYOPTIONS パラメータで制御されます。</p> |

**例** RMTASK REPLICAT, GROUP initrep

# RMTTRAIL

**適用対象** Extract

RMTTRAIL パラメータでは、GGSCI の ADD RMTTRAIL コマンドで作成されているリモート・トレイルを指定します。RMTTRAIL で指定するトレイルは、関連する TABLE よりも先に配置する必要があります。異なるリモート・トレイルを定義するために、複数の RMTTRAIL 文を使用できます。RMTTRAIL より先に RMTHOST パラメータを配置する必要があります。

## ファイル・バージョンングについて

すべての Oracle GoldenGate プロセスは分離されるので、異なる Oracle GoldenGate リリースのものである可能性があることから、各証跡ファイルまたは抽出ファイルでは、バージョンがファイル・ヘッダー内に格納されています。デフォルトでは、証跡のバージョンは、ファイルを作成したプロセスの現在のバージョンになります。証跡のバージョンを設定するには、EXTTRAIL、EXTTRAIL、RMTTRAIL または RMTFILE パラメータの FORMAT オプションを使用します。

異なる Oracle GoldenGate プロセス・バージョンの上位および下位互換性が保証されるようにするには、ファイルのヘッダー・フィールドは標準のトークン形式で記述される必要があります。プロセスの新しいバージョンで作成される新しいトークンは、古いバージョンでは無視されるようにできるので、下位互換性が保証されます。同様に新しい Oracle GoldenGate バージョンでは古いトークンがサポートされます。さらに、トークンが新しいプロセス・バージョンで推奨されない場合、デフォルト値がトークンに割り当てられるので、古いバージョンは引き続き適切に機能します。ファイル・バージョンを指定するトークンは COMPATIBILITY で、Logdump ユーティリティで表示可能です。また、@GETENV 関数の GGFILEHEADER オプションで取得することもできます。

## パラメータの依存関係

RECOVERYOPTIONS パラメータと EXTTRAIL、RMTTRAIL、EXTFILE および RMTFILE の FORMAT オプション間には依存性が存在します。RECOVERYOPTIONS が APPENDMODE に設定される場合、FORMAT オプションは RELEASE 10.0 以上に設定される必要があります。RECOVERYOPTIONS が OVERWRITEMODE に設定される場合、FORMAT オプションは RELEASE 9.5 以下に設定される必要があります。

**デフォルト** なし

**構文** RMTTRAIL <name>  
[, FORMAT RELEASE <major>.<minor>]

| 引数     | 説明                                                                                               |
|--------|--------------------------------------------------------------------------------------------------|
| <name> | トレイルの完全修飾パス名。名前には 2 文字を使用します。エーシングされたトレイル・ファイルは、この名前に 6 桁の順序番号が追加されます (例: /ggs/dirdat/rt000001)。 |



| 引数                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FORMAT<br>RELEASE <major>.<minor> | <p>Extract により証跡、ファイルまたは（リモート・タスクの場合）別のプロセスに送信されるデータのメタデータ形式を指定します。形式は Oracle GoldenGate プロセスのリリースに依存します。古い Oracle GoldenGate リリースには新しいものよりも含まれるメタデータの数が少ない、または異なります。メタデータによって、データ・レコードがサポートされるリリースのものであるかどうかプロセスに伝達されます。</p> <ul style="list-style-type: none"> <li>◆ FORMAT は、必須のキーワードです。</li> <li>◆ RELEASE では Oracle GoldenGate リリースが指定されます。&lt;major&gt; はメジャー・リリース番号で、&lt;minor&gt; はマイナー・リリース番号です。有効な値は 9.0 から現在の Oracle GoldenGate リリース番号の間です。（9.0 より前の Oracle GoldenGate リリースを使用している場合は、9.0 か 9.5 のいずれかを指定します。）プログラムとして、リリースは適切な証跡形式の互換性レベルにマッピングされます。デフォルト値は、この証跡に書き込まれるプロセスの現在のリリースです。</li> </ul> <p>追加モードが Extract で有効でない場合、デフォルトのファイルのバージョン番号は自動的に 9（Oracle GoldenGate リリース 9.x）になります。Extract モードは、RECOVERYOPTIONS パラメータで制御されます。</p> <p>証跡または抽出ファイルには、それらが読み込まれるプロセスのバージョン以下のバージョンが必要です。それ以外の場合はプロセスが異常終了します。さらに、Oracle GoldenGate では、データ・ポンプの出力証跡またはファイルのバージョンが、入力証跡またはファイルのバージョンと同じであることが強制されます。再起動の際、Extract では新しいファイルが証跡にロールされるので、各ファイルが 1 つのバージョンのみ（ファイルが空の場合を除く）であることが保証されます。</p> |

- 例 1 RMTTRAIL /ggs/dirdat/ny
- 例 2 RMTTRAIL /ggs/dirdat/ny, FORMAT RELEASE 10.4

## ROLLOVER

### 適用対象 Extract

ROLLOVER パラメータでは、トレイル・ファイルをいつエージングし、新しいファイルを作成するかを指定します。ROLLOVER はグローバル・パラメータで、パラメータ・ファイルの RMTTRAIL または RMTFILE 文で定義されているすべてのトレイルに適用されます。

ROLLOVER を使用して、特定の期間を表す（例：各日）トレイル・ファイルを作成します。これにより、継続処理を促進し、出力を体系化する手段を提供します。また、1 つのファイルを非アクティブ化し、次の実行用に新しいファイルを作成することにより、バッチ処理を体系化する手段を提供します。

ファイルは、トランザクションの途中でなく、トランザクションとトランザクションの間でロールオーバーされるので、データの整合性が確保されます。ファイルがロールオーバーされるときにチェックポイントが作成されるため、処理の際に以前のファイルは不要になります。

ロールオーバーは、実行中にロールオーバーの条件が満たされた場合にのみ行われます。たとえば、ROLLOVER ON TUESDAY が指定され、データ抽出が火曜日に開始される場合、（より詳しい ROLLOVER ルールが指定されていないかぎり）ロールオーバーは次の火曜日まで行われません。最大 30 のロールオーバー・ルールを指定できます。



AT または ON のいずれかのオプションを指定する必要があります。両方のオプションと一緒に、任意の順番で使用できます。AT のみで ON を指定しない場合、指定した時刻に毎日新しいトレイル・ファイルが作成されます。

**デフォルト** デフォルトのファイル・サイズに到達したときか、ADD RMTTRAIL または ADD EXTTRAIL コマンドの MEGABYTES オプションで指定されているサイズに到達したときにロールオーバーする。

**構文** ROLLOVER {AT <hh:mi> | ON <day> | AT <hh:mi> ON <day>} [REPORT]

| 引数         | 説明                                                                                                                                                              |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AT <hh:mi> | <p>ファイルをエージングする時刻。</p> <p>有効な値：</p> <ul style="list-style-type: none"> <li>◆ hh は 24 時間表記の時間で、1 ～ 23 までの値が有効です。</li> <li>◆ mi は 00 ～ 59 までの値を受け付けます。</li> </ul> |
| ON <day>   | <p>ファイルをエージングする曜日。</p> <p>有効な値：</p> <p>SUNDAY<br/>MONDAY<br/>TUESDAY<br/>WEDNESDAY<br/>THURSDAY<br/>FRIDAY<br/>SATURDAY</p> <p>大 / 小文字は区別されません。</p>             |
| REPORT     | <p>最後のレポート生成以降に各表から抽出されたレコード数のレポートを生成します。このレポートは、REPORT パラメータによって他のレポートが生成されないかぎり、対応するトレイルにレコード出力数を示します。</p>                                                    |

**例 1** 次に、トレイルを毎日午後 3:00 にエージングする例を示します。

ROLLOVER AT 15:00

**例 2** 次に、トレイルを毎週日曜日午前 8:00 にエージングする例を示します。

ROLLOVER AT 08:00 ON SUNDAY

## SEQUENCE

**適用対象** Extract

SEQUENCE パラメータでは、Oracle GoldenGate トレイルへの伝播および別のデータベースへの送信のために、順序値をトランザクション・ログから抽出します。現在 Oracle GoldenGate は、Oracle データベースで順序をサポートしています。

**注意** シーケンス (CREATE、ALTER、DROP、RENAME) の DDL サポートは複製シーケンス値と互換性がありますが、必須ではありません。シーケンス値のみを複製するには、Oracle GoldenGateDDL サポート環境をインストールする必要はありません。単に SEQUENCE パラメータを使用できます。

Oracle GoldenGate では、ターゲット・シーケンスの値が次のように設定されます。

- 増分間隔が正の値である場合に、ソース値よりも大きい。
- 増分間隔が負の値である場合に、ソース値よりも小さい。

増分方向に基づき、Replicat では、挿入を実行する際のテストとして次の式のいずれかが適用されます。

```
source_highwater_value + (source_cache_size * source_increment_size * source_RAC_nodes) <=
target_highwater_value
```

または

```
source_highwater_value + (source_cache_size * source_increment_size * source_RAC_nodes) >=
target_highwater_value
```

式の結果が FALSE である場合、ターゲット・シーケンスはソース値よりも大きい値（シーケンスが増分される場合）またはソース値よりも小さな値（シーケンスが減分される場合）に更新されます。ターゲットは常に、式のカッコ内の式の前にある、またはそれと等しいものである必要があります。たとえば、ソースの水位標値が 40、CACHE が 20、ソースの INCREMENTBY 値が 1、および 2 つのソース RAC ノードがある場合、ターゲットの水位標値は最低 80 である必要があります。

```
40 + (20*1*2) < 80
```

ターゲットの水位標値が 80 未満である場合、Oracle GoldenGate は、ターゲットがソースの前にあり続けるように、シーケンスを更新して水位標値を増分します。現在の水位標値を取得するには、次の問合せを実行します。

```
SELECT last_number FROM all_sequences WHERE sequence_owner=upper('SEQUENCEOWNER')
AND sequence_name=upper('SEQUENCENAME');
```

## サポートされている処理モード

- Oracle GoldenGate オンラインおよびバッチ (SPECIALRUN) 変更同期方法では、順序値のレプリケーションがサポートされています。
- SOURCEISTABLE が、Extract パラメータとして、または ADD EXTRACT 内に含まれる Oracle GoldenGate 初期ロード方法では、順序値のレプリケーションはサポートされていません。
- Oracle GoldenGate は、双方向構成での順序値のレプリケーションをサポートしていません。
- Oracle GoldenGate は、高可用性構成で順序値をサポートしています。この構成には、両方のサーバー（プライマリ・サーバーとターゲット・フェイルオーバー・サーバー）にプライマリ Extract、データ・ポンプ、および Replicat が含まれます。この構成では、フェイルオーバー・サーバー上の Extract プロセスは非アクティブである必要があり、取得対象の順序は含まれません。高可用性構成での Oracle GoldenGate の構成および操作方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。
- データ・ポンプに書き込むプライマリ Extract に対して SEQUENCE を使用する場合は、データ・ポンプが PASSTHRU モードか NOPASSTHRU モードであるかにかかわらず、データ・ポンプに対して同一の SEQUENCE パラメータを使用する必要があります。ただし、DDL パラメータを使用し、同じデータ・ポンプを介して（順序またはその他のオブジェクトの）DDL 操作を伝播する場合は、データ・ポンプは PASSTHRU モードで動作する必要があります。

## SEQUENCE 使用のガイドライン

- ソース・シーケンスとターゲット・シーケンスのキャッシュ・サイズおよび増分間隔は同一である必要があります。
- キャッシュ・サイズは、0（ゼロ、NOCACHE）を含む、任意のサイズにできます。
- シーケンスは、サイクルするようにもしないようにも設定できますが、ソース・データベースとターゲット・データベースで同じように設定する必要があります。
- DDL サポートが有効な構成に SEQUENCE を追加するには、Oracle GoldenGate DDL オブジェクトを INITIALSETUP モードで再インストールする必要があります。

## エラー処理

- Extract は、順序名を解決できない場合、操作を無視します。
- Replicat による順序のエラー処理を有効化するには、REPEROR パラメータを使用します。このパラメータは、MAP パラメータのオプションとしても、単独のパラメータとしても使用できます。REPEROR は、ターゲットで順序が削除されたかどうかを検出できるので、このパラメータを使用して順序が再作成されるまで順序操作を再試行できます。
- REPEROR は、起動時に行方不明のオブジェクトを処理しません。DDLERROR と IGNOREMISSINGTABLES を使用してください。

## その他の重要情報

- データベースによる順序の維持手法にギャップは固有で、発生が予想されるため、Oracle GoldenGate がレプリケートする順序値にはギャップが存在する可能性があります。ただし、NOCHECKSEQUENCEVALUE を使用しないかぎり、ターゲット値は常にソース値より大きくなります（138 ページ）を参照してください。
- Extract が RAC システム上でシングルスレッド・モードで実行中で、ラグがあるノード上で順序が更新された場合、順序の取得に時間がかかる場合があります。これは正常な動作です。
- フェイルオーバーでは、トランザクション・ログまたは Oracle GoldenGate トレイル・ファイルのデータの損失または破損を引き起こす問題が発生すると、レプリケートされた順序の更新値が失われます。
- SEND EXTRACT および SEND REPLICAT を REPORT オプションとともに使用するときに表示される統計には、順序操作は UPDATE として表示されます。

## オブジェクトの名前と所有者

ソースおよびターゲット・オブジェクト名は、Oracle GoldenGate パラメータ・ファイルで、scott.emp のようにスキーマおよび名前両方で完全に修飾する必要があります。

## 大 / 小文字区別

Oracle GoldenGate は、Oracle では大 / 小文字を区別せず、小文字または大 / 小文字が混在する所有者、データベース、およびオブジェクト名を大文字に変換します。データベース自体が大 / 小文字の区別をサポートするように設定されているかにかかわらず、大文字に変換します。

## サポートされる文字

Oracle GoldenGate では、オブジェクト名およびキー列とキー以外の列の名前で英数字の文字がサポートされます。また、Oracle GoldenGate では、Oracle GoldenGate でキーとして使用されない列について、次の英数字以外の文字もサポートされます。

表 45 オブジェクト名およびキー以外の列の名前でサポートされる英数字以外の文字<sup>1</sup>

| 文字  | 説明            |
|-----|---------------|
| ~   | チルド           |
| <>  | 大なりおよび小なり記号   |
| /   | フォワード・スラッシュ   |
| \   | バックスラッシュ      |
| !   | 感嘆符           |
| @   | @記号           |
| #   | シャープ記号        |
| \$  | ドル記号          |
| %   | パーセント記号       |
| ^   | 脱字記号          |
| ()  | 丸カッコ（開きおよび閉じ） |
| _   | 下線            |
| -   | ダッシュ          |
| +   | プラス記号         |
| =   | 等号            |
|     | パイプ           |
| []  | 角カッコ（開きおよび閉じ） |
| { } | 中カッコ（開きおよび閉じ） |

<sup>1</sup> Oracle GoldenGate によって使用されるキーのタイプは、指定される表の定義、および KEYCOLS 句によるオーバーライドがあるかどうかによって異なります。Oracle GoldenGate では、主キー（ある場合）または一意キー / 索引（データベースにより異なる）が使用されます。これらの定義がない場合、表のすべての列が使用されますが、KEYCOLS 句は存在するすべてのキー・タイプよりも優先されます。Oracle GoldenGate によってキーとして使用される列の場合、名前の文字が WHERE 句への包含に有効である必要があります。このリストにはすべてが含まれていますが、データベースのプラットフォームでこれらの文字がサポートされる場合とされない場合があります。

## サポートされない文字

Oracle GoldenGate では、次の文字はオブジェクトまたは列の名前でサポートされません。

表 46 オブジェクトおよび列の名前でサポートされない文字<sup>1</sup>

| 文字  | 説明             |
|-----|----------------|
| &   | アンパサンド         |
| *   | アスタリスク         |
| ?   | 疑問符            |
| :   | コロン            |
| ;   | セミコロン          |
| ,   | コンマ            |
| '   | 一重引用符          |
| “ ” | 二重引用符          |
| ‘ ’ | アクセント記号（発音区別符） |
| .   | ピリオド           |
|     | 空白             |

<sup>1</sup> このリストにはすべてが含まれていますが、データベースのプラットフォームでこれらの文字がサポートされる場合とされない場合があります。

## ワイルドカードの使用

TABLE、SEQUENCE および MAP パラメータでは、1 つの文内で複数のオブジェクトを指定する場合にアスタリスクのワイルドカードを使用できます。ワイルドカードはオブジェクト名には使用できますが、スキーマまたはデータベース名には使用できません。ターゲット・オブジェクトがターゲット・データベースに存在する必要があります。

### ワイルドカードの解決方法

ワイルドカードはデフォルトで次の方法で解決されます。

**ソース・オブジェクト :** ソース・オブジェクトの名前が明示的に記述されている場合、そのオブジェクトおよびターゲット・オブジェクトの解決はプロセスの開始時に行われます。ソース・オブジェクトの名前でワイルドカードが使用されている場合、このソース・オブジェクトおよびそのターゲット・オブジェクトの解決は、ソース・オブジェクトの最初の行が処理されるときに行われます。

**ターゲット・オブジェクト :** ターゲット・オブジェクトでワイルドカードが使用されている場合、Oracle GoldenGate は、このワイルドカードを対応するソース・オブジェクトの名前で置き換えます。（「ワイルドカードの使用に関するルール」を参照してください。）

このデフォルトの動作により、Oracle GoldenGate による、処理の開始後に作成されたソース・オブジェクトの取得が可能になります。ワイルドカードのルールを変更するには、WILDCARDRESOLVE パラメータを使用します。デフォルト値は、DYNAMIC です。

TABLEEXCLUDE および MAPEXCLUDE パラメータを使用して、ワイルドカード・オブジェクトの選択の使用と、明示的オブジェクトの除外を組み合わせることができます。

### ワイルドカードの使用に関するルール

ワイルドカードを使用する際、次のルールが適用されます。

- アスタリスク (\*) は、任意の文字数（ゼロ・シーケンスを含む）と一致します。
- ソース・オブジェクトの場合は、名前の一部にワイルドカードを付けた形式を使用できます。たとえば、次のソース指定は有効です。

```
hq.t_*;
```

- ターゲット・オブジェクトの場合、名前の一部にワイルドカードを付けた形式は、ワイルドカードが使用されたターゲット名のアスタリスクがソース・オブジェクトの名前と置換されるので、使用できません。たとえば、次の文は無効です。

```
TABLE hq.t_*, TARGET rpt.t_*;
MAP hq.t_*, TARGET rpt.t_*;
```

これらは、ターゲット指定内のワイルドカードがソース・オブジェクトの名前である T\_TEST と置換されるので、次のような解決結果になります。

```
TABLE HQ.T_TEST1, TARGET RPT.T_T_TEST1;
MAP HQ.T_TEST1, TARGET RPT.T_T_TEST1;
```

次の例は、正しいワイルドカードの使用方法を示すものです。

```
TABLE hq.t_*, TARGET rpt.*;
MAP hq.t_*, TARGET rpt.*;
```

これにより、次のような正しい結果が得られます。

```
TABLE HQ.T_TEST1, TARGET RPT.T_TEST1;
MAP HQ.T_TEST1, TARGET RPT.T_TEST1;
```

以下に示すスキーマでのワイルドカードの使用例では、データベースに、USER1、USER2、A132B、AB、AX および XB のスキーマがあります。また、USER1.SALES、USER2.SALES、A132B.SALES、AB.SALES、AX.SALES、XB.SALES の表があります。TABLE 文は次のようになります。

```
TABLE A*B.SALES;
```

この構成では、次の表が取得されます。

```
A132B.SALES
```

```
AB.SALES
```

次の表は取得されません。

```
AX.SALES
```

```
XB.SALES
```

**デフォルト**    なし

**構文**           SEQUENCE <owner>.<sequence>;

| 引数                          | 説明                                                                   |
|-----------------------------|----------------------------------------------------------------------|
| SEQUENCE <owner>.<sequence> | ソース順序の所有者（スキーマ）および名前を指定します。 <b>297</b> ページの「オブジェクトの名前と所有者」を参照してください。 |
| ;                           | SEQUENCE パラメータ文を終了します。セミコロンはオプションです。                                 |

**例**             SEQUENCE hr.employees\_seq;

## SETENV

**適用対象**       Extract および Replicat

SETENV パラメータでは、任意の環境変数の値を設定します。Extract または Replicat は、起動時に、環境で設定された値ではなく、このパラメータで指定された値を使用します。

設定する変数ごとに 1 つの SETENV 文を使用します。SETENV 文で設定されたすべての変数は、オペレーティング・システム・レベルで設定されているすべての既存の変数よりも優先されます。

**デフォルト**     なし

**構文**           SETENV (  
          {<environment\_variable> |  
          GGS\_CacheRetryCount |  
          GGS\_CacheRetryDelay}  
          = "<value>"  
          )

| オプション                  | 説明                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <environment_variable> | 設定する環境変数名。                                                                                                                                                                                                                                                                                                                                                                                                           |
| "<value>"              | 指定する変数の値。この値は引用符で囲む必要があります。                                                                                                                                                                                                                                                                                                                                                                                          |
| GGS_CacheRetryCount    | <p>(SQL Server) システム・アクティビティが多いために、Extract によるソース・トランザクション・ログ・ファイルの読み取りがブロックされる場合に、この読み取りの再試行回数を制御する Oracle GoldenGate 環境変数です。デフォルトの再試行回数は 10 回です。Extract は、指定された回数の試行後、次のようなエラーとともに異常終了します。</p> <p>GGS ERROR 600 [CFileInfo::Read] Timeout expired after 10 retries with 1000 ms delay waiting to read transaction log or backup files.</p> <p>レポート・ファイルやエラー・ログにタイムアウト・メッセージが記録され続ける場合は、このパラメータで再試行回数を増やしてください。</p> |
| GGS_CacheRetryDelay    | <p>(SQL Server) 前回の試行が失敗したときに、Extract のトランザクション・ログ読み取りの再試行までの待機時間（ミリ秒）を制御する Oracle GoldenGate 環境パラメータです。デフォルトの待機時間は 1000 ミリ秒です。</p>                                                                                                                                                                                                                                                                                  |

- 例 1** 独立した SETENV 文を使用すると、環境設定を変更せずに、単一の Oracle GoldenGate インスタンスを複数の Oracle データベース・インスタンスに接続させることができます。次のパラメータ文では、ORACLE\_HOME および ORACLE\_SID の値を設定します。

```
SETENV (ORACLE_HOME = "/home/oracle/ora9/product")
SETENV (ORACLE_SID = "ora9")
```

- 例 2** 次のパラメータ文では、Extract が異常終了する前に実行するトランザクション・ログ読み取り再試行回数を最大 20 回、再試行間隔を 3000 ミリ秒として、SQL Server 環境の Oracle GoldenGate の値を設定します。

```
SETENV (GGS_CacheRetryCount = 20)
SETENV (GGS_CacheRetryDelay = 3000)
```

## SHOWSYNTAX

### 適用対象 Replicat

各 Replicat SQL 文の適用前に表示することができるインタラクティブ・セッションを開始するには、SHOWSYNTAX パラメータを使用します。失敗した SQL 文の構文を確認することで、問題の原因を診断できる場合があります。たとえば、WHERE 句で索引が付けられていない列が使用されていることを検知できます。

### SHOWSYNTAX 使用の要件

- SHOWSYNTAX を初めて使用するときは、Oracle Support アナリストにサポートをリクエストしてください。これはデバッグ・パラメータで、適切に使用しないと望ましくない結果が生じる場合があります。手動操作が要求され自動処理が中断されるため、処理速度が低下し、バックログや遅延が発生することがあります。
- SHOWSYNTAX を使用するためには、オペレーティング・システムのコマンド・シェルから Replicat を起動する必要があります。Replicat を GGSCI から起動している場合は、SHOWSYNTAX を使用しないでください。
- SHOWSYNTAX はテスト環境で使用してください。本番環境に影響がないように、Replicat グループとターゲット表の複製を作成してください。

### SHOWSYNTAX の使用方法

1. Replicat パラメータ・ファイルに、パラメータごとに 1 行を使用して次に示す順序で次のパラメータを含めます。
  - NOBINARYCHARS
  - NODYNSQL
  - SHOWSYNTAX

**注意** NOBINARYCHARS は、Oracle GoldenGate にバイナリ・データを NULL で終わる文字列として処理させる、ドキュメントに記載されていないパラメータです。このパラメータを使用する前に、Oracle サポートに連絡してください。NODYNSQL を使用すると、Replicat は動的 SQL とバインド変数を使用するかわりに、リテラル SQL 文を使用します。サポートの詳細は、<http://support.oracle.com> を参照してください。



2. Oracle GoldenGate ホーム・ディレクトリで、オペレーティング・システムのコマンド・シェルから、次の構文を使用して **Replicat** を起動します。**reportfile** オプションは使用しないでください。画面に出力する必要があります。

```
replicat paramfile dirprm/<Replicat_name>.prm
```

3. 最初の **SQL** 文がプロンプトとともに表示されます。
  - 現在の文を実行し、次の文を表示するには、**Keep Displaying**（デフォルト）を選択します。
  - 通常処理を再開して画面への **SQL** 文の出力を停止するには、**Stop Display** を選択します。
4. 構文の表示が終了したら、パラメータ・ファイルから **SHOWSYNTAX**、**NOBINARYCHARS** および **NODYNSQL** を削除します。

デフォルト なし  
構文 SHOWSYNTAX

## SOURCEDB

適用対象 Manager、Extract、DEFGEN

**SOURCEDB** は、接続情報の一部としてデータ・ソース名または識別子を必要とするデータベースまたはデータ・セット用のパラメータです。**SOURCEDB** に続く **TABLE** 文で指定された表は、指定したデータ・ソースのものであるとみなされます。

データ・ソースで要求される認証方法に応じて、**SOURCEDB** とともに **USERID** パラメータの使用が必要になることがあります。

- データベース・ログイン情報が要求される場合は、同じパラメータ文内で（必要な場合）**SOURCEDB** を **USERID** とともに使用する必要があります。
- **SQL/MX** データベースの場合は、**SOURCEDB** ではカタログを指定し、**USERID** ではスキーマを指定します。スキーマが省略されると、**SOURCEDB** はグループに関連付けられているスキーマをデフォルトで使用します。
- **c-tree** データベースの場合、**SOURCEDB** ではサーバーの別名を指定します。
- オペレーティング・システム・レベルでの認証を許可するデータベースの場合は、**USERID** なしで **SOURCEDB** を指定できます。

**Manager** に対しては、**PURGEOLDEXTRACTS** など、**Manager** とソース・データベースとのやり取りを伴う Oracle GoldenGate パラメータを使用するときのみ、**SOURCEDB** を使用します。

**DB2 LUW** の場合は、**SOURCEDB** 文は別名ではなく実際の名前でデータベースを参照する必要があります。

デフォルト なし  
構文 SOURCEDB <data source>

| 引数            | 説明        |
|---------------|-----------|
| <data source> | データ・ソース名。 |

例 1 SOURCEDB mydb

例 2 SOURCEDB mydb, USERID ggs, PASSWORD ggs123

## SOURCEDEFS

**適用対象** Extract データ・ポンプおよび Replicat

SOURCEDEFS パラメータでは、ソース表またはファイルの定義が含まれるファイル名を指定します。ソース定義は、ソースおよびターゲットの表構造が異なる異種同期環境で Oracle GoldenGate を使用する場合に必要です。お使いの Oracle GoldenGate 構成に応じて、次の1つまたは複数のプロセスで SOURCEDEFS を使用します。

- ターゲット・システム上の Replicat プロセス
- ソースまたは仲介システム上のデータ・ポンプ

ソース定義ファイルを作成するには、DEFGEN ユーティリティを使用します。データ・ポンプまたは Replicat を起動する前に、仲介システムまたはターゲット・システムにこのファイルを送信してください。

たとえば、各 SOURCEDEFS ファイルに別々のアプリケーションの定義が保持されている場合など、複数のソース定義ファイルを使用するときは、パラメータ・ファイルで複数の SOURCEDEFS 文を使用できます。

**デフォルト** なし

**構文** SOURCEDEFS <file name>

| 引数          | 説明                      |
|-------------|-------------------------|
| <file name> | ソース・データ定義を含むファイルの完全修飾名。 |

**例 1** SOURCEDEFS c:\ggs\dirdef\tcust.def

**例 2** SOURCEDEFS /ggs/dirdef/source\_defs

## SOURCEISTABLE

**適用対象** Extract

SOURCEISTABLE パラメータでは、別の表またはファイルへのロードの準備のために、ソース表から直接完全なレコードを抽出します。SOURCEISTABLE は、TABLE 文内に指定されているすべての列データを抽出します。

このパラメータは、次の初期ロード方法に適用されます。

- ファイルから Replicat へのデータのロード
- ファイルからデータベース・ユーティリティへのデータのロード

このパラメータは、次の初期ロード方法では使用しないでください。

- Extract がファイルを使用せずに Replicat プロセスにロード・データを直接送信する Oracle GoldenGate ダイレクト・ロード
- SQL\*Loader への Oracle GoldenGate ダイレクト・バルク・ロード

これらの処理の場合、SOURCEISTABLE は、パラメータ・ファイルでは使用せずに、ADD EXTRACT の引数として指定します。初期データ・ロード方法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

SOURCEISTABLE を使用する場合は、Extract パラメータ・ファイルの最初のパラメータ文にする必要があります。

SOURCEISTABLE を使用するには、Extract および Replicat パラメータ・ファイルから DDL パラメータを削除し、DDL 抽出およびレプリケーションを無効にします。詳細は、149 ページを参照してください。

**デフォルト** なし  
**構文** SOURCEISTABLE

## SPACESTONULL | NOSPACESTONULL

**適用対象** Replicat

SPACESTONULL および NOSPACESTONULL パラメータでは、空白のみを含むソース列をターゲット表で NULL に変換するかどうかを制御します。ターゲット列が NULL 値を受け付ける場合、SPACESTONULL は空白を NULL に変換します。NOSPACESTONULL は、空白をターゲット列で単一の空白文字に変換します。

これらのパラメータは表に固有です。一方のパラメータは、もう一方のパラメータが見つかるまで、それ以降のすべての MAP 文に有効です。このパラメータは、Oracle のみサポートしています。

**デフォルト** NOSPACESTONULL  
**構文** SPACESTONULL | NOSPACESTONULL

## SPECIALRUN

**適用対象** Extract および Replicat

このパラメータの実装は、プロセスによって異なります。

### Extract 用 SPECIALRUN

Extract パラメータ・ファイルの SPECIALRUN パラメータでは、次のいずれかの構成のワнтаイト処理実行を指定します。

- バッチ変更同期実行
- Oracle GoldenGate ロールバック・セッション

これらのプロセスの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

SPECIALRUN は、Extract にチェックポイントを作成しないように指示します。ワнтаイト実行には開始と終了があるため、チェックポイントは不要です。（オンライン処理では SPECIALRUN のかわりに EXTRACT パラメータを使用します。）

SPECIALRUN を使用する場合は、パラメータ・ファイルの最初のパラメータにする必要があります。SPECIALRUN を指定するときは、BEGIN および END パラメータを使用する必要があります。

このパラメータは、次のデータベースに対してはサポートされていません。

- Ingres
- SQL/MX
- SQL Server
- Sybase

デフォルト なし

構文 SPECIALRUN, <data source>

| 引数                             | 説明                                                                                                                                                                                                               |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <data source>                  | 次のいずれかを指定できます。                                                                                                                                                                                                   |
| TRANLOG [<bsds>]               | データ・ソースとしてトランザクション・ログを指定します。z/OS システム上の DB2 では、<bsds> オプションを使用してトランザクション・ログのブートストラップ・データ・セットのファイル名を指定します。指定する BSDS 名が、 <i>Extract</i> プロセスの接続先 DB インスタンスのものであることを確認してください。Oracle GoldenGate は、BSDS 指定の検証を実行しません。 |
| EXTTRAILSOURCE<br><trail name> | データ・ソースとしてトレイルを指定します。EXTTRAIL または RMTTRAIL パラメータで指定されているトレイルの完全修飾名を指定します。このオプションは、データ・ポンプ <i>Extract</i> プロセスとともに使用します。                                                                                          |
| EXTFILESOURCE <file name>      | データ・ソースとして抽出ファイルを指定します。EXTFILE または RMTFILE パラメータで指定されているファイルの完全パス名を指定します。このオプションは、データ・ポンプ <i>Extract</i> プロセスとともに使用します。                                                                                          |

例 1 SPECIALRUN

例 2 SPECIALRUN TRANLOG

例 3 SPECIALRUN EXTTRAILSOURCE c:\ggs\dir\dat\aa

例 4 SPECIALRUN EXTFILESOURCE c:\ggs\dir\dat\datafile

## Replicat 用 SPECIALRUN

Replicat パラメータ・ファイルのワнтаイム処理実行用の SPECIALRUN パラメータは、Replicat にチェックポイントを作成しないように指示します。ワнтаイム実行には開始と終了があるため、チェックポイントは不要です。

SPECIALRUN は、次のときに使用します。

- ファイルから Replicat にデータをロードする初期データ・ロード方法
- バッチ変更同期実行

Replicat が SPECIALRUN モードの場合は、GGSCI の START REPLICAT コマンドで Replicat を起動しないでください。初期ロード・タスク中に自動的に起動するか、バッチ実行でコマンド・ラインから起動する必要があります。

これらの構成の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

SPECIALRUN を指定するときは、END パラメータを使用する必要があります。Replicat パラメータ・ファイルには、REPLICAT (280 ページを参照してください) または SPECIALRUN のいずれかを指定する必要があります。REPLICAT では、オンライン処理を指定します。

デフォルト なし

構文 SPECIALRUN

## SQLDUPERR

**適用対象** Replicat

SQLDUPERR パラメータでは、重複行を検出したときにデータベースによって返される数値エラー・コードを指定します。重複レコード・エラーは、データベース内の既存レコードと一致する主キーを使用した挿入操作が試みられたときに発生します。

SQLDUPERR は、OVERRIDEDUPS パラメータを使用して特別な重複レコード処理を指定するときに使用する必要があります。複数のデータベース・タイプにレプリケートする場合は、複数の SQLDUPERR インスタンスを使用します。

**デフォルト** なし**構文** SQLDUPERR <error number>

| 引数             | 説明                            |
|----------------|-------------------------------|
| <error number> | 重複レコードが検出されたときに返される数値エラー・コード。 |

**例** 次の文は、Microsoft Access および SQL Server の重複レコード・エラー・コードを示しています。

SQLDUPERR -1605

SQLDUPERR -2601

## SQLEXEC

**適用対象** Extract および Replicat

SQLEXEC パラメータは、次のように使用します。

- パラメータ・ファイルのルート・レベルで、SQL ストアド・プロシージャまたは問合せを実行する単独文として使用します。単独文としての SQLEXEC は、Oracle GoldenGate 処理中に TABLE または MAP 文から独立して実行されます。
- データベース・コマンドを実行する単独文として使用します。

**注意** SQLEXEC は、TABLE（Extract 用）または MAP（Replicat 用）文の一部として、SQL ストアド・プロシージャまたは問合せを実行するためにも使用できます。この使用方法の詳細は、この章のアルファベット順リファレンスの TABLE および MAP の項を参照してください。

SQLEXEC により、Oracle GoldenGate はデータベースと通信し、データベースによってサポートされているファンクションを実行できます。データベース・ファンクションは、データ抽出およびレプリケーション・プロセスと統合することも、これらのプロセスから独立して使用することもできます。

## SQLEXEC でサポートされているデータベースとデータ型

以下は、SQLEXEC によってサポートされるデータベース、および入力と出力パラメータにサポートされるデータ・タイプです。

### DB2 LUW および z/OS

- CHAR
- VARCHAR
- DATE
- すべての数値データ・タイプ
- BLOB データ・タイプ

### Ingres

LOB 以外のすべてのデータ・タイプ

### MySQL

TEXT および BLOB 以外のすべてのデータ・タイプ

### Oracle

次のものを除く、すべての Oracle タイプがサポートされます。

- BFILE
- BLOB
- CFILE
- CLOB
- NCLOB
- NTY

### SQL Server

- CHAR
- VARCHAR
- DATETIME
- すべての数値データ・タイプ
- 長さが 200 バイト未満のイメージおよびテキスト・データ・タイプ
- TIMESTAMP パラメータ・タイプはネイティブにサポートされませんが、他のデータ・タイプをパラメータとして使用し、ストアード・プロシージャ内の TIMESTAMP 形式に値を変換できます。

### Sybase

TEXTIMAGE および UDT 以外のすべてのデータ・タイプ

### Teradata

Oracle GoldenGate でサポートされる、すべての Teradata データ・タイプ

## 単独 SQLEXEC パラメータ使用のガイドライン

- 単独 SQLEXEC 文は、他のパラメータも考慮して、パラメータ・ファイルにリストされている順序で実行されます。
- SQLEXEC プロシージャまた問合せすべてには、すべての例外処理を含める必要があります。

- 問合せまたはプロシージャは、SQLEXEC 文の実行時に、データベースの有効な構文を使用して正しく構築されている必要があります。そうでない場合、Replicat は、設定されているエラー処理ルールにかかわらず異常終了します。許可されている SQL 構文の詳細は、データベース・ベンダーによって提供されている SQL リファレンス・ガイドを参照してください。
- SQLEXEC 句よりも先に、Oracle GoldenGate ユーザーのデータベース・ログイン情報を指定する必要があります。Extract の場合は、データベースに適切な SOURCEDB および USERID パラメータを使用します。Replicat の場合は、適切な TARGETDB および USERID パラメータを使用します。
- Oracle GoldenGate プロセスを実行しているユーザーが、SQL を実行するユーザーです。このユーザーは、コマンドおよびストアド・プロシージャの実行、およびデータベース提供のプロシージャのコール権限を持っている必要があります。
- 単独 SQLEXEC 文は、レコードから入力パラメータを受け付けたり、出力パラメータを渡したりするためには使用できません。パラメータの受け渡しにストアド・プロシージャおよび問合せを使用するには、TABLE または MAP 文内で SQLEXEC 文を使用します。
- 単独 SQLEXEC 文によって影響を受けるすべてのオブジェクトは、Oracle GoldenGate プロセスの起動前に存在する必要があります。そのため、これらのオブジェクトに対する DDL サポートを無効にする必要があります。無効にしない場合には、SQLEXEC プロシージャまたは問合せがこれらのオブジェクト上で実行される前に、DDL 操作によってオブジェクト構造の変更やオブジェクトの削除が行われることがあります。

Oracle GoldenGate でのストアド・プロシージャおよび問合せの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**注意** Oracle GoldenGate は、z/OS 上の DB2 に対して、ODBC SQLExecDirect ファンクションを使用して SQL 文を動的に実行します。つまり、接続先のデータベース・サーバーは、SQL 文を動的に準備できる必要があります。ODBC は、実行のたびに（リクエストされる間隔で）SQL 文を準備します。通常は、このことが Oracle GoldenGate ユーザーの問題になることはありません。詳細は、DB2 のマニュアルを参照してください。

|           |                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>構文</b> | <p>Procedures:</p> <pre> SQLEXEC "exec &lt;procedure name&gt;()" [EVERY &lt;n&gt; {SECONDS   MINUTES   HOURS   DAYS}] [ONEXIT] </pre>   |
| <b>構文</b> | <p>Queries:</p> <pre> SQLEXEC "&lt;sql query&gt;" [EVERY &lt;n&gt; {SECONDS   MINUTES   HOURS   DAYS}] [ONEXIT] </pre>                  |
| <b>構文</b> | <p>Database commands:</p> <pre> SQLEXEC "&lt;database command&gt;" [EVERY &lt;n&gt; {SECONDS   MINUTES   HOURS   DAYS}] [ONEXIT] </pre> |

| コンポーネント                                            | 説明                                                                                                                                                                  |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| "exec <procedure name> ()"                         | 実行するストアド・プロシージャ名を指定します。この文は、二重引用符で囲む必要があります。exec キーワードは必須です。<br>例：<br>SQLEXEC "exec prc_job_count ()"                                                               |
| "<sql query>"                                      | 実行する問合せ名を指定します。問合せは引用符で囲む必要があります。複数の行にわたる問合せでは、各行で引用符を使用します。最良の結果を得るために、各開始引用符の後ろおよび各終了引用符の前に（または少なくとも各終了引用符の前に）空白を入力してください。<br>例：<br>SQLEXEC " select x from dual" |
| EVERY <n><br>{SECONDS   MINUTES  <br>HOURS   DAYS} | 単独ストアド・プロシージャまたは問合せを、次の例のように、定義済間隔で実行します。<br>SQLEXEC "exec prc_job_count ()" EVERY 30 SECONDS                                                                       |
| ONEXIT                                             | Extract または Replicat プロセスが正常に停止したときに SQL を実行します。                                                                                                                    |
| "<database command>"                               | データベース・コマンドを実行します。                                                                                                                                                  |

- 例 1      SQLEXEC "exec prc\_job\_count ()"
- 例 2      SQLEXEC " select x from dual"
- 例 3      SQLEXEC "exec prc\_job\_count ()" EVERY 30 SECONDS
- 例 4      SQLEXEC "exec prc\_job\_count ()" ONEXIT
- 例 5      SQLEXEC "SET TRIGGERS OFF"

## STARTUPVALIDATIONDELAY[CSECS]

**適用対象**      Manager

STARTUPVALIDATIONDELAY または STARTUPVALIDATIONDELAYCSECS パラメータでは、START EXTRACT または START REPLICAT コマンドで起動されたプロセスのステータスを Manager が検証するまでの遅延時間を設定します。指定した遅延時間の経過後にプロセスが実行されていない場合、GGSCI プロンプトにエラー・メッセージが表示されます。

これらのパラメータによって、プロセス起動に必要なメモリー不足などが原因で、エラー・メッセージまたはレポートを生成できずに失敗するプロセスが明らかになります。起動時に検証を実施することで、Oracle GoldenGate ユーザーはこのような失敗を把握できます。

**デフォルト**      0 秒（起動ステータスを検証しない）

**構文**      STARTUPVALIDATIONDELAY <seconds> | STARTUPVALIDATIONDELAYCSECS <centiseconds>

| 引数                         | 説明                                |
|----------------------------|-----------------------------------|
| <seconds>   <centiseconds> | プロセスのステータスを確認するまでの遅延時間（秒またはセンチ秒）。 |



**例** 次に、Manager が START コマンドの発行後 10 センチ秒待機してからプロセスのステータスを確認する例を示します。

```
STARTUPVALIDATIONDELAYSECS 10
```

## STATOPTIONS

**適用対象** Extract および Replicat

STATOPTIONS パラメータでは、STATSEXTRACT または STATSREPLICAT コマンドによって生成される統計表示に含める情報を指定します。このパラメータのオプションは、必要に応じてこれらのコマンドの引数としても有効化できます。

**デフォルト** 各オプションを参照してください。

**構文**

```
STATOPTIONS
[, REPORTDETAIL | NOREPORTDETAIL]
[, REPORTFETCH | NOREPORTFETCH]
[, RESETREPORTSTATS | NORESETREPORTSTATS]
```

| 引数                                    | 説明                                                                                                                                                                                                                                                                                                               |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| REPORTFETCH   NOREPORTFETCH           | Extract に有効です。REPORTFETCH は、FETCHCOLS 句（332 ページを参照してください）によってトリガーされるフェッチや、トランザクション・レコードに十分な情報が含まれていないときに実行する必要があるフェッチなどの行フェッチの統計を返します。NOREPORTFETCH は、フェッチに関する統計のレポートを無効にします。デフォルトは NOREPORTFETCH です。                                                                                                            |
| REPORTDETAIL   NOREPORTDETAIL         | Replicat に有効です。REPORTDETAIL は、衝突エラーが原因でレプリケートされなかった操作の統計を返します。出力を有効にすると、このような操作が通常の統計（実行された INSERT、UPDATE および DELETE 操作）、および詳細な表示の統計でレポートされます。たとえば、10 レコードが INSERT 操作で、キーの重複が原因でこれらすべてのレコードが無視された場合、レポートには INSERT 操作数 10、衝突が原因で破棄された操作数 10 と出力されます。NOREPORTDETAIL は、衝突に関する統計のレポートを無効にします。デフォルトは REPORTDETAIL です。 |
| RESETREPORTSTATS   NORESETREPORTSTATS | Extract および Replicat に有効です。新しいプロセス・レポートが作成されるときに、REPORT パラメータによって生成される統計をリセットするかどうかを制御します。デフォルトの NORESETREPORTSTATS では、REPORTROLLOVER パラメータに基づいてレポートがロールオーバーされるときに、レポートから次のレポートに統計を繰り越します。統計をリセットするには、RESETREPORTSTATS を使用します。                                                                                   |

# SYSLOG

**適用対象** GLOBALS、Manager

SYSLOG パラメータを使用して、Oracle GoldenGate から Windows または UNIX システム上のシステム・ログに送信されるメッセージのタイプを制御できます。以下が可能です。

- すべての Oracle GoldenGate メッセージを含める
- すべての Oracle GoldenGate メッセージを抑止する
- 情報、警告、エラー・メッセージ、またはこれらの任意の組合せが含まれるようにフィルタを設定する

SYSLOG を GLOBALS または Manager パラメータのいずれか、あるいは両方として使用できます。GLOBALS パラメータ・ファイルに含まれる場合、システム上のすべての Oracle GoldenGate プロセスのメッセージ・フィルタ処理を制御します。Manager パラメータ・ファイルに含まれる場合は、Manager プロセスのみのメッセージ・フィルタ処理を制御します。GLOBALS と Manager パラメータ・ファイルの両方で使用される場合は、Manager プロセスに関しては Manager 設定の方が、GLOBALS 設定よりも優先されます。これにより、Manager プロセスとその他すべての Oracle GoldenGate プロセス間で別の設定を使用できます。

**デフォルト** すべての Oracle GoldenGate メッセージをシステム・ログに書き込む。

**構文** SYSLOG {[ALL | NONE] | [, INFO] [, WARN] [, ERROR]}

| 引数    | 説明                                                                                                                                  |
|-------|-------------------------------------------------------------------------------------------------------------------------------------|
| ALL   | すべての INFO（情報）、WARN（警告）および ERROR（エラー）メッセージをシステム・ログに送信します。これはデフォルトで、次の設定と同じです。<br>SYSLOG INFO, WARN, ERROR<br>他のオプションと組み合わせることはできません。 |
| NONE  | Oracle GoldenGate メッセージをシステム・ログに書き込みません。他のオプションと組み合わせることはできません。                                                                     |
| INFO  | ERROR としてレポートされるメッセージをシステム・ログに送信します。WARN および ERROR と組み合わせることができます（順序は任意）。                                                           |
| WARN  | WARN としてレポートされるメッセージをシステム・ログに送信します。INFO および ERROR と組み合わせることができます（順序は任意）。                                                            |
| ERROR | ERROR としてレポートされるメッセージをシステム・ログに送信します。INFO および WARN と組み合わせることができます（順序は任意）。                                                            |

**例** 次の例はともに、警告およびエラー・メッセージをシステム・ログに送信しますが、情報メッセージは送信しません。

```
SYSLOG WARN, ERROR
```

または

```
SYSLOG ERROR, WARN
```

## DEFGEN 用 TABLE

DEFGEN パラメータ・ファイルの TABLE パラメータでは、このユーティリティを実行するソース表を指定します。各 TABLE 文は、セミコロンで終了する必要があります。

**デフォルト** なし

**構文** TABLE <[owner.]table>[, DEF <definitions template>];

| 引数                         | 説明                                                                                                                                                                                      |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <[owner.]table>            | 表の所有者（オプション）および名前。Oracle のログベースの抽出を使用する場合を除き、所有者はオプションです。このパラメータは、表についてのみワイルドカード（*）引数を受け付けます。Oracle GoldenGate は、自動的に内部記憶域を拡大し、最大で 100,000 のワイルドカード・エントリを追跡します。                         |
| DEF <definitions template> | この表の定義を基盤にする定義テンプレートを指定します。同じ定義を持つ新しい表は、DEFGEN を実行せずに、かつ Oracle GoldenGate プロセスの停止と起動を伴わずに後で追加できます。テンプレートは、TABLE または MAP 文の DEF または TARGETDEF オプションで指定されます。このオプションは、初期ロードではサポートされていません。 |

**例 1** TABLE fin.account;

**例 2** TABLE fin.acc\*;

**例 3** TABLE fin.acctl, DEF acctdefs;

## Extract 用 TABLE

**適用対象** Extract

Extract パラメータ・ファイルの TABLE パラメータでは、Oracle GoldenGate によって抽出されるオブジェクトを指定します。TABLE は、次に対して有効です。

- データ変更の取得をサポートするオンライン Extract。
- ソース表からの完全なデータ・レコードの抽出をサポートする初期ロード Extract。

**注意** TABLE で取得するソース・オブジェクトをレプリケーションのためにターゲット・オブジェクトにマップするには、Replicat パラメータ・ファイルの MAP パラメータ文でソースおよびターゲット表を指定します。

TABLE では、次のオブジェクトを指定できます。

- 索引
- トリガー
- マテリアライズド・ビュー
- 表

**注意** 取得する順序を指定するには、SEQUENCE パラメータを使用します。

## サポートの制限

表に対しては、すべての TABLE オプションを使用できます。次のことを実行できます。

- 表の行を選択およびフィルタする
- 表の列をマップする
- データを変換する
- キー列を指定する
- ストアド・プロシージャおよび問合せを実行する
- ユーザー・トークンを定義する
- 末尾を切り捨てる
- ユーザー・イグジットにパラメータを渡す

索引、トリガー、およびマテリアライズド・ビューに対しては、TABLE では取得するオブジェクトの指定のみを行います。

**注意** Oracle GoldenGate は、Oracle マテリアライズド・ビューの実際のデータ値のレプリケーションをサポートしています。Oracle GoldenGate は、Oracle と Teradata の索引およびトリガーの DDL レプリケーションをサポートしていますが、これらのオブジェクトのコンテンツはサポートしていません。DDL サポートの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** なし

**構文**

```
TABLE <table spec> [, TARGET <table spec>]
[, DEF <definitions template>]
[, TARGETDEF <definitions template>]
[, COLMAP (<column mapping expression>)]
[, {COLS | COLSEXCEPT} (<column specification>)]
[, EVENTACTIONS <action>]
[, EXITPARAM "<parameter string>"]
[, {FETCHCOLS | FETCHCOLSEXCEPT} (column specification)]
[, {FETCHMODCOLS | FETCHMODCOLSEXCEPT} (<column spec>)]
[, FETCHBEFOREFILTER]
[, FILTER (<filter specification>)]
[, KEYCOLS (<column specification>)]
[, SQLEXEC (<SQL specification>)]
[, SQLPREDICATE "WHERE <where clause>"]
[, TOKENS (<token specification>)]
[, TRIMSPACES | NOTRIMSPACES]
[, WHERE (<where clause>)]
;
```

表 47 TABLE 構文コンポーネントの概要

| コンポーネント                          | 説明                                                                                                                                                                                                                                                                |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TABLE<br><table spec>            | ソース表を指定します。この表に続いて記載されているガイドラインを参照してください。                                                                                                                                                                                                                         |
| TARGET <table spec>              | <p>ソース表をマップするターゲット表を指定します。</p> <p>Extract が変換を実行するためにターゲット定義ファイル（TARGETDEFS パラメータで指定）を参照する必要がある場合、および COLMAP オプションを使用する場合は、TABLE 文に TARGET を指定する必要があります。それ以外の場合は省略できます。TARGET を使用することによって、定義ファイルまたは列マップに指定されているレコードの構造を反映して、ソース構造ではなくターゲット構造に基づいて抽出データを特定します。</p> |
|                                  | <p>ソース・システムへのオーバーヘッドの追加を防止するために、列マッピングおよび変換をターゲット・システム上で実行できます。ただし、Windows または UNIX システムから NonStop システムへのレプリケーションでは、これらのファンクションをソースで実行する必要があります。</p>                                                                                                              |
|                                  | <p>ソースが複数でターゲットが 1 つのときは、ソースでマッピングと変換を実行するほうが適切な場合もあります。このようなケースで、特に新しいファイルの生成が必要になる頻繁なアプリケーション変更が発生するときは、ターゲットに転送する必要がある各ソース・データベースの複数のソース定義を管理するよりも、各ソースに転送する 1 つのターゲット・ファイルを管理するほうが容易です。</p> <p>ネーミング規則の詳細は、この表に続いて記載されているガイドラインを参照してください。</p>                 |
| DEF <definitions template>       | ソース定義テンプレートを指定します。                                                                                                                                                                                                                                                |
| TARGETDEF <definitions template> | ターゲット定義テンプレートを指定します。                                                                                                                                                                                                                                              |
| COLMAP                           | 異なるソース列およびターゲット列間でレコードをマップします。                                                                                                                                                                                                                                    |
| COLS   COLSEXCEPT                | 処理する列を選択または除外します。                                                                                                                                                                                                                                                 |
| EVENTACTIONS (<action>)          | 指定されているフィルタ・ルールを満たすレコードに基づいて、アクションをトリガーします。                                                                                                                                                                                                                       |
| EXITPARAM                        | リテラル文字列形式でパラメータをユーザー・イグジットに渡します。                                                                                                                                                                                                                                  |
| FETCHCOLS<br>FETCHCOLSEXCEPT     | トランザクション・レコードに値が存在しない場合に、ソース・データベースからの列値のフェッチを有効にします。                                                                                                                                                                                                             |
| FETCHBEFOREFILTER                | フィルタの実行前に、FETCHCOLS または FETCHCOLSEXCEPT アクションを実行します。                                                                                                                                                                                                              |

表 47 TABLE 構文コンポーネントの概要（続き）

| コンポーネント                                                        | 説明                                                                           |
|----------------------------------------------------------------|------------------------------------------------------------------------------|
| <code>FETCHMODCOLS</code>  <br><code>FETCHMODCOLSEXCEPT</code> | トランザクション・ログに列が存在する場合でも、データベースから列値をフェッチさせます。                                  |
| <code>FILTER</code>                                            | 数値に基づいてレコードを選択します。 <code>FILTER</code> は、 <code>WHERE</code> よりも柔軟に使用できます。   |
| <code>KEYCOLS</code>                                           | 行を一意に特定する列を指定します。                                                            |
| <code>SQLEXEC</code>                                           | ストアド・プロシージャおよび問合せを実行します。                                                     |
| <code>SQLPREDICATE</code>                                      | 初期ロードのために <code>WHERE</code> 句を使用して行を選択します。                                  |
| <code>TOKENS</code>                                            | ユーザー・トークンを定義します。                                                             |
| <code>TRIMSPACES</code>  <br><code>NOTRIMSPACES</code>         | <code>CHAR</code> 列を <code>VARCHAR</code> 列にマッピングする際に、末尾の空白を切り捨てるかどうかを制御します。 |
| <code>WHERE</code>                                             | 条件演算子に基づいてレコードを選択します。                                                        |

## オブジェクト名およびオーナー

MySQL では、データベース名がディレクトリ名、テーブル名がファイル名に、それぞれマッピングされます。ソースおよびターゲット・オブジェクト名は、`fin.emp` のように、Oracle GoldenGate パラメータ・ファイルで完全に修飾される必要があります。Oracle GoldenGate では、次のように大 / 小文字がサポートされます。

## 大 / 小文字の区別

次は、Oracle GoldenGate に関連するオブジェクト名の大 / 小文字の区別に関する一般的なガイドラインです。データベースまたは基盤となるオペレーティング・システムで大 / 小文字の区別がサポートされるかにより、これらのガイドラインがデータベースに適用される場合と、されない場合があります。MySQL では、大 / 小文字が区別される表の名前は MySQL が実行されるオペレーティング・システムによって異なります。MySQL については、Windows プラットフォームでは大 / 小文字が混じった表の名前は識別されず、また、たいていの UNIX プラットフォームでは大 / 小文字が区別されます。大 / 小文字の区別（またはその区別なし）は、ソース・データベースに適用されてもターゲット・データベースには適用されない（あるいはその逆）場合があります。

- システムまたはデータベースで大 / 小文字が区別される場合、Oracle GoldenGate では、データベースの名前、オーナーとスキーマ名、オブジェクト名、列の名前およびユーザー名について、大 / 小文字の区別がサポートされます。
- システムまたはデータベースで大 / 小文字が区別されない場合（または大 / 小文字が区別されないように設定されている場合）、Oracle GoldenGate ではすべての名前が大文字に変換されます。例外は Oracle 11g で、パスワードが必要とされる Oracle GoldenGate 入力で、パスワードの大 / 小文字が区別されます。

### Oracle GoldenGate 構成で大 / 小文字の区別を保持する方法

Oracle GoldenGate パラメータ・ファイルで、大 / 小文字が区別される名前をデータベースで表示されるとおりに指定します。他のデータベース（大 / 小文字が区別されるオブジェクトのソースまたはター

ゲット) で大 / 小文字が区別されない場合、TABLE および MAP パラメータで、大 / 小文字が区別される名前を二重引用符で囲みます。

大 / 小文字が区別されないソースから大 / 小文字が区別されるターゲットに複製する場合、Extract ではこれらは大文字で証跡に書き込まれるので、Replicat MAP 文にソース名を大文字で入力します。

例：

```
MAP SALES.CUSTOMER, TARGET "Sales.Account";
```

**注意** 引用符で囲まれた列名は、リテラルとして扱われます。Oracle GoldenGate による列名の大 / 小文字区別への対応の詳細は、321 ページの「COLMAP の使用」を参照してください。

## サポートされる文字

Oracle GoldenGate では、オブジェクト名およびキー列とキー以外の列の名前で英数字の文字がサポートされます。また、Oracle GoldenGate では、Oracle GoldenGate でキーとして使用されない列について、次の英数字以外の文字もサポートされます。

表 48 オブジェクト名およびキー以外の列の名前でサポートされる英数字以外の文字<sup>1</sup>

| 文字 | 説明            |
|----|---------------|
| ~  | チルド           |
| <> | 大なりおよび小なり記号   |
| /  | フォワード・スラッシュ   |
| \  | バックスラッシュ      |
| !  | 感嘆符           |
| @  | @記号           |
| #  | シャープ記号        |
| \$ | ドル記号          |
| %  | パーセント記号       |
| ^  | 脱字記号          |
| () | 丸カッコ（開きおよび閉じ） |
| _  | 下線            |
| -  | ダッシュ          |
| +  | プラス記号         |
| =  | 等号            |

表 48 オブジェクト名およびキー以外の列の名前でサポートされる英数字以外の文字<sup>1</sup>

| 文字  | 説明            |
|-----|---------------|
|     | パイプ           |
| []  | 角カッコ（開きおよび閉じ） |
| { } | 中カッコ（開きおよび閉じ） |

<sup>1</sup> Oracle GoldenGate によって使用されるキーのタイプは、指定される表の定義、および KEYCOLS 句によるオーバーライドがあるかどうかによって異なります。Oracle GoldenGate では、主キー（ある場合）または一意キー / 索引（データベースにより異なる）が使用されます。これらの定義がない場合、表のすべての列が使用されますが、KEYCOLS 句は存在するすべてのキー・タイプよりも優先されます。Oracle GoldenGate によってキーとして使用される列の場合、名前の文字が WHERE 句への包含に有効である必要があります。このリストにはすべてが含まれていますが、データベースのプラットフォームでこれらの文字がサポートされる場合とされない場合があります。

## サポートされない文字

Oracle GoldenGate では、次の文字はオブジェクトまたは列の名前でサポートされません。

表 49 オブジェクトおよび列の名前でサポートされない文字<sup>1</sup>

| 文字  | 説明             |
|-----|----------------|
| &   | アンパサンド         |
| *   | アスタリスク         |
| ?   | 疑問符            |
| :   | コロン            |
| ;   | セミコロン          |
| ,   | コンマ            |
| '   | 一重引用符          |
| “ ” | 二重引用符          |
| ‘ ’ | アクセント記号（発音区別符） |
| .   | ピリオド           |
|     | 空白             |

<sup>1</sup> このリストにはすべてが含まれていますが、データベースのプラットフォームでこれらの文字がサポートされる場合とされない場合があります。

## ワイルドカードの使用

TABLE、SEQUENCE および MAP パラメータでは、1 つの文内で複数のオブジェクトを指定する場合にアスタリスクのワイルドカードを使用できます。ワイルドカードはオブジェクト名には使用できますが、スキーマまたはデータベース名には使用できません。ターゲット・オブジェクトがターゲット・データ



ベースに存在する必要があります。

## ワイルドカードの解決方法

ワイルドカードはデフォルトで次の方法で解決されます。

**ソース・オブジェクト** : ソース・オブジェクトの名前が明示的に記述されている場合、そのオブジェクトおよびターゲット・オブジェクトの解決はプロセスの開始時に行われます。ソース・オブジェクトの名前でワイルドカードが使用されている場合、このソース・オブジェクトおよびそのターゲット・オブジェクトの解決は、ソース・オブジェクトの最初の行が処理されるときに行われます。

**ターゲット・オブジェクト** : ターゲット・オブジェクトでワイルドカードが使用されている場合、Oracle GoldenGate は、このワイルドカードを対応するソース・オブジェクトの名前で置き換えます。(「ワイルドカードの使用に関するルール」を参照してください。)

このデフォルトの動作により、Oracle GoldenGate による、処理の開始後に作成されたソース・オブジェクトの取得が可能になります。ワイルドカードのルールを変更するには、WILDCARDRESOLVE パラメータを使用します。デフォルト値は、DYNAMIC です。

TABLEEXCLUDE および MAPEXCLUDE パラメータを使用して、ワイルドカード・オブジェクトの選択の使用と、明示的オブジェクトの除外を組み合わせることができます。

## ワイルドカードの使用に関するルール

ワイルドカードを使用する際、次のルールが適用されます。

- アスタリスク (\*) は、任意の文字数（ゼロ・シーケンスを含む）と一致します。
- ソース・オブジェクトの場合は、名前の一部にワイルドカードを付けた形式を使用できます。たとえば、次のソース指定は有効です。

```
hq.t_*;
```

- ターゲット・オブジェクトの場合、名前の一部にワイルドカードを付けた形式は、ワイルドカードが使用されたターゲット名のアスタリスクがソース・オブジェクトの名前と置換されるので、使用できません。たとえば、次の文は無効です。

```
TABLE hq.t_*, TARGET rpt.t_*;
MAP hq.t_*, TARGET rpt.t_*;
```

これらは、ターゲット指定内のワイルドカードがソース・オブジェクトの名前である T\_TEST と置換されるので、次のような解決結果になります。

```
TABLE HQ.T_TEST1, TARGET RPT.T_T_TEST1;
MAP HQ.T_TEST1, TARGET RPT.T_T_TEST1;
```

次の例は、正しいワイルドカードの使用方法を示すものです。

```
TABLE hq.t_*, TARGET rpt.*;
MAP hq.t_*, TARGET rpt.*;
```

これにより、次のような正しい結果が得られます。

```
TABLE HQ.T_TEST1, TARGET RPT.T_TEST1;
MAP HQ.T_TEST1, TARGET RPT.T_TEST1;
```

以下に示すスキーマでのワイルドカードの使用例では、データベースに、USER1、USER2、A132B、AB、AX および XB のスキーマがあります。また、USER1.SALES、USER2.SALES、A132B.SALES、AB.SALES、AX.SALES、

XB.SALES の表があります。TABLE 文は次のようになります。

```
TABLE A*B.SALES;
```

この構成では、次の表が取得されます。

```
A132B.SALES
```

```
AB.SALES
```

次の表は取得されません。

```
AX.SALES
```

```
XB.SALES
```

## TABLE 文での Unicode およびネイティブ・エンコーディングの使用

Oracle GoldenGate では、Unicode、または Windows、文字列の列内の文字を UNIX および Linux オペレーティング・システムのネイティブ・キャラクタ・エンコーディングで表示するためのエスケープ・シーケンスの使用がサポートされます。エスケープ・シーケンスは、TABLE または MAP 文内の次の要素で使用されます。

- WHERE 句
- Unicode キャラクタを Unicode 列に割り当てる、またはネイティブ・エンコーディングの文字を列に割り当てる COLMAP 句
- COLMAP 句内の Oracle GoldenGate 列変換関数

Oracle GoldenGate では、次のタイプのエスケープ・シーケンスがサポートされます。

- \uFFFF Unicode エスケープ・シーケンス
- \377 8 進数エスケープ・シーケンス
- \xFF 16 進数エスケープ・シーケンス

次の制限が適用されます。

- このサポートは、7ビットASCIIに相当するU+0000～U+007Fの間のUTF-16コード・ポイントに制限されます。
- ソース列およびターゲット列の両方が Unicode である必要があります。
- ソースおよびターゲットのデータ・タイプが同一である必要があります(NCHAR から NCHAR への変換など)。

### エスケープ・シーケンスの使用手順

各エスケープ・シーケンスの先頭にリバース・ソリダスを付け（コード・ポイント U+005C）、次にキャラクタ・コード・ポイントを指定します。（リバース・ソリダスは、一般にはバックスラッシュ記号として知られています。）パラメータ文内の入力文字列または列変換関数内で、実際の文字のかわりにエスケープ・シーケンスを使用します。

#### \uFFFF Unicode エスケープ・シーケンスの使用手順

- 小文字の u を先頭に付け、4 桁の 16 進数のみを指定します。
- サポートされる範囲は次のとおりです。
  - 0 ～ 9 (U+0030 ～ U+0039)
  - A ～ F (U+0041 ～ U+0046)

- a ～ f (U+0061 ～ U+0066)
- NCHAR および NVARCHAR 列に使用可能なエスケープ・シーケンスは、これのみです。
- サロゲート・ペアはサポートされていません。

**例**      \u20ac は、ユーロ通貨記号用の Unicode エスケープ・シーケンスです。

**注意**      クロスプラットフォームのサポートを確実なものにするには、Unicode エスケープ・シーケンスを使用します。8 進数および 16 進数のエスケープ・シーケンスは、異種のオペレーティング・システム間では標準化されていません。

### \\377 8 進数エスケープ・シーケンスの使用手順

- 3 つの 8 進数が含まれている必要があります。
- サポートされる範囲は次のとおりです。
  - 最初の桁の範囲が 0 ～ 3 (U+0030 ～ U+0033) です。
  - 2 番目と 3 番目の桁の範囲は 0 ～ 7 (U+0030 ～ U+0037) です。

**例**      \200 は、Microsoft Windows 環境でのユーロ通貨記号用の 8 進数エスケープ・シーケンスです。

### \\xFF 16 進数エスケープ・シーケンスの使用手順

- 小文字の x を先頭に付け、2 桁の 16 進数のみを指定します。
- サポートされる範囲は次のとおりです。
  - 0 ～ 9 (U+0030 ～ U+0039)
  - A ～ F (U+0041 ～ U+0046)
  - a ～ f (U+0061 ～ U+0066)

**例**      \x80 は、Microsoft Windows 環境でのユーロ通貨記号用の 16 進数エスケープ・シーケンスです。

## COLMAP の使用

COLMAP では、ソース列を異なる名前のターゲット列に明示的にマップするか、ソース名とターゲット名が同一の場合にデフォルトの列マッピングを指定します。COLMAP は、列データの選択、変換、および移動方法を提供します。データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。

**注意**      後続の TABLE 文のすべての表に適用するグローバル・ルールを作成するには、COLMATCH パラメータを使用します。

### データ定義の生成

構造が同一でないソース表とターゲット表に対して COLMAP を使用する場合は、ソース表のデータ定義を生成し、ターゲットに転送し、SOURCEDEFS パラメータを使用して定義ファイルを特定する必要があります。

ソースおよびターゲットが同一の構造とみなされるためには、同一の列名（該当する場合は大 / 小文字区別を含む）とデータ型が含まれ、かつ各表の列の順序が同一である必要があります。また、両方の表は、文字用の列の列長セマンテック（バイトまたは文字）が同一である必要があります。

両方の表が同一の構造を持ち、変換などのその他の機能のために COLMAP を使用する場合は、ソース定義は必要ありません。かわりに ASSUMETARGETDEFS パラメータを使用できます。

詳細は、次を参照してください。

- SOURCEDEFS は 304 ページ

- ASSUMETARGETDEFS は 115 ページ
- 『Oracle GoldenGate Windows and UNIX 管理者ガイド』のデータ定義ファイルの作成に関する項

### デフォルトの列マッピングの使用

ソース列と対応するターゲット列の名前が同一の場合は、明示的なマッピング文ではなく、デフォルト・マッピングを使用できます。デフォルト・マッピングでは、Oracle GoldenGate は名前が同じ列を自動的にマップします。適切な場合は、データ変換も自動的に行います。

デフォルト・マッピングを使用するには、USEDEFAULTS オプションを使用します。デフォルト・マッピングは、明示的なマッピング文でマップされていない列にのみ有効です。

Sybase および SQL Server を除くすべてのデータベースでは、列名は名前比較のために大文字に変更されます。Sybase および SQL Server では、USEDEFAULTS は次の方法で大 / 小文字区別をサポートします。

- 名前がターゲット列と正確に一致するソース列が見つかった場合、2 つの列はマッピングされます。
- 大 / 小文字が一致する列が見つからない場合、大 / 小文字の一致にかかわらずターゲット列と名前が一致する最初の候補のソース列を使用してマップが作成されます。

たとえば、次は大 / 小文字を区別する列を含むソース表とターゲット表です。

#### ソース表 USER1.SM01

id  
owner  
created  
changed  
creator  
modifiedBy  
comment  
COMMENT

#### ターゲット表 USER3.SM01

ID  
owner  
id  
Creator  
comment  
ModifiedBy  
creationDate  
alterationDate  
Comment  
COMMENT

次に示すこれらの表に対する列マップには、明示的およびデフォルトの列マッピングが含まれています。

```
TABLE USER1.SM01, TARGET USER3.SM01,
COLMAP (USEDEFAULTS,
 ID = id,
 creationDate = created,
 alterationDate = changed,
);
```

このマップの結果を次に示します。デフォルト・マッピングでは、適切な場合には大 / 小文字の区別が識別され、それ以外の場合には名前のみが照合されます。2 つのターゲット列は、明示的にマップされておらず、デフォルト・マッピングも確立できなかったため、マップされません。

#### マッピングのタイプ

明示的なマッピング:

#### マッピングの結果

ID = id,  
creationDate = created,  
alterationDate = changed

デフォルト・マッピング:                   owner = owner,  
                                                 comment = comment,  
                                                 COMMENT = COMMENT,  
                                                 Creator = creator,  
                                                 ModifiedBy = modifiedby

マッピングされないターゲット列:           id, Comment

列マッピングの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文**

```
TABLE <table spec>, TARGET <table spec>,
COLMAP (
[USEDEFAULTS,]
<target column> = <source expression>
[, ...]
);
```

| コンポーネント                               | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <table spec>                          | ソースまたはターゲット表。                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <target column> = <source expression> | <p>ソースおよびターゲット列間のマップを明示的に定義します。列でサポートされている文字の詳細は、317 ページの「サポートされる文字」を参照してください。Unicode またはネイティブ・エンコーディングの文字列または列を使用するときに適用されるルールの詳細は、320 ページを参照してください。</p> <p>&lt;target column&gt; はターゲット列名です。</p> <p>&lt;source expression&gt; には、次のいずれかを指定できます。</p> <ul style="list-style-type: none"> <li>◆ ソース列名（例：ORD_DATE）</li> <li>◆ 数値定数（例：123）</li> <li>◆ 引用符で囲まれた文字列定数（例："ABCD"）</li> <li>◆ Oracle GoldenGate 列変換ファンクションを使用する式（例：@STREXT(COL1,1,3)）。列変換ファンクションの詳細は、第 4 章を参照してください。</li> </ul> |
| USEDEFAULTS                           | <p>明示的な列マップで指定されていない場合に、同一の名前を持つソース列とターゲット列を自動的にマップします。1 つの列セットには、明示的マップまたは USEDEFAULTS を使用し、両方を使用しないでください。詳細は、322 ページの「デフォルトの列マッピングの使用」を参照してください。USEDEFAULTS は、明示的な列マップの前に指定します。</p>                                                                                                                                                                                                                                                                                                   |

**例 1**           TABLE ggs.tran, TARGET ggs.tran2, COLMAP (loc2 = loc, type2 = type);

**例 2**           TABLE ggs.tran, TARGET ggs.tran2, COLMAP COLMAP ( EUROVAL = "\u20ac0" );

**例 3**           TABLE ggs.tran, TARGET ggs.tran2, COLMAP (SECTION = @STRCAT("\u00a7", SECTION ));

## COLS および COLSEXCEPT の使用

COLS および COLSEXCEPT では、列選択を制御します。

- COLS では、同期するデータが含まれる列を指定します。他のすべての列は Oracle GoldenGate に無視されます。

- COLSEXCEPTでは、同期から除外する列を指定します。他のすべての列はOracle GoldenGateに処理されます。多数の列が含まれる表では、COLS を使用して各列を指定するよりも、COLSEXCEPT を使用するほうが効率的な場合があります。キー列は除外しないでください。

次に、COLS を使用するための条件を示します。

- 表に1つ以上のキー列が含まれているか、TABLE のKEYCOLS オプションで代替キーが定義されている必要があります。
- キー列またはKEYCOLS で指定されている列がCOLS で指定されている列リストに含まれている必要があります。含まれていない場合、これらの列は取得されず、処理中にエラーが生成されます。  
(注意: COLS を指定しない場合、キー列は自動的に取得されます。)

主キーまたは一意キーがない、またはその両方が存在しない場合に TABLE 文の KEYCOLS 句がない場合、Oracle GoldenGate は表のすべての列を使用するため、COLS の指定は不要になります。

データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。

**構文** TABLE <table spec>, {COLS | COLSEXCEPT} (<column> [, ...]) ;

| コンポーネント  | 説明                                                                                                                                                                                                        |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <column> | <p>列名。複数の列を指定するには、次の例のようにコンマ区切りリストを作成します。</p> <p>次の例では、列 1 および 3 のみを処理します。</p> <pre>TABLE hq.acct, COLS (col1, col3);</pre> <p>次の例では、列 4 を除きすべての列を処理します。</p> <pre>TABLE hq.acct, COLSEXCEPT (col4);</pre> |

**注意** データベースが（変更されないかぎり列値が記録されない）圧縮更新を使用している場合、COLS で抽出するように指定した列を使用できないことがあります。このような列値を使用するには、TABLE 文の FETCHCOLS オプションを使用するか、列のサブリメンタル・ロギングを有効にします。

## DEF の使用

DEF では、ソース定義テンプレートを指定します。定義テンプレートは、特定のソース表に対して DEFGEN が実行されるときに、このオブジェクトの定義に基づいて作成されます。テンプレートが作成されると、この表と同一の定義を持つ新しいソース表は、DEFGEN を実行せずに、かつ Extract の停止と起動を伴わずに追加できます。DEF で指定されたテンプレートの定義は、定義の参照に使用されます。DEFGEN の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文** TABLE <table spec>, DEF <definitions template>;

| 引数                     | 説明                                                                                                 |
|------------------------|----------------------------------------------------------------------------------------------------|
| <definitions template> | DEFGEN パラメータ・ファイルの TABLE の DEF オプションで指定されている定義テンプレート名。テンプレートに含まれる定義は、この TABLE 文の表の定義と同一である必要があります。 |

**例** TABLE acct.cust\*, DEF custdef;

## EVENTACTIONS の使用

EVENTACTIONS では、特定のフィルタリング・ルールに適合する、イベント・レコードと呼ばれるトランザクション・ログ内のレコードに基づいて、Extract に定義済のアクションを実行させます。このシステムを使用して、データベース・イベントに基づいて Oracle GoldenGate 処理をカスタマイズできます。

**警告** EVENTACTIONS は、ソース・データベースが Teradata で、Extract が最大パフォーマンス・モードで構成されている場合にはサポートされません。

このシステムの使用方法の例には、プロセスの開始または停止、変換の実行、統計のレポートなどがあります。イベント・マーカー・システムは、次のような目的で使用されます。

- SQLEXEC またはユーザー・イグジット関数が実行される同期化ポイントを確立する。
- データ検証スクリプトを実行するシェル・コマンドを実行する。
- 特定のアカウント番号が検出される際の証跡を有効化する。
- ラグ履歴を取得する。
- バッチ・プロセスまたは営業後レポート・プロシージャを開始する時点を確立する。

イベント・マーカー機能は、データ変更のレプリケーションにはサポートされていますが、初期ロードにはサポートされていません。

### イベント・マーカー・システムの使用方法：

システムには、次の入力コンポーネントが必要です。

1. 操作をトリガーするイベント・レコードを指定します。これを行うには、次のいずれか 1 つのパラメータ文内に FILTER か WHERE 句、または SQLEXEC 問合せかプロシージャを含めます。
  - Extract パラメータ・ファイル内の TABLE 文
  - Replicat パラメータ・ファイル内の MAP 文
  - ソース表をターゲット表にマッピングすることなく EVENTACTIONS 操作の実行を可能にする、Replicat パラメータ・ファイル内の特別な TABLE 文
2. イベント・レコードを指定した同じ TABLE または MAP 文で、EVENTACTIONS パラメータを適切なオプションとともに含めて、プロセスに適用される操作を指定します。

**注意** すべてではありませんが、多くの EVENTACTIONS オプションは（Extract 用）TABLE および（Replicat 用）MAP 両方に適用されるため、ここでは両方のプロセスのすべてのオプションを説明します。一方のみに適用されるオプションには、その旨記載しています。

### 複数の操作の組合せ方法：

- すべてではありませんが、多くの EVENTACTIONS オプションは組み合わせて使用できます。たいいていの場合、目的を達成するのに 2 つ以上の操作を組み合わせる必要があります。
- EVENTACTIONS 文全体を最初に解析し、優先順位に応じて実行するオプションを指定します。次のリストでは、Process the record の前にリストされている操作は、レコードが証跡に書き込まれる、またはターゲットに適用される（プロセスに基づく）前に発生します。Process the record の後にリストされる操作は、レコードが処理された後に実行されます。
  - TRACE
  - LOG
  - CHECKPOINT BEFORE
  - IGNORE
  - DISCARD

- SHELL
- ROLLOVER
- (Process the record)
- REPORT
- ABORT
- CHECKPOINT AFTER
- FORCESTOP
- STOP

#### イベント・レコード自体の処理の制御方法：

イベント・レコード自体が通常の方法で処理されないようにするには、IGNORE または DISCARD オプションを使用します。IGNORE および DISCARD は、レコード自体の前に評価されるので、レコードが処理されるのを防ぐことができます。これらのオプションがない場合、Extract によりレコードが証跡に書き込まれ、Replicatにより、このレコード内に含まれている操作がターゲット・データベースに適用されます。

トランザクションに、イベント操作をトリガーする 2 つ以上のレコードが含まれている可能性がある点を考慮する必要があります。そのような場合、特定の EVENTACTIONS の指定が複数実行されることがあります。たとえば、2 つの連続的な ROLLOVER 操作をトリガーする 2 つの有効なレコードを実行すると、Extract で証跡が 2 回ロールオーバーされ、2 つのうち 1 つが実質、空のままになります。

#### 構文

```
EVENTACTIONS (
[STOP | ABORT | FORCESTOP]
[IGNORE [TRANSACTION [INCLUDEEVENT]]]
[DISCARD]
[LOG [INFO | WARNING]]
[REPORT]
[ROLLOVER]
[SHELL <command>]
[TRACE <trace file> [TRANSACTION] [PURGE | APPEND]]
[CHECKPOINT [BEFORE | AFTER | BOTH]]
[, ...]
)
```

| 操作   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STOP | <p>指定されたイベント・レコードが発生した場合に、プロセスを正常に停止します。プロセスは、停止前にオープン・トランザクションが完了するのを待ちます。トランザクションが Replicat によってグループ化またはバッチ化されたトランザクションである場合、プロセスが正常に停止する前に現在のグループのトランザクションが適用されます。プロセスは、レコードがトランザクションの終わりであることも意味するかぎり、イベント・レコード後の次のレコードで再開されます。</p> <p>プロセスでは、トランザクションがまだオープンであるためにすぐに停止できない場合、メッセージをログに記録します。ただし、長い間実行されているオープン・トランザクション内でイベント・レコードが発生した場合は、トランザクションの未コミット状態を警告する警告メッセージは表示されません。したがって、STOP イベントにかかわらず、プロセスは長期間実行されたままになることがあります。</p> <p>STOP は、ABORT および FORCESTOP 以外の他の EVENTACTIONS オプションと組み合わせることができます。</p> |



| 操作                                     | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ABORT                                  | <p>指定されたイベント・レコードが発生した場合、オープン・トランザクションがある、ないにかかわらず、プロセスがただちに終了されます。イベント・レコードは処理されません。致命的なエラーは、ログに書き込まれ、DISCARD も指定されている場合は、イベント・レコードは破棄ファイルに書き込まれます。プロセスについては、起動時にリカバリが実行されます。</p> <p>ABORT は、CHECKPOINT BEFORE、DISCARD、SHELL および REPORT とのみ、組み合わせることができます。</p>                                                                                                                                                                                                                                                                                   |
| FORCESTOP                              | <p>指定されたイベント・レコードが発生すると、イベント・レコードが、トランザクションで最後の操作である場合、またはトランザクション内の唯一のレコードである場合にのみ、プロセスが正常に停止されます。レコードは通常どおり書き込まれます。</p> <p>イベント・レコードが、長い間実行されているオープン・トランザクション内で発生した場合、ABORT で指定されるように、プロセスはログに警告メッセージを書き込み、すぐに終了します。この場合、起動時のリカバリが必要になる場合があります。FORCESTOP 操作が、長い間事項されているトランザクションの途中でトリガーされた場合、プロセスは警告メッセージなしに終了します。</p> <p>FORCESTOP は、ABORT、STOP、CHECKPOINT AFTER および CHECKPOINT BOTH 以外の他の EVENTACTIONS オプションと組み合わせることができます。ROLLOVER とともに使用される際、プロセスが正常に停止する場合にロールオーバーのみが発生します。</p>                                                            |
| IGNORE<br>[TRANSACTION [INCLUDEEVENT]] | <p>デフォルトでは、プロセスで指定されたイベント・レコードが無視されます。いずれの警告またはメッセージもログに書き込まれませんが、Oracle GoldenGate 統計が更新され、レコードが無視されたことが示されます。</p> <ul style="list-style-type: none"> <li>◆ TRANSACTION を使用して、イベントをトリガーしたレコードが含まれるトランザクション全体を無視できます。TRANSACTION が使用される場合、イベント・レコードはトランザクション内で最初の者である必要があります。トランザクションを無視する際、デフォルトでイベント・レコードも無視されます。TRANSACTION は、TRANS に短縮できます。</li> <li>◆ INCLUDEEVENT を TRANSACTION とともに使用して、イベント・レコードを証跡またはターゲットに伝播することができますが、残りの関連するトランザクションは無視されます。</li> </ul> <p>IGNORE は、ABORT および DISCARD 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p> |

| 操作                   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DISCARD              | <p>プロセスで次の処理が行われます。</p> <ul style="list-style-type: none"> <li>◆ 指定されたイベント・レコードを破棄ファイルに書き込む。</li> <li>◆ Oracle GoldenGate 統計を更新して、レコードが破棄されたことを示す。</li> </ul> <p>プロセスは、証跡の次のレコードから処理を再開します。このオプションを使用した場合、DISCARDFILE パラメータを使用して破棄ファイルの名前を指定できます。デフォルトでは、破棄ファイルは作成されません。</p> <p>DISCARD は、IGNORE 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p>                                                                                     |
| LOG [INFO   WARNING] | <p>指定されたイベント・レコードが発生した場合に、プロセスによってイベントがログに記録されます。メッセージはレポート・ファイル、Oracle GoldenGate エラー・ログおよびシステム・イベント・ログに書き込まれます。</p> <p>次のオプションを使用して、メッセージの重大度を指定します。</p> <ul style="list-style-type: none"> <li>◆ INFO では、重大度の低い情報メッセージを指定します。これはデフォルトです。</li> <li>◆ WARNING では、重大度の高い警告メッセージを指定します。</li> </ul> <p>LOG は、ABORT 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。ABORT を使用する場合、プロセスが終了される前に ABORT によってログに致命的エラーが記録されるので、LOG は必要ありません。</p> |
| REPORT               | <p>指定されたイベント・レコードが発生した場合に、プロセスでレポート・ファイルが生成されます。これは、GGSCI で SEND コマンドを REPORT オプションとともに使用した場合と同じです。</p> <p>REPORT メッセージは、イベント・レコードが処理された後に発生するので (DISCARD、IGNORE または ABORT が使用されていないかぎり)、レポート・データにイベント・レコードが含まれます。</p> <p>REPORT は、他のすべての EVENTACTIONS オプションと組み合わせることができます。</p>                                                                                                                                                      |

| 操作                                                      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ROLLOVER                                                | <p>Extract にのみ有効です。指定されたイベント・レコードが発生した場合に、Extract により証跡ファイルが新しいファイルにロール・オーバーされます。ROLLOVER 操作は、Extract によるイベント・レコードの証跡ファイルへの書込み前に発生するので、DISCARD、IGNORE または ABORT も使用されていないかぎり、レコードは新しいファイル内で最初のものになります。</p> <p>ROLLOVE は、ABORT 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>注意：</p> <p>次の理由により、ROLLOVER を ABORT と組み合わせることはできません。</p> <ul style="list-style-type: none"> <li>◆ ROLLOVER では、プロセスによるチェックポイントへの書込みはできない。</li> <li>◆ ROLLOVER は、ABORT の前に発生する。</li> </ul> <p>ROLLOVER チェックポイントがない場合、ABORT により、Extract は再起動時に、以前の証跡ファイル内にある以前のチェックポイントを確認しようとします。この結果、ロールオーバーが取り消されます。</p> |
| SHELL <command>                                         | <p>指定されたイベント・レコードが発生した場合に、プロセスによって指定されたシェル・コマンドが実行されます。</p> <ul style="list-style-type: none"> <li>◆ &lt;command&gt; には、実行されるシステムまたはシェル・コマンドを指定します。</li> <li>◆ シェル・コマンドが成功すると、プロセスでは情報メッセージがレポート・ファイルおよびイベント・ログに書き込まれます。成功したかどうかは、UNIX シェル言語に準拠するコマンドのイグジット状態に基づきます。この言語では、ゼロは成功を示します。</li> <li>◆ システム呼出しに失敗すると、プロセスは致命的エラーを出して異常終了します。UNIX シェル言語では、ゼロ以外は失敗です。</li> </ul> <p>SHELL は、他のすべての EVENTACTIONS オプションと組み合わせることができます。</p>                                                                                                                                                         |
| TRACE <trace file><br>[TRANSACTION]<br>[PURGE   APPEND] | <p>指定されたイベント・レコードが発生した場合に、プロセスにより証跡情報が証跡ファイルに書き込まれます。</p> <p>デフォルトでは、証跡はプロセスが終了されるまで有効になります。証跡レベルを設定するには、Oracle GoldenGate TRACE または TRACE2 パラメータを使用します。</p> <ul style="list-style-type: none"> <li>◆ &lt;trace file&gt; には、証跡ファイルの名前を指定し、TRACE キーワード後すぐに表示される必要があります。一意の証跡ファイルを指定するか、またはスタンドアロンの TRACE または TRACE2 パラメータで指定されるデフォルトの証跡ファイルを使用できます。</li> </ul> <p>EVENTACTIONS TRACE が使用されている異なる TABLE または MAP 文で、同じ証跡ファイルを使用できます。複数の TABLE または MAP 文で同じ証跡ファイル名が指定されているが、TRACE オプションが統一して使用されていない場合、この証跡ファイルが含まれている最後に解決された TABLE または MAP 内のオプションが優先されます。</p>                              |

| 操作                                    | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHECKPOINT<br>[BEFORE   AFTER   BOTH] | <ul style="list-style-type: none"> <li>◆ プロセスが終了されるのではなく、現在のトランザクションの終わりまで証跡を有効化するには、TRANSACTION を使用します。Replicat の場合、トランザクション境界は、通常の Replicat によるグループ化またはバッチ化されたターゲット・トランザクションではなく、ソース・トランザクションに基づきます。TRANSACTION は、TRANS に短縮できます。</li> <li>◆ PURGE を使用して、追加の証跡レコードの書込み前に証跡ファイルの切捨てを行うか、または、APPEND を使用して、既存のレコードの終わりに新しい証跡レコードを書き込みます。APPEND がデフォルトです。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                                       | <p>TRACE は、ABORT 以外の他のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>指定された証跡ファイルへの証跡を無効化するには、GGSCI の SEND &lt;process&gt; コマンドを TRACE OFF &lt;filename&gt; オプションとともに実行します。</p> <p>指定されたイベント・レコードが発生した場合に、プロセスでチェックポイントが書き込まれます。チェックポイント操作により、TABLE または MAP 文で定義されている処理に関するコンテキストが提供されます。このコンテキストとは開始ポイントおよび終了ポイントであるので、SQLEXEC およびユーザー・イグジットで実行される関数のマッピング用の同期化ポイントが与えられます。</p> <ul style="list-style-type: none"> <li>◆ BEFORE<br/>Extract の場合、BEFORE では、Extract による証跡へのイベント・レコードの書込み前にチェックポイントの書込みがプロセスによって行われます。</li> <li>Replicat の場合、BEFORE では、Replicat によってレコード内に含まれている SQL 操作のターゲットへの適用前に、チェックポイントの書込みがプロセスによって行われます。</li> <li>BEFORE では、イベント・レコードがトランザクション内の最初のレコードである必要があります。最初のレコードでない場合、プロセスは異常終了します。BEFORE を使用して、これより前のすべてのトランザクションが、イベント・レコードがコミットされた時点で開始されるように設定できます。</li> <li>CHECKPOINT BEFORE は、すべての EVENTACTIONS オプションと組み合わせることができます。</li> </ul> |

| 操作 | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>◆ AFTER</p> <p>Extract の場合、AFTER では、Extract による証跡へのイベント・レコードの書き込み後にチェックポイントの書き込みがプロセスによって行われます。</p> <p>Replicat の場合、AFTER では、Replicat によってレコード内に含まれている SQL 操作のターゲットへの適用後に、チェックポイントの書き込みがプロセスによって行われます。</p> <p>AFTER では、チェックポイント・リクエストが勧告としてフラグされます。つまり、プロセスで、次の実質的な機会があった場合にのみチェックポイントが実行されます。たとえば、イベント・レコードが複数のレコードのトランザクションの 1 つである場合、チェックポイントは、Oracle GoldenGate データ整合性モデルに従って、次のトランザクション境界で発生します。</p> <p>CHECKPOINT AFTER は、ABORT 以外のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>◆ BOTH</p> <p>BOTH は、BEFORE と AFTER を組み合わせたものです。Extract または Replicat プロセスでは、イベント・レコードが処理される前後にチェックポイントの書き込みが行われます。</p> <p>CHECKPOINT BOTH は、ABORT 以外のすべての EVENTACTIONS オプションと組み合わせることができます。</p> <p>CHECKPOINT は、CP に短縮できます。</p> |

**例 1** 次に、特定の受注番号の挿入操作を含むトランザクションのトレースを有効にする例を示します。

```
TABLE source.order, FILTER (@GETENV ("GGHEADER", "OPTYPE") = "INSERT" AND
order_no = 1), EVENTACTIONS (TRACE order_1.trc TRANSACTION);
```

**例 2** 次に、定義された処理期間の終了時に、トレイルを順序の次のファイルにロールオーバーする例を示します。一連のトレイル・ファイルは、処理期間に基づいてユニットとして結合できます。その動作を説明します。

1. Extract が FILTER 句を満たすレコードを検出すると、ROLLOVER イベント・アクションがソース・データベースに記録されます。
2. トランザクション・ログでこのレコード (イベント・レコード) を検出すると、Extract は現在のトレイル・ファイルを閉じ、新しいトレイル・ファイルを開きます。
3. ROLLOVER イベント・アクションと IGNORE アクションを組み合わせることで、イベント・レコード自体のトレイル・ファイルへの書き込みを防ぎます。

```
TABLE source.event_table, FILTER (@GETENV ("GGHEADER", "OPTYPE") = "INSERT" AND
order_no = 10,000), EVENTACTIONS (ROLLOVER, IGNORE);
```

イベント・マーカー・システムの他の使用例および詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

## EXITPARAM の使用

EXITPARAM では、TABLE 文からのレコードを検出するたびにユーザー・イグジット・ルーチンにパラメータを渡します。データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。ユーザー・イグジットの詳細は、第 5 章を参照してください。

**構文**           TABLE <table spec>, EXITPARAM "<parameter string>";

| コンポーネント              | 説明                                                            |
|----------------------|---------------------------------------------------------------|
| "<parameter string>" | リテラル文字列のパラメータ。パラメータは二重引用符で囲みます。パラメータ文字列では、最大で 100 文字まで指定できます。 |

## FETCHCOLS および FETCHCOLSEXCEPT の使用

FETCHCOLS および FETCHCOLSEXCEPT では、トランザクション・ログ・レコードに値が存在しない場合に、データベースから列値をフェッチします。このオプションは、データベースが（変更されないかぎり列値は記録されない）圧縮更新を使用しているものの、FILTER 操作に必要な他の列値を使用できるようにする必要があります場合に使用します。

- FETCHCOLS では、指定した列をフェッチします。
- FETCHCOLSEXCEPT では、指定した列を除くすべての列をフェッチします。多数の列が含まれる表では、FETCHCOLS を使用して各列を指定するよりも、FETCHCOLSEXCEPT を使用するほうが効率的な場合があります。

FETCHCOLS および FETCHCOLSEXCEPT は、Oracle GoldenGate にサポートされているすべてのデータベースに有効です。

- Oracle データベースの場合、Oracle GoldenGate は、Oracle の Flashback Query メカニズムを使用して、UNDO 表領域から値をフェッチします。フラッシュバック問合せは、特定の時刻または SCN 時点の列の読取り一貫性イメージを提供します。Oracle GoldenGate の Flashback Query 使用法の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。
- FETCHCOLS または FETCHCOLSEXCEPT を使用するかわりに、必要な列のサブリメンタル・ロギングを有効にするほうが効率的なこともあります。

Sybase の場合は、Oracle GoldenGate によって Sybase 暗号化データがサポートされていないため、暗号化列データはこれらのパラメータにサポートされていません。

フェッチの制御、およびフェッチ対象に指定した列が見つからない場合のレスポンスを指定するには、FETCHOPTIONS パラメータを使用します。STATS EXTRACT コマンドによって生成される統計表示にフェッチ結果を含めるには、STATOPTIONS パラメータを使用します。

FETCHCOLS または FETCHCOLSEXCEPT で指定した値がトランザクション・ログに存在する場合、データベースのフェッチは実行されません。これにより、データベースのオーバーヘッドが軽減されます。

データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。

**構文**           TABLE <table spec>, {FETCHCOLS | FETCHCOLSEXCEPT} (<column> [, ...]) ;

| コンポーネント  | 説明                                                                                                                                          |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <column> | 次のいずれかを指定できます。 <ul style="list-style-type: none"> <li>◆ 列名、または (COL1, COL2) のような列名のコンマ区切りリスト。</li> <li>◆ (*) のようなアスタリスク・ワイルドカード。</li> </ul> |

## FETCHMODCOLS および FETCHMODCOLSEXCEPT の使用

FETCHMODCOLS および FETCHMODCOLSEXCEPT では、列がトランザクション・ログに存在する場合でも、列値をデータベースからフェッチさせます。データベースのタイプに応じて、ログ・レコードに表のすべての列が含まれている場合と、特定のトランザクション操作で変更された列のみが含まれている場合があります。

- FETCHMODCOLS では、指定した列をフェッチします。
- FETCHMODCOLSEXCEPT では、指定した列を除き、トランザクション・ログに存在するすべての列をフェッチします。多数の列が含まれる表では、FETCHMODCOLS を使用して各列を指定するよりも、FETCHMODCOLSEXCEPT を使用するほうが効率的な場合があります。

データ・ポンプ Extract グループにパススルー・モードで処理される表 (パラメータ・ファイルの PASSTHRU パラメータを使用) には、このオプションを使用しないでください。この処理モードでは、データベース・ログインはサポートされていません。

このオプションは、Oracle GoldenGate によってサポートされているすべてのデータベースに有効です。

Sybase の場合は、Oracle GoldenGate によって Sybase 暗号化データがサポートされていないため、暗号化列データはこれらのパラメータにサポートされていません。

**デフォルト**   なし

**構文**           TABLE <table spec>, {FETCHMODCOLS | FETCHMODCOLSEXCEPT} (<column spec>);

| 引数              | 説明                                                                                                                                          |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| (<column spec>) | 次のいずれかを指定できます。 <ul style="list-style-type: none"> <li>◆ 列名、または (COL1, COL2) のような列名のコンマ区切りリスト。</li> <li>◆ (*) のようなアスタリスク・ワイルドカード。</li> </ul> |

## FETCHBEFOREFILTER の使用

FETCHBEFOREFILTER では、FILTER が実行される前に、FETCHCOLS または FETCHCOLSEXCEPT で指定された列をフェッチします。     事前にフェッチすることにより、フィルタに必要な値を使用できます。FETCHBEFOREFILTER を指定しない場合、FETCHCOLS または FETCHCOLSEXCEPT で指定したフェッチは、フィルタが実行されるまで実行されません。

データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。

**構文**           TABLE <table spec>, FETCHCOLS (<column> [, ...]),  
                  FETCHBEFOREFILTER,  
                  FILTER <filter clause>  
                  ;

## FILTER の使用

FILTER では、数値に基づいてレコードを選択または除外します。フィルタ式では、条件演算子、Oracle GoldenGate 列変換ファンクション、またはその両方を使用できます。

**注意** 文字列に基づいてフィルタするには、Oracle GoldenGate 文字列ファンクションの 1 つを使用するか（第 4 章を参照してください）、WHERE オプションを使用します。

FILTER コンポーネントはすべてコンマで区切ります。FILTER 句には、次を含めることができます。

- 数字
- 数字を含む列
- 数字を返すファンクション
- 算術演算子：
  - + (加算)
  - (減算)
  - \* (乗算)
  - / (除算)
  - \ (余り)
- 比較演算子：
  - > (より大きい)
  - >= (以上)
  - < (より少ない)
  - <= (以下)
  - = (等しい)
  - <> (等しくない)

比較から導出した結果はゼロ (FALSE を示す) またはゼロ以外 (TRUE を示す) になります。

- カッコ (式の結果をグループ化)
- 結合演算子: AND、OR

データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。

### 構文

```
TABLE <table spec>
, FILTER (
[, ON INSERT | ON UPDATE | ON DELETE]
[, IGNORE INSERT | IGNORE UPDATE | IGNORE DELETE]
, <filter clause>
);
```

| コンポーネント         | 説明                                                                                                                                                                                                                                                                                                                                   |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <filter clause> | <p>次のように、式に基づいてレコードを選択します。</p> <pre>FILTER ((PRODUCT_PRICE*PRODUCT_AMOUNT)&gt;10000))</pre> <p>FILTER 句内のテキスト文字列は二重引用符で囲む必要があります<br/>(例: <code>FILTER (@STRFIND(NAME, "JOE") &gt; 0)</code>)。</p> <p>フィルタ句内では、次の例のように、Oracle GoldenGate の列変換ファンクションを使用できます。</p> <pre>FILTER (@COMPUTE (PRODUCT_PRICE*PRODUCT_AMOUNT)&gt;10000)</pre> |



| コンポーネント                                             | 説明                                                                                                                                                                                        |
|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ON INSERT  <br>ON UPDATE  <br>ON DELETE             | レコードのフィルタリングを、指定した操作に限定します。次のように、操作はコマンドで区切ります。<br><br>FILTER (ON UPDATE, ON DELETE, @COMPUTE<br>(PRODUCT_PRICE*PRODUCT_AMOUNT)>10000)<br><br>この例では、更新および削除に対してフィルタを実行しますが、挿入に対しては実行しません。 |
| IGNORE INSERT  <br>IGNORE UPDATE  <br>IGNORE DELETE | 指定した操作にフィルタを適用しません。次のように、操作はコマンドで区切ります。<br><br>FILTER (IGNORE INSERT, @COMPUTE<br>(PRODUCT_PRICE*PRODUCT_AMOUNT)>10000)<br><br>この例では、更新および削除に対してフィルタを実行しますが、挿入は無視します。                     |

## KEYCOLS の使用

KEYCOLS では、ターゲット表の 1 つ以上の列を一意列として定義します。主に KEYCOLS は、ターゲット表で主キーまたは一意索引が使用できないときに、代替主キーを定義するために使用します。

ソースおよびターゲットのキー列または一意索引列は、データベースで定義されている場合も、KEYCOLS によって代替キーが指定されている場合でも、一致する必要があります。ソース表には、少なくともターゲット表と同じ数のキー列または索引列が含まれている必要があります。そうでなければ、ソースのキー列または索引列を更新する際に、Replicat は余剰なターゲット列のビフォア・イメージを取得できません。

キーを定義する際は、次のガイドラインに従ってください。

- ソース表とターゲット表両方にキーまたは一意索引が含まれていない場合は、TABLE および MAP 文両方で KEYCOLS を使用し、一致する列セットを指定します。
- いずれか一方の表にキーまたは一意索引が含まれていない場合は、その表に対して KEYCOLS を使用し、もう一方の表の実際のキーまたは索引列に一致する列を指定します。一致する列セットを定義できない場合は、TABLE および MAP 文両方で KEYCOLS を使用し、一意の値が含まれる一致する列セットを指定します。KEYCOLS の指定は、既存のキーまたは索引よりも優先されます。
- ターゲット表にソース表よりも大きなキー（または多くの一意索引列）が含まれている場合は、TABLE 文で KEYCOLS を使用し、実際のソースのキーまたは索引列に加え、余剰なターゲット列と一致するソース列を指定する必要があります。表に主キーまたは一意索引が含まれている場合、KEYCOLS の指定はこれらよりも優先されるため、余剰な列のみを指定しないでください。このように KEYCOLS を使用すると、キーまたは索引列は更新のときにビフォア・イメージを利用できます。

KEYCOLS を使用するときは、指定した列をトランザクション・ログ に記録し、Replicat がトレイルで使えるようにしてください。この設定は、データベース・インタフェースを使用するか、ADD TRANDDATA コマンドの COLS オプションを使用して行えます（Oracle ログベース抽出のみ）。

ターゲット表では、KEYCOLS で定義したキー列に一意索引を作成します。索引によって、Oracle GoldenGate が処理する必要があるターゲット行をより高速に特定できます。

データ・ポンプ Extract グループにパススルー・モードで処理される表には、KEYCOLS を使用しないでください。

**構文**           TABLE <table spec>, KEYCOLS (<column> [, ... ]);

| コンポーネント    | 説明                                                                                                                                      |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| (<column>) | 代替主キーとして使用する列を定義します。複数の列を指定するには、次のようにコンマ区切りリストを作成します。<br><br>KEYCOLS (id, name)<br><br>主キーまたは一意索引が存在する場合は、これらの列を KEYCOLS 指定に含める必要があります。 |

## SQLEXEC の使用

SQLEXEC では、Oracle GoldenGate 処理中に TABLE 文内から SQL ストアド・プロシージャまたは問合せを実行します。SQLEXEC により、Oracle GoldenGate はデータベースと直接通信し、データベースによってサポートされているファンクションを実行できます。データベース・ファンクションは、列変換のための値取得などの同期プロセスの一部として使用することも、データの抽出またはレプリケーションと関係なく使用することもできます。

TABLE 文内で使用する場合、実行されるプロシージャまたは問合せは、ソースまたはターゲット行から入力パラメータを受け付け、出力パラメータを渡すことができます。

### SQLEXEC の依存関係および制約事項

- SQL は、Oracle GoldenGate プロセスを実行しているユーザーによって実行されます。このユーザーは、ストアド・プロシージャの実行、およびデータベース提供のプロシージャの呼び出し権限を持っている必要があります。
- 問合せまたはプロシージャは、SQLEXEC 文の実行時に、データベースの有効な構文を使用して正しく構築されている必要があります。そうでない場合、設定されているエラー処理ルールにかかわらず、Oracle GoldenGate は異常終了します。許可されている SQL 構文の詳細は、データベース・ベンダーによって提供されている SQL リファレンス・ガイドを参照してください。
- SQLEXEC は、主キー列の値を変更するために使用しないでください。主キーの値は、Extract から Replicat に渡されます。主キーの値がない場合、Replicat 操作は完了できません。主キーの値を SQLEXEC で変更する必要がある場合は、元のキーの値を別の列にマッピングした後、KEYCOLS オプションでこの列を代替キーとして定義することにより、エラーを回避できます。335 ページの「KEYCOLS の使用」を参照してください。
- z/OS 上の DB2 に対して、Oracle GoldenGate は、ODBC SQLExecDirect ファンクションを使用して SQL 文を動的に実行します。つまり、接続先のデータベース・サーバーは、SQL 文を動的に準備できる必要があります。ODBC は、実行のたびに（リクエストされる間隔で）SQL 文を準備します。通常は、このことが Oracle GoldenGate ユーザーの問題になることはありません。詳細は、z/OS 上の DB2 のマニュアルを参照してください。
- データ・ポンプ Extract グループにパススルー・モードで処理される表には、SQLEXEC を使用しないでください。
- Oracle GoldenGate DDL サポートを使用する場合は、SQL の実行前に、ストアド・プロシージャまたは問合せに影響を受けるすべてのオブジェクトが、正しい構造で存在している必要があります。したがって、これらのオブジェクトの構造に影響する DDL（CREATE や ALTER など）は、SQLEXEC の実行前に実行する必要があります。

## **SQLEXEC でサポートされているデータベースとデータ型**

以下は、SQLEXEC によってサポートされるデータベース、および入力と出力パラメータにサポートされるデータ・タイプです。

### **DB2 LUW および z/OS**

- CHAR
- VARCHAR
- DATE
- すべての数値データ・タイプ
- BLOB データ・タイプ

### **Ingres**

LOB 以外のすべてのデータ・タイプ

### **MySQL**

TEXT および BLOB 以外のすべてのデータ・タイプ

### **Oracle**

次のものを除く、すべての Oracle タイプがサポートされます。

- BFILE
- BLOB
- CFILE
- CLOB
- NCLOB
- NTY

### **SQL Server**

- CHAR
- VARCHAR
- DATETIME
- すべての数値データ・タイプ
- 長さが 200 バイト未満のイメージおよびテキスト・データ・タイプ
- TIMESTAMP パラメータ・タイプはネイティブにサポートされませんが、他のデータ・タイプをパラメータとして使用し、ストアド・プロシージャ内の TIMESTAMP 形式に値を変換できます。

### **Sybase**

TEXTIMAGE および UDT 以外のすべてのデータ・タイプ

### **Teradata**

Oracle GoldenGate でサポートされる、すべての Teradata データ・タイプ

Oracle GoldenGate でのストアド・プロシージャおよび問合せの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

## SQLEXEC とストアド・プロシージャの使用

TABLE 文内からストアド・プロシージャを実行するには、SPNAME 句を使用します。

### 構文

```
SQLEXEC (
 SPNAME <sp name>
 [, ID <logical name>]
 {, PARAMS <param spec> | NOPARAMS}
 [, <option>] [, ...]
)
```

| コンポーネント                           | 説明                                                                                                                                                                                    |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <sp name>                         | ストアド・プロシージャ名を指定します。                                                                                                                                                                   |
| ID <logical name>                 | プロシージャの論理名を定義します。このオプションは、TABLE 文内でプロシージャを複数回実行するときに使用します。1 つの TABLE 文で、最大 20 のストアド・プロシージャで実行できます。プロシージャの実行が 1 回のみの場合は必要ありません。                                                        |
| PARAMS <param spec>  <br>NOPARAMS | プロシージャがパラメータを受け付けるかどうかを定義します。<br>PARAMS <param spec> または NOPARAMS のいずれかを使用する必要があります。<br><param spec> では、入力パラメータおよび入力ソースを定義します。                                                        |
| <option>                          | ストアド・プロシージャの影響を制御するために、単独または他のオプションと組み合わせて使用できる、次のいずれかのオプションを指定します。<br><br>AFTERFILTER   BEFOREFILTER<br>ALLPARAMS<br>DBOP<br>ERROR<br>EXEC<br>MAXVARCHARLEN<br>PARAMBUFSIZE<br>TRACE |

SQLEXEC コンポーネントの説明は、アルファベット順に 342 ページから記載されています。

## SQLEXEC と問合せの使用

TABLE 文内から問合せを実行するには、ID および QUERY 句を使用します。

### 構文

```
SQLEXEC (
 ID <logical name>
 , QUERY "<sql query>"
 {, PARAMS <param spec>| NOPARAMS}
 [, <option>] [, ...]
)
```

| コンポーネント                          | 説明                                                                                                                                                                                                                                                                                                              |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID <logical name>                | 問合せの論理名を定義します。問合せの結果から値を抽出するには、論理名が必要です。ID <logical name> は、問合せから返された列値を参照します。                                                                                                                                                                                                                                  |
| QUERY "<sql query>"              | データベースに対して実行する SQL 問合せの構文を指定します。問合せは、問合せを実行するデータベースの有効な標準問合せ言語を使用している必要があります。<br><br>問合せは、SELECT 文の結果を返すか、INSERT、UPDATE、または DELETE 文を実行できます。SELECT 文の出力を生成する問合せの場合は、SELECT によって最初に返される行のみが処理されます。SELECT 文には "INTO ..." 句を指定しないでください。問合せはすべてを 1 行に収め、引用符で囲む必要があります。最良の結果を得るために、各開始引用符の後ろおよび各終了引用符の前に空白を入力してください。 |
| PARAMS<br><param spec>  NOPARAMS | 問合せがパラメータを受け付けるかどうかを定義します。これらのいずれかのオプションを使用する必要があります。<param spec> では、入力パラメータおよび入力ソースを定義します。                                                                                                                                                                                                                     |
| <option>                         | 問合せの影響を制御するために、単独または他のオプションと組み合わせて使用できる、次のいずれかのオプションを指定します。<br><br>AFTERFILTER   BEFOREFILTER<br>ALLPARAMS<br>DBOP<br>ERROR<br>EXEC<br>MAXVARCHARLEN<br>PARAMBUFSIZE<br>TRACE                                                                                                                                   |

SQLEXEC コンポーネントの説明は、アルファベット順に 342 ページから記載されています。

## 入力パラメータ用のプレースホルダの使用

ほとんどの問合せは、入力パラメータ用のプレースホルダを必要とします。問合せ内でのパラメータの指定方法は、データベースのタイプによって異なります。

- Oracle の場合は、次の例のように、入力パラメータはコロン (:) を使用して指定し、その後にパラメータ名を続けます。  

```
"SELECT NAME FROM ACCOUNT WHERE SSN = :SSN AND ACCOUNT = :ACCT"
```
- 他のデータベースの場合は、次の例のように、入力パラメータは疑問符を使用して指定します。  

```
"SELECT NAME FROM ACCOUNT WHERE SSN = ? AND ACCOUNT = ?"
```

どのデータベースでも、パラメータ名を引用符で囲む必要はありません。

## パラメータの値渡し

Oracle GoldenGate は、入力値および出力値をプロシージャまたは問合せとやり取りするためのオプションを提供しています。

- ストアド・プロシージャまたは問合せ内で入力パラメータにデータ値を渡すには、SQLEXEC の PARAMS オプションを使用します (347 ページを参照してください)。
- ストアド・プロシージャまたは問合せから入力として値を FILTER または COLMAP 句に渡すには、次の構文を使用します。  

```
{<procedure name> | <logical name>}.<parameter>
```

### 条件:

- <procedure name> は、ストアド・プロシージャの実際の名前で、SQLEXEC 文の SPNAME に指定している値と一致する必要があります。この引数は、Oracle GoldenGate 実行中にプロシージャを 1 回実行する場合にのみ指定します。
- <logical name> は、SQLEXEC 文の ID オプションで指定した論理名です。この引数は、TABLE 文内でプロシージャを複数回実行する場合に、問合せまたはストアド・プロシージャのインスタンスから値を渡すときに使用します。
- <parameter> は、参照表の列などのパラメータ名か、返された値を抽出する場合の RETURN\_VALUE のいずれかです。

上記の構文のかわりとして、@GETVAL ファンクションを使用できます。詳細は、419 ページを参照してください。

入力パラメータのネーミングには、次のような異なる構成があります。

- Oracle は、次の例のように、入力パラメータに論理名を付けることを許可します。  

```
SQLEXEC (ID appphone, QUERY " select per_type from ps_personal_data "
" where emplid = :vemplid "
" and per_status = 'N' and per_type = 'A' ",
PARAMS (vemplid = emplid)),
TOKENS (applid = @GETVAL(appphone.per_type));
```

- 他のデータベースでは、次の例のように、入力パラメータに P1、P2 のような名前を付け、入力パラメータごとに番号を増分させる必要があります。

```
SQLEXEC (ID appphone, QUERY " select per_type from ps_personal_data "
 " where emplid = ? "
 " and per_status = 'N' and per_type = 'A' ",
 PARAMS (p1 = emplid)),
TOKENS (applid = @GETVAL(appphone.per_type));
```

次に、Oracle のソース表、ターゲット表、および参照表と、これらの表のパラメータをストアド・プロシージャの単一のインスタンス、およびストアド・プロシージャの複数のインスタンスに渡す方法の例を示します。

#### ソース表 "cust"

| 列名                      | 説明      |
|-------------------------|---------|
| custid                  | Number  |
| current_residence_state | Char(2) |
| birth_state             | Char(2) |

#### ターゲット表 "cust\_extended"

| 列名                           | 説明          |
|------------------------------|-------------|
| custid                       | Number      |
| current_residence_state_long | Varchar(30) |
| birth_state_long             | Varchar(30) |

#### 参照表 "state\_lookup"

| 列名           | 説明          |
|--------------|-------------|
| abbreviation | Char(2)     |
| long_name    | Varchar(30) |

- 例 1** 次に、参照表から値を取得するために 1 回実行されるストアド・プロシージャの使用例を示します。この値は、COLMAP 文のターゲット列にマッピングされます。

```
TABLE sales.cust, TARGET sales.cust_extended, &
SQLEXEC (SPNAME lookup, &
PARAMS (long_name = birth_state)), &
COLMAP (custid = custid, &
birth_state_long = lookup.long_name);
```

**例 2** 次に、参照表から値を取得するストアド・プロシージャを複数回実行する例を示します。値は、ターゲット列にマッピングされます。

```
TABLE sales.cust, TARGET sales.cust_extended, &
SQLEXEC (SPNAME lookup, ID lookup1, &
PARAMS (long_name = current_residence_state)), &
SQLEXEC (SPNAME lookup, ID lookup2, &
PARAMS (long_name = birth_state)), &
COLMAP (custid = custid, &
current_residence_state_long = lookup1.long_name, &
birth_state_long = lookup2.long_name);
```

### AFTERFILTER および BEFOREFILTER の使用

AFTERFILTER および BEFOREFILTER では、TABLE 文の FILTER 句との関連でストアド・プロシージャまたは問合せをいつ実行するかを指定します。

**構文** AFTERFILTER | BEFOREFILTER

| ルール          | 説明                                                                            |
|--------------|-------------------------------------------------------------------------------|
| AFTERFILTER  | FILTER 文の後に SQL 文を実行します。これにより、フィルタが成功しない場合に SQL 実行のオーバーヘッドを回避できます。これはデフォルトです。 |
| BEFOREFILTER | FILTER 文の前に SQL 文を実行します。したがって、プロシージャまたは問合せの結果をフィルタで使用できます。                    |

**例** SQLEXEC (SPNAME check, NOPARAMS, BEFOREFILTER)

### ALLPARAMS の使用

ALLPARAMS は、ストアド・プロシージャまたは問合せの実行のために、指定されたすべてのパラメータが存在している必要があるかどうかを決定するグローバル・ルールとして使用します。ALLPARAMS で設定するグローバル・ルールよりも、PARAMS 句内で設定する個別のパラメータのルールのほうが優先されます。

**構文** ALLPARAMS {OPTIONAL | REQUIRED}

| ルール      | 説明                                                   |
|----------|------------------------------------------------------|
| OPTIONAL | すべてのパラメータが存在しているかどうかにかかわらず、SQL の実行を許可します。これはデフォルトです。 |
| REQUIRED | SQL を実行するために、すべてのパラメータが存在している必要があります。                |

**例** SQLEXEC (SPNAME lookup,  
PARAMS (long\_name = birth\_state, short\_name = state),  
ALLPARAMS OPTIONAL)



## DBOP の使用

DBOP では、ストアド・プロシージャまたは問合せ内で実行された INSERT、UPDATE、DELETE、および SELECT 文をコミットします。コミットしない場合、これらの文はロールバックされる可能性があります。Oracle GoldenGate は、ソース・トランザクションと同じトランザクション境界内でコミットを発行します。

**警告** データベース、特に本番環境のデータベースに対して SQLEXEC プロシージャを実行する場合は、注意して使用してください。プロシージャによってコミットされた変更は、既存のデータを上書きすることがあります。

**構文** DBOP  
**例** SQLEXEC (SPNAME check, NOPARAMS, DBOP)

## ERROR の使用

ERROR では、ストアド・プロシージャまたは問合せに関連するエラーに対するレスポンスを定義します。明示的なエラー処理の指定がない場合、Oracle GoldenGate プロセスはエラーを検出すると異常終了します。プロシージャからプロセスにエラーを返させ、ERROR でレスポンスを指定するようにしてください。

**構文** ERROR <action>

| アクション  | 説明                                                                                                                                                                                                       |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IGNORE | Oracle GoldenGate に、ストアド・プロシージャまたは問合せに関連するすべてのエラーを無視させ、処理を継続させます。このパラメータ抽出結果は、" 列行方不明 " 状態になります。これはデフォルトです。                                                                                              |
| REPORT | ストアド・プロシージャまたは問合せに、関連するすべてのエラーを破棄ファイルにレポートさせます。DISCARDFILE パラメータで、破棄ファイルを指定しておく必要があります。このレポートは、エラーの原因の追跡に役立ちます。ここには、エラーの説明と、プロシージャまたは問合せとやり取りしたパラメータの値両方が含まれます。Oracle GoldenGate は、エラーをレポートした後、処理を継続します。 |
| RAISE  | REPERERROR パラメータで設定されたルールに従ってエラーを処理します。Oracle GoldenGate は、エラーを処理する前に、現在の TABLE 文に関連する他のストアド・プロシージャまたは問合せの処理を継続します。                                                                                      |
| FINAL  | RAISE と似ていますが、プロシージャまたは問合せに関連するエラーを検出したときに、残りのストアド・プロシージャおよび問合せがバイパスされます。エラー処理は、エラーの直後に起動されます。                                                                                                           |
| FATAL  | プロシージャまたは問合せに関連するエラーを検出したときに、即座に Oracle GoldenGate を異常終了させます。                                                                                                                                            |

**例** SQLEXEC (SPNAME check, NOPARAMS, ERROR REPORT)

## EXEC の使用

EXEC では、TABLE 文のストアド・プロシージャまたは問合せを実行する頻度、および出力パラメータを抽出する場合に結果を有効とみなす期間を制御します。

### 構文

EXEC <frequency>

| 頻度          | 説明                                                                                                                                                                         |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAP         | プロシージャまたは問合せが指定されている各ソース・ターゲットのマップで、プロシージャまたは問合せを 1 回実行します。MAP を使用する場合、同一のソース表を持つそれ以降のマップでは結果は無効になります。たとえば、ソース表が複数のターゲット表と同期を取る場合、結果は最初のソース・ターゲットのマップでのみ有効です。MAP はデフォルトです。 |
| ONCE        | Oracle GoldenGate の実行中、関連する TABLE 文の最初の呼び出し時にプロシージャまたは問合せを 1 回実行します。結果は、プロセスが実行しているかぎり有効です。                                                                                |
| TRANSACTION | プロシージャまたは問合せをソース・トランザクションで 1 回実行します。結果は、トランザクションのすべての操作に有効です。                                                                                                              |
| SOURCEROW   | プロシージャまたは問合せをソース行操作で 1 回実行します。このオプションは、ソース表を複数のターゲット表と同期し、プロシージャまたは問合せの結果がソース・ターゲット・マップングのたびに呼び出される場合に使用します。                                                               |

**例 1** 次に、ONCE の使用例を示します。

```
TABLE sales.cust, TARGET sales.cust_extended, &
SQLEXEC (SPNAME lookup, &
PARAMS (long_name = birth_state), &
EXEC ONCE), &
COLMAP (custid = custid,
birth_state_long = lookup.long_name);
```

**例 2** 次に、TRANSACTION の使用例を示します。

```
TABLE sales.cust, TARGET sales.cust_extended, &
SQLEXEC (SPNAME lookup, PARAMS (long_name = birth_state), EXEC TRANSACTION), &
COLMAP (custid = custid, &
birth_state_long = lookup.long_name);
```

**例 3** 次に、デフォルト (MAP) の誤った使用例を示します。2 つの TABLE 文で、同一のソース表と 2 つの異なるターゲット表を同期します。しかし、プロシージャ lookup の結果は、2 番目の MAP の実行までに無効になってしまい、2 番目の MAP は "列行方不明" の状態になります。この機能を正しく実装するには、SOURCEROW を使用する必要があります。

```
TABLE sales.srctab, TARGET sales.targetab, &
SQLEXEC (SPNAME lookup, PARAMS (param1 = srccol)), &
COLMAP (targcol = lookup.param2); &
TABLE sales.srctab, TARGET sales.targetab2, &
COLMAP (targcol2 = lookup.param2);
```

**例 4** 次に、SOURCEROW の使用例を示します。プロシージャが各ソース行操作で実行されるため、2 番目の MAP は有効な値を返します。

```
TABLE sales.srctab, TARGET sales.targettab, &
SQLEXEC (SPNAME lookup, &
PARAMS (param1 = srccol), EXEC SOURCEROW), &
COLMAP (targcol = lookup.param2);

TABLE sales.srctab, TARGET sales.targettab2, &
COLMAP (targcol2 = lookup.param2);
```

### ID の使用

ID は、TABLE 文内の問合せおよびストアド・プロシージャに対して、次のように使用します。

- 問合せでは、ID <logical name> を使用し、Oracle GoldenGate がこの名前を使用して問合せから返される列値を参照できるようにします。
- ストアド・プロシージャでは、ID <logical name> を使用して、たとえば 2 つの異なる列のマッピングなどのために、1 つの TABLE 文内でプロシージャを複数回呼び出せるようにします。それ以外の場合、ID は必要ありません。1 つの TABLE 文で、最大 20 のストアド・プロシージャで実行できます。これらは、パラメータ・ファイルにリストされた順に実行されます。

**構文** ID <logical name>

| コンポーネント        | 説明                                                                                      |
|----------------|-----------------------------------------------------------------------------------------|
| <logical name> | ストアド・プロシージャまたは問合せの論理名。たとえば、"lookup" という名前のプロシージャの論理名を "lookup1"、"lookup2" などのように指定できます。 |

**例 1** 次に、ID <logical name> の使用例を示します。各列マッピングが、lookup1 および lookup2 を使用して lookup という名前のストアド・プロシージャを個別にコールし、それぞれの結果を参照します。

```
TABLE sales.srctab, TARGET sales.targettab, &
SQLEXEC (SPNAME lookup, ID lookup1, PARAMS (param1 = srccol)), &
COLMAP (targcol1 = lookup1.param2), &
SQLEXEC (SPNAME lookup, ID lookup2, PARAMS (param1 = srccol)), &
COLMAP, (targcol2 = lookup2.param2);
```

**例 2** 次に、lookup という名前のストアド・プロシージャを 1 回実行する例を示します。このケースでは、実際のプロシージャ名を使用します。論理名は必要ありません。

```
TABLE sales.tab1, TARGET sales.tab2, &
SQLEXEC (SPNAME lookup), PARAMS (param1 = srccol)), &
COLMAP (targcol = lookup.param1);
```

**例 3** 次に、Oracle および SQL Server の問合せにそれぞれ ID <logical name> を使用する例を示します。この例では、このドキュメントのスペースの制約によって、SQLEXEC 文が複数の行にまたがっています。実際の SQLEXEC 文は、1 行内に含める必要があります。

```
TABLE sales.account, TARGET sales.newacct, &
SQLEXEC (ID lookup, &
QUERY "select desc_col into desc_param from lookup_table &
where code_col = :code_param", &
```

```
PARAMS (code_param = account_code)), &
COLMAP (newacct_id = account_id, &
newacct_val = lookup.desc_param);

TABLE sales.account, TARGET sales.newacct, &
SQLEXEC (ID lookup, &
QUERY "select desc_col into desc_param from lookup_table &
where code_col = ?", &
PARAMS (p1 = account_code)), &
COLMAP (newacct_id = account_id, &
newacct_val = lookup.desc_param);
```

### MAXVARCHARLEN の使用

MAXVARCHARLEN では、プロシージャまたは問合せの出力パラメータに割り当てる最大長を指定します。これを超える出力値は切り捨てられます。

**構文** MAXVARCHARLEN <num bytes>

| コンポーネント     | 説明                                                                      |
|-------------|-------------------------------------------------------------------------|
| <num bytes> | 出力パラメータに許可する最大バイト数を定義します。明示的に MAXVARCHARLEN 句を使用しない場合のデフォルトは、255 バイトです。 |

**例** MAXVARCHARLEN 100

### NOPARAMS の使用

NOPARAMS は、ストアド・プロシージャまたは問合せがパラメータを必要としない場合に、PARAMS のかわりに使用します。PARAMS 句または NOPARAMS のいずれかを指定する必要があります。

**構文** NOPARAMS

**例** SQLEXEC (SPNAME check, NOPARAMS)

### PARAMBUFSIZE の使用

PARAMBUFSIZE では、入力および出力パラメータを含むパラメータ情報を保持するメモリー・バッファの最大値を指定します。Oracle GoldenGate は、パラメータに割り当てられたメモリーと最大値との差が 500 バイト以内に達するたびに警告を発行します。

**構文** PARAMBUFSIZE <num bytes>

| コンポーネント     | 説明                                                                          |
|-------------|-----------------------------------------------------------------------------|
| <num bytes> | メモリー・バッファに許可する最大バイト数を定義します。明示的に PARAMBUFSIZE 句を使用しない場合のデフォルトは、10,000 バイトです。 |

**例** PARAMBUFSIZE 15000

## PARAMS の使用

PARAMS では、入力を受け付けるストアド・プロシージャまたは問合せのパラメータ名と、入力を提供するソース列名または Oracle GoldenGate 列変換ファンクション名を指定します。PARAMS 句または NOPARAMS のいずれかを指定する必要があります。

以下は、SQLEXEC によってサポートされるデータベース、および入力と出力パラメータにサポートされるデータ・タイプです。

### DB2 LUW および z/OS

- CHAR
- VARCHAR
- DATE
- すべての数値データ・タイプ
- BLOB データ・タイプ

### Ingres

LOB 以外のすべてのデータ・タイプ

### MySQL

TEXT および BLOB 以外のすべてのデータ・タイプ

### Oracle

次のものを除く、すべての Oracle タイプがサポートされます。

- BFILE
- BLOB
- CFILE
- CLOB
- NCLOB
- NTY

### SQL Server

- CHAR
- VARCHAR
- DATETIME
- すべての数値データ・タイプ
- 長さが 200 バイト未満のイメージおよびテキスト・データ・タイプ
- TIMESTAMP パラメータ・タイプはネイティブにサポートされませんが、他のデータ・タイプをパラメータとして使用し、ストアド・プロシージャ内の TIMESTAMP 形式に値を変換できます。

### Sybase

TEXTIMAGE および UDT 以外のすべてのデータ・タイプ

### Teradata

Oracle GoldenGate でサポートされる、すべての Teradata データ・タイプ

デフォルトでは、出力パラメータは、パラメータ当たり 255 バイトで切り捨てられます。プロシージャがこれより長いパラメータを必要とする場合は、MAXVARCHARLEN オプションを使用します。

```
構文 PARAMS (
 [OPTIONAL | REQUIRED]
 <param name> = {<source column> | <GG function>}
 [, ...]
)
```

| コンポーネント                                                                         | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OPTIONAL   REQUIRED                                                             | <p>パラメータ値が見つからないときに、プロシージャまたは問合せを実行するかどうかを決定します。</p> <ul style="list-style-type: none"> <li>◆ <b>OPTIONAL</b> は、<b>SQL</b> の実行にパラメータ値が必要ないことを示します。必要なソース列がデータベース操作で見つからない場合、またはソース列が見つからないために列変換ファンクションを完了できない場合でも、<b>SQL</b> は実行されます。</li> <li>◆ <b>OPTIONAL</b> は、Oracle 以外のすべてのデータベースのデフォルトです。Oracle では、ストアド・プロシージャ定義を取得するときに、パラメータがオプションかどうか自動的に判別されます。</li> <li>◆ <b>REQUIRED</b> は、パラメータ値が存在している必要があることを示します。パラメータ値が存在しない場合、<b>SQL</b> は実行されません。</li> </ul>                                                                                                                                                                                                                                                                                                                                                            |
| <pre>&lt;param name&gt; = {   &lt;source column&gt;     &lt;GG function }</pre> | <p>パラメータ名を、入力を提供する列またはファンクションにマップします。</p> <p>条件：</p> <ul style="list-style-type: none"> <li>◆ <b>&lt;param name&gt;</b> は、次のいずれかです。           <ul style="list-style-type: none"> <li>ストアド・プロシージャの場合、ストアド・プロシージャ内の入力を受け付ける任意のパラメータ名（参照表の列など）です。</li> <li>Oracle 問合せの場合、先行するコロンを除いた、問合せの任意の入力パラメータ名です。たとえば、<code>:param1</code> の場合は、<b>PARAMS</b> 句では <code>param1</code> と指定します。</li> <li>Oracle 以外の問合せの場合は、<b>Pn</b> と指定します（<i>n</i> は、1 から始まる文内のパラメータ番号です）。たとえば、2 つの入力パラメータを持つ問合せの場合、<b>&lt;param name&gt;</b> エントリは <b>P1</b> および <b>P2</b> になります。</li> </ul> </li> <li>◆ <b>&lt;source column&gt;</b> は、ソース列名です。デフォルトでは、指定した列が（圧縮更新であるために）ログに存在しない場合、このパラメータはプロシージャまたは問合せでこのパラメータに指定されているデフォルト値を使用します。</li> <li>◆ <b>&lt;GG function&gt;</b> は、Oracle GoldenGate 列変換ファンクション名です。列変換ファンクションの詳細は、第 4 章を参照してください。</li> </ul> |

**例** 次に、`account` 表からターゲット表 `newacct` にデータをマップする例を示します。`account` 表からのレコードを処理するとき、Oracle GoldenGate は、列マップを実行する前に `lookup` ストアド・プロシージャを実行します。プロシージャの `code_param` パラメータは、`account_code` ソース列から入力を受け付けます。

```
TABLE sales.account, TARGET sales.newacct, &
SQLEXEC (SPNAME lookup, PARAMS (code_param = account_code)), &
COLMAP (newacct_id = account_id, &
newacct_val = lookup.desc_param);
```

## TRACE の使用

TRACE では、入力および出力パラメータをレポート・ファイルに記録します。

SQLEXEC 証跡が有効化されているサンプルの破棄ファイル：

```
Input parameter values...

LMS_TABLE: INTERACTION_ATTR_VALUES
KEY1: 2818249
KEY2: 1
Report File:

From Table MASTER.INTERACTION_ATTR_VALUES to MASTER.INTERACTION_ATTR_VALUES:
inserts: 0
updates: 0
deletes: 0
discards: 1

Stored procedure GGS_INTERACTION_ATTR_VALUES:
 attempts: 2
 successful: 0
```

**構文**      TRACE {ALL | ERROR}

| アクション | 説明                                                                |
|-------|-------------------------------------------------------------------|
| ALL   | 呼び出された各プロシージャまたは問合せの入力および出力パラメータをレポート・ファイルに書き込みます。これはデフォルトです。     |
| ERROR | SQL エラーの発生後にのみ、呼び出された各プロシージャまたは問合せの入力および出力パラメータをレポート・ファイルに書き込みます。 |

**例**      SQLEXEC (SPNAME lookup, PARAMS (long\_name = birth\_state, short\_name = state), TRACE ERROR)

## SQLPREDICATE の使用

SQLPREDICATE では、Extract が初期ロードの準備で表からデータを選択するときに使用する SELECT 文に、通常の SQL WHERE 句を含めます。SQLPREDICATE では、選択条件で返されるレコードがキー値によって順序付けされます。

SQLPREDICATE は、WHERE または FILTER オプションよりも初期ロードに適した選択方法です。これは SQL 文を直接適用し、他のオプションのように Oracle GoldenGate がフィルタリング前にすべてのレコードをフェッチする必要がないため、はるかに処理が高速です。

Oracle 表の場合、SQLPREDICATE を使用すると UNDO セグメントに保持されるデータ量が削減され、スナップショットが古すぎるために発生するエラーが低減します。非常に大きな表をロードするときに有益です。

SQLPREDICATE 句を使用することにより、複数の Extract プロセス間で大きな表の行をパーティション化できます。この構成では、並列配信ロード処理も活用できます。

SQLPREDICATE では、タイムスタンプやその他の基準に基づいてデータを選択し、抽出してターゲット表にロードする行を制限するためにも使用できます。また SQLPREDICATE では、ORDER BY 句やその他のタイプの選択句も使用できます。

最適なパフォーマンスを得るためには、WHERE 句の一部として指定する列がキーまたは索引の一部である必要があります。そうでない場合は、完全な表スキャンが必要になるため、SELECT 文の効率が低下します。

このパラメータは、Oracle、DB2 LUW および z/OS、SQL Server、Teradata データベースに有効です。表から直接レコードを選択する SELECT 文が使用されるとみなされるため、変更データ同期ではなく、初期ロード・プロセスでのみこれを使用してください。データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。

**構文** TABLE <table spec>, SQLPREDICATE "WHERE <where clause>";

| コンポーネント   | 説明               |
|-----------|------------------|
| WHERE     | 必須のキーワード。        |
| <WHERE 句> | 有効な SQL WHERE 句。 |

**例** SQLPREDICATE "where state = 'CO' and city = 'DENVER'"

## TARGETDEF の使用

TARGETDEF では、ターゲット定義テンプレートを指定します。定義テンプレートは、特定のターゲット表に対して DEFGEN が実行されるときに、このオブジェクトの定義に基づいて作成されます。テンプレートが作成されると、この表と同一の定義を持つ新しいターゲット表は、DEFGEN を実行せずに、かつ Extract の停止と起動を伴わずに追加できます。TARGETDEF で指定されたテンプレートの定義は、定義の参照に使用されます。DEFGEN の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文** TABLE <table spec>, TARGETDEF <definitions template>;

| 引数                     | 説明                                                                                                 |
|------------------------|----------------------------------------------------------------------------------------------------|
| <definitions template> | DEFGEN パラメータ・ファイルの TABLE の DEF オプションで指定されている定義テンプレート名。テンプレートに含まれる定義は、この TABLE 文の表の定義と同一である必要があります。 |

**例** TABLE acct.cust\*, TARGET acc.cust\*, DEF custdef, TARGETDEF tcustdef;

## TOKENS の使用

TOKENS では、ユーザー・トークンを定義し、データと関連付けます。トークンを使用すると、データを抽出し、トレイル・レコード・ヘッダーのユーザー・トークン・エリア内に保持できます。トークン・データを様々な方法で取得および使用することにより、Oracle GoldenGate のデータ提供方法をカスタマイズできます。たとえば、列マップ、SQLEXEC で呼び出されるストアド・プロシージャ、またはマクロでトークン・データを使用できます。

ターゲット表で定義済トークン・データを使用するには、Replicat 用 MAP 文の COLMAP 句で、@TOKEN 列変換ファンクションを使用します。@TOKEN ファンクションは、トークン名をターゲット列にマップします。

データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。

トークンの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。



**構文**           TABLE <table spec>, TOKENS (<token name> = <token data> [, ...]) ;

| コンポーネント      | 説明                                                                                |
|--------------|-----------------------------------------------------------------------------------|
| <token name> | トークンに付ける名前。任意の英数字を使用できます（大 / 小文字区別なし）。                                            |
| <token data> | 2000 バイトまでの文字列。データは、二重引用符で囲まれた定数、または Oracle GoldenGate 列変換ファンクションの結果のいずれかを指定できます。 |

**例**           次に、TK-OSUSER、TK-GROUP、および TK-HOST という名前のトークンを作成し、これらのトークンを @GETENV ファンクションで取得したトークン・データにマップする例を示します。

```
TABLE ora.oratest, TOKENS (
TK-OSUSER = @GETENV ("GGENVIRONMENT" , "OSUSERNAME"),
TK-GROUP = @GETENV ("GGENVIRONMENT" , "GROUPNAME")
TK-HOST = @GETENV ("GGENVIRONMENT" , "HOSTNAME"));
```

## TRIMSPACES および NOTRIMSPACES の使用

TRIMSPACES および NOTRIMSPACES では、末尾の空白を切り捨てるかどうかを制御します。これらのパラメータは、CHAR から VARCHAR への列マッピングにのみ影響します。デフォルトは TRIMSPACES です。

TRIMSPACES および NOTRIMSPACES は、異なる TABLE 文および文グループに対する切捨て機能を有効化または無効化するために、パラメータ・ファイルのルート・レベルでも使用できます。

**構文**           TABLE <table spec>, {TRIMSPACES | NOTRIMSPACES};

**例**           次に、最初の 2 つの表に対してはデフォルトの末尾切捨てを行い、最後の 2 つでは末尾切捨てを行わない例を示します。

```
TABLE fin.src1;
TABLE fin.src2;
TABLE fin.src3, NOTRIMSPACES;
TABLE fin.src4, NOTRIMSPACES;
```

## WHERE の使用

WHERE では、条件文に基づいてレコードを選択します。データ・ポンプ Extract グループにパススルー・モードで処理される表には、このオプションを使用しないでください。拡張 ASCII または出力不可能文字を含む Unicode 列または文字列を使用するには、320 ページを参照してください。WHERE 句の使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**構文**           TABLE <table spec>, WHERE (<where clause>);

| コンポーネント   | 説明                                                                                                                                                    |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <WHERE 句> | 次の例のように、条件に基づいてレコードを選択します。<br>WHERE (branch = "NY")<br><br>次の表に、許可されている WHERE 演算子を示します。<br><br>WHERE は、主キー更新操作の一部として条件文の主キー列のビフォア・イメージの評価をサポートしていません。 |

表 50 許可されている WHERE 演算子

| 演算子             | 例:                                                                                        |
|-----------------|-------------------------------------------------------------------------------------------|
| 列名              | PRODUCT_AMT                                                                               |
| 数値              | -123, 5500.123                                                                            |
| 引用符で囲まれたリテラル文字列 | "AUTO", "Ca"                                                                              |
| 列テスト            | @NULL, @PRESENT, @ABSENT (レコードの列が NULL、存在、不在かのテスト)。これらのテストは、Oracle GoldenGate に組み込まれています。 |
| 比較演算子           | =, <>, >, <, >=, <=                                                                       |
| 結合演算子           | AND, OR                                                                                   |
| グループ化用カッコ       | 複数の要素を論理的にグループ化するには、開きおよび閉じカッコを使用します。                                                     |

**例** 次の WHERE 句は、TABLE または MAP 文に含まれる場合の WHERE 句での Unicode エスケープ・シーケンスの使用を示したものです。それぞれ、「Élise」、「Zoé」および「Véronique」の入力文字内の別の場所でアキュート・アクセントが付いたラテン文字の小文字の e (コード・ポイント U+00E9) がエスケープされています。

```
WHERE (FIRSTNAME <> "\u00e9lise")
WHERE (FIRSTNAME <> "Zo\u00e9")
WHERE (FIRSTNAME <> "V\u00e9ronique")
```

## Replicat 用 TABLE

Replicat パラメータ・ファイルの TABLE パラメータでは、EVENTACTIONS で指定されているイベント・アクションの対象にするデータ・レコードをトレイルから限定するフィルタリング・ルールを指定します。

**警告** EVENTACTIONS は、ソース・データベースが Teradata で、Extract が最大パフォーマンス・モードで構成されている場合にはサポートされません。

Replicat 用 TABLE 文は、Replicat 用 MAP 文に似ていますが、この文では TARGET 句によってデータ・レコードのソース表からターゲット表へのマッピングを行いません。Replicat 用の TABLE は、イベント・レコードを検出したときに Replicat が実行するデータ操作以外のアクションをトリガーする手段としてのみ使用します。

ターゲット表が指定されないため、次のような制限があります。

- Replicat が表名または列をターゲット表にマップするためのオプションや、Replicat がデータを操作するためのオプションは使用できません。

- ASSUMETARGETDEFS を使用するには、Replicat が表定義を問合わせるターゲット表名が必要になるため、Replicat 用 TABLE 文が含まれていると同じパラメータ・ファイルでは ASSUMETARGETDEFS パラメータを使用できません。Replicat にソース表の定義を提供するために、ソース定義ファイルを作成する必要があります。このファイルをターゲット・システムに送信し、Replicat パラメータ・ファイルで SOURCEDEFS パラメータを使用してファイルのパス名を指定します。
- イベント・レコード自体は、Replicat によってターゲット・データベースに適用されません。EVENTACTIONS オプションの IGNORE または DISCARD を指定する必要があります。

TABLE 文は、セミコロンを使用して終了します。

#### 構文

```
TABLE <table spec>,
[, SQLEXEC (<SQL specification>), BEFOREFILTER]
[, FILTER (<filter specification>)]
[, WHERE (<where clause>)]
{ , EVENTACTIONS ({IGNORE | DISCARD} [<action>])}
;
```

表名でサポートされている文字、および次に示す構文オプションの説明は、Replicat 用 MAP パラメータの説明を参照してください。

#### 例

次に、特定の受注番号 (order\_no=1) に対する挿入操作が含まれる受注トランザクションを Replicat にトレースさせる例を示します。トレース情報は、order\_1.trc トレース・ファイルに書き込まれます。MAP パラメータでは、ソース表とターゲット表のマッピングを指定します。

```
MAP sales.order, TARGET rpt.order;
```

```
TABLE sales.order,
FILTER (@GETENV ("GGHEADER", "OPTYPE") = "INSERT" AND order_no = 1), &
EVENTACTIONS (TRACE order_1.trc TRANSACTION);
```

イベント・マーカー・システムの他の使用例および詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

## TABLEEXCLUDE

**適用対象** Extract

TABLEEXCLUDE パラメータは、TABLE および SEQUENCE パラメータとともに使用して、ワイルドカード指定から明示的に表と順序を除外します。TABLEEXCLUDE は、除外するオブジェクトを含むすべての TABLE および SEQUENCE 文よりも先に指定する必要があります。

**デフォルト** なし

**構文**           TABLEEXCLUDE <exclude specification> [NORENAME]

| 引数                      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <exclude specification> | 除外するオブジェクトの名前またはワイルドカード指定。TABLE または SEQUENCE 文でオブジェクトをワイルドカード指定するときと同じワイルドカード・ルールが TABLEEXCLUDE に適用されます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| NORENAME                | <p>(DDL サポートが有効な場合、Oracle および Teradata) この TABLEEXCLUDE 文のオブジェクトが Oracle GoldenGate 構成に含まれる名前に変更されることを許可しません。この状況が発生する例を示します。</p> <p>TableA は除外され、tableB は除外されていないときに、tableA の名前が tableB に変更される。</p> <p>NORENAME は、構成に含まれないオブジェクトが、構成に含まれる名前に変更されると警告を生成します。この状況が最も発生しやすいのは、TABLE 文でワイルドカードが使用されているときです。NORENAME は、たとえば、名前が変更されたオブジェクトが Oracle GoldenGate にサポートされない構造を持つ場合や、レプリケート対象外のオブジェクトのときに、エラーの発生を防止するために有効に利用できます。</p> <p>Oracle RAC では、NORENAME によって、除外オブジェクトのステータスが変更されたときに、このオブジェクトの追跡のために通常なら必要になる無駄なプロセスも排除できます。また、NORENAME によって、名前が変更されて構成に含まれたオブジェクトが、サポートされていない構造をもつ場合にもエラーを防止できます。</p> <p>NORENAME の代替としては、DDOPTIONS を NOCROSSRENAME オプションとともに使用します。NOCROSSRENAME は、パラメータ・ファイル内のすべての TABLE、SEQUENCE、および TABLEEXCLUDE 文全体にグローバルに処理除外を適用します。159 ページを参照してください。</p> |

**例**           次の例では、TABLE 文は TEST という名前の表を除くすべての表を取得します。

```
TABLEEXCLUDE fin.TEST
TABLE fin.*;
```

## TARGETDB

**適用対象**   Replicat

TARGETDB は、接続情報の一部としてデータ・ソース名を必要とするデータベース用のパラメータです。TARGETDB エントリに続く MAP 文に指定されているターゲット表は、指定したデータベースのものであるとみなされます。

このパラメータは、データベースへのログインで要求される認証方法に応じて、次に示すように USERID パラメータとともに使用する必要がある場合があります。

- データベース・ログイン情報が要求されるデータベースの場合は、同じパラメータ文内で（必要な場合）TARGETDB を USERID とともに使用する必要があります。
- c-tree データベースの場合、TARGETEDB ではサーバーの別名を指定します。

- オペレーティングシステム・レベルでの認証を許可するデータベースの場合は、USERID なしで TARGETDB を指定できます。

**デフォルト** なし

**構文** TARGETDB <data source>

| 引数            | 説明        |
|---------------|-----------|
| <data source> | データ・ソース名。 |

**例 1** TARGETDB mydb

**例 2** TARGETDB mydb, USERID ggs, PASSWORD ggs123

## TARGETDEFS

**適用対象** Extract（プライマリおよびデータ・ポンプ）

TARGETDEFS パラメータは、ターゲットが **Enscribe** ファイルの場合に使用します。TARGETDEFS には、ターゲット・システム上に存在する表およびファイルのデータ定義を含む、ソース・システムまたは仲介システム上のファイル名を指定します。ターゲットとして **Enscribe** ファイルが指定されている **TABLE** 文の前に、少なくとも 1 つの TARGETDEFS エントリを指定します。

ターゲット定義ファイルを生成するには、**DEFGEN** ユーティリティを使用します。**Extract** を起動する前に、このファイルをソース・システムまたは仲介システムに送信してください。

たとえば、各 TARGETDEFS ファイルに別々のアプリケーションの定義が保持されている場合など、異なる定義を使用するために複数のターゲット定義ファイルが必要なときは、パラメータ・ファイルで複数の TARGETDEFS 文を使用できます。

**デフォルト** なし

**構文** TARGETDEFS <file name>

| 引数          | 説明             |
|-------------|----------------|
| <file name> | データ定義の取得元ファイル。 |

**例 1** TARGETDEFS C:\repodbc\sales.def

**例 2** TARGETDEFS /ggs/dirdef/ODBC/tandem\_defs

## TCPSOURCETIMER | NOTCPSOURCETIMER

**適用対象** Extract

TCPSOURCETIMER および NOTCPSOURCETIMER パラメータでは、レポートでの使用を目的として、Oracle GoldenGate 環境内にレプリケートされた操作のタイムスタンプを管理します。

デフォルトは TCPSOURCETIMER です。これにより、データ・レコードが他のシステムに送信されるときにタイムスタンプが調整されるため、同期ラグの理解が容易になります。

NOTCPSOURCETIMER では、元のタイムスタンプ値を保持します。NOTCPSOURCETIMER は、双方向構成でタイムスタンプベースでの競合解決を使用するときに使用します。NOTCPSOURCETIMER は、@GETENV 列変換関数 "GGHEADER"、"COMMITTIMESTAMP" を参照するユーザー・トークンを使用するときに使用します。

TCPSOURCETIMER および NOTCPSOURCETIMER はグローバル・パラメータで、Extract パラメータ・ファイル内のすべての TABLE 文に適用されます。

**デフォルト** TCPSOURCETIMER

**構文** TCPSOURCETIMER | NOTCPSOURCETIMER

## THREADOPTIONS

**適用対象** Extract

THREADOPTIONS パラメータでは、Oracle RAC (Real Application Cluster) 環境での Extract の動作方法を制御します。

### パフォーマンス・オプションの使用

Oracle GoldenGate は、ターゲット・システムに送信する前に、データをメモリーのキューに入れます。THREADOPTIONS パラメータの INQUEUE SIZE および OUTQUEUE SIZE オプションでは、キューに入れるデータ量を決定します。この値が大きいほど、大量のデータを処理するときのパフォーマンスが向上します。この値を小さくすると、アクティビティが非常に少ない環境でデータがターゲットにより迅速に送信されます。デフォルト値から使用を開始してください。一般的な値は、100 ~ 1500 です。ほとんどの環境では、各オプションとも 1000 で十分なパフォーマンスが得られるはずです。

この 2 つのパラメータに加え、AIX ユーザーは、環境変数 AIXTHREAD\_SCOPE を S (システム・スコープ) に設定して、プロセスの同時実行をサポートする複数 CPU の使用を指定することにより、パフォーマンスを改善できることがあります。システム範囲を使用するには、Manager プロセスを起動するユーザーの .profile ファイルに次を追加するか、GGSCI を起動する前に、この変数を手動でエクスポートします。

```
AIXTHREAD_SCOPE=S
export AIXTHREAD_SCOPE
```

GGSCI、Manager、および Extract を停止および再起動して、変更を有効にします。

**デフォルト** なし

**構文**

```

THREADOPTIONS
[EOFDELAYMS <milliseconds>]
[IOLATENCY <milliseconds>]
[INQUEUESIZE <n>]
[MAXCOMMITPROPAGATIONDELAY <milliseconds>]
[OUTQUEUESIZE <n>]

```

| 引数                                          | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EOFDELAYMS <milliseconds>                   | <p>REDO ログの論理的な末尾に到達した後に、<b>Extract</b> プロセスが処理する追加データを探すまでの遅延時間を指定します。デフォルトは <b>250</b> ミリ秒です。この値を増やすと、ソース表に変更が行われた時間とターゲット表に変更が適用される時間のラグが増える可能性があります。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| INQUEUESIZE <n>                             | <p>各 <b>Extract</b> スレッドの入力キューのキュー・エントリ数を指定します。有効な値は <b>16 ~ 65535</b> です。デフォルトは <b>255</b> です。</p> <p>ほとんどのケースでは、デフォルト値で問題なく機能するはずです。<b>INQUEUESIZE</b> は、少なくとも <b>OUTQUEUESIZE</b> の 2 倍に設定する必要があります。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| IOLATENCY <milliseconds>                    | <p>データベースで構成されている最大コミット伝播遅延と、<b>Oracle GoldenGate</b> によって使用される内部値との時間差を指定します。有効な値は、<b>0 ~ 180000</b> ミリ秒（3 分）です。内部 I/O 遅延を表すデフォルト値は、<b>1500</b>（1.5 秒）です。</p> <p><b>Extract</b> がアーカイブ・ログ専用（<b>ALO</b>）モードの場合は無効です。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| MAXCOMMITPROPAGATIONDELAY<br><milliseconds> | <p>トランザクションがコミットされた時間から、アイドルの <b>REDO</b> ログを読み込む時間までの遅延時間を指定します。この遅延を指定することにより、<b>Oracle</b> が <b>REDO</b> ログにデータを書き込む時点から、<b>Oracle GoldenGate</b> が <b>REDO</b> ログからデータを読み取るまでの時間差を考慮できます。共有 <b>RAC</b> データベース・ドライブへのアクセスに競合がある場合、この時間差は大きくなる可能性があります。キャッシングおよびシリアルライズ機能を持ち、このような遅延を最小化または排除する <b>SAN</b> および <b>NFS</b> デバイス上にデータベースが存在する場合は、低い値が適切です。</p> <p>この値は、0 より大きく、<b>90000</b> ミリ秒（90 秒）未満に設定する必要があります。デフォルトは 3 秒です。この値は、<b>Oracle</b> の同じ名前のパラメータの最小値である <b>2000</b> ミリ秒よりも常に大きい値に設定し、また <b>Oracle</b> 値よりも絶対に低い値に設定しないでください。<b>Oracle</b> の値を確認するには、<b>DBA</b> 権限を持つユーザーとして接続し、<b>SQL*Plus</b> で次のコマンドを発行します。</p> <pre>show parameter max_commit</pre> <p><b>Extract</b> がアーカイブ・ログ専用（<b>ALO</b>）モードの場合は無効です。</p> |
| OUTQUEUESIZE <n>                            | <p>各 <b>Extract</b> スレッドの出力キューのキュー・エントリ数を指定します。有効な値は <b>8 ~ 65535</b> です。デフォルトは <b>127</b> です。</p> <p>ほとんどのケースでは、デフォルト値で問題なく機能するはずです。<b>OUTQUEUESIZE</b> は、<b>INQUEUESIZE</b> の約半分の値に設定する必要があります。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

## TRACE/TRACE2

**適用対象** Extract および Replicat

Extract または Replicat 処理情報を取得して、処理のボトルネックを識別するには、TRACE および TRACE2 パラメータを使用します。

- TRACE では、段階ごとの処理情報が表示されます。
- TRACE2 では、最も多くの時間を消費している Extract または Replicat 上のコード・セグメントが識別されます。

両方とも、DML および DDL の証跡がサポートされます。

またトレースは、GGSCI の SEND EXTRACT または SEND REPLICAT コマンドを使用して、有効化と無効化を切り替えることもできます。GGSCI コマンドの詳細は、第 1 章を参照してください。

トレースによって重大な処理のボトルネックが明らかになった場合は、Oracle サポートに連絡してください。詳細は、<http://support.oracle.com> を参照してください。

**デフォルト** トレースしない

**構文** TRACE | TRACE2  
[, DDL[INCLUDE] | DDLONLY]  
[, [FILE] <file name>]

| 引数                     | 説明                                                                                                                                                                                                                                                                  |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DDL[INCLUDE]   DDLONLY | <p>(Replicat のみ) DDL トレースを有効化し、トレース・レポートに DDL トレースを取得する方法を指定します。</p> <ul style="list-style-type: none"> <li>◆ DDL[INCLUDE] では、DDL のトレースに加え、トランザクション・データ処理もトレースします。DDL または DDLINCLUDE のいずれかが有効です。</li> <li>◆ DDLONLY では、DDL をトレースし、トランザクション・データはトレースしません。</li> </ul> |
| [FILE]<br><file name>  | <p>Oracle GoldenGate のトレース情報の記録先ファイルの完全修飾名。FILE キーワードはオプションですが、次の例のように、ファイル名の後に他のオプションを指定する場合は使用する必要があります。</p> <p>TRACE FILE &lt;file name&gt; DDLINCLUDE</p> <p>次の例のように、ファイル名の後に他のオプションを指定しない場合は、FILE キーワードを省略できます。</p> <p>TRACE DDLINCLUDE &lt;file name&gt;</p>  |

**例** TRACE /home/ggs/dirrpt/trace.trc



# TRACETABLE | NOTRACETABLE

**適用対象** Extract および Replicat

TRACETABLE および NOTRACETABLE パラメータは、Oracle データベースとともに使用し、ADD TRACETABLE コマンドで作成されたトレース表を特定します。TRACETABLE は、トレース表がデフォルトの GGS\_TRACE 以外の名前で作成されている場合にのみ必要です。GGS\_TRACE という名前のトレース表がデータベースに存在する場合は、トレース表機能は自動的に有効になり、TRACETABLE の指定は必要ありません。

トレース表は、双方向同期で Extract に Replicat トランザクションを識別させるために使用します。

TRACETABLE が使用される場合、Extract および Replicat パラメータ・ファイルの両方に存在している必要があります。

- Replicat パラメータ・ファイルでは、TRACETABLE は、Replicat によって各トランザクションの開始部分に証跡表への操作が書き込まれるようにします。
- Extract パラメータ・ファイルでは、TRACETABLE は、Extract で、この証跡表に対する操作で開始される任意のトランザクションが Replicat トランザクションとして識別されるようにします。

NOTRACETABLE を指定すると、Replicat はトレース表に操作を書き込まないため、Extract は Replicat トランザクションを認識できません。

Replicat トランザクションを Extract に抽出させるか無視させるかを制御するには、GETREPLICATES および IGNOREREPLICATES パラメータを使用します。199 ページを参照してください。

双方向同期の構成方法は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

**デフォルト** GGS\_TRACE

**構文** TRACETABLE [<owner>.]<table name> | NOTRACETABLE

| 引数                     | 説明                                                                             |
|------------------------|--------------------------------------------------------------------------------|
| [<owner>.]<table name> | トレース表の所有者（オプション）および名前。所有者を指定しない場合は、USERID パラメータで指定されているユーザーのスキーマに所有されるとみなされます。 |

**例** TRACETABLE ggs.excl\_trans

# TRANLOGOPTIONS

**適用対象** Extract

TRANLOGOPTIONS パラメータでは、Extract とトランザクション・ログとのやり取り方法を制御します。TRANLOGOPTIONS 文を同一のパラメータ・ファイル内で複数回使用することも、（オプションに許可されている場合は）同一の TRANLOGOPTIONS 文内で複数のオプションを指定することもできます。

各 TRANLOGOPTIONS オプションは、対象のデータベースに対してのみ使用してください。

**デフォルト** なし

**構文**

```
TRANLOGOPTIONS {
[ALTARCHIVEDLOGFORMAT <string>] [INSTANCE <instance_name>] [THREADID <id>]
[ALTARCHIVELOGDEST [PRIMARY] [INSTANCE <instance_name>] <path name>]
[ALTARCHIVELOGDEST ("<Backup Path>" [FILESPEC "<File Pattern>"]
[[NOT] RECURSIVE] [PRIMARY]])]
[ALTONLINELOGS (<path> [, <path>] [...])]
[ARCHIVEDLOGONLY]
[ASMBUFSIZE <size>]
[ASMUSER SYS@<ASM_instance>, ASMPASSWORD <password>
[, ENCRYPTKEY DEFAULT | ENCRYPTKEY <keyname>]]
[BUFSIZE <size>]
[COMPLETEARCHIVEDLOGONLY]
[DBLOGREADER]
[DBLOGREADERBUFSIZE <buffer size>]
[EXCLUDETRANS <trans name>]
[EXCLUDEUSER <user name>]
[EXCLUDEUSERID <Oracle uid>]
[FILTERTABLE <table_name>]
[IGNOREDATA_CAPTURECHANGES | NOIGNOREDATA_CAPTURECHANGES]
[IGNOREDIRECTLOADINSERTS]
[LOGSOURCE <platform>]
[MANAGESECONDARYTRUNCATIONPOINT | NOMANAGESECONDARYTRUNCATIONPOINT]
[MAXWARNEOF <seconds>]
[NODDLCHANGEWARNING]
[NOFLUSH]
[PATHMAP <NFS mount point> <log path>]
[PURGEORPHANEDTRANSACTIONS | NOPURGEORPHANEDTRANSACTIONS]
[QUERYRETRYCOUNT <number of retries>] |
[READBUFFER <byte length>]
[READTIMEOUT <milliseconds>]
[REQUIRELONGDATA_CAPTURECHANGES | NOREQUIRELONGDATA_CAPTURECHANGES]
[TRANSCLEANUPFREQUENCY <minutes>]
[UNPRIVILEGED]
[VAMCOMPATIBILITY {1 | 2}]
}
[, ...]
```

| オプション                                                                                          | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALTARCHIVEDLOGFORMAT<br><string><br>[INSTANCE <instance_name>]<br>[THREADID <id>]              | <p>(Oracle) ソース・データベースのアーカイブ・ログ・フォーマットをオーバーライドする文字列を指定します。&lt;string&gt; は、Oracle パラメータの LOG_ARCHIVE_FORMAT と同じ指定子を受け付けます。Extract は、指定されるフォーマット指定子を使用してログ・ファイル名を導出します。</p> <p>例：</p> <pre>arch_%T.arc</pre> <p>RAC 上で ALTARCHIVEDLOGFORMAT を使用するときは、各ノード上で ALTARCHIVEDLOGFORMAT パラメータを使用します。</p> <p>Extract にログ・ストリームを識別させるには、次の条件の 1 つ満たす必要があります。</p> <ul style="list-style-type: none"> <li>◆ INSTANCE または THREADID オプションを使用して、スレッド指定子 ("%t" または "%T") を含むログ名フォーマットを指定します。</li> <li>◆ または、各スレッドに一意のログ・ディレクトリを使用します。この目的には、ALTARCHIVELOGDEST を使用します。（スレッド指定子を使用するログ名フォーマットを使用している場合も、一意のディレクトリの使用をお勧めします。）</li> </ul> <p>注意：Extract がユーザー定義ログ・フォーマットを見つけられない場合や、スレッドにデフォルト・フォーマットが指定されていない場合は、データベースの 1 つのスレッドに問合せで取得したデフォルトのログ・フォーマットが別のスレッドにも適用されます。</p> <p>次のオプションは、RAC とともに使用します。Extract は、データベース・カタログに対して、提供された入力を検証します。</p> <ul style="list-style-type: none"> <li>◆ INSTANCE &lt;instance_name&gt; は、特定の Oracle インスタンスに ALTARCHIVEDLOGFORMAT を適用します。</li> </ul> <p>例：</p> <pre>TRANLOGOPTIONS ALTARCHIVEDLOGFORMAT &amp;<br/>INSTANCE rac1 log_%t_%s_%r.arc</pre> <ul style="list-style-type: none"> <li>◆ THREADID &lt;id&gt; は、指定のログ・フォーマットを持つインスタンスのスレッド番号を指定します。</li> </ul> <p>例：</p> <pre>TRANLOGOPTIONS ALTARCHIVEDLOGFORMAT &amp;<br/>THREADID 2 log_%t_%s_%r.arc</pre> |
| ALTARCHIVELOGDEST<br>[PRIMARY]<br>[INSTANCE <instance_name>]<br>[THREADID <id>]<br><path name> | <p>(Oracle) デフォルト以外の場所に存在するときに、Extract にアーカイブ・ログまたはバックアップ Oracle トランザクション・ログの場所を示します。Extract はまず指定された場所を確認し、次にデフォルトの場所を確認します。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

| オプション | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <ul style="list-style-type: none"> <li>◆ &lt;path name&gt; には、代替ディレクトリのアーカイブ・ログの完全修飾パスを指定します。このディレクトリは、<b>Oracle GoldenGate</b> が実行されているノードに NFS マウントされている必要があります。このマウント・ポイントを <b>ALTARCHIVELOGDEST</b> に使用します。</li> </ul> <p>オプション：</p> <ul style="list-style-type: none"> <li>◆ <b>PRIMARY</b> では、代替場所でログが見つからない場合に、<b>Extract</b> がデフォルトのログの場所を確認することを防ぎます。<b>ALTARCHIVELOGDEST</b> パスのみが確認されます。<b>PRIMARY</b> は、アーカイブ・ログ専用（<b>ALO</b>）モードで実行している <b>Extract</b> のデフォルトで、それ以外の場合にはオプションです。</li> <li>◆ <b>INSTANCE &lt;instance_name&gt;</b> は、特定の <b>Oracle</b> インスタンスに、指定された <b>ALTARCHIVELOGDEST</b> 動作を適用します。<b>RAC</b> でこのオプションを使用する場合は、各ノードに <b>ALTARCHIVELOGDEST</b> パラメータを指定する必要があります。</li> <li>◆ <b>THREADID &lt;id&gt;</b> は、特定のスレッド番号に、指定された <b>ALTARCHIVELOGDEST</b> 動作を適用します。</li> </ul> <p>どの <b>Oracle</b> インスタンスにも、複数の <b>ALTARCHIVELOGDEST</b> パラメータを指定できます。そのような場合、<b>Extract</b> は <b>ALTARCHIVELOGDEST</b> で指定されている順番で各場所を検索します。</p> <p>次に例を示します。</p> <pre>TRANLOGOPTIONS ALTARCHIVELOGDEST PRIMARY INSTANCE rac1 /disk1/node1/arch, ALTARCHIVELOGDEST INSTANCE rac1 /disk2/node1/arch, ALTARCHIVELOGDEST INSTANCE rac2 /disk1/node2/arch</pre> <p>この例では、<b>Extract</b> はインスタンス "<b>rac1</b>" に関連するログを <b>/disk1/node1/arch</b> で検索し、最初の検索に失敗すると、次は <b>/disk2/node1/arch</b> で検索します。<b>Extract</b> は、"<b>rac1</b>" のデフォルトの場所を確認しません。"<b>rac2</b>" については、まず <b>/disk1/node2/arch</b> で検索し、次にデフォルトの場所で検索します。</p> |

| オプション                                                                                                                           | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALTARCHIVELOGDEST (" <b>&lt;backup path&gt;</b> "<br>[FILESPEC " <b>&lt;file pattern&gt;</b> "] [[NOT]<br>RECURSIVE] [PRIMARY]) | <p>(SQL Server) デフォルト以外の場所に存在するときに、Extract にアーカイブまたはバックアップ・ログの場所を示します。Extract はまず指定された場所を確認し、次にデフォルトの場所を確認します。</p> <ul style="list-style-type: none"> <li>◆ パラメータ引数はカッコで囲みます。</li> <li>◆ "<b>&lt;backup path&gt;</b>" には、バックアップ・ログのパス名を指定します (二重引用符で囲みます)。最後のバックスラッシュ (\) ・デリミタの後に、ワイルドカード記号を使用できます。<br/>アスタリスク (*) は、ゼロまたはそれ以上の文字と一致します。<br/>疑問符 (?) は一文字と一致します。<br/>NOT RECURSIVE を使用している場合は、このオプションを使用しないでください。</li> </ul> <p>SQL Server の Extract パラメータ・ファイルには、1 つの TRANLOGOPTIONS ALTARCHIVELOGDEST エントリのみ指定できます。複数のエントリが存在する場合は、最後のエントリのみが使用されます。</p> <p>オプション:</p> <ul style="list-style-type: none"> <li>◆ FILESPEC "<b>&lt;file pattern&gt;</b>" には、"<b>&lt;backup path&gt;</b>" 内のファイル・パターンを指定します。ファイル・パターンは二重引用符で囲みます。<br/>アスタリスク (*) は、ゼロまたはそれ以上の文字と一致します。<br/>疑問符 (?) は一文字と一致します。<br/>バックスラッシュ (\) ・デリミタは使用しないでください。<br/>バックスラッシュを使用すると他のパスも指定できますが、これは無効です。</li> <li>◆ [[NOT] RECURSIVE] では、"<b>&lt;backup path&gt;</b>" で指定されているファイルを再帰的に検索 (すべてのサブディレクトリも検索) するかどうかを指定します。</li> <li>◆ PRIMARY では、代替場所でログが見つからない場合に、Extract がデフォルトのログの場所を確認することを防ぎます。<br/>ALTARCHIVELOGDEST パスのみが確認されます。これはデフォルトです。</li> </ul> |
| ALTONLINELOGS (<br><b>&lt;path&gt;</b><br>[, <b>&lt;path&gt;</b> ]<br>[...]<br>)                                                | <p>(SQL Server) SQL Server オンライン・トランザクション・ログの代替場所を指定します。複数のオンライン・ログ・ファイルを指定するには、各ファイルをコンマで区切ります。パスに空白が含まれる場合は、次の例のように、引用符で囲みます。</p> <p>ALTONLINELOGS ("c:\SQL Server\Data\NorthWnd1.ldf")</p> <p>入力指定全体をカッコで囲みます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

| オプション                                                                                                     | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ARCHIVEDLOGONLY                                                                                           | <p>(Oracle) ARCHIVEDLOGONLY では、v\$log および v\$archived_log などのシステム・ビューのログの問合せや検証を行わずに、Extract にアーカイブ・ログからのみ読み取りを実行させます。このパラメータにより、Extract はアーカイブ・ログ専用 (ALO) モードに入ります。詳細は、『Oracle GoldenGate Oracle インストレーションおよびセットアップ・ガイド』を参照してください。</p> <p>デフォルトでは、Extract は接続先が物理スタンバイ・データベースの場合でも、アーカイブ・ログ専用モードを使用しません。</p>                                                                                                                                                                                                                                                                                                                                                                            |
| ASMBUFSIZE <size>                                                                                         | <p>(Oracle) ASM インスタンスから抽出されたデータを保持するバッファのサイズを制御します。値を大きくすると抽出速度は向上しますが、Extract はより多くのメモリーを消費します。値を小さくすると、メモリーの使用量は減りますが、Extract はキャッシュ・サイズを超えるデータをディスクに保持する必要があるため、I/O は増加します。</p> <p>次に、BUFSIZE の有効範囲とデフォルト・サイズ (バイト) を示します。</p> <p>Oracle:</p> <ul style="list-style-type: none"> <li>◆ 最小: REDO ログの 1 ブロックのサイズ</li> <li>◆ 最大: 28672</li> <li>◆ デフォルト: 28672</li> </ul>                                                                                                                                                                                                                                                                                                               |
| ASMUSER SYS@<ASM_instance>,<br>ASMPASSWORD <password><br>[, ENCRYPTKEY DEFAULT  <br>ENCRYPTKEY <keyname>] | <p>(Oracle) ASM インスタンスへのログインのためのユーザーおよびパスワードを指定します。このユーザーは SYS である必要があります。パスワードに大/小文字区別がある場合は、そのとおりに入力します。</p> <ul style="list-style-type: none"> <li>◆ ENCRYPTKEY DEFAULT は、オプションの引数なしの ENCRYPT PASSWORD コマンドによって、パスワードがデフォルトの Oracle GoldenGate 鍵で暗号化されている場合に必要です。</li> <li>◆ ENCRYPTKEY &lt;keyname&gt; は、ENCRYPT PASSWORD ENCRYPTKEY &lt;keyname&gt; コマンドによってユーザー定義鍵を使用してパスワードが暗号化されているときに必要です。ENCKEYS ファイルに含まれているものと同じ論理鍵名を指定します。</li> </ul> <p><b>注意:</b> このパラメータは、標準の USERID パラメータのかわりには使用できません。ASM 環境では、両方のパラメータが必要です。ASMUSER は、DBLOGREADER オプションを使用している場合は不要です。</p> <p>追加の ASM 要件、および Oracle GoldenGate のセキュリティ機能の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p> |
| BUFSIZE <size>                                                                                            | <p>(DB2 LUW、Oracle) トランザクション・ログの各読み取り結果を保持する内部バッファのサイズ (バイト) を制御します。API によってこのバッファに収まるだけレコードが返されるため、このオプションは注意して使用してください。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

| オプション                                               | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                     | <p>次に、BUFSIZE の有効範囲とデフォルト・サイズ（バイト）を示します。</p> <p><i>Oracle:</i></p> <p>最小 : 8,192</p> <p>最大 : 10,000,000</p> <p>デフォルト : 1,024,000</p> <p><i>DB2 LUW:</i></p> <p>最小 : 40,960</p> <p>最大 : 33,554,432</p> <p>デフォルト : 131,072</p> <p>上記の値は、4096 ページ・サイズの倍数である必要があります。指定された値がこの要件を満たさない場合、Extract は値を切り捨てて倍数にします。</p> <p>システム管理者に依頼して、新しいバッファ・サイズをサポートするために十分な ECSA 領域を確保してください。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| COMPLETEARCHIVEDLOGONLY   NOCOMPLETEARCHIVEDLOGONLY | <p>(Oracle) Extract のデフォルトのアーカイブ・ログ処理をオーバーライドします。</p> <ul style="list-style-type: none"> <li>◆ <b>通常モードのデフォルト :</b> Extract は、アーカイブ・ログが使用可能になると、ディスクに書き込まれるまで待機せずに、即座にアーカイブ・ログの処理を開始します。</li> </ul> <p><b>通常モードのオーバーライド :</b> COMPLETEARCHIVEDLOGONLY を使用して、読取りを開始する前に、アーカイブ・ログが完全にディスクに書き込まれるまで Extract を待機させます。</p> <ul style="list-style-type: none"> <li>◆ <b>アーカイブ・ログ専用 (ALO) モードのデフォルト :</b> アーカイブ・ログが完全にディスクに書き込まれるまで Extract を待機させます。</li> </ul> <p><b>ALO モードのオーバーライド :</b> NOCOMPLETEARCHIVEDLOGONLY を使用して、アーカイブ・ログが使用可能になり次第、Extract に即座にアーカイブ・ログの処理を開始させます。</p> <p>このパラメータは、本番（ソース）アーカイブ・ログをセカンダリ・データベースにコピーし、データ・ソースとして使用するときに応用します。一部の Oracle プログラムは、最初のバイトから最後のバイトまで順序どおりにアーカイブ・ログを構築せずに、たとえばまず最初の 500MB、次に最後の 500MB、そして最後に中間の 1000MB をコピーする場合があります。Extract は、最初のバイトから読取りを開始した場合、バイト・シーケンスが中断する部分に到達したときに異常終了します。ファイル全体が書き込まれるまで待機させることにより、この問題を防止できます。</p> |

| オプション                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMPLETEARCHIVEDLOGTIMEOUT <seconds> | <p>(Oracle) COMPLETEARCHIVEDLOGONLY モードの Extract が、REDO ログが完全にディスクに書き込まれたかどうかを検証できないときに、検証の再試行を待機する秒数を制御します。このオプションは、TRANLOGOPTIONS の COMPLETEARCHIVEDLOGONLY オプションとともに使用します。デフォルトでは、このオプションは無効化されており、Extract はファイルがディスクに書き込まれたかどうかを検証できない場合、10 秒後に異常終了します。検証では、最後のブロックからブロック・ヘッダーを読み取り、予測される順序番号と照合することにより、最後のブロックが書き込まれたかどうかを判断します。&lt;seconds&gt; には、0 よりも大きい任意の値を使用します。</p>                                                                                                                                                                               |
| DBLOGREADER                          | <p>(Oracle) Extract に Oracle 10.2.0.5 以降で使用可能な新しい API を使用させます。この API は、データベース・サーバーを使用して REDO ログおよびアーカイブ・ログにアクセスするため、ASM に常駐する必要がありません。使用するデータベースは、この API モジュールを含むライブラリを含み、かつ実行中である必要があります。</p> <p>DBLOGREADER を使用する場合、Extract は最大 4MB の読取りバッファを使用できます。デフォルトの OCI バッファを使用するときの最大バッファ・サイズは、1024000 バイトです。REDO 率が高い場合、大きなバッファを使用することによって Extract のパフォーマンスが改善することがあります。バッファ・サイズを指定するには、DBLOGREADERBUFSIZE オプションを使用します。</p> <p>DBLOGREADER を使用するときは、TRANLOGOPTIONS の ASMUSER および ASMPASSWORD オプションを使用しないでください。この API は、USERID パラメータで指定されているユーザーおよびパスワードを使用します。</p> |
| DBLOGREADERBUFSIZE <buffer size>     | <p>(Oracle) DBLOGREADER を指定しているときに、Extract が Oracle ASM からの取得に使用する読取りバッファ・サイズを指定します。4MB までのバッファを指定します。デフォルトは 1024000 バイトです。1024000 よりも大きいバッファ・サイズを使用するときは、BUFSIZE パラメータを、DBLOGREADERBUFSIZE で指定する値以上に変更する必要があります。</p>                                                                                                                                                                                                                                                                                                                                         |
| EXCLUDETRANS <trans name>            | <p>(Sybase、SQL Server) Replicat データベース・ユーザーまたは他の任意のユーザーのトランザクション名を指定し、Extract によるこれらのトランザクションの取得を防ぎます。双方向処理で、データベース間でのデータのループを防止するために使用します。</p> <p>Replicat が使用するデフォルトのトランザクション名は ggs_repl ですが、EXCLUDETRANS では任意のトランザクションを指定できます。双方向同期の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>                                                                                                                                                                                                                                                          |



| オプション                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>EXCLUDEUSER &lt;user name&gt;</p> | <p>(DB2 LUW、DB2 z/OS、Ingres、Oracle 10g 以上、Sybase)<br/>GETREPLICATES または IGNOREREPLICATES パラメータのルール適用対象となるトランザクションを特定するフィルタとして使用する、Replicat データベース・ユーザーまたは他の任意のユーザーの名前を指定します。</p> <p>通常このオプションは、双方向またはカスケード処理構成で、除外または取得する Replicat トランザクションを特定するために使用します。ただしこのパラメータは、特定のビジネス・アプリケーションなど、その他のユーザーのトランザクションの特定にも使用できます。</p> <p><b>z/OS 上の DB2 での考慮事項：</b></p> <p>z/OS 上の DB2 では、このユーザーは常にトランザクションのプライマリ許可 ID で、通常はログオンした元の RACF ユーザーの ID ですが、トランザクション・プロセッサまたは DB2 イグジットによって変更されているときは、別の許可 ID の場合もあります。</p> <p><b>Oracle での考慮事項：</b></p> <p>Oracle データベースでは、複数の EXCLUDEUSER 文を使用できます。指定されたすべてのユーザーは、GETREPLICATES または IGNOREREPLICATES ルールの対象になるという意味で、Replicat ユーザーと同一とみなされます。</p> <p>1 つの文でユーザーの範囲を指定するために、ワイルドカードは使用できません。</p> <p>EXCLUDEUSER と EXCLUDEUSERID は、同一のパラメータ・ファイルで使用できます。</p> <p>ユーザー名は有効である必要があります。Oracle GoldenGate は、データベースに問合せて関連するユーザー ID を取得し、この数値識別子をユーザー名にマップします。そのため、名前解決がデフォルトの DYNAMICRESOLUTION (178 ページ) に設定されている間に、指定されたユーザーが削除および再作成された場合、EXCLUDEUSER は有効のままです。名前解決が NODYNAMICRESOLUTION に設定されているときに同一の DDL が実行された場合、EXCLUDEUSER は無効になるため、EXCLUDEUSER を有効にするために Extract を停止および再起動する必要があります。</p> <p>Oracle 9i 以前では、Replicat トランザクションを識別するために、トレース表を使用する必要があります。359 ページを参照してください。</p> |

| オプション                                                      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EXCLUDEUSERID <Oracle uid>                                 | <p>(Oracle 10g 以上) GETREPLICATES または IGNOREREPLICATES パラメータのルール適用対象となるトランザクションを特定するフィルタとして使用する、Replicat データベース・ユーザーまたは他の任意のユーザーの Oracle ユーザー ID (UID) を指定します。</p> <p>使用方法は EXCLUDEUSER と同じです。</p> <p>ユーザー ID は正の整数で、最大値は 2147483638 です。ユーザー ID は、いくつかのシステム・ビューに問合せで取得できます。最も単純なビューは ALL_USERS です。Oracle GoldenGate は、ユーザー ID の検証を行いません。</p> <p>注意：特定のユーザー ID に関連付けられているユーザーが削除および再作成された場合、このユーザーには新しいユーザー ID が割り当てられるため、このユーザーには EXCLUDEUSERID は無効になります。</p> |
| FILTERTABLE <table_name>                                   | <p>(c-tree および SQL/MX) このオプションでは、Replicat が使用するチェックポイント表名を指定します。チェックポイント表上の操作は、ソースへのデータのループバックを防止する手段として、ローカル Extract によって無視されます。チェックポイント表の作成の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>                                                                                                                                                                                                                                                  |
| IGNOREDATA CAPTURECHANGES  <br>NOIGNOREDATA CAPTURECHANGES | <p>(DB2 LUW) IGNOREDATA CAPTURECHANGES は、DATA CAPTURE CHANGES が設定されていない表を無視します。表がワイルドカードで指定されている場合に、変更取得セットが含まれている表の処理を継続するために使用します。スキップされた表に対する警告がエラー・ログに発行されます。デフォルトは NOIGNOREDATA CAPTURECHANGES です。</p>                                                                                                                                                                                                                                                  |
| IGNOREDIRECTLOADINSERTS                                    | <p>(Oracle) Extract にすべての Oracle ダイレクト・ロード INSERT を無視させます。デフォルトの動作（このパラメータなし）では、Oracle ダイレクト・ロード INSERT を取得します。このオプションは、Oracle 10g 以上のログと互換性がある Oracle ログに適用されます。</p>                                                                                                                                                                                                                                                                                        |
| IGNOREMISSINGTABLES                                        | <p>(DB2 LUW の Extract) DML 操作を受信した後に表が削除されたことが原因で行方不明になった表を Extract に無視させます。この状況は、次のときに発生することがあります。</p> <p>DB2 LUW ログ・レコードには表名が含まれていないため、Extract は TABLE および TABLEEXCLUDE 文を処理するために、表領域 ID および表 ID を使用してデータベースから表名前を検索します。処理中のログ・レコードが生成された後に表が削除されると、Extract は表が行方不明のためにこのレコードをスキップします。IGNOREMISSINGTABLES はデフォルトです。</p>                                                                                                                                    |

| オプション                                                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LOGSOURCE <platform>                                                 | <p>(Oracle) REDO ログまたはアーカイブ・ログ、またはその両方が、データベースをホストしているプラットフォーム以外のプラットフォームに保持されているときに、そのオペレーティング・システムを指定します。次の値が有効です。</p> <ul style="list-style-type: none"> <li>◆ AIX</li> <li>◆ HPUX</li> <li>◆ LINUX</li> <li>◆ MVS</li> <li>◆ SOLARIS</li> <li>◆ VMS</li> <li>◆ WINDOWS</li> <li>◆ S390</li> </ul> <p>また、LOGSOURCE を PATHMAP オプションとともに使用し、ログのパスも指定することができます。</p> <p>正しいデータ位置を保持するために、LOGSOURCE プラットフォームと Extract を実行するプラットフォームは、次の要素が同一である必要があります。</p> <ul style="list-style-type: none"> <li>◆ エンディアン・オーダー</li> <li>◆ ビット幅 (32 ビット、64 ビットなど)</li> </ul> <p>次に互換性のあるエンディアン・プラットフォームを示します。</p> <ul style="list-style-type: none"> <li>◆ ビッグ・エンディアン: AIX、HPUX、MVS、SOLARIS、S290</li> <li>◆ リトル・エンディアン: LINUX、VMS、WINDOWS</li> </ul> <p>たとえば、Extract を HPUX で実行している場合、LOGSOURCE を AIX に設定すると有効ですが、LINUX は無効です。</p> <p>LOGSOURCE を使用するときは、TRANLOGOPTIONS 文全体を 1 行にまとめてください。アンパサド (&amp;) 行終了文字を使用して行を複数に分割しないでください。</p> |
| MANAGESECONDARYTRUNCATIONPOINT  <br>NOMANAGESECONDARYTRUNCATIONPOINT | <p>(SQL Server 2005 および Sybase) Oracle GoldenGate が 2 次切捨てポイントを維持するかどうかを指定します。デフォルトは MANAGESECONDARYTRUNCATIONPOINT です。</p> <p><b>SQL Server 2005</b></p> <p>MANAGESECONDARYTRUNCATIONPOINT は、Oracle GoldenGate が SQL Server レプリケーションと同時に実行されないため、Oracle GoldenGate が 2 次切捨てポイントを維持するときに使用します。NOMANAGESECONDARYTRUNCATIONPOINT は、Oracle GoldenGate が SQL Server レプリケーションと同時に実行されるときに使用します。SQL Server レプリケーションに 2 次切捨てポイントを管理させます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| オプション                                   | 説明                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                         | <p><b>Sybase</b></p> <p>MANAGESECONDARYTRUNCATIONPOINT は、Oracle GoldenGate が Sybase Replication Server と同時に実行されないときに使用します。Extract が 2 次切捨てポイントを管理します。</p> <p>NOMANAGESECONDARYTRUNCATIONPOINT は、Sybase トランザクション・ログを切り捨てないときに使用します。Extract は 2 次切捨てポイントを管理しません。このオプションは、Extract がデバッグ目的で以前のログ位置から Sybase トランザクション・ログの再読取りを行う必要があるときに使用できます。</p>                      |
| MAXWARNEOF <seconds>                    | <p>(Oracle) 警告メッセージを生成するまで、Extract が新しいログ・ファイルを待機する秒数を指定します。Extract は、特定の順序番号に対して警告メッセージを 1 つのみ生成します。MAXWARNEOF を指定しない場合、Extract はデフォルトで 1 時間待機します。値 0 を指定すると、Extract が待機した時間にかかわらず、警告メッセージは生成されません。</p>                                                                                                                                                            |
| NODDLCHANGEWARNING                      | <p>(SQL Server) Extract がデータを取得するソース・オブジェクトに DDL 操作が実行されたときに、Extract が警告を記録することを防ぎます。デフォルトでは、問題を修正できるように警告がレポートされます。Oracle GoldenGate は、SQL Server に対して DDL 取得およびレプリケーションをサポートしていないため、ソースおよびターゲットのメタデータは一定のままであるとみなします。DDL 変更によって Extract が異常終了しなくても、このような変更が発生するたびに警告が記録されます。</p> <p>NODDLCHANGEWARNING を使用すると、Oracle GoldenGate ログにこのようなメッセージが蓄積されることを防止できます。</p> |
| NOFLUSH                                 | <p>(DB2 z/OS) ログ・バッファのフラッシュを防止します。</p>                                                                                                                                                                                                                                                                                                                              |
| PATHMAP <NFS mount point><br><log path> | <p>(Oracle) REDO ログまたはアーカイブ・ログ、またはその両方が、データベースをホストしているシステム以外のシステムに保持されているときに、その場所を指定します。NFS マウント・ポイントに続き、Oracle ログ構造のパスを指定します。1 つ以上の PATHMAP 文を使用できます。</p> <p>このシステムがデータベースをホストしているシステムとは異なるプラットフォームの場合は、LOGSOURCE オプションとともに使用できます。</p> <p>PATHMAP を使用するときは、TRANLOGOPTIONS 文全体を 1 行にまとめてください。アンバサド (&amp;) 行終了文字を使用して行を複数に分割しないでください。</p>                           |

| オプション                                                      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PURGEORPHANEDTRANSACTIONS  <br>NOPURGEORPHANEDTRANSACTIONS | (Oracle RAC) ノードに障害が発生し、Extract がロールバックを取得できないときに発生する孤立トランザクションのページを有効化または無効化します。トランザクションは、ページの実行前に、自身の開始時間とノードの開始時間の比較によって孤立しているかどうかを検証され、ノードよりも開始時間が早いトランザクションはページされます。デフォルトは PURGEORPHANEDTRANSACTIONS です。                                                                                                                                                                                                            |
| QUERYRETRYCOUNT <number of retries>                        | <p>(SQL Server の Extract) タイムアウト後に表のメタデータを取得するために問合せを再試行する回数を指定します。デフォルトは、30 秒の待機後に 1 回の再試行で、この再試行が失敗するとプロセスは異常終了します。</p> <p>QUERYRETRYCOUNT では、指定される入力値に従って 30 秒間隔で複数回再試行するように指定できます。すべての再試行が失敗すると、Extract は通常の接続エラー・メッセージとともに異常終了します。表を作成した長時間におよぶトランザクションでは、タイムアウトが発生することがあります。システム表はロックされ、Extract の問合せは完了できなくなります。</p> <p>例：<br/>TRANLOGOPTIONS QUERYRETRYCOUNT 4</p> <p>この例では、Extract は 30 秒間隔で 4 回問い合わせを試行します。</p> |
| READBUFFER <byte length>                                   | <p>(c-tree) c-tree によって保持される、レプリケーション・ログ・データを格納するバッファの長さ (バイト) を指定します。このバッファは、デフォルトでパフォーマンス機能として使用されます。Extract がこのバッファ内のデータをリクエストすると、通常はパフォーマンスが大幅に向上します。</p> <p>デフォルトのバッファ・サイズは、最小の許可サイズの 8KB です。最大サイズは 64KB です。値ゼロは、レプリケーション・ログ読み取りバッファを無効にします。</p>                                                                                                                                                                    |
| READTIMEOUT <milliseconds>                                 | <p>(c-tree) 変更レコードが現在の位置に存在しない場合に、c-tree レプリケーション・エンジンが (READBUFFER によって制御される) トランザクション・バッファに変更レコードを送信するために待機する時間を決定するタイムアウト (ミリ秒) を指定します。このタイムアウト間隔が経過する前に変更レコードが到着しなかった場合、データベースによってエラーが返されます。このパラメータで指定するタイムアウトによって、次のレコードが存在しない場合に Extract が待機する必要がある時間が決まります。</p> <p>デフォルトは 1000 ミリ秒です。最小値は 1 ミリ秒です。最大値は 30000 ミリ秒です。</p>                                                                                           |

| オプション                                                                | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| REQUIRELONGDATA_CAPTURECHANGES  <br>NOREQUIRELONGDATA_CAPTURECHANGES | <p>(DB2 LUW) 次の状況での Extract のレスポンスを制御します。</p> <ul style="list-style-type: none"> <li>◆ DATA_CAPTURE が NONE、または INCLUDE LONGVAR COLUMNS なしで CHANGES に設定されている<br/>および</li> <li>◆ パラメータ・ファイルに、一部またはすべての列値のビフォア・イメージを必要とする Oracle GoldenGate パラメータ (GETBEFOREUPDATES、NOCOMPRESSUPDATES、および NOCOMPRESSDELETES) が含まれている</li> </ul> <p>この 2 つの DATA_CAPTURE 設定では、LONGVAR 列のビフォア値が記録されません。Extract がこれらの列を使用できない場合、ターゲット・データの整合性に影響が及ぶことがあります。</p> <ul style="list-style-type: none"> <li>◆ REQUIRELONGDATA_CAPTURECHANGES<br/>Extract はエラーとともに異常終了します。</li> <li>◆ NOREQUIRELONGDATA_CAPTURECHANGES<br/>Extract は警告を出しますが、データ・レコードの処理を継続します。</li> </ul> |
| TRANSCLEANUPFREQUENCY <minutes>                                      | <p>(Oracle RAC) Oracle GoldenGate が孤立トランザクションのスキャン、再スキャン、および削除の一連の動作を実行する間隔 (分) を指定します。最初のスキャンでは、孤立していると思われるトランザクションをマーク付けします。2 回目のスキャンで孤立していることを確認し、次にこれらのトランザクションを削除します。有効な値は 1 ～ 43200 分です。デフォルトは 10 分です。</p>                                                                                                                                                                                                                                                                                                                                                                                                                               |
| UNPRIVILEGED                                                         | <p>(z/OS 上の DB2) Extract の拡張属性を、APF (許可プログラム機能) に許可されないに変更します。これは、システム・コマンド extattr -a extract を発行することと同じです。APF 許可の拡張属性の削除は、特別の権限を必要とせずに実行できることがあります。このオプションにより、Extract に、Sysplex データ共有環境をサポートしていない、バージョン 8.0 以前の API を使用するように指示します。</p>                                                                                                                                                                                                                                                                                                                                                                                                       |

| オプション                    | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VAMCOMPATIBILITY {1   2} | <p>(c-tree、MySQL、SQL M/X、SQL Server、Sybase、Teradata)</p> <p><i>Vendor Access Module</i> (VAM) と呼ばれる Oracle GoldenGate API 間で、各 VAM 実装の必要性に応じて、異なるメタデータ構造が渡されることを許可します。</p> <ul style="list-style-type: none"> <li>◆ 値 1 は、元の VAM API メタデータ構造の使用を指定します。このメタデータ構造は、Oracle GoldenGate for Teradata のために実装されています。Teradata モジュールは、共有ライブラリとして構成されており、新しいバージョンのようにプログラムによって設定されないため、TRANLOGOPTIONS VAMCOMPATIBILITY の使用が必要な唯一の VAM モジュールです。TRANLOGOPTIONS VAMCOMPATIBILITY は、古い TAM モジュールとともに新しい Oracle GoldenGate for Teradata の Extract を使用している場合に、古いモジュールとの下位互換性をサポートするために使用します。VAM 互換性を VAMInitialize で設定している場合は、TRANLOGOPTIONS で設定する必要はありません。このパラメータは、Extract と TAM モジュールが同じバージョンの場合には必要ありません。</li> <li>◆ 値 2 は、元の構造をベースにフィールドが追加されている、拡張 VAM API メタデータ構造の使用を指定します。この構造は、現在 Oracle GoldenGate for Sybase 製品で使用されています。この値は VAM 内でプログラムによって設定されるため、TRANLOGOPTIONS VAMCOMPATIBILITY を使用する必要はありません。</li> </ul> |

- 例 1** 次の例では、アーカイブ・ログの場所を指定します。
- ```
TRANLOGOPTIONS ALTARCHIVELOGDEST /fs1/oradata/archive/log2
```
- 例 2** 次の Oracle の例では、2 ユーザーを（1 つは名前、1 つはユーザー ID で）フィルタし、GETREPLICATES または IGNOREREPLICATES ルールに従ってこれらのユーザーのトランザクションを処理し、新しいトランザクション・バッファ・サイズを設定します。
- ```
TRANLOGOPTIONS EXCLUDEUSER ggsrep, EXCLUDEUSERID 90, BUFSIZE 100000
```
- 例 3** 次の例では、SQL Server または Sybase 環境で Replicat トランザクション名を除外します。
- ```
TRANLOGOPTIONS EXCLUDETRANS "ggs_repl"
```
- 例 4** 次に、データベースをホストするプラットフォームと別のプラットフォーム上にあるトランザクション・ログを処理する方法を示します。注意：この文は、このドキュメントのスペースの制約によって、複数の行にまたがっています。
- ```
TRANLOGOPTIONS, LOGSOURCE VMS, PATHMAP DKA200:[RDBMS.ORACLE.ORA9201I.64.ADMIN.GGS.ARCH]
/net/deltan/uservol1/RDBMS.DIR/ORACLE.DIR/ORA9201I.DIR/
64.DIR/admin.DIR/ggs.DIR/ARCH.dir PATHMAP
DKA200:[RDBMS.ORACLE.ORA9201I.64.ORADATA.GGS]
/net/deltan/uservol1/rdbms.dir/oracle.dir/ora9201I.DIR/
64.dir/oradata.dir/ggs.dir
```
- 例 5** 次に、ALTONLINELOGS のいくつかの使用例を示します。空白を含むパスは、二重引用符で囲まれています。
- ```
TRANLOGOPTIONS ALTONLINELOGS ("third one/log1.txt")
TRANLOGOPTIONS ALTONLINELOGS( "first one/log1.txt", second_one/log2.txt )
TRANLOGOPTIONS ALTONLINELOGS ( sixth_one/log1.txt, seventh_one/log2.txt, &
"eighth one/log2.txt")
```

TRANSACTIONTIMEOUT

適用対象 Replicat

未コミットの Replicat ターゲット・トランザクションが、ターゲット・データベース上でロックを保持しないようにし、不要なリソースの消費を防ぐには、TRANSACTIONTIMEOUT パラメータを使用します。このパラメータの値は、Replicat が既存のアプリケーションのタイムアウト内、および個のターゲットの他のデータベース要件内で機能するように変更できます。

TRANSACTIONTIMEOUT では、トランザクション内の最後のソース・トランザクションに関するトランザクション終了時レコードが未受信の場合に、Replicat でターゲット・トランザクションが保持される時間が制限されます。デフォルトでは、Replicat は複数のソース・トランザクションを 1 つのターゲット・トランザクションにグループ化するので、パフォーマンスが向上しますが、部分的なソース・トランザクションはコミットせず、最後のレコードまで無制限に待機します。Replicat パラメータの GROUPTRANSOPS により、グループ化されたターゲット・トランザクションの最小サイズが制御されます。

次のイベントは、TRANSACTIONTIMEOUT をトリガーするのに十分なほど長く継続することがあります。

- ネットワークの問題により、証跡データがターゲット・システムに送信されない。
- システム上のディスク領域が不足し、証跡データへの書き込みができない。
- コレクタが異常終了した（稀なイベント）。
- Extract が異常終了したか、トランザクションに関するレコードの書き込み中に停止した。
- Extract データ・ポンプが異常終了したか、または停止した。
- 停電またはシステム・クラッシュなど、ソース・システムの障害が発生した。

TRANSACTIONTIMEOUT の動作

通常の操作では、Replicat は、トランザクションが異常終了し、再試行が必要になるときに備えて、現在のターゲット・トランザクション内の最初のソース・トランザクションの開始位置を記憶しています。TRANSACTIONTIMEOUT を有効にすると、Replicat は、現在のソース・トランザクションの最初のレコードの位置も記憶し、TRANSACTIONTIMEOUT がトリガーされた場合に、この位置を論理的な終了（EOL）位置として使用します。

TRANSACTIONTIMEOUT は、トリガーされると次のことを行います。

1. 現在のターゲット・トランザクションを中止します。
2. 中止されたターゲット・トランザクション内の最初のソース・トランザクションの開始位置に移動します。
3. 論理終了位置（最後の完了していないソース・トランザクションの開始位置）まで、すべてのトレイル・レコードを処理します。
4. 論理 EOF 位置でトランザクションをコミットします。
5. 新しいトレイル・データを待機してから、次のトレイル・レコードを処理します。

TRANSACTIONTIMEOUT は、データの到着を遅らせ、TRANSACTIONTIMEOUT をトリガーする問題の性質に応じて、同一のソース・トランザクションに対して何度もトリガーできます。

TRANSACTIONTIMEOUT 状況の発生の確認

TRANSACTIONTIMEOUT が有効な場合に、Replicat がソース・トランザクションの残りの部分を待機しているかどうかを確認するには、SEND REPLICAT コマンドを STATUS オプションとともに発行します。次のステータスは、Replicat が待機していることを示します。

```
Performing transaction timeout recovery
Waiting for data at logical EOF after transaction timeout recovery
```

デフォルト 無効

構文 TRANSACTIONTIMEOUT <n> <units>

オプション	説明
<n>	待機する間隔を指定する整数。有効な値は 1 秒～1 週間（7 日間）です。この値は、プライマリ Extract および関連するすべてのデータ・ポンプ両方で設定されている EOFDELAY パラメータの値より大きくする必要があります。
<units>	次のいずれか 1 つを使用します: S、SEC、SECS、SECOND、SECONDS、MIN、MINS、MINUTE、MINUTES、HOUR、HOURS、DAY、DAYS。

例 TRANSACTIONTIMEOUT 5 S

TRANSMEMORY

適用対象 z/OS 上の DB2 および NonStop SQL/MX の場合の Extract

TRANSMEMORY パラメータは、コミットされていないトランザクション・データのキャッシュに使用可能なメモリの量および一時ディスク領域の制御に使用します。Oracle GoldenGate では、コミットされているトランザクションのみがターゲット・データベースに送信されるため、コミットまたはロールバック・インジケータのいずれかが受信されるまで、ソース・システム上でトランザクション・データを格納するのに十分なシステム・メモリが必要になります。

このパラメータは、z/OS 上の DB2 データベースおよび NonStop SQL/M データベース用です。その他のデータベースには、CACHEMGR パラメータを使用します。

TRANSMEMORY を使用したメモリ管理について

TRANSMEMORY は、Oracle GoldenGate トランザクションのキャッシュ・サイズの調整、およびキャッシュのサイズを超えるデータを格納するためのディスク上における一時的な場所の定義を可能にします。キャッシュ・サイズの合計、トランザクションごとのメモリ・サイズ、初期と増分メモリの割当て、およびディスク格納領域の定義のための各オプションがあります。

トランザクションが RAM により指定されるメモリ・プールに追加され、それぞれ、TRANSRAM に達するとディスクにフラッシュされます。メモリの初期量は、INITTRANSRAM に基づいて各トランザクションに割り当てられ、必要に応じて RAMINCREMENT で指定されている量ずつ、最大 TRANSRAM で設定されている量まで増加します。このため、TRANSRAM の値は、(INITTRANSRAM + RAMINCREMENT) の合計で割り切れる数値である必要があります。

現在の TRANSMEMORY 設定を表示するには、GGSCI の VIEW REPORT <group> コマンドを使用します。

z/OS の場合の注意事項

z/OS システムでは、RAM オプションにより、すべてのキャッシュ済仮想メモリの割当ての総量のみでなく、起動中に割り当てられるヒープ・メモリのサイズも制御されます。デフォルト値が大きいと、仮想メモリ・プール内のフラグメンテーションは回避されますが、一部のインストールでは、特にアプリケーションで主に小規模のトランザクションが生成される場合に仮想メモリが不必要に消費されることがあります。大量のヒープ・メモ리를割り当てることで、z/OS で割当てが完了するまで起動時の Extract プロセスが応答なし状態になることもあります。

Extract のパフォーマンスに影響せずに十分なトランザクション・アクティビティを保持するのに必要な量に、z/OS の RAM を設定します。この値が小さすぎると、Extract プロセスによってトランザクション・データがディスク内に書き込まれ、Extract プロセス速度の低下およびディスク領域の消費につながる場合があります。最適な値を決定するには、何度かテストを行う必要があります。

デフォルト なし

構文
TRANSMEMORY
[RAM <size>]
[TRANSRAM <size>]
[TRANSALLSOURCES <size>]
[INITTRANSRAM <size>]
[RAMINCREMENT <size>]
[DIRECTORY (<directory name>, <max dir size>, <max file size>)]

オプション	説明
RAM <size>	すべてのキャッシュ済トランザクションに使用されるメモリの総量を指定します。z/OS では、トランザクションごとに割り当てられる初期メモリの量も指定されます。 デフォルト値は 200MB です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。 GB MB KB G M K gb mb kb g m k
TRANSRAM <size>	単一のキャッシュ済トランザクションに使用されるメモリの総量を指定します。デフォルト値は 50MB です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。 GB MB KB G M K gb mb kb g m k 最適な結果を得るには、TRANSRAM を INITTRANSRAM と RAMINCREMENT の両方で割り切れる値に設定する必要があります。
TRANSALLSOURCES <size>	単一のキャッシュ済トランザクションに使用されるメモリの総量およびディスク領域を指定します。デフォルト値は、使用可能なメモリ（メモリおよびディスク）の総量の 50% です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。 GB MB KB G M K gb mb kb g m k
INITTRANSRAM <size>	(NonStop システムのみ) トランザクションに割り当てられる初期メモリの量を指定します。デフォルト値は 50KB です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。 GB MB KB G M K gb mb kb g m k

オプション	説明
RAMINCREMENT <size>	<p>トランザクションに追加のメモリーが必要な場合に、増分するメモリーの量を指定します。デフォルト値は 50KB です。この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m k</p>
DIRECTORY (<directory name>, <max dir size>, <max file size>)	<p>トランザクション・データ用のディスク領域が TRANSRAM で指定した最大値を超える場合に、一時的なディスク領域を指定します。複数の DIRECTORY を指定できます。</p> <ul style="list-style-type: none"> ◆ <directory name> は、ディレクトリの完全修飾名です。デフォルト値は、Oracle GoldenGate ディレクトリの dirtmp サブディレクトリです。 ◆ <max dir size> は、このディレクトリ内のすべてのファイルの最大サイズです。デフォルト値は 2GB です。指定した領域が使用できない場合は、使用可能なディスク領域の 75% が使用されます。 ◆ <max file size> は、このディレクトリ内の各ファイルの最大サイズです。デフォルト値は 200MB です。 <p>この値は、次のようにギガバイト、メガバイトまたはキロバイトで指定します。</p> <p>GB MB KB G M K gb mb kb g m k</p> <p>ディレクトリ・サイズおよびファイル・サイズは、RAM で指定したメモリーのサイズより大きな値である必要があります。</p> <p>ファイル名には、次の形式を使用します。</p> <p><group>_trans_00001.mem または <PID>_trans_00001.mem</p> <p>グループ名はオンライン・プロセスに使用されます。システム・プロセス ID 番号 (PID) は、SPECIALRUN パラメータで指定されるバッチ実行数に使用されます。</p> <p>スレッド化された Extract の場合の形式は、データベースごとに以下のような形式になります。</p> <p><group>_<thread #>_00001.mem</p>

例 1 次の例では、**INITTRANSRAM** で初期割当て値を指定した後、**RAMINCREMENT** で 9 回の増分を許可した場合に、データのディスクへのフラッシュ前に各トランザクションを 10 回増分できるようになります。

```
TRANSMEMORY DIRECTORY(c:\test\dirtmp, 3000000000,
3000000000), RAM 8000K, TRANSRAM 1000K, INITTRANSRAM 100K,
RAMINCREMENT 100K
```

例 2 次の例は、前述の例と同じですが、2 番目のディレクトリを追加しています。

```
TRANSMEMORY DIRECTORY(c:\test\dirtmp, 3000000000,
3000000000), DIRECTORY(c:\test\dirtmp2, 1000000000,
50000000), RAM 8000K, TRANSRAM 1000K, INITTRANSRAM 100K,
RAMINCREMENT 100K
```

注意 これらの例では、スペースの制約のため、パラメータ指定文が複数行にわたって記述されています。実際のパラメータ・ファイルでは、複数行にわたってパラメータを指定する場合は、各行の末尾にアンパサンド文字（&）を含める必要があります。

TRIMSPACES | NOTRIMSPACES

適用対象 Extract および Replicat

TRIMSPACES および NOTRIMSPACES パラメータでは、ターゲット表に送信するときに末尾の空白を切り捨てるかどうかを制御します。これらのパラメータは、CHAR から VARCHAR への列マッピングにのみ影響します。

TRIMSPACES および NOTRIMSPACES は、パラメータ・ファイル内の異なる TABLE または MAP 文の個別スイッチとして使用できます。また、個別の TABLE または MAP 文内で使用して、特定の MAP または TABLE 文のグローバル設定よりも優先させることができます。

デフォルト TRIMSPACES

構文 TRIMSPACES | NOTRIMSPACES

例 次の例では、最後の MAP 文を除き、デフォルトの末尾空白切捨てを行います。

```
MAP fin.src1, TARGET fin.tgt1;
MAP fin.src2, TARGET fin.tgt2;
MAP fin.src3, TARGET fin.tgt3;
NOTRIMSPACES
MAP fin.src4, TARGET fin.tgt4;
```

UNLOCKEDTRAILFILES

適用対象 GLOBALS

UNLOCKEDTRAILFILES パラメータでは、ターゲット・システムに影響を与えるネットワーク障害が発生している場合に、Collector によってロックされているトレイル・ファイルの問題を回避します。Oracle GoldenGate リリース 10.4 以上では、Collector はトレイルへのデータの書き込み中に、同一のトレイルへの書き込みを試みる他のプロセスによってトレイルが破損されないように、トレイル・ファイルをロックします。UNLOCKEDTRAILFILES では、Collector によるトレイル・ファイルのロックを防止します。

このパラメータは、次のように使用できます。

- ソース・システム上で GLOBALS パラメータ・ファイルで使用する場合、UNLOCKEDTRAILFILES はこのシステム上のすべての Extract プロセスによって生成されるすべてのトレイルに影響します。ターゲット上の関連する Collector プロセスは、これらのトレイル・ファイルをロックしません。
- ターゲット・システム上で GLOBALS パラメータ・ファイルで使用する場合、UNLOCKEDTRAILFILES はローカル・システム上のすべての Collector によって書き込まれるすべてのトレイルに影響します。ローカル・トレイル・ファイルは、このシステム上のどの Collector にもロックされません。
- 特定の Extract プロセスのトレイルのみに適用するには、GLOBALS パラメータとして UNLOCKEDTRAILFILES を使用するかわりに、RMTHOST の PARAMS-UL オプションを使用します。関連する Collector プロセス（ターゲットが複数の場合は複数の Collector プロセス）は、これらのトレイル・ファイルをロックしません。

デフォルト 無効

構文 UNLOCKEDTRAILFILES

UPDATEDELETES | NOUPDATEDELETES

適用対象	Replicat
	UPDATEDELETES パラメータでは、パラメータ・ファイル内でこのパラメータ以降のすべての MAP 文の削除操作を更新操作に変換します。UPDATEDELETES を無効にするには、NOUPDATEDELETES を使用します。
デフォルト	NOUPDATEDELETES
構文	UPDATEDELETES NOUPDATEDELETES

UPREPORT

適用対象	Manager
	UPREPORTMINUTES または UPREPORHOURS パラメータでは、Manager が実行中の Extract および Replicat プロセスをレポートする間隔を指定します。これらのいずれかのプロセスが開始または停止するたびに、イベントが生成されます。エラー・ログが非常に大きい場合には、ログ内のこうしたメッセージは見逃されてしまいがちです。UPREPORTMINUTES および UPREPORHOURS は、このようなプロセス・ステータスが見逃されないように、定期的にレポートします。
	停止されたプロセスをレポートするには、DOWNREPORT パラメータを使用します。詳細は、176 ページを参照してください。
デフォルト	実行中のプロセスをレポートしない。
構文	UPREPORTMINUTES <minutes> UPREPORHOURS <hours>

引数	説明
<minutes>	実行中のプロセスをレポートする間隔（分）。
<hours>	実行中のプロセスをレポートする間隔（時間）。

例	次の例では、30 分間隔でレポートが生成されます。 UPREPORTMINUTES 30
----------	---

USEDATEPREFIX

適用対象	Replicat
	USEDATEPREFIX パラメータは、Teradata-ODBC ドライバの 3.02.0.02 より前のバージョンを使用する Teradata データベースで使用します。このパラメータは、DATE データ型のデータ値に DATE リテラル接頭辞を追加します。
	USEDATEPREFIX を使用するときは、USETIMEPREFIX および USETIMESTAMPPREFIX パラメータを使用してください。これらのパラメータを使用するときは、NODYNSQL パラメータ（178 ページを参照してください）を指定する必要があります。
	Teradata-ODBC ドライバ 3.02.0.02 以上では、このパラメータまたは NODYNSQL を使用する必要はありません。
デフォルト	無効

構文 USEDATEPREFIX

USERID

適用対象 Manager、Extract、Replicat、DEFGEN

USERID パラメータでは、データベースにログインするときに使用する Oracle GoldenGate プロセスの認証タイプの指定、およびパスワード暗号化情報の指定を行います。Oracle GoldenGate の暗号化の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

注意 パスワード暗号化は、SQL/MX に対してはサポートされていません。

USERID は、パラメータ・ファイルのどの TABLE エントリよりも先に指定します。

USERID を使用するケースと使用方法

USERID は常に必要なわけではなく、PASSWORD は USERID が必要なときに常に必要なわけではありません。データベース認証の構成方法に応じては、USERID の使用のみで十分な場合、SOURCEDB または TARGETDB パラメータのみでも十分な場合があります。

SOURCEDB および TARGETDB を参照してください。

データベース・タイプ別の USERID の要件

注意 USERID で指定するユーザーに必要な権限の詳細は、お使いのデータベースの Oracle GoldenGate インストール・ガイドを参照してください。

c-tree

USERID は、PASSWORD とともに、c-tree データベースに接続するすべての Oracle GoldenGate プロセスで使用します。また、SOURCEDB または TARGETDB パラメータを USERID とともに使用して、c-tree サーバーの別名を指定します。

DB2 for LUW

USERID は、PASSWORD とともに、データベース認証を使用して DB2 LUW データベースに接続するすべての Oracle GoldenGate プロセスで使用します。データベースがオペレーティングシステム・レベルでの認証を許可するように構成されている場合は、USERID および PASSWORD を省略できます (SOURCEDB または TARGETDB のみを使用できます)。この場合は、オペレーティング・システムのユーザーが『Oracle GoldenGate DB2 LUW インストレーションおよびセットアップ・ガイド』で説明されている適切な権限を持っている必要があります。

DB2 for z/OS データベース

Oracle GoldenGate プロセスに割り当てられているユーザーが、プロセスを正常に機能させるために必要な DB2 権限を持っていない場合に、USERID を PASSWORD とともに使用します。

MySQL

USERID は、PASSWORD とともに、MySQL データベースに接続するすべての Oracle GoldenGate プロセスで使用します。

Oracle

USERID は、Oracle データベースに接続する Oracle GoldenGate プロセスで使用します。

- オペレーティング・システムのログインを使用するには、USERID を / 引数とともに使用します。
- データベース・ユーザー名およびパスワードを使用するには、USERID を PASSWORD とともに使用します。
- オプションで、ログインするユーザーを sysdba として指定できます。

Ingres Database

- オペレーティング・システムのログインを使用する場合、USERID は必要ありません。SOURCEDB または TARGETDB パラメータのみを USERID とともに使用して、ODBC データ・ソースを指定します。
- データベース・ユーザー名およびパスワードを使用するには、USERID を PASSWORD とともに使用します。また、SOURCEDB または TARGETDB パラメータを USERID とともに使用して、ODBC データ・ソースを指定します。

SQL/MX

- ソース SQL/MX データベースに接続する Oracle GoldenGate プロセスでは、USERID を PASSWORD なしで使用して、デフォルト・スキーマを指定します。また、SOURCEDB パラメータを使用して、カタログ名を指定します。
- ターゲット SQL/MX データベースに接続する Oracle GoldenGate プロセスでは、USERID を PASSWORD とともに使用します。また、TARGETDB パラメータを使用して、ターゲット ODBC データ・ソースを指定します。

SQL Server

Oracle GoldenGate プロセスによって使用される ODBC データソース接続がデータベース認証を提供するように構成されている場合は、USERID を PASSWORD とともに使用します。USERID には、このプロセス、またはシステム管理者かサーバー管理者固有サーバー・ロールのアカウントの任意のメンバーに割り当てられている特定のログインを使用できます。

- ソース SQL Server システムでは、SOURCEDB パラメータも使用してソース ODBC データ・ソースを指定します。
- ターゲット SQL Server システムでは、TARGETDB パラメータも使用してターゲット ODBC データ・ソースを指定します。

Sybase

USERID および PASSWORD は、Sybase データベースに接続するすべての Oracle GoldenGate プロセスで使用します。

Teradata

USERID と PASSWORD は、Teradata データベースに接続するすべての Oracle GoldenGate プロセスで使用します。

- ソース Teradata システムでは、SOURCEDB パラメータも使用してソース ODBC データ・ソースを指定します。
- ターゲット Teradata システムでは、TARGETDB パラメータも使用してターゲット ODBC データ・ソースを指定します。

TimesTen

Replicat にによって使用される ODBC データソース接続がデータベース認証を提供するように構成されている場合は、USERID を PASSWORD とともに使用します。また、TARGETDB パラメータを使用して、ターゲット ODBC データ・ソースを指定します。

デフォルト なし

構文 USERID {/ | <user id>[, PASSWORD <password>]}
[ENCRYPTKEY DEFAULT | ENCRYPTKEY <keyname>] [SYSDBA]

引数	説明
/	<p>(Oracle) データベース・ユーザー・ログインではなく、Oracle 用のオペレーティングシステム・ログインを使用するように Oracle GoldenGate に指示します。この引数は、データベースによってオペレーティングシステム・レベルでの認証が許可されている場合にのみ使用します。データベースレベルの認証をバイパスすることにより、アプリケーションのパスワードが頻繁に変更される場合に、Oracle GoldenGate パラメータ・ファイルを更新する必要がなくなります。</p> <p>このオプションを使用するには、Oracle OS_AUTHENT_PREFIX 初期化パラメータとの関連で、正しいユーザー名がデータベースに存在している必要があります。OS_AUTHENT_PREFIX で指定されている値は、ユーザーのオペレーティング・システム・アカウント名の先頭に追加され、データベース名と比較されます。この 2 つの名前は一致する必要があります。</p> <p>OS_AUTHENT_PREFIX が " " (NULL 文字列) に設定されている場合は、ユーザー名を "identified externally" として作成する必要があります。</p> <p>たとえば、OS ユーザー名が "ogg" の場合は、次のようにしてデータベース・ユーザーを作成します。</p> <pre>CREATE USER ogg IDENTIFIED EXTERNALLY;</pre> <p>OS_AUTHENT_PREFIX が OPS\$ または別の文字列に設定されている場合は、ユーザー名は次のフォーマットで作成する必要があります。</p> <pre><OS_AUTHENT_PREFIX_value><OS_user_name></pre> <p>たとえば、OS ユーザー名が "ogg" の場合は、次のようにしてデータベース・ユーザーを作成します。</p> <pre>CREATE USER ops\$ogg IDENTIFIED BY oggpassword;</pre>
<user id> [, PASSWORD <password>]	<p>Oracle GoldenGate にデータベースレベルの認証を使用するように指示します。</p> <ul style="list-style-type: none"> ◆ <user id> には、データベース・ユーザー名、スキーマ (SQL/MX) または SQL*Net 接続文字列 (Oracle) を指定します。 ◆ <password> には、データベースに要求される場合にこのユーザーのパスワードを指定します。 <p>パスワードが ENCRYPT PASSWORD コマンドによって暗号化されている場合は、暗号化されたパスワードを使用します。それ以外の場合は、クリアテキストのパスワードを使用します。パスワードに大/小文字区別がある場合は、そのとおりに入力します。</p> <p>ユーザー ID またはパスワードが変更される場合は、Oracle GoldenGate パラメータ・ファイルにこの変更を加える必要があります。</p>
ENCRYPTKEY DEFAULT	<p>オプションの引数なしの ENCRYPT PASSWORD コマンドによって、パスワードがデフォルトの Oracle GoldenGate 鍵で暗号化されている場合に必要です。</p>

引数	説明
ENCRYPTKEY <keyname>	ENCRYPT PASSWORD ENCRYPTKEY <keyname> コマンドによって、ユーザー定義鍵を使用してパスワードが暗号化されているときに必要です。ENCKEYS ファイルに含まれているものと同じ論理鍵名を指定します。
SYSDBA	(Oracle) ユーザーが sysdba としてログインするように指定します。

- 例 1** USERID /
- 例 2** USERID ggs, PASSWORD ggs123
- 例 3** USERID ggs@ora1.ora, PASSWORD ggs123
- 例 4** USERID data1
- 例 5** USERID ggs, PASSWORD AACAAAAAAAAAAAAIALCKDZIRHOJBHOJUH, ENCRYPTKEY superx128
- 例 6** USERID ggs, PASSWORD AACAAAAAAAAAAAAIALCKDZIRHOJBHOJUH, ENCRYPTKEY default

USETIMEPREFIX

適用対象 Replicat

USETIMEPREFIX パラメータは、Teradata-ODBC ドライバの 3.02.0.02 より前のバージョンを使用する Teradata データベースで使用します。このパラメータは、TIME データ型のデータ値に TIME リテラル接頭辞を追加します。

USETIMEPREFIX を使用するときは、USEDATEPREFIX および USETIMESTAMPPREFIX パラメータを使用してください。これらのパラメータを使用するときは、NODYNSQL パラメータ（178 ページを参照してください）を指定する必要があります。

Teradata-ODBC ドライバ 3.02.0.02 以上では、このパラメータまたは NODYNSQL を使用する必要はありません。

デフォルト 無効

構文 USETIMEPREFIX

USETIMESTAMPPREFIX

適用対象 Replicat

USETIMESTAMPPREFIX パラメータは、Teradata-ODBC ドライバの 3.02.0.02 より前のバージョンを使用する Teradata データベースで使用します。このパラメータは、TIMESTAMP データ型のデータ値に TIMESTAMP リテラル接頭辞を追加します。

USETIMESTAMPPREFIX を使用するときは、USETIMEPREFIX および USEDATEPREFIX パラメータを使用してください。これらのパラメータを使用するときは、NODYNSQL パラメータ（178 ページを参照してください）を指定する必要があります。

Teradata-ODBC ドライバ 3.02.0.02 以上では、このパラメータまたは NODYNSQL を使用する必要はありません。

デフォルト 無効

構文 USETIMESTAMPPREFIX

VAM

適用対象 Extract

VAM パラメータでは、Teradata Access Module (TAM) を使用して Extract プロセスのためのデータ取得ファンクションを実行し、Extract API (VAM) にデータを送信することを指定します。

デフォルト なし

構文 VAM <library name>, PARAMS ("<param>" [, "<param>"] [, ...])

引数	説明
<library name>	データベース・ベンダーによって Windows DLL または UNIX 共有オブジェクトとして提供されているライブラリ名。このライブラリが Oracle GoldenGate ディレクトリ以外のディレクトリに存在する場合は、完全パス名を使用します。Teradata は、このライブラリを <i>Teradata Access Module (TAM)</i> と呼びます。このプログラム (ライブラリ) は、Oracle GoldenGate API (VAM) と通信します。
PARAMS <param>	<ul style="list-style-type: none"> PARAMS は必須のキーワードです。 <param> は、TAM 初期化ファイルや実行時パラメータなど、Oracle GoldenGate API に渡される、引用符で囲まれた任意のパラメータです。次の PARAMS オプション を参照してください。
SQL/MX PARAMS オプション	
ARLIBError <error>, <action>	<p>(オプション) VAM による TMFARLIB エラーの処理方法を指定します。</p> <ul style="list-style-type: none"> <error> は、ARLIB エラー番号です。 <action> には、ABEND WARN IGNORE を指定できます。 <p>デフォルトは ABEND です。エラー -1000 および -2000 は、他にアクションが指定されていても、常に ABEND になります。</p> <p>例：</p> <pre>Vam Params (arliberror (-16,-14), Warn) Vam Params (arliberror -2000, Abend) Vam Params (arliberror -1000, Abend)</pre>
ARErrorReportInterval <seconds>	<p>同一の TMFARLIB エラーが Extract にレポートで戻される間隔 (秒) を設定します。これにより、蓄積される各エラー・タイプのメッセージ量が削減されます。</p> <ul style="list-style-type: none"> <seconds> は、ゼロ以上である必要があります。 デフォルトは 60 秒です。

引数	説明
Teradata PARAMS オプション	
infile, <ini file>, callbackLib, extract.exe	<p>Teradata TAM に必須のパラメータです。</p> <ul style="list-style-type: none"> ◆ infile は、次のパラメータに TAM 初期化ファイルが指定されることを示します。 ◆ <ini file> は、TAM 初期化ファイル名です。このファイルがプログラムのインストール先と同じディレクトリに存在する場合を除き、完全修飾パス名を指定します。 ◆ callbackLib は、次のパラメータに TAM とのインタフェースになるプログラムが指定されることを示します。このパラメータは、大 / 小文字が区別され、ここに示したとおりに正確に入力する必要があります。 ◆ extract.exe は Extract プログラムで、TAM にとってコールバック・プログラムです。

例 VAM tam.dll, PARAMS (infile, tam.ini, callbackLib, extract.exe)

VARWIDTHNCHAR | NOVARWIDTHNCHAR

適用対象 Extract、Replicat、DEFGEN

VARWIDTHNCHAR および NOVARWIDTHNCHAR パラメータでは、NCHAR データのトレイルへの書込み方法と Replicat による解釈方法を制御します。

- VARWIDTHNCHAR を使用すると、NCHAR、NVARCHAR2 または NCLOB 文字セットは、可変長文字セット (UTF-8) として処理されます。
- NOVARWIDTHNCHAR を使用すると、NCHAR、NVARCHAR2 または NCLOB 文字セットは、UTF-16 として処理されます。
- いずれのオプションも指定しない場合は、NCHAR、NVARCHAR2 または NCLOB 文字セットの処理方法を決定するために、データベースの NLS_NCHAR_CHARACTERSET プロパティ値が使用されます。

デフォルト データベースから NLS_NCHAR_CHARACTERSET プロパティを使用する

構文 VARWIDTHNCHAR | NOVARWIDTHNCHAR

WARNLONGTRANS

適用対象 Extract

Extract で、トランザクションが長い間実行しているという警告メッセージが生成される前にトランザクションがオープンになる時間の長さを指定するには、WARNLONGTRANS パラメータを使用します。また、WARNLONGTRANS を使用して、Oracle GoldenGate による長い間実行されているトランザクションのチェックの回数を制御できます。

WARNLONGTRANS を指定すると、Oracle GoldenGate は指定されたしきい値を満たすトランザクションを確認し、発見した最初のトランザクションを Oracle GoldenGate エラー・ログ、Extract レポート・ファイル、およびシステム・ログにレポートします。デフォルトでは、Oracle GoldenGate はこの確認を 5 分ごとに繰り返します。

オープンしているトランザクションのリストのオンデマンド表示、トランザクション詳細のファイルへの出力、これらのトランザクションの中止またはトレイルへの強制的な書き込みの方法の詳細は、SEND EXTRACT コマンド (33 ページ) を参照してください。

このパラメータは、Oracle にのみ有効です。

デフォルト 1 時間（および別の処理スレッドを使用して 5 分間隔で確認）

構文
WARNLONGTRANS <duration><unit>
[, CHECKINTERVAL <interval><unit>]
[, NOUSETHEADS]
[, USELASTREADTIME]

引数	説明
<duration><unit>	<p>オープンしているトランザクションを長時間におよぶトランザクションとみなすまでの時間を設定します。デフォルトは 1 時間です。</p> <ul style="list-style-type: none"> ◆ <duration> には、時間の長さを整数で指定します。 ◆ <unit> には、second、minutes、hours または days を完全なスペリングまたは短縮形式で指定します。 <p>S SEC SECS SECOND SECONDS M MIN MINS MINUTE MINUTES H HOUR HOURS D DAY DAYS</p> <p><duration> および <unit> の間にスペースを入れないでください。</p>
CHECKINTERVAL <interval><unit>	<p>Oracle GoldenGate が WARNLONGTRANS を満たすトランザクションを確認し、レポートする間隔を設定します。</p> <ul style="list-style-type: none"> ◆ <interval> には、確認する時間間隔の長さを整数で指定します。 ◆ <unit> には、second、minutes、hours または days を完全なスペリングまたは短縮形式で指定します。 <p>S SEC SECS SECOND SECONDS M MIN MINS MINUTE MINUTES H HOUR HOURS D DAY DAYS</p> <p><interval> および <unit> の間にスペースを入れないでください。デフォルトは最小値の 5 分です。</p>

引数	説明
NOUSETHEADS	メイン・プロセス・スレッドによってモニタリングを実施することを指定します。デフォルトでは、モニタリングは個々のスレッドによって実施されます。NOUSETHEADS は、システムがマルチスレッドをサポートしていない場合にのみ使用するようになっています。
USELASTREADTIME	(Oracle のみ) トランザクションが長時間におよんでいるかどうかの判断基準として、Extract に常に REDO ログの最後の読取り時間を使用させます。デフォルトでは、Extract は REDO ログから読み取った最後のレコードのタイムスタンプを使用します。このパラメータは、ARCHIVEDLOGONLY オプションを使用して TRANLOGOPTIONS でアーカイブ・ログ専用モードで実行している Extract に適用されます。

例 WARNLONGTRANS 2h, CHECKINTERVAL 3m

WARNRATE

適用対象 Replicat

WARNRATE パラメータを使用して、プロセス・レポートおよびエラー・ログへのレポート前に、ターゲット表で許容可能な SQL エラー数のしきい値を設定できます。エラーは警告としてレポートされます。お使いの環境で多くのエラーが許容可能な場合は、WARNRATE の値を増やすことにより、これらのファイルのサイズを最小化できます。

デフォルト 100 エラー

構文 WARNRATE <num errors>

引数	説明
<num errors>	警告発行のしきい値にする SQL エラー数。

例 WARNRATE 1000

WILDCARDRESOLVE

適用対象 Extract および Replicat

WILDCARDRESOLVE パラメータでは、TABLE、SEQUENCE、または MAP 文でワイルドカードで指定されている表を処理するルールを変更します。WILDCARDRESOLVE は、パラメータ・ファイル内の関連する TABLE、SEQUENCE、または MAP 文よりも先に指定する必要があります。

ターゲット・オブジェクトは、ワイルドカードの解決を試みるときに、ターゲット・データベース上に存在している必要があります。ターゲット・オブジェクトが存在しない場合、Replicat は異常終了します。

デフォルト DYNAMIC

構文 WILDCARDRESOLVE {DYNAMIC | IMMEDIATE}

引数	説明
DYNAMIC	<p>ワイルドカード定義に一致するソース・オブジェクトは、ワイルドカード・ルールに一致するたびに解決されます。これはデフォルトです。</p> <p>SOURCEISTABLE または GENLOADFILES を指定している場合は、このオプションを使用しないでください。これらのパラメータを使用すると、WILDCARDRESOLVE は常に暗黙的に IMMEDIATE に設定されます。</p> <p>これは、Teradata に必須の引数です。</p> <p>SEQUENCE パラメータを指定し、ワイルドカードを使用して Oracle 順序をレプリケートするときは、DYNAMIC を使用する必要があります。</p> <p>デフォルトの動作が要求される場合、明示的な WILDCARDRESOLVE DYNAMIC パラメータ文はオプションです。ただし指定しておく、パラメータ・ファイルを確認する担当者が、使用されている方法を明確に把握できます。</p>
IMMEDIATE	<p>ワイルドカード定義に一致するソース・オブジェクトは、起動時に処理されます。このオプションは、Teradata ではサポートされていません。SOURCEISTABLE を使用する場合は、これが強制的にデフォルトになります。</p> <p>このオプションは、Oracle インターバル・パーティション機能をサポートしていません。Oracle GoldenGate が新しいパーティションを見つけられるように、動的な解決を行う必要があります。</p>

例 次に、起動時にワイルドカードを解決する例を示します。

```
WILDCARDRESOLVE IMMEDIATE
TABLE hq.acct_*;
```

第 3 章

Collector パラメータ

.....

この章では、Collector プロセス・パラメータについて説明します。Collector プロセスは、ターゲット・システム上で動作し、受信データを受け取ってトレイルに書き込みます。

動的 Collector

通常 Oracle GoldenGate ユーザーは、この Collector プロセスとやり取りを行いません。これは Manager プロセスによって動的に起動されます。これが *動的 Collector* と呼ばれます。

静的 Collector

静的 Collector は、次に示す構文と入力パラメータを使用してコマンドラインで SERVER プログラムを実行することにより、手動で実行できます。

構文 server <parameter> [<parameter>] [...]

Collector パラメータは、大 / 小文字が区別され、先頭にダッシュを付ける必要があります。

表 51 Collector パラメータ

パラメータ	説明
-cp <checkpoint file>	Collector が別名 Extract グループのために保持するチェックポイント・ファイルの名前を指定します。<checkpoint file> は、別名 Extract グループに関連付けられているパッシブ Extract グループ名です。 チェックポイント・ファイルは、パッシブ Extract グループが実行中かどうかを確認するために使用されます。チェックポイント・ファイルが Collector (server プログラム) によってロックされている場合は実行中です。-h および -p パラメータとともに使用する必要があります。 パッシブおよび別名 Extract グループの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。
-d <definitions file>	エクスポートされた表定義を含む、DEFGEN ユーティリティによって生成されたローカル・ファイル。
-E	受信ヘッダーおよびデータを、ASCII から EBCDIC フォーマットに変換します。デフォルトでは、Oracle GoldenGate はデータ変換を行いません。

.....

表 51 Collector パラメータ (続き)

パラメータ	説明
-e <version error type> <action>	<p>Collector にカスタマイズした方法で特定のフォーマット・エラーにレスポンスするように指示します。たいいてい場合はデフォルト値で問題ありません。複数のエラー・タイプを指定するときは、-e を複数回使用します。次に例を示します。</p> <pre>-e OLD CONTINUE -e NEW DISCARD.</pre> <p><version error type> には、次のいずれかを指定できます。</p> <p>NEW 想定よりも多くのデータ（現在の定義よりも多くの列）を含むレコードを確認します。Collector プロセスは、更新されたソース表が必要になることがあります（つまり DEFGEN をもう一度実行する必要があります）。デフォルトのアクションは ABEND です。</p> <p>OLD 想定より少ないデータを含むレコードを確認します。これは、通常はレコードに含まれる列が表の現在の定義よりも少ない状況で、正常な状態と考えられます。デフォルトのアクションは CONTINUE です。</p> <p>OUTOFSYNC 提供されている定義に従って変換できないレコードを確認します。デフォルトのアクションは ABEND です。</p> <p><action> には、次のいずれかを指定できます。</p> <p>ABEND レコードを破棄し、Extract プロセスに即座に終了するよう指示します。</p> <p>CONTINUE 検出された変換エラーにかかわらず、（可能な場合）レコードを処理します。</p> <p>DISCARD （破棄ファイルが -x で指定されている場合）、レコードを破棄ファイルに出力します。Collector は、エラー・ファイルに最初に破棄されたレコードの警告を送信し、レコードの処理を継続します。</p> <p>-ENCRYPT <encrypt type></p> <p>Extract パラメータ・ファイルの RMTHOST パラメータで指定され、Extract プロセスから渡される暗号化のタイプ。</p> <p>次の値が有効です。</p> <ul style="list-style-type: none"> ◆ NONE ◆ BLOWFISH <p>BLOWFISH を使用するときは、-KEYNAME オプションも指定します。Oracle GoldenGate のセキュリティの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。</p>

表 51 Collector パラメータ (続き)

パラメータ	説明
-f	Extract プロセスに成功ステータスを返す前に、常にファイルへの書き込みをディスクにフラッシュさせます。操作のたびにディスクにフラッシュするよりも効率的なため、デフォルトではファイル・システムは I/O をバッファします。I/O 成功の確認後、バッファがディスクにフラッシュされる前にシステムが停止し、データが損失するという小さなリスクよりも、通常はパフォーマンスのメリットのほうが重要です。このリスクが許容できない場合は、Oracle GoldenGate のパフォーマンスが低下する可能性を理解した上で、-f を使用してください。
-g	2GB より大きなファイルをサポートします (Solaris のみ)。
-h <host name or IP address>	ソース・システムの名前または IP アドレスを指定します。このオプションは、ソース上でパッシブ・モードで実行されている Extract と関連付けられているターゲット上の別名 Extract を使用するときを使用します。これにより、Collector を接続モードで動作させます。このモードでは、ソース Extract からの接続リクエストを待機するかわりに、ソース Extract への TCP/IP 接続を開始します。-p Collector オプションとともに使用する必要があります。 パッシブおよび別名 Extract グループの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。
-k	サービス提供先の Extract プロセスが接続を切断したときに終了するように Collector に指示します。このオプションは、Collector プロセスを起動するときに Manager プロセスによって使用されます。
-KEYNAME <name>	ローカル ENCKEYS 参照ファイルで定義されている鍵名を指定します。-ENCRYPT に BLOWFISH を指定しているときに使用します。
-l <file name>	指定したファイルに出力を記録します。完全修飾名を使用します。
-m	割り当てる最大ログ・ファイル数を指定します。
-P <parameter file>	Collector パラメータを含むローカル・ファイル。このファイルのパラメータは、Extract プロセスから送信されるパラメータより優先されます。
-p <port number>	次のように指定する TCP/IP ポート番号： <ul style="list-style-type: none"> ◆ 通常の Extract または通常のデータ・ポンプの場合：Collector プロセスが Extract からの接続リクエストをリスニングするポート。 ◆ パッシブ・モードで実行されている Extract またはデータ・ポンプの場合：Extract またはデータ・ポンプが Collector からの接続リクエストをリスニングするポート。この場合は、-h <host> パラメータとともに使用する必要があります。 デフォルトはポート 7819 です。 パッシブおよび別名 Extract グループの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。
-R <alternate value>	無効な数値 ASCII データを代替値に置換します。デフォルトは 0 に置換します。代替値には、次のいずれかを指定できます。

表 51 Collector パラメータ (続き)

パラメータ	説明
	<number> 代替値を指定します。
	NULL ターゲット列が NULL 値を受け付ける場合は NULL を指定し、それ以外の場合はゼロに置換します。
	UNPRINTABLE 出力不可能なデータを含むすべての列を拒否します。プロセスは停止し、不正な値をレポートします。
	NONE 数値データの置換を行いません。Oracle GoldenGate はそのままのデータのレプリケートを試みます。
-x <discard file>	フォーマットできないレコードを出力する破棄ファイルを指定します。完全修飾名を使用します。
-UL	<p>ターゲット・システムに影響を与えるネットワーク障害が発生している場合に、Collector によってロックされているトレイル・ファイルの問題を回避します。Oracle GoldenGate リリース 10.4 以上では、Collector はトレイルへのデータの書き込み中に、同一のトレイルへの書き込みを試みる他のプロセスによってトレイルが破損されないように、トレイル・ファイルをロックします。</p> <p>-UL オプションは、Collector によるトレイル・ファイルのロックを防止します。この機能は、ソースまたはターゲット・システムで GLOBALS パラメータとしても使用できます。UNLOCKEDTRAILFILES パラメータの項を参照してください。</p>

第 4 章

列変換ファンクション

.....

Oracle GoldenGate の列変換ファンクションを使用すると、ソース値をターゲット列に適切なフォーマットに変換できます。これらのファンクションでは、数字と文字の操作、テストの実行、パラメータ値の抽出、環境情報のリターンなどを行うことができます。

列変換ファンクションでの Unicode およびネイティブ・エンコーディングの使用

Oracle GoldenGate では、Unicode、または Windows、文字列の列内の文字を UNIX および Linux オペレーティング・システムのネイティブ・キャラクタ・エンコーディングで表示するためのエスケープ・シーケンスの使用がサポートされます。エスケープ・シーケンスは、TABLE または MAP 文内の次の要素で使用されます。

- WHERE 句
- Unicode キャラクタを Unicode 列に割り当てる、またはネイティブ・エンコーディングの文字を列に割り当てる COLMAP 句
- COLMAP 句内の Oracle GoldenGate 列変換関数

Oracle GoldenGate では、次のタイプのエスケープ・シーケンスがサポートされます。

- \uFFFF Unicode エスケープ・シーケンス
- \377 8 進数エスケープ・シーケンス
- \xFF 16 進数エスケープ・シーケンス

次の制限が適用されます。

- このサポートは、7ビットASCIIに相当するU+0000～U+007Fの間のUTF-16コード・ポイントに制限されます。
- ソース列およびターゲット列の両方が Unicode である必要があります。
- ソースおよびターゲットのデータ・タイプが同一である必要があります(NCHAR から NCHAR への変換など)。

エスケープ・シーケンスの使用手順

各エスケープ・シーケンスの先頭にリバース・ソリダスを付け（コード・ポイント U+005C）、次にキャラクタ・コード・ポイントを指定します。（リバース・ソリダスは、一般にはバックスラッシュ記号として知られています。）パラメータ文内の入力文字列または列変換関数内で、実際の文字のかわりにエスケープ・シーケンスを使用します。

.....

\uFFFF Unicode エスケープ・シーケンスの使用手順

- 小文字の **u** を先頭に付け、4 桁の 16 進数のみを指定します。
- サポートされる範囲は次のとおりです。
 - 0 ～ 9 (U+0030 ～ U+0039)
 - A ～ F (U+0041 ～ U+0046)
 - a ～ f (U+0061 ～ U+0066)
- NCHAR および NVARCHAR 列に使用可能なエスケープ・シーケンスは、これのみです。
- サロゲート・ペアはサポートされていません。

例 \u20ac は、ユーロ通貨記号用の Unicode エスケープ・シーケンスです。

注意 クロスプラットフォームのサポートを確実なものにするには、Unicode エスケープ・シーケンスを使用します。8 進数および 16 進数のエスケープ・シーケンスは、異種のオペレーティング・システム間では標準化されていません。

\377 8 進数エスケープ・シーケンスの使用手順

- 3 つの 8 進数が含まれている必要があります。
- サポートされる範囲は次のとおりです。
 - 最初の桁の範囲が 0 ～ 3 (U+0030 ～ U+0033) です。
 - 2 番目と 3 番目の桁の範囲は 0 ～ 7 (U+0030 ～ U+0037) です。

例 \200 は、Microsoft Windows 環境でのユーロ通貨記号用の 8 進数エスケープ・シーケンスです。

\xFF 16 進数エスケープ・シーケンスの使用手順

- 小文字の **x** を先頭に付け、2 桁の 16 進数のみを指定します。
- サポートされる範囲は次のとおりです。
 - 0 ～ 9 (U+0030 ～ U+0039)
 - A ～ F (U+0041 ～ U+0046)
 - a ～ f (U+0061 ～ U+0066)

例 \x80 は、Microsoft Windows 環境でのユーロ通貨記号用の 16 進数エスケープ・シーケンスです。

例 次の例では、列変換ファンクションを使用して、節記号文字 (U+00A7 出表現) および Unicode 列 SECTION を結合しています。

```
COLMAP ( SECTION = @STRCAT( "\u00a7", SECTION ) );
```

列変換ファンクションの概要

この概要は、Oracle GoldenGate ファンクションで実行できる処理タイプ別に構成されています。
アルファベット順リファレンスは 397 ページから記載しています。

表 52 テストの実行

ファンクション	説明
CASE	一連の値テストに応じて値を選択します。
EVAL	一連の独立テストに基づいて値を選択します。
IF	条件文から TRUE または FALSE のどちらを返されるかによって、2 つの値から 1 つを選択します。

表 53 行方不明列の処理

ファンクション	説明
COLSTAT	列が MISSING、NULL、または INVALID であることのインジケータを返します。
COLTEST	列が PRESENT、MISSING、NULL、または INVALID かどうかをテストする条件計算を実行します。

表 54 日付の操作

ファンクション	説明
DATE	ソース列に渡されたフォーマットに基づいて日付と時刻を返します。
DATEDIFF	2 つの日付または時刻の差を返します。
DATENOW	現在の日付と時刻を返します。

表 55 算術計算を実行します。

ファンクション	説明
COMPUTE	算術式の結果を返します。

表 56 文字列の操作

ファンクション	説明
NUMBIN	バイナリ文字列を数字に変換します。
NUMSTR	文字列を数字に変換します。
STRCAT	1 つ以上の文字列を連結します。
STRCMP	2 つの文字列を比較します。
STREXT	文字列の一部を抽出します。
STREQ	2 つの文字列が等しいかどうかを確認します。
STRFIND	文字列内の文字列を検索します。
STRLEN	文字列の長さを返します。
STRLTRIM	先行する空白を切り捨てます。
STRNCAT	1 つ以上の文字列を最大長まで連結します。
STRNCMP	指定された文字数に基づいて 2 つの文字列を比較します。
STRNUM	数字を文字列に変換します。
STRRTRIM	末尾の空白を切り捨てます。
STRSUB	文字列を別の文字列に置換します。
STRTRIM	先行の空白と末尾の空白を切り捨てます。
STRUP	文字列を大文字に変更します。
VALONEOF	文字列または文字列用の列を値のリストを比較します。

表 57 その他のファンクション

ファンクション	説明
BINARY	ソース列が文字用の列として定義されているときに、ソース列のバイナリ・データをターゲット列でバイナリ・データとして保持します。
BINTOHEX	バイナリ文字列を 16 進数文字列に変換します。
GETENV	環境情報を返します。
GETVAL	FILTER または COLMAP 句の入力としてストアド・プロシージャからパラメータを抽出します。
HEXTOBIN	16 進数文字列をバイナリ文字列に変換します。
HIGHVAL LOWVAL	値を最大値または最小値までに抑止します。
RANGE	並列処理のために行を複数のデータ・グループに分割します。
TOKEN	トレイル・レコード・ヘッダーからトークン・データを取得します。

BINARY

@BINARY ファンクションは、列変換ファンクションに参照されるソース列が、文字用の列として定義されているものの、ターゲットではバイナリのまま保持する必要があるバイナリ・データを含んでいるときに使用します。デフォルトでは、文字用の列のバイナリ・データは（必要な場合に）ASCII に変換され、NULL で終了された文字列とみなされます。@BINARY ファンクションは、任意のバイナリ・データをターゲット列にコピーします。

構文 @BINARY (<column name>)

引数	説明
<column name>	データのコピー先のターゲット列名。

例 次に、@BINARY を使用してソース列 ACCT_CREATE_DATE からターゲット列 ACCT_CHIEF_COMPLAINT にデータをコピーする方法の例を示します。

```
ACCT_CHIEF_COMPLAINT =
@IF ( @NUMBIN (ACCT_CREATE_DATE ) < 48633, "xxxxxxx",
@BINARY (ACCT_CHIEF_COMPLAINT) )
```

BINTOHEX

@BINTOHEX ファンクションでは、指定されたバイナリ・データを対応する 16 進数に変換します。

構文 @BINTOHEX (<data>)

引数	説明
<data>	ソース列名、式、または引用符で囲まれたリテラル文字列。

例 @BINTOHEX ("12345") は "3132333435" に変換されます。

CASE

@CASE ファンクションでは、一連の値テストに応じて値を 1 つ選択します。@CASE でテストできるケース数に制限はありません。ケース数が多い場合は、パフォーマンスを最適化するために最も頻繁に検出される条件を最初にリストします。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @CASE (<value>, <test value1>, <test result1>
[, <test value2>, <test result2>] [, ...]
[, <default result>])

引数	説明
<value>	テストする値 (列名など)。リテラルは引用符で囲みます。
<test value>	<value> に有効な結果。リテラルは引用符で囲みます。
<test result>	<test value> の値に基づいて返される値。リテラルは引用符で囲みます。
<default result>	<value> がどの <test value> 値にも該当しない場合に返されるデフォルト値。リテラルは引用符で囲みます。

例 1 次の例では、PRODUCT_CODE が "CAR" の場合は "A car" が返され、PRODUCT_CODE が "TRUCK" の場合は "A truck" が返されます。PRODUCT_CODE が最初の 2 つのケースのいずれにも該当しない場合は、デフォルト値が指定されていないため、FIELD_MISSING インジケーションが返されます。

```
@CASE (PRODUCT_CODE, "CAR", "A car", "TRUCK", "A truck")
```

例 2 次の例は上記の例に似ていますが、ここではデフォルト値が指定されています。PRODUCT_CODE が "CAR" にも "TRUCK" にも該当しない場合、このファンクションは "A vehicle" を返します。

```
@CASE (PRODUCT_CODE, "CAR", "A car", "TRUCK", "A truck", "A vehicle")
```


COLSTAT

列が欠落している、NULLまたは無効な Extract あるいは Replicat にインジケータを返すには、@COLSTAT 関数を使用します。このインジケータは、追加の変換関数を使用する、より大きな操作式の一部として使用できます。

- 構文

@COLSTAT ({MISSING | NULL | INVALID})
- 例 1

次の例では、ターゲット列 ITEM に NULL が返されます。

ITEM = @COLSTAT (NULL)
- 例 2

次の @IF 計算では、PRICE および QUANTITY がゼロ未満である場合に @COLSTAT を使用して NULL がターゲット列に返されます。

ORDER_TOTAL = PRICE * QUANTITY, @IF (PRICE < 0 AND QUANTITY < 0, @COLSTAT (NULL))

COLTEST

@COLTEST ファンクションでは、1 つ以上の列条件を使用したテストによる条件演算を有効にします。条件が満たされた場合、@COLTEST は TRUE を返します。条件演算を実行するには、@IF ファンクションを使用します。

- 構文

@COLTEST (<source column>, <test item> [, <test item>] [, ...])

引数	説明
<source column>	ソース列名。
<test item>	次の値が有効です。 PRESENT 列がソース・レコードに存在し、NULL でないことを示します。データベースが、変更されない NULL と同じではない列の値を記録しない場合、列の値は MISSING になることがあります。 NULL 列がソース・レコードに存在し、NULL であることを示します。 MISSING 列がソース・レコードに存在しないこと示します。 INVALID 列がソース・レコードに存在するが無効なデータを含んでいることを示します。

- 例 1

次の例では、@IF を使用して、ソース・レコードに BASE_SALARY 列が存在し（および NULL でなく）、かつ 250000 より大きい場合に、HIGH_SALARY 列に値をマップします。それ以外の場合は、NULL が返されます。

HIGH_SALARY =
@IF (@COLTEST (BASE_SALARY, PRESENT) AND
BASE_SALARY > 250000,
BASE_SALARY, @COLSTAT (NULL))

例 2

次の例では、AMT 列が MISSING または INVALID の場合には 0 が返され、それ以外の場合には AMT の値が返されます。

```
AMOUNT = @IF (@COLTEST (AMT, MISSING, INVALID), 0, AMT)
```

COMPUTE

@COMPUTE ファンクションでは、算術式の値をターゲット列に返します。ファンクションから返される値は、文字列形式です。

算術式の値を別の Oracle GoldenGate ファンクションに返すときは、次の例のように @COMPUTE 句を省略できます。

```
@STRNUM ((AMOUNT1 + AMOUNT2), LEFT)
```

上記は、次と同じ結果を返します。

```
@STRNUM ((@COMPUTE (AMOUNT1 + AMOUNT2), LEFT)
```

算術式では、次の要素を組み合わせることができます。

- 数値
- 数値が含まれる列の名前
- 数値を返す関数
- 算術演算子：
 - + (加算)
 - (減算)
 - * (乗算)
 - / (除算)
 - \ (剰余算)
- 比較演算子：
 - > (大なり)
 - >= (以上)
 - < (小なり)
 - <= (以下)
 - = (等しい)
 - <> (等しくない)

比較から返される結果。ゼロ (FALSE を示す) またはゼロ以外 (TRUE)。

- カッコ (式内で結果をグループにまとめる)
- 論理演算子の AND、OR。Oracle GoldenGate では、論理式の必要な部分のみが評価されます。文が FALSE になると、この式の残りの部分は無視されます。これは、欠落しているまたは NULL のフィールドを評価する場合に便利です。たとえば、COL1 の値が 25 で、COL2 の値が 10 の場合、次が予想されます。

@COMPUTE (COL1 > 0 AND COL2 < 3) が 0 を返す。

@COMPUTE (COL1 < 0 AND COL2 < 3) が 0 を返す。COL2 < 3 は評価されない。

@COMPUTE ((COL1 + COL2)/5) が 7 を返す。

構文 @COMPUTE(<expression>)

引数	説明
<expression>	有効な算術式。

例 1 AMOUNT_TOTAL = @COMPUTE (AMT + AMT2)
例 2 AMOUNT_TOTAL = @IF (AMT >= 0, AMT * 100, 0)
例 3 ANNUAL_SALARY = @COMPUTE (MONTHLY_SALARY * 12)

DATE

@DATE ファンクションでは、ソース列に渡された形式に基づいて、ターゲット列に様々な形式で日付と時間を返します。@DATE は、実質上すべてのタイプの入力を有効な **SQL** 日付に変換します。@DATE は、日付列の一部の抽出や、日付に基づいた数値タイムスタンプの計算にも使用できます。

構文 @DATE ("<output descriptor>", "<input descriptor>", <source col>
 [, "<input descriptor>", <source col>] [, ...])

引数	説明
"<output descriptor>"	ターゲット列が必要とする日付ディスクリプタおよびオプションのリテラル値（空白やコロンなど）を含むターゲット文字列。日付ディスクリプタは、必要に応じて結合できます。402 ページの表 58 の日付ディスクリプタの説明を参照してください。形式ディスクリプタは、ターゲットの date/time/timestamp 形式と一致する必要があります。Oracle GoldenGate は、必要な場合、正しい形式にするために指定された形式を上書きします。
"<input descriptor>"	日付ディスクリプタおよびオプションのリテラル値（空白やコロンなど）を含むソース文字列。日付ディスクリプタは、必要に応じて結合できます。次に例を示します。 ディスクリプタ文字列 "YYYYMMDD" は、次に指定される数字または文字用の列に、順番に 4 桁の西暦（YYYY）、月（MM）および日（DD）が含まれていることを示します。 ディスクリプタ文字列 "DD/MM/YY" は、フィールドに日、スラッシュ、月、スラッシュ、西暦の下 2 桁が含まれていることを示します。 日付ディスクリプタの詳細は、表 58 を参照してください。
<source col>	上記の入力を提供するソース列名。

表 58 日付ディスクリプタ

ディスクリプタ	説明	有効
CC	100 年	入力 / 出力
YY	西暦の下 2 桁	入力 / 出力
YYYY	4 桁の西暦	入力 / 出力
MM	数字の月	入力 / 出力
MMM	英語の月 (APR、OCT など)	入力 / 出力
DD	月単位の日付	入力 / 出力
DDD	年単位の日付 (001、365 など)	入力 / 出力
DOW0	数値で表す曜日 (Sunday = 0)	入力 / 出力
DOW1	数値で表す曜日 (Sunday = 1)	入力 / 出力
DOWA	英語で表す曜日 (SUN、MON、TUE など)	入力 / 出力
HH	時間	入力 / 出力
MI	分	入力 / 出力
SS	秒	入力 / 出力
JTSLCT	すでにローカル時間のユリウス・タイムスタンプに対して、またはユリウス・タイムスタンプに変換するときにローカル時間を保持するために使用します。	入力 / 出力
JTSGMT	ユリウス・タイムスタンプ (JTS と同じ)。	入力 / 出力
JTS	ユリウス・タイムスタンプ。JUL および JTS は、数式で利用できる数字を生成します。単位はマイクロ秒です。Windows のタイムスタンプはミリ秒の粒度を使用するので、Windows マシンでは値にゼロが埋め込まれます。	入力 / 出力
JUL	ユリウス日。JUL および JTS は、数式で利用できる数字を生成します。	入力 / 出力
TTS	NonStop の 48 ビット・タイムスタンプ	入力
PHAMIS	PHAMIS アプリケーション日付形式	入力
FFFFFF	端数 (マイクロ秒まで)	入力 / 出力
STRATUS	STRATUS アプリケーション・タイムスタンプ	入力 / 出力

表 58 日付ディスクリプタ (続き)

ディスクリプタ	説明	有効
CDATE	エポック以降の C タイムスタンプ (秒)	入力 / 出力
例 1	<p>下 2 桁の西暦が提供されているインスタンスで、出力で 4 桁の西暦が必要な場合には、複数のオプションを使用して正しい 100 年を取得できます。</p> <ul style="list-style-type: none"> 100 年を次のようにハードコードできます。 "CC", 19 or "CC", 20 @IF ファンクションを使用して、次のように条件を設定できます。 "CC", @IF (YY > 70, 19, 20) <p>これにより、下 2 桁の西暦が 70 より大きい場合に、100 年を 19 に設定し、それ以外の場合は 20 に設定できます。</p> <ul style="list-style-type: none"> システムは 100 年を自動的に計算できます。下 2 桁の西暦が 50 未満の場合、システムは 100 年を 20 とし、それ以外の場合は 100 年を 19 とし算出します。 	
例 2	<p>次の例では、西暦、月、および日列を日付に変換します。</p> <pre>date_col = @DATE ("YYYY-MM-DD", "YY", date1_yy, "MM", date1_mm, "DD", date1_dd)</pre>	
例 3	<p>次の例では、日付と時間を変換し、秒をデフォルトでゼロにします。</p> <pre>date_col = @DATE ("YYYY-MM-DD:HH:MI:00", "YYMMDD", date1, "HHMI", time1)</pre>	
例 4	<p>次の例では、YYYYMMDDHHMISS として保持されている数値列を SQL 日付に変換します。</p> <pre>datetime_col = @DATE ("YYYY-MM-DD:HH:MI:SS", "YYYYMMDDHHMISS", numeric_date)</pre>	
例 5	<p>次の例では、YYYYMMDDHHMISS として保持されている数値列をユリウス・タイムスタンプに変換します。</p> <pre>julian_ts_col = @DATE ("JTS", "YYYYMMDDHHMISS", numeric_date)</pre>	
例 6	<p>次の例では、ユリウス・タイムスタンプ列を 2 つの異なる列 (YYYY-MM-DD:HH:MI:SS 形式の日付時間列と、タイムスタンプのマイクロ秒部分を保持する端数列) に変換します。</p> <pre>datetime_col = @DATE ("YYYY-MM-DD:HH:MI:SS", "JTS", jts_field), fraction_col = @DATE ("FFFFFF", "JTS", jts_field)</pre>	
例 7	<p>次の例では、受注が処理された時間を生成します。内部の @DATE 式は、order_taken 列をユリウス・タイムスタンプに変更した後、マイクロ秒に変換された order_minutes 列をこのタイムスタンプに追加します。この式は、新しいユリウス・タイムスタンプとして外部 @DATE 式に戻され、読みやすい日付と時間に再度変換されます。</p> <pre>order_filled = @DATE ("YYYY-MM-DD:HH:MI:SS", "JTS", @DATE ("JTS", "YYMMDDHHMISS", order_taken) + order_minutes * 60 * 1000000)</pre>	

例 8 次の例では、完全な時間の計算を実行します。ソース日付列（名前 "dt"）に 5 時間を足して変換された日時が、ターゲット列（名前 "dt5"）に返されます。また、ソース・タイムスタンプ列（名前 "ts"）に 5 時間を足して変換されたタイムスタンプが、ターゲット列（名前 "ts5"）に返されます。

```
MAP scratch.t4, TARGET scratch.t4_copy,  
COLMAP (USEDEFAULTS,  
dt5 = @DATE ("YYYY-MM-DD:HH:MI:SS", "JTS",  
@COMPUTE (@DATE ("JTS", "YYYY-MM-DD:HH:MI:SS", dt) + 180000000000 ) ),  
ts5 = @DATE ("YYYY-MM-DD:HH:MI:SS.FFFFFFFF", "JTS",  
@COMPUTE ( @DATE ("JTS", "YYYY-MM-DD:HH:MI:SS.FFFFFFFF", ts) + 180000000000 ) )  
);
```

DATEDIFF

@DATEDIFF ファンクションでは、2 つの日付または日時の差異を日数または秒数で計算します。

構文 @DATEDIFF ("difference", "<date>", "<date>")

引数	説明
<difference>	指定した日付の差異。次の値が有効です。 DD 差異を日数で計算します。 SS 差異を秒数で計算します。
<date>	引用符で囲んだ YYYY-MM-DD[*HH:MI[:SS]] 形式の文字列（* はコロン (:) または空白でも可）、または現在の日付を返す引用符なしの @DATENOW ファンクション。

- 例 1** 次の例では、2006 年 1 月 1 日からの日数を計算します。
YTD = @DATEDIFF ("DD", "2006-01-01", @DATENOW ())
- 例 2** 次の例では、その年の現在までの日数を計算します（@DATEDIFF は、2006-01-01 に対して 0 を返します）。
todays_day = @COMPUTE (@DATEDIFF ("DD", "2006-01-01", @DATENOW ()) +1)

DATENOW

@DATENOW ファンクションでは、現在の日時を YYYY-MM-DD HH:MI:SS 形式で返します。日時は、夏時間の調整後、ローカル時間で返されます。@DATENOW は引数を取りません。

構文 @DATENOW ()

EVAL

@EVAL ファンクションでは、一連の独立したテストに基づいて値を選択します。テストできる条件数に制限はありません。ケース数が多い場合は、パフォーマンスを最適化するために最も頻繁に検出される条件を最初にリストします。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

```
構文      @EVAL (<condition1>, <result1>
          [<condition2>, <result2>] [, ....]
          [, <default result>])
```

引数	説明
<condition>	標準の条件演算子を使用する条件テスト。
<result>	条件テストの結果に基づいて返される値または文字列。リテラルは二重引用符で囲みます。
<default result>	どの条件も満たされなかったときに返されるデフォルトの結果。デフォルト結果はオプションです。

例 1 次の例では、AMOUNT 列が 10000 より大きい場合に、"high amount" が返されます。AMOUNT が 5000 より大きい（かつ 10000 以下）の場合（前の条件が満たされなかった場合）は、"somewhat high" が返されます。いずれの条件も満たさない場合、デフォルト結果が指定されていないため、COLUMN_MISSING インジケーションが返されます。

```
AMOUNT_DESC = @EVAL (AMOUNT > 10000, "high amount", AMOUNT > 5000, "somewhat high" )
```

例 2 次に、上記を変更した例を示します。同じ結果が返されますが、デフォルト値が指定されたため、AMOUNT が 5000 以下の場合には結果 "lower" が返されます。

```
@EVAL (AMOUNT > 10000, "high amount", AMOUNT > 5000, "somewhat high", "lower")
```

GETENV

@GETENV ファンクションでは、Oracle GoldenGate 環境に関する情報を返します。この情報は、次への入力として使用できます。

- ストアド・プロシージャまたは問い合わせ（SQLEXEC を使用）
- 列マップ（TABLE または MAP の COLMAP オプションを使用）
- ユーザー・トークン（TABLE の TOKENS オプションで定義され、@TOKENS ファンクションによってターゲット列にマッピング済）
- GET_ENV_VALUE ユーザー・イグジット・ファンクション（466 ページを参照）

表 59. GETENV オプションの概要

情報タイプ	説明
一般情報タイプ	
("LAG" , "<unit>")	ラグ情報を返します。
("LASTERR" , "<option>")	最後にレプリケートされた操作（詳細なエラー情報を含む）に関する情報を返します。
("JULIANTIMESTAMP")	現在のシステム時間をユリウス形式で返します。
("RECSOUTPUT")	起動以降に Extract がトレイル・ファイルに書き込んだレコード数を返します。
Oracle GoldenGate 情報タイプ	
("GGENVIRONMENT", "<option>")	Oracle GoldenGate 環境情報を返します。
("GGFILEHEADER", "<option>")	ファイル・ヘッダーに保持されている Oracle GoldenGate トレイル・ファイルのフォーマットおよびプロパティを返します。
("GGHEADER", "<option>")	Oracle GoldenGate レコード・ヘッダー情報を返します。
("RECORD", "<option>")	Oracle GoldenGate トレイル・ファイルのレコードの場所を示す順序番号および RBA を返します。
データベース情報タイプ	
("DBENVIRONMENT", "<option>")	グローバル・データベース環境情報を返します。
("TRANSACTION", "<option>")	ソース・トランザクションに関する情報を返します。
オペレーティング・システム情報タイプ	
("OSVARIABLE", "<variable>")	指定されたオペレーティングシステム環境変数の文字列値を返します。
Base 24 情報タイプ	
("TLFKEY", "SYSKEY" "<unique key>")	一意キーを ACI の Base24 アプリケーションの TLF/PTLF レコードに関連付けます。

例 次の例では、TABLE 文の TOKENS 句で @GETENV ファンクションを使用して、Oracle GoldenGate レコード・ヘッダー内にユーザー・トークンを移入します。次のようにこのファンクションの複数のオプションを組み合わせて使用して、特定の情報を返すことができます。

```
TABLE fin.product, TOKENS (  
  TKN-OSUSER = @GETENV ("GGENVIRONMENT", "OSUSERNAME"),  
  TKN-DOMAIN = @GETENV ("GGENVIRONMENT", "DOMAINNAME"),  
  TKN-COMMIT-TS = @GETENV ("GGHEADER", "COMMITTIMESTAMP"),  
  TKN-TABLE = @GETENV ("GGHEADER", "TABLENAME"),  
  TKN-OP-TYPE = @GETENV ("GGHEADER", "OPTYPE"),  
  TKN-LENGTH = @GETENV ("GGHEADER", "RECORDLENGTH"),  
  TKN-LAG-SEC = @GETENV ("LAG", "SECONDS"),  
  TKN-DB-USER = @GETENV ("DBENVIRONMENT", "DBUSER"),  
  TKN-DB-VER = @GETENV ("DBENVIRONMENT", "DBVERSION"),  
  TKN-ROWID = @GETENV ("RECORD", "GDVN"));
```

LAG 情報タイプの使用

@GETENV の LAG オプションでは、ラグ情報を返します。ラグは、Extract または Replicat がレコードを処理した時間と、データ・ソースのそのレコードのタイムスタンプとの差異です。LAG と <environment value> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("LAG", "<unit>")

環境値	戻り値
"SEC"	ラグ（秒）を返します。LAG に対して単位が明示的に指定されていないときは、これがデフォルトです。
"MSEC"	ラグ（ミリ秒）を返します。
"MIN"	ラグ（分）を返します。

LASTERR 情報タイプの使用

@GETENV の LASTERR オプションでは、Replicat が最後に失敗した操作の情報を返します。このオプションはエラー情報を提供します。LASTERR は、Replicat プロセスのみとの使用が有効です。LASTERR と <environment value> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("LASTERR", "<return value>")

環境値	戻り値
"DBERRNUM"	失敗した操作に関連するデータベース・エラー番号を返します。
"DBERRMSG"	失敗した操作に関連するデータベース・エラー・メッセージを返します。
"OPTYPE"	試行した操作タイプを返します。Oracle GoldenGate 操作タイプのリストは、『Oracle GoldenGate Windows and UNIX 管理者ガイド』の付録を参照してください。
"ERRTYPE"	エラーのタイプを返します。次のような結果になります。 <ul style="list-style-type: none"> ◆ DB (データベース・エラーの場合) ◆ MAP (マッピング・エラーの場合)

JULIANTIMESTAMP 情報タイプの使用

@GETENV の JULIANTIMESTAMP オプションでは、現在の時刻をユリウス形式で返します。単位はマイクロ秒（百万分の 1 秒）です。Windows のタイムスタンプはミリ秒（1000 分の 1 秒）の粒度を使用するので、Windows マシンでは値にゼロが埋め込まれます。次に、一般的な列マッピングの例を示します。

```
MAP dbo.tab8451, Target targ.tabjts, COLMAP (USEDEFAULTS, &
JTSS = @GETENV ("JULIANTIMESTAMP")
JTSFFFFFF = @date ("yyyy-mm-dd:hh:mi:ss.ffffff", "JTS", &
@getenv ("JULIANTIMESTAMP") ) )
;
```

JTSS および JTSFFFFFF 列は、次のような値になります。

```
212096320960773000 2008-12-17:16:42:40.773000
212096321536540000 2008-12-17:16:52:16.540000
212096322856385000 2008-12-17:17:14:16.385000
212096323062919000 2008-12-17:17:17:42.919000
212096380852787000 2008-12-18:09:20:52.787000
```

数字の最後の 3 桁（マイクロ秒）はすべて、埋め込まれたゼロが含まれています。

構文 @GETENV ("JULIANTIMESTAMP")

RECSOUTPUT 情報タイプの使用

@GETENV の RECSOUTPUT オプションでは、起動後に Extract がトレイル・ファイルに書き込んだレコード数の現在のカウンタを取得します。返される値は、表またはトランザクションではなく、Extract セッション自体に固有の値です。Extract が停止して再起動されると、カウンタは 1 にリセットされます。

構文 @GETENV ("RECSOUTPUT")

GGENVIRONMENT 情報タイプの使用

@GETENV の GGENVIRONMENT オプションでは、Oracle GoldenGate 環境に関する情報を返します。このオプションは、Extract および Replicat プロセスに有効です。GGENVIRONMENT と <environment value> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("GGENVIRONMENT", "<return value>")

環境値	戻り値
"DOMAINNAME"	(Windows のみ) プロセスを開始したユーザーに関連付けられているドメイン名を返します。
"GROUPDESCRIPTION"	GGSCI の ADD コマンドでのグループ作成時に DESCRIPTION オプションで説明が指定されている場合に、チェックポイント・ファイルから取得したグループの説明。
"GROUPNAME"	プロセス・グループ名を返します。
"GROUPTYPE"	プロセスのタイプ (EXTRACT または REPLICAT) を返します。
"HOSTNAME"	Extract または Replicat プロセスが実行されているシステムの名前を返します。
"OSUSERNAME"	プロセスを起動したオペレーティング・システムのユーザー名を返します。
"PROCESSID"	オペレーティング・システムによってプロセスに割り当てられているプロセス ID。

GGHEADER 情報タイプの使用

@GETENV の GGHEADER オプションでは、Oracle GoldenGate トレイル・レコードのヘッダー部分の情報を返します。Oracle GoldenGate トレイル内のすべてのデータ・レコードには、レコードのトランザクション環境が記述されたヘッダーが含まれています。レコード・ヘッダーの詳細は、『Oracle GoldenGate Windows and UNIX リファレンス・ガイド』を参照してください。

このオプションは、Extract および Replicat プロセスに有効です。GGHEADER と <environment value> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("GGHEADER", "<return value>")

環境値	戻り値
"BEFOREAFTERINDICATOR"	レコードがビフォア・イメージかアフター・イメージかを示すビフォアまたはアフター・インジケータを返します。次のような結果になります。 ◆ BEFORE (ビフォア・イメージ) ◆ AFTER (アフター・イメージ)
"COMMITTIMESTAMP"	次の例のように、YYY-MM-DD HH:MI:SS.FFFFFFFF 形式で表現されたトランザクション・タイムスタンプ（トランザクションがコミットされた時間）を返します。 2006-01-24 17:08:59.000000
"LOGPOSITION"	データ・ソース内の Extract プロセスの位置を返します。（LOGRBA オプションを参照してください。）
"LOGRBA"	LOGRBA および LOGPOSITION は、データ・ソース内のレコードの詳細な位置を保持します。トランザクション・ログベースの製品では、LOGRBA は順序番号、LOGPOSITION は相対バイト・アドレスです。ただしこれらの値は、取得方法とデータベースのタイプによって異なります。
"OBJECTNAME" "TABLENAME"	表名または（順序の場合）オブジェクト名を返します。
"OPTYPE"	操作のタイプを返します。次のような結果になります。 INSERT UPDATE DELETE ENSCRIBE COMPUPDATE SQL COMPUPDATE PK UPDATE TRUNCATE 操作が上記のいずれのタイプでもない場合、このファンクションは単語 TYPE およびこのタイプに割り当てられている数字を返します。レコード・タイプの詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』の付録を参照してください。
"RECORDLENGTH"	レコード長（バイト）を返します。

環境値	戻り値
"TRANSACTIONINDICATOR"	<p>トランザクション・インジケータを返します。この値は、Logdump ユーティリティを使用して表示できる、レコード・ヘッダーの TransInD フィールドに相当します（『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください）。</p> <p>次のような結果になります。</p> <ul style="list-style-type: none"> ◆ BEGIN (TransInD が 0、トランザクションの最初のレコードを示します) ◆ MIDDLE (TransInD が 1、トランザクションの中間のレコードを示します) ◆ END (TransInD が 2、トランザクションの最後のレコードを示します) ◆ WHOLE (TransInD が 3、トランザクションの唯一のレコードを示します)

GGFILEHEADER 情報タイプの使用

@GETENV の GGFILEHEADER オプションでは、ファイル・ヘッダーに保持されている Oracle GoldenGate の抽出ファイルまたはトレイル・ファイルの属性を返します。トレイル内のすべてのファイルには、このヘッダーが含まれています。ヘッダーには、ファイル自体、およびファイルが使用されている環境について記述されています。

ファイル・ヘッダーは、データ・レコードに先行する証跡ファイルの先頭部分にレコードとして格納されます。証跡のヘッダーに格納されているレコードに関する情報によって、GoldenGate プロセスは、各レコードが GoldenGate の現行リリースでサポートされる形式であるかどうかを判断できます。

証跡のヘッダー・フィールドは、トークンとして格納されます。トークンの形式は、GoldenGate のすべてのリリースで同じです。あるリリースの GoldenGate でいずれかのトークンがサポートされない場合、そのトークンは無視されます。以前のリリースの GoldenGate との互換性を確保するため、非推奨のトークンにはデフォルト値が割り当てられます。

このオプションは、Replicat プロセスに有効です。GGFILEHEADER と <environment value> は、両方とも二重引用符で囲む必要があります。

注意 データベース、オペレーティング・システム、または Oracle GoldenGate リリースがトークンに関する情報を提供しない場合は、NULL 値が返されます。

構文 @GETENV ("GGFILEHEADER", "<return_value>")

環境値	戻り値
"COMPATIBILITY"	<p>トレイル・ファイルの Oracle GoldenGate 互換性レベルを返します。トレイル・ファイルのデータ・レコードを読み取るためには、現在の Oracle GoldenGate リリースの互換性レベルがトレイル・ファイルの互換性レベル以上である必要があります。現在有効な値は、0 または 1 です。</p> <ul style="list-style-type: none"> ◆ 1 は、トレイル・ファイルが Oracle GoldenGate リリース 10.0 以上（ファイル・バージョン情報を含むファイル・ヘッダーをサポート）であることを示します。 ◆ 0 は、トレイル・ファイルが Oracle GoldenGate リリース 10.0 よりも古いことを示します。これらのリリースでは、ファイル・ヘッダーはサポートされていません。値 0 は、これらの Oracle GoldenGate リリースとの下位互換性を維持するために使用されます。

環境値	戻り値
トレイル・ファイルに関する情報	
"CHARSET"	トレイル・ファイルのグローバル・キャラクタ・セットを返します。次に例を示します。 WCP1252-1
"CREATETIMESTAMP"	トレイルが作成された時間を INIT64 のローカル GMT ユリウス時間で返します。
"FILENAME"	トレイル・ファイル名を返します。ファイル・システムに応じて、フォワード・スラッシュまたはバックワード・スラッシュを使用した絶対パスまたは相対パスの場合があります。
"FILEISTRAIL"	トレイル・ファイルが単一のファイル（バッチ実行で作成されたファイルなど）か、オンライン継続処理用トレイルの順序番号付けされたファイルかを示す、True/false フラグを返します。false の場合、SeqNum サブトークンは無効です。
"FILESEQNO"	トレイル・ファイルの順序番号（先行ゼロを除く）を返します。たとえば、ファイルの順序番号が aa000026 の場合、FILESEQNO は 26 を返します。
"FILESIZE"	トレイル・ファイルのサイズを返します。アクティブ・ファイルの場合は NULL を返し、ファイルが一杯になりトレイルがロール・オーバーされているときはサイズ値を返します。
"FIRSTRECCSN"	トレイル・ファイルの最初のレコードのコミット順序番号（CSN）を返します。トレイル・ファイルが完成するまで、値は NULL です。CSN の詳細は、495 ページの付録 1 を参照してください。
"LASTRECCSN"	トレイル・ファイルの最後のレコードのコミット順序番号（CSN）を返します。トレイル・ファイルが完成するまで、値は NULL です。CSN の詳細は、495 ページの付録 1 を参照してください。
"FIRSTRECIOTIME"	最初のレコードがトレイル・ファイルに書き込まれた時間を返します。トレイル・ファイルが完成するまで、値は NULL です。
"LASTRECIOTIME"	最後のレコードがトレイル・ファイルに書き込まれた時間を返します。トレイル・ファイルが完成するまで、値は NULL です。

環境値	戻り値
"URI"	<p>トレイル・ファイルを作成したプロセスのユニバーサル・リソース識別子を次の形式で返します。</p> <pre><host_name>:<dir>[:<dir>][:<dir_n>]<group_name></pre> <p>条件:</p> <ul style="list-style-type: none"> ◆ host_name は、プロセスをホストするサーバー名です。 ◆ dir は、Oracle GoldenGate インストール・パスのサブディレクトリです。 ◆ group_name は、プロセスにリンクされているプロセス・グループ名です。 <p>例:</p> <pre>sys1:home:oracle:v9.5:extora</pre> <p>トレイルがどこでのプロセスに処理されたかを示します。以前の実行の履歴も含まれます。</p>
"URIHISTORY"	<p>現在のプロセス以前にトレイル・ファイルに書き込んだプロセスの URI のリストを返します。</p> <ul style="list-style-type: none"> ◆ プライマリ Extract の場合、このフィールドは空です。 ◆ データ・ポンプの場合は、このフィールドは URIHistory+ 入力トレイル・ファイルの URI です。
トレイル・ファイルを作成した Oracle GoldenGate プロセスに関する情報	
"GROUPNAME"	<p>トレイルを作成した Extract グループに関連付けられているグループ名を返します。グループ名は、ADD EXTRACT コマンドで指定された名前です。たとえば、"ggext" です。</p>
"DATASOURCE"	<p>プロセスによって読み取られたデータ・ソースを返します。次のいずれかになります。</p> <ul style="list-style-type: none"> ◆ DS_EXTRACT_TRAILS (ソースは、変更データが移入された Oracle GoldenGate 抽出ファイルです) ◆ DS_DATABASE (ソースは SOURCEISTABLE 主導の初期ロードで使用され、直接選択されてトレイルに書き込まれたデータベース表です) ◆ DS_TRAN_LOGS (ソースはデータベース・トランザクション・ログです) ◆ DS_INITIAL_DATA_LOAD (ソースは Extract で、データはソース表から直接取得されました) ◆ DS_VAM_EXTRACT (ソースは Vendor Access Module です) ◆ DS_VAM_TWO_PHASE_COMMIT (ソースは VAM トレイルです)
"GGMAJORVERSION"	<p>トレイルを作成した Extract プロセスのメジャー・バージョン (整数で表現) を返します。たとえば、バージョン 1.2.3 の場合は 1 を返します。</p>
"GGMINORVERSION"	<p>トレイルを作成した Extract プロセスのマイナー・バージョン (整数で表現) を返します。たとえば、バージョン 1.2.3 の場合は 2 を返します。</p>
"GGVERSIONSTRING"	<p>トレイルを作成した Extract プロセスのメンテナンス (またはパッチ) レベル (整数で表現) を返します。たとえば、バージョン 1.2.3 の場合は 3 を返します。</p>

環境値	戻り値
"GGMAINTENANCELEVEL"	プロセスのメンテナンス・バージョン (xx.xx.xx) を返します。
"GGBUGFIXLEVEL"	プロセスのパッチ・バージョン (xx.xx.xx.x) を返します。
"GGBUILDNUMBER"	プロセスのビルド番号を返します。
トレイル・ファイルのローカル・ホストに関する情報	
"HOSTNAME"	<p>トレイルに書き込んだ Extract が実行されているマシンの DNS 名を返します。次に例を示します。</p> <ul style="list-style-type: none"> ◆ sysa ◆ sysb ◆ paris ◆ hq25
"OSVERSION"	<p>トレイルに書き込んだ Extract が実行されているマシンのオペレーティング・システムのメジャー・バージョンを返します。次に例を示します。</p> <ul style="list-style-type: none"> ◆ Versions10_69 ◆ #1 SMP Fri Feb 24 16:56:28 EST 2006 ◆ 5.00.2195 Service Pack 4
"OSRELEASE"	<p>トレイルに書き込んだ Extract が実行されているマシンのオペレーティング・システムのリリース・バージョンを返します。たとえば、OSVERSION で示した例のリリース・バージョンは次のようになることがあります。</p> <ul style="list-style-type: none"> ◆ 5.10 ◆ 2.6.9-34.ELsmp ◆ 2000 Advanced Server
"OSTYPE"	<p>トレイルに書き込んだ Extract が実行されているマシンのオペレーティング・システムのタイプを返します。次に例を示します。</p> <ul style="list-style-type: none"> ◆ SunOS ◆ Linux ◆ Microsoft Windows
"HARDWARETYPE"	<p>トレイルに書き込んだ Extract が実行されているマシンのハードウェアのタイプを返します。次に例を示します。</p> <ul style="list-style-type: none"> ◆ sun4u ◆ x86_64 ◆ x86
トレイル・ファイルのデータを生成したデータベースに関する情報	
"DBNAME"	データベース名 (findb など) を返します。
"DBINSTANCE"	データベースのタイプに適切な場合、データベース・インスタンス名 (ORA1022A など) を返します。

環境値	戻り値
"DBTYPE"	トレイル・ファイルのデータを生成したデータベースのタイプを返します。 次のいずれかになります。 DB2 UDB DB2 ZOS CTREE INGRES MSSQL MYSQL ORACLE SQLMX SYBASE TERADATA NONSTOP ENSCRIBE ODBC
"DBCHARSET"	トレイル・ファイルのデータを生成したデータベースで使用されているキャラクタ・セットを返します。（一部のデータベースでは、これは空になります。）
"DBMAJORVERSION"	トレイル・ファイルのデータを生成したデータベースのメジャー・バージョンを返します。
"DBMINORVERSION"	トレイル・ファイルのデータを生成したデータベースのマイナー・バージョンを返します。
"DBVERSIONSTRING"	トレイル・ファイルのデータを生成したデータベースのメンテナンス（パッチ）・レベルを返します。
"DBCLIENTCHARSET"	データベース・クライアントに使用されているキャラクタ・セットを返します。
"DBCLIENTVERSIONSTRING"	データベース・クライアントのメンテナンス（パッチ）・レベルを返します。 （一部のデータベースでは、これは空になります。）
以前のトレイル・ファイルから繰り越されるリカバリ情報	
"RECOVERYMODE"	内部で Oracle GoldenGate に使用されるリカバリ情報を返します。
"LASTCOMPLETECSN"	内部で Oracle GoldenGate に使用されるリカバリ情報を返します。
"LASTCOMPLETEXIDS"	内部で Oracle GoldenGate に使用されるリカバリ情報を返します。
"LASTCSN"	内部で Oracle GoldenGate に使用されるリカバリ情報を返します。
"LASTXID"	内部で Oracle GoldenGate に使用されるリカバリ情報を返します。
"LASTCSNTS"	内部で Oracle GoldenGate に使用されるリカバリ情報を返します。

RECORD 情報タイプの使用

@GETENV の RECORD オプションでは、Oracle GoldenGate トレイル・ファイルのレコードの場所を返します。このファンクションは、ファイルの順序番号、およびそのファイルの相対バイト・アドレスを返すことができます。この 2 つの値から、特定のレコードに関連する一意の値が得られます。

このオプションは、Extract データ・ポンプおよび Replicat プロセスに有効です。RECORD と <environment value> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("RECORD", "<environment value>")

環境値	戻り値
"FILESEQNO"	トレイル・ファイルの順序番号（先行ゼロを除く）を返します。
"FILERBA"	FILESEQNO ファイル内のレコードの相対バイト・アドレスを返します。

DBENVIRONMENT 情報タイプの使用

@GETENV の DBENVIRONMENT オプションでは、データベースのグローバル環境情報を返します。このオプションは、Extract および Replicat プロセスに有効です。DBENVIRONMENT と <environment value> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("DBENVIRONMENT", "<return value>")

環境値	戻り値
"DBNAME"	データベース名を返します。
"DBVERSION"	データベース・バージョンを返します。
"DBUSER"	データベース・ログイン・ユーザーを返します。
"SERVERNAME"	サーバー名を返します。

TRANSACTION 情報タイプの使用

@GETENV の TRANSACTION オプションでは、ソース・トランザクションに関する情報を返します。このオプションは、Extract プロセスに有効です。TRANSACTION と <environment value> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("TRANSACTION", "<return value>")

環境値	戻り値
"TRANSACTIONID" "XID"	トランザクション ID 番号を返します。トランザクション ID および CSN は、各トランザクションの最初のレコードに関連付けられており、トレイル・レコードにトークンとして保持されています。各トランザクション ID には、CSN が関連付けられています。トランザクション ID トークンは、相対値として評価されることはないため、どのプラットフォームでもゼロは埋め込まれません。これらは、一致するかしないかのみでの評価が行われます。トレイルでは、トランザクション ID トークンは TRANID として表示されることに注意してください。
"CSN"	<p>コミット順序番号 (CSN) を返します。CSN は、Oracle、DB2 LUW、および DB2 z/OS データベースに対して返される場合は、ゼロが埋め込まれません。サポートされている他のすべてのデータベースの場合は、CSN にゼロが埋め込まれます。Sybase CSN の場合、ドット (.) で区切られている各サブ文字列は、そのサブ文字列が変更されない長さまでゼロが埋め込まれます。</p> <p>トレイルでは、CSN トークンは LOGCSN として表示されることに注意してください。CSN トークンの補足情報は、TRANSACTIONID XID 環境値を参照してください。</p> <p>CSN 自体の詳細は、495 ページの付録 1 を参照してください。</p>
"TIMESTAMP"	トランザクションのコミット・タイムスタンプを返します。
"NAME"	使用可能な場合、トランザクション名を返します。
"USERID"	(Oracle) 最後のトランザクションをコミットしたデータベース・ユーザーの Oracle ユーザー ID を返します。
"USERNAME"	(Oracle) 最後のトランザクションをコミットしたデータベース・ユーザーの Oracle ユーザー名を返します。
"RSN"	レコード順序番号を返します。
"PLANNAME"	(z/OS 上の DB2) 現在のトランザクションを最初に実行した計画名を返します。計画名は、リカバリ・ログ・レコードの開始ユニットに含まれています。

OSVARIABLE 情報タイプの使用

@GETENV の OSVARIABLE オプションでは、指定されたオペレーティングシステム環境変数の文字列値を返します。このオプションは、**Extract** および **Replicat** に有効です。OSVARIABLE と <variable> は、両方とも二重引用符で囲む必要があります。

構文 @GETENV ("OSVARIABLE", "<variable>")

環境値	戻り値
"<variable>"	<p>変数名。指定された変数と正確に一致するものが検索されます。たとえば、UNIX の grep コマンドでは、次のすべての変数が返されますが、@GETENV ("OSVARIABLE", "HOME") では HOME の値のみが返されます。</p> <pre> ANT_HOME=/usr/local/ant JAVA_HOME=/usr/java/j2sdk1.4.2_10 HOME=/home/judyd ORACLE_HOME=/rdbms/oracle/ora1022i/64 </pre> <p>オペレーティング・システムが大 / 小文字区別をサポートしている場合、検索では大 / 小文字が区別されます。</p>

TLFKEY 情報タイプの使用

@GETENV の TLFKEY オプションでは、一意キーと ACI の Base24 アプリケーションの TLF/PTLF レコードを関連付けます。64 ビット・キーは、次のアイテムが連結されて構成されます。

- 2000 年以降の秒数。
- TLF/PTLF ブロックのレコードのブロック数に 10 を掛けた値。
- ユーザーによって指定されたノード (0 ～ 255 である必要があります)。

このオプションは、**Extract** および **Replicat** プロセスに有効です。TLFKEY は二重引用符で囲む必要があります。

構文 @GETENV ("TLFKEY", SYSKEY <unique key>)

環境値	戻り値
<unique key>	<p>ソース TLF/PTLF ファイルの NonStop ノード番号。</p> <p>例：</p> <pre> GETENV ("TLFKEY", SYSKEY, 27) </pre>

GETVAL

@GETVAL ファンクションでは、ストアド・プロシージャまたは問合せから値を抽出し、MAP または TABLE 文の FILTER または COLMAP 句の入力として使用できるようにします。

@GETVAL を使用してパラメータ値を抽出できるかどうかは、次のことに依存します。

- 1. ストアド・プロシージャまたは問合せの実行の成否。
- 2. ストアド・プロシージャまたは問合せの結果の失効の有無。

行方不明列値の処理

値を抽出できない場合、@GETVAL ファンクションの結果は " 列行方不明 " になります。一般的にこのような状態は、データベースが変更されたログのみを記録する場合に更新操作で発生します。

通常、これは列をマッピングできないことを意味します。行方不明の列値をテストするには、@COLTEST ファンクションを使用して @GETVAL の結果をテストした後、必要な場合には、行方不明の値を補うために、列に代替値をマップします。または、列値を使用可能にするために、TABLE または MAP パラメータの FETCHCOLS または FETCHCOLSEXCEPT オプションを使用して、値がログに存在しない場合にデータベースから値をフェッチさせます。必要な列のサブリメンタル・ロギングを有効にする方法も有益です。

構文 @GETVAL (<name>.<parameter>)

引数	説明
<name>	<p>ストアド・プロシージャまたは問合せ名。SQLEXEC を使用してプロシージャまたは問合せを使用するときは、次の値が有効です。</p> <p>問合せ：SQLEXEC 句の ID オプションで指定されている論理名を使用します。ID は、問合せで必須の SQLEXEC 引数です。</p> <p>ストアド・プロシージャ：TABLE または MAP 文内でプロシージャを実行する回数に応じて、次のいずれかを使用します。</p> <ul style="list-style-type: none">◆ 複数回実行するときは、SQLEXEC 文の ID 句で定義されている論理名を使用します。ID は、1 つのプロシージャを複数回実行するときに必須です。◆ 1 回実行するときは、ストアド・プロシージャの実際の名前を使用します。
<パラメータ>	<p>有効な値は、次のいずれかです。</p> <ul style="list-style-type: none">◆ データを抽出し、列マップに渡すストアド・プロシージャまたは問合せ内のパラメータ名。◆ ストアド・プロシージャまたは問合せが返す値を抽出するときは、RETURN_VALUE。

例 1 次の例では、各 MAP 文が @GETVAL ファンクション内で論理名 lookup1 および lookup2 を参照することによって、ストアド・プロシージャ lookup をコールし、各結果セットを適切に参照できるようにします。

```
MAP schema.srctab, TARGET schema.targtab,
SQLEXEC (SPNAME lookup, ID lookup1, PARAMS (param1 = srccol)),
COLMAP (targcol1 = @GETVAL (lookup1.param2));
MAP schema.srctab, TARGET schema.targtab2,
SQLEXEC (SPNAME lookup, ID lookup2, PARAMS (param1 = srccol)),
COLMAP (targcol2= @GETVAL (lookup2.param2));
```

例 2 次の例では、ストアド・プロシージャ `lookup` を 1 回実行します。このケースでは、実際のプロシージャ名を使用します。論理名は必要ありません。

```
MAP schema.tab1, TARGET schema.tab2,  
SQLEXEC (SPNAME lookup, PARAMS (param1 = srccol)),  
COLMAP (targcol = @GETVAL (lookup.param1));
```

例 3 次の例では、問合せを実行し、取得した値を `@GETVAL` を使用してマップします。

```
MAP sales.account, TARGET sales.newacct,  
SQLEXEC (ID lookup,  
QUERY " select desc_col into desc_param from lookup_table "  
" where code_col = :code_param ",  
PARAMS (code_param = account_code)),  
COLMAP (newacct_id = account_id, newacct_val = @GETVAL (lookup.desc_param));
```

代替構文

`SQLEXEC` では、`@GETVAL` キーワードを明示的に使用せずにパラメータ結果を取得できます。次の形式で、プロシージャ名（問い合わせまたはプロシージャの複数のインスタンスを使用する場合は論理名）およびパラメータを参照します。

構文 {<procedure name> | <logical name>}.<parameter>

例 1 次の例では、`@GETVAL` キーワードを使用せずに、`@GETVAL` が暗黙的に `proc1.p2` で呼び出されます。

```
MAP test.tab1, TARGET test.tab2,  
SQLEXEC (SPNAME proc1, ID myproc, PARAMS (p1 = sourcecol1)),  
COLMAP (targcol1 = proc1.p2);
```

例 2 次の例では、`@GETVAL` キーワードを使用せずに、`@GETVAL` が暗黙的に `lookup.desc_param` で呼び出されます。

```
MAP sales.account, TARGET sales.newacct,  
SQLEXEC (ID lookup,  
QUERY " select desc_col into desc_param from lookup_table "  
" where code_col = :code_param ",  
PARAMS (code_param = account_code)),  
COLMAP (newacct_id = account_id, newacct_val = lookup.desc_param);
```

HEXTOBIN

`@HEXTOBIN` ファンクションでは、指定された 16 進数データ文字列を `RAW` 形式に変換します。

構文 `@HEXTOBIN (<data>)`

引数	説明
<データ>	ソース列名、式、または引用符で囲まれたリテラル文字列。

例 `@HEXTOBIN ("414243")` は、3 バイト (`0x41 0x42 0x43`) に変換されます。

HIGHVAL | LOWVAL

@HIGHVAL および @LOWVAL ファンクションは、値を生成する必要があり、その値を上限または下限内で抑制する場合に使用します。これらのファンクションは、同一の名前の COBOL ファンクションをエミュレートします。

@HIGHVAL および @LOWVAL は、文字列およびバイナリ・データ型とのみ使用します。文字列と使用する場合は、@STRNCMP のみ有効です。10 進数または日付データ型、もしくは SQLEXEC 操作とともに使用すると、エラーを引き起こすことがあります。DOUBLE データ型では、-1 または 0 (Oracle NUMBER の精度指定なし、位取り指定なし) になります。

構文 @HIGHVAL ([<length>]) | @LOWVAL ([<length>])

引数	説明
<length>	オプションです。バイナリ出力長 (バイト) を指定します。<length> の最大値は、ターゲット列長です。

例 次の例では、group_level 列のサイズを 5 バイトとみなします。

ファンクション文	結果
group_level = @HIGHVAL()	{0xFF, 0xFF, 0xFF, 0xFF, 0xFF}
group_level = @LOWVAL()	{0x00, 0x00, 0x00, 0x00, 0x00}
group_level = @HIGHVAL(3)	{0xFF, 0xFF, 0xFF}
group_level = @LOWVAL(3)	{0x00, 0x00, 0x00}

IF

@IF ファンクションでは、条件に基づいて 2 つの値のうち 1 つを返します。@IF ファンクションは、他の Oracle GoldenGate ファンクションとともに使用して、1 つ以上の例外条件をテストする条件引数を開始できます。テストの結果に基づいて処理を実行するように指示できます。@IF 文は、必要に応じてネストできます。

構文 @IF (<conditional expression>, <value if non-zero>, <value if zero>)

引数	説明
<conditional expression>	有効な条件式または Oracle GoldenGate ファンクション。数値演算子 (=、>、< など) は、数値比較でのみ使用します。文字比較では、文字比較ファンクションの 1 つを使用します。
<value if non-zero>	非ゼロを true とみなします。
<value if zero>	ゼロ (0) を false とみなします。

例 1 次の例では、AMT 列がゼロより大きい場合にのみ量が返され、それ以外の場合はゼロが返されます。
AMOUNT_COL = @IF (AMT > 0, AMT, 0)

例 2 次の例では、STATE 列が CA、AZ、または NV の場合に WEST が返され、それ以外の場合は EAST が返されます。

```
REGION = @IF (@VALONEOF (STATE, "CA", "AZ", "NV"), "WEST", "EAST")
```

例 3 次の例では、2 つの列が両方ともゼロより大きい場合に、PRICE 列に QUANTITY 列を掛けた結果が返されます。それ以外の場合は、@COLSTAT(NULL) ファンクションによって、ターゲット列に NULL 値が生成されます。

```
ORDER_TOTAL = @IF (PRICE > 0 AND QUANTITY > 0, PRICE * QUANTITY,
@COLSTAT (NULL))
```

NUMBIN

@NUMBIN ファンクションでは、8 バイト以下のバイナリ文字列を数字に変換します。このファンクションは、ソース列に定義されているバイト・ストリームが、実際には文字列として表現されている数字のときに使用します。

構文 @NUMBIN (<source column>)

引数	説明
<source column>	変換する文字列を含むソース列名。

例 次の例では、@NUMBIN と @DATE を組み合わせて、48 ビット Tandem 列を 64 ビットのローカル時間のユリウス値に変換します。

```
DATE = @DATE ("JTSLCT", "TTS" @NUMBIN (DATE))
```

NUMSTR

@NUMSTR ファンクションでは、文字列（文字）用の列または値を数字に変換します。@NUMSTR では、次のいずれかの操作を行います。

- 文字列（文字）を数字にマップする。
- 数字のみを含む文字列用の列を算術式で使用する。

構文 @NUMSTR (<input>)

引数	説明
<入力>	次のいずれかを指定できます。 <ul style="list-style-type: none"> ◆ 文字用の列名。 ◆ 引用符で囲まれたリテラル文字列。

例 PAGE_NUM = @NUMSTR (ALPHA_PAGE_NO)

RANGE

2 つ以上の Oracle GoldenGate プロセス間で表の行を分割するには、@RANGE 関数を使用します。大きく頻繁にアクセスされる表のスループットを向上するのに使用でき、また、異なる送信先にデータ・セットを分配するのに使用できます。TABLE または MAP 文の FILTER 句で各範囲を指定します。

@RANGE には安全性および拡張性があります。同じ行が常に同じプロセス・グループで処理されるので、データの整合性が保持されます。

@RANGE は、入力で指定された列のハッシュ値を計算します。列が指定されない場合、TABLE または MAP 文に KEYCOLS 句が指定されていれば、この列を使用します。それ以外の場合は、主キー列を使用します。

Oracle GoldenGate は、指定された範囲数で均等に分散できるように、合計範囲数を調整します。

このファンクションには任意の列を指定できるため、相互に関係制約を持つ表の行は、参照整合性を維持するために、同一のプロセスまたはトレイルにグループ化する必要があります。

注意 範囲は、Extract を使用して計算するほうが、Replicat を使用するよりも効率的です。ターゲット側で範囲を計算するには、Replicat がトレイル全体を読み取り、各範囲の指定を満たすデータを見つける必要があります。

構文 @RANGE (<range>, <total ranges> [, <column>] [, <column>] [, ...])

引数	説明
<range>	指定のプロセスまたはトレイルに割り当てる範囲。<total ranges> で定義する値を最大値として、1、2、3 のように指定します。
<total ranges>	割り当てる範囲の合計数 たとえば、データを 3 つのグループに分割するには、値 3 を使用します。
<column>	範囲割当ての基準にする列名。この引数はオプションです。使用されない場合、Oracle GoldenGate は表の主キーに基づいて範囲を割り当てます。

例 1 次の例では、レプリケーション・ワークロードは、ソース表 acct の ID 列に基づいて、3 つの範囲（3 つの Replicat プロセス間）に分割されます。

 (Replicat グループ 1 のパラメータ・ファイル)

MAP sales.acct, TARGET sales.acct, FILTER (@RANGE (1, 3, ID));

 (Replicat グループ 2 のパラメータ・ファイル)

MAP sales.acct, TARGET sales.acct, FILTER (@RANGE (2, 3, ID));

 (Replicat グループ 3 のパラメータ・ファイル)

MAP sales.acct, TARGET sales.acct, FILTER (@RANGE (3, 3, ID));

例 2 次の例では、1 つの Extract プロセスが処理ロードを 2 つのトレイルに分割します。範囲計算の基準にする列が定義されていないため、Oracle GoldenGate は主キー列を使用します。

RMTTRAIL /ggs/dirdat/aa

TABLE fin.account, FILTER (@RANGE (1, 2));

RMTTRAIL /ggs/dirdat/bb

TABLE fin.account, FILTER (@RANGE (2, 2));

例 3 次の例では、2 つの表で order_ID 列に基づく関連操作が行われます。order_master 表は、キー order_ID を持ち、order_detail 表は、キー order_ID および item_number を持ちます。キー order_ID によって関係が確立されるため、参照整合性を維持するために、両方の表の @RANGE フィルタでこのキーが使用されます。ロードは 2 つの範囲に分割されます。

```
(パラメータ・ファイル #1)

MAP sales.order_master, TARGET sales.order_master,
FILTER (@RANGE (1, 2, order_ID));
MAP sales.order_detail, TARGET sales.order_detail,
FILTER (@RANGE (1, 2, order_ID));
```

```
(パラメータ・ファイル #2)

MAP sales.order_master, TARGET sales.order_master,
FILTER (@RANGE (2, 2, order_ID));
MAP sales.order_detail, TARGET sales.order_detail,
FILTER (@RANGE (2, 2, order_ID));
```

STRCAT

@STRCAT ファンクションでは、1 つ以上の文字列または文字列（文字）用の列を連結します。リテラル文字列は引用符で囲みます。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRCAT (<string1>, <string2> [, ...])

引数	説明
<string1>	連結する最初の列またはリテラル文字列。
<string2>	連結する次の列またはリテラル文字列。

例 次の例では、3 つの列から電話番号を作成し、リテラル・フォーマット値を含めます。

```
PHONE_NO = @STRCAT (AREA_CODE, PREFIX, "-", PHONE)
```

STRCMP

@STRCMP ファンクションでは、2 つの文字用の列またはリテラル文字列を比較します。リテラルは引用符で囲みます。

@STRCMP は、次を返します。

- -1 (最初の文字列が 2 つ目の文字列より小さい場合)
- 0 (2 つの文字列が等しい場合)
- -1 (最初の文字列が 2 つ目の文字列より大きい場合)

文字列の比較の前に、末尾の空白が切り捨てられます。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。このファンクションは、CHAR と NCHAR など、異なる文字データ型を比較できます。詳細は、393 ページを参照してください。

構文 @STRCMP (<string1>, <string2>)

引数	説明
<string1>	比較する最初の列またはリテラル文字列。
<string2>	比較する 2 つ目の列またはリテラル文字列。

例 次の例では、2 つのリテラル文字列を比較し、最初の文字列が 2 つ目の文字列より大きいために 1 が返されます。

```
@STRNCMP ("JOHNSON", "JONES")
```

STREQ

@STREQ ファンクションでは、2 つの文字列（文字）用の列またはリテラル文字列が等しいかどうかを確認します。リテラルは引用符で囲みます。@STREQ は、次を返します。

- 1 (true) (2 つの文字列が等しい場合)
- 0 (false) (2 つの文字列が等しくない場合)

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。このファンクションは、CHAR と NCHAR など、異なる文字データ型を比較できます。詳細は、393 ページを参照してください。

構文 @STREQ (<string1>, <string2>)

引数	説明
<string1>	比較する最初の列またはリテラル文字列。
<string2>	比較する 2 つ目の列またはリテラル文字列。

例 次の例では、`region` 列の値と、リテラル値 `"EAST"` を比較します。`region = EAST` の場合、レコードはフィルタに渡されます。

```
FILTER (@STREQ (region, "EAST"))
```

次の例に示すように、`@STREQ` は結果を決定するための比較にも使用できます。`state` が `"NY"` の場合、式は `"East Coast"` を返します。それ以外の場合は、`"Other"` を返します。

```
@IF (@STREQ (state, "NY"), "East Coast", "Other")
```

STREXT

`@STREXT` ファンクションでは、文字列の一部を抽出します。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が **Unicode** で指定される場合、ターゲット列は **SQL Unicode** データ型である必要があります。詳細は、393 ページを参照してください。

構文 `@STREXT (<string>, <begin position>, <end position>)`

引数	説明
<string>	抽出元の文字列。この文字列には、文字の列の名前またはリテラル文字列を指定できます。リテラルは引用符で囲みます。
<begin position>	抽出を開始する文字の位置。
<end position>	抽出を終了する文字の位置。終了位置は抽出に含まれます。

次の例では、`@STREXT` ファンクションを使用して、電話番号を 3 つの異なる列に抽出します。

```
AREA_CODE = @STREXT (PHONE, 1, 3),  
PREFIX = @STREXT (PHONE, 4, 6),  
PHONE_NO = @STREXT (PHONE, 7, 10)
```

STRFIND

@STRFIND ファンクションでは、文字列用の列内の文字列の位置を確認し、文字列が見つからないときはゼロを返します。オプションで、@STRFIND は文字列内の開始位置を受け付けます。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRFIND (<string>, "search string" [, < begin position>])

引数	説明
<string>	検索元の文字列。ここには、文字用の列名またはリテラル文字列を指定できます。リテラルは引用符で囲みます。
"<search string>"	検索する文字列。検索する文字列は引用符で囲みます。
<begin position>	検索を開始する文字の位置。

例 ACCT 列の文字列を ABC123ABC とすると、次のような結果になります。

ファンクション文	結果
@STRFIND (ACCT, "23")	5
@STRFIND (ACCT, "ZZ")	0
@STRFIND (ACCT, "ABC", 2)	7 (検索が 2 番目の文字から開始されたため)

STRLEN

@STRLEN ファンクションでは、文字数で表された文字列長を返します。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRLEN (<string>)

引数	説明
<string>	文字列（文字）用の列名、またはリテラル文字列。リテラルは引用符で囲みます。

例 @STRLEN (ID_NO)

STRLTRIM

@STRLTRIM ファンクションでは、先行する空白を切り捨てます。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRLTRIM (<string>)

引数	説明
<string>	文字用の列名、またはリテラル文字列。リテラルは引用符で囲みます。

例 birth_state = @strltrim (state)

STRNCAT

@STRCAT ファンクションでは、1 つ以上の文字列を最大長まで連結します。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRNCAT (<string>, <max length> [, <string>, <max length>] [, ...])

引数	説明
<string>	文字列（文字）用の列名、またはリテラル文字列。リテラルは引用符で囲みます。
<max length>	最大文字列長（文字数）。

例 次の例では、2 つの文字列を連結し、結果が "ABC123" になります。

```
PHONE_NO = @STRNCAT ("ABCDEF", 3, "123456", 3)
```

STRNCMP

@STRNCMP ファンクションでは、特定の文字数に基づいて 2 つの文字列を比較します。文字列には、文字列（文字）用の列名、または引用符で囲んだりテラル文字列を指定できます。比較は、文字列の最初の文字から開始されます。

@STRNCMP は、次を返します。

- -1（最初の文字列が 2 つ目の文字列より小さい場合）
- 0（2 つの文字列が等しい場合）
- -1（最初の文字列が 2 つ目の文字列より大きい場合）

構文 @STRNCMP (<string1>, <string2>, <max length>)

引数	説明
<string1>	比較する最初の文字列。
<string2>	比較する 2 つ目の文字列。
<max length>	文字列内の比較する最大文字数。

例 次の例では、<max length> で 2 が指定されているため、各文字列の最初の 2 文字を比較し、2 つの文字セットが同一なので 0 を返します。

```
@STRNCMP ("JOHNSON", "JONES", 2)
```

STRNUM

@STRNUM ファンクションでは、数字を文字列に変換し、出力形式および埋込みを指定します。

構文 @STRNUM (<column>, {LEFT | LEFTSPACE, | RIGHT | RIGHTZERO} [<length>])

引数	説明
<column>	ソース数字列名。
LEFT	左に寄せます（埋込みなし）。
LEFTSPACE	左に寄せ、ターゲット列の残りを空白で埋めます。
RIGHT	右に寄せ、ターゲット列の残りを空白で埋めます。列の値が負の値の場合、空白はマイナス記号の前に追加されます。たとえば、ターゲット列で 7 桁が許可されているとすると、列値 -1.27 に strnum(Col1, right) を使用した場合の結果は、###-1.27 になります。マイナス記号は桁数にカウントされませんが、10 進数はカウントされます。
RIGHTZERO	右に寄せ、ターゲット列の残りをゼロで埋めます。列の値が負の値の場合、ゼロはマイナス記号の後ろで数値の前に追加されます。たとえば、ターゲット列で 7 桁が許可されているとすると、列値 -1.27 に strnum(Col1, rightzero) を使用した場合の結果は、-0001.27 になります。マイナス記号は桁数にカウントされませんが、10 進数はカウントされます。
<length>	埋込みを指定するオプション（LEFT 以外）を使用しているときに、出力長を指定します。次に例を示します。 <ul style="list-style-type: none">◆ 列値 -1.27 に strnum(Col1, right, 6) を使用した場合の結果は、##-1.27 になります。マイナス記号は桁数にカウントされませんが、10 進数はカウントされます。◆ 列値 -1.27 に strnum(Col1, rightzero, 6) が使用されている場合は、-001.27 になります。マイナス記号は桁数にカウントされませんが、10 進数はカウントされます。

例 次に、ソース列 NUM の値が 15 で、ターゲット列の最大長が 5 文字の場合に、フォーマット・オプションを使用して様々な結果を取得する例を示します。

ファンクション文	結果（# は空白を表します）
CHAR1 = @STRNUM (NUM, LEFT)	15
CHAR1 = @STRNUM (NUM, LEFTSPACE)	15###
CHAR1 = @STRNUM (NUM, RIGHTZERO)	00015
CHAR1 = @STRNUM (NUM, RIGHT)	###15

上記の例で出力の <length> を 4 とすると、次のように異なるタイプの結果が返されます。

ファンクション文	結果（# は空白を表します）
CHAR1 = @STRNUM (NUM, LEFTSPACE, 4)	15##
CHAR1 = @STRNUM (NUM, RIGHTZERO, 4)	0015
CHAR1 = @STRNUM (NUM, RIGHT, 4)	##15

STRRTRIM

@STRRTRIM ファンクションでは、末尾の空白を切り捨てます。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRRTRIM (<string>)

引数	説明
<string>	文字用の列名、またはリテラル文字列。リテラルは引用符で囲みます。

例 street_address = @strrtrim (address)

STRSUB

@STRSUB ファンクションでは、文字列（文字）用の列または定数内の文字を置換します。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRSUB
 (<source string>, <search string>, <substitute string>
 [, <search string>, <substitute string>] [, ...])

引数	説明
<source string>	置換される文字を含むソース文字列または列。
<search string>	置換される文字列。
<substitute string>	検索した文字列と置換する文字列。

例 1 次の例では、xxABCxx が返されます。
 @STRSUB ("123ABC123", "123", "xx")

例 2 次の例では、023zBC023 が返されます。
 @STRSUB ("123ABC123", "A", "z", "1", "0")

STRTRIM

@STRTRIM ファンクションでは、先行する空白と末尾の空白を切り捨てます。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRTRIM (<string>)

引数	説明
<string>	文字用の列名、またはリテラル文字列。リテラルは引用符で囲みます。

例 pin_no = @strtrim (custpin)

STRUP

@STRUP ファンクションでは、英数字文字列または文字列（文字）用の列を大文字に変換します。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @STRUP (<string>)

引数	説明
<string>	文字用の列名、またはリテラル文字列。リテラルは引用符で囲みます。

例 次の例では、"SALESPERSON" が返されます。

@STRUP ("salesperson")

TOKEN

@TOKEN ファンクションでは、Oracle GoldenGate レコード・ヘッダーのユーザー・トークン・エリアに保持されているトークン・データを取得します。トークン・データをターゲット列にマップするには、COLMAP 句のソース式で @TOKEN を使用します。そのかわりの方法としては、@TOKEN を SQLEXEC 文、Oracle GoldenGate マクロ、またはユーザー・イグジットで使用できます。

トークン・データを定義するには、Extract パラメータ・ファイルの TABLE 文で TOKENS 句を使用します。トークンの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

構文

@TOKEN ("`<token name>`")

引数	説明
<code>"<token name>"</code>	引用符で囲まれた、データを取得するトークン名。

例

次の例では、10 のトークンをターゲット列にマップします。

```
MAP ora.oratest, TARGET ora.rpt,
COLMAP (
  host = @token ("tk_host"),
  gg_group = @token ("tk_group"),
  osuser = @token ("tk_osuser"),
  domain = @token ("tk_domain"),
  ba_ind = @token ("tk_ba_ind"),
  commit_ts = @token ("tk_commit_ts"),
  pos = @token ("tk_pos"),
  rba = @token ("tk_rba"),
  tablename = @token ("tk_table"),
  optype = @token ("tk_optype")
);
```

VALONEOF

@VALONEOF ファンクションでは、文字列または文字列（文字）用の列を値のリストと比較します。値または列がリストにある場合は 1 が返され、それ以外の場合は 0 が返されます。

Oracle GoldenGate では、このファンクションで、Unicode、および Microsoft Windows、UNIX、Linux オペレーティング・システムのネイティブ・エンコーディングの文字列を格納する列内の文字を表すために、エスケープ・シーケンスの使用をサポートしています。引数が Unicode で指定される場合、ターゲット列は SQL Unicode データ型である必要があります。詳細は、393 ページを参照してください。

構文 @VALONEOF (<expression>, <value> [, <value>] [, ...])

引数	説明
<expression>	文字用の列名、または引用符で囲まれたリテラル文字列。
<value>	基準値。

例 次の例では、STATE が CA または NY の場合、値が非ゼロ（true）のときに @IF によって返されるレスポンスである "COAST" が返されます。それ以外の場合は、"MIDDLE" が返されます。

 @IF (@VALONEOF (STATE, "CA", "NY"), "COAST", "MIDDLE")

第 5 章

ユーザー・イグジット・ファンクション

.....

この章では、Oracle GoldenGate ユーザー・イグジット・ファンクションとその構文について説明します。Oracle GoldenGate ユーザー・イグジットの使用の詳細は、『Oracle GoldenGate Windows and UNIX 管理者ガイド』を参照してください。

ユーザー・イグジットのコール

Oracle GoldenGate 処理内の定義済 exit ポイントで、Windows DLL または UNIX 共有オブジェクトから C プログラミング・コードで記述されたカスタム exit ルーチン呼び出すには、CUSEREXIT パラメータを使用します。ユーザーの exit ルーチンは、Extract および Replicat プロセスからの様々なイベントや情報の受信、必要に応じて情報の処理、およびコール元（それを呼び出した Oracle GoldenGate プロセス）へのレスポンスや情報の返信が可能である必要があります。

CUSEREXIT パラメータの情報および構文は、142 ページを参照してください。

インストールされるユーザー・イグジット・ファイルについて

usrdecs.h ファイルについて

usrdecs.h ファイルは、ユーザー・イグジット API のインクルード・ファイルです。タイプ定義、返されるステータス値、コールバック・ファンクション・コード、その他の多くの定義が含まれています。usrdecs.h ファイルは、Oracle GoldenGate ディレクトリにインストールされます。このファイルは変更しないでください。

ユーザー・イグジットのアップグレード

usrdecs.h ファイルはバージョンングされるので、新機能や構造の変更などの機能強化やアップグレードが新しい Oracle GoldenGate リリースに追加されたときに既存のユーザー・イグジットとの下位互換性が保証されます。usrdecs.h ファイルのバージョンは、Replicat または Extract の起動時にレポート・ファイルに出力されます。

新しいユーザー・イグジット機能を使用するには、新しい usrdecs ファイルを含めるようにルーチンを再コンパイルする必要があります。新しい機能を使用しないルーチンの場合は、再コンパイルする必要はありません。

.....

ユーザー・イグジット・ファンクションの使用例の表示

Oracle GoldenGate では、次のサンプル・ユーザー・イグジット・ファイルが Oracle GoldenGate インストール・ディレクトリの UserExitExamples ディレクトリ内にインストールされます。

- `exitdemo.c` では、ユーザー・イグジットの初期化、あるイグジット・ポイントにおけるコールバックの実行およびデータの変更方法が示されます。このデモは、データベース・タイプに特定されません。
- `exitdemo_passthru.c` では、Extract データ・ポンプにおける CUSEREXIT パラメータの PASSTHRU オプションの使用方法が示されます。
- `exitdemo_more_recs.c` では、同じ入力レコードを複数回使用して、いくつかのターゲット・レコードを生成する方法の例が示されます。
- `exitdemo_lob.c` では、LOB データへの読取りアクセスの取得方法の例が示されます。
- `exitdemo_pk_befores.c` では、主キーの更新レコードの前後のイメージ部分、および通常の更新のイメージ前（キー以外の更新）にアクセスする方法が示されます。また、競合の検出手段として、Replicat パラメータ・ファイル内の `SQLEXEC` を使用してターゲット列を取得する方法も示されます。ターゲットからフェッチされる値は、ユーザー・イグジットに入るときにターゲット・レコードとしてマッピングされます。

各ディレクトリには、`.c` ファイル、`Make` ファイルおよび `readme.txt` ファイルが含まれます。

ファンクションの概要

パラメータ	説明
EXIT_CALL_TYPE	処理中にいつルーチンをコールするかを指定します。
EXIT_CALL_RESULT	ルーチンにレスポンスを提供します。
EXIT_PARAMS	ルーチンに情報を提供します。
ERCALLBACK	コールバック・ルーチンを実装します。コールバック・ルーチンは、レコードおよび Oracle GoldenGate コンテキスト情報を取得し、データ・レコードの内容を変更します。

EXIT_CALL_TYPE の使用

EXIT_CALL_TYPE では、処理中にいつ Extract または Replicat プロセス（コール元）がユーザー・イグジット・ルーチンをコールするかを指定します。プロセスは、次のコールを使用してルーチンをコールできます。

表 60 ユーザー・イグジット・コール

コール・タイプ	処理ポイント
EXIT_CALL_START	処理開始時にコールされます。ユーザー・イグジットは、ファイルのオープンや変数の初期化などの初期化作業を実行できます。
EXIT_CALL_STOP	プロセスが正常に停止または異常終了する前にコールされます。ユーザー・イグジットは、ファイルのクローズや合計の出力などの完了作業を実行できます。
EXIT_CALL_BEGIN_TRANS	Replicat トランザクションの開始直前にコールされます。
EXIT_CALL_END_TRANS	Replicat トランザクションの最後のレコード直後にコールされます。
EXIT_CALL_CHECKPOINT	Extract または Replicat チェックポイントが書き込まれた直後にコールされます。
EXIT_CALL_PROCESS_RECORD	<ul style="list-style-type: none"> ◆ Extract の場合は、レコード・バッファがトレイルに出力される前にコールされます。 ◆ Replicat の場合は、レプリケートされた操作が実行される直前にコールされます。 <p>このコールは、ほとんどのユーザー・イグジット処理の基盤です。EXIT_CALL_PROCESS_RECORD がコールされると、レコード・バッファおよびその他のレコード情報がコールバック・ルーチンに使用可能になります。ソースとターゲットのマッピングがパラメータ・ファイルで指定されている場合、マッピングは EXIT_CALL_PROCESS_RECORD イベントが発生する前に実行されます。ユーザー・イグジットは、レコード内の他のすべての操作を、マップ、変換、削除、または実行できます。ユーザー・イグジットは、コール元がレコードを処理または無視する必要があるかを示すステータスを返すことができます。</p>
EXIT_CALL_PROCESS_MARKER	Replicat の処理中に、NonStop サーバーのマーカーがトレイルから読み取られ、マーカー履歴ファイルに書き込まれる前にコールされます。
EXIT_CALL_DISCARD_RECORD	<p>Replicat の処理中に、レコードが破棄ファイルに書き込まれる前にコールされます。レコードは、Oracle GoldenGate 変更レコードの値がターゲット表の現在のバージョンと異なっているときなど、複数の理由で破棄されます。関連する破棄バッファは、コールバック・ルーチンを使用してユーザー・イグジットによって取得および操作できます。</p> <p>このコール・タイプは、Extract プロセスでは使用できません。</p>

表 60 ユーザー・イグジット・コール（続き）

コール・タイプ	処理ポイント
EXIT_CALL_DISCARD_ASCII_RECORD	Extract の処理中に、ASCII 入力レコードが破棄ファイルに書き込まれる前にコールされます。関連する ASCII バッファは、コールバック・ルーチンを使用してユーザー・イグジットによって取得および操作できます。 このコール・タイプは、Replicat プロセスでは使用できません。
EXIT_CALL_FATAL_ERROR	Extract または Replicat の処理中に致命的なエラーが発生後、Oracle GoldenGate が停止する直前にコールされます。
EXIT_CALL_RESULT	各イグジット・コールの完了時に、コール元にレスポンス方法を指示するためにユーザー・イグジット・ルーチンによって設定されます。

EXIT_CALL_RESULT の使用

EXIT_CALL_RESULT では、ルーチンにレスポンスを提供します。

表 61 ユーザー・イグジット・レスポンス

コール結果	説明
EXIT_OK_VAL	ルーチンがイベントに対して何も実行しない場合は、EXIT_OK_VAL とみなされます。イグジット・コール・タイプが次のいずれかの場合 EXIT_CALL_PROCESS_RECORD EXIT_CALL_DISCARD_RECORD EXIT_CALL_DISCARD_ASCII_RECORD ... かつ EXIT_OK_VAL が返される場合、Oracle GoldenGate はユーザー・イグジットから返されたレコード・バッファを処理します。
EXIT_IGNORE_VAL	レコードのそれ以上の処理を拒否します。EXIT_IGNORE_VAL は、ユーザー・イグジットが特定のレコードに必要なすべての処理を実行し、データ・レコードの出力やレプリケートが必要ないときに適切です。
EXIT_STOP_VAL	コール元に処理を正常に停止するように指示します。EXIT_STOP_VAL または EXIT_ABEND_VAL は、ユーザー・イグジットでエラー状態が発生したときに適切です。
EXIT_ABEND_VAL	コール元に即座に停止するように指示します。
EXIT_PROCESSED_REC_VAL	Extract または Replicat に、レコードをスキップし、レポート・ファイルに出力される統計の該当する表および操作タイプを更新するように指示します。

EXIT_PARAMS の使用

EXIT_PARAMS では、ユーザー・イグジット・ルーチンにプログラム名やユーザー定義パラメータなどの情報を提供します。単一のデータ・レコードを複数回処理できます。

表 62 ユーザー・イグジット入力

イグジット・パラメータ	説明
PROGRAM_NAME	コール元プロセスの完全パスおよび名前を指定します (\ggs\extract や \ggs\replicat など)。このパラメータは、Windows API を使用して Oracle GoldenGate コールバック・ルーチンをロードするときや、ユーザー・イグジットが Extract および Replicat 両方の処理で使用されているときに、コール元を特定するために使用します。
FUNCTION_PARAM	<ul style="list-style-type: none"> ◆ リテラル文字列のパラメータをユーザー・イグジットに渡すために使用します。パラメータを渡す側である TABLE または MAP 文の EXITPARAM オプションでパラメータを指定します。236 ページを参照してください。これは、特定のレコードを処理するイグジット・コール中にのみ有効です。 ◆ FUNCTION_PARAM は、CUSEREXIT パラメータの PARAMS オプションで指定されているパラメータを渡すために、イグジット・コール起動イベントでも使用できます。(142 ページを参照してください。) これは、イグジットの起動時にグローバル・パラメータを提供するためにのみ有効です。
MORE_RECS_IND	イグジットからのリターンに対して設定します。データベース・レコードに対して、Extract または Replicat プロセスがレコードをもう一度処理するかどうかを決定します。これにより、Enscribe を SQL に変換する（データ正規化）ときの一般的なファンクションで、ユーザー・イグジットは Extract によって処理される各レコードに対して多くのレコードを出力できます。同一のレコードをもう一度リクエストするには、MORE_RECS_IND を CHAR_NO_VAL または CHAR_YES_VAL に設定します。

ERCALLBACK の使用

ERCALLBACK では、コールバック・ルーチンを実行します。ユーザー・コールバック・ルーチンは、Extract または Replicat プロセスからコンテキスト情報を取得し、コール・タイプが次のいずれかのときにレコード自体を含むコンテキスト値を設定します。

- EXIT_CALL_PROCESS_RECORD
- EXIT_CALL_DISCARD_RECORD
- EXIT_CALL_DISCARD_ASCII_RECORD

構文 ERCALLBACK (<function_code>, <buffer>, <result_code>);

引数	説明
<function_code>	コールバック・ルーチンによって実行されるファンクション。ユーザー・コールバック・ルーチンは、コールバック・ルーチンに渡されるファンクション・コードに基づいて異なる動作をします。一部のファンクションは、Extract および Replicat 両方が使用できますが、各プロセスでのファンクションの有効性は、コールバック・ルーチン中にそのファンクションに対して設定される入力パラメータに依存します。使用可能なファンクション・コードの詳細は、442 ページの「ファンクション・コード」を参照してください。
<buffer>	指定するファンクション・コードに関連付けられている事前定義済構造体を含むバッファへの void ポインタ。
<result_code>	コールバック・ルーチンによって実行されるファンクションのステータス。コールバック・ルーチンによって返される結果コードは、コールバック・ファンクションが成功したかどうかを示します。結果コードは、表 63 の値の 1 つになります。

表 63 結果コード

コード	説明
EXIT_FN_RET_OK	コールバック・ファンクションは成功しました。
EXIT_FN_RET_INVALID_COLUMN	ファンクション・コールで、存在しない列が参照されました。
EXIT_FN_RET_COLUMN_NOT_FOUND	圧縮更新レコードで列が見つかりませんでした。
EXIT_FN_RET_TABLE_NOT_FOUND	無効な表名が指定されました。
EXIT_FN_RET_BAD_COLUMN_DATA	列データを取得または設定中に無効なデータを検出しました。
EXIT_FN_RET_INVALID_CONTEXT	コールバック・ファンクションが不適切なときにコールされました。
EXIT_FN_RET_INVALID_PARAM	無効なパラメータがコールバック・ファンクションに渡されました。
EXIT_FN_RET_INVALID_CALLBACK_FUNC_CD	無効なコールバック・ファンクション・コードがコールバック・ルーチンに渡されました。
EXIT_FN_RET_NOT_SUPPORTED	このファンクションは、このプロセスに対してサポートされていません。
EXIT_FN_RET_FETCH_ERROR	レコードをフェッチできませんでした。エラー・メッセージを表示して理由を確認してください。
EXIT_FN_RET_EXCEEDED_MAX_LENGTH	表または列名が割り当てられたバッファに収まらなかったため、メタデータを取得できませんでした。
EXIT_FN_RET_TOKEN_NOT_FOUND	指定されたトークンがレコードに見つかりませんでした。
EXIT_FN_RET_INCOMPLETE_DDL_REC	DDL レコードの処理中に内部エラーが発生しました。レコードが不完全の可能性があります。
EXIT_FN_RET_ENV_NOT_FOUND	指定された環境値がレコードに見つかりませんでした。
EXIT_FN_RET_INVALID_COLUMN_TYPE	ルーチンは、その目的では Oracle GoldenGate にサポートされていないデータ・タイプを操作しようとしています。

ファンクション・コード

ファンクション・コードは、コールバック・ルーチンの出力を決定します。コールバック・ルーチンは、データ・バッファの内容が指定されたファンクション・コードの構造体と一致するとみなします。コールバック・ルーチン・ファンクション・コードとそのデータ・バッファは、次の項で説明します。次に、使用可能なファンクションの概要を示します。

表 64 Oracle GoldenGate ファンクション・コードの概要

ファンクション・コード	説明
COMPRESS_RECORD	COMPRESS_RECORD ファンクションは、マッピングの後にターゲット表の列がすべてでなく一部のみ存在する場合で、個々の列値ではなくレコード全体を操作する必要があるときに使用します。
DECOMPRESS_RECORD	DECOMPRESS_RECORD ファンクションは、マッピングの後にターゲット表の列がすべてでなく一部のみ存在する場合で、個々の列値ではなくレコード全体を操作する必要があるときに使用します。
FETCH_CURRENT_RECORD	FETCH_CURRENT_RECORD では、処理中のソース・レコードと同じキーを持つターゲット表内のレコードを、ロックせずに取得します。
FETCH_CURRENT_RECORD_WITH_LOCK	FETCH_CURRENT_RECORD_WITH_LOCK では、処理中のソース・レコードと同じキーを持つターゲット表内のレコードを取得し、同時にこのレコードを更新のためにロックします。
GET_BEFORE_AFTER_IND	GET_BEFORE_AFTER_IND ファンクションでは、レコードがデータベース操作のビフォア・イメージかアフター・イメージかを確認します。
GET_COL_METADATA_FROM_INDEX	GET_COL_METADATA_FROM_INDEX ファンクションでは、特定の列索引に関連付けられている列メタデータを確認します。
GET_COL_METADATA_FROM_NAME	GET_COL_METADATA_FROM_NAME ファンクションでは、特定の列名に関連付けられている列メタデータを確認します。
GET_COLUMN_INDEX_FROM_NAME	GET_COLUMN_INDEX_FROM_NAME ファンクションでは、特定の列名に関連付けられている列索引を確認します。
GET_COLUMN_NAME_FROM_INDEX	GET_COLUMN_NAME_FROM_INDEX ファンクションでは、特定の列索引に関連付けられている列名を確認します。
GET_COLUMN_VALUE_FROM_INDEX	GET_COLUMN_VALUE_FROM_INDEX ファンクションでは、指定する列索引を使用して、データ・レコードから列値を取得します。
GET_COLUMN_VALUE_FROM_NAME	GET_COLUMN_VALUE_FROM_NAME ファンクションでは、特定の列名を使用して、データ・レコードから列値を取得します。
GET_DDL_RECORD_PROPERTIES	GET_DDL_RECORD_PROPERTIES ファンクションでは、DDL 操作に関する情報を取得します。

表 64 Oracle GoldenGate ファンクション・コードの概要（続き）

ファンクション・コード	説明
GET_ENV_VALUE	GET_ENV_VALUE ファンクションでは、Oracle GoldenGate 環境に関する情報を返します。
GET_ERROR_INFO	GET_ERROR_INFO ファンクションでは、破棄レコードに関連付けられているエラー情報を取得します。
GET_GMT_TIMESTAMP	GET_GMT_TIMESTAMP ファンクションでは、操作コミット・タイムスタンプを GMT フォーマットで取得します。
GET_MARKER_INFO	GET_MARKER_INFO ファンクションでは、データを送信するときにマーカー情報を取得します。マーカーは、ユーザー・イグジット内でカスタム処理をトリガーするために使用します。
GET_OPERATION_TYPE	GET_OPERATION_TYPE ファンクションでは、レコードに関連付けられている操作のタイプを確認します。
GET_POSITION	GET_POSITION ファンクションでは、Oracle GoldenGate トレイル内の Extract データ・ポンプまたは Replicat の読取り位置を取得します。
GET_RECORD_BUFFER	GET_RECORD_BUFFER ファンクションでは、カスタム列変換に関する情報を取得します。
GET_RECORD_LENGTH	GET_RECORD_LENGTH ファンクションでは、データ・レコードの長さを取得します。
GET_RECORD_TYPE	GET_RECORD_TYPE ファンクションでは、処理されているレコードのタイプを取得します。
GET_STATISTICS	GET_STATISTICS ファンクションでは、Extract または Replicat プロセスの現在の処理統計を取得します。
GET_TABLE_COLUMN_COUNT	GET_TABLE_COLUMN_COUNT ファンクションでは、表内の列の合計数を取得します。
GET_TABLE_METADATA	GET_TABLE_METADATA ファンクションでは、処理中のレコードに関連付けられている表のメタデータを取得します。
GET_TABLE_NAME	GET_TABLE_NAME ファンクションでは、処理中のレコードに関連付けられているソースまたはターゲット表の名前を取得します。
GET_TIMESTAMP	GET_TIMESTAMP ファンクションでは、ソース・データ・レコードに関連付けられている I/O タイムスタンプを取得します。

表 64 Oracle GoldenGate ファンクション・コードの概要（続き）

ファンクション・コード	説明
GET_TRANSACTION_IND	GET_TRANSACTION_IND ファンクションでは、データ・レコードがトランザクションの最初、最後、または中間の操作かを確認します。
GET_USER_TOKEN_VALUE	GET_USER_TOKEN_VALUE ファンクションでは、トレイル・レコードからユーザー・トークン値を取得します。
OUTPUT_MESSAGE_TO_REPORT	OUTPUT_MESSAGE_TO_REPORT ファンクションでは、レポート・ファイルにメッセージを出力します。
RESET_USEREXIT_STATS	RESET_USEREXIT_STATS ファンクションでは、Oracle GoldenGate プロセスの統計をリセットします。
SET_COLUMN_VALUE_BY_INDEX	SET_COLUMN_VALUE_BY_INDEX ファンクションでは、データ・レコード全体を操作せず、単一の列値のみを変更します。
SET_COLUMN_VALUE_BY_NAME	SET_COLUMN_VALUE_BY_NAME ファンクションでは、データ・レコード全体を操作せず、単一の列値のみを変更します。
SET_OPERATION_TYPE	SET_OPERATION_TYPE ファンクションでは、レコードに関連付けられている操作のタイプを変更します。
SET_RECORD_BUFFER	SET_RECORD_BUFFER ファンクションは、HP NonStop ユーザー・イグジットとの互換性の維持、および複雑なデータ・レコード操作のために使用します。
SET_TABLE_NAME	SET_TABLE_NAME ファンクションでは、レコードに関連付けられている表名を変更します。

COMPRESS_RECORD

適用対象 Extract および Replicat

COMPRESS_RECORD ファンクションでは、DECOMPRESS_RECORD ファンクションで解凍されたレコードを再度圧縮します。COMPRESS_RECORD は、DECOMPRESS_RECORD を使用した後にのみ使用します。

構文

```
#include "usrdecs.h"
short result_code;
compressed_rec_def compressed_rec;
ERCALLBACK (COMPRESS_RECORD, &compressed_rec, &result_code);
```

バッファ

```
typedef struct
{
char *compressed_rec;
long compressed_len;
char *decompressed_rec;
long decompressed_len;
short *columns_present;
short source_or_target;
char requesting_before_after_ind;
} compressed_rec_def;
```

入力 次のようになります。

入力	説明
decompressed_rec	圧縮前レコードが含まれるバッファへのポインタ。レコードは、デフォルトの Oracle GoldenGate 正規フォーマットとみなされます。
decompressed_len	解凍されたレコードの長さ。
source_or_target	ソースまたはターゲットのどちらのレコードが圧縮されるかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL
requesting_before_after_ind	内部入力として使用されます。設定は不要です。設定しても無視されます。
columns_present	圧縮レコード内に存在する列を示す値の配列。たとえば、圧縮レコードに 1 番目、3 番目、6 番目の列が存在し、表内の列数の合計が 7 の場合、配列に次を含む必要があります。 1, 0, 1, 0, 0, 1, 0 表内の列数を取得するには、GET_TABLE_COLUMN_COUNT ファンクションを使用します（480 ページを参照してください）。

出力 次のようになります。

出力	説明
compressed_rec	圧縮フォーマットで返されるレコードへのポインタ。通常、compressed_rec はタイプ exit_rec_buf_def のバッファへのポインタです。exit_rec_buf_def バッファには、Extract または Replicat によってまもなく処理される実際のレコードが含まれます。このバッファは、コール・タイプが EXIT_CALL_DISCARD_RECORD のときに指定されます。イグジット・ルーチンは、たとえばカスタム・マッピング・ファンクションを実行するために、このバッファの内容を変えることがあります。コール元は、compressed_rec に割り当てる適切な量のメモリを確保する必要があります。
compressed_len	返される圧縮レコードの長さ。

返される値 EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_OK
EXIT_FN_RET_INVALID_PARAM

DECOMPRESS_RECORD

適用対象 Extract および Replicat

DECOMPRESS_RECORD ファンクションは、GET_RECORD_BUFFER（473 ページを参照してください）または SET_RECORD_BUFFER ファンクション（492 ページを参照してください）で更新レコード全体を取得または操作する必要があるものの、レコードが圧縮されている場合に使用します。DECOMPRESS_RECORD は、レコードを論理レイアウトに配置することにより、圧縮レコードの処理とマップを容易にします。存在する列は、索引および長さインジケータ（「圧縮レコード・フォーマット」を参照してください）なしで所定の位置に配置されます。行方不明の列は、ゼロとして表現されます。DECOMPRESS_RECORD を使用するときは、他の操作が発生する前に呼び出す必要があります。ユーザー・イグジットが処理を完了したら、Oracle GoldenGate プロセスに返す前に、COMPRESS_RECORD ファンクション（445 ページを参照してください）を使用してレコードを再度圧縮してください。

このファンクションは、UPDATE 操作の処理にのみ有効です。削除、挿入、および更新は、バッファに完全なレコード・イメージで保持されます。

圧縮レコード・フォーマット

圧縮 SQL 更新は、次のフォーマットを持ちます。

```
<index><length><value>[<index><length><value>][...]
```

条件:

- <index> は、表の列のリストへの 2 バイトの索引（最初の列はゼロ）です。
- <length> は、表の 2 バイトの長さです。
- <value> は、実際の列値で、適切な場合に次の 2 バイトの NULL インジケータの 1 つが含まれます。0 は非 NULL です。-1 は NULL です。

構文

```
#include "usrdecs.h"
short result_code;
compressed_rec_def compressed_rec;
ERCALLBACK (DECOMPRESS_RECORD, &compressed_rec, &result_code);
```

.....

バッファ typedef struct
 {
 char *compressed_rec;
 long compressed_len;
 char *decompressed_rec;
 long decompressed_len;
 short *columns_present;
 short source_or_target;
 char requesting_before_after_ind;
 } compressed_rec_def;

入力 次のようになります。

入力	説明
compressed_rec	圧縮フォーマットのレコードへのポインタ。この値を取得するには、GET_RECORD_BUFFER ファンクションを使用します（473 ページを参照してください）。
compressed_len	圧縮レコードの長さ。この値を取得するには、GET_RECORD_BUFFER（473 ページを参照してください）または GET_RECORD_LENGTH ファンクション（476 ページを参照してください）を使用します。
source_or_target	ソースまたはターゲットのどちらのレコードが解凍されるかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL
requesting_before_after_ind	内部入力として使用されます。設定は不要です。設定しても無視されます。

出力 次のようになります。

出力	説明
decompressed_rec	解凍済フォーマットで返されるレコードへのポインタ。レコードは、デフォルトの Oracle GoldenGate 内部正規フォーマットとみなされます。コール元は、decompressed_rec に割り当てる適切な量のメモリーを確保する必要があります。
decompressed_len	返される解凍済レコードの長さ。
columns_present	圧縮レコード内に存在する列を示す値の配列。たとえば、圧縮レコードに 1 番目、3 番目、6 番目の列が存在し、表内の列数の合計が 7 の場合、配列に次を含む必要があります。 1, 0, 1, 0, 0, 1, 0 マッピング・ファンクションは、この配列によって圧縮列をマップするかどうかとその時期を決定します。

返される値 EXIT_FN_RET_INVALID_CONTEXT
 EXIT_FN_RET_OK
 EXIT_FN_RET_INVALID_PARAM

FETCH_CURRENT_RECORD

適用対象 Replicat（現在は c-tree データベースをサポート）

FETCH_CURRENT_RECORD では、処理中のソース・レコードと同じキーを持つターゲット表内のレコードを、ロックせずに取得します。ターゲット・イメージに対して指定された次のいずれかのレコード取得ファンクションへのコールが完了すると、ユーザー・イグジットによるレコード・バッファの読取りが可能になります。

- GET_RECORD_BUFFER
- GET_COLUMN_VALUE_FROM_INDEX
- GET_COLUMN_VALUE_FROM_NAME

ターゲット表からのフェッチは、Replicat によってのみ、マップされたターゲット・バッファが使用可能なときにのみサポートされます。Replicat は、キーによって現在のレコードをフェッチします。

構文

```
#include "usrdecs.h"
short result_code;
error_info_def error_info;
ERCALLBACK (FETCH_CURRENT_RECORD, &error_info, &result_code);
```

バッファ

```
typedef struct
{
    long error_num;
    char *error_msg;
    long max_length;
    long actual_length;
    short msg_truncated;
} error_info_def;
```

入力 次のようになります。

入力	説明
error_msg	返されるエラー・メッセージを受け付けるバッファへのポインタ。
max_length	結果のエラー・メッセージを受け付けるために割り当てた error_msg バッファの最大長。これは NULL 終了文字列として返されます。

出力 次のようになります。

出力	説明
error_num	破棄ファイルに関連付けられている SQL またはシステム・エラー番号。
error_msg	破棄レコードに関連付けられている NULL で終了するエラー・メッセージ文字列へのポインタ。
actual_length	NULL 終了文字を含まないエラー・メッセージ長。

出力	説明
msg_truncated	エラー・メッセージが切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、エラー・メッセージと NULL 終了文字を足した長さが、最大バッファ長を超えるときに行われます。

返される値

```
EXIT_FN_RET_OK
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_NOT_SUPPORTED
EXIT_FN_RET_FETCH_ERROR
```

FETCH_CURRENT_RECORD_WITH_LOCK

適用対象 Extract および Replicat (現在は c-tree データベースをサポート)

FETCH_CURRENT_RECORD_WITH_LOCK では、処理中のソース・レコードと同じキーを持つターゲット表内のレコードを取得し、同時にこのレコードを更新のためにロックします。ターゲット・イメージに対して指定された次のいずれかのレコード取得ファンクションへのコールが完了すると、ユーザー・イグジットによるレコード・バッファの読取りが可能になります。

- GET_RECORD_BUFFER
- GET_COLUMN_VALUE_FROM_INDEX
- GET_COLUMN_VALUE_FROM_NAME

ターゲット表からのフェッチは、Replicat によってのみ、マップされたターゲット・バッファが使用可能なときのみサポートされます。Replicat は、キーによって現在のレコードをフェッチします。

構文

```
#include "usrdecs.h"
short result_code;
error_info_def error_info;
ERCALLBACK (FETCH_CURRENT_RECORD_WITH_LOCK, &error_info, &result_code);
```

バッファ

```
typedef struct
{
    long error_num;
    char *error_msg;
    long max_length;
    long actual_length;
    short msg_truncated;
} error_info_def;
```

入力 次のようになります。

入力	説明
error_msg	返されるエラー・メッセージを受け付けるバッファへのポインタ。
max_length	結果のエラー・メッセージを受け付けるために割り当てた error_msg バッファの最大長。これは NULL 終了文字列として返されます。

出力 次のようになります。

出力	説明
error_num	破棄ファイルに関連付けられている SQL またはシステム・エラー番号。
error_msg	破棄レコードに関連付けられている NULL で終了するエラー・メッセージ文字列へのポインタ。
actual_length	NULL 終了文字を含まないエラー・メッセージ長。
msg_truncated	エラー・メッセージが切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、エラー・メッセージと NULL 終了文字を足した長さが、最大バッファ長を超えるときに行われます。

返される値

```

EXIT_FN_RET_OK
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_NOT_SUPPORTED
EXIT_FN_RET_FETCH_ERROR

```

GET_BEFORE_AFTER_IND

適用対象 Extract および Replicat

GET_BEFORE_AFTER_IND ファンクションでは、レコードがデータベース操作のビフォア・イメージかアフター・イメージかを確認します。挿入の場合はアフター・イメージ、削除の場合はビフォア・イメージ、更新の場合はアフターまたはビフォア・イメージのいずれかになります (Extract および Replicat パラメータの GETUPDATEBEFORES および GETUPDATEAFTERS を参照してください)。更新のビフォア・イメージが抽出される場合は、同一の更新内でビフォア・イメージがアフター・イメージよりも先行します。

構文

```

#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (GET_BEFORE_AFTER_IND, &record, &result_code);

```

バッファ

```

typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCTURE_VERSION */
    char requesting_before_after_ind;
} record_def;

```

入力 なし

出力 次のようになります。

出力	説明
before_after_ind	レコードがビフォア・イメージかアフター・イメージかを示す次の一方。 BEFORE_IMAGE_VAL AFTER_IMAGE_VAL

返される値 EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_OK

GET_COL_METADATA_FROM_INDEX

適用対象 Extract および Replicat

GET_COL_METADATA_FROM_INDEX ファンクションでは、該当の列の索引を指定して列メタデータを取得します。

構文

```
#include "usrdecs.h"
short result_code;
col_metadata_def column_meta_rec;
ERCALLBACK (GET_COL_METADATA_FROM_INDEX, &column_meta_rec, &result_code);
```

バッファ

```
typedef struct
{
    short column_index;
    char *column_name;
    long max_name_length;
    short native_data_type;
    short gg_data_type;
    short gg_sub_data_type;
    short is_nullable;
    short is_part_of_key;
    short key_column_index;
    short length;
    short precision;
    short scale;
    short source_or_target;
} col_metadata_def;
```

入力 次のようになります。

入力	説明
column_index	返される列値の列索引。
max_name_length	返される列名の最大長。通常、最大長は名前のバッファの長さです。返される名前は NULL で終了しているため、最大長は列名の最大長と同じになるはずです。
source_or_target	ソースまたはターゲットのどちらのレコードが圧縮されているかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
column_name	返される列値の列名。
native_data_type	列の（データベースに）ネイティブなデータ型。プロセスに応じて、次のように native_data_type または dd_data_type が返されます。 ◆ Extract がソース列に対してコールバック・リクエストを行っている場合は、 native_data_type が返されます。 Extract がマップされたターゲット列をリクエストしている場合は、（システム上にターゲット定義があるとみなされ） gg_data_type が返されます。 ◆ Extract データ・ポンプがソース列に対してコールバック・リクエストを行い、ローカル・データベースが存在する場合は、 native_data_type が返されます。データベースがない場合は、（システム上にソース定義ファイルがあるとみなされ） gg_data_type が返されます。データ・ポンプがマップされたターゲット列をリクエストしている場合は、（システム上にターゲット定義が存在するとみなされ） gg_data_type が返されます。 ◆ Replicat がソース列に対してコールバック・リクエストを行っている場合は、 gg_data_type が返されます（システム上にソース定義が存在するとみなされます）。 Replicat がソース列をリクエストし、パラメータ・ファイルで ASSUMETARGETDEFS が使用されている場合は、 native_data_type が返されます。 Replicat がターゲット列をリクエストしている場合は、 native_data_type が返されます。
gg_data_type	列の Oracle GoldenGate データ型。
gg_sub_data_type	列の Oracle GoldenGate サブデータ型。
is_nullable	列が NULL 値を許可するかどうかを示すフラグ（TRUE または FALSE）。
is_part_of_key	列が Oracle GoldenGate によって使用されているキーの一部かどうかを示すフラグ（TRUE または FALSE）。

出力	説明										
key_column_index	<p>索引での列の順序を示します。たとえば次の表は、主キーで宣言されている順序とは異なる順序で存在する 2 つのキー列を持ちます。</p> <pre>CREATE TABLE ABC (cust_code VARCHAR2(4), name VARCHAR2(30), city VARCHAR2(20), state CHAR(2), PRIMARY KEY (city, cust_code) USING INDEX);</pre> <p>論理列順に各列に対してコールバック・ファンクションを実行すると、次が返されます。</p> <table> <tr> <th>列名</th><th>返されるキー索引値</th></tr> <tr> <td>cust_code</td><td>1</td></tr> <tr> <td>name</td><td>-1</td></tr> <tr> <td>city</td><td>0</td></tr> <tr> <td>state</td><td>-1</td></tr> </table> <ul style="list-style-type: none"> ◆ 列がキーの一部の場合、返される値はキー内の列の順序です。 ◆ 列がキーの一部ではない場合は、値 -1 が返されます。 	列名	返されるキー索引値	cust_code	1	name	-1	city	0	state	-1
列名	返されるキー索引値										
cust_code	1										
name	-1										
city	0										
state	-1										
length	列の長さを返します。										
precision	数値データ型の場合、列の精度を返します。										
scale	数値データ型の場合、位取りを返します。										

返される値

```
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_EXCEEDED_MAX_LENGTH
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_OK
```

GET_COL_METADATA_FROM_NAME

適用対象

Extract および Replicat

GET_COL_METADATA_FROM_NAME ファンクションでは、該当の列の名前を指定して列メタデータを取得します。

構文

```
#include "usrdecs.h"
short result_code;
col_metadata_def column_meta_rec;
ERCALLBACK (GET_COL_METADATA_FROM_NAME, &column_meta_rec, &result_code);
```

バッファ

```
typedef struct
{
    short column_index;
    char *column_name;
    long max_name_length;
    short native_data_type;
    short gg_data_type;
    short gg_sub_data_type;
    short is_nullable;
    short is_part_of_key;
    short key_column_index;
    short length;
    short precision;
    short scale;
    short source_or_target;
} col_metadata_def;
```

入力

次のようになります。

入力	説明
column_name	返される列値の列名。
source_or_target	ソースまたはターゲットのどちらのレコードが圧縮されているかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明										
column_index	返される列値の列索引。										
source_or_target	ソースまたはターゲットのどちらのレコードが圧縮されているかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL										
native_data_type	列の（データベースに）ネイティブなデータ型。										
gg_data_type	列の Oracle GoldenGate データ型。										
gg_sub_data_type	列の Oracle GoldenGate サブデータ型。										
is_nullable	列が NULL 値を許可するかどうかを示すフラグ（TRUE または FALSE）。										
is_part_of_key	列が Oracle GoldenGate によって使用されているキーの一部かどうかを示すフラグ（TRUE または FALSE）。										
key_column_index	索引での列の順序を示します。たとえば、次の表は、表での定義の順序と索引での定義の順序が異なる 2 つのキー列を持ちます。 <pre>CREATE TABLE tcustmer (cust_code VARCHAR2(4), name VARCHAR2(30), city VARCHAR2(20), state CHAR(2), PRIMARY KEY (city, cust_code) USING INDEX);</pre> 返される値は次のようになります。 <table> <tr> <th>列名</th><th>返されるキー索引値</th></tr> <tr> <td>cust_code</td><td>1</td></tr> <tr> <td>name</td><td>-1</td></tr> <tr> <td>city</td><td>0</td></tr> <tr> <td>state</td><td>-1</td></tr> </table> <ul style="list-style-type: none"> ◆ 列がキーの一部の場合は、索引での順序が整数で返されます。 ◆ 列がキーの一部ではない場合は、値 -1 が返されます。 	列名	返されるキー索引値	cust_code	1	name	-1	city	0	state	-1
列名	返されるキー索引値										
cust_code	1										
name	-1										
city	0										
state	-1										
length	列の長さを返します。										
precision	数値データ型の場合、列の精度を返します。										
scale	数値データ型の場合、位取りを返します。										

返される値

```
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_EXCEEDED_MAX_LENGTH
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_OK
```

GET_COLUMN_INDEX_FROM_NAME

適用対象 Extract および Replicat

GET_COLUMN_INDEX_FROM_NAME ファンクションでは、特定の列名に関連付けられている列索引を確認します。

構文

```
#include "usrdecs.h"
short result_code;
env_value_def env_value;
ERCALLBACK (GET_COLUMN_INDEX_FROM_NAME, &env_value, &result_code);
```

バッファ

```
typedef struct
{
    char *buffer;
    long max_length;
    long actual_length;
    short value_truncated;
    short index;
    short source_or_target;
} env_value_def;
```

入力 次のようになります。

入力	説明
buffer	列名へのポインタ。
actual_length	バッファ内の列名の長さ。
source_or_target	列名情報を参照するためにソース表とターゲット表のどちらを使用するかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
index	指定した列名に対して返される列索引。

返される値

```
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
```

GET_COLUMN_NAME_FROM_INDEX

適用対象 Extract および Replicat

GET_COLUMN_NAME_FROM_INDEX ファンクションでは、特定の列索引に関連付けられている列名を確認します。

構文

```
#include "usrdecs.h"
short result_code;
env_value_def env_value;
ERCALLBACK (GET_COLUMN_NAME_FROM_INDEX, &env_value, &result_code);
```

バッファ

```
typedef struct
{
    char *buffer;
    long max_length;
    long actual_length;
    short value_truncated;
    short index;
    short source_or_target;
} env_value_def;
```

入力 次のようになります。

入力	説明
buffer	返される列名を受け付けるバッファへのポインタ。列名は NULL で終了します。
max_length	結果の列名を受け付けるために割り当てた buffer の最大長。NULL 終了文字列として返されます。
index	返される列名の列索引。
source_or_target	列名情報を参照するためにソース表とターゲット表のどちらを使用するかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
buffer	NULL で終了している列名。
actual length	返される列名の文字列長。実際の長さに NULL 終了文字は含まれません。
value_truncated	値が切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、列名と NULL 終了文字を足した長さが、最大バッファ長を超えるときに行われます。

返される値

```
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
```

GET_COLUMN_VALUE_FROM_INDEX

適用対象 Extract および Replicat

GET_COLUMN_VALUE_FROM_INDEX ファンクションでは、指定する列索引を使用して、データ・レコードから列値を取得します。列値は、ユーザー・イグジット内のほとんどのロジックの基本です。データ・レコード内の各列の値は、複雑なロジックの基準にできます。ユーザー・イグジットでは、列値が ASCII で返されるか、内部 Oracle GoldenGate 正規フォーマットで返されるかを指定できます。

構文

```
#include "usrdecs.h"
short result_code;
column_def column;
ERCALLBACK (GET_COLUMN_VALUE_FROM_INDEX, &column, &result_code);
```

バッファ

```
typedef struct
{
char *column_value;
unsigned short max_value_length;
unsigned short actual_value_length;
short null_value;
short remove_column;
short value_truncated;
short column_index;
char *column_name;
short ascii_or_internal;
short source_or_target;
/* Version 2 CALLBACK_STRUCT_VERSION */
char requesting_before_after_ind;
char more_lob_data;
} column_def;
```

入力 次のようになります。

入力	説明
column_value	返された列値を受け付けるバッファへのポインタ。
max_value_length	返される列値の最大長。通常、最大長は列値バッファの長さです。 ascii_or_internal で ASCII フォーマットが指定される場合、列値は NULL で終了し、最大長は列値の最大長と同じになるはずです。
column_index	返される列値の列索引。
ascii_or_internal	列値が ASCII で返されるか Oracle GoldenGate 内部正規フォーマットで返されるかを示す次の一方。 EXIT_FN_ASCII_FORMAT EXIT_FN_INTERNAL_FORMAT 内部正規フォーマットには、適切な場合に、2 バイトの NULL インジケータおよび 2 バイトの変数データ長が含まれます。
source_or_target	列値の取得にソースまたはターゲット・データ・レコードのどちらを使用するかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL
requesting_before_after_ind	<p>アフター・イメージ・レコードの処理中に、更新または主キー更新のビフォア・イメージ列値を必要とする場合に設定します。</p> <p>主キー更新または通常の（非キー）更新レコードの "アフター・イメージ" を処理中に、列の "ビフォア" 値を取得するには、requesting_before_after_ind フラグを BEFORE_IMAGE_VAL に設定します。</p> <ul style="list-style-type: none"> ◆ 主キー更新のキー列のビフォア・イメージにアクセスする場合は、他の設定は不要です。 ◆ 主キー更新の非キー列または通常の更新の任意の列にアクセスするには、ビフォア・イメージが使用可能である必要があります。 <p>requesting_before_after_ind の入力が明示的に指定されていない場合、デフォルト設定は AFTER_IMAGE_VAL（列のアフター・イメージの取得）です。</p> <p>ビフォア・イメージを使用可能にするには、GETUPDATEBEFORES パラメータを使用するか、USEREXIT パラメータ文内の INCLUDEUPDATEBEFORES オプションを使用します。</p> <p>次の点に注意してください。</p> <ul style="list-style-type: none"> ◆ GETUPDATEBEFORES を使用すると、Extract プロセスはトレイルにビフォア・イメージを書き込み、ビフォア・イメージを使用してユーザー・イグジットへの EXIT_CALL_PROCESS_RECORD コールも行います。 ◆ INCLUDEUPDATEBEFORES を使用すると、ユーザー・イグジットへの EXIT_CALL_PROCESS_RECORD コールは行われず、Extract の場合はトレイルへのビフォア・イメージの書き込みも行われません。

出力 次のようになります。

出力	説明
column_value	<p>返される列値へのポインタ。ascii_or_internal が EXIT_FN_ASCII_FORMAT として指定されている場合、列値は NULL で終了する ASCII 文字列として返され、それ以外の場合は Oracle GoldenGate 内部正規フォーマットで返されます。ASCII フォーマット場合、日付は次のフォーマットで返されます。</p> <p>YYYY-MM-DD HH:MI:SS.FFFFFFFF</p> <p>秒の端数が含まれるかどうかは、データベースに依存します。</p>
actual_value_length	<p>返される列名の文字列長。ascii_or_internal が EXIT_FN_ASCII_FORMAT として指定されている場合、実際の長さに NULL 終了文字は含まれません。</p>
null_value	<p>列値が NULL かどうかを示すフラグ (0 または 1)。null_value フラグが 1 の場合、列値バッファは NULL バイトで埋められます。</p>
value_truncated	<p>値が切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、列値の長さが最大バッファ長を超えたときに行われます。ascii_or_internal が EXIT_FN_ASCII_FORMAT として指定されている場合は、列長に NULL 終了文字が含まれます。</p>
char more_lob_data;	<p>ベース・レコードに保持可能な初期の 4K の制限よりも大きい LOB データが存在するかどうかを示すフラグ。LOB は、4K の上限を越えている場合、LOB フラグメントに保持されます。</p> <p>返される値を保持するために適切な量のメモリを割り当てる必要があります。Oracle GoldenGate は、8K までのデータの LOB 列に常にアクセスし、ユーザー・イグジットによって割り当てられた量までバッファを埋めます。割り当てられた量よりも LOB が大きい場合は、すべてのデータがユーザー・イグジットに送信されるまで、後続のコールバックを使用してすべての列データを取得する必要があります。</p> <p>データの終端を確認するには、more_lob_data を検証します。ユーザー・イグジットは、新しい列にアクセスする前に、このフラグを CHAR_NO_VAL または CHAR_YES_VAL に設定します。最初のコールバック後にこのフラグがまだ初期化状態で、CHAR_YES_VAL または CAR_NO_VAL にも設定されていない場合は、次のいずれかに該当します。</p> <ul style="list-style-type: none"> ◆ LOB の処理に十分なメモリが割り当てられていた。 ◆ LOB でなかった。 ◆ ベース・トレイル・レコード・サイズの 4K の制限を越えていなかった。 <p>列が LOB かどうかを確認するために、ソース表メタデータを取得することをお勧めします。</p>

返される値

```
EXIT_FN_RET_BAD_COLUMN_DATA
EXIT_FN_RET_COLUMN_NOT_FOUND
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
```

GET_COLUMN_VALUE_FROM_NAME

適用対象 Extract および Replicat

GET_COLUMN_VALUE_FROM_NAME ファンクションでは、指定する列名を使用して、データ・レコードから列値を取得します。列値は、ユーザー・イグジット内のほとんどのロジックの基本です。データ・レコード内の各列の値は、複雑なロジックの基準にできます。ユーザー・イグジットでは、列値が ASCII で返されるか、内部 Oracle GoldenGate 正規フォーマットで返されるかを指定できます。

構文

```
#include "usrdecs.h"
short result_code;
column_def column;
ERCALLBACK (GET_COLUMN_VALUE_FROM_NAME, &column, &result_code);
```

バッファ

```
typedef struct
{
    char *column_value;
    unsigned short max_value_length;
    unsigned short actual_value_length;
    short null_value;
    short remove_column;
    short value_truncated;
    short column_index;
    char *column_name;
    short ascii_or_internal;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
    char more_lob_data;
} column_def;
```

入力 次のようになります。

入力	説明
column_value	返された列値を受け付けるバッファへのポインタ。
max_value_length	返される列値の最大長。通常、最大長は列値バッファの長さです。ASCII フォーマットが指定された場合 (ascii_or_internal を参照)、列値は NULL で終了し、最大長は列値の最大長と同じになります。
column_name	返される列値の列名。
ascii_or_internal	<p>列値が ASCII で返されるか Oracle GoldenGate 内部正規フォーマットで返されるかを示す次の一方。</p> <p>EXIT_FN_ASCII_FORMAT EXIT_FN_INTERNAL_FORMAT</p> <p>内部正規フォーマットには、適切な場合に、2 バイトの NULL インジケータおよび 2 バイトの変数データ長が含まれます。</p>
source_or_target	<p>列値の取得にソースまたはターゲット・データ・レコードのどちらを使用するかを示す次の一方。</p> <p>EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL</p>
requesting_before_after_ind	<p>アフター・イメージ・レコードの処理中で、更新または主キー更新のビフォア列を必要とする場合に設定します。</p> <p>主キー更新または通常の（非キー）更新レコードの "アフター・イメージ" を処理中に、列の "ビフォア" 値を取得するには、requesting_before_after_ind フラグを BEFORE_IMAGE_VAL に設定します。</p> <ul style="list-style-type: none"> ◆ 主キー更新のキー列のビフォア・イメージにアクセスする場合は、他の設定は不要です。 ◆ 主キー更新の非キー列または通常の更新の任意の列にアクセスするには、ビフォア・イメージが使用可能である必要があります。 <p>requesting_before_after_ind の入力が明示的に指定されていない場合、デフォルト設定は AFTER_IMAGE_VAL（列のアフター・イメージの取得）です。</p> <p>ビフォア・イメージを使用可能にするには、GETUPDATEBEFORES パラメータを使用するか、CUSEREXIT パラメータ文内の INCLUDEUPDATEBEFORES オプションを使用します。</p> <p>次の点に注意してください。</p> <ul style="list-style-type: none"> ◆ GETUPDATEBEFORES を使用すると、Extract プロセスはトレイルにビフォア・イメージを書き込み、ビフォア・イメージを使用してユーザー・イグジットへの EXIT_CALL_PROCESS_RECORD コールも行います。 ◆ INCLUDEUPDATEBEFORES を使用すると、ユーザー・イグジットへの EXIT_CALL_PROCESS_RECORD コールは行われず、Extract の場合はトレイルへのビフォア・イメージの書き込みも行われません。

出力 次のようになります。

出力	説明
column_value	<p>返される列値へのポインタ。ascii_or_internal が EXIT_FN_ASCII_FORMAT として指定されている場合、列値は NULL で終了する ASCII 文字列として返され、それ以外の場合は Oracle GoldenGate 内部正規フォーマットで返されます。ASCII フォーマット場合、日付は次のフォーマットで返されます。</p> <p>CCYY-MM-DD HH:MI:SS.FFFFFFFF</p> <p>秒の端数が含まれるかどうかは、データベースに依存します。</p>
actual_length	<p>返される列名の文字列長。ascii_or_internal が EXIT_FN_ASCII_FORMAT として指定されている場合、実際の長さに NULL 終了文字は含まれません。</p>
null_value	<p>列値が NULL かどうかを示すフラグ (0 または 1)。null_value フラグが 1 の場合、列値バッファは NULL バイトで埋められます。</p>
value_truncated	<p>値が切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、列値の長さが最大バッファ長を超えたときに行われます。ascii_or_internal が EXIT_FN_ASCII_FORMAT として指定されている場合は、t 列長に NULL 終了文字が含まれます。</p>
char more_lob_data;	<p>ベース・レコードに保持可能な初期の 4K の制限よりも大きい LOB データが存在するかどうかを示すフラグ。LOB は、4K の上限を越えている場合、LOB フラグメントに保持されます。</p> <p>返される値を保持するために適切な量のメモリを割り当てる必要があります。Oracle GoldenGate は、8K までのデータの LOB 列に常にアクセスし、ユーザー・イグジットによって割り当てられた量までバッファを埋めます。割り当てられた量よりも LOB が大きい場合は、すべてのデータがユーザー・イグジットに送信されるまで、後続のコールバックを使用してすべての列データを取得する必要があります。</p> <p>データの終端を確認するには、more_lob_data を検証します。ユーザー・イグジットは、新しい列にアクセスする前に、このフラグを CAR_NO_VAL または CHAR_YES_VAL に設定します。最初のコールバック後にこのフラグがまだ初期化状態で、CHAR_YES_VAL または CAR_NO_VAL にも設定されていない場合は、次のいずれかに該当します。</p> <ul style="list-style-type: none"> ◆ LOB の処理に十分なメモリが割り当てられていた。 ◆ LOB でなかった。 ◆ ベース・トレイル・レコード・サイズの 4K の制限を越えていなかった。 <p>列が LOB かどうかを確認するために、ソース表メタデータを取得することをお勧めします。</p>

返される値

EXIT_FN_RET_BAD_COLUMN_DATA
EXIT_FN_RET_COLUMN_NOT_FOUND
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK

```

例
memset (&col_meta, 0, sizeof(col_meta));
if (record.mapped)
col_meta.source_or_target = EXIT_FN_TARGET_VAL;
else
col_meta.source_or_target = EXIT_FN_SOURCE_VAL;
col_meta.source_or_target = EXIT_FN_SOURCE_VAL;
col_meta.column_name = (char *)malloc(100);
col_meta.max_name_length = 100;
col_meta.column_index = 1;

call_callback (GET_COL_METADATA_FROM_NAME, &col_meta, &result_code);

```

GET_DDL_RECORD_PROPERTIES

適用対象 Extract および Replicat (DDL レプリケーションがサポートされているデータベース)

DDL が実行されたオブジェクトに関する情報などの DDL 操作、および DDL 文自体のテキストが返されるようにするには、GET_DDL_RECORD_PROPERTIES 関数を使用します。Extract プロセスでは、ソース表のレイアウトのみ取得可能です。Replicat プロセスでは、ソースまたはターゲットのレイアウトの取得が可能です。

```

構文
#include "usrdecs.h"
short result_code;
ddl_record_def ddl_rec;
ERCALLBACK (GET_DDL_RECORD_PROPERTIES, &ddl_rec, &result_code);

```

```

バッファ
typedef struct
{
char *ddl_type;
long ddl_type_max_length; /* Maximum Description length PASSED IN BY USER */
long ddl_type_length; /* Actual length */

char *object_type;
long object_type_max_length; /* Maximum Description length PASSED IN BY USER */
long object_type_length; /* Actual length */

char *object_name;
long object_max_length; /* Maximum Description length PASSED IN BY USER */
long object_length; /* Actual length */

char *owner_name;
long owner_max_length; /* Maximum Description length PASSED IN BY USER */
long owner_length; /* Actual length */

char *ddl_text;
long ddl_text_max_length; /* Maximum Description length PASSED IN BY USER */
long ddl_text_length; /* Actual length */

short ddl_text_truncated; /* Was value truncated? */
short source_or_target; /* Source or target value? */
} ddl_record_def;

```

入力 次のようになります。

入力	説明
ddl_type_length object_type_length object_length owner_length ddl_text_length	返される列値を受け付ける各アイテム用の 1 つのバッファへのポインタ。 アイテムは次のとおりです。 <ul style="list-style-type: none"> ◆ ddl_type_length には、DDL 操作タイプの長さ（CREATE または ALTER など）を含めます。 ◆ object_type_length には、DDL 操作によって影響を受けるオブジェクトのタイプの長さ（TABLE または INDEX など）を含めます。 ◆ object_length には、オブジェクト名の長さを含めます。 ◆ owner_length には、オブジェクトの所有者（スキーマまたはデータベース）の長さを含めます。 ◆ ddl_text_length には、実際の DDL 文テキストの長さを含めます。
ddl_type_max_length	* ddl_type によって返される DDL 操作タイプの最大の長さを指定します。 DDL タイプは、データベースに対して有効な任意の DDL コマンド（ALTER など）です。
object_type_max_length	* object_type によって返されるオブジェクト・タイプの最大の長さを指定します。 オブジェクト・タイプは、データベースに有効な任意のオブジェクト（TABLE、INDEX、TRIGGER など）です。
object_max_length	* object_name によって返されるオブジェクト名の最大の長さを指定します。
owner_max_length	* owner_name によって返される所有者名の最大の長さを指定します。
ddl_text_max_length	* ddl_text によって返される DDL 文のテキストの最大の長さを指定します。
source_or_target	ソースまたはターゲット・データ・レコードのどちらの操作タイプを返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
ddl_type_length object_type_length object_length owner_length ddl_text_length	これらのフィールドはすべて、リクエストされた値の実際の長さを返します。（説明は入力のセクションを参照してください。）
ddl_text_truncated	DDL テキストが切り捨てられたかどうかを示すフラグ（0 または 1）。切捨ては、DDL テキストと NULL 終了文字を足した長さが、最大バッファ長を超えるときに行われます。

返される値

```
EXIT_FN_RET_OK
EXIT_FN_RET_NOT_SUPPORTED
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INCOMPLETE_DDL_REC
```

GET_ENV_VALUE

適用対象 Extract および Replicat

GET_ENV_VALUE ファンクションでは、Oracle GoldenGate 環境に関する情報を返します。指定される情報は、@GETENV 列変換ファンクションと同じで、同一の入力値を使用して指定されます。有効な情報タイプ、環境変数、および戻り値の詳細は、405 ページの @GETENV の項を参照してください。

構文

```
#include "usrdecs.h"
short result_code;
getenv_value_def env_ptr;
ERCALLBACK (GET_ENV_VALUE, &env_ptr, &result_code);
```

バッファ

```
typedef struct
{
char *information_type;
char *env_value_name;
char *return_value;
long max_return_length;
long actual_length;
short value_truncated;
} getenv_value_def;
```

入力 次のようになります。

入力	説明
information_type	返される情報タイプ ("GGENVIRONMENT" や "GGHEADER" など)。情報タイプは、二重引用符で囲って指定する必要があります。情報タイプのリストと詳細な説明は、405 ページを参照してください。
env_value_name	情報タイプから取得する環境値。環境値は、二重引用符で囲って指定する必要があります。環境値の詳細は、405 ページから開始されている、使用する情報タイプの 環境値 のリストを参照してください。たとえば、"GGENVIRONMENT" 情報タイプを使用している場合は、"GROUPNAME" が有効な環境値の 1 つです。
max_return_length	このデータのバッファの最大長。

出力 次のようになります。

出力	説明
return_value	指定する環境変数の説明のセクションで、この環境値に対してリストされている有効な 戻り値 。
actual_length	このバッファのデータの実際の長さ。
value_truncated	値が切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、値と NULL 終了文字を足した長さが、最大バッファ長を超えたときに行われます。

返される値

```
EXIT_FN_RET_OK
EXIT_FN_RET_ENV_NOT_FOUND
EXIT_FN_RET_INVALID_PARAM
```

GET_ERROR_INFO

適用対象 Extract および Replicat

GET_ERROR_INFO ファンクションでは、破棄レコードに関連付けられているエラー情報を取得します。ユーザー・イグジットは、この情報をカスタム・エラー処理ロジックで使用できます。たとえば、ユーザー・イグジットは詳細なエラー情報を含む電子メール・メッセージを送信できます。

構文

```
#include "usrdecs.h"
short result_code;
error_info_def error_info;
ERCALLBACK (GET_ERROR_INFO, &error_info, &result_code);
```

バッファ

```
typedef struct
{
    long error_num;
    char *error_msg;
    long max_length;
    long actual_length;
    short msg_truncated;
} error_info_def;
```

入力 次のようになります。

入力	説明
error_msg	返されるエラー・メッセージを受け付けるバッファへのポインタ。
max_length	結果のエラー・メッセージを受け付けるために割り当てた error_msg バッファの最大長。これは NULL 終了文字列として返されます。

出力 次のようになります。

出力	説明
error_num	破棄ファイルに関連付けられている SQL またはシステム・エラー番号。
error_msg	破棄レコードに関連付けられている NULL で終了するエラー・メッセージ文字列へのポインタ。
actual_length	NULL 終了文字を含まないエラー・メッセージ長。
msg_truncated	エラー・メッセージが切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、エラー・メッセージと NULL 終了文字を足した長さが、最大バッファ長を超えるときに行われます。

返される値 EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_OK

GET_GMT_TIMESTAMP

適用対象 Extract および Replicat

GET_GMT_TIMESTAMP ファンクションでは、操作コミット・タイムスタンプを GMT フォーマットで取得します。このファンクションを使用するには、usrdecs.h バージョン 2 を使用して再コンパイルする必要があります。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (GET_GMT_TIMESTAMP, &record, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
} record_def;
```

入力 なし

出力 次のようになります。

出力	説明
timestamp	返される GMT フォーマットの 64 ビット I/O タイムスタンプ。
io_datetime	ローカル I/O 日付および時刻を含む、次のフォーマットの NULL 終了文字列。 YYYY-MM-DD HH:MI:SS.FFFFFFFF

返される値 EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_OK

GET_MARKER_INFO

適用対象 Extract（データ・ポンプのみ）および Replicat

GET_MARKER_INFO ファンクションでは、Replicat がデータを適用しているときに NonStop ソース・システムから送信されたマーカー情報を取得します。マーカーは、ユーザー・イグジット内でカスタム処理をトリガーするために使用します。

構文

```
#include "usrdecs.h"
short result_code;
marker_info_def marker_info;
ERCALLBACK (GET_MARKER_INFO, &marker_info, &result_code);
```

バッファ

```
typedef struct
{
    char *processed;
    char *added;
    char *text;
    char *group;
    char *program;
    char *node;
} marker_info_def;
```

入力 次のようになります。

出力	説明
processed	processed 戻り値を受け付けるバッファへのポインタ。
added	added 戻り値を受け付けるバッファへのポインタ。
text	text 戻り値を受け付けるバッファへのポインタ。
group	group 戻り値を受け付けるバッファへのポインタ。
program	program 戻り値を受け付けるバッファへのポインタ。

出力	説明
node	node 戻り値を受け付けるバッファへのポインタ。

出力 次のようになります。

出力	説明
processed	マーカが処理されたローカルの日付と時刻を示す YYYY-MM-DD HH:MI:SS フォーマットの NULL 終了文字列。
added	マーカが追加されたローカルの日付と時刻を示す YYYY-MM-DD HH:MI:SS フォーマットの NULL 終了文字列。
text	マーカに関連付けられたテキストを含む NULL 終了文字列。
group	マーカを処理した Replicat グループを示す NULL 終了文字列。
program	マーカを処理したプログラムを示す NULL 終了文字列。
node	マーカが作成された Himalaya ノードを表す NULL 終了文字列。

返される値 EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_OK

GET_OPERATION_TYPE

適用対象 Extract および Replicat

GET_OPERATION_TYPE ファンクションでは、レコードに関連付けられている操作のタイプを確認します。ユーザー・イグジットで操作のタイプを把握できることは有益です。たとえばユーザー・イグジットは、削除操作を検出するたびに複雑な検証を実行できます。また、ユーザー・イグジットが完全なデータ・レコードを操作する場合は、圧縮レコードがいつ処理されるかを把握することも重要です。

かわりの方法として、GET_RECORD_BUFFER ファンクションでも操作タイプを確認できます (473 ページを参照してください)。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (GET_OPERATION_TYPE, &record, &result_code);
```



```
バッファ      typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
} record_def;
```

入力 次のようになります。

入力	説明
source_or_target	ソースまたはターゲット・データ・レコードのどちらの操作タイプを返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
io_type	次のいずれかが返されます。 DDL タイプ SQL_DDL_VAL DML タイプ DELETE_VAL INSERT_VAL UPDATE_VAL 圧縮 Enscribe 更新 UPDATE_COMP_ENSCRIBE_VAL 圧縮 SQL 更新 UPDATE_COMP_SQL_VAL UPDATE_COMP_PK_SQL_VAL その他 TRUNCATE_TABLE_VAL

返される値 EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK

GET_POSITION

適用対象 Extract（データ・ポンプのみ）および Replicat

GET_POSITION ファンクションでは、Oracle GoldenGate トレイル内の Extract データ・ポンプまたは Replicat の読取り位置を取得します。

構文

```
#include "usrdecs.h"
short result_code;
ERCALLBACK (GET_POSITION &position_def, &result_code);
```

バッファ

```
typedef struct
{
    char *position;
    long position_len;
    short position_type;
    short ascii_or_internal;
} position_def;
```

入力 次のようになります。

入力	説明
position_len	位置の長さへの割当ての長さ。
position_type	次のいずれかになります。 <ul style="list-style-type: none"> ◆ STARTUP_CHECKPOINT トレイルの開始位置。 ◆ CURRENT_CHECKPOINT トレイルの最後の読取り位置。
ascii_or_internal	列値が渡されたフォーマットのインジケータ。現在はデフォルトの Oracle GoldenGate 正規フォーマットのみサポートされており、次のように表されます。 EXIT_FN_INTERNAL_FORMAT

出力

出力	説明
*position	位置の値を表すバッファへのポインタ。このバッファは、position_def で、char フィールドの 8 バイトの seqnorba として、2 つのバイナリ値（符号なし int32t および int32t）として宣言されます。ユーザー・イグジットは、このデータを正しいデータ型に移行する必要があります。リトル・エンディアン・プラットフォームでこのファンクションを使用すると、プロセスは 2 つのそれぞれのフィールドで " バイトを反転 " させます。

返される値

```
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_NOT_SUPPORTED
EXIT_FN_RET_OK
```

GET_RECORD_BUFFER

適用対象 Extract および Replicat

GET_RECORD_BUFFER ファンクションでは、カスタム列変換に関する情報を取得します。MAP または TABLE パラメータの COLMAP オプションでは十分に対応できないときは、ユーザー・イグジットを使用して異なるソースおよびターゲット・レコード間でデータをマッピングできます。たとえば、ユーザー・イグジットを使用して Enscribe データベースの固有の日付フィールド（たとえば YYDDD）をターゲット・レコードで標準の SQL 日付に変換し、他の列は COLMAP オプションを使用して Extract プロセスでマップできます。

SET_RECORD_BUFFER ファンクション（492 ページを参照してください）では、GET_RECORD_BUFFER で取得したデータを変更できます。ただし、これには内部 Oracle GoldenGate 正規フォーマットで書き込まれたデータ・レコードに対する理解が必要です。かわりの方法として、SET_COLUMN_VALUE_BY_INDEX ファンクション（487 ページを参照してください）または SET_COLUMN_VALUE_BY_NAME ファンクション（489 ページを参照してください）を使用して、データ・レコードの列値を設定できます。

削除、挿入、および更新は、バッファに完全なレコード・イメージで保持されます。

圧縮 SQL 更新は、次のフォーマットを持ちます。

```
<index><length><value>[<index><length><value>][...]
```

条件:

- <index> は、表の列のリストへの 2 バイトの索引（最初の列はゼロ）です。
- <length> は、表の 2 バイトの長さです。
- <value> は、実際の列値で、適切な場合に次の 2 バイトの NULL インジケータの 1 つが含まれます。0 は非 NULL です。-1 は NULL です。

SQL レコードの場合は、DECOMPRESS_RECORD ファンクション（446 ページを参照してください）を使用して操作のためにレコードを解凍した後、COMPRESS_RECORD ファンクション（445 ページを参照してください）を使用して、プロセスに予期されるとおりにもう一度圧縮できます。

圧縮 Enscribe 更新は、次のフォーマットを持ちます。

```
<offset><length><value>[<offset><length><value>][...]
```

条件:

- <offset> は変更された Enscribe レコードのデータ・フラグメントのオフセットです。
- <length> はフラグメント長です。
- <value> はデータです。フラグメントはフィールド境界にまたがることもあるため、（圧縮が無効かまたは FETCHCOMPS が使用されている場合を除き）完全なフィールドが常に取得されるわけではありません。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (GET_RECORD_BUFFER, &record, &result_code);
```

```
バッファ      typedef struct
                {
                char *table_name;
                char *buffer;
                long length;
                char before_after_ind;
                short io_type;
                short record_type;
                short transaction_ind;
                int64_t timestamp;
                exit_ts_str io_datetime;
                short mapped;
                short source_or_target;
                /* Version 2 CALLBACK_STRUCT_VERSION */
                char requesting_before_after_ind;
                } record_def;
```

入力 次のようになります。

入力	説明
source_or_target	ソースまたはターゲット・データ・レコードのどちらのレコード・バッファを返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL
requesting_before_after_ind	オプションです。io_type が UPDATE_COMP_PK_SQL_VAL（主キー更新）のレコードのレコード・バッファをリクエストしているときに設定します。次のいずれかを使用して、主キー更新のどの部分にアクセスするかを指定します。デフォルトは AFTER_IMAGE_VAL です。 BEFORE_IMAGE_VAL AFTER_IMAGE_VAL

出力 次のようになります。

出力	説明
buffer	レコード・バッファへのポインタ。通常 buffer は、exit_rec_buf_def タイプのバッファへのポインタです。exit_rec_buf_def バッファには、Extract または Replicat にまもなく処理される実際のレコードが含まれます。このバッファは、コール・タイプが EXIT_CALL_DISCARD_RECORD のときに指定されます。イグジット・ルーチンは、たとえばカスタム・マッピング・ファンクションを実行するために、このバッファの内容を変えることがあります。
length	返されるレコード・バッファの長さ。
io_type	次のいずれかが返されます。 DDL タイプ SQL_DDL_VAL DML タイプ DELETE_VAL INSERT_VAL UPDATE_VAL 圧縮 Enscribe 更新 UPDATE_COMP_ENSCRIBE_VAL 圧縮 SQL 更新 UPDATE_COMP_SQL_VAL UPDATE_COMP_PK_SQL_VAL その他 TRUNCATE_TABLE_VAL
mapped	これがマップされたレコード・バッファかどうかを示すフラグ (0 または 1)。
before_after_ind	レコードがビフォア・イメージかアフター・イメージかを示す次の一方。 BEFORE_IMAGE_VAL AFTER_IMAGE_VAL

返される値 EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK

GET_RECORD_LENGTH

適用対象 Extract および Replicat

GET_RECORD_LENGTH ファンクションでは、データ・レコードの長さを取得します。かわりの方法として、GET_RECORD_BUFFER ファンクションでもデータ・レコードの長さを取得できます。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (GET_RECORD_LENGTH, &record, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
} record_def;
```

入力 次のようになります。

入力	説明
source_or_target	ソースまたはターゲット・データ・レコードのどちらのレコード長を返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
length	返されるデータ・レコードの長さ。

返される値

```
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
```

GET_RECORD_TYPE

適用対象 Extract および Replicat

GET_RECORD_TYPE ファンクションでは、処理されているレコードのタイプを取得します。レコードは、SQL または Enscribe レコードです。各レコード・タイプはフォーマットが異なるため、レコード・バッファを操作するときにレコード・タイプは重要です。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (GET_RECORD_TYPE, &record, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
} record_def;
```

入力 次のようになります。

入力	説明
source_or_target	ソースまたはターゲット・データ・レコードのどちらのレコード・タイプを返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
record_type	返されるレコード・タイプ。次のいずれかになります。 SQL レコードの場合： EXIT_REC_TYPE_SQL Enscribe レコードの場合： EXIT_REC_TYPE_ENSCRIBE

返される値

EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK

GET_STATISTICS

適用対象 Extract および Replicat

GET_STATISTICS ファンクションでは、Extract または Replicat プロセスの現在の処理統計を取得します。たとえばユーザー・イグジットは、Extract または Replicat 処理中に致命的なエラーが発生した場合に、カスタム・レポートに統計を出力できます。

統計は、データをリクエストしたプロセスのタイプに基づいて自動的に処理されます。

- **Extract** プロセスは、常にリクエストをソース表として扱い、出力回数にかかわらずこの表を 1 回カウントします。
- **Replicat** プロセスは、常にリクエストをターゲット表のセットとして扱います。このセットには、ソース表の数にかかわらず、ターゲットへのすべてのカウントが含まれます。

構文

```
#include "usrdecs.h"
short result_code;
statistics_def statistics;
ERCALLBACK (GET_STATISTICS, &statistics, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    short group;
    exit_timestamp_string start_datetime;
    long num_inserts;
    long num_updates;
    long num_befores;
    long num_deletes;
    long num_discards;
    long num_ignores;
    long total_db_operations;
    long total_operations;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    long num_truncates;
} statistics_def;
```


入力 次のようになります。

入力	説明
table_name	ソース表名を指定する NULL 終了文字列。統計は常にソース・レコードに対して記録されます。
group	次のいずれかになります。
EXIT_STAT_GROUP_STARTUP	Oracle GoldenGate プロセスが最後に起動したときからの統計を取得します。
EXIT_STAT_GROUP_DAILY	現在の日付の開始からの統計を取得します。
EXIT_STAT_GROUP_HOURLY	現在の時間の開始からの統計を取得します。
EXIT_STAT_GROUP_RECENT	GGSCI を使用して統計をリセットしてからの統計を取得します。
EXIT_STAT_GROUP_REPORT	最後のレポートが生成されてからの統計を取得します。
EXIT_STAT_GROUP_USEREXIT	ユーザー・イグジットが RESET_USEREXIT_STATS を使用して最後に統計をリセットしてからの統計を取得します。

出力 次のようになります。

出力	説明
start_datetime	指定のグループに対して統計の記録が開始されたローカルの日付と時刻を示す YYYY-MM-DD HH:MI:SS フォーマットの NULL 終了文字列。
num_inserts	返される Extract または Replicat が処理した挿入数。
num_updates	返される Extract または Replicat が処理した更新数。
num_befores	返される Extract または Replicat が処理した更新のビフォア・イメージ数。
num_deletes	返される Extract または Replicat が処理した削除数。
num_discards	返される Extract または Replicat が破棄したレコード数。
num_ignores	返される Extract または Replicat が無視したレコード数。
total_db_operations	返される Extract または Replicat が処理したデータベース操作合計数。
total_operations	返される Extract または Replicat が処理した合計操作数（破棄および無視を含む）。
num_truncates	返される Extract または Replicat が処理した切捨て数。

返される値

```
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_TABLE_NOT_FOUND
EXIT_FN_RET_OK
```

GET_TABLE_COLUMN_COUNT

適用対象 Extract および Replicat

GET_TABLE_COLUMN_COUNT ファンクションでは、表内の列の合計数（キー列数を含む）を取得します。

構文

```
#include "usrdecs.h"
short result_code;
table_def table;
ERCALLBACK (GET_TABLE_COLUMN_COUNT, &table, &result_code);
```

バッファ

```
typedef struct
{
    short num_columns;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    short num_key_columns;
} table_def;
```

入力 次のようになります。

入力	説明
source_or_target	ソースまたはターゲット表のどちらの列合計数を返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
num_columns	返される指定の表の列の合計数。
num_key_columns	返される Oracle GoldenGate に指定の表のキーとして使用されている列の合計数。

返される値

```
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
```

GET_TABLE_METADATA

適用対象 Extract および Replicat

GET_TABLE_METADATA ファンクションでは、処理中のレコードに関連付けられている表のメタデータを取得します。

構文

```
#include "usrdecs.h"
short result_code;
table_metadata_def tbl_meta_rec;
ERCALLBACK (GET_TABLE_METADATA, &tbl_meta_rec, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    short value_truncated;
    long max_name_length;
    long actual_name_length;
    short num_columns;
    short num_key_columns;
    short *key_columns;
    short num_keys_returned;
    BOOL using_pseudo_key;
    short source_or_target;
} table_metadata_def;
```

入力 次のようになります。

入力	説明
table_name	table_name 戻り値を受け付けるバッファへのポインタ。
key_columns	key_columns 索引の配列へのポインタ。
max_name_length	返される表名の最大長。通常、最大長は表名のバッファの長さです。返される表名は NULL で終了しているため、最大長は表名の最大長と同じになるはずです。
source_or_target	ソースまたはターゲット表のどちらの名前を返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
table_name	処理中のレコードに関連付けられている表の名前。
value_truncated	値が切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、表名と NULL 終了文字を足した長さが、最大バッファ長を超えるときに行われます。
actual_name_length	返される表名の文字列長。実際の長さに NULL 終了文字は含まれません。
num_columns	表内の列数。
num_key_columns	Oracle GoldenGate に使用されているキー内の列数。
key_columns	キー列の値。キー数に列の長さを掛けた値を把握し、適切な量のバッファを割当ててする必要があります。
num_keys_returned	リクエストされているキー列の数。
using_pseudo_key	KEYCOLS で指定されている列がキーとして使用されているかどうかを示すフラグ。TRUE または FALSE を返します。

返される値

```

EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_EXCEEDED_MAX_LENGTH
EXIT_FN_RET_OK

```

GET_TABLE_NAME

適用対象 Extract および Replicat

GET_TABLE_NAME ファンクションでは、処理中のレコードに関連付けられているソースまたはターゲット表の名前を取得します。

構文

```

#include "usrdecs.h"
short result_code;
env_value_def env_value;
ERCALLBACK (GET_TABLE_NAME, &env_value, &result_code);

```

バッファ

```

typedef struct
{
    char *buffer;
    long max_length;
    long actual_length;
    short value_truncated;
    short index;
    short source_or_target;
} env_value_def;

```

入力 次のようになります。

入力	説明
buffer	返された表名を受け付けるバッファへのポインタ。表名は NULL で終了します。
max_length	表名を受け付けるために割り当てたバッファの最大長。これは NULL 終了文字列として返されます。
source_or_target	ソースまたはターゲット表のどちらの名前を返すかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL

出力 次のようになります。

出力	説明
buffer	NULL で終了している表名。
actual length	返される表名の文字列長。実際の長さに NULL 終了文字は含まれません。
value_truncated	値が切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、表名と NULL 終了文字を足した長さが、最大バッファ長を超えるときに行われます。

返される値

```
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
```

GET_TIMESTAMP

適用対象 Extract および Replicat

GET_TIMESTAMP ファンクションでは、ソース・データ・レコードに関連付けられている I/O タイムスタンプを ASCII 日付時刻フォーマットで取得します。タイムスタンプは、その後ローカル時間に変換され、元のデータベース操作の時間に概算されます。

注意 ASCII コミット・タイムスタンプは、地域によって異なる夏時間の使用によって変化する可能性があります。ユーザー・イグジット・コールバックは、この差異を回避するために、GMT 時間として ASCII 日付時刻を返す必要があります。Oracle GoldenGate トレイルは、GMT フォーマットを使用します。[GET_GMT_TIMESTAMP](#) を参照してください。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (GET_TIMESTAMP, &record, &result_code);
```

バッファ typedef struct
 {
 char *table_name;
 char *buffer;
 long length;
 char before_after_ind;
 short io_type;
 short record_type;
 short transaction_ind;
 int64_t timestamp;
 exit_ts_str io_datetime;
 short mapped;
 short source_or_target;
 /* Version 2 CALLBACK_STRUCT_VERSION */
 char requesting_before_after_ind;
 } record_def;

入力 なし

出力 次のようになります。

出力	説明
timestamp	返される ASCII フォーマットの 64 ビット I/O タイムスタンプ。
io_datetime	ローカル I/O 日付および時刻を含む、次のフォーマットの NULL 終了文字列。 YYYY-MM-DD HH:MI:SS.FFFFFFFF

返される値 EXIT_FN_RET_INVALID_CONTEXT
 EXIT_FN_RET_OK

GET_TRANSACTION_IND

適用対象 Extract および Replicat

GET_TRANSACTION_IND ファンクションでは、データ・レコードがトランザクションの最初、最後、または中間の操作かを確認します。これは、たとえばユーザー・イグジットが各トランザクションの詳細を作成し、特別の概要レコードを出力するときに役立ちます。

構文 #include "usrdecs.h"
 short result_code;
 record_def record;
 ERCALLBACK (GET_TRANSACTION_IND, &record, &result_code);

バッファ typedef struct
 {
 char *table_name;
 char *buffer;
 long length;
 char before_after_ind;
 short io_type;
 short record_type;
 short transaction_ind;
 int64_t timestamp;
 exit_ts_str io_datetime;
 short mapped;
 short source_or_target;
 /* Version 2 CALLBACK_STRUCT_VERSION */
 char requesting_before_after_ind;
 } record_def;

入力 なし

出力 次のようになります。

出力	説明
transaction_ind	返されるトランザクション・インジケータで、次のいずれかで表されます。
BEGIN_TRANS_VAL	レコードはトランザクションの最初です。
MIDDLE_TRANS_VAL	レコードはトランザクションの中間です。
END_TRANS_VAL	レコードはトランザクションの最後です。
WHOLE_TRANS_VAL	レコードはトランザクションの唯一のレコードです。

返される値 EXIT_FN_RET_INVALID_CONTEXT
 EXIT_FN_RET_OK

GET_USER_TOKEN_VALUE

適用対象 Extract および Replicat

GET_USER_TOKEN_VALUE ファンクションでは、トレイル・レコードからユーザー・トークン値を取得します。

構文 `#include "usrdecs.h"`

バッファ

```
typedef struct
{
    char *token_name;
    char *token_value;
    long max_length;
    long actual_length;
    short value_truncated;
} token_value_def;
```

入力 次のいずれかになります。

入力	説明
token_name	トークン名を表すバッファへのポインタ。
max_length	結果のトークン値を受け付けるために割り当てた token_name バッファの最大長。これは NULL 終了文字列として返されます。

出力 次のようになります。

入力	説明
token_value	(ある場合) トークンの戻り値を表すバッファへのポインタ。
actual_length	返されるトークン値の実際の長さ。トークンが見つかっていても値が存在しないときは、値 0 が返されます。
value_truncated	トークン値が切り捨てられたかどうかを示すフラグ (0 または 1)。切捨ては、表名と NULL 終了文字を足した長さが、最大バッファ長を超えたときに行われます。

返される値

```
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_TOKEN_NOT_FOUND
EXIT_FN_RET_OK
```


OUTPUT_MESSAGE_TO_REPORT

適用対象 Extract および Replicat

OUTPUT_MESSAGE_TO_REPORT ファンクションでは、レポート・ファイルにメッセージを出力します。

構文

```
#include "usrdecs.h"
short result_code;
char message[500];
ERCALLBACK (OUTPUT_MESSAGE_TO_REPORT, message, &result_code);
```

バッファ なし

入力 次のようになります。

入力	説明
message	NULL 終了文字列。

出力 なし

返される値 EXIT_FN_RET_OK

RESET_USEREXIT_STATS

適用対象 Extract および Replicat

RESET_USEREXIT_STATS ファンクションでは、最後の GET_STATISTICS へのコールが処理された後の Oracle GoldenGate プロセスの EXIT_STAT_GROUP_USEREXIT 統計をリセットします。このファンクションにより、ユーザー・イグジットは GET_STATISTICS ファンクションによって返されるグループ統計をいつリセットするかを制御できますが、他の統計のリセットは許可されません。

構文

```
#include "usrdecs.h"
short result_code;
call_callback (RESET_USEREXIT_STATS, NULL, &result_code);
```

入力 なし

出力 なし

返される値 なし

SET_COLUMN_VALUE_BY_INDEX

適用対象 Extract および Replicat

SET_COLUMN_VALUE_BY_INDEX または SET_COLUMN_VALUE_BY_NAME ファンクションでは、データ・レコード全体を操作せず、単一の列値のみを変更します。

構文

```
#include "usrdecs.h"
short result_code;
column_def column;
ERCALLBACK (SET_COLUMN_VALUE_BY_INDEX, &column, &result_code);
```

```

バッファ      typedef struct
                {
                char *column_value;
                unsigned short max_value_length;
                unsigned short actual_value_length;
                short null_value;
                short remove_column;
                short value_truncated;
                short column_index;
                char *column_name;
                short ascii_or_internal;
                short source_or_target;
                /* Version 2 CALLBACK_STRUCT_VERSION */
                char requesting_before_after_ind;
                char more_lob_data;
                } column_def;
    
```

入力 次のようになります。

入力	説明
column_value	新しい列値を表すバッファへのポインタ。
actual_value_length	新しい列値の長さ。新しい列値が ASCII フォーマットの場合、実際の長さに NULL 終了文字は含めません。
null_value	新しい列値が NULL かどうかを示すフラグ (0 または 1)。null_value フラグが 1 に設定されている場合、データ・レコードの列値は NULL に設定されます。
remove_column	存在する場合に、圧縮更新の列を削除するかどうかを示すフラグ (0 または 1)。このフラグは、レコードの操作タイプが UPDATE_COMP_SQL_VAL、PK_UPDATE_SQL_VAL、または UPDATE_COMP_ENSCRIBE_VAL の場合にのみ設定する必要があります。
column_index	データ・レコード・バッファにコピーする新しい列値の列索引。列索引はゼロから開始されます。
ascii_or_internal	<p>新しい列が ASCII またはデフォルトの Oracle GoldenGate 正規フォーマットのどちらで渡されるかを示す次の一方。</p> <p>EXIT_FN_ASCII_FORMAT EXIT_FN_INTERNAL_FORMAT</p> <p>Oracle GoldenGate 正規フォーマットには、適切な場合に、2 バイトの NULL インジケータおよび 2 バイトの変数データ長を含める必要があります。</p>
source_or_target	<p>ソースまたはターゲット・レコードが変更されるかを示す次の一方。</p> <p>EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL</p>

入力	説明
requesting_before_after_ind	<p>io_type が UPDATE_COMP_PK_SQL_VAL（主キー更新）のレコードの列値を設定するときに設定します。次のいずれかを使用して、主キー更新のどの部分にアクセスするかを指定します。デフォルトは AFTER_IMAGE_VAL です。</p> <ul style="list-style-type: none"> ◆ BEFORE_IMAGE_VAL ◆ AFTER_IMAGE_VAL

出力 なし

返される値

```
EXIT_FN_RET_BAD_COLUMN_DATA
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
EXIT_FN_RET_NOT_SUPPORTED
EXIT_FN_RET_INVALID_COLUMN_TYPE
```

SET_COLUMN_VALUE_BY_NAME

適用対象 Extract および Replicat

SET_COLUMN_VALUE_BY_NAME または SET_COLUMN_VALUE_BY_INDEX ファンクションでは、データ・レコード全体を操作せず、単一の列値のみを変更します。

構文

```
#include "usrdecs.h"
short result_code;
column_def column;
ERCALLBACK (SET_COLUMN_VALUE_BY_NAME, &column, &result_code);
```

バッファ

```
typedef struct
{
    char *column_value;
    unsigned short max_value_length;
    unsigned short actual_value_length;
    short null_value;
    short remove_column;
    short value_truncated;
    short column_index;
    char *column_name;
    short ascii_or_internal;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
    char more_lob_data;
} column_def;
```

入力 次のようになります。

入力	説明
column_value	新しい列値を表すバッファへのポインタ。
actual_value_length	新しい列値の長さ。新しい列値が ASCII フォーマットの場合、実際の長さに NULL 終了文字は含めません。
null_value	新しい列値が NULL かどうかを示すフラグ (0 または 1)。null_value フラグが 1 に設定されている場合、データ・レコードの列値は NULL に設定されます。
remove_column	存在する場合に、圧縮更新の列を削除するかどうかを示すフラグ (0 または 1)。このフラグは、レコードの操作タイプが UPDATE_COMP_SQL_VAL、PK_UPDATE_SQL_VAL、または UPDATE_COMP_ENSCRIBE_VAL の場合にのみ設定する必要があります。
column_name	データ・レコード・バッファにコピーする新しい列値に対応する列名。
ascii_or_internal	新しい列が ASCII または Oracle GoldenGate 内部正規フォーマットのどちらで渡されるかを示す次の一方。 EXIT_FN_ASCII_FORMAT EXIT_FN_INTERNAL_FORMAT Oracle GoldenGate 正規フォーマットには、適切な場合に、2 バイトの NULL インジケータおよび 2 バイトの変数データ長を含める必要があります。
source_or_target	ソースまたはターゲット・データ・レコードが変更されるかを示す次の一方。 EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL
requesting_before_after_ind	io_type が UPDATE_COMP_PK_SQL_VAL (主キー更新) のレコードの列値を設定するときに設定します。次のいずれかを使用して、主キー更新のどの部分にアクセスするかを指定します。デフォルトは AFTER_IMAGE_VAL です。 ◆ BEFORE_IMAGE_VAL ◆ AFTER_IMAGE_VAL

出力 なし

返される値

EXIT_FN_RET_BAD_COLUMN_DATA
EXIT_FN_RET_INVALID_COLUMN
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
EXIT_FN_RET_NOT_SUPPORTED
EXIT_FN_RET_INVALID_COLUMN_TYPE

SET_OPERATION_TYPE

適用対象 Extract および Replicat

SET_OPERATION_TYPE ファンクションでは、レコードに関連付けられている操作のタイプを変更します。たとえば、特定の表での削除を別の表で挿入に変更できます。レコード・ヘッダーのビフォア/アフター・インジケータは、挿入および削除操作に対して適切に変更されます。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (SET_OPERATION_TYPE, &record, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
} record_def;
```

入力 次のようになります。

入力	説明
io_type	<p>削除、挿入、および更新に、次の 1 つがそれぞれ返されます。</p> <p>DELETE_VAL INSERT_VAL UPDATE_VAL</p> <p>圧縮 Enscribe 更新の場合は、次が返されます。</p> <p>UPDATE_COMP_ENSCRIBE_VAL</p> <p>圧縮 SQL 更新の場合は、次が返されます。</p> <p>UPDATE_COMP_SQL_VAL</p> <p>新しい操作タイプが挿入または削除の場合、レコードのビフォア/アフター・インジケータは次のいずれかに設定されます。</p> <p>挿入: AFTER_IMAGE_VAL (アフター・イメージ) 削除: BEFORE_IMAGE_VAL (ビフォア・イメージ)</p>
source_or_target	<p>ソースまたはターゲット・データ・レコードのどちらの操作タイプを設定するかを示す次の一方。</p> <p>EXIT_FN_SOURCE_VAL EXIT_FN_TARGET_VAL</p>

出力 なし

返される値

EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK

SET_RECORD_BUFFER

適用対象 Extract および Replicat

SET_RECORD_BUFFER ファンクションは、ユーザー・イグジットとの互換性の維持、および複雑なデータ・レコード操作のために使用します。このファンクションでは、レコード全体を操作します。データ・レコード・バッファを直接正確に変更するには、Oracle GoldenGate 内部レコード・フォーマットを把握する必要があるため、変更を行うときは、レコード全体ではなく個々の列値を変更する方法が最適です。列値を変更するには、SET_COLUMN_VALUE_BY_INDEX および SET_COLUMN_VALUE_BY_NAME ファンクションを使用します。ユーザー・イグジット内のこれらのファンクションで、ほとんどのカスタム・マッピングを十分に処理できます。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (SET_RECORD_BUFFER, &record_def, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
} record_def;
```

入力 次のようになります。

入力	説明
buffer	新しいレコード・バッファへのポインタ。通常、buffer はタイプ exit_rec_buf_def のバッファへのポインタです。exit_rec_buf_def バッファには、Extract または Replicat にまもなく処理される実際のレコードが含まれます。このバッファは、コール・タイプが EXIT_CALL_DISCARD_RECORD のときに指定されます。イグジット・ルーチンは、たとえばカスタム・マッピング・ファンクションを実行するために、このバッファの内容を変えることがあります。
length	新しいレコード・バッファの長さ。

出力 なし

返される値

```
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
EXIT_FN_RET_NOT_SUPPORTED
```

SET_TABLE_NAME

適用対象 Extract およびデータ・ポンプ

SET_TABLE_NAME ファンクションでは、レコードに関連付けられている表名を変更します。たとえば、特定の表での削除を履歴表への挿入に変更できます。表名は、Extract 処理中にのみ変更できます。

構文

```
#include "usrdecs.h"
short result_code;
record_def record;
ERCALLBACK (SET_TABLE_NAME, &record_def, &result_code);
```

バッファ

```
typedef struct
{
    char *table_name;
    char *buffer;
    long length;
    char before_after_ind;
    short io_type;
    short record_type;
    short transaction_ind;
    int64_t timestamp;
    exit_ts_str io_datetime;
    short mapped;
    short source_or_target;
    /* Version 2 CALLBACK_STRUCT_VERSION */
    char requesting_before_after_ind;
} record_def;
```

入力 次のようになります。

入力	説明
table_name	データ・レコードに関連付ける新しい表名を指定する NULL 終了文字列。

出力 なし

返される値

```
EXIT_FN_RET_INVALID_CONTEXT
EXIT_FN_RET_INVALID_PARAM
EXIT_FN_RET_OK
```


付録 1

Oracle GoldenGate コミット順序番号について

.....

Oracle GoldenGate で作業する際、コミット・シーケンス番号 (CSN) を参照する必要がある場合があります。CSN は、トランザクション・ログ内に **Extract** を配置する、証跡内に **Replicat** を再配置する、またはその他の目的に必要となることがあります。これは、一部の変換関数で返され、レポートおよび特定の GGSCI 出力内に含まれます。

CSN は、トランザクションにおける一貫性およびデータの整合性を維持する目的でトランザクションを識別するために Oracle GoldenGate によって構築される識別子です。トランザクションがデータベースにコミットされる特定の時間を一意に識別します。

各データベース管理システムでは、それぞれのトランザクションの完了時に、何らかの種類の一意のシリアル番号が独自に生成され、トランザクションが一意的に識別されます。CSN にはこの同じ識別情報が取得されます。CSN は内部では一連のバイトで表現されますが、プラットフォームに依存せずに処理されます。それぞれが、同じログ・ストリーム内のトランザクションのコミット・レコードにバインドされている CSN 番号のうち、いずれか 2 つを比較することで、この 2 つのトランザクションの完了順序が正しく示されます。

CSN 値は、トランザクションの開始を識別する証跡レコード内でトークンとして格納されます。この値は、@GETENV 列変換関数を使用して取得でき、Logdump ユーティリティを使用して表示できます。

表 65 Oracle GoldenGate データベースごとの CSN 値¹

データベース	CSN 値
c-tree	<log number>.<byte offset> 条件: <ul style="list-style-type: none">◆ <log number> は、10 進数の c-tree ログ・ファイルの番号で、先行 0(ゼロ) が埋め込まれます。◆ <byte offset> は、ファイル (0 ベース) の先頭部分からの 10 進数の相対バイト位置で、先行 0(ゼロ) が埋め込まれます。 例: 0000000068.0000004682
DB2 for i	DB2 for i について、Oracle GoldenGate による抽出 (取得) はサポートされていないので、このデータベースの CSN はありません。
DB2 LUW	<LSN> 条件: <ul style="list-style-type: none">◆ <LSN> は、可変長、10 進数ベースの DB2 ログ・シーケンス番号です。 例: 1234567890

表 65 Oracle GoldenGate データベースごとの CSN 値¹ (続き)

データベース	CSN 値
DB2 z/OS	<p><RBA></p> <p>条件:</p> <ul style="list-style-type: none">◆ <RBA> は、トランザクション・ログ内にあるコミット・レコードの 6 バイトの相対バイト・アドレスです。 <p>例:</p> <p>1274565892</p>
MySQL	<p><LogNum>:<LogPosition></p> <p>条件:</p> <ul style="list-style-type: none">◆ <RBA> は、識別されるトランザクションの START TRANSACTION レコードが含まれているログ・ファイルの名前です。◆ <LogPosition> は、そのレコードのイベント・オフセット値です。イベント・オフセット値は、ログ・レコードのレコード・ヘッダー・セクションに格納されています。 <p>たとえば、ログ番号が 12 で、ログの位置が 121 の場合、CSN は次のようになります。</p> <p>000012:0000000000000121</p>
Oracle	<p><system change number></p> <p>条件:</p> <ul style="list-style-type: none">◆ <system change number> は、Oracle SCN 値です。 <p>例:</p> <p>6488359</p>
SQL/MX	<p><sequence number>.<RBA></p> <p>条件:</p> <ul style="list-style-type: none">◆ <sequence number> は、6 進数の NonStop TMF 監査証跡シーケンス番号で、先行 0(ゼロ) が埋め込まれます。◆ <RBA> は、そのファイル内における 10 進数の相対バイト・アドレスで、先行 0(ゼロ) が埋め込まれます。 <p>これら 2 つで、TMF マスター監査証跡 (MAT) 内の場所が指定されます。</p> <p>例:</p> <p>000042.0000068242</p>

表 65 Oracle GoldenGate データベースごとの CSN 値¹ (続き)

データベース	CSN 値
SQL Server	<p>データベースの返し方に従い、次のいずれかになります。</p> <ul style="list-style-type: none"> ◆ コロンで区切られた 16 進数の文字列で、先行 0(ゼロ) および 0x 接頭辞が埋め込まれます (8:8:4)。 ◆ コロンで区切られた 10 進数の文字列で、先行 0(ゼロ) が埋め込まれます (10:10:5)。 ◆ 0x 接頭辞付き、先行 0(ゼロ) なしのコロンで区切られた 16 進数の文字列 ◆ 先行 0(ゼロ) なしのコロンで区切られた 10 進数の文字列 ◆ 10 進数の文字列 <p>条件:</p> <ul style="list-style-type: none"> ◆ 最初の値は仮想ログ・ファイルの番号で、2 番目は仮想ログ内のセグメント番号、3 番目はエントリ番号です。 <p>例:</p> <pre>0X00000d7e:0000036b:01bd 0000003454:0000000875:00445 0Xd7e:36b:1bd 3454:875:445 3454000000087500445</pre>
Sybase	<p><time_high>.<time_low>.<page>.<row></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <time_high> および <time_low> は、ログ・ページのインスタンス ID を示します。各データベースのログ・ページのヘッダーに格納されます。<time_high> は 2 バイトで、<time_low> は 4 バイトで、それぞれ先行 0(ゼロ) が埋め込まれています。 ◆ <page> は、データベースの論理ページ番号で、0(ゼロ) が埋め込まれます。 ◆ <row> は行番号で、0(ゼロ) が埋め込まれます。 <p>これらのコンポーネントをあわせて、ログ・ストリームの一意の場所が示されます。timestamp-high の 2 バイトの整数に有効な範囲は 0 ～ 65535 です。timestamp-low の 4 バイトの整数の場合は 0 ～ 4294967295 です。</p> <p>例:</p> <pre>00001.0000067330.0000013478.00026</pre>
Teradata	<p><sequence ID></p> <p>条件:</p> <ul style="list-style-type: none"> ◆ <sequence ID> は、汎用固定長の印刷可能シーケンス ID です。 <p>例:</p> <pre>0x0800000000000000D700000021</pre>
TimesTen	<p>TimesTen について、Oracle GoldenGate による抽出 (取得) はサポートされていないので、このデータベースの CSN はありません。</p>

¹ Oracle、DB2 LUW および DB2 z/OS 以外のすべてのデータベース・プラットフォームには固定長の CSN があります。CSN は、必要に応じて先行 0(ゼロ) が埋め込まれます。Sybase CSN など、複数のフィールドが含まれている CSN を各フィールド内に埋め込むことができます。

用語集

.....

以下は、このドキュメントで使用されている用語の説明です。

用語	定義
異常終了 (abend)	異常な終了。コンピュータ・システム上で実行されているプロセスの障害または予期しない終了。
変更後イメージ (after image)	挿入または更新の実行後におけるデータベース内の行の値。
エイリアス Extract (alias Extract)	ソース・システムよりセキュアなネットワーク・ゾーン内に存在するターゲット・システムで動作する Extract グループ。エイリアス Extract の目的は、信頼度の低いソースに対してターゲットから TCP/IP 接続を開始することです。一度接続が確立されると、データはソース・システムで動作するパッシブ Extract グループによって通常どおり処理され、ネットワークを通じて転送されます。
追加モード (append mode)	証跡に対するデフォルトの書き込み方法。この方法では、Extract は、障害後に古いデータを上書きせずに、証跡ファイルに再読取りデータを追加します。
アーカイブ・ログ専用モード (Archived Log Only mode: ALO)	Extract の操作モード。プロセスは、本番データベース・システムまたはスタンバイ・データベース・システムのアーカイブ・トランザクション・ログから排他的に読取りを行うように構成されます。
バッチ Replicat 処理モード (batch Replicat processing mode)	バッチ・モードでは、Replicat は、同様の SQL 文を配列に編成し、それらを高速に適用します。Replicat は、メモリー・キュー内に複数の文をバッチとしてまとめ、各バッチを 1 回のデータベース操作で適用します。このモードの動作は、BATCHSQL パラメータによって制御されます。「標準 Replicat 処理モード」も参照してください。
監査証跡 (audit trail)	レプリケーションおよびリカバリの目的で、データベースに行われた変更を格納する NonStop Server システムのファイル。
バッチ実行 (batch run)	開始と終了が明確な 1 回かぎりの処理実行。特定の終了ポイントを持たないオンライン変更同期などの継続処理と対比されます。
変更前イメージ (before image)	SQL 操作がデータベース内の行に対して実行される前に、その行に存在していた値。
双方向同期 (bidirectional synchronization)	複数のデータベースおよびサーバー全体で負荷分散が行われます。この環境では、通常、異なるユーザーが同じデータセットを変更可能で、それらの変更が Oracle GoldenGate によって同期されます。

用語	定義
BLOB	「LOB」 を参照してください。
制限リカバリ (Bounded Recovery)	Extract リカバリ・システムの一部。制限リカバリによって、 Extract が予期せずに停止して再起動する場合に、 Extract の停止時点で存在していたオープン・トランザクションの数やそれらの経過時間にかかわらず、効率的なリカバリが保証されます。制限リカバリにより、 Extract が停止した時点までリカバリして通常の処理を再開するまでに要する最大時間の上限が設定されます。
コール元 (caller)	ユーザー・イグジット ・ルーチンを実行する Oracle GoldenGate プロセス。
正規形式 (canonical format)	Oracle GoldenGate が 証跡 または 抽出ファイル にデータを格納する場合に使用するデータ形式。この形式によって、 異機種 データベース間でデータを高速かつ正確に交換できます。
カスケード同期 (cascading synchronization)	データが ソース ・システムから 1 つ以上の 中間システム に送信され、さらにそれらのシステムから 1 つ以上の 同期 状態の他のシステムに送信される Oracle GoldenGate 構成。
変更同期 (change synchronization)	あるシステムのデータベースで行われたデータ変更を、1 つ以上の他のシステムに存在する同様のデータセットと 同期 するプロセス。
チェックポイント・ファイル (checkpoint file)	Oracle GoldenGate プロセスによって生成された チェックポイント を格納するディスク上のファイル。
チェックポイント表 (checkpoint table)	Replicat の チェックポイント を保持する ターゲット ・データベースに作成される表。オプションで、ディスク上の標準の チェックポイント ・ファイルと組み合わせて使用されます。
チェックポイント (checkpoints)	Oracle GoldenGate プロセスの現在の読み取り位置と書き込み位置を記録する内部インジケータ。チェックポイントは、 オンライン変更同期 でデータの正確性とフォルト・トレランスを保証するために、 Extract プロセスおよび Replicat プロセスによって使用されます。
CLOB	「LOB」 を参照してください。
CMDSEC ファイル (CMDSEC file)	GGSCI コマンド権限のルールを格納した Oracle GoldenGate ファイル。
Collector	TCP/IP を通じて Extract プロセスからデータを受信し、そのデータを ターゲット ・システムの 証跡 または 抽出ファイル に書き込むプロセス。
衝突 (collisions)	Oracle GoldenGate によってレプリケートされたデータ変更が ターゲット 表に適用される場合に、 ターゲット 行が欠落または重複していると発生するエラー。
列 (column)	データベース表によって記述されるエンティティに割り当てられた一連の属性のうちの 1 つ。たとえば、「従業員」というエンティティには、名前、住所および電話番号の列を使用できます。
列マップ (column map)	「マップ」 を参照してください。

用語	定義
列変換関数 (column-conversion functions)	データを選択または操作する目的で比較、テスト、計算などの処理を実行する Oracle GoldenGate 組込みの処理関数。
コミット (commit)	トランザクションを終了させ、そのトランザクション内の SQL 文で実行された変更を永続化するトランザクション制御文。
コミット順序番号 (Commit Sequence Number: CSN)	CSN は、トランザクション一貫性とデータ整合性を維持する目的でトランザクションを識別するために Oracle GoldenGate が作成する識別子です。CSN によって、トランザクションがデータベースにコミットされた特定の時点が一意に識別されます。CSN の構成および値は、トランザクションを生成したデータベースのタイプに応じて異なります。CSN は、データベースがトランザクションを識別する際に使用する一意の情報を取得して、それを内部的に一連のバイト列として表現します。ただし、Oracle GoldenGate は、CSN をプラットフォームに依存しない方法で処理します。
圧縮更新 (compressed update)	SQL の更新操作を記録する方法。この方法では、更新の結果として変更された列値のみがトランザクション・ログに記録されます。
競合解決 (conflict resolution)	同じ SQL 操作が (ほぼ) 同時に 2 つ以上のデータベースの同じ行に適用される場合の処理ルールおよびエラー処理ルールを提供する、双方向同期で使用される手順。
統合同期 (consolidated synchronization)	異なるデータを 2 つ以上のデータベースから 1 つの中央データベース (データ・ウェアハウスなど) にレプリケートするプロセス。
変換 (conversion)	「トランスフォーメーション」を参照してください。
データ定義ファイル (data definitions file)	「ソース定義ファイル」および「ターゲット定義ファイル」を参照してください。
データ・ポンプ (data pump)	抽出ファイルまたは証跡から読取りを行うセカンダリ Extract プロセス。証跡へのデータ移入は、データソースから読取りを行うプライマリ Extract プロセスによって行われます。
データソース (data source)	Oracle GoldenGate によって処理されるデータ変更のコンテナ。次のデータソースを使用できます。 <ul style="list-style-type: none"> ◆ データベースのトランザクション・ログ ◆ ベンダー・アクセス・モジュール

用語	定義
データソース名 (data source name: DSN)	<p>DSN では、データベースに対する ODBC 接続を定義します。DSN は、データベースに応じて、データベース名、データベース・ディレクトリ、データベース ODBC ドライバ名、データベース認証情報などの情報で構成されます。DSN によって、アプリケーションはアプリケーション・プログラム内の必須情報をエンコードせずにデータベースに接続できるため、Oracle GoldenGate などの外部アプリケーションでは DSN が必要です。</p> <p>DSN には次の 3 つのタイプがあります。</p> <ul style="list-style-type: none"> ◆ システム DSN は、マシンにアクセスする任意のエンティティで使用できます。これはシステム構成内に格納されます。 ◆ ユーザー DSN は、特定のユーザーのみが使用できます。これはシステム構成内に格納されます。 ◆ ファイル DSN は、.dsn 拡張子付きのテキスト・ファイルに格納されます。これは、必要な ODBC ドライバがインストールされている異なるシステム間で共有できます。
データ型 (data type)	<p>個々のデータについて、そのデータの種別およびそのデータに実行可能な操作の種別を識別する属性。たとえば、整数データ型は数値を、文字データ型は文字を格納します。</p>
DDL	<p>データ定義言語。データベースの構造を定義するデータであり、行、列、表、索引およびデータベース詳細（ファイルの場所、ユーザー、権限、記憶域パラメータなど）が含まれます。</p>
DEFGEN	<p>データ定義ファイルを生成する Oracle GoldenGate ユーティリティ。</p>
廃棄ファイル (discard file)	<p>失敗した SQL 操作に関する情報が含まれる Oracle GoldenGate ファイル。このファイルは、レコードを処理できない場合に作成されますが、DISCARDFILE パラメータがパラメータ・ファイル内に存在し、ファイルの場所が指定されている必要があります。</p>
DML	<p>データ操作言語。データベースのデータを取得および操作します。SQL の場合、その操作は選択、挿入、更新および削除です。</p>
DSN	<p>「データソース名」を参照してください。</p>
動的 Collector (dynamic Collector)	<p>Manager プロセスによって自動的に起動される Collector プロセス。静的 Collector と対比されます。</p>
EMSCNT	<p>Windows などのサポートされるオペレーティング・システムで発生した Oracle GoldenGate のシステム・エラー・メッセージを NonStop Server の EMS（イベント管理サブシステム）サーバーに配信する Oracle GoldenGate ユーティリティ。</p>
ENCKEYS ファイル (ENCKEYS file)	<p>暗号化鍵を格納した Oracle GoldenGate 参照ファイル。</p>
暗号化 (encryption)	<p>解読するためのパスワードまたは復号化コードを所有するユーザー以外には判読不可能な形式にデータをエンコードする方法。</p>

用語	定義
エラー・ログ (error log)	Oracle GoldenGate によって生成されたイベント、メッセージ、エラーおよび警告の処理を示すファイル。ファイル名は ggserr.log であり、Oracle GoldenGate のルート・ディレクトリに配置されます。
イベント・マーカ・システム (event marker system)	Oracle GoldenGate をカスタマイズして、フィルタ基準に適合するレコードに基づいて処理中に特定のアクションを実行するシステム。たとえば、特定のレコードが検出された場合にそのレコードをスキップするか、Oracle GoldenGate プロセスを停止できます。 「イベント・レコード」も参照してください。
イベント・レコード (event record)	特定のフィルタ基準を満たし、処理中に特定のアクションを起動するために使用されるトランザクション・ログ内のレコード。 「イベント・マーカ・システム」も参照してください。
例外マップ (exceptions map)	エラー処理専用で使用する特別な MAP パラメータ 。このパラメータは、エラーの後にのみ実行されて、エラー・データを 例外表 に送信します。
例外表 (exceptions table)	失敗した SQL 操作 に関する情報が 例外マップ の結果として書き込まれるデータベース 表 。エラー処理で使用されます。
Extract	データソース 、 ソース表 、ローカル 証跡 またはローカル・ファイルからデータを読み取る Oracle GoldenGate プログラム。 Extract は、 ターゲット・システム に配信するためにデータを処理します。プライマリ Extract は、データソースまたはデータベース表を読み取り、 データ・ポンプ Extract は、プライマリ Extract によってデータを移入されたローカル証跡を読み取ります。
抽出ファイル (extract file)	Oracle GoldenGate によって書き込まれるファイルであり、 バッチ実行 または 初期ロード による後続の処理を一時的に待機しているデータが格納されます。
抽出 (extraction)	後続の処理または ターゲット・データベース への転送（あるいはその両方）に備えてデータベース 表 または データソース からデータを読み取る処理。
フェッチ (fetch)	トランザクション・ログのレコードを処理する場合に Extract プロセスによってデータベースに発行される問合せ。フェッチは、 SQL 操作 を完了するために必要なデータ値がレコードに存在しない場合に必要です。
ファイル・ヘッダー (file header)	「 ヘッダー 」を参照してください。
フィルタリング (filtering)	抽出 または レプリケーション のためにデータを選択および除外するルールを使用すること。
関数 (function)	アプリケーションまたは ルーチン 内で実行できるコード部分。「 列変換関数 」も参照してください。
GGSCI	GoldenGate ソフトウェア・コマンド・インタフェース。Oracle GoldenGate を構成、制御および監視するコマンドを発行するための主要インタフェースです。

用語	定義
GLOBALS ファイル (GLOBALS file)	Oracle GoldenGate インスタンスに全体的に適用されるパラメータを格納した Oracle GoldenGate のルート・ディレクトリにあるテキスト・ファイル。 Extract や Replicat などのプロセスに固有のランタイム・パラメータと対比されます。
グループ (group)	プロセス・グループとも呼ばれます。グループは、Oracle GoldenGate プロセス (Extract または Replicat) と、そのプロセスに関連するパラメータ・ファイル、 チェックポイント ・ファイルおよびその他のファイルで構成されます。
ヘッダー (header)	ヘッダーには次の種類があります。 <ul style="list-style-type: none"> ◆ レコード・ヘッダー: レコードのトランザクション環境に関する情報を格納した Oracle GoldenGate の証跡ファイルに含まれるレコードの先頭部分にある領域。 ◆ ファイル・ヘッダー: 証跡の各ファイルの先頭部分または抽出ファイルの先頭部分にある領域。このヘッダーには、Oracle GoldenGate のリリースなど、ファイル自体の情報が含まれます。
異機種 (heterogeneous)	データの交換が、異なるタイプのアプリケーション間、異なるタイプのデータベース間、異なるオペレーティング・システム間、またはこれらの組合せを対象に行われるデータ環境。
同機種 (homogeneous)	データの交換が、同一タイプのアプリケーション、データベースおよびオペレーティング・システム間で行われるデータ環境。
初期ロード (initial load)	2 つのデータベースを同一にするために ソース ・データを ターゲット ・データベースに複製すること。
中間システム (intermediary system)	ソース ・システムと ターゲット ・システム間の中継場所として機能するネットワーク上のシステム。このシステムは、 トランスフォーメーション などの追加処理アクティビティのホストに指定できます。
キー (key)	表 の行の一意の識別子として使用されるその表の 1 つ以上の 列 。Oracle GoldenGate では、 ターゲット ・データベースでの適切な行の検出と ソース ・データベースからのフェッチにこのキーが使用されます。Oracle GoldenGate でキーとして指定できるのは、 主キー 、 一意キー 、 代替キー 、または表のすべての列 (定義された識別子がない場合) です。
KEYCOLS	Oracle GoldenGate が 表 の特定の 行 を検索するために一意の識別子として使用する 1 つ以上の 列 を定義する TABLE 文または MAP 文の句。
KEYGEN	暗号化鍵 を生成する Oracle GoldenGate ユーティリティ。
ラグ (lag)	Extract ラグは、Extract によってレコードが処理された時刻と、 データソース におけるそのレコードのタイムスタンプとの間の差異です。 Replicat ラグは、Replicat によって 証跡 の最後のレコードが処理された時刻と、証跡におけるそのレコードのタイムスタンプとの間の差異です。
待機時間 (latency)	変更が ソース ・データで発生した時点と、その変更が ターゲット ・データに反映された時点との間の時間的差異。

用語	定義
LOB	ラージ・オブジェクト。大きすぎて文字フィールドに収まらない非構造化オブジェクト（Microsoft Word 文書や映像 / 音声ファイルなど）を表すデータベースのデータ型です。LOB のサブセットとして、文字データを格納する CLOB（キャラクタ・ラージ・オブジェクト）と、バイナリ・データを格納する BLOB（バイナリ・ラージ・オブジェクト）があります。
ログベース抽出 (log-based extraction)	データベースのトランザクション・ログからデータ変更を抽出する方法。
論理名 (logical name)	ストアド・プロシージャの実行インスタンスを表すそのプロシージャの名前。プロシージャの実際の名前と対比されます。たとえば、lookup という名前のプロシージャの論理名は、lookup1 や lookup2 などになります。
LUW	Linux、UNIX、Windows。これらのどのプラットフォームでも実行されるアプリケーションを説明する頭字語（DB2 LUW など）。
マクロ (macro)	パラメータおよびコマンドの実装などのタスクを自動化するコンピュータ・プログラム。
Manager	Oracle GoldenGate 処理のための制御プログラム。
マップ (map)	ソース・データのセットとターゲット・データのセットとの間の対応付け。マップには、データの選択基準と変換基準を含めることができます。これらのマップは、Replicat の MAP パラメータで指定します。
MAP 文 (MAP statement)	ソース表とターゲット表間の関係と、それらの表の処理ルールを指定する Replicat パラメータ。
マーカー (marker)	Extract および Replicat の処理に関連してアプリケーション固有のイベントを識別するために、NonStop Server の監査証跡に挿入されるレコード。 「イベント・マーカー・システム」も参照してください。
標準 Replicat 処理モード (normal Replicat processing mode)	Replicat のデフォルトの処理モード。標準モードでは、Replicat は、複数のソース・トランザクションによる操作を（トランザクション順に）蓄積し、それらをターゲットの 1 つのトランザクション内のグループとして適用することでパフォーマンスを向上します。GROUPTRANSOPS パラメータによってこのトランザクション内の操作の数を制御できますが、その境界は、グループの最後のトランザクションによるすべての操作が含まれるように Replicat によって自動的に調整される可能性があります。「バッチ Replicat 処理モード」および「ソース Replicat 処理モード」も参照してください。
オブジェクト (object)	このドキュメントにおいては、オブジェクトという語は、データの格納（表など）、所有権および権限の定義（ロールなど）、他のオブジェクトに対するアクションの実行（トリガーなど）といった目的でユーザーが認識および作成できるデータベースの任意の論理コンポーネントを示します。
オブジェクト・レコード (object record)	Oracle GoldenGate で処理するために構成された表および他のデータベース・オブジェクトの属性（列 ID やデータ型など）を格納したファイル。

用語	定義
ODBC	<i>Open Database Connectivity</i> 。アプリケーションが統一された方法で異なるタイプのデータベースに接続できるようにする標準インタフェースの頭字語です。ODBC の目的は、データベースに接続するプロセスをプログラミング言語、データベース・システムおよびオペレーティング・システムから分離することです。
オンライン変更同期 (online change synchronization)	Extract プロセスと Replicat プロセスが、Oracle GoldenGate ユーザーによって停止されないかぎり、データ変更を同期するために継続的に実行される Oracle GoldenGate の処理方法。オンライン・プロセスでは、 証跡 に チェックポイント が保持されます。
オンライン Extract (online Extract)	オンライン変更同期のために構成された Extract グループ。
オンライン処理 (online processing)	「 オンライン変更同期 」を参照してください。
オンライン Replicat (online Replicat)	オンライン変更同期のために構成された Replicat グループ。
操作 (operation)	単一の作業単位。通常は、データに行われる SQL 変更またはデータベースの オブジェクト 構造に行われる変更を示しますが、コンピュータ・プロセスによって実行される任意の作業を示すこともあります。
Oracle GoldenGate Director	<p>Oracle GoldenGate ユーザーによる Oracle GoldenGate プロセスの監視および管理を可能にするグラフィカル・ユーザー・インタフェース・ソフトウェア。</p> <p>Oracle GoldenGate Director には、次のコンポーネントがあります。</p> <p>Oracle GoldenGate Director Administrator: 管理者が Oracle GoldenGate のユーザーおよびインスタンスを定義するために使用するユーティリティ。</p> <p>Oracle GoldenGate Director Server: Oracle GoldenGate プロセスに関するデータを収集するソフトウェア・モジュール。</p> <p>Oracle GoldenGate Director Client: Oracle GoldenGate Director に対するインタフェースとしてユーザーのシステムにインストールされるソフトウェア。</p> <p>Oracle GoldenGate Director Web: Oracle GoldenGate Director に対するブラウザベースのユーザー・インタフェース (ソフトウェアのインストール不要)。</p>
Oracle GoldenGate ロー ルバック (Oracle GoldenGate Rollback)	変更前イメージ を使用して、データベースに加えられた変更を元に戻すユーティリティ。
上書きモード (overwrite mode)	リリース 10.0 より前のリリースの Oracle GoldenGate で使用されていた 証跡 へのデータの書き込み方法。このモードでは、 Extract は、リカバリ時に証跡ファイルの最後にデータを追加するのではなく、既存のデータを上書きします。

用語	定義
所有者 (owner)	データベース・オブジェクトが組織階層の一部として割り当てられるデータベースの論理ネームスペース。データベース・オブジェクトの所有権はデータベース・タイプごとに異なる方法で管理されるため、このドキュメントで使用される 所有者 という語は、オブジェクト名の修飾子としてデータベースによって認識されるすべてのエンティティ（通常はユーザー名またはスキーマ名）を示します。たとえば、修飾された Oracle 表名の scott.emp において、所有者は scott です。
パラメータ (parameter)	コンピュータ・プログラム (Oracle GoldenGate のようなアプリケーション、 ストアド・プロシージャ 、 マクロ 、スクリプトまたは他の処理命令のコードなど) の入力値または出力値。
パラメータ・ファイル (parameter file)	Oracle GoldenGate プロセスの動作を制御する パラメータ を格納したファイル。パラメータ・ファイルのデフォルトの場所は、Oracle GoldenGate のインストール・ディレクトリの dirprm ディレクトリです。
パススルー・データ・ポンプ (pass-through data pump)	データ定義を参照する必要性を避けるために、 PASSTHRU パラメータで構成された データ・ポンプ 。これにより、処理が高速化し、データベースが存在しない 中間システム でポンプを使用できます。
パススルー Extract (pass-through Extract)	「 パススルー・データ・ポンプ 」を参照してください。
パッシブ Extract (passive Extract)	エイリアス Extract が ターゲット で使用されている場合に、 ソース ・システムで動作する Extract プロセス。この Oracle GoldenGate 構成が必要になるのは、ターゲットがよりセキュアなネットワーク・ゾーン内に存在するためにセキュリティ・ルールで（通常の Extract が行うような）ソース・システムからの TCP/IP 接続の開始が許可されない場合です。パッシブ Extract は、使用中は データ・ポンプ になります。それ以外の場合、 プライマリ Extract になります。
プライマリ Extract (primary Extract)	データソース から、または直接データベース表から読取りを行う Extract グループ。プライマリ Extract は、後から データ・ポンプ Extract によって読み取られるローカル 証跡 に書き込みを行うことができます。または、TCP/IP を通じてデータを ターゲット ・システムに送信できます。
主キー (primary key)	現在および将来の 表 に存在する（可能性のある）すべての 行 を一意に識別する 1 つ以上の 列 で構成された整合性制約。1 つの表には 1 つの主キーのみを指定できます。主キーには、暗黙的な NOT NULL 制約が含まれます。
プロセス・レポート (process report)	プロセス構成と実行時の統計およびイベントに関する情報を提供する、 Extract 、 Replicat および Manager に対して生成されるレポート。プロセス・レポートのデフォルトの場所は、Oracle GoldenGate のインストール・ディレクトリの dirrpt ディレクトリです。
レコード (record)	データベースの 行 に対して実行された単一の SQL 操作 に関する情報を格納する トランザクション・ログ または 証跡 の情報単位。 レコード という語は、表の特定の行に含まれる情報を説明する場合にも使用されます。
レコード・ヘッダー (record header)	「 ヘッダー 」を参照してください。

用語	定義
リモート・ファイル (remote file)	リモート・システムの抽出ファイル。
リモート証跡 (remote trail)	リモート・システムの証跡。
Replicat	データをターゲット表に適用するか、データを別のアプリケーションまたは宛先に移動する Oracle GoldenGate プロセス。
レプリケーション (replication)	ソース・データベースの操作を再作成してターゲット・データベースに適用するプロセス。
レポート (report)	「プロセス・レポート」を参照してください。
レポート・ファイル (report file)	「プロセス・レポート」を参照してください。
ロールバック (rollback)	コミットされていないトランザクション内の SQL 文によって実行されたデータ変更を元に戻すアクション。
ロールオーバー (rollover)	一連のファイル内 (証跡など) の 1 つのファイルを閉じ、同じ一連のファイル内の新規ファイルを開くこと。
ルーチン (routine)	値の取得と返却を行ってレスポンスを戻す関数をコールする、Oracle GoldenGate などのアプリケーション内で実行されるコード部分。 「ユーザー・イグジット」も参照してください。
行 (row)	データベース表内に格納されるエンティティ (従業員など) の単一のインスタンスに関する情報。たとえば、John Doe に関する情報は、1 つの行に格納されますが、その行は会社の John およびその他の従業員に関する情報を格納するより広い範囲の行のコレクションに含まれます。行は、一般的にレコードとも呼ばれます。
ソース (source)	Oracle GoldenGate が抽出を行う元のデータの場所 (ソース・データベースやソース・システムなど)。
ソース定義ファイル (source definitions file)	ソース表の定義を格納したファイルであり、ターゲット・システムに転送されます。このファイルは、ソース表とターゲット表が異なる場合に、データ変換のために Replicat プロセスによって使用されます。
ソース Replicat 処理モード (source Replicat processing mode)	ソース処理モードでは、Replicat は、ソースで使用された範囲と同じトランザクション境界内で SQL 操作を適用します。「標準 Replicat 処理モード」も参照してください。
特別実行 (special run)	「バッチ実行」を参照してください。
文 (statement)	コンピュータ・プログラミング言語における基本命令 (SQL 文、パラメータ文、コマンド文など)。

用語	定義
静的 Collector (static Collector)	Manager プロセスによって自動的に起動されるかわりに、Oracle GoldenGate ユーザーによって手動で起動される Collector プロセス。
ストアド・プロシージャ (stored procedure)	データベースに格納され、ビジネス・ルールの適用、アプリケーション・ロジックの追加、または他の必要な作業の実行のためにプロセスまたはアプリケーションによって必要時にコールされる SQL、PL/SQL または Java の 文 のグループ。
代替キー (substitute key)	表の 行 を一意に識別できるその表内の任意の 列 で構成された一意識別子。代替キーは、表の定義では設定されません。TABLE 文または MAP 文に KEYCOLS 句を指定することで作成します。
同期 (synchronization)	2 つ以上のデータセットの一貫性を相互に確立または維持するプロセス。一貫性を確保するため、一方のセットは、他方と同一になるか、他方を再編成、再フォーマットまたは拡張したバージョンになりますが、情報それ自体の本質は維持されます。
表 (table)	行 および 列 で構成されたデータベースの記憶域の論理単位。行と列の組合せによって、特定のエンティティ (従業員など) のインスタンスと、そのエンティティの属性 (名前や住所など) が識別されます。
TABLE 文 (TABLE statement)	データベースからデータを 抽出 する 1 つ以上のソース表を指定する Extract パラメータ。
Teradata アクセス・モジュール (Teradata Access Module: TAM)	Teradata データベースの Change Data Capture (CDC) コンポーネントと Extract プロセス間のインタフェース。これにより、Oracle GoldenGate は、Teradata レプリケーション・コンポーネントと通信できます。
ターゲット (target)	Oracle GoldenGate によって処理されるデータの宛先 (ターゲット・データベースやターゲット・システムなど)。
ターゲット定義ファイル (target definitions file)	ターゲット 表の定義を格納したファイル。このファイルは、 ソース ・システムに転送され、ソース表とターゲット表が異なる場合に、データ変換のために Extract プロセスによって使用されます。
タスク (task)	Extract プロセスが、 Collector プロセスまたは 証跡 を使用せずに、TCP/IP を通じて Replicat プロセスと直接通信する バッチ実行 の特別なタイプ。
トークン (token)	Oracle GoldenGate の 証跡 ファイルに含まれる レコード の ヘッダー 部分に格納されたユーザー定義の情報。トークン・データを使用して、Oracle GoldenGate による情報の配信方法をカスタマイズできます。
トレース表 (trace table)	Oracle Database で Oracle GoldenGate が使用するために作成される特別な表。この表を パラメータ 設定と組み合わせて使用し、 双方向同期 構成で レプリケート されたデータが ソース に戻されることを防止します。
証跡 (trail)	後続の処理に備えて Oracle GoldenGate が一時的にデータを格納するディスク上の一連のファイル。Oracle GoldenGate は、 オンライン変更同期 のために証跡に チェックポイント を記録します。

用語	定義
トランザクション (transaction)	開始および終了のトランザクション制御文のセット内で論理的な作業単位として実行される 1 つ以上の SQL 操作 (または 文) のグループ。トランザクション内の各 SQL 文 は、すべて一体として実行に成功する必要があります。それ以外の場合、どの文も実行できません。トランザクションは、データおよび構造の整合性を確保するデータベース機能のシステムの一部です。
トランザクション・ログ (transaction log)	データのリカバリまたは レプリケーション を行う目的で、データベースに対して実行されたすべての SQL 変更操作 を記録する一連のファイル。
トランスフォーメーション (transformation)	「変換」とも呼ばれます。 ターゲット の表またはアプリケーションで必要とされる形式に ソース ・データを処理するプロセス (日付の変換や算術計算の実行など) です。Oracle GoldenGate 列変換関数 を使用してトランスフォーメーションを実行できます。
単方向同期 (unidirectional synchronization)	データ変更がソースからターゲットへと一方向にレプリケートされる構成。 双方向 構成のように同じデータが変更されてソースに戻されることはありません。
一意キー (unique key)	現在および将来の表に存在する (可能性のある) すべての 行 を一意に識別する 1 つ以上の 列 で構成された整合性制約。暗黙的な NOT NULL 制約が含まれない点で、 主キー とは異なります。1 つの表には複数の一意キーを指定できます。
作業単位 (Unit of Work)	データベースで論理的な単位として実行されるデータ操作のセット。すべての操作が成功する必要があります。それ以外の場合はどの操作も実行できません。IBM の用語では、 作業単位 という語は、他のタイプのデータベースにおける トランザクション という語と同義です。
ユーザー・イグジット (user exit)	カスタム処理 (データの変換、データベース・イベントに対するレスポンス、無効なデータの修復など) を実行するために Oracle GoldenGate の処理中にコールされる C プログラミング・コードで記述されたユーザー作成プログラム。
ベンダー・アクセス・モジュール (Vendor Access Module: VAM)	特定の種類のデータベースと通信するために Oracle GoldenGate プロセス・モジュールによって使用される API インタフェース。
VAM 証跡 (VAM trail)	トランザクション・ログと同様に、必要に応じて自動的に作成およびエージングされる一連のファイル。同時トランザクションによるデータ操作は、発生した順に時系列に記録されますが、必ずしもトランザクション順ではありません。Teradata の最大保護コミット・プロトコルをサポートするために使用されます。
ワイルドカード (wildcard)	不明または未指定の文字 (文字セット) のプレースホルダ。ワイルドカードは、 パラメータ文 または コマンド文 で複数の名前を指定する手段です。Oracle GoldenGate では、任意の数の不明な文字を表すアスタリスク・ワイルドカード (*) がサポートされます。

索引

記号

! コマンド 86

マクロ文字 215

数字

16 進数データ, バイナリに変換 420

A

ABEND オプション, REPEROR 241, 275

ADDTRANDATA オプション, DDLOPTIONS 160

ADD コマンド

CHECKPOINTTABLE 80

EXTRACT 17

EXTTRAIL 66

REPLICAT 47

RMTTRAIL 67

TRACETABLE 83

TRANDATA 74

AFTERCSN オプション, START REPLICAT 60

AFTERFILTER オプション, SQL EXEC 249, 342

AIXTHREAD_SCOPE 変数 356

ALLOCFILES パラメータ 114

ALLOWDUPTARGETMAP パラメータ 115

ALLOWLOBDATATRUNCATE オプション, DBOPTIONS 144

ALLOWNESTED パラメータ 88

ALLOWUNUSED COLUMN オプション, DBOPTIONS 144

ALLPARAMS オプション, SQLEXEC 249, 342

ALLPROCESSES オプション

INFO EXTRACT 32

INFO REPLICAT 54

STATUS EXTRACT 46

STATUS REPLICAT 65

ALL オプション, DDL 151, 156, 168

ALLOWARNEOF オプション, TRANLOGOPTIONS 370

ALTARCHIVEDLOGFORMAT オプション,
TRANLOGOPTIONS 361

ALTER コマンド

EXTRACT 25

EXTTRAIL 68

REPLICAT 50

RMTTRAIL 68

ALTID オプション, Extract 用 MAP 216

ALTONLINELOGS オプション, TRANLOGOPTIONS 363

APPEND オプション

DEFSFILE 173

DISCARDFILE 174

RMTFILE 285

APPEND ヒント, Oracle 207, 238

ARCHIVEDLOGONLY オプション, TRANLOGOPTIONS 364

ARSTATS オプション, SEND EXTRACT 41

ASCII

EBCDIC に変換 115

保存 190

無効, 置換 391

ASCIITOEBCDIC パラメータ 115

ASMBUFSIZE オプション, TRANLOGOPTIONS 364

ASMUSER オプション, TRANLOGOPTIONS 364

ASM インスタンス

代替 API 366

読取りバッファ・サイズ 366

ログイン 364

ASSUMETARGETDEFS パラメータ 115

ATCSN オプション, START REPLICAT 60

AT オプション

REPORT 281

REPORTROLLOVER 283

ROLLOVER 295

AUTORESTART パラメータ 116

AUTOSTART パラメータ 117

B

Base24 レコード, キーの関連付け 418

BATCHERRORMODE オプション, **BATCHSQL** 120

BATCHESPERQUEUE オプション, **BATCHSQL** 121

BATCHSQL パラメータ 118

BATCHTRANSOPS オプション, **BATCHSQL** 121

BCP/DTS, ファイル生成 195

BCP オプション, **FORMATASCII** 191

BEFOREFILTER オプション, **SQLEXEC** 249, 342

BEGIN

ADD EXTRACT オプション 20

ADD REPLICAT オプション 48

パラメータ 122

BINARYINPUT オプション, **MAP** 227

BINARY ファンクション 397

BINTOHEX ファンクション 398

BLOBMEMORY パラメータ 122

BLOWFISH 暗号化 288, 290

BOOTDELAYMINUTES パラメータ 122

Bounded Recovery 123

BRDIR オプション, **BR** 128

BRINTERVAL オプション, **BR** 128

BR パラメータ 123

BUFSIZE オプション, **TRANLOGOPTIONS** 364

BULKLOAD パラメータ 129

BYTESPERQUEUE オプション, **BATCHSQL** 121

C

CACHEBUFFERSIZE

オプション, **CACHEMGR** 133

統計 131

CACHEDIRECTORY オプション, **CACHEMGR** 134

CACHEMGR

オプション, **SEND EXTRACT** 34

パラメータ 129

CACHEPAGEOUTSIZE

オプション, **CACHEMGR** 134

統計 131

CACHEPOOL 統計, **SEND EXTRACT** 34

CACHEQUEUES 統計, **SEND EXTRACT** 34

CACHESIZE

オプション, **CACHEMGR** 133

統計 131

CACHESIZEMAX 統計 132

CACHESTATS 統計, **SEND EXTRACT** 34

CASE ファンクション 398

CHECKINTERVAL オプション, **WARNLONGTRANS** 386

CHECKMINUTES パラメータ 136, 270

CHECKPARAMS パラメータ 136

CHECKPOINTSECS パラメータ 137

CHECKPOINTTABLE

オプション, **ADD REPLICAT** 49

パラメータ 137

CHECKSEQUENCEVALUE パラメータ 138

CHECK 制約, **SQL Server** ターゲットでの抑止 148

CHILDSTATUS オプション, **SEND MANAGER** 15

CLEANUP コマンド

CHECKPOINTTABLE 81

EXTRACT 26

REPLICAT 50

CMDTRACE パラメータ 139

Collector プロセス・パラメータ 389

COLMAP オプション

MAP 224

TABLE 321

COLMATCH パラメータ 139

COLS (EXCEPT) オプション, **TABLE** 323

COLSTAT ファンクション 399

COLS オプション, **ADD TRANDATA** 77

COLTEST ファンクション 399

COMMENT パラメータ 140

COMMITTEDTRANLOG オプション, **DSOPTIONS** 176

COMPLETEARCHIVEDLOGONLY オプション,
TRANLOGOPTIONS 365

COMPLETEARCHIVEDLOGTIMEOUT オプション,
TRANLOGOPTIONS 366

COMPRESS_RECORD ファンクション 445

COMPRESSDELETES パラメータ 141

COMPRESSTHRESHOLD オプション

RMTHOST 288
RMTHOSTOPTIONS 290

COMPRESSUPDATES パラメータ 141**COMPRESS オプション**

RMTHOST 288
RMTHOSTOPTIONS 290

COMPUTE ファンクション 400**CONNECTIONPORT オプション, DBOPTIONS 145****-cp パラメータ 389****CREATE SUBDIRS コマンド 88****CREATETRANLOG オプション, DSOPTIONS 177****CSN, 「変更順序番号」を参照****c-tree**

Extract 開始位置 21
REPLICATE 属性, 有効化 74
サーバー別名, 指定 303, 354
トランザクション・バッファ
 受信レコードのタイムアウト 371
 調整 371
認証, 指定 380
レコードをフェッチするユーザー・イグジット 448, 449

CUSEREXIT パラメータ 142**D****DATA CAPTURE CHANGES 75, 368****DATE**

オプション, FORMATASCII 191
ファンクション 401

DATEDIFF ファンクション 404**DATENOW ファンクション 404****DB2**

ADD TRANDATA オプション 75
LONGVARCHAR のビフォア値 372
LONGVAR のビフォア値が記録されないときの警告 372
双方向の同期化 200
トランザクション・バッファ, 制御 364
トランザクション・メモリー, 管理 375
非特権 APF 属性の設定 372
ブートストラップ・データ・セット, ADD EXTRACT 19
行方不明の表 368
ログイン要件 380
ログ・バッファ, フラッシュの防止 370

DBENVIRONMENT オプション, @GETENV 416**DBLOGIN コマンド 72****DBLOGREADERBUFSIZE オプション, TRANLOGOPTIONS 366****DBLOGREADER オプション, TRANLOGOPTIONS 366****DBOPTIONS パラメータ 143****DBOP オプション, SQLEXEC 249, 343****DDL**

エラー, 処理 154
処理オプション, 設定 159
スキーマ, 指定 203
トレース 58, 358
フィルタリング 149
変更の警告を抑止 370
文字列置換 166
履歴
 ページ 266
 表示 85

DDLERROR パラメータ 154**DDLINCLUDE オプション**

SEND REPLICAT 57
TRACE/TRACE2 358

DDLONLY オプション

SEND REPLICAT 57
TRACE/TRACE2 358

DDLOPTIONS パラメータ 159**DDLSTBST パラメータ 166****DDLTABLE パラメータ 171****DDL の exclusion 句 150, 155, 168****DDL の inclusion 句 150, 155, 168****DDL パラメータ 149****DECOMPRESS_RECORD ファンクション 446**

DECRYPTTRAIL パラメータ 172, 179
DEFAULTUSERPASSWORD オプション, DDLOPTIONS 162
DEFERAPPLYINTERVAL パラメータ 172
DEFERREFCONST オプション, DBOPTIONS 145
DEFSFILE パラメータ 173
DEF オプション, MAP 228
DELETE コマンド
 CHECKPOINTTABLE 81
 EXTRACT 26
 EXTTRAIL 69
 REPLICAT 51
 RMTTRAIL 69
 TRACETABLE 84
 TRANDATA 79
DELIMITER オプション, FORMATASCII 191
DESC オプション
 ADD EXTRACT 23
 ADD REPLICAT 49
DETAIL オプション, INFO コマンド
 Extract 32
 Replicat 54
DIRECTORY オプション
 LOBMEMORY 212
 TRANSMEMORY 377
DISABLELOBCACHING オプション, DBOPTIONS 145
DISCARDFILE パラメータ 173
DISCARDROLLOVER パラメータ 174
DISCARD オプション, REPERROR 241, 275
DOWNCritical パラメータ 175
DOWNREPORT パラメータ 176
DSOPTIONS パラメータ 176
DUMPDDL パラメータ 85
DYNAMICPORTLIST パラメータ 177
DYNAMICPORTREASSIGNDELAY パラメータ 178
DYNAMICRESOLUTION パラメータ 178
DYNAMIC オプション WILDCARDRESOLVE 388
DYNSQL パラメータ 178
-d パラメータ 389

E

EBCDIC, 変換 115, 389
EDIT PARAMS コマンド 71, 96

EMPTYLOBSTRING オプション, DBOPTIONS 145
-ENCRYPT Collector パラメータ 390
ENCRYPT PASSWORD コマンド 73
ENCRYPTKEY オプション
 DDLOPTIONS 162, 164
 ENCRYPT PASSWORD 73
 USERID 383
ENCRYPTTRAIL パラメータ 179
ENCRYPT オプション
 RMTHOST 288
 RMTHOSTOPTIONS 290
END パラメータ 180
Enscribe
 単一列として定義されたレコード 227
 定義ファイル, 指定 355
EOFDELAY(CSECS) パラメータ 181
EOFDELAYMS オプション, THREADOPTIONS 357
EOF オプション, ADD EXTRACT 22
ERROR オプション, SQLEXEC 250, 343
ER コマンド 65
ETOLDFORMAT パラメータ 181
ETROLLOVER オプション, ALTER EXTRACT 25
EVAL ファンクション 405
EVENTACTIONS オプション
 MAP 228
 TABLE (Extract) 325
 TABLE (Replicat) 352
EVERY オプション, SQLEXEC 310
EXCEPTIONSONLY オプション, MAP 235
EXCEPTION オプション, REPERROR 241, 275
EXCLUDELIST オプション, SEND EXTRACT 41
EXCLUDELONG オプション, ADD TRANDATA 77
EXCLUDETRANS オプション, TRANLOGOPTIONS 200, 366
EXCLUDEUSERID オプション, TRANLOGOPTIONS 201, 368
EXCLUDEUSER オプション, TRANLOGOPTIONS 201, 367
EXCLUDE オプション
 DDL 150, 155, 168
 DDLSUBST 155, 167
EXEC オプション, SQLEXEC 250, 344
EXIT_CALL_ パラメータ
 RESULT 438
 TYPE 437

EXIT_PARAMS ファンクション 439

EXITPARAM オプション

MAP 236

TABLE 332

EXTFILE

オプション, ADD REPLICAT 48

パラメータ 182

EXTFILESOURCE

オプション, ADD EXTRACT 20

オプション, SPECIALRUN 306

EXTRACOLS オプション, **FORMATASCII** 191

Extract

起動 43

コマンドの概要 17

実行履歴, 削除 26

ステータス, 表示 46

中断 32

停止

 オンライン処理 46

 バッチ実行 180

統計, 表示 43

トレース 358

ラグ, 表示 27, 32

レポート, 表示 94

「Extract グループ」も参照

Extract グループ

最大数 17

削除 26

追加 17

パラメータ・ファイルに指定 184

変更 25

EXTRACT パラメータ 184

Extract 用 **MAP** パラメータ 216

EXTRBA オプション

ADD EXTRACT 21

ADD REPLICAT 49

EXTSEQNO オプション

ADD EXTRACT 21

ADD REPLICAT 49

EXTTRAIL

オプション, ADD REPLICAT 48

パラメータ 185

EXTTRAILSOURCE オプション

ADD EXTRACT 20

SPECIALRUN 306

-E パラメータ 389

-e パラメータ 390

F

FC コマンド 88

FETCH_CURRENT_RECORD_WITH_LOCK ファンクション 449

FETCH_CURRENT_RECORD ファンクション 448

FETCHBATCHSIZE オプション, **DBOPTIONS** 146

FETCHBEFOREFILTER オプション, **TABLE** 333

FETCHCOLS (EXCEPT) オプション, **TABLE** 332

FETCHLOBS オプション, **DBOPTIONS** 146

FETCHMODCOLS オプション, **TABLE** 333

FETCHOPTIONS パラメータ 187

FILE オプション, **TRACE/TRACE2** 358

FILTERDUPS パラメータ 188

FILTERTABLE オプション, **TRANLOGOPTIONS** 200, 368

FILTER オプション

MAP 236

TABLE 334

FLUSH(C)SECS パラメータ 189

FORCESTOP オプション

SEND EXTRACT 34

SEND REPLICAT 56

FORCETRANS オプション, **SEND EXTRACT** 34, 46

FORMATASCII パラメータ 190

FORMATSQL パラメータ 192

FORMATXML パラメータ 193

FORMAT オプション

EXTFILE 184

EXTTRAIL 186

RMTFILE 286

RMTTRAIL 292, 294

FREQUENCY オプション, **PURGEOLDEXTRACTS** 267, 269, 271

FUNCTIONSTACKSIZE パラメータ 194

-f パラメータ 391

G

GENLOADFILES パラメータ 195

GET_ ファンクション

BEFORE_AFTER_IND ファンクション 450
 COL_METADATA_FROM_INDEX 451
 COL_METADATA_FROM_NAME 454
 COLUMN_INDEX_FROM_NAME 456
 COLUMN_NAME_FROM_INDEX 457
 COLUMN_VALUE_FROM_INDEX 458
 COLUMN_VALUE_FROM_NAME 461
 DDL_RECORD_PROPERTIES 464
 ENV_VALUE 443, 466
 ERROR_INFO 467
 GMT_TIMESTAMP 468
 MARKER_INFO 469
 OPERATION_TYPE 470
 POSITION 472
 RECORD_BUFFER 473
 RECORD_LENGTH 476
 RECORD_TYPE 477
 STATISTICS 478
 TABLE_COLUMN_COUNT 480
 TABLE_METADATA 443, 481
 TABLE_NAME 482
 TIMESTAMP 483
 TRANSACTION_IND 484
 USER_TOKEN_VALUE 486

GETAPPLOPS

オプション, DDLOPTIONS 162
 パラメータ 197

GETDELETES パラメータ 198**GETENV**

パラメータ 198
 ファンクション 405

GETINSERTS パラメータ 199**GETLAG オプション**

SEND EXTRACT 35
 SEND REPLICAT 56

GETPORTINFO オプション, SEND MANAGER 15**GETPURGEOLDEXTRACTS オプション, SEND MANAGER** 15**GETREPLICATES**

オプション, DDLOPTIONS 162
 パラメータ 199

GETTCPSTATS オプション, SEND EXTRACT 35**GETTRUNCATES パラメータ** 201**GETUPDATEAFTERS パラメータ** 202**GETUPDATEBEFORES パラメータ** 202**GETUPDATES パラメータ** 203**GETVAL ファンクション** 419**GGENVIRONMENT オプション, @GETENV** 409**GGFILEHEADER オプション, @GETENV** 182, 185, 285, 293, 411**GGHEADER オプション, @GETENV** 410**GGSCacheRetryCount オプション, SETENV** 301**GGSCacheRetryDelay オプション, SETENV** 301**GGSCDDL_tables** 85**GGSCHEMA パラメータ** 203**GGSCI コマンド** 14**ggserr.log ファイル, 表示** 94**GGSEVT コマンド** 94**GROUPTRANSOPS パラメータ** 203**-g パラメータ** 391**H****HANDLECOLLISIONS**

再起動時に適用 284

使用オプション

MAP 文 237

SEND REPLICAT 56

グローバル・レベル 204

HANDLETPKUPDATE パラメータ 206**HELP コマンド** 13, 90**HEXTOBIN ファンクション** 420**HIGHVAL ファンクション** 421**HISTORY コマンド** 90**HOST オプション, DBOPTIONS** 146**-h パラメータ** 391**I**

IDENTITY シード, SQL Server ターゲットで更新されな
 い 148

ID オプション, SQLEXEC 252, 345

IF ファンクション 421

IGNOREAPPLOPS

オプション, DDLOPTIONS 162

パラメータ 197

IGNOREDATACAPTURECHANGES パラメータ 368

IGNOREDELETES パラメータ 198

IGNOREGETUPDATEAFTERS パラメータ 202

IGNOREINSERTS パラメータ 199

IGNOREMISSINGTABLES オプション, TRANLOGOPTIONS 368

IGNOREREPLICATES

- オプション, DDLOPTIONS 162
- パラメータ 199

IGNORETRUNCATES パラメータ 201

IGNOREUPDATEBEFORES パラメータ 202

IGNOREUPDATES パラメータ 203

IGNORE オプション, REPERROR 241, 275

IMMEDIATE オプション, WILDCARDRESOLVE 388

INCLUDELIST オプション, SEND EXTRACT 41

INCLUDELONG オプション, ADD TRANDATA 77

INCLUDEUPDATEBEFORES オプション, CUSEREXIT 143

INCLUDE オプション

- DDL 150, 155, 168
- DDLSUBST 155, 167

INCLUDE パラメータ 207

INCONSISTENTROW オプション, REPFETCHEDCOLOPTIONS 277

INFO コマンド

- ALL 90
- CHECKPOINTTABLE 82
- ER 65
- EXTRACT 27
- EXTTRAIL 70
- MANAGER 15
- MARKER 91
- REPLICAT 51
- RMTTRAIL 70
- TRACETABLE 84
- TRANDATA 79

Ingres

- Replicat トランザクションを除外 367
- ログイン・パラメータ 381

INITTRANSRAM オプション

- LOBMEMORY 212
- TRANSMEMORY 376

INLINEPROPERTIES オプション, FORMATXML 194

INQUEUE SIZE オプション, THREADOPTIONS 357

INSERTALLRECORDS

- MAP オプション 238
- パラメータ 208

INSERTAPPEND

- オプション, MAP 238
- パラメータ 207

INSERTDELETES パラメータ 208

INSERTMISSINGUPDATES パラメータ 208

INSERTUPDATES パラメータ 209

INSTRCOMMENTSWORDS オプション, DDL 153, 158, 171

INSTRCOMMENTS オプション, DDL 152, 157, 170

INSTRWORDS オプション, DDL 153, 158, 171

INSTR オプション, DDL 152, 157, 170

IOLATENCY オプション, THREADOPTIONS 357

K

KEYCOLS オプション

- MAP 239
- TABLE 335

KEYNAME オプション

- RMTHOST 288
- RMTHOSTOPTIONS 291

-KEYNAME パラメータ 391

KILL コマンド

- ER 65
- EXTRACT 32
- REPLICAT 55

-k パラメータ 391

L

LAGCRITICAL パラメータ 209

LAGINFO パラメータ 210

LAGREPORT パラメータ 210

LAG オプション, @GETENV 407

LAG コマンド

- ER 65
- EXTRACT 32
- REPLICAT 55

LASTERR オプション, @GETENV 408

LATESTROWVERSION オプション, REPFETCHEDCOLOPTIONS 277

LIMITROWS オプション, DBOPTIONS 146

LIST TABLES コマンド 74

LIST パラメータ 211

LOB

埋込みバッファ・サイズ 146

空 145

ロギング, 制御 77

LOBBUFSIZE オプション, DBOPTIONS 146

LOBMEMORY パラメータ 211

LOBS オプション, ADD TRANDATA 77

LOBWRITESIZE オプション, DBOPTIONS 147

LOGEND オプション, SEND EXTRACT 36

LOGNUM オプション, ADD EXTRACT 21

LOGSOURCE オプション, TRANLOGOPTIONS 369

LOWVAL ファンクション 421

LSN オプション, ADD EXTRACT 22

-l パラメータ 391

M

MACROCHAR パラメータ 215

MACRO パラメータ 213

Manager

起動 16

コマンドの概要 15

実行中プロセスの検証 310

ステータス 15, 16

停止 17

名前, 指定 262

ポート

Manager, 指定 266

動的リスト 177

メンテナンス間隔 136

MANAGESECONDARYTRUNCATIONPOINT オプション, TRANLOGOPTIONS 369

MAPDERIVED オプション, DDLOPTIONS 163

MAPEXCEPTION オプション, MAP 240

MAPEXCLUDE パラメータ 258

MAPPED オプション, DDL 151, 156, 168

MAPSESSIONSCHEMA オプション, DDLOPTIONS 163

MARKERTABLE パラメータ 259

MAXBYTES オプション, DISCARDFILE 174

MAXCOMMITPROPAGATIONDELAY オプション, THREADOPTIONS 357

MAXDISCARDRECS パラメータ 259

MAXFETCHSTATEMENTS パラメータ 260

MAXFILES オプション 285

EXTFILE 183

RMTFILE 286

MAXKEEP オプション

PURGEDDLHISTORY 267

PURGEMARKERHISTORY 268

MAXSQLSTATEMENTS パラメータ 260

MAXTRANSOPS パラメータ 261

MAXVARCHARLEN オプション, SQLEXEC 253, 346

MEGABYTES オプション

ADD EXTTRAIL 66

ADD RMTTRAIL 67

ALTER EXTTRAIL 68

ALTER RMTTRAIL 69

DISCARDFILE 174

EXTFILE 183

RMTFILE 286

MGRPORT オプション

ADD EXTRACT 23

RMTHOST 288

MGRSERVNAME パラメータ 262

Microsoft SQL Server, 「SQL Server」を参照

MINKEEP オプション

PURGEDDLHISTORY 267

PURGEMARKERHISTORY 268

PURGEOLDEXTRACTS 271

MISSINGROW オプション

FETCHOPTIONS 187

REFFETCHEDCOLOPTIONS 278

-m パラメータ 391

N

NAMES オプション, FORMATASCII 191

NCHAR データ, トレイルのフォーマット 385

NOALLOWDUPTARGETMAP パラメータ 115

NOALLOWLOBDATATRUNCATE オプション, DBOPTIONS 144

NOBATCHERRORMODE オプション, BATCHSQL 120

NOBINARYCHARS パラメータ 129, 179, 302

NOCATALOGCONNECT オプション, DBOPTIONS 145

NOCHECKSEQUENCEVALUE パラメータ 138
NOCOMPRESSDELETES パラメータ 141
NOCOMPRESSUPDATES パラメータ 141
NOCROSSRENAME オプション, **DDOPTIONS** 163
NODBCHECKPOINT オプション, **ADD REPLICAT** 49
NODDLCHANGEWARNING パラメータ 370
NODYNSQL パラメータ 178
NOENCRYPTTRAIL パラメータ 179
NOFETCHLOBS オプション, **DBOPTIONS** 146
NOFETCH オプション
 FETCHOPTIONS 187
 REFETCHEDCOLOPTIONS 277
NOFILTERDUPS パラメータ 188
NOFLUSH オプション, **TRANLOGOPTIONS** 370
NOHANDLECOLLISIONS
 オプション, **SEND REPLICAT** 56
 パラメータ 204
NOHDRFIELDS オプション, **FORMATASCII** 191
NOHEADERS パラメータ 262
NOIGNOREDATAACAPTURECHANGES パラメータ 368
NOINSERTAPPEND
 オプション, **MAP** 238
 パラメータ 207
NOINSERTDELETES パラメータ 208
NOINSERTMISSINGUPDATES パラメータ 208
NOINSERTUPDATES パラメータ 209
NOKEY オプション, **ADD TRANDATA** 78
NOLIMITROWS オプション, **DBOPTIONS** 146
NOLIST パラメータ 211
NOMANAGESECONDARYTRUNCATIONPOINT オプション,
 TRANLOGOPTIONS 369
NOMAPDERIVED オプション, **DDOPTIONS** 163
NONAMES オプション
 FORMATASCII 191
 FORMATSQL 193
NONE オプション, **REPLACEBADCHAR** 279
NOOVERRIDEDUPS パラメータ 264
NOPARAMS オプション, **SQLEXEC** 253, 346
NOPASSTHRUMESSAGES パラメータ 266
NOPASSTHRU パラメータ 265

NOPKUPDATES オプション, **FORMATSQL** 193
NOPURGEORPHANEDTRANSACTIONS オプション
 SEND EXTRACT 40
 TRANLOGOPTIONS 371
NOQUOTE オプション, **FORMATASCII** 191
NORENAME オプション, **TABLEEXCLUDE** 354
NOREPORTDETAIL オプション, **STATS REPLICAT** 64
NOREPORTFETCH オプション, **STATOPTIONS** 311
NOREPORT オプション, **DDOPTIONS** 164
NOREQUIRELONGDATAACAPTURECHANGES オプション,
 TRANLOGOPTIONS 372
NORESETREPORTSTATS パラメータ 311
NORESTARTCOLLISIONS パラメータ 284
NOSPACESTONULL パラメータ 305
NOSPTHREAD オプション, **DBOPTIONS** 147
NOSUPPRESSTRIGGERS オプション, **DBOPTIONS** 147
NOT FOR REPLICATION フラグ, 有効化 148
NOTCPSOURCETIMER パラメータ 356
NOTRANSTMTS オプション, **FORMATASCII** 191
NOTRIMSPACES
 オプション, **MAP** 257
 オプション, **TABLE** 351
 パラメータ 378
NOUPDATEDELETES パラメータ 379
NOUSECHECKPOINTS オプション,
 PURGEOLDEXTRACTS 271
NOUSEKEY オプション, **FETCHOPTIONS** 188
NOUSELATESTVERSION オプション, **FETCHOPTIONS** 188
NOUSEROWID オプション, **FETCHOPTIONS** 188
NOUSESNAPOSHOT オプション, **FETCHOPTIONS** 188
NOUSETHEADS オプション, **WARNLONGTRANS** 387
NOVARWIDTHNCHAR パラメータ 385
NULL
 オプション
 REPLACEBADCHAR 279
 REPLACEBADNUM 280
 空白から変換 305
NULLISSPACE オプション, **FORMATASCII** 191
NUMBIN ファンクション 422
NUMFILES パラメータ 263
NUMSTR ファンクション 422

O

OBEY

コマンド 92
ネスト 88
パラメータ 263

OBJNAME オプション, DDL 151, 156, 169

OBJTYPE オプション, DDL 151, 156, 169

ODBC, Replicat 接続オプション 148

ODBC データ・ソース, 指定 303, 354

OLDFORMAT オプション

ADD TRANDATA 78
DELETE TRANDATA 79

OLDFORMAT オプション, TRANDATA コマンド 76

OLE DB, Replicat 接続オプション 148

ONEXIT オプション, SQLEXEC 310

ON オプション

REPORT 281
REPORTROLLOVER 283
ROLLOVER 295

OPENTRANS オプション, ADD EXTRACT 41

OPSPERBATCH オプション, BATCHSQL 121

OPSPERQUEUE オプション, BATCHSQL 121

OPTYPE オプション, DDL 151, 156, 168

Oracle

ASM バッファ・サイズ 364

DDL

エラー処理 154
オプション 159
スキーマ 203
表, 指定 171
フィルタリング 149
マーカー表, 指定 259
履歴のページ 266

Extract 開始位置 21

LOB キャッシング, 無効化 145

RAC

孤立トランザクション, パージ 40
スレッド, 指定 23
スレッド・オプション 356

REDO ログ

代替パス 370
代替プラットフォーム 369

SQL*Loader, パラメータ 129, 192, 195

アーカイブ・ログ

場所, 指定 361
フォーマット 361
唯一のデータ・ソース 364

オープンしているトランザクション, 表示 36

行更新, 制限 146

サブリメンタル・ロギング

新しい表に自動的に有効化 160
起動前に有効化 75

順序, レプリケート 295

準備済問合せ, 数 260

トランザクション, スキップ 37

トレース表

作成および保持 83
指定 359

認証, 指定 162, 364, 380

日付および時刻フォーマットの変換 193

Oracle GoldenGate

環境, 表示 93

サブディレクトリ, サクセイ 88

ORACLE オプション, FORMATSQ 193

Oracle ビフォア・イメージを追加するトリガー 74

Oracle 用 ALTARCHIVELOGDEST オプション, TRANLOGOPTIONS 361

order_no 353
OSVARIABLE オプション, @GETENV 418
OTHER オプション, DDL 151, 156, 168
OUTPUT_MESSAGE_TO_REPORT ファンクション 487
OUTPUTFILEUMASK パラメータ 264
OUTQUEUESIZE オプション, THREADOPTIONS 357
OVERRIDEDUPS パラメータ 264

P

PAGE オプション, ADD EXTRACT 23
PARAMBUFSIZE オプション, SQLEXEC 253, 346
PARAMS オプション
 ADD EXTRACT 23
 ADD REPLICAT 49
 CUSEREXIT 143
 RMTHOST 289
 RMTHOSTOPTIONS 291
 SQLEXEC 254, 347
 VAM 384
PASSIVE オプション, ADD EXTRACT 23
PASSTHRU
 オプション, CUSEREXIT 142
 パラメータ 265
PASSTHRUMESSAGES パラメータ 266
PASSWORD オプション, USERID 382
PATHMAP オプション, TRANLOGOPTIONS 370
PLACEHOLDERS オプション, FORMATASCII 191
PORT
 オプション, RMTHOST 289
 パラメータ 266
PROCESS VM AVAIL FROM OS 統計 131
PTLF レコード, キーの関連付け 418
PURGEDDLHISTORY パラメータ 266
PURGEMARKERHISTORY パラメータ 268
PURGEOLDEXTRACTS パラメータ 269
PURGEOLDTASKS パラメータ 272
PURGEORPHANEDTRANSACTIONS オプション
 SEND EXTRACT 40
 TRANLOGOPTIONS 371

PURGE オプション
 DEFSFILE 173
 DISCARDFILE 174
 RMTHOST 285
-P パラメータ 391
-p パラメータ 391

Q

QUERYRETRYCOUNT オプション, TRANLOGOPTIONS 371
QUERY オプション, SQLEXEC 246, 339

R

RAC, Oracle
 スレッド, 指定 23
 チューニング・オプション 356
RAISEERROR オプション, FILTER 句 237
RAMINCREMENT オプション
 LOBMEMORY 212
 TRANSMEMORY 377
RAM オプション
 LOBMEMORY 212
 TRANSMEMORY 376
RANGE ファンクション 423
RBA
 Extract 開始位置 21
 Replicat 開始位置 49
READBUFFER オプション, TRANLOGOPTIONS 371
READTIMEOUT オプション, TRANLOGOPTIONS 371
RECORD オプション, @GETENV 416
RECOVERYOPTIONS パラメータ 273
RECSOUTPUT オプション, GETENV 408
REDUNDANTROW オプション,
 REFETCHEDCOLOPTIONS 278
REMOVECOMMENTS オプション, DDLOPTIONS 164
REPERROR
 オプション, MAP 241
 パラメータ 274
REFETCHEDCOLOPTIONS パラメータ 277
REPLACEBADCHAR パラメータ 279
REPLACEBADNUM パラメータ 280

Replicat

- Oracle SQL の APPEND ヒント 207, 238
- エラー処理 241, 274, 387
- 起動 58
- 構文, 表示 302
- コマンド 47
- 実行履歴, 削除 50
- ステータス, 表示 64
- 双方向レプリケーション・パラメータのメタデータを更新 165
- 停止
 - オンライン処理 55, 65
 - バッチ実行 180
- 統計, 表示 63
- トランザクション
 - 分離 197
 - 無視 359, 366, 367, 368
- トランザクション, タイムアウト 374
- トランザクションの遅延 172
- トランザクション名 200
- トレース 358
- トレイル内開始位置 58
- ラグ, 表示 51
- レポート, 表示 94
- 「Replicat グループ」も参照

REPLICATEPASSWORD オプション, DDLOPTIONS 164**Replicat グループ**

- 最大数 47
- 削除 51
- 追加 47
- パラメータ・ファイルに指定 280
- 変更 50

REPLICAT パラメータ 280**Replicat 用 MAP パラメータ**

- 使用 217
- 重複, 許可 115

REPORTCOUNT パラメータ 282**REPORTDETAIL オプション, STATOPTIONS 311****REPORTDETAIL オプション, STATS REPLICAT 64****REPORTFETCH オプション**

- STATOPTIONS 311
- STATS EXTRACT 45

REPORTRATE オプション

- STATS EXTRACT 45
- STATS REPLICAT 64

REPORTROLLOVER パラメータ 283**REPORT オプション**

- ADD EXTRACT 23
- ADD REPLICAT 49
- DDLOPTIONS 164
- ROLLOVER 295
- SEND EXTRACT 36
- SEND REPLICAT 57

REPORT パラメータ 281**RESET_USEREXIT_STATS ファンクション 487****RESETMINUTES オプション, AUTORESTART 117****RESETREPORTSTATS パラメータ 311****RESET オプション, REPERROR 276****RESTARTAPPEND オプション, DSOPTIONS 177****RESTARTCOLLISIONS パラメータ 284****RESTARTSKIP オプション, DDLERROR 154****RETRIES オプション, AUTORESTART 117****RETRYDELAY パラメータ 284****RETRYOP オプション, REPERROR 242, 275****RMTFILE パラメータ 284****RMTHOST**

- オプション, ADD EXTRACT 23
- パラメータ 287

RMTHOSTOPTIONS パラメータ 290**RMTNAME オプション, ADD EXTRACT 23****RMTTASK パラメータ 292****RMTTRAIL パラメータ 293****ROLLOVER**

- オプション, SEND EXTRACT 36
- パラメータ 294

RUNTIME オプション, END 180**-R パラメータ 391****S****SAVE オプション**

- CLEANUP EXTRACT 26
- CLEANUP REPLICAT 50

SEND コマンド

ER 65
EXTRACT 33
MANAGER 15
REPLICAT 56

SEQUENCE パラメータ 295**SET EDITOR コマンド** 71, 96**SET_ ファンクション**

COLUMN_VALUE_BY_INDEX 487
COLUMN_VALUE_BY_NAME 489
OPERATION_TYPE 491
RECORD_BUFFER 473, 492
TABLE_NAME 494

SETENV パラメータ 301**SETIFMISSING オプション, REPFETCHEDCOLOPTIONS** 278**SHELL コマンド** 93**SHOW**

オプション, DUMPDDL 86
コマンド 93

SHOWCH オプション

INFO EXTRACT 31
INFO REPLICAT 54

SHOWINFOMESSAGES オプション, DBOPTIONS 147**SHOWSYNTAX パラメータ** 302**SHOWTRANS オプション, SEND EXTRACT** 36, 46**SHOWWARNINGS オプション, DBOPTIONS** 147**SKIPTRANSACTION オプション, START REPLICAT** 59**SKIPTRANS オプション, SEND EXTRACT** 37, 46**SKIPTRIGGERERROR オプション, DDLERROR** 154**SNAPSHOTROW オプション, REPFETCHEDCOLOPTIONS** 279**SORTTRANLOG オプション, DSOPTIONS** 177**SOURCEDB パラメータ** 303, 309**SOURCEDEFS パラメータ** 304**SOURCEISTABLE**

オプション, ADD EXTRACT 19
パラメータ 304

SPACESTONULL パラメータ 305**SPACE オプション, REPLACEBADCHAR** 279**SPECIALRUN**

EXTRACT 用パラメータ 305
REPLICAT 用パラメータ 306
オプション, ADD REPLICAT 48

SPTHREAD オプション, DBOPTIONS 147**SQL**

Replicat, 表示 302
エラー警告率 387
実行頻度 250, 344
出力フォーマット 192
処理中に実行 243, 336
重複行エラー 307
バッチ処理 118
文, 数 260
リテラル文, 使用 178

SQL Server

2 次切捨てポイント, 管理 369
Integration Services (SSIS) 191
Replicat 接続オプション 148
Replicat トランザクションの除外パラメータ 366
大 / 小文字区別 225
環境パラメータ 301
切捨て, サポート 201
更新行数の制限 146
信頼できる接続, 使用 148
双方向の同期化 200
代替場所のログ 363
メタデータ問合せ再試行パラメータ 371
ログイン・パラメータ 381

SQL Server 用 ALTARCHIVELOGDEST オプション, TRANLOGOPTIONS 363**SQL*Loader, ファイル生成** 195**SQL/MX**

Extract 開始位置 21
カタログおよびスキーマ・パラメータ 303, 381
双方向サポート 368
パスワード暗号化 73

SQLDUPERR パラメータ 307**SQLEXEC**

MAP 文 243
TABLE 文 336
グローバル 307

SQLID オプション, DBLOGIN 73**SQLLOADER オプション, FORMATASCII** 192**SQLPREDICATE オプション, TABLE** 349**SQL 文のキャッシング** 118**STARTUPVALIDATIONDELAY パラメータ** 310

START コマンド

ER 65
EXTRACT 43
MANAGER 16
REPLICAT 58

STATOPTIONS パラメータ 311**STATS コマンド**

ER 65
EXTRACT 43
REPLICAT 63

STATUS オプション

SEND EXTRACT 38
SEND REPLICAT 57

STATUS コマンド

ER 65
EXTRACT 46
MANAGER 16
REPLICAT 64

STOP オプション

SEND EXTRACT 39
SEND REPLICAT 57

STOP コマンド

ER 65
EXTRACT 46
MANAGER 17
REPLICAT 65

STRCAT ファンクション 424**STRCMP ファンクション 425****STREQ ファンクション 425****STREXT ファンクション 426****STRFIND ファンクション 427****STRLEN ファンクション 427****STRLTRIM ファンクション 428****STRNCAT ファンクション 428****STRNCMP ファンクション 429****STRNUM ファンクション 430****STRRTRIM ファンクション 431****STRSUB ファンクション 431****STRTRIM ファンクション 432****STRUP ファンクション 432****SUPPRESSTRIGGERS オプション, DBOPTIONS 147****Sybase**

2 次切捨てポイント 369

LOB

空 145

切捨て, 制御 144

伝播, 制御 77

ロギング 77

Replicat トランザクション, 識別 366

TDS パケット・サイズ 148

行, 更新制限 146

サーバー・メッセージ, エラー・ログに出力 147

認証, 指定 381

表レプリケーション, 有効化 74

レプリケーション, 表のマーク付け 75

SYSDBA オプション, DBLOGIN 73**SYSLOG パラメータ 312****T****TABLEEXCLUDE パラメータ 353****TABLE オプション**

STATS EXTRACT 45

STATS REPLICAT 64

TABLE パラメータ

DEFGEN 313

Extract 313

Replicat 352

多数を許可 263

TARGETDB パラメータ 309, 354**TARGETDEFS パラメータ 355****TASKS オプション**

INFO EXTRACT 32

INFO REPLICAT 54

STATUS EXTRACT 46

STATUS REPLICAT 64

TCP/IP

統計, 表示 35

ポート, Manager 266, 288

TCPBUFSIZE オプション

RMTHOST 289

RMTHOSTOPTIONS 291

TCPFLUSHBYTES オプション

RMTHOST 289

RMTHOSTOPTIONS 291

TCPSOURCETIMER パラメータ 356

TDSPACKETSIZE オプション, DBOPTIONS 148

TDS パケット・サイズ, 増加 148

Teradata

- DDL
 - エラー処理 154
 - 構成オプション 159
 - フィルタリング 149
- 構成オプション 176
- コマンド, データベースに送信 41
- 処理モード 176
- データ・ソース 20, 384
- 日時パラメータ 379, 383
- 認証, 指定 381

THREDOPTIONS パラメータ 356

THREADS オプション, ADD EXTRACT 23

THREAD オプション, ADD EXTRACT 25

TIMEOUT オプション

- RMTHOST 289
- RMTHOSTOPTIONS 291

TIME オプション, FORMATASCII 191

TLFKEY オプション, @GETENV 418

TLF レコード, キーの関連付け 418

TOKENS オプション, TABLE 350

TOKEN ファンクション 433

TOTALONLY オプション

- SEND REPLICAT 64
- STATS EXTRACT 45

TRACE

- BATCHSQL オプション 121
- SEND EXTRACT オプション 40
- SEND REPLICAT オプション 57
- SQLEXEC オプション 256, 349
- パラメータ 358

TRACEINIT オプション

- SEND EXTRACT 40
- SEND REPLICAT 58

TRACE オプション

- DDL 57

Trandata コマンド 74

TRANLOGOPTIONS

- オプション, SEND EXTRACT 40
- パラメータ 360

TRANLOG オプション

- ADD EXTRACT 19
- SPECIALRUN 306

TRANSABORT オプション, REPERORR 242, 276

TRANSACTIONTIMEOUT パラメータ 374

TRANSACTION オプション, @GETENV 417

TRANSALLSOURCES オプション

- LOBMEMORY 212
- TRANSMEMORY 376

TRANSCLEANUPFREQUENCY オプション

- SEND EXTRACT 40
- TRANLOGOPTIONS 372

TRANSMEMORY パラメータ 375

TRANSRAM オプション

- LOBMEMORY 212
- TRANSMEMORY 376

TRANS オプション, FORMATXML 194

TRIMSPACES

- オプション 257, 351
- パラメータ 378

TRUSTEDCONNECTION オプション, DBOPTIONS 148

TS オプション, FORMATASCII 191

U

-UL パラメータ 392

umask, 出力ファイルの設定 264

UNDO セグメント

- 問合せでの量の削減 349
- フェッチ元 188

Unicode 列および文字列 223

UNLOCKEDTRAILFILES パラメータ 378

UNMAPPED オプション, DDL 151, 156, 168

UNPRINTABLE オプション

- REPLACEBADCHAR 279
- REPLACEBADNUM 280

UNPRIVILEGED オプション, TRANLOGOPTIONS 372

UPDATEDELETES パラメータ 379

UPDITEMETADATA オプション, DDLOPTIONS 165

UPREPORT パラメータ 379

USECHECKPOINTS オプション, PURGEOLDEXTRACTS 271

USEDATEPREFIX パラメータ 379

USEDEFAULTS オプション

MAP 225, 227, 322

TABLE 323

USEKEY オプション, FETCHOPTIONS 188

USELASTREADTIME オプション, WARNLONGTRANS 387

USELATESTVERSION オプション, FETCHOPTIONS 188

USEODBC オプション, DBOPTIONS 148

USEOWNERFORSESSION オプション, DDLOPTIONS 165

USEREPLICATIONUSER オプション, DBOPTIONS 148

USERID

オプション, DBLOGIN 73

パラメータ 380

USEROWID オプション, FETCHOPTIONS 188

USESNAAPSHOT オプション, FETCHOPTIONS 188

USESTOPSTATUS 引数, PURGEOLDTASKS 273

USETIMEPREFIX パラメータ 383

USETIMESTAMPMPREFIX パラメータ 383

V

VALONEOF ファンクション 434

VAM

オプション, ADD EXTRACT 20

互換性, 指定 373

トレイル

作成 177

データ・ソース 20

パラメータ 384

VAMCOMPATIBILITY オプション, TRANLOGOPTIONS 373

VAMMESSAGE オプション, SEND EXTRACT 41

VAMTRAILSOURCE オプション, ADD EXTRACT 20

VARWIDTHNCHAR パラメータ 385

VERSIONS コマンド 94

VIEW GGSEVT コマンド 94

VIEW PARAMS コマンド 72

VIEW REPORT コマンド 94

W

WAITMINUTES オプション, AUTORESTART 117

WARNLONGTRANS パラメータ 386

WARNRATE パラメータ 387

WHERE オプション

MAP 257

TABLE 351

WHERE 句

MAP 文 257

TABLE 文 351

初期ロードの選択 349

X

XML

埋込み, バッファ 149

トレイルに出力 193

XMLBUFSIZE オプション, DBOPTIONS 149

-x パラメータ 392

ア

アーカイブ・ログ

異なるデータベースから読取り 216

処理オプション 360

アクション, 処理中にトリガー 228, 325, 352

アスタリスクワイルドカード文字 222, 300, 319

圧縮, 使用 288, 290

アフター・イメージ, 含める 202

アフター・インジケータ, 返す 410

暗号化

IDENTIFIED BY のパスワード 162

TCP/IP 288, 290

パスワード・データベース 73

トレイル 179

イ

イグジット, 「ユーザー・イグジット」を参照

一時主キー更新 206

イベント, 処理中にトリガー 228, 325, 352

イベント, 表示 94

イベント・マーカ・システム 228, 325, 352

イベント・レコード 228, 325, 352

引用符, ASCII 出力から除外 191

ウ

上書きモード・リカバリ・オプション 273

エ

エディタ, 変更 71

エディタ、変更 96

エラー処理

Collector 390

DDL 154

FILTER 句 237

MAP 文 241

エラー情報, プロセスに返す 408

大きな LOB 144

警告率 387

衝突 56, 204, 284

ストアド・プロシージャおよび問合せ 250, 343

重複行 307

重複レコード 264

例外 MAP 235, 240

レスポンス, 指定 274

エラー・メッセージ

大きすぎる Sybase LOB 144

失敗時に生成されない 310

表示 94

ファンクションに返す 408, 448

エラー・ログ, 表示 94**オ****大 / 小文字区別**

DDL 文字列置換 166

コマンド 88

順序名 297

トークン名 351

パスワード 73, 162, 364

マクロ・パラメータ 214

列マッピング 226

列マップ 225, 322

大文字, 変換 432**オブジェクト ID, 名前にマッピング 216****オブジェクト・レコード, 構築ルール 178****オペレーティング・システム**

異種のトランザクション・ログ 369

タイプ, 表示 94

変数, プロセスに返す 418

ログイン 382

オンライン・ヘルプ, 入手 13**オンライン処理**

開始

Extract 43

Replicat 58

グループ, ツイカ

Extract 17

Replicat 47

指定

Extract パラメータ・ファイル 184

Replicat パラメータ・ファイル 280

停止

Extract 46

Replicat 65

カ**カーソル, 指定**

動的 SQL 260

フェッチ問合せ 260

開始位置

オンライン処理 20, 48

バッチ実行 122

外部キー制約, SQL Server ターゲットでの無効化 148**鍵**

暗号化 288, 291, 391

数

REDO ログ・スレッド, 指定 23

グループ, 最大 47

カスケード操作, ターゲットでの無効化 147, 148**仮想メモリ, 管理 129****間隔, チェックポイント 123****環境**

Oracle GoldenGate, 表示 93

情報, 取得 405

変数

設定 301

表示 198

キ

キー

- TLF/PTLF 418
- 一時更新 206
- サプリメンタル・ロギングの抑止 78
- 代替 239
- 名前, サポートされる文字 220, 298, 317

キャッシュ, メモリー 129

キュー, Extract

- 出力 357
- 入力 357

行

- 初期ロードの選択でパーティション化 349
 - すべて抽出 19, 304
 - 選択数, 制限 146
 - ソースに基づいて挿入 208
 - 重複
 - SQL コード 307
 - 上書き 264
 - 範囲に分割 423
 - フィルタリング
 - FILTER 文 236, 334
 - 条件文 257, 351
 - 列のフェッチ 332
- 切捨て, 処理制御 201

ク

空白

- オブジェクトおよび列の名前 221, 299, 318
- 切捨て
 - 先行 428
 - 先行および末尾 432
 - 末尾 431

グループ, 「Extract グループ」または「Replicat グループ」を参照

ケ

警告, 抑止

- ソース・オブジェクトへの DDL 実行時 370
- ログ・ファイルの不在時 370

計算

- 算術 400
- 日付の差異 404

結果コード, ユーザー・イグジット 441

コ

更新

- 圧縮 141
- 圧縮, 列のフェッチ 332
- アフター・イメージ, 処理 202
- 一時主キー 206
- 挿入に変換 208, 209
- ビフォア・イメージ, 処理 202
- フィルタリング 203
- 複数, 防止 146

構文

- Replicat, 表示 302
- パラメータ
 - 検証 136
 - 表示 72

コマンド

- Extract 17
- Manager 15
- Replicat 47
- SQL/MX, 送信 41
- Teradata, 送信 41
- 一般 86
- 繰返し 86, 88
- シェル, 実行 93
- チェックポイント表 80
- データベース 72, 307
- トランザクション・データ (Trandata) 74
- トレース表 83
- トレイル 66
- パラメータ編集 71
- ファイルから実行 92
- ヘルプの使用 90
- ユニットとしての Extract, Replicat 65
- 履歴, 表示 90

コミット・タイムスタンプ, カエス 410

コメント

- DDL 164, 166
- パラメータ・ファイル 140

コールバック・ルーチン, ユーザー・イグジット 440

サ

差異, 計算

算術 400

日付 404

最小値, 抑制 421

最大値, 抑制 421

削除

Extract グループ 26

Replicat グループ 51

圧縮 141

カスケード, ターゲットで延期 145

チェックポイント表 81

トレース表 84

トレイル 69

フィルタリング 198

複数, 防止 146

変換

更新 379

挿入 208

補足トランザクション・データ 79

作成

Extract グループ 17

Oracle トレース表 83

Replicat グループ 47

チェックポイント表 80

トレイル 66, 67

破棄ファイル 173

パラメータ・ファイル 96

サブリメンタル・ロギング

ステータス, 検証 79

属性の変更 76

フェッチの代替手段 332

無効化 79

有効化

新しい表に自動的 160

起動前 74

算術演算子

COMPUTE ファンクション 400

FILTER 句 236, 334

参照整合性制約, ターゲットで延期 145

シ

時間, 変換 401

システム起動, 処理までの遅延 122

システム・ログ, メッセージのフィルタリング 312

主キー, 「キー」を参照

順序

ターゲットの検証 138

レプリケート 295

条件文

WHERE 句 257, 351

ファンクション 421

フィルタ句 236, 334

証跡

形式およびプロパティ, 返す 182, 185, 285, 293

衝突

解決 56, 204

起動後に処理 284

除外

ASCII 出力からデータ 191

DDL レプリケーションからのオブジェクト 150, 155, 168

MAP 文からオブジェクト 258

Replicat ユーザー 367

TABLE 文からオブジェクト 353

取得から Replicat トランザクション 199, 366

取得からトランザクション 368

取得からレコード 334

マッピングから列 324

レポートからマクロ 211

初期ロード

SQL*Loader パラメータ 129

WHERE 句でレコードを選択 349

ファイル, 実行および制御 195

衝突, 解決 56, 204

ダイレクト・ロード方法 19, 292

重複レコード, 上書き 264

ファイルから 304

処理されたレコード数 282

ス

数字

- 置換 280
- 変換
 - バイナリ文字列 422
 - 文字列 422
- 文字に変換 430

スキーマ

- セッションのマッピング 163
- 未修飾のオブジェクトに割り当て 165

ストアド・プロシージャ

- 値, 抽出 419
- 実行
 - MAP 文 243
 - TABLE 文から 336
 - 単独文 307

スナップショットが古すぎるエラー 349

セ

静的 Collector, 定義 389

制約, ターゲットで延期 145

セキュリティ

- データ暗号化 288, 290, 390
- パスワード暗号化 73
- ファイル暗号化 179

セッション・スキーマ, マッピング 163

接続

- 信頼, SQL Server 148
- 複数, 防止 145

ソ

ソース表, 「表」を参照

ソース列, 「列」を参照

操作, データ

圧縮

- 更新 141
- 削除 141

再試行 284

タイプ, 返す 410

フィルタ基準 237, 335

フィルタリング

- 切捨て 201
- 更新 203
- 削除 198
- 挿入 199

変換

- 更新から挿入 208, 209
- 削除から更新 379
- 削除から挿入 208
- 履歴, 保持 208

相対バイト・アドレス, 「RBA」を参照

挿入

- Oracle, APPEND ヒント 207, 238
- 削除から作成 208
- 重複 264, 307
- フィルタリング 199
- 変更操作 208, 238

双方向レプリケーション・パラメータ 197, 366, 367

タ

ターゲット・システム, 指定 287

ターゲット表, 「表」を参照

ターゲット列, 「列」を参照

代替

- キー列 239, 335
- 無効な数字 280
- 無効な文字 279

タイムスタンプ

XML 出力 194

開始

- オンライン処理 20
- バッチ処理 122
- コミット, カエス 410
- 別のシステムに合わせて調整 356

ダイレクト・ロード, 指定 129, 292

タスク

削除 272
作成 292
表示 46, 64

チ**チェックポイント**

Bounded Recovery 123
初期, 作成 17, 47
ページ基準 271
表示 52
 Extract 28, 31
 Replicat 54
表に保持 80
頻度, 制御 123, 137

チェックポイント表

Extract への指定 200
GLOBALS ファイルに指定 137
クリーンアップ 81
削除 81
情報, 表示 82
追加 80
優先 49

遅延

Oracle GoldenGate 起動 122
Replicat トランザクション 172

遅延, 「ラグ」を参照**置換**

文字列 431

抽出トレイル

暗号化 179
削除 69
追加 66
パラメータ・ファイルに指定 185
変更 68
「トレイル」も参照

抽出ファイル

umask, 設定 264
暗号化 179
データ・ソース
 Replicat 48
 データ・ポンプ 20
パラメータ・ファイルに指定 182

長時間におよぶトランザクション

警告 386
表示 36
リカバリ 123

ツ**追加**

Extract グループ 17
Oracle トレース表 83
Replicat グループ 47
チェックポイント表 80
トレイル 66, 67
パラメータ 98
補足トランザクション・データ 74
追加モード・リカバリ・オプション 273

テ**定義, 「データ定義」を参照****定義テンプレート, 定義 228****ディレクトリ**

Oracle GoldenGate, サブディレクトリ作成 88
アーカイブ・ログの代替 362
パラメータ・ファイル, 指定 23, 49
メモリー・ページング 132
レポート・ファイル, 指定 23, 49

テキスト

EBCDIC に変換 115
FILTER 句 237, 334
エディタ, 変更 71
パラメータ・ファイルのコメント 140

テキスト・エディタ, 変更 96**適用延期機能 172****テスト**

値選択 398, 405
条件 421
列の存在 399

データ

- 16 進数, バイナリに変換 420
- 圧縮 288, 290
- 暗号化 288, 290
- 外部フォーマットで出力 190, 192, 193
- バイナリ
 - 16 進数に変換 398
 - 保持 397
- 範囲に分割 423
- 変換
 - 変換ファンクションを使用 393
 - ユーザー・イグジット 142
- マッピング 217, 313
- 文字, 「文字データ」を参照
- ループ, 防止
 - Oracle 74
 - 他のデータベース 360, 367

データ・ソース, ODBC 303, 354**データ・ソース, 指定**

- Extract プロセス 19
- Replicat プロセス 48

データ定義

- ソースに基づく 115
- ファイル名パラメータ
 - Collector 389
 - DEFGEN 173
 - Replicat 304

データベース

- イベント, トリガー 228, 325, 352
- オプション, 設定 143
- 環境, 返す 416
- コマンド 72, 307
- バージョン, 表示 94
- パスワード, 暗号化 73
- ログイン
 - ASM インスタンス 364
 - GGSCI 72
 - Oracle GoldenGate プロセス 380

データ・ポンプ

- 削除 26
- 作成 17
- 実行履歴, 削除 26
- ソース, 指定 185
- パススルー 265
- 変更 25

テンプレート, SQL*Loader, BCP 195**ト****トークン, ユーザー**

- 指定 350
- 取得 433

問合せ

- DDL 履歴表 85
- 値を抽出 419
- 実行
 - MAP 文 243
 - TABLE 文から 336
 - 単独文 307
- 準備数 260
- ブレースホルダ 247, 340

統計

- Extract 43, 94
- Replicat 63, 94
- 一時的
 - Extract 36
 - Replicat 57
- すべてのプロセス 65
- ネットワーク, 表示 35
- 表示, 制御 311
- メモリー・キャッシュ 34
- レコード数 282
- レポート, リセット 311

動的 Collector, 定義 389**動的ポート**

- 再割当て待機 178
- 指定 177
- リストの表示 15

特別実行

- 開始時刻 122
- 指定 48, 305, 306
- 終了時刻 180

トランザクション

Replicat でスキップ 58

オープン

コミット 34

スキップ 37

表示 36

孤立, パージ 40, 372

情報 417

除外 197, 366, 367

ターゲット

1 番目をスキップ 59

タイムアウト 374

遅延 172

分割 261

長時間におよぶ, 警告 386

長時間におよぶ, リカバリ 123

バッファ, 管理 129, 364

無視 368

トランザクション・インジケータ, 返す 411**トランザクション・ログ**

異種プラットフォーム 369

位置, 返す 410

異なるデータベースから読取り 216

抽出オプション 360

データ・ソース 19, 306

補足データ, 取得 74

読取りバッファ・サイズ 364

トリガー, 抑止

Oracle ターゲット上 147

SQL Server ターゲット上 148

トレイル

umask, 設定 264

開始位置、指定 21, 48

旧フォーマット 181

削除 69

情報 70

追加 66, 67, 177

データ・ソース 20, 48

バージョン, 指定 182, 185, 285, 293

パラメータ・ファイルに指定 185, 293

ファイル

暗号化 179

エージング 294

サイズ, 指定 66, 67

パージ 269

フォーマット, 指定 182, 185, 285, 293

フォーマットおよびプロパティ, 返す 411

変更 68

レコードの場所 416

ロールオーバー 25

ロック, 防止 378, 392

トレース・オプション

BATCHSQL 121

DDL 40, 358

SQLEXEC パラメータ 256, 349

処理のボトルネック 358

プロセス・ボトルネック 40

マクロ展開 139

トレース表

検証 84

削除 84

作成 83

指定 359

ナ**～内のコメント**

パラメータ・ファイル 97

名前

サポートされない文字 221, 299, 318

サポートされる文字 220, 298, 317

導出 163

トリガーおよびログ・グループ 76

ワイルドカードの使用 221, 299, 318

ニ**認証**

データ・ソース名 303, 354

データベース・ユーザー 380

ネ**ネイティブ・エンコーディング 223****ネットワーク、統計の表示 35**

ハ

バイナリ・データ

- 16 進数に変換 398
- 保持 397

バイナリ・データのゼロ 227

バイナリ文字, 変換

- Enscribe 227
- 数字へ 422

配列処理, 使用 118

破棄ファイル

- umask, 設定 264
- エージング 174
- サイズ, 制約 259
- 指定 173, 392

バージョン, 表示 94, 182, 185, 285, 293

パスワード, データベース

- 暗号化 73
- 指定 382

パッシブ Extract

- TCP/IP オプション 290
- 作成 23

ハッシュ, 定義 423

バッチ実行

- エンド・ポイント 180
- 開始位置 122
- 指定 48, 305

バッチ処理, Replicat 操作 118

バッファ

- DB2, フラッシュの防止 370
- Extract, フラッシュ 189, 391
- 埋込み LOB 146
- 埋込み XML 149
- サイズ, SQLEXEC パラメータ 253, 346
- メモリー・プール, 管理 129
- ログ読取り, 管理 364

パラメータ

SQLEXEC

- 指定 254, 347
- 問合せのブレースホルダ 247, 340
- プロシージャまたは問合せから抽出 419
- 渡し 247, 340

マクロ 214

ユーザー・イグジット 236, 332

パラメータ, Oracle GoldenGate プロセス

- Collector 389
- DEFGN 114
- Extract 102
- GLOBALS 99
- Manager 100
- Replicat 107

表示 72, 94

頻繁に使用 263

ユーザー・イグジット 435

パラメータ・ファイル

- 記憶域, 代替 23, 49
- 検証 136
- コマンド 71
- コメント 140
- テキスト・エディタ, 変更 71
- 表示 72
- 編集 71

範囲, 割当て 423

ヒ

比較演算子, FILTER 句 236, 334

日付

- 現在, 返す 404
- 差異, 計算 404
- 操作 401

ビフォア・イメージ

- WHERE 句 257
- アフター・イメージと比較 202
- トレイル 202

ビフォア・インジケータ, 返す 410

表

DDL 履歴
 指定 171
 ページ 266
 表示 85
Oracle トレース
 コマンド 83
 指定 359
一覧表示 74
指定
 抽出 313
 定義ファイル 313
ソースからターゲットへのマッピング 217
チェックポイント
 Extract への指定 200
 GLOBALS ファイルに指定 137
 コマンド 80
データ・ソース 304
定義
 出力ファイル 304, 355
 デフォルト 115
動的解決 178
名前, 返す 410
名前をオブジェクト ID にマッピング 216
マーカー
 指定 259
 ページ 268
例外 235, 240
ワイルドカード指定から除外 258, 353

フ

ファンクション, 「変換ファンクション」を参照

フィルタリング

DDL 149
EVENTACTION 用ルール 352
Replicat 操作 368
取得する行 334, 351
初期ロードで選択する行 349
ターゲットに送信する行 236, 257
トランザクション 366, 367, 368

フェッチ

統計, 表示 45
動作, 制御 187, 277

ブレースホルダ

問合せ 247, 340
行方不明の列 191

プロセス, Oracle GoldenGate

環境, 返す 409
起動
 異常終了後 116
 自動 117
起動の遅延 122
子 15
情報, すべて表示 90
すべてを制御および表示 65
レポートの表示 94
「Extract」, 「Manager」, または 「Replicat」も参照

へ

ページング, 管理 129

ヘッダー, レコード

値を返す 410
除外 262

別名 Extract グループ

作成 23
チェックポイント・ファイル 389

変換, 実装

SQL 文 243, 336
変換ファンクション 393
ユーザー・イグジット 142
列マッピング文 224, 321

変換ファンクション

使用 393
メモリー割当て 194

変更

DDL 表名 171
Extract グループ 25
Manager ポート番号 266
Manager 名 262
Replicat グループ 50
テキスト・エディタ 96
トレイル 68
トレイル・フォーマット 185, 293
パラメータ 98
ファイル・フォーマット 182, 284
マーカー表名 259
「修正」も参照 25
「変更」も参照

変更された列, フェッチ 333

変更順序番号

Replicat 開始位置 58
情報 495

編集

以前の GGSCI コマンド 88
パラメータ・ファイル 71, 98

変数, 「環境変数」を参照

ホ**ポート番号**

Collector 391
Manager 266
再割当て待機 178
動的な割当て 177
マルチデーモン MySQL 145
リモート 288

ホスト

MySQL マルチデーモン 146
名前, 取得 409
別名 Extract のソース 391
リモート, 指定 287

マ**マーカー**

イグジット・コールのトリガー 437
表示 91

マーカー表, ページ 268

マクロ

作成 213
代替文字 215
展開, トレース 139
ライブラリ, 含める
パラメータ・ファイル 207
レポート・ファイル 211

マッピング

環境情報 405
導出オブジェクト 163
表, ソースからターゲット 217
ユーザー・トークン 433
列
グローバル 139
個別 224, 321

マルチデーモン MySQL オプション 145, 146

ム

無効なデータ, 置換 279, 280, 391

メ**メッセージ**

システム・ログでフィルタリング 312
送信先
Extract 33
Manager 15
Replicat 56

メモリー, 管理

Extract バッファ 189, 391
SQLEXEC 用パラメータ 253, 346
グローバル・プール 129
表マッピング 114, 263
変換ファンクション 194

メンテナンス

DDL マーカー表 268
DDL 履歴表 266
Manager, 間隔 136
実行履歴, 削除
 Extract 26
 Replicat 50
トレイル 69, 269
ラグ統計
 確認間隔 210
 しきい値 209
 レポート間隔 210

モ**文字**

Unicode 223
オブジェクト名でのサポート 220, 298, 317
数
 グループ名 19, 48
 パラメータ文字列 236, 332
ネイティブ・エンコーディング 223
マクロおよびパラメータ 214, 215
ワイルドカードとの一致 222, 300, 319

文字, マクロ 215

文字データ

NCHAR, トレイルのフォーマット 385
無効, 置換 279

文字列

位置, 決定 427
一部, 抽出 426
空白, 切捨て 428, 431, 432
置換
 DDL 166
 空のLOB 145
 文字から文字 431
長さ, 返す 427
比較
 値 425, 434
 文字数 425, 429
変換
 大文字 432
 数字から文字 430
 バイナリから数字 422
 文字から数字 422
連結 424, 428

ユ**ユーザー**

Oracle ASM, 指定 364
指定 380
除外 366, 367, 368
トランザクション, 無視 368
パスワード, 暗号化 73

ユーザー・イグジット

使用 435
パラメータ渡し
 MAP 文 236
 TABLE 文 332

ユーザー・トークン, 「トークン」を参照

ラ**ライブラリ, マクロ**

パラメータ・ファイル 207
レポート・ファイル 211

ラグ

- Replicat に定義した間隔 172
- 確認間隔 210
- しきい値 209
- 情報, プログラムによって取得 407
- タイムスタンプの調整 356
- 表示
 - Extract 27, 32
 - Replicat 51, 55, 56
 - すべてのプロセス 65, 90
- レポート間隔 210

リ

- リカバリ, **Extract** 123
- リカバリ・モード, **設定** 273
- リモート・タスク, **作成** 292
- リモート・トレイル
 - 削除 69
 - 追加 67
 - パラメータ・ファイルに指定 293
 - 変更 68
 - 「トレイル」も参照
- リモート・ファイル, **指定** 284
- リモート・ホスト, **指定** 287

履歴

- DDL
 - 操作, パージ 266
 - 表示 85
 - マーカー, パージ 268
- GGSCI コマンド 90
- トランザクション 208
- プロセス, 削除 26, 50

履歴表

- DDL
 - パージ 266
 - 表示 85
- 行, パージ 266, 268

ル

- ループ, **防止** 366, 367
 - Oracle 74
 - 他のデータベース 360, 367

レ**例外**

- MAP 文, 指定 235, 240
- ルール, エラー処理 241, 275

レコード

- 「行」も参照
- 処理数 282
- デリミタ 191
- 長さ, 返す 410
- 不正な順番 188

レコード・ヘッダー

- 値, 返す 410
- 抑止 262

列

- Unicode 223
- キー, 代替 239
- サブリメンタル, ログイン 77
- 選択 323
- データベースからフェッチ 332, 333
- テストおよび変換
 - ファンクションを使用 393
 - ユーザー・イグジット 435
- マッピング 139
- 未使用, 許可 144
- 列マップの名称 227, 323

列変換ファンクション

- 概要 393
- メモリー, 割当て 194

列マッピング

- グローバル・ルール 139
- 作成 224, 321
- デフォルト 227, 323

レプリケーション, 表のマーク付け 75**レポート**

- SQLEXEC パラメータ 256, 349
- 一時的な統計
 - Extract 36
 - Replicat 57
- 起動時からの処理レコード 282
- 最後のレポートからのレコード数 295
- 正常なプロセス終了 175
- プロセス情報 94
- ラグ 32, 55

レポート・ファイル

エージング 283
表示 94
別の保管場所 23, 49
「レポート」も参照

□

ローカル・トレイル, 「抽出トレイル」を参照

ロギング, Oracle サプリメンタル 75

ログ, イベント 94

ログイン, オペレーティング・システム 382

ログイン, データベース

ASM 364
GGSCI 72
Oracle GoldenGate プロセス 380
暗号化 73

ログ・ファイル, 数 391

ロード・ユーティリティ用実行ファイル 195

ロード・ユーティリティ用制御ファイル 195

論理名, SQLEXEC 252, 345

ワ**ワイルドカード**

DATA CAPTURE CHANGES が指定されていない表 368

使用 221, 299, 318

含めない 258, 353

*** ワイルドカード文字 222, 300, 319**

空白

NULL に変換 305

切捨て

末尾 257, 351, 378