

# Oracle8i

レプリケーション・マネージメント API リファレンス

リリース 8.1

2000 年 2 月

部品番号 : J00935-01

---

Oracle8i レプリケーション・マネージメント API リファレンス, リリース 8.1

部品番号 : J00935-01

原本名 : Oracle8i Replication Management API Reference, Release 2 (8.1.6)

原本部品番号 : A76958-01

原本著者 : William Creekbaum and Randy Urbano

原本協力者 : Valarie Moore, Nimar Arora, Alan Downing, Al Demers, Ira Greenberg, Denis Goddard, Maria Pratt, Jim Stamos, Curt Elsbernd, Jairaj Galagali, Viswanathan Krishnamurthy, Jing Liu, Pat McElroy, Arvind Rajaram, Wayne Smith, Eric Vandeveld, Lik Wong

Copyright © 1996, 1999, Oracle Corporation. All rights reserved.

Printed in Japan.

#### 制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されています。

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

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

#### 危険な用途への使用について

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

#### Restricted Rights Notice

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

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

---

---

# 目次

はじめに .....	xiii
<b>1 レプリケーションの概要</b>	
レプリケート環境構築の概要 .....	1-2
作業を開始する前に .....	1-4
グローバル名 .....	1-4
ジョブ・プロセス .....	1-4
<b>2 レプリケーション・サイトの構築</b>	
レプリケーション・サイト設定の概要 .....	2-2
マスター・サイトの設定 .....	2-4
スナップショット・サイトの設定 .....	2-14
<b>3 マスター・グループの作成</b>	
マスター・グループ作成の概要 .....	3-2
作業を開始する前に .....	3-3
マスター・グループの作成 .....	3-5
<b>4 配置テンプレートの作成</b>	
Oracle 配置テンプレートの概念 .....	4-2
配置テンプレートの作成 .....	4-2
インスタンスエーションのためのパッケージ化 .....	4-10
テンプレートのパッケージ化 .....	4-12
インスタンスエーション・スクリプトのファイルへの保存 .....	4-14

ファイルの配布 .....	4-15
配置テンプレートのインスタンス化 .....	4-16
インスタンス化後のリフレッシュ .....	4-17

## 5 スナップショット・グループの作成

スナップショット・グループ作成の概要 .....	5-2
スナップショット・グループの作成 .....	5-4

## 6 競合解消

競合解消の準備 .....	6-2
プラン .....	6-2
更新競合の競合解消方法の作成 .....	6-2
上書きによる方法と廃棄による方法 .....	6-3
最小値による方法と最大値による方法 .....	6-4
タイムスタンプによる方法 .....	6-6
加算による方法と平均による方法 .....	6-9
優先グループ .....	6-11
サイトの優先順位 .....	6-13
一意性競合の競合解消方法の作成 .....	6-16
削除競合の競合回避方法の作成 .....	6-21
競合解消の監査 .....	6-24
競合解消統計の収集 .....	6-24
競合解消統計の表示 .....	6-24
競合解消統計の取消し .....	6-24
統計情報の削除 .....	6-25

## 7 API を使用したレプリケート環境の管理

マスター・サイトの管理 .....	7-2
マスター定義サイトの変更 .....	7-2
マスター・サイトの追加 .....	7-3
マスター・サイトの削除 .....	7-4
スナップショット・サイトの管理 .....	7-5
グループの所有者の使用 .....	7-5
スナップショット・グループのマスター・サイトの変更 .....	7-9
スナップショット・サイトの削除 .....	7-10

スナップショット・ログの管理 .....	7-17
<b>遅延トランザクションの管理</b> .....	7-24
遅延トランザクション・キューのプッシュ .....	7-24
遅延トランザクション・キューのパージ .....	7-25
<b>エラー・キューの管理</b> .....	7-26
受信者としてのエラー・トランザクションの再実行 .....	7-26
代替ユーザーとしてのエラー・トランザクションの再実行 .....	7-27
<b>レプリケート・オブジェクトの変更</b> .....	7-27
<b>エクスポートおよびインポートを使用したオフライン・インスタンスの実行</b> .....	7-29
マスター・サイト .....	7-29
スナップショット・サイト .....	7-34
<b>レプリケート表間の違いの判別</b> .....	7-41
DIFFERENCES .....	7-41
RECTIFY .....	7-41
<b>データ・ディクショナリ・ビュー内のコメント・フィールドの更新</b> .....	7-45

## 8 レプリケーション・マネージメント API リファレンス

<b>パッケージ</b> .....	8-2
<b>Oracle レプリケーション・マネージメント API の使用例</b> .....	8-2
考慮事項 .....	8-3
Replication Manager および Oracle レプリケーション・マネージメント API .....	8-3
<b>DBMS_DEFER パッケージ</b> .....	8-4
サブプログラムの概要 .....	8-4
CALL プロシージャ .....	8-4
COMMIT_WORK プロシージャ .....	8-5
datatype_ARG プロシージャ .....	8-6
TRANSACTION プロシージャ .....	8-7
<b>DBMS_DEFER_QUERY パッケージ</b> .....	8-9
サブプログラムの概要 .....	8-9
GET_ARG_FORM ファンクション .....	8-9
GET_ARG_TYPE ファンクション .....	8-10
GET_CALL_ARGS プロシージャ .....	8-11
GET_datatype_ARG ファンクション .....	8-12
<b>DBMS_DEFER_SYS パッケージ</b> .....	8-15
サブプログラムの概要 .....	8-15
ADD_DEFAULT_DEST プロシージャ .....	8-16

DELETE_DEFAULT_DEST プロシージャ .....	8-17
DELETE_DEF_DESTINATION プロシージャ .....	8-17
DELETE_ERROR プロシージャ .....	8-18
DELETE_TRAN プロシージャ .....	8-18
DISABLED ファンクション .....	8-19
EXCLUDE_PUSH プロシージャ .....	8-20
EXECUTE_ERROR プロシージャ .....	8-21
EXECUTE_ERROR_AS_USER プロシージャ .....	8-21
PURGE ファンクション .....	8-22
PUSH ファンクション .....	8-24
REGISTER_PROPAGATOR プロシージャ .....	8-27
SCHEDULE_PURGE プロシージャ .....	8-27
SCHEDULE_PUSH プロシージャ .....	8-29
SET_DISABLED プロシージャ .....	8-30
UNREGISTER_PROPAGATOR プロシージャ .....	8-31
UNSCHEDULE_PURGE プロシージャ .....	8-32
UNSCHEDULE_PUSH プロシージャ .....	8-32
<b>DBMS_OFFLINE_OG パッケージ .....</b>	<b>8-34</b>
サブプログラムの概要 .....	8-34
BEGIN_INSTANTIATION プロシージャ .....	8-34
BEGIN_LOAD プロシージャ .....	8-35
END_INSTANTIATION プロシージャ .....	8-37
END_LOAD プロシージャ .....	8-38
RESUME_SUBSET_OF_MASTERS プロシージャ .....	8-39
<b>DBMS_OFFLINE_SNAPSHOT パッケージ .....</b>	<b>8-42</b>
サブプログラムの概要 .....	8-42
BEGIN_LOAD プロシージャ .....	8-42
END_LOAD プロシージャ .....	8-43
<b>DBMS_RECTIFIER_DIFF パッケージ .....</b>	<b>8-45</b>
サブプログラムの概要 .....	8-45
DIFFERENCES プロシージャ .....	8-45
RECTIFY プロシージャ .....	8-48
<b>DBMS_REFRESH パッケージ .....</b>	<b>8-51</b>
サブプログラムの概要 .....	8-51
ADD プロシージャ .....	8-51
CHANGE プロシージャ .....	8-52
DESTROY プロシージャ .....	8-54
MAKE プロシージャ .....	8-55

REFRESH プロシージャ .....	8-57
SUBTRACT プロシージャ .....	8-58
<b>DBMS_REPCAT パッケージ</b> .....	8-59
サブプログラムの概要 .....	8-59
ADD_GROUPED_COLUMN プロシージャ .....	8-63
ADD_MASTER_DATABASE プロシージャ .....	8-64
ADD_PRIORITY_datatype プロシージャ .....	8-65
ADD_SITE_PRIORITY_SITE プロシージャ .....	8-67
ADD_conflicttype_RESOLUTION プロシージャ .....	8-68
ALTER_MASTER_PROPAGATION プロシージャ .....	8-71
ALTER_MASTER_REPOBJECT プロシージャ .....	8-73
ALTER_PRIORITY プロシージャ .....	8-74
ALTER_PRIORITY_datatype プロシージャ .....	8-75
ALTER_SITE_PRIORITY プロシージャ .....	8-77
ALTER_SITE_PRIORITY_SITE プロシージャ .....	8-78
ALTER_SNAPSHOT_PROPAGATION プロシージャ .....	8-79
CANCEL_STATISTICS プロシージャ .....	8-80
COMMENT_ON_COLUMN_GROUP プロシージャ .....	8-81
COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY プロシージャ .....	8-82
COMMENT_ON_REPGROUP プロシージャ .....	8-83
COMMENT_ON_REPOBJECT プロシージャ .....	8-84
COMMENT_ON_REPSITES プロシージャ .....	8-85
COMMENT_ON_SNAPSHOT_REPSITES プロシージャ .....	8-86
COMMENT_ON_conflicttype_RESOLUTION プロシージャ .....	8-87
COMPARE_OLD_VALUES プロシージャ .....	8-88
CREATE_MASTER_REPGROUP プロシージャ .....	8-90
CREATE_MASTER_REPOBJECT プロシージャ .....	8-91
CREATE_SNAPSHOT_REPGROUP プロシージャ .....	8-94
CREATE_SNAPSHOT_REPOBJECT プロシージャ .....	8-95
DEFINE_COLUMN_GROUP プロシージャ .....	8-97
DEFINE_PRIORITY_GROUP プロシージャ .....	8-98
DEFINE_SITE_PRIORITY プロシージャ .....	8-99
DO_DEFERRED_REPCAT_ADMIN プロシージャ .....	8-100
DROP_COLUMN_GROUP プロシージャ .....	8-101
DROP_GROUPED_COLUMN プロシージャ .....	8-102
DROP_MASTER_REPGROUP プロシージャ .....	8-103
DROP_MASTER_REPOBJECT プロシージャ .....	8-105
DROP_PRIORITY プロシージャ .....	8-106

DROP_PRIORITY_GROUP プロシージャ .....	8-107
DROP_PRIORITY_datatype プロシージャ .....	8-108
DROP_SITE_PRIORITY プロシージャ .....	8-109
DROP_SITE_PRIORITY_SITE プロシージャ .....	8-110
DROP_SNAPSHOT_REPGROUP プロシージャ .....	8-111
DROP_SNAPSHOT_REPOBJECT プロシージャ .....	8-112
DROP_conflicttype_RESOLUTION プロシージャ .....	8-113
EXECUTE_DDL プロシージャ .....	8-114
GENERATE_REPLICATION_SUPPORT プロシージャ .....	8-116
GENERATE_SNAPSHOT_SUPPORT プロシージャ .....	8-117
MAKE_COLUMN_GROUP プロシージャ .....	8-119
PURGE_MASTER_LOG プロシージャ .....	8-120
PURGE_STATISTICS プロシージャ .....	8-121
REFRESH_SNAPSHOT_REPGROUP プロシージャ .....	8-122
REGISTER_SNAPSHOT_REPGROUP プロシージャ .....	8-123
REGISTER_STATISTICS プロシージャ .....	8-124
RELOCATE_MASTERDEF プロシージャ .....	8-125
REMOVE_MASTER_DATABASES プロシージャ .....	8-126
REPCAT_IMPORT_CHECK プロシージャ .....	8-128
RESUME_MASTER_ACTIVITY プロシージャ .....	8-129
SEND_OLD_VALUES プロシージャ .....	8-129
SET_COLUMNS プロシージャ .....	8-131
SUSPEND_MASTER_ACTIVITY プロシージャ .....	8-132
SWITCH_SNAPSHOT_MASTER プロシージャ .....	8-133
UNREGISTER_SNAPSHOT_REPGROUP プロシージャ .....	8-134
VALIDATE ファンクション .....	8-135
WAIT_MASTER_LOG プロシージャ .....	8-137
<b>DBMS_REPCAT_ADMIN パッケージ</b> .....	8-139
サブプログラムの概要 .....	8-139
GRANT_ADMIN_ANY_SCHEMA プロシージャ .....	8-139
GRANT_ADMIN_SCHEMA プロシージャ .....	8-140
REGISTER_USER_REPGROUP プロシージャ .....	8-141
REVOKE_ADMIN_ANY_SCHEMA プロシージャ .....	8-142
REVOKE_ADMIN_SCHEMA プロシージャ .....	8-143
UNREGISTER_USER_REPGROUP プロシージャ .....	8-144
<b>DBMS_REPCAT_INSTANTIATE パッケージ</b> .....	8-146
サブプログラムの概要 .....	8-146
DROP_SITE_INSTANTIATION プロシージャ .....	8-146



INstantiate_Offline ファンクション .....	8-147
Instantiate_Offline_Repapi ファンクション .....	8-149
Instantiate_Online ファンクション .....	8-151
<b>DBMS_REPCAT_RGT パッケージ</b> .....	8-154
サブプログラムの概要 .....	8-154
ALTER_REFRESH_TEMPLATE プロシージャ .....	8-156
ALTER_TEMPLATE_OBJECT プロシージャ .....	8-158
ALTER_TEMPLATE_PARM プロシージャ .....	8-160
ALTER_USER_AUTHORIZATION プロシージャ .....	8-162
ALTER_USER_PARM_VALUE プロシージャ .....	8-163
COMPARE_TEMPLATES ファンクション .....	8-165
COPY_TEMPLATE ファンクション .....	8-166
CREATE_OBJECT_FROM_EXISTING ファンクション .....	8-168
CREATE_REFRESH_TEMPLATE ファンクション .....	8-169
CREATE_TEMPLATE_OBJECT ファンクション .....	8-171
CREATE_TEMPLATE_PARM ファンクション .....	8-173
CREATE_USER_AUTHORIZATION ファンクション .....	8-176
CREATE_USER_PARM_VALUE ファンクション .....	8-177
DELETE_RUNTIME_PARMS プロシージャ .....	8-179
DROP_ALL_OBJECTS プロシージャ .....	8-179
DROP_ALL_TEMPLATE_PARMS プロシージャ .....	8-180
DROP_ALL_TEMPLATE_SITES プロシージャ .....	8-181
DROP_ALL_TEMPLATES プロシージャ .....	8-182
DROP_ALL_USER_AUTHORIZATIONS プロシージャ .....	8-182
DROP_ALL_USER_PARM_VALUES プロシージャ .....	8-183
DROP_REFRESH_TEMPLATE プロシージャ .....	8-184
DROP_SITE_INSTANTIATION プロシージャ .....	8-185
DROP_TEMPLATE_OBJECT プロシージャ .....	8-186
DROP_TEMPLATE_PARM プロシージャ .....	8-187
DROP_USER_AUTHORIZATION プロシージャ .....	8-188
DROP_USER_PARM_VALUE プロシージャ .....	8-189
GET_RUNTIME_PARM_ID ファンクション .....	8-190
INSERT_RUNTIME_PARMS プロシージャ .....	8-190
Instantiate_Offline ファンクション .....	8-192
Instantiate_Offline_Repapi ファンクション .....	8-194
Instantiate_Online ファンクション .....	8-197
LOCK_TEMPLATE_EXCLUSIVE プロシージャ .....	8-199
LOCK_TEMPLATE_SHARED プロシージャ .....	8-199

<b>DBMS_REPUTIL パッケージ</b> .....	8-200
サブプログラムの概要 .....	8-200
REPLICATION_OFF プロシージャ .....	8-200
REPLICATION_ON プロシージャ .....	8-201
REPLICATION_IS_ON ファンクション .....	8-201
FROM_REMOTE ファンクション .....	8-201
GLOBAL_NAME ファンクション .....	8-202
MAKE_INTERNAL_PKG プロシージャ .....	8-202
SYNC_UP_REP プロシージャ .....	8-203
<b>DBMS_SNAPSHOT パッケージ</b> .....	8-204
サブプログラムの概要 .....	8-204
BEGIN_TABLE_REORGANIZATION プロシージャ .....	8-204
END_TABLE_REORGANIZATION プロシージャ .....	8-205
I_AM_A_REFRESH ファンクション .....	8-206
PURGE_DIRECT_LOAD_LOG プロシージャ .....	8-206
PURGE_LOG プロシージャ .....	8-206
PURGE_SNAPSHOT_FROM_LOG プロシージャ .....	8-207
REFRESH プロシージャ .....	8-209
REFRESH_ALL_MVIEWS プロシージャ .....	8-211
REFRESH_DEPENDENT プロシージャ .....	8-212
REGISTER_SNAPSHOT プロシージャ .....	8-214
UNREGISTER_SNAPSHOT プロシージャ .....	8-215

## 9 データ・ディクショナリ・ビュー

<b>レプリケーション・カタログ・ビュー</b> .....	9-2
ALL_REPCATLOG .....	9-6
ALL_REPCAT_REFRESH_TEMPLATES .....	9-7
ALL_REPCAT_TEMPLATE_OBJECTS .....	9-8
ALL_REPCAT_TEMPLATE_PARS .....	9-10
ALL_REPCAT_TEMPLATE_SITES .....	9-12
ALL_REPCAT_USER_AUTHORIZATIONS .....	9-13
ALL_REPCAT_USER_PARM_VALUES .....	9-14
ALL_REPCOLUMN .....	9-16
ALL_REPCOLUMN_GROUP .....	9-17
ALL_REPCONFLICT .....	9-17
ALL_REPDDL .....	9-18
ALL_REPGENOBJECTS .....	9-18

ALL_REPGROUP .....	9-19
ALL_REPGROUP_PRIVILEGES .....	9-20
ALL_REPGROUPED_COLUMN .....	9-21
ALL_REPKEY_COLUMNS .....	9-21
ALL_REPOBJECT .....	9-22
ALL_REPPARAMETER_COLUMN .....	9-23
ALL_REPPRIORITY .....	9-24
ALL_REPPRIORITY_GROUP .....	9-26
ALL_REPPROP .....	9-26
ALL_REPRESOL_STATS_CONTROL .....	9-27
ALL_REPRESOLUTION .....	9-28
ALL_REPRESOLUTION_METHOD .....	9-29
ALL_REPRESOLUTION_STATISTICS .....	9-29
ALL_REPSITES .....	9-30
DBA_REPCATLOG .....	9-31
DBA_REPCAT_REFRESH_TEMPLATES .....	9-31
DBA_REPCAT_TEMPLATE_OBJECTS .....	9-32
DBA_REPCAT_TEMPLATE_PARS .....	9-32
DBA_REPCAT_TEMPLATE_SITES .....	9-32
DBA_REPCAT_USER_AUTHORIZATIONS .....	9-32
DBA_REPCAT_USER_PARM_VALUES .....	9-32
DBA_REPCOLUMN .....	9-33
DBA_REPCOLUMN_GROUP .....	9-33
DBA_REPCONFLICT .....	9-33
DBA_REPDDL .....	9-33
DBA_REPGENOBJECTS .....	9-33
DBA_REPGROUP .....	9-33
DBA_REPGROUP_PRIVILEGES .....	9-34
DBA_REPGROUPED_COLUMN .....	9-34
DBA_REPKEY_COLUMNS .....	9-34
DBA_REPOBJECT .....	9-34
DBA_REPPARAMETER_COLUMN .....	9-34
DBA_REPPRIORITY .....	9-34
DBA_REPPRIORITY_GROUP .....	9-35
DBA_REPPROP .....	9-35
DBA_REPRESOL_STATS_CONTROL .....	9-35
DBA_REPRESOLUTION .....	9-35
DBA_REPRESOLUTION_METHOD .....	9-35

DBA_REPRESOLUTION_STATISTICS .....	9-35
DBA_REPSITES .....	9-36
USER_REPCATLOG .....	9-36
USER_REPCAT_REFRESH_TEMPLATES .....	9-36
USER_REPCAT_TEMPLATE_OBJECTS .....	9-36
USER_REPCAT_TEMPLATE_PARS .....	9-37
USER_REPCAT_TEMPLATE_SITES .....	9-37
USER_REPCAT_USER_AUTHORIZATIONS .....	9-37
USER_REPCAT_USER_PARM_VALUES .....	9-37
USER_REPCOLUMN .....	9-38
USER_REPCOLUMN_GROUP .....	9-38
USER_REPCONFLICT .....	9-38
USER_REPDDL .....	9-38
USER_REPGENOBJECTS .....	9-38
USER_REPGROUP .....	9-38
USER_REPGROUP_PRIVILEGES .....	9-39
USER_REPGROUPED_COLUMN .....	9-39
USER_REPKY_COLUMNS .....	9-39
USER_REPOBJECT .....	9-39
USER_REPPARAMETER_COLUMN .....	9-39
USER_REPPRIORITY .....	9-40
USER_REPPRIORITY_GROUP .....	9-40
USER_REPPROP .....	9-40
USER_REPRESOL_STATS_CONTROL .....	9-40
USER_REPRESOLUTION .....	9-41
USER_REPRESOLUTION_METHOD .....	9-41
USER_REPRESOLUTION_STATISTICS .....	9-41
USER_REPSITES .....	9-41
<b>遅延トランザクション・ビュー</b> .....	9-42
DEFCALL .....	9-43
DEFCALLDEST .....	9-43
DEFDEFAULTDEST .....	9-43
DEFERRCOUNT .....	9-44
DEFERROR .....	9-44
DEFLOB .....	9-44
DEFPROPAGATOR .....	9-45
DEFSCHEDULE .....	9-45
DEFTRAN .....	9-46

DEFTRANDEST .....	9-46
スナップショット・ビューおよびスナップショット・リフレッシュ・グループ・ビュー .....	9-47
ALL_REFRESH .....	9-48
ALL_REFRESH_CHILDREN .....	9-49
ALL_REGISTERED_SNAPSHOTS .....	9-50
ALL_SNAPSHOT_LOGS .....	9-50
ALL_SNAPSHOT_REFRESH_TIMES .....	9-51
ALL_SNAPSHOTS .....	9-52
DBA_REFRESH .....	9-54
DBA_REFRESH_CHILDREN .....	9-54
DBA_REGISTERED_SNAPSHOT_GROUPS .....	9-54
DBA_REGISTERED_SNAPSHOTS .....	9-54
DBA_SNAPSHOT_LOGS .....	9-54
DBA_SNAPSHOT_LOG_FILTER_COLS .....	9-54
DBA_SNAPSHOT_REFRESH_TIMES .....	9-55
DBA_SNAPSHOTS .....	9-55
USER_REFRESH .....	9-55
USER_REFRESH_CHILDREN .....	9-55
USER_REGISTERED_SNAPSHOTS .....	9-55
USER_SNAPSHOTS .....	9-55
USER_SNAPSHOT_LOGS .....	9-56
USER_SNAPSHOT_REFRESH_TIMES .....	9-56

## A セキュリティ・オプション

マルチマスター・レプリケーションのセキュリティ設定 .....	A-2
トラステッド・セキュリティとアントラステッド・セキュリティ .....	A-2
スナップショット・レプリケーションのセキュリティ設定 .....	A-7
トラステッド・セキュリティとアントラステッド・セキュリティ .....	A-8

## B ユーザー定義の競合解消方法

ユーザー定義の競合解消方法 .....	B-2
競合解消方法のパラメータ .....	B-2
更新競合の解消 .....	B-3
一意性競合の解消 .....	B-3
削除競合の解消 .....	B-4
制限事項 .....	B-4

ユーザー定義の競合解消方法の例 .....	B-4
<b>ユーザー定義の競合通知方法</b> .....	B-6
競合通知ログの作成 .....	B-6
競合通知パッケージの作成 .....	B-7
<b>競合解消情報の表示</b> .....	B-10

## 索引

---

# はじめに

ここでは、次の項目を説明します。

- 概要
- 対象読者
- 構成
- 表記規則

『Oracle8i レプリケーション・マネージメント API リファレンス』では、Oracle8i および Oracle8i Enterprise Edition 製品の機能を説明します。Oracle8i と Oracle8i Enterprise Edition の基本的な機能は同一です。ただし、Enterprise Edition のみで使用可能な高度な機能がいくつかあります。これらの一部はオプションです。たとえば、パーティション化機能を使用するには、Enterprise Edition および Partitioning Option が必要です。

---

## 概要

このリファレンスでは、レプリケーション・マネージメント API を説明します。このリファレンスは、読者が『Oracle8i レプリケーション・ガイド』で説明されているレプリケーションの概念をよく理解していることを前提としています。

このリファレンスは、レプリケーション・マネージメント API の使用方法の説明を目的としており、レプリケーション・マネージメント API のクイック・リファレンス・ガイドとして利用できます。

このリファレンスの情報は、あらゆるオペレーティング・システムで稼働中の Oracle8i サーバーに適用できます。次に示す項目が含まれています。

- レプリケーション・サイトの設定
- マスター・グループの作成
- 配置テンプレートの作成
- スナップショット・グループの作成
- 競合解消
- レプリケーション環境の管理
- レプリケーション・マネージメント API リファレンス
- データ・ディクショナリ・ビュー
- セキュリティ・オプション

## 対象読者

このリファレンスは、Oracle8i のレプリケーション環境の開発およびメンテナンスを行うデータベース管理者やアプリケーション開発者を対象としています。

このリファレンスは、読者がリレーショナル・データベースの概念、分散データベースの管理方法、PL/SQL（プロシージャ型レプリケーションを使用する場合）、および Oracle のレプリケート環境を実行するオペレーティング・システムについてよく理解していることを前提としています。

また、このリファレンスを読む前に、次のドキュメントを読み、内容を理解しておく必要があります。

- 『Oracle8i 概要』
- 『Oracle8i 管理者ガイド』
- 『Oracle8i 分散システム』
- 『Oracle8i PL/SQL ユーザーズ・ガイドおよびリファレンス』
- 『Oracle8i レプリケーション・ガイド』



---

# 構成

このリファレンスは、次の章および付録から構成されます。

## 第1章「レプリケーションの概要」

レプリケーション・マネージメント API を使用したレプリケート環境の構築プロセスの概要を説明します。また、レプリケート環境の構築の前提条件も説明します。

## 第2章「レプリケーション・サイトの構築」

マスター・サイトとスナップショット・サイトの設定プロセスの詳細を説明します。この章の内容は、新しいレプリケート環境を構築する場合や、新しいマスター・サイトまたはスナップショット・サイトを既存のレプリケート環境に追加する場合に必要になります。

## 第3章「マスター・グループの作成」

マルチマスター・レプリケーション用のマスター・グループの作成方法や、スナップショット・サイト用のマスターとしてのマスター・グループの作成方法を説明します。第3章ではマスター・グループを作成し、第2章で設定した3つのマスター・サイトの間でデータをレプリケートする方法を説明します。

## 第4章「配置テンプレートの作成」

配置テンプレートを使用してスナップショット環境を構築する方法を説明します。配置テンプレートを使用すると、任意の数のスナップショット・サイトに対し、最も効果的にスナップショット環境を配布できます。

## 第5章「スナップショット・グループの作成」

スナップショット・グループを使用してスナップショット環境を構築する方法を説明します。配置テンプレートのみでは必要な環境が構築できない場合は、第5章を参照してください。この章では、スナップショット・サイトに対してスナップショット環境を構築する方法を説明します。

## 第6章「競合解消」

データの競合が発生した場合にすべてのサイトのデータを収束させるために必要な、競合解消方法を説明します。

## 第7章「API を使用したレプリケート環境の管理」

レプリケート環境を管理するために必要な管理タスクを説明します。マスター・グループの管理、レプリケート・オブジェクトの変更、オフライン・インスタンスエーションおよびその他の情報が含まれています。

## 第8章「レプリケーション・マネージメント API リファレンス」

レプリケート環境をインプリメントするためのパッケージ・プロシージャおよびファンクションのパラメータと、発生する可能性のある例外を説明します。

## 第9章「データ・ディクショナリ・ビュー」

遅延トランザクション、読取り専用スナップショットおよび Oracle レプリケーションに関するビューを説明します。

## 付録 A「セキュリティ・オプション」

レプリケーション・マネージメント API を使用したマルチマスター・レプリケーションとスナップショット・レプリケーションのセキュリティ設定を説明します。

---

### 付録 B「ユーザー定義の競合解消方法」

レプリケーション・マネージメント API を使用したユーザー定義の競合解消方法および通知  
ファンクションの作成方法を説明します。

### 変更点

主要な変更点は次のとおりです。

- 付録 A「セキュリティ・オプション」および付録 B「ユーザー定義の競合解消方法」を  
追加しました。

## 表記規則

このリファレンスでは、数種類のフォントを使用して、異なるタイプの情報を示していま  
す。

### 特別な注意

本文では、次の 3 つのレベルの注記を使用して、特に重要な情報を示しています。

---

---

**注意：** 特別な情報または補足的な情報を示します。

---

---

**関連項目：** さらに詳しい情報の参照先を示します。

---

---

**注意：** システムやデータに発生する可能性のあるダメージの重要な情報  
を示します。

---

---

### 文字表記について

次の項では、このリファレンスでの表記規則を説明します。

#### 大文字

大文字は、文のキーワード、オブジェクト名、初期化パラメータ名、  
データ・ディクショナリ・ビューなどを強調するために使用されます。

たとえば、" プライベート・ロールバック・セグメントを作成する場合、  
名前を ROLLBACK\_SEGMENTS 初期化パラメータに含める必要があり  
ます。" と表記します。

---

## コード例

SQL、Server Manager 行モード、SQL\*Plus コマンドおよび SQL\*Plus 文は、段落内のテキスト部分とは別の行に固定幅フォントで表示されます。次にその例を示します。

```
INSERT INTO emp (empno, ename) VALUES (1000, 'SMITH');
```

```
ALTER TABLESPACE users ADD DATAFILE 'users2.ora' SIZE 50K;
```

サンプル文には、カンマや引用符などの区切り記号を使用できます。文中のすべての区切り記号は必須です。文の終了は、セミコロン (;) で示されています。アプリケーションによっては、文の終了を示すためのセミコロンや記号が不要な場合もあります。

文の例で使用されている大文字は、Oracle SQL のキーワードであることを示します。ただし、文を実際に実行した場合、キーワードの大文字と小文字は区別されません。文の例で使用されている小文字は、例のための内容であることを示しています。たとえば、表名、列名、ファイル名などが小文字で表されます。



---

# レプリケーションの概要

この章では、レプリケーション・マネージメント API を使用してレプリケート環境を構築するプロセスを説明します。次に、項目を示します。

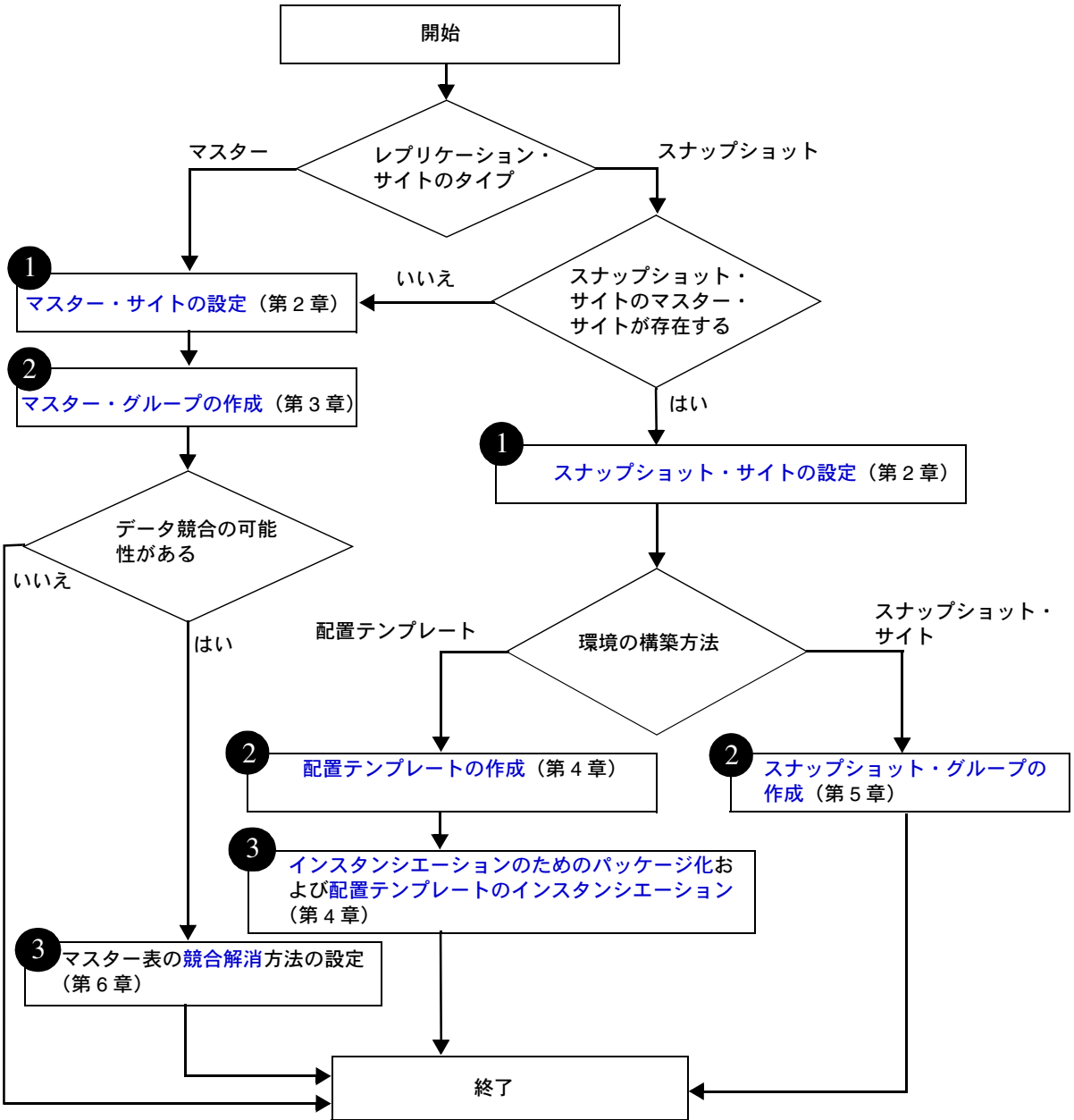
- [レプリケート環境構築の概要](#)
- [作業を開始する前に](#)

## レプリケート環境構築の概要

[図 1-1](#) は、レプリケート環境を構築するための基本的なステップを示します。どのようなタイプのレプリケーション・サイトを構築する場合でも、まずサイトを設定します。

レプリケーション・サイトを設定した後、マスター・グループとスナップショット・グループを作成します。レプリケーション環境が構築できた後、[第 6 章](#)と[第 7 章](#)を参照して、競合解消とレプリケート環境の管理を確実に理解します。

図 1-1 レプリケート環境構築のプロセス



## 作業を開始する前に

レプリケーション・サイトを設定する前に、次の事柄を確認します。

- 初期化パラメータ・ファイルの GLOBAL\_NAMES が TRUE に設定されていること
- 各マスター・サイトに必要なジョブ・プロセスが割り当てられていること

## グローバル名

レプリケート環境が正しく機能するためには、初期化パラメータ・ファイルの GLOBAL\_NAMES 初期化パラメータを TRUE に設定する必要があります。初期化パラメータ・ファイルでは、次のようにコメントとパラメータが記述されています。

```
# Global Naming -- enforce that a dblink has same name as the db it connects to
global_names = TRUE
```

この初期化パラメータは、構築したレプリケート環境（マスター・サイトとスナップショット・サイトの両方を含みます）に関係するデータベースごとに、TRUE に設定する必要があります。

## ジョブ・プロセス

レプリケート環境で自動化処理を行うためには、十分なジョブ・プロセス（SNP バックグラウンド・プロセスと呼ばれる場合もあります）を割り当てする必要があります。自動化処理とは、遅延トランザクション・キューの自動的な伝播、パージおよびスナップショットのリフレッシュなどです。

マルチマスター・レプリケーションでは、サイトごとに他のマスター・サイトに対するスケジュール・リンクがあります。たとえば、マスター・サイトが6つある場合、サイトごとに他の5つのサイトに対するスケジュール・リンクがあります。通常、1つのスケジュール・リンクには1つのプロセスが必要です。また、遅延トランザクション・キューやその他のユーザー定義ジョブをパージするためのジョブ・プロセスを追加することもできます。

スナップショット・レプリケーションの場合は、各スナップショット・サイトには通常、マスター・データベースに対する1つのスケジュール・リンク、および最低1つのジョブ・プロセスが必要です。通常、パージ・スケジュール、ユーザー定義ジョブおよびスケジュール・リンクに応じて、スナップショット・サイトには1～3つのジョブ・プロセスが必要です。一方、ユーザーがアプリケーションのインターフェースを使用して手動でスナップショットをリフレッシュする場合は、スケジュール・リンクを作成する必要はなく、スナップショット・サイトで必要なのは通常より1つ少ないジョブ・プロセスとなります。

ジョブ・プロセスの数の他に、ジョブ間隔も定義する必要があります。ジョブ間隔は、ジョブ・プロセスをアクティブにして、保留になっている操作（キューのプッシュなど）を実行する間隔を決定します。デフォルト値の60秒は、ほとんどのレプリケート環境では適切な値です。ただし、パフォーマンスを最適化するために、要求によってはこの値を変更した方がよい場合もあります。たとえば、20秒おきに変更を伝播させる場合は、60秒よりも間隔



を狭めます。これに対して、変更を日に 1 回伝播させる場合は、SNP プロセスを使用して保留になっている操作を 1 時間に 1 回調べます。

通常、ジョブ・プロセスは、初期化パラメータ・ファイルの "Oracle replication" ヘッダーの次に定義されています。

```
job_queue_processes = 7  
job_queue_interval = 60
```

初期化パラメータ・ファイルの内容を変更した場合は、データベースを再起動して新しい設定を反映させます。

**関連項目：** Oracle レプリケーションで重要な初期化パラメータの詳細は、『Oracle8i レプリケーション・ガイド』の第 7 章「レプリケーション環境の計画」を参照してください。データベースの再起動の詳細は、『Oracle8i 管理者ガイド』を参照してください。



---

## レプリケーション・サイトの構築

この章では、レプリケーション・マネージメント API を使用して、マスター・レプリケーション・サイトとスナップショット・レプリケーション・サイトを設定する方法を説明します。次に、項目を示します。

- [レプリケーション・サイト設定の概要](#)
- [マスター・サイトの設定](#)
- [スナップショット・サイトの設定](#)

## レプリケーション・サイト設定の概要

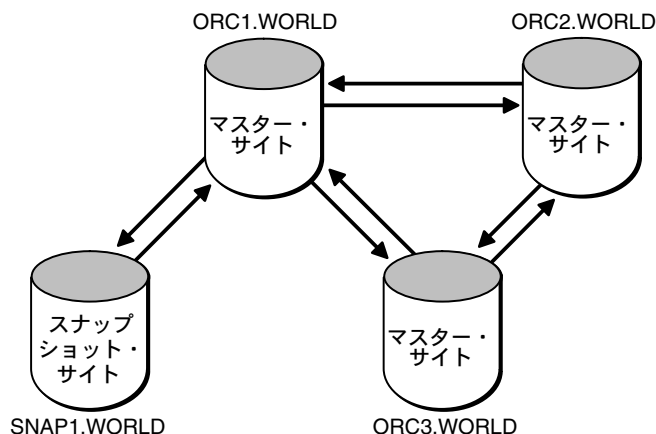
レプリケート環境を構築する前に、その環境に関するサイトを設定する必要があります。[図 2-2](#) と [図 2-3](#) に示すように、マスター・サイトとスナップショット・サイトでは設定方法が異なります。

この章では、次のデータベースが設定されていることを前提とします。

- 4 つのデータベース
  - ORC1.WORLD
  - ORC2.WORLD
  - ORC3.WORLD
  - SNAP1.WORLD

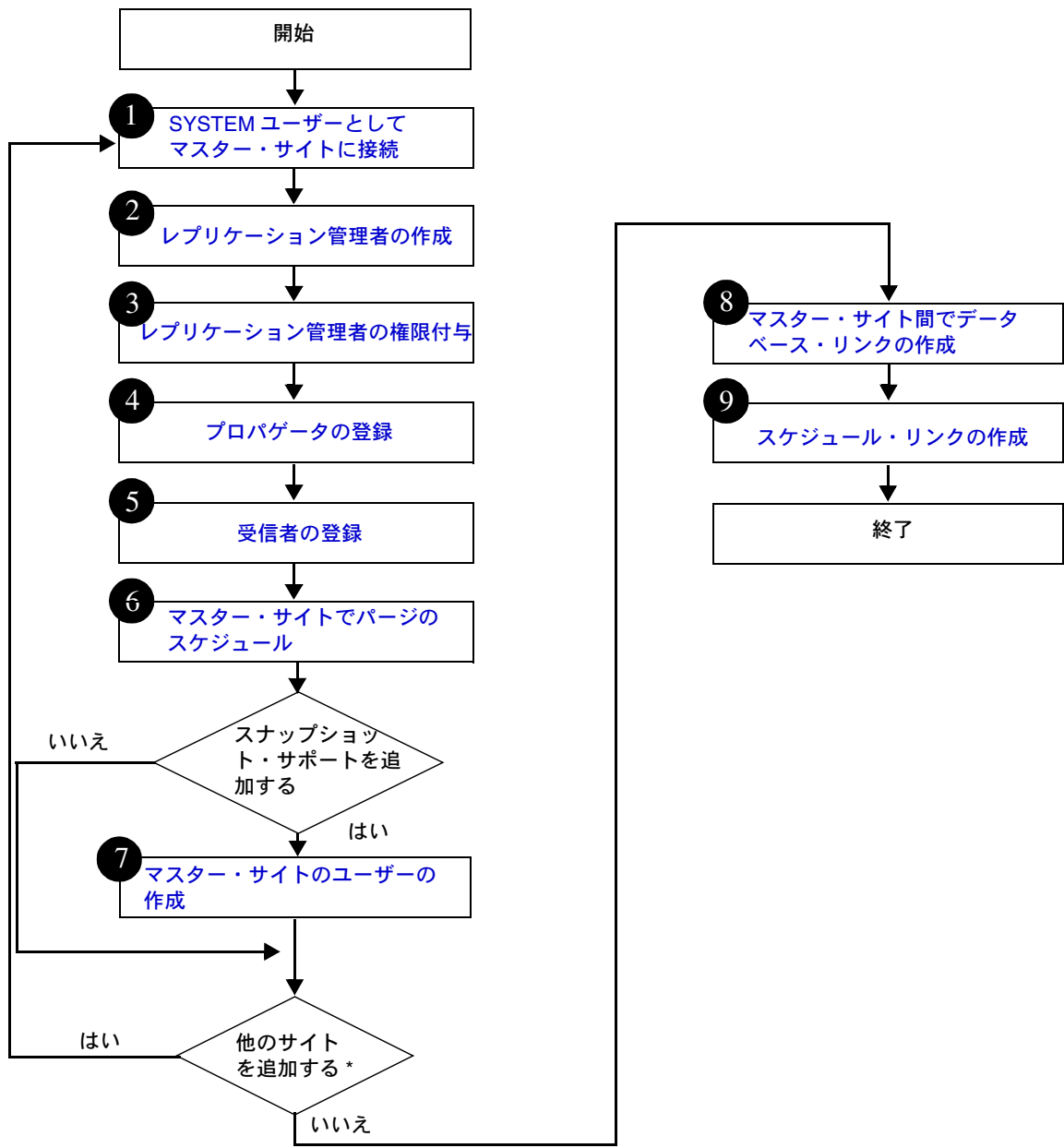
第 2 章～第 6 章では、[図 2-1](#) に示すレプリケーション環境で作業します。この環境をこの章で構築していきます。

**図 2-1 3 つのマスター・サイトと 1 つのスナップショット・サイト**



新しいマスター・サイトを構築する場合は[図 2-2](#)、新しいスナップショット・サイトを構築する場合は[図 2-3](#) の手順に従います。

図 2-2 マスター・サイトの設定



\* 複数のマスター・サイト(マルチマスター・レプリケーション)は、Oracle Enterprise Edition のみで使用可能です。

## マスター・サイトの設定

```

/*****
ステップ 1 @ ORC1.WORLD:
SYSTEM ユーザーとしてマスター・サイトに接続
*****/

-- レプリケーション用の設定を行うデータベースに、SYSTEM として接続します。
-- ORC1.WORLD を設定した後に、ORC2.WORLD サイトおよび ORC3.WORLD サイトを
-- 設定するには、それぞれ 2-7 ページおよび 2-9 ページの
-- ステップ 1 から再び作業を実行します。

CONNECT system/manager@orc1.world

/*****
ステップ 2 @ ORC1.WORLD:
レプリケーション管理者の作成
*****/

-- レプリケーション管理者には、レプリケート環境を作成して管理するために
-- 必要な権限が与えられます。レプリケート環境の一部となる各データベースごとに、
-- レプリケーション管理者が必要です。

CREATE USER repadmin IDENTIFIED BY repadmin;

/*****
ステップ 3 @ ORC1.WORLD:
レプリケーション管理者の権限付与

GRANT ADMIN_ANY_SCHEMA API の詳細は、8-139 ページの
「GRANT_ADMIN_ANY_SCHEMA プロシージャ」を参照してください。
*****/

-- GRANT ADMIN_ANY_SCHEMA プロシージャ API を実行すると、
-- レプリケート環境を作成して管理するための強力な権限が、
-- レプリケーション管理者に与えられます。

BEGIN
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
        USERNAME => 'repadmin');
END;
/

-- レプリケーション管理者 (REPADMIN) がレプリケート表のためのスナップショット・ログを
-- 作成するには、REPADMIN に、COMMENT ANY TABLE
-- および LOCK ANY TABLE 権限を与えます。
```

```

/*****
ステップ 4 @ ORCL.WORLD:
プロパゲータの登録

REGISTER_PROPAGATOR API の詳細は、8-27 ページの
「REGISTER_PROPAGATOR プロシージャ」を参照してください。
*****/

-- プロパゲータには、他のマスター・サイトへ遅延トランザクション・キューを
-- 伝播する責任があります。

BEGIN
    DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
        USERNAME => 'repadmin');
END;
/

/*****
ステップ 5 @ ORCL.WORLD:
受信者の登録

REGISTER_USER_REPGROUP API の詳細は、8-141 ページの
「REGISTER_USER_REPGROUP プロシージャ」を参照してください。
*****/

-- 受信者は、プロパゲータによって他のマスター・サイトから送信された、
-- 伝播遅延トランザクションを受信します。

BEGIN
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        USERNAME => 'repadmin',
        PRIVILEGE_TYPE => 'receiver',
        LIST_OF_GNAMES => NULL);
END;
/

```

```
/******  
ステップ 6 @ ORCL.WORLD:  
マスター・サイトでパージのスケジュール
```

SCHEDULE\_PURGE API の詳細は、8-27 ページの  
「[SCHEDULE\\_PURGE プロシージャ](#)」を参照してください。

```
*****/
```

```
-- チェック中の遅延トランザクション・キューのサイズを保つために、正常に  
-- 終了したトランザクションをパージする必要があります。  
-- SCHEDULE_PURGE API で、パージ・プロセスが自動化されます。  
-- このプロシージャは、レプリケーション管理者として実行する必要があります。
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
BEGIN
```

```
    DBMS_DEFER_SYS.SCHEDULE_PURGE (  
        NEXT_DATE => SYSDATE,  
        INTERVAL  => 'SYSDATE + 1/24',  
        DELAY_SECONDS => 0,  
        ROLLBACK_SEGMENT => '');
```

```
END;
```

```
/
```

```
/******
```

```
ステップ 7:  
マスター・サイトのユーザーの作成
```

```
*****/
```

```
-- ステップ 7a: プロキシ・スナップショット管理者の作成  
-- プロキシ・スナップショット管理者は、スナップショット・サイトに存在する  
-- スナップショット管理者のかわりに、ターゲットのマスター・サイトでタスクを実行します。  
-- 『Oracle8i レプリケーション・ガイド』の  
-- 「スナップショット・レプリケーションのセキュリティ設定」を参照してください。
```

```
CONNECT system/manager@orc1.world
```

```
CREATE USER proxy_snapadmin IDENTIFIED BY proxy_snapadmin;
```

```
BEGIN
```

```
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (  
        USERNAME => 'proxy_snapadmin',  
        PRIVILEGE_TYPE => 'proxy_snapadmin',  
        LIST_OF_GNAMES => NULL);
```

```
END;
```

```
/
```

```
-- ステップ 7b: プロキシ・リフレッシャの作成
```



-- プロキシ・リフレッシャは、スナップショット・サイトに存在するリフレッシャのかわりに、  
 -- マスター・サイトでタスクを実行します。

```
CREATE USER proxy_refresher IDENTIFIED BY proxy_refresher;
```

```
GRANT CREATE SESSION TO proxy_refresher;
GRANT SELECT ANY TABLE TO proxy_refresher;
```

```

/*****
ステップ 1 @ ORC2.WORLD:
SYSTEM ユーザーとして接続
*****/

```

-- **注意:**  
 -- 複数のマスター・サイト（マルチマスター・レプリケーション）は、  
 -- Oracle8i Enterprise Edition のみで使用できます。Oracle8i Enterprise Edition を  
 -- 使用していない場合は、2-11 ページのステップ 8 に進んでください。

-- レプリケーション用の設定を行うデータベースには、SYSTEM として接続する  
 -- 必要があります。ORC2.WORLD を設定した後に、ORC3.WORLD サイトを  
 -- 設定するには、2-9 ページのステップ 1 から再び作業を実行します。

```
CONNECT system/manager@orc2.world
```

```

/*****
ステップ 2 @ ORC2.WORLD:
レプリケーション管理者の作成
*****/

```

-- レプリケーション管理者には、レプリケート環境を作成して管理するために  
 -- 必要な権限が与えられます。レプリケート環境の一部となる各データベースごとに、  
 -- レプリケーション管理者が必要です。

```
CREATE USER repadmin IDENTIFIED BY repadmin;
```

```

/*****
ステップ 3 @ ORC2.WORLD:
レプリケーション管理者の権限付与

```

GRANT ADMIN\_ANY\_SCHEMA API の詳細は、8-139 ページの  
[「GRANT\\_ADMIN\\_ANY\\_SCHEMA プロシージャ」](#)を参照してください。

```
*****/
```

-- GRANT\_ADMIN\_ANY\_SCHEMA プロシージャ API を実行すると、  
 -- レプリケート環境を作成して管理するための強力な権限が、  
 -- レプリケーション管理者に与えられます。

```
BEGIN
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
        USERNAME => 'repadmin');
END;
/

-- レプリケーション管理者 (REPADMIN) がレプリケート表のためのスナップショット・ログを
-- 作成するには、REPADMIN に、COMMENT ANY TABLE および LOCK ANY TABLE 権限を与えます。
/*****
ステップ 4 @ ORC2.WORLD:
プロパゲータの登録

REGISTER_PROPAGATOR API の詳細は、8-27 ページの
「REGISTER_PROPAGATOR プロシージャ」を参照してください。
*****/

-- プロパゲータには、他のマスター・サイトへ遅延トランザクション・キューを
-- 伝播する責任があります。

BEGIN
    DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
        USERNAME => 'repadmin');
END;
/

/*****
ステップ 5 @ ORC2.WORLD:
受信者の登録

REGISTER_USER_REPGROUP API の詳細は、8-141 ページの
「REGISTER_USER_REPGROUP プロシージャ」を参照してください。
*****/

-- 受信者は、プロパゲータによって他のマスター・サイトから送信された、
-- 伝播遅延トランザクションを受信します。

BEGIN
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        USERNAME => 'repadmin',
        PRIVILEGE_TYPE => 'receiver',
        LIST_OF_GNAMES => NULL);
END;
/

/*****
ステップ 6 @ ORC2.WORLD:
マスター・サイトでバージのスケジュール
```

```

SCHEDULE_PURGE API の詳細は、8-27 ページの
「SCHEDULE\_PURGE プロシージャ」を参照してください。
*****/

-- チェック中の遅延トランザクション・キューのサイズを保つために、正常に
-- 終了したトランザクションをパージする必要があります。
-- SCHEDULE_PURGE プロシージャ API で、パージ・プロセスが自動化されます。
-- このプロシージャは、レプリケーション管理者として実行する必要があります。

CONNECT repadmin/repadmin@orc2.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PURGE (
        NEXT_DATE => SYSDATE,
        INTERVAL => 'SYSDATE + 1/24',
        DELAY_SECONDS => 0,
        ROLLBACK_SEGMENT => '');
END;
/

/*****
ステップ 1 @ ORC3.WORLD:
SYSTEM ユーザーとして接続
*****/

-- 注意:
-- 複数のマスター・サイト（マルチマスター・レプリケーション）は、
-- Oracle8i Enterprise Edition のみで使用できます。Oracle8i Enterprise Edition を
-- 使用していない場合は、2-11 ページのステップ 8 に進んでください。

-- レプリケーション用の設定を行うデータベースには、SYSTEM として接続する
-- 必要があります。

CONNECT system/manager@orc3.world

/*****
ステップ 2 @ ORC3.WORLD:
レプリケーション管理者の作成
*****/

-- レプリケーション管理者には、レプリケート環境を作成して管理するために
-- 必要な権限が与えられます。レプリケート環境の一部となるデータベースごとに、
-- レプリケーション管理者が必要です。

CREATE USER repadmin IDENTIFIED BY repadmin;

```

```

/*****
ステップ 3 @ ORC3.WORLD:
レプリケーション管理者への権限付与

GRANT_ADMIN_ANY_SCHEMA API の詳細は、8-139 ページの
「GRANT_ADMIN_ANY_SCHEMA プロシージャ」を参照してください。
*****/

--GRANT_ADMIN_ANY_SCHEMA プロシージャ API を実行すると、
-- レプリケート環境を作成して管理するための強力な権限が、
-- レプリケーション管理者に与えられます。

BEGIN
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
        USERNAME => 'repadmin');
END;
/

-- レプリケーション管理者 (REPADMIN) がレプリケート表のためのスナップショット・ログを
-- 作成するには、REPADMIN に、COMMENT ANY TABLE
-- および LOCK ANY TABLE 権限を与えます。
/*****
ステップ 4 @ ORC3.WORLD:
プロパゲータの登録

REGISTER_PROPAGATOR API の詳細は、8-27 ページの
「REGISTER_PROPAGATOR プロシージャ」を参照してください。
*****/

-- プロパゲータには、他のマスター・サイトへ遅延トランザクション・キューを
-- 伝播する責任があります。

BEGIN
    DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
        USERNAME => 'repadmin');
END;
/

/*****
ステップ 5 @ ORC3.WORLD:
受信者の登録

REGISTER_USER_REPGROUP API の詳細は、8-141 ページの
「REGISTER_USER_REPGROUP プロシージャ」を参照してください。
*****/

-- 受信者は、プロパゲータによって他のマスター・サイトから送信された、
```

-- 伝播遅延トランザクションを受信します。

```
BEGIN
  DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
    USERNAME => 'repadmin',
    PRIVILEGE_TYPE => 'receiver',
    LIST_OF_GNAMES => NULL);
END;
/
```

/\*\*\*\*\*

ステップ 6 @ ORC3.WORLD:

マスター・サイトでパージのスケジュール

SCHEDULE\_PURGE API の詳細は、8-27 ページの  
「[SCHEDULE\\_PURGE プロシージャ](#)」を参照してください。

\*\*\*\*\*/

-- チェック中の遅延トランザクション・キューのサイズを保つために、正常に  
-- 終了したトランザクションをパージする必要があります。  
-- SCHEDULE\_PURGE プロシージャ API で、パージ・プロセスが自動化されます。  
-- このプロシージャは、レプリケーション管理者として実行する必要があります。

CONNECT repadmin/repadmin@orc3.world

```
BEGIN
  DBMS_DEFER_SYS.SCHEDULE_PURGE (
    NEXT_DATE => SYSDATE,
    INTERVAL => 'SYSDATE + 1/24',
    DELAY_SECONDS => 0,
    ROLLBACK_SEGMENT => '');
END;
/
```

/\*\*\*\*\*

ステップ 7:

マスター・サイト間でデータベース・リンクの作成

データベース・リンクは、レプリケーション・サイト間でデータをレプリケートするのに  
必要な分散メカニズムを提供します。

詳細は、『Oracle8i 分散システム』を参照してください。

\*\*\*\*\*/

-- プライベート・データベース・リンクを作成する前に、各プライベート・データベース・リンクに  
-- 使用されることになる、パブリック・データベース・リンクを  
-- 作成する必要があります。その後、設定した各マスター・サイトの  
-- すべてのレプリケーション管理者を結ぶデータベース・リンクを作成します。

```
CONNECT system/manager@orc1.world
CREATE PUBLIC DATABASE LINK orc2.world USING 'orc2.world';
CREATE PUBLIC DATABASE LINK orc3.world USING 'orc3.world';

CONNECT repadmin/repadmin@orc1.world
CREATE DATABASE LINK orc2.world CONNECT TO repadmin IDENTIFIED BY repadmin;
CREATE DATABASE LINK orc3.world CONNECT TO repadmin IDENTIFIED BY repadmin;

CONNECT system/manager@orc2.world
CREATE PUBLIC DATABASE LINK orc1.world USING 'orc1.world';
CREATE PUBLIC DATABASE LINK orc3.world USING 'orc3.world';

CONNECT repadmin/repadmin@orc2.world
CREATE DATABASE LINK orc1.world CONNECT TO repadmin IDENTIFIED BY repadmin;
CREATE DATABASE LINK orc3.world CONNECT TO repadmin IDENTIFIED BY repadmin;

CONNECT system/manager@orc3.world
CREATE PUBLIC DATABASE LINK orc1.world USING 'orc1.world';
CREATE PUBLIC DATABASE LINK orc2.world USING 'orc2.world';

CONNECT repadmin/repadmin@orc3.world
CREATE DATABASE LINK orc1.world CONNECT TO repadmin IDENTIFIED BY repadmin;
CREATE DATABASE LINK orc2.world CONNECT TO repadmin IDENTIFIED BY repadmin;
```

/\*\*\*\*\*

ステップ 8:

スケジュール・リンクの作成

[SCHEDULE\\_PUSH](#) プロシージャを実行するとデータベース・リンクが定義され、スケジュール・リンクが作成されます。

詳細は 8-29 ページの「[SCHEDULE\\_PUSH プロシージャ](#)」を参照してください。

\*\*\*\*\*/

- スケジュール・リンクで、遅延トランザクション・キューが他のマスター・サイトに
- 伝播される頻度を決定します。ステップ 7 で作成した各データベース・リンクに対し、
- [SCHEDULE\\_PUSH](#) プロシージャを実行する必要があります。対象の
- データベース・リンクは、[SCHEDULE\\_PUSH](#) プロシージャの DESTINATION
- パラメータに指定します。

```
CONNECT repadmin/repadmin@orc1.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        DESTINATION => 'orc2.world',
        INTERVAL => 'SYSDATE + 10 / (24 * 60)',
        NEXT_DATE => SYSDATE);
END;
/
```

```

BEGIN
DBMS_DEFER_SYS.SCHEDULE_PUSH (
    DESTINATION => 'orc3.world',
    INTERVAL => 'SYSDATE + 10 / (24 * 60)',
    NEXT_DATE => SYSDATE);
END;
/

CONNECT repadmin/repadmin@orc2.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        DESTINATION => 'orc1.world',
        INTERVAL => 'SYSDATE + 10 / (24 * 60)',
        NEXT_DATE => SYSDATE);
END;
/

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        DESTINATION => 'orc3.world',
        INTERVAL => 'SYSDATE + 10 / (24 * 60)',
        NEXT_DATE => SYSDATE);
END;
/

CONNECT repadmin/repadmin@orc3.world

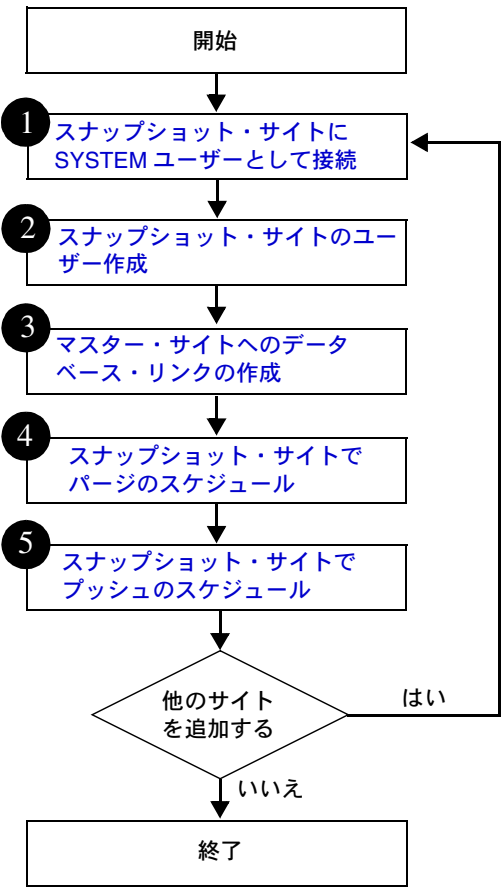
BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        DESTINATION => 'orc1.world',
        INTERVAL => 'SYSDATE + 10 / (24 * 60)',
        NEXT_DATE => SYSDATE);
END;
/

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        DESTINATION => 'orc2.world',
        INTERVAL => 'SYSDATE + 10 / (24 * 60)',
        NEXT_DATE => SYSDATE);
END;
/

```

# スナップショット・サイトの設定

図 2-3 スナップショット・サイトの設定





```

/*****
ステップ 1:
スナップショット・サイトに SYSTEM ユーザーとして接続
*****/
-- スナップショット・サイトとして設定するデータベースには、SYSTEM として
-- 接続する必要があります。

CONNECT system/manager@snap1.world

/*****
ステップ 2:
スナップショット・サイトのユーザー作成
*****/

-- スナップショット・サイトに必要なユーザーには、何種類かあります。
-- 必要なユーザーは、次のとおりです。
--   スナップショット管理者
--   プロパゲータ
--   リフレッシャ

-- ステップ 2a: スナップショット管理者の作成
-- スナップショット管理者には、スナップショット・サイトを作成して管理する
-- 責任があります。GRANT ADMIN ANY_SCHEMA プロシージャを実行すると、
-- 適切な権限がスナップショット管理者に与えられます。

CREATE USER snapadmin IDENTIFIED BY snapadmin;

BEGIN
  DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
    USERNAME => 'snapadmin');
END;
/

-- ステップ 2b: プロパゲータの作成
-- プロパゲータには、遅延トランザクション・キューをターゲット・マスター・サイトに
-- 伝播する責任があります。

CREATE USER propagator IDENTIFIED BY propagator;

BEGIN
  DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
    USERNAME => 'propagator');
END;
/

-- ステップ 2c: リフレッシャの作成
-- リフレッシャには、ターゲット・マスター・サイトでレプリケート表に行われた

```

```
-- 変更を、スナップショット・サイトに受信する責任があります。

CREATE USER refresher IDENTIFIED BY refresher;

GRANT CREATE SESSION TO refresher;
GRANT ALTER ANY SNAPSHOT TO refresher;

/*****
ステップ 3:
マスター・サイトへのデータベース・リンクの作成
*****/

-- ステップ 3a: パブリック・データベース・リンクの作成

CONNECT system/manager@snap1.world

CREATE PUBLIC DATABASE LINK orcl.world USING 'orcl.world';

-- ステップ 3b: スナップショット管理者のデータベース・リンクの作成
-- スナップショット・サイトのスナップショット管理者からマスター・サイトの
-- プロキシスナップショット管理者への、データベース・リンクを作成する必要がある
-- あります。

CONNECT snapadmin/snapadmin@snap1.world;

CREATE DATABASE LINK orcl.world
  CONNECT TO proxy_snapadmin IDENTIFIED BY proxy_snapadmin;

-- ステップ 3c: プロパゲータ受信者のデータベース・リンクの作成
-- スナップショット・サイトのプロパゲータからマスター・サイトの受信者への、
-- データベース・リンクを作成する必要があります。(受信者は、マスター・グループを
-- 作成したときに定義されています。詳細は、2-8 ページの
-- 「受信者の登録」を参照してください。)

CONNECT propagator/propagator@snap1.world

CREATE DATABASE LINK orcl.world
  CONNECT TO repadmin IDENTIFIED BY repadmin;

/*****
ステップ 4:
スナップショット・サイトでページのスケジュール

SCHEDULE_PURGE API の詳細は、8-27 ページの「SCHEDULE\_PURGE プロシージャ」
を参照してください。
*****/
```

```
-- チェック中の遅延トランザクション・キューのサイズを保つために、正常に
-- 終了したトランザクションをパージする必要があります。
-- SCHEDULE_PURGE プロシージャ API で、パージ・プロセスが自動化されます。
-- スナップショット・サイトに読み取り専用のスナップショットのみが含まれている場合は、
-- このプロシージャを実行する必要はありません。
```

```
CONNECT snapadmin/snapadmin@snap1.world
```

```
BEGIN
```

```
    DBMS_DEFER_SYS.SCHEDULE_PURGE (
        NEXT_DATE => SYSDATE,
        INTERVAL => 'SYSDATE + 1/24',
        DELAY_SECONDS => 0,
        ROLLBACK_SEGMENT => '');
```

```
END;
```

```
/
```

```
/*****
```

```
ステップ 5:
```

```
スナップショット・サイトでプッシュのスケジュール
```

```
SCHEDULE_PUSH API の詳細は、8-29 ページの「SCHEDULE\_PUSH プロシージャ」  
を参照してください。
```

```
*****/
```

```
-- 遅延トランザクション・キューをターゲット・マスター・サイトに伝播する必要が  
-- あるときに、SCHEDULE_PUSH プロシージャ API でスケジューリングします。
```

```
CONNECT snapadmin/snapadmin@snap1.world
```

```
BEGIN
```

```
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        DESTINATION => 'orc1.world',
        INTERVAL => 'SYSDATE + 1/24',
        NEXT_DATE => SYSDATE,
        STOP_ON_ERROR => FALSE,
        DELAY_SECONDS => 0,
        PARALLELISM => 0);
```

```
END;
```

```
/
```



---

## マスター・グループの作成

この章では、マスター・レプリケーション・サイトでマスター・グループを作成する方法を説明します。次に、項目を示します。

- マスター・グループ作成の概要
- マスター・グループの作成

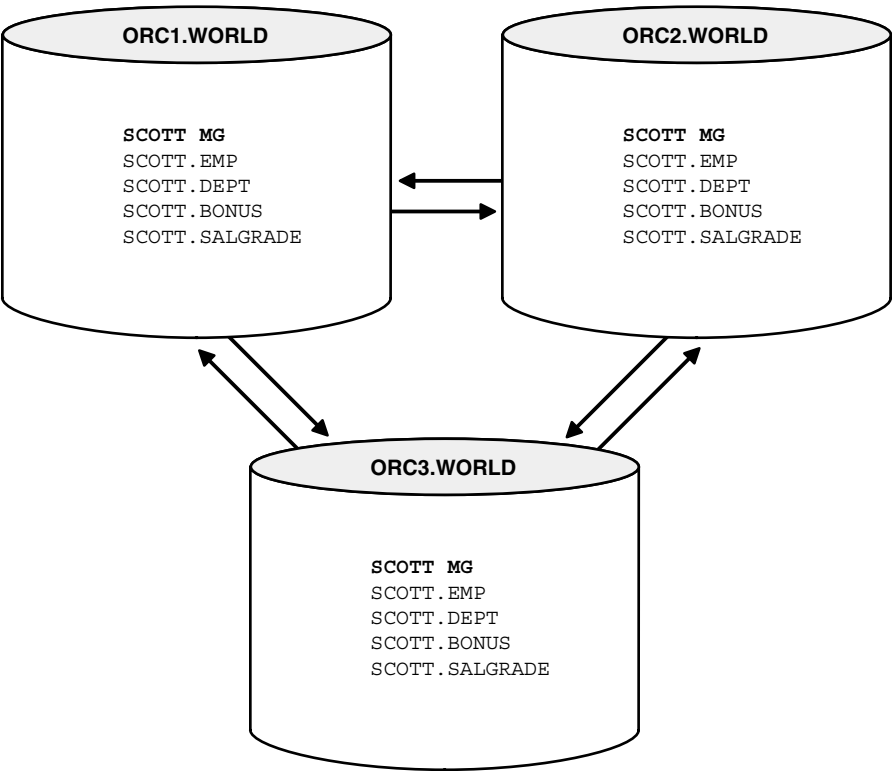
# マスター・グループ作成の概要

マスター・サイトを設定した後、マスター・グループを作成できます。レプリケート環境を正しく構築するには、[図 3-2](#) に示すステップに従う必要があります。

**関連項目：** マスター・サイトの設定の詳細は、2-1 ページの「[レプリケーション・サイトの構築](#)」を参照してください。

この章では、SCOTT\_MG マスター・グループを作成して、[図 3-1](#) に示すオブジェクトをレプリケートします。

**図 3-1 EMP、DEPT、BONUS および SALGRADE の全サイトでのレプリケーション**



## 作業を開始する前に

この章で示すスクリプトを正しく実行するには、SCOTT スキーマが ORC1.WORLD（任意で ORC2.WORLD と ORC3.WORLD）に存在し、次のオブジェクトを含むことを前提とします。

- EMP
- DEPT
- BONUS
- SALGRADE

SCOTT スキーマが ORC1.WORLD に存在しないか、または SCOTT オブジェクトが存在しない場合は、Oracle データベースで提供されるスクリプトを実行して、サンプルのスキーマ SCOTT とそれに対応するオブジェクトを作成します。

スクリプトを実行するには、次のステップに従います。

1. ORC1.WORLD にユーザー SYSTEM として接続します。

```
CONNECT system/manager@orc1.world
```

2. SCOTT スキーマが存在する場合は、ステップ 3 に進みます。存在しない場合は、次のように入力して、ユーザー SCOTT を作成します。

```
CREATE USER scott IDENTIFIED BY tiger;
```

3. `ORACLE_HOME\%rdbs%\admin` ディレクトリにある、`utlsamp1.sql` スクリプトを実行します。

このスクリプトを正しく実行するには、スキーマ SCOTT が存在することと、IDENTIFIED BY tiger が指定されていることが必要です。このスキーマが存在しない場合は、ステップ 2 に従ってください。

---

**注意：** 同じコンピュータ上で複数のデータベース・インスタンスを実行している場合は、`utlsamp1.sql` スクリプト内の `CONNECT` 文字列を変更して、ターゲット・データベースを含むようにします。たとえば、次の文字列

```
CONNECT scott/tiger
```

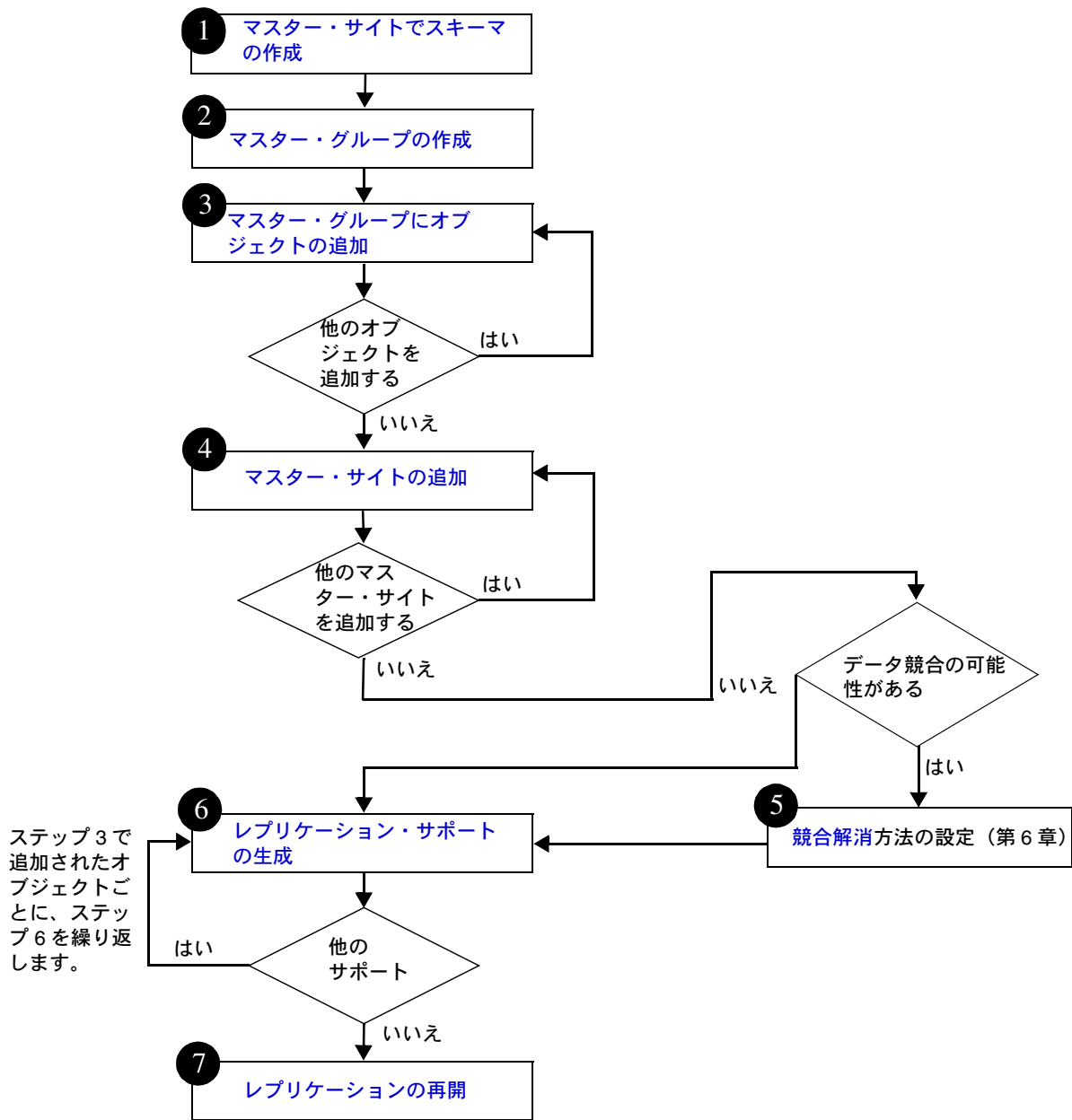
を次の文字列に置き換えます。

```
CONNECT scott/tiger@orc1.world
```

---

ここまでのステップを完了すると、EMP、DEPT、BONUS および SALGRADE の各表の最新コピーが作成されます。

図 3-2 マスター・グループの作成





# マスター・グループの作成

```

/*****
ステップ 1:
マスター・サイトでスキーマの作成
*****/

```

```

CONNECT system/manager@orc2.world;
CREATE USER scott IDENTIFIED BY tiger;
GRANT CONNECT, RESOURCE TO scott;

```

```

CONNECT system/manager@orc3.world;
CREATE USER scott IDENTIFIED BY tiger;
GRANT CONNECT, RESOURCE TO scott;

```

```

/*****
ステップ 2:
マスター・グループの作成
*****/

```

```

-- 新規マスター・グループを作成するために、CREATE_MASTER_REPGROUP
-- という API を使用します。マスター・グループにオブジェクトを追加したり、
-- その他のレプリケーション管理タスクを実行するときには、このステップ中に定義した
-- マスター・グループ名を参照してください。次のプロシージャを実行できるのは、
-- レプリケーション管理者のみです。

```

```

CONNECT repadmin/repadmin@orc1.world

```

```

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPGROUP (
        GNAME => 'scott_mg');
END;
/

```

```

/*****
ステップ 3:
マスター・グループにオブジェクトの追加
*****/

```

```

--CREATE_MASTER_REPOBJECT という API で、マスター・グループに
-- オブジェクトを追加します。ほとんどの場合はマスター・グループに表を
-- 追加することになりますが、索引、プロシージャ、ビュー、シノニムなども
-- 追加できます。詳細は、8-91 ページの
-- 「CREATE_MASTER_REPOBJECT プロシージャ」を参照してください。

```

```

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        GNAME => 'scott_mg',

```

```
TYPE => 'table',
ONAME => 'emp',
SNAME => 'scott',
USE_EXISTING_OBJECT => TRUE,
COPY_ROWS => TRUE);

END;
/

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    GNAME => 'scott_mg',
    TYPE => 'table',
    ONAME => 'dept',
    SNAME => 'scott',
    USE_EXISTING_OBJECT => TRUE,
    COPY_ROWS => TRUE);

END;
/

--EMP 表および DEPT 表には主キーがありますが、BONUS 表および
--SALGRADE 表には主キーがありません。レプリケーションが正常に行われる
-- ためには、レプリケート表に主キーまたは " 設定列 " のいずれかが必要です。
-- マルチマスター・レプリケーションのみをサポートする場合は
--DBMS_REPCAT.SET_COLUMNS プロシージャのみで十分ですが、高速の
-- リフレッシュ可能なスナップショットもサポートする予定ならば、主キーが必要です。
-- マスター・グループにオブジェクトを追加する前に、オブジェクトを変更して
-- おくほうが簡単です。

ALTER TABLE scott.bonus ADD (CONSTRAINT bonus_pk PRIMARY KEY(ename));

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    GNAME => 'scott_mg',
    TYPE => 'table',
    ONAME => 'bonus',
    SNAME => 'scott',
    USE_EXISTING_OBJECT => TRUE,
    COPY_ROWS => TRUE);

END;
/

-- 前のステップで SCOTT.BONUS オブジェクトを変更した場合とまったく同じように、
--SCOTT.SALGRADE オブジェクトを変更する必要があります。

ALTER TABLE scott.salgrade ADD (CONSTRAINT salgrade_pk PRIMARY KEY(grade));

BEGIN
```

```

DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    GNAME => 'scott_mg',
    TYPE => 'table',
    ONAME => 'salgrade',
    SNAME => 'scott',
    USE_EXISTING_OBJECT => TRUE,
    COPY_ROWS => TRUE);
END;
/

/*****
ステップ 4:
マスター・サイトの追加
*****/

-- これから使用するマスター・グループをマスター定義サイトで定義すると、
-- 同じレプリケーション環境の一部となる他のサイトを定義できます。
-- (特に指定しないかぎり、マスター・グループが作成されたサイトが、
-- マスター定義サイトになります。) このレプリケート環境には、これから
-- ORC2.WORLD および ORC3.WORLD サイトが追加されます。

BEGIN
    DBMS_REPCAT.ADD_MASTER_DATABASE (
        GNAME => 'scott_mg',
        MASTER => 'orc2.world',
        USE_EXISTING_OBJECTS => TRUE,
        COPY_ROWS => TRUE,
        PROPAGATION_MODE => 'ASYNCHRONOUS');
END;
/

/*****
注意: DBA_REPSITES ビューに ORC2.WORLD が表示されてから、次に進んでください。
別の SQL*Plus のセッションで、次の SELECT 文を実行して確認します。
*****/

SELECT * FROM dba_repsites WHERE gname = 'scott_mg';
*****/

PAUSE Press <RETURN> to continue.

BEGIN
    DBMS_REPCAT.ADD_MASTER_DATABASE (
        GNAME => 'scott_mg',
        MASTER => 'orc3.world',
        USE_EXISTING_OBJECTS => TRUE,
        COPY_ROWS => TRUE,
        PROPAGATION_MODE => 'ASYNCHRONOUS');
END;
/

```

```
/******
```

**注意:** DBA\_REPSITES ビューに ORC2.WORLD が表示してから、次に進んでください。別の SQL\*Plus のセッションで、次の SELECT 文を実行して確認します。

```
SELECT * FROM dba_repsites WHERE gname = 'scott_mg';
*****/
PAUSE Press <RETURN> to continue.
```

```
/******
```

**注意:** グループの作成中に 1 つ以上の表をマスター・グループに追加した場合は、レプリケーション・アクティビティをすぐに再開しないでください。レプリケーション競合の可能性を考慮して、グループのレプリケート表に対する競合解消を設定してください。マスター・グループ・オブジェクトの競合解消の設定の詳細は、[第 6 章「競合解消」](#)を参照してください。

```
*****/
```

```
/******
```

ステップ 5:  
レプリケーション・サポートの生成

```
*****/
```

```
BEGIN
  DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
    SNAME => 'scott',
    ONAME => 'emp',
    TYPE => 'TABLE',
    MIN_COMMUNICATION => TRUE);
END;
/
```

```
BEGIN
  DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
    SNAME => 'scott',
    ONAME => 'dept',
    TYPE => 'TABLE',
    MIN_COMMUNICATION => TRUE);
END;
/
```

```
BEGIN
  DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
    SNAME => 'scott',
    ONAME => 'bonus',
    TYPE => 'TABLE',
    MIN_COMMUNICATION => TRUE);
END;
/
```

```

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'salgrade',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/

/*****
注意: DBA_REPCATLOG ビューが空になってから、マスター・グループの
アクティビティを再開してください。
次の SELECT 文によって DBA_REPCATLOG ビューの内容を確認できます。

SELECT * FROM dba_repcatlog WHERE gname = 'scott_mg';
*****/
PAUSE Press <RETURN> to continue.

/*****
ステップ 6:
レプリケーションの再開
*****/

-- マスター・グループの作成、レプリケーション・オブジェクトの追加、
-- レプリケーション・サポートの生成およびマスター・データベースの追加を
-- すべて終了した後に、レプリケーション・アクティビティを再開する必要があります。
-- RESUME MASTER ACTIVITY プロシージャで、指定されたマスター・グループ
-- に対するレプリケーションをオンにします。

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

```



---

## 配置テンプレートの作成

この章では、レプリケーション・マネージメント API を使用して、配置テンプレートを作成する方法を説明します。次に、項目を示します。

- [Oracle 配置テンプレート](#)の概念
- 配置テンプレートの作成
- インスタンス化のためのパッケージ化
- 配置テンプレートのインスタンス化

---

### 配置テンプレートをインスタンス化する前に

- 配置テンプレートで作成したスナップショットの高速リフレッシュを実行する場合は、そのスナップショットのマスター表にスナップショット・ログがあることを確認します。スナップショットの副問合せで指定されたマスター表に対する、スナップショット・ログも作成する必要があります。次の例は、SCOTT.EMP マスター表に対するスナップショット・ログを作成する方法を示します。

```
CREATE SNAPSHOT LOG ON scott.emp;
```

- 作成したスナップショット・サイトでのアクティビティによって、マスター・サイトで競合が発生する可能性がある場合は、配置テンプレートをインスタンス化する前に、スナップショットのマスター表に対して競合解消を設定します。競合解消の設定の詳細は、[第 6 章「競合解消」](#)を参照してください。
-

## Oracle 配置テンプレートの概念

Oracle で提供される配置テンプレートを使用すると、データベース管理者はスナップショット環境をパッケージ化して、その配布と導入を簡単かつ安全に行ったり、カスタマイズしたりできます。配置テンプレートは、1つのスナップショットと固定データ・セットのみで単純に構成することも、数百のスナップショットと1つ以上の変数に基づく動的データ・セットで複雑に構成することもできます。この目的は、環境を1回定義して、必要に応じてその配置テンプレートを配置することです。Oracle 配置テンプレートの特徴を次に示します。

- 集中管理
- スナップショット環境を繰り返し配置可能
- テンプレート・パラメータを使用したリモート・サイトでのデータのサブセット化
- テンプレートのインスタンス化とデータ・アクセスを制御する許可されたユーザーのリスト

スナップショット環境を適用するには、データベース管理者がマスター・サイトに配置テンプレートを作成します。このテンプレートには、スナップショット環境を適用するために必要なすべての情報を格納します。たとえば、リモート・サイトとターゲット・リフレッシュ・グループでオブジェクトを作成するための DDL などを格納します。また、このテンプレートには、ユーザーのセキュリティ情報やカスタム・スナップショットを作成するためのテンプレート・パラメータへのリンクも含まれています。

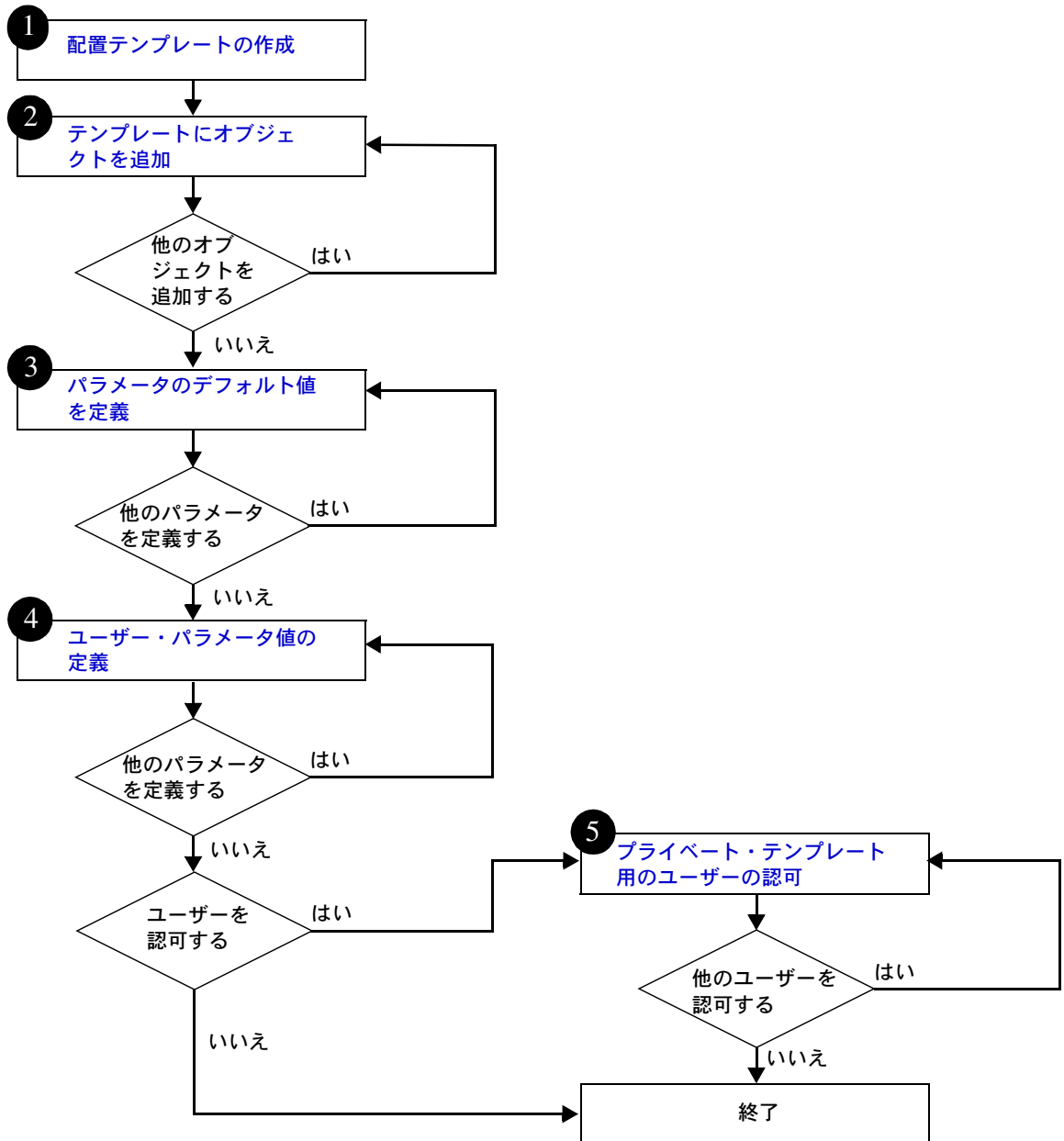
## 配置テンプレートの作成

この項では、レプリケーション・マネージメント API を使用して配置テンプレートを作成する、完全なスクリプト例を示します。

**関連項目：** 配置テンプレートの概要と構成は、『Oracle8i レプリケーション・ガイド』の第4章「配置テンプレートの概念およびアーキテクチャ」を参照してください。



図 4-1 配置テンプレートの作成



スクリプト内のコメントにも、必ず目を通してください。コメントには、レプリケーション・マネージメント API で配置テンプレートを作成するための重要で役立つ情報が記述されています。

---

---

**注意：** レプリケーション・マネージメント API を使用した場合、垂直パーティション化は実行できません。詳細は、『Oracle8i レプリケーション・ガイド』の「垂直パーティション化」を参照してください。

---

---

-- このスクリプトで、4 個のテンプレート・オブジェクト、2 個のテンプレート・パラメータ、  
-- ユーザー・パラメータ値のセット、および認可ユーザーが  
-- 含まれるプライベート配置テンプレートが作成されます。テンプレートは、次の順序で  
-- 作成されます。

-- ステップ 1: リフレッシュ・グループのテンプレートの定義  
-- ステップ 2: DT\_PERSONNEL へのテンプレート・オブジェクトの追加  
-- ステップ 3: パラメータのデフォルトおよびプロンプト・テキストの定義  
-- ステップ 4: ユーザー・パラメータ値の定義  
-- ステップ 5: プライベート・テンプレートのためのユーザーの認可

CONNECT repadmin/repadmin@orc3.world

/\*\*\*\*\*  
ステップ 1:  
配置テンプレートの作成  
\*\*\*\*\*/

-- 配置テンプレートのコンポーネントを組み立てる前に、  
-- CREATE\_REFRESH\_TEMPLATE プロシージャを使用して、Public または  
-- Private のステータス指定、ターゲット・リフレッシュ・グループ、所有者などの  
-- テンプレートの特性と一緒に、配置テンプレートの名前を定義します。

```
DECLARE
  a NUMBER;
BEGIN
  a := DBMS_REPCAT_RGT.CREATE_REFRESH_TEMPLATE (
    OWNER => 'scott',
    REFRESH_GROUP_NAME => 'personnel',
    REFRESH_TEMPLATE_NAME => 'dt_personnel',
    TEMPLATE_COMMENT => 'personnel deployment template',
    PUBLIC_TEMPLATE => 'N');
END;
/
```

```

/*****

```

```

ステップ 2:

```

```

テンプレートにオブジェクトを追加

```

```

*****/

```

```

-- ステップ 2a: EMP スナップショットの作成

```

```

-- 次のプロシージャでは、DBMS_LOB パッケージが使用されていることに
-- 注意してください。このパッケージでは、CREATE_TEMPLATE_OBJECT ファンクションの
--DDL_TEXT パラメータに値を挿入する必要があります。
-- 値のデータ型は CLOB です。CLOB 型パラメータに値を挿入することが必要な
-- 場合は、DBMS_LOB パッケージが常に使用されます。DBMS_LOB パッケージの
-- 使用方法の詳細と LOB 全般については、『Oracle8i アプリケーション開発者ガイド 基礎編』
-- を参照してください。

```

```

DECLARE
    tempstring VARCHAR2(300);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);
    tempstring := 'CREATE SNAPSHOT scott.snap_emp AS SELECT
        empno, ename, job, mgr, hiredate, sal, comm, deptno
    FROM scott.emp@:dblink WHERE deptno = :dept';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
        REFRESH_TEMPLATE_NAME => 'dt_personnel',
        OBJECT_NAME => 'snap_emp',
        OBJECT_TYPE => 'SNAPSHOT',
        DDL_TEXT => templob,
        master_rollback_seg => 'RBS');
    DBMS_LOB.FREETEMPORARY(templob);
END;
/

```

```

-- スナップショットの作成時には常に、そのスナップショットに対する問合せで表所有者のスキーマ名を
-- 指定します。前述の例では、SCOTT が EMP 表の所有者として指定されています。

```

```

-- ステップ 2b: DEPT スナップショットの作成

```

```

DECLARE
    tempstring VARCHAR2(300);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);

```

```
tempstring := 'CREATE SNAPSHOT scott.snap_dept AS SELECT
    deptno, dname, loc
    FROM scott.dept@dblink';
DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
    REFRESH_TEMPLATE_NAME => 'dt_personnel',
    OBJECT_NAME => 'snap_dept',
    OBJECT_TYPE => 'SNAPSHOT',
    DDL_TEXT => templob,
    MASTER_ROLLBACK_SEG => 'RBS');
DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

-- ステップ 2c: SALGRADE スナップショットの作成

```
DECLARE
    tempstring VARCHAR2(300);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);
    tempstring := 'CREATE SNAPSHOT scott.snap_salgrade AS SELECT
        grade, losal, hisal
        FROM scott.salgrade@dblink';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
        REFRESH_TEMPLATE_NAME => 'dt_personnel',
        OBJECT_NAME => 'snap_salgrade',
        OBJECT_TYPE => 'SNAPSHOT',
        DDL_TEXT => templob,
        MASTER_ROLLBACK_SEG => 'RBS');
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

-- ステップ 2d: BONUS スナップショットの作成

```
DECLARE
    tempstring VARCHAR2(300);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);
    tempstring := 'CREATE SNAPSHOT scott.snap_bonus AS SELECT
        ename, job, sal, comm
        FROM scott.bonus@dblink';
```

```

DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
    REFRESH_TEMPLATE_NAME => 'dt_personnel',
    OBJECT_NAME => 'snap_bonus',
    OBJECT_TYPE => 'SNAPSHOT',
    DDL_TEXT => templob,
    MASTER_ROLLBACK_SEG => 'RBS');
DBMS_LOB.FREETEMPORARY(templob);
END;
/

/*****
ステップ 3:
パラメータのデフォルト値を定義
*****/

-- テンプレートのパラメータ値とプロンプトの文字列を定義する
-- ためには、他のステップの説明にある CREATE ファンクションと
-- プロシージャではなく、ALTER_TEMPLATE_PARM プロシージャを
-- 使用します。実際のパラメータはステップ 2 で作成されているので、
-- ALTER プロシージャを使用します。ステップ 2 では、DDL_TEXT パラメータに
-- :DBLINK および :DEPT パラメータを定義しました。Oracle では DDL 内で
-- このようなパラメータが検出され、テンプレートのパラメータが自動的に
-- 作成されます。残りのテンプレート・パラメータ情報（デフォルトのパラメータ値
-- およびプロンプト文字列）を定義するために、
-- ALTER_TEMPLATE_PARM プロシージャを使用します。

-- ステップ 3a: DEPT パラメータ

DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);
    tempstring := '20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM (
        REFRESH_TEMPLATE_NAME => 'dt_personnel',
        PARAMETER_NAME => 'dept',
        NEW_DEFAULT_PARM_VALUE => templob,
        NEW_PROMPT_STRING => 'Enter your department number:',
        NEW_USER_OVERRIDE => 'Y');
    DBMS_LOB.FREETEMPORARY(templob);
END;
/

-- ステップ 3b: DBLINK パラメータ

```

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);
    tempstring := 'ORC2.WORLD';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM (
        REFRESH_TEMPLATE_NAME => 'dt_personnel',
        PARAMETER_NAME => 'dblink',
        NEW_DEFAULT_PARM_VALUE => templob,
        NEW_PROMPT_STRING => 'Enter target database link:',
        NEW_USER_OVERRIDE => 'N');
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

```
/*****
```

ステップ 4:

ユーザー・パラメータ値の定義

```
*****/
```

```
-- 個々のリモート・スナップショット・サイトでカスタム・データ・セットの
-- インスタンスエーションを自動化するために、指定されたユーザーが
-- ターゲット・テンプレートをインスタンスエートするときに自動的に使用される
-- USER_PARAMETER 値を定義できます。
-- CREATE_USER_PARM_VALUE によって、特定のユーザーに
-- パラメータ値を割り当てることができます。
```

```
-- ステップ 4a: ユーザー SCOTT 用のユーザー・パラメータ値の定義
```

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);
    tempstring := '30';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE (
        REFRESH_TEMPLATE_NAME => 'dt_personnel',
        PARAMETER_NAME => 'dept',
        USER_NAME => 'scott',
        PARM_VALUE => templob);
    DBMS_LOB.FREETEMPORARY(templob);
```

```

END;
/
-- ステップ 4b: ユーザー SCOTT 用のユーザー・パラメータ値の定義

DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, dbms_lob.session);
    tempstring := 'ORC2.WORLD';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE (
        REFRESH_TEMPLATE_NAME => 'dt_personnel',
        PARAMETER_NAME => 'dblink',
        USER_NAME => 'scott',
        PARM_VALUE => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/

/*****
ステップ 5:
プライベート・テンプレート用のユーザーの認可
*****/

-- ステップ 1 で定義した DBMS_REPCAT_RGT.CREATE_REFRESH_TEMPLATE ファンクションに
-- PUBLIC_TEMPLATE => 'N' が指定されていることからわかるように、
-- これはプライベート・テンプレートなので、DT_PERSONNEL 実行テンプレートを
-- インスタンスエートするにはユーザーを認可する必要があります。
-- DBMS_REPCAT_RGT.CREATE_USER_AUTHORIZATION ファンクションを
-- 使用して、認可ユーザーを作成します。

DECLARE
    a NUMBER;
BEGIN
    a := DBMS_REPCAT_RGT.CREATE_USER_AUTHORIZATION (
        USER_NAME => 'scott',
        REFRESH_TEMPLATE_NAME => 'dt_personnel');
END;
/

COMMIT;

```

## インスタンスーションのためのパッケージ化

配置テンプレートを作成した後、テンプレートをパッケージ化してインスタンスーションを行います。ここでは、オンラインとオフラインのインスタンスーション・プロシージャの使用例を示します。オンラインとオフラインのインスタンスーション・プロシージャは非常に類似しており、必要に応じて `INSTANTIATE_ONLINE` ファンクションと `INSTANTIATE_OFFLINE` ファンクションのいずれかを使用します。ここでは、2つの作業を行います。まず、インスタンスーション・スクリプトを作成して、次にそのスクリプトをファイルに保存します。

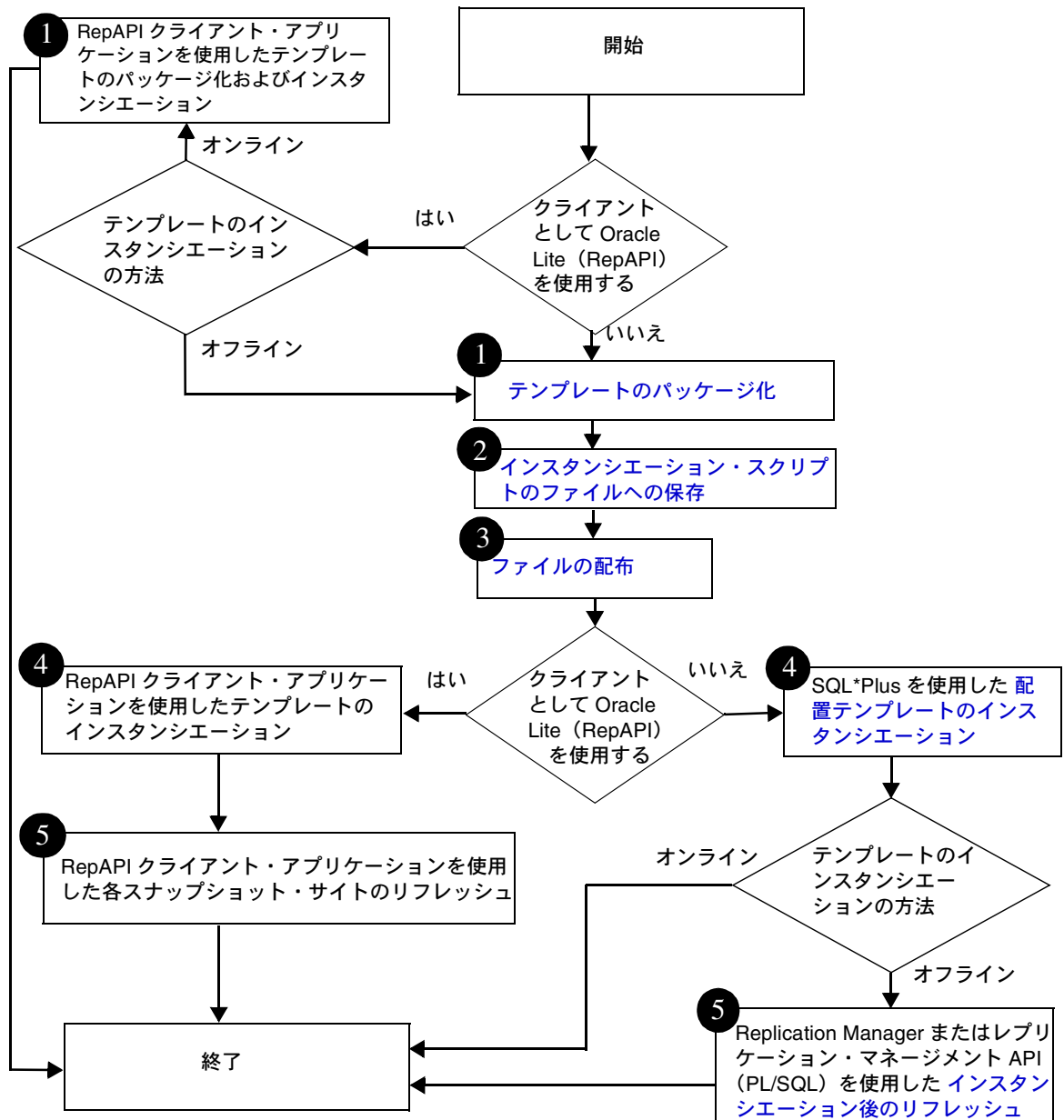
---

**関連項目：** 図 4-2 の作業の一部では、RepAPI クライアント・アプリケーションを使用して Oracle8i Lite スナップショット・サイト用の配置テンプレートのパッケージ化やインスタンスーションを行う必要があります。これらの作業を実行するには、Oracle8i Lite で提供されている Oracle Client Replication Tool などのクライアント・アプリケーションを使用します。詳細は、Oracle8i Lite のドキュメントを参照してください。Java RepAPI クライアントをサポートするサーバー構成の詳細は、『Oracle8i レプリケーション・ガイド』の付録 C「RepAPI を使用する場合の Oracle8i Server の構成」を参照してください。

---



図 4-2 配置テンプレートのパッケージ化およびインスタンスエーション



## テンプレートのパッケージ化

INSTANTIATE\_OFFLINE ファンクションまたは INSTANTIATE\_ONLINE ファンクションを実行すると、Oracle では USER\_REPCAT\_TEMP\_OUTPUT ビューにスクリプトが移入されて、リモート・スナップショット環境が構築されます。オンラインとオフラインのスクリプトのいずれにも、配置テンプレートで指定されたオブジェクトを作成するための SQL 文が含まれます。ただし、オフライン・インスタンスーション・スクリプトには、オブジェクトに移入するデータが含まれます。オンライン・インスタンスーション・スクリプトには、このデータは含まれません。オンライン・インスタンスーションでは、スナップショット・サイトがマスター・サイトに接続して、このデータがダウンロードされます。

必要に応じて、「[オフライン・インスタンスーションによるパッケージ化](#)」または「[オンライン・インスタンスーションによるパッケージ化](#)」に示すスクリプトを実行してください。これらの項の説明は、Oracle8i Enterprise Edition、Oracle8i Standard Edition または Oracle8i Personal Edition がインストールされたスナップショット・サイトのテンプレートのパッケージ化にのみ適用されます。Oracle8i Lite がインストールされたスナップショット・サイトには適用されません。

**関連項目：** Oracle8i Lite がインストールされたスナップショット・サイトでオフライン・インスタンスーション用に配置テンプレートをパッケージ化する方法は、『Oracle8i レプリケーション・ガイド』の付録 C 「RepAPI を使用する場合の Oracle8i Server の構成」を参照してください。Oracle Client Replication Tool などのクライアント・アプリケーションを使用して、オンラインでの配置テンプレートのパッケージ化およびインスタンスーションを行う方法は、Oracle8i Lite のドキュメントを参照してください。

## オフライン・インスタンスーションによるパッケージ化

INSTANTIATE\_OFFLINE ファンクションは、指定された配置テンプレートの内容に従って、スナップショット環境を構築するスクリプトを生成します。このスクリプトには、スナップショット環境を構築するための DDL (CREATE 文) の他に、スナップショット環境に適切なデータ・セットを移入するための DML (INSERT 文) が含まれています。

---

**注意：** 配置テンプレートをパッケージ化するマスター・サイトに、配置テンプレート用のターゲット・マスター・オブジェクトが含まれている場合は、ループバック・データベース・リンクを作成する必要があります。

---

```
-- リモート・スナップショット・サイトによるオフライン・インスタンスーション用テンプレートを
-- パッケージ化するために、INSTANTIATE_OFFLINE ファンクションを
-- 使用します。このプロシージャを実行すると、
-- スナップショット環境が作成されると同時に、その環境に適切な
-- データ・セットが移入されます。このスクリプトは、一時的な
-- USER_REPCAT_TEMP_OUTPUT ビューに格納されます。
```

```

SET SERVEROUTPUT ON
DECLARE
    dt_num NUMBER;
BEGIN
    dt_num := DBMS_REPCAT_RGT.INSTANTIATE_OFFLINE(
        REFRESH_TEMPLATE_NAME => 'dt_personnel',
        USER_NAME => 'scott',
        SITE_NAME => 'la_regional',
        NEXT_DATE => SYSDATE,
        INTERVAL => 'SYSDATE + (1/144)');
    DBMS_OUTPUT.PUT_LINE('Template ID = ' || dt_num);
END;
/
COMMIT;
/

```

DT\_NUM 変数に返される値を記録しておきます。この値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから選択して、生成されたスクリプトを取り出すときに使用する必要があります。ここに示したスクリプトを実行した後、「[インスタンスエーション・スクリプトのファイルへの保存](#)」に示すステップに従ってください。このスクリプトは個々のスナップショット・サイトに固有のもので、他のスナップショット・サイトでは使用できません。

## オンライン・インスタンスエーションによるパッケージ化

INSTANTIATE\_ONLINE ファンクションは、指定された配置テンプレートの内容に従って、スナップショット環境を構築するスクリプトを生成します。このスクリプトがリモート・スナップショット・サイトで実行されると、スクリプト内の DDL (CREATE 文) に従ってスナップショット・サイトが作成され、マスター・サイトから適切なデータ・セットが移入されます。この処理では、リモート・スナップショット・サイトが、マスター・サイトにオンラインで接続されている必要があります。

**関連項目：** スナップショット・サイトのこの他の必要事項は、『Oracle8i レプリケーション・ガイド』の第7章「レプリケーション環境の計画」を参照してください。

```

-- リモート・スナップショット・サイトによるオンライン・インスタンスエーション用テンプレートを
-- パッケージ化するために、INSTANTIATE_ONLINE ファンクションを
-- 使用します。このプロシージャを実行すると、
-- スナップショット環境が作成されます。このスクリプトは、一時的な
--USER_REPCAT_TEMP_OUTPUT ビューに格納されます。

```

```

SET SERVEROUTPUT ON
DECLARE
    dt_num NUMBER;
BEGIN
    dt_num := DBMS_REPCAT_RGT.INSTANTIATE_ONLINE(
        REFRESH_TEMPLATE_NAME => 'dt_personnel',

```

```
        USER_NAME => 'scott',
        SITE_NAME => 'snap1.world',
        NEXT_DATE => SYSDATE,
        INTERVAL => 'SYSDATE + (1/144)');
DBMS_OUTPUT.PUT_LINE('Template ID = ' || dt_num);
END;
/
COMMIT;
/
```

DT\_NUM 変数に戻される値を記録しておきます。この値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから選択して、生成されたスクリプトを取り出すときに使用する必要があります。ここに示したスクリプトを実行した後、「[インスタンスーション・スクリプトのファイルへの保存](#)」に示すステップに従ってください。

## インスタンスーション・スクリプトのファイルへの保存

USER\_REPCAT\_TEMP\_OUTPUT ビューの内容を保存する最も簡単な方法は、SQL\*Plus のスプール機能を使用して、SELECT 文の結果を保存することです。次に示すステップに従って、配置テンプレート・スクリプトをファイルに保存します。

---

---

**注意：** 次に示すステップは、INSTANTIATE\_OFFLINE ファンクションまたは INSTANTIATE\_ONLINE ファンクションをコールした直後に行う必要があります。USER\_REPCAT\_TEMP\_OUTPUT ビューの内容は、一時的なものだからです。4-12 ページの「[テンプレートのパッケージ化](#)」に示したスクリプトを実行していない場合は、それを実行してから、次に示すステップを行ってください。

---

---

1. SPOOL *filename.sql* と入力します。 *filename* は作成するスクリプト名です。複数のインスタンスーション・ファイルを生成する必要がある場合もあるので、ファイルにはわかりやすい名前を付けます。たとえば、次のように入力します。

```
SQL> SPOOL d:\snap1_world.sql
```

インスタンスーション・スクリプトは、特に指定がない限り、前述の例のように snap1\_world.sql という名前で、Oracle ホーム・ディレクトリに保存されます。別のディレクトリに保存する場合は、フルパスでファイル名を指定します。

2. 次の SELECT 文を実行します。

```
SQL> SELECT DBMS_REPCAT_RGT.VC2_FROM_CLOB(text) text
        FROM user_repcat_temp_output
        WHERE output_id = dt_num ORDER BY LINE;
```

*dt\_num* は、INSTANTIATE\_ONLINE ファンクションまたは INSTANTIATE\_OFFLINE ファンクション（4-12 ページの「[テンプレートのパッケージ化](#)」を参照してください）を実行したときに戻された値です。

3. スプールを停止するには次のように入力します。

```
SQL> SPOOL OFF
```

ステップ 1 で指定したファイルが、指定したディレクトリに保存されます。このファイルには、スナップショット環境を構築するためのスクリプトが記述されています。

## ファイルの配布

インスタンスエーション・スクリプトを作成して、それをファイルに保存した後、テンプレートをインスタンスエートするリモート・スナップショット・サイトにこのファイルを配布する必要があります。ファイルを配布するには、FTP サイトにファイルを送信するか、CD-ROM、フロッピー・ディスクなどに保存します。

## 配置テンプレートのインスタンスエーション

インスタンスエーション・スクリプトがリモート・スナップショット・サイトに配布された後、そこで配置テンプレートをインスタンスエートできます。

**関連項目：** 配置テンプレートをインスタンスエートするために、スナップショット・サイトで満たされる必要がある要件は、『Oracle8i レプリケーション・ガイド』の第7章「レプリケーション環境の計画」を参照してください。

次のスクリプトは、Oracle8i Enterprise Edition、Oracle8i Standard Edition または Oracle8i Personal Edition がインストールされたリモート・スナップショット・サイトでインスタンスエーション・プロセスを実行する方法を示しています。Oracle8i Lite がインストールされたスナップショット・サイトには適用されません。Oracle8i Lite スナップショット・サイトでは、Oracle Client Replication Tool などのクライアント・アプリケーションを使用して配置テンプレートをインスタンスエートします。

**関連項目：** Oracle8i Lite スナップショット・サイトでの配置テンプレートのインスタンスエーションの詳細は、Oracle8i Lite のドキュメントを参照してください。

```
/*****
```

```
ステップ 1:
```

```
スキーマとデータベース・リンクの作成
```

```
*****/
```

```
-- リモート・スナップショット・サイトでインスタンスエーション・スクリプトを実行する前に、  
-- レプリケート・オブジェクトが含まれるスキーマを作成する必要があります。  
--SQL*Plus を使用して、ステップ 1 およびステップ 2 を実行します。
```

```
CONNECT system/manager@snap1.world
```

```
CREATE USER scott IDENTIFIED BY tiger;
```

```
GRANT CONNECT, RESOURCE TO scott;
```

```
-- プライベート・データベース・リンクを作成する前に、  
-- パブリック・データベース・リンクを作成しておく必要があります。
```

```
CREATE PUBLIC DATABASE LINK orc3.world USING 'orc3.world';
```

```
-- ターゲット・ユーザーの scott として接続し、ターゲット・マスター・サイトへの  
-- プライベート・データベース・リンクを作成します。ターゲットのユーザーも  
-- マスター・サイトに存在していることが必要です。
```

```
CONNECT scott/tiger@snap1.world
```

```
CREATE DATABASE LINK orc3.world  
CONNECT TO scott IDENTIFIED BY tiger;
```

```
/******  
ステップ 2:  
インスタンス化・スクリプトの実行  
*****/  
  
@d:\snapl_world.sql
```

構築するスナップショット環境の規模やロードされるデータ量によっては、インスタンス化・プロセスにかなり時間がかかる場合があります。

## インスタンス化後のリフレッシュ

オフラインで配置テンプレートをインスタンス化した場合は、できるだけ早い時点でリフレッシュを実行する必要があります。次の EXECUTE 文を実行します。

```
EXECUTE DBMS_REFRESH.REFRESH('personnel');
```





---

## スナップショット・グループの作成

この章では、リモート・スナップショット・レプリケーション・サイトで、スナップショット・グループを作成する方法を説明します。次に、項目を示します。

- [スナップショット・グループ作成の概要](#)
- [スナップショット・グループの作成](#)

---

**注意：** 作成したスナップショット・サイトでのアクティビティによってマスター・サイトで競合が発生する可能性がある場合は、マスター表に対して競合解消を設定します。競合解消の設定の詳細は、[第6章「競合解消」](#)を参照してください。

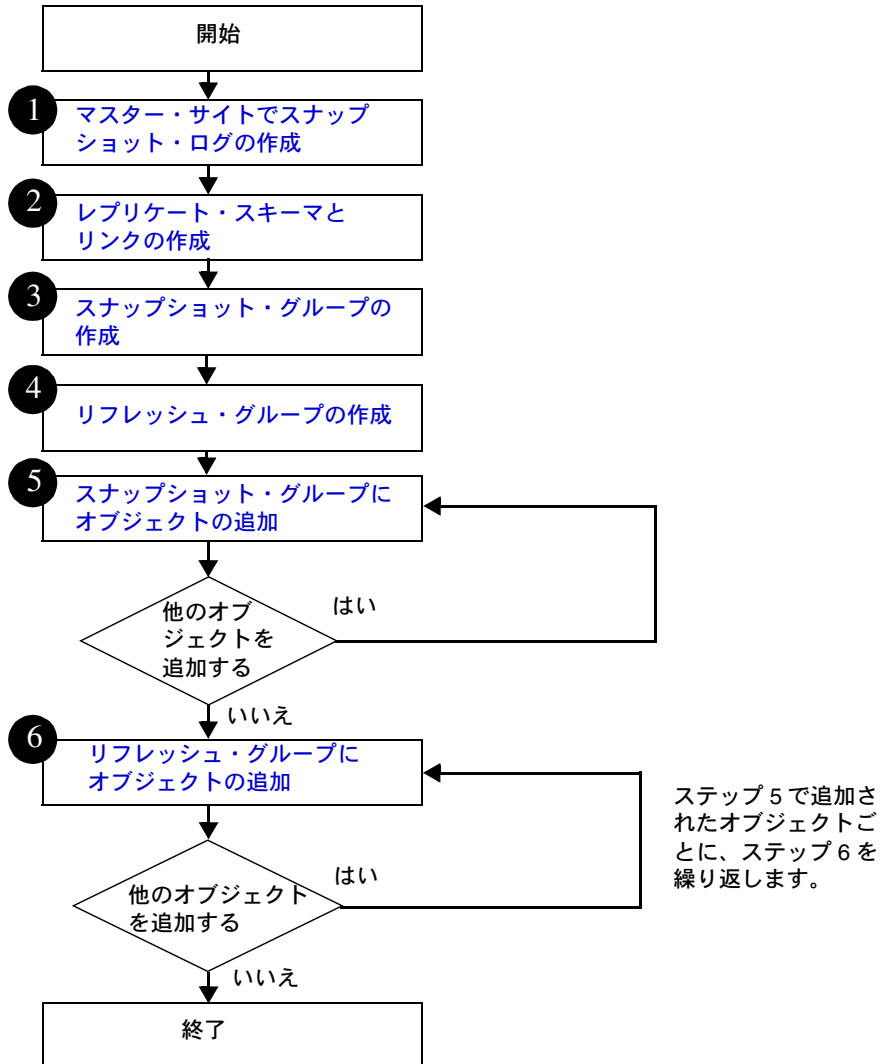
---

## スナップショット・グループ作成の概要

スナップショット・サイトを設定し、マスター・グループを最低 1 つ作成した後、リモート・スナップショット・サイトでスナップショット・グループを作成できます。[図 5-1](#) は、スナップショット・グループを作成するためのプロセスを示します。

**関連項目：** スナップショット・サイトの設定の詳細は、[第 2 章「レプリケーション・サイトの構築」](#)を参照してください。マスター・グループの作成の詳細は、[第 3 章「マスター・グループの作成」](#)を参照してください。

図 5-1 スナップショット・グループの設定



## スナップショット・グループの作成

```
/*****
```

```
ステップ 1:
```

```
マスター・サイトでスナップショット・ログの作成
```

この SQL 文の詳細は、『Oracle8i SQL リファレンス』の

「CREATE SNAPSHOT LOG」を参照してください。

```
*****/
```

```
-- マスター・サイトの 1 つでスナップショット・サイトがサポートされるようにする場合は、
-- スナップショットにレプリケートされる各マスター表に対し、スナップショット・ログを
-- 作成する必要があります。2-2 ページの図 2-1 にあるように、ORCL.WORLD は
-- ORCL.WORLD スナップショット・サイトのターゲット・マスター・サイトになります。
-- 必要なスナップショット・ログは ORCL.WORLD で作成される必要があります。
```

```
CONNECT scott/tiger@orcl.world
```

```
CREATE SNAPSHOT LOG ON scott.emp;
CREATE SNAPSHOT LOG ON scott.dept;
CREATE SNAPSHOT LOG ON scott.bonus;
CREATE SNAPSHOT LOG ON scott.salgrade;
```

```
/*****
```

```
ステップ 2:
```

```
レプリケート・スキーマとリンクの作成
```

```
*****/
```

```
-- スナップショット・グループの作成を始める前に、リモート・スナップショット・サイトに
-- レプリケート・スキーマが存在し、しかも必要なデータベース・リンクが
-- 作成されていることを確認してください。
```

```
CONNECT system/manager@snap1.world
```

```
CREATE USER scott IDENTIFIED BY tiger;
GRANT connect, resource TO scott;
```

```
CONNECT scott/tiger@snap1.world
```

```
-- スナップショットの所有者には、スナップショット・サイトが設定されたときに
-- 作成された、proxy_refresher をポイントするデータベース・リンクが必要です。
-- 詳細は、2-6 ページの「マスター・サイトのユーザーの作成」を参照してください。
```

```
CREATE DATABASE LINK orcl.world
  CONNECT TO proxy_refresher IDENTIFIED BY proxy_refresher;
```

```

/*****
ステップ 3:
スナップショット・グループの作成
*****/

-- 次のプロシージャは、リモート・スナップショット・サイトで、スナップショット管理者
-- のみが実行できます。

CONNECT snapadmin/snapadmin@snap1.world

--GNAME パラメータに指定するマスター・グループは、ターゲットのマスター・サイトで
-- レプリケート中のマスター・グループの名前である必要があります。

BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP (
        GNAME => 'scott_mg',
        MASTER => 'orcl.world',
        PROPAGATION_MODE => 'ASYNCHRONOUS');
END;
/

```

## スナップショット・グループの作成

---

```

/*****
ステップ 4:
リフレッシュ・グループの作成
*****/

```

```

-- 特定のリフレッシュ・グループに追加されたすべてのスナップショットは、
-- 同時にリフレッシュされます。この機能によって、リフレッシュ・グループ内で
-- 関連性のあるスナップショット間の、トランザクション一貫性が保証されます。

```

```

BEGIN
    DBMS_REFRESH.MAKE (
        NAME => 'snapadmin.scott_rg',
        LIST => '',
        NEXT_DATE => SYSDATE,
        INTERVAL => 'SYSDATE + 1/24',
        IMPLICIT_DESTROY => FALSE,
        ROLLBACK_SEG => '',
        PUSH_DEFERRED_RPC => TRUE,
        REFRESH_AFTER_ERRORS => FALSE);
END;
/

```

```

/*****
ステップ 5:
スナップショット・グループにオブジェクトの追加
*****/

```

```

-- スナップショットの作成時は常に、そのスナップショットに対する問合せで表所有者の
-- スキーマ名を指定します。前述の例では、SCOTT が各問合せの表の所有者として指定さ
-- れています。

```

```

BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        ONAME => 'bonus',
        TYPE => 'SNAPSHOT',
        DDL_TEXT => 'CREATE SNAPSHOT scott.bonus REFRESH FAST WITH
                    PRIMARY KEY FOR UPDATE AS SELECT * FROM
                    scott.bonus@orcl.world',
        MIN_COMMUNICATION => TRUE);
END;
/

```

```

BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
        GNAME => 'scott_mg',

```

```

        SNAME => 'scott',
        ONAME => 'dept',
        TYPE => 'SNAPSHOT',
        ddl_text => 'CREATE SNAPSHOT scott.dept REFRESH FAST WITH
                    primary key for update as select * from
                    scott.dept@orc1.world',
        MIN_COMMUNICATION => 'TRUE');
END;
/

BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'SNAPSHOT',
        DDL_TEXT => 'CREATE SNAPSHOT scott.emp REFRESH FAST WITH
                    primary key for update as select * from
                    scott.emp@orc1.world',
        MIN_COMMUNICATION => 'TRUE');
END;
/

BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        ONAME => 'salgrade',
        TYPE => 'SNAPSHOT',
        DDL_TEXT => 'CREATE SNAPSHOT scott.salgrade REFRESH FAST WITH
                    primary key for update as select * from
                    scott.salgrade@orc1.world',
        MIN_COMMUNICATION => 'TRUE');
END;
/

/*****
ステップ 6:
リフレッシュ・グループにオブジェクトの追加
*****/

-- リフレッシュ・グループに追加するスナップショット・グループの各オブジェクトは、
-- 関連性があるスナップショット間での参照整合性を保つために、
-- 同時にリフレッシュされます。

BEGIN
    DBMS_REFRESH.ADD (

```

```
        NAME => 'snapadmin.scott_rg',
        LIST => 'scott.bonus',
        LAX => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        NAME => 'snapadmin.scott_rg',
        LIST => 'scott.dept',
        LAX => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        NAME => 'snapadmin.scott_rg',
        LIST => 'scott.emp',
        LAX => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        NAME => 'snapadmin.scott_rg',
        LIST => 'scott.salgrade',
        LAX => TRUE);
END;
/
```



---

## 競合解消

この章では、レプリケート環境に競合解消方法を定義する方法を説明します。次に、項目を示します。

- [競合解消の準備](#)
- [更新競合の競合解消方法の作成](#)
- [一意性競合の競合解消方法の作成](#)
- [削除競合の競合回避方法の作成](#)
- [競合解消の監査](#)

## 競合解消の準備

レプリケート環境に複数のサイトがある場合、競合が起こることがあります。競合を回避するようにデータベースとフロントエンド・アプリケーションを設計しても、競合を皆無にはできません。レプリケーションで重要な点は、レプリケート環境に関係するすべてのサイトで、確実にデータを収束させることです。

データの競合が発生した場合、競合がビジネス・ルールに従って解消され、すべてのサイトでデータを正常に収束させるためのメカニズムが必要です。

Oracle レプリケーションでは、ビジネス・ルールに従って競合を解消する、データベース用の競合解消システムを定義できます。Oracle の事前作成競合解消方法で競合を解消できない場合は、独自の競合解消方法を作成し、それを使用して競合を解消できます。

**関連項目：** 競合解消方法の概要および各方法に対するデータ収束の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

## プラン

レプリケート表に競合解消方法をインプリメントする前に、システム内のデータを分析し、競合の起こりやすい領域を確認します。たとえば、従業員番号などの静的データは、頻繁には変更されません。このため、競合はあまり起こりません。これに対して、従業員の担当顧客は頻繁に変わるので、データの競合が起こりやすくなります。

競合の起こりやすい領域を特定した後、その解消方法を決定します。たとえば、最新の変更を優先させたり、特定のサイトを優先させたりします。

この章では、各種の競合解消方法を説明します。この説明を参考にして、解消方法に最も適した方法を使用します。また、発生する可能性のある競合の解消方法についても考慮します。

問題の起こりうる領域を特定し、その問題を解決するビジネス・ルールを決定した後、Oracle の競合解消方法（またはユーザー独自の方法のいずれか）を使用して競合解消システムをインプリメントします。

## 更新競合の競合解消方法の作成

データの競合は、複数のサイトで同時に同じ行が更新された場合、または、あるサイトの遅延トランザクションが他のサイトに正しく伝播される前によく起こります。

更新競合を回避する方法の1つは、同期レプリケート環境をインプリメントすることです。ただし、この方法には、ネットワーク・リソースがかなり必要です。

もう1つの方法は、Oracle 競合解消方法を使用することです。これらの方法を使用すると、同じ行が複数の更新を受信した場合に起こる更新競合に対処できます。

## 上書きによる方法と廃棄による方法

上書きによる方法と廃棄による方法は、レプリケーション元またはレプリケーション先のサイトからの値を無視します。このため、複数のマスター・サイトでの確実なデータ収束は保証されません。これらの方法は、1つのマスター・サイトと複数のスナップショット・サイトで使用するか、あるいは、なんらかのユーザー定義の通知機能と使用することを前提としています。

上書きによる方法は、レプリケーション元の新しい値で、レプリケーション先のカレント値を置き換えます。これに対して、廃棄による方法は、レプリケーション元からの新しい値を無視します。

**関連項目：** 詳細は、8-68 ページの「[ADD\\_conflicttype\\_RESOLUTION プロシージャ](#)」および『Oracle8i レプリケーション・ガイド』の「上書きによる方法と廃棄による方法」を参照してください。

---

---

**注意：** この項では、このマニュアルの他のスクリプトでは使用されていないオブジェクトを使用しています。ORC1.WORLD、ORC2.WORLD、ORC3.WORLD および SNAP1.WORLD という構成には、3つのマスター・サイトと1つのスナップショット・サイトが含まれており、上書きによる方法と廃棄による方法には適していないためです。

---

---

-- 次のプロシージャを実行できるのは、レプリケーション管理者のみです。

```
CONNECT repadmin/repadmin@saturn.universe
```

-- 競合解消方法を定義する前に、その方法を適用する表が含まれている

-- マスター・グループを停止しておく必要があります。

```
BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'titan_mg');
END;
/
```

-- すべての Oracle 競合解消方法は、" 列グループ " という用語で表される論理的な  
 -- 列のグループに配置されます。DBMS\_REPCAT.「[MAKE\\_COLUMN\\_GROUP プロシージャ](#)」を  
 -- 使用して、ターゲットとなる表の列グループを作成します。

```
BEGIN
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        SNAME => 'titan',
        ONAME => 'planet',
        COLUMN_GROUP => 'planet_cgl',
        LIST_OF_COLUMN_NAMES => 'order,circumference,moons');
END;
/
```

-- 指定された表のための競合解消方法を定義するために、  
-- DBMS\_REPCAT.ADD\_UPDATE\_RESOLUTION という API を使用します。  
-- この例では、OVERWRITE（上書き）による競合解消方法が作成されます。

```
BEGIN
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        SNAME => 'titan',
        ONAME => 'planet',
        COLUMN_GROUP => 'planet_cg1',
        SEQUENCE_NO => 1,
        METHOD => 'OVERWRITE',
        PARAMETER_COLUMN_NAME => 'order,circumference,moons');
END;
/
```

-- 競合解消方法を定義した後、その方法を受け入れる表の  
-- レプリケーション・サポートを再生成します。

```
BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'titan',
        ONAME => 'planet',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/
```

-- レプリケーション・サポートを再生成した後に、「RESUME\_MASTER\_ACTIVITY プロシージャ」で  
-- レプリケーション・アクティビティを再開します。

```
BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'titan_mg');
END;
/
```

## 最小値による方法と最大値による方法

アドバンスド・レプリケーション機能によって列グループの競合が検出され、最小値による方法または最大値による方法がコールされると、列グループ内の該当する列で、レプリケーション元の新しい値とレプリケーション先のカレント値が比較されます。競合解消方法を定義する場合は、この列を指定します。

レプリケーション先の列の新しい値が、カレント値よりも小さい、または大きい場合（使用する方法により異なります）、レプリケーション元にある列グループの値がレプリケーショ

ン先に適用されます。その場合、その行に対しては、他のすべてのエラーが解決されていることを前提とします。そうでない場合は、行が変更されずにそのまま保持されます。

-- 次のプロシージャを実行できるのは、レプリケーション管理者のみです。

```
CONNECT repadmin/repadmin@orc1.world
```

-- 競合解消方法を定義する前に、その方法を適用する表が含まれている

-- マスター・グループを停止しておく必要があります。

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
```

```
END;
```

```
/
```

-- すべての Oracle 競合解消方法は、" 列グループ " という用語で表される論理的な

-- 列のグループに配置されます。DBMS\_REPCAT. 「MAKE\_COLUMN\_GROUP プロシージャ」を

-- 使用して、ターゲットとなる表の列グループを作成します。

```
BEGIN
```

```
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        SNAME => 'scott',
        ONAME => 'salgrade',
        COLUMN_GROUP => 'salgrade_cg1',
        LIST_OF_COLUMN_NAMES => 'losal');
```

```
END;
```

```
/
```

-- 指定された表のための競合解消方法を定義するために、

--DBMS\_REPCAT.ADD\_UPDATE\_RESOLUTION という API を使用します。

-- この例では、MINIMUM (最小値) による競合解消方法が作成されます。

```
BEGIN
```

```
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        SNAME => 'scott',
        ONAME => 'salgrade',
        COLUMN_GROUP => 'salgrade_cg1',
        SEQUENCE_NO => 1,
        METHOD => 'MINIMUM',
        PARAMETER_COLUMN_NAME => 'losal');
```

```
END;
```

```
/
```

-- 競合解消方法を定義した後、その方法を受け入れる表の

-- レプリケーション・サポートを再生成します。

```
BEGIN
```

```
DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
    SNAME => 'scott',
    ONAME => 'salgrade',
    TYPE => 'TABLE',
    MIN_COMMUNICATION => TRUE);
END;
/

-- レプリケーション・サポートを再生成した後に、「RESUME_MASTER_ACTIVITY プロシージャ」で
-- レプリケーション・アクティビティを再開します。

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
```

## タイムスタンプによる方法

最古のタイムスタンプによる方法と最新のタイムスタンプによる方法は、最小値による方法と最大値による方法に類似しています。タイムスタンプによる方法を使用するには、DATE型のレプリケート表にある列を指定する必要があります。アプリケーションで列グループのいずれかの列を更新する場合は、レプリケーション先のタイムスタンプ列の値をローカルのSYSDATE値で更新する必要があります。他のサイトから適用された変更に対しては、タイムスタンプ値をレプリケート元のタイムスタンプ値に設定します。

タイムスタンプの競合解消が正しく機能するための要素は次のとおりです。

- コンピュータ間で同期化された時刻の設定
  - タイムスタンプを自動的に記録するためのタイムスタンプ・フィールドとトリガー
- 次のプロシージャを実行できるのは、レプリケーション管理者のみです。

```
CONNECT repadmin/repadmin@orc1.world

-- 競合解消方法を定義する前に、その方法を適用する表が含まれている
-- マスター・グループを停止しておく必要があります。

BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

-- ターゲットの表にタイムスタンプ・フィールドがまだ含まれていない場合、
-- 行の挿入または更新時にタイムスタンプ値を記録できるように、
-- 表にタイムスタンプ用の列を追加します。また、該当 DDL を
```

-- ターゲットの表に適用するために、ALTER\_MASTER\_REPOBJECT API を  
 -- 使用する必要があります。(単純に DDL を発行すると、レプリケート・オブジェクトが  
 -- 無効になる場合があります。)

```
BEGIN
  DBMS_REPCAT.ALTER_MASTER_REPOBJECT (
    SNAME => 'scott',
    ONAME => 'emp',
    TYPE => 'TABLE',
    DDL_TEXT => 'ALTER TABLE scott.emp ADD (timestamp DATE)');
END;
/
```

-- 新しい列をレプリケート・オブジェクトに挿入した後に、影響を受ける  
 -- オブジェクトのためのレプリケーション・サポートを、必ず再生成して  
 -- ください。このステップは、レプリケーション・オブジェクトを変更した後、  
 -- ただちに実行する必要があります。

```
BEGIN
  DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
    SNAME => 'scott',
    ONAME => 'emp',
    TYPE => 'TABLE',
    MIN_COMMUNICATION => TRUE);
END;
/
```

-- タイムスタンプ・フィールドを作成した後に、行が挿入または更新された  
 -- 時間を記録するトリガーを作成します。記録された値は、  
 --TIMESTAMP (タイムスタンプ) 方法に基づく競合解消で使用されます。DDL を  
 -- 直接実行するかわりに、DBMS\_REPCAT.「[CREATE\\_MASTER\\_REPOBJECT プロシージャ](#)」を  
 -- 使用して、トリガーを作成し、マスター・グループに追加する  
 -- 必要があります。

```
BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    GNAME => 'scott_mg',
    TYPE => 'TRIGGER',
    ONAME => 'insert_time',
    SNAME => 'scott',
    DDL_TEXT => 'CREATE TRIGGER scott.insert_time
                BEFORE
                INSERT OR UPDATE ON scott.emp FOR EACH ROW
                BEGIN
                IF DBMS_REPUTIL.FROM_REMOTE = FALSE THEN
                  :NEW.TIMESTAMP := SYSDATE;
                END IF;
                END;
```

```
        END IF;
        END;');

END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'insert_time',
        TYPE => 'TRIGGER',
        MIN_COMMUNICATION => TRUE);
END;
/

-- すべての Oracle 競合解消方法は、" 列グループ " という用語で表される論理的な
-- 列のグループに配置されます。DBMS_REPCAT.「MAKE\_COLUMN\_GROUP プロシージャ」を
-- 使用して、ターゲットとなる表の列グループを作成します。

BEGIN
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        SNAME => 'scott',
        ONAME => 'emp',
        COLUMN_GROUP => 'emp_cg1',
        LIST_OF_COLUMN_NAMES => 'mgr, hiredate, sal, timestamp');
END;
/

-- 指定された表のための競合解消方法を定義するために、
-- DBMS_REPCAT.ADD_UPDATE_RESOLUTION という API を使用します。
-- この例では、以前作成した TIMESTAMP 列が使用され、LATEST_TIMESTAMP
-- (最新タイムスタンプ) という競合解消方法が指定されます。

BEGIN
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        SNAME => 'scott',
        ONAME => 'emp',
        COLUMN_GROUP => 'emp_cg1',
        SEQUENCE_NO => 1,
        METHOD => 'LATEST_TIMESTAMP',
        PARAMETER_COLUMN_NAME => 'timestamp');
END;
/

-- 競合解消方法を定義した後、その方法を受け入れる表の
-- レプリケーション・サポートを再生成します。

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
```



```

        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/

-- 競合解消方法を定義した後、その方法を受け入れる表の
-- レプリケーション・サポートを再生成します。

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

```

## 加算による方法と平均による方法

加算による方法と平均による方法は、1つの数値列のみから構成される列グループに使用します。この競合解消方法は、一方の値のみを受け付けるのではなく、比較対象の2つの値を加算するか、またはその2つの平均を取ります。

-- 次のプロシージャを実行できるのは、レプリケーション管理者のみです。

```
CONNECT repadmin/repadmin@orc1.world
```

-- 競合解消方法を定義する前に、その方法を適用する表が含まれている  
-- マスター・グループを停止しておく必要があります。

```

BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

```

-- すべての Oracle 競合解消方法は、"列グループ" という用語で表される論理的な  
-- 列のグループに配置されます。DBMS\_REPCAT.「[MAKE\\_COLUMN\\_GROUP](#) プロシージャ」を  
-- 使用して、ターゲットとなる表の列グループを作成します。

```

BEGIN
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        SNAME => 'scott',
        ONAME => 'bonus',
        COLUMN_GROUP => 'bonus_cg1',
        LIST_OF_COLUMN_NAMES => 'sal');
END;
/

```

-- 指定された表のための競合解消方法を定義するために、  
-- DBMS\_REPCAT.ADD\_UPDATE\_RESOLUTION という API を使用します。  
-- この例では、SAL 列が使用される ADDITIVE（加算）という競合解消方法が  
-- 指定されます。

```
BEGIN
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        SNAME => 'scott',
        ONAME => 'bonus',
        COLUMN_GROUP => 'bonus_cg1',
        SEQUENCE_NO => 1,
        METHOD => 'ADDITIVE',
        PARAMETER_COLUMN_NAME => 'sal');
END;
/
```

-- 競合解消方法を定義した後、その方法を受け入れる表の  
-- レプリケーション・サポートを再生成する必要があります。

```
BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'bonus',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/
```

-- レプリケーション・サポートを再生成した後に、「RESUME\_MASTER\_ACTIVITY プロシージャ」で  
-- レプリケーション・アクティビティを再開します。

```
BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
```

## 優先グループ

優先グループでは、特定の列にある値ごとに優先レベルを指定できます。Oracle によって競合が検出されると、優先列で優先順位の低い値を持つ表が、優先順位の高い値を持つ表のデータで更新されます。

```
CONNECT repadmin/repadmin@orc1.world
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
```

```
END;
```

```
/
```

```
--JOB フィールドが列グループの一部であること、さらにその列グループのために
-- サイト優先順位に関する競合解消のメカニズムが使用されることを確認しておく
-- 必要があります。「ADD_GROUPED_COLUMN プロシージャ」を使用して、
-- このフィールドを既存の列グループに追加します。列グループをまだ作成していなければ、
--DBMS_REPCAT. 「MAKE_COLUMN_GROUP プロシージャ」で新しい列グループを
-- 作成できます。
```

```
BEGIN
```

```
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        SNAME => 'scott',
        ONAME => 'emp',
        COLUMN_GROUP => 'emp_cg1',
        LIST_OF_COLUMN_NAMES => 'mgr, hiredate, sal, job');
```

```
END;
```

```
/
```

```
-- 優先順位の値を表内の値に割り当てる前に、定義した値を保持する
-- 優先グループを作成しておく必要があります。
```

```
BEGIN
```

```
    DBMS_REPCAT.DEFINE_PRIORITY_GROUP (
        GNAME => 'scott_mg',
        PGROUP => 'job_pg',
        DATATYPE => 'VARCHAR2');
```

```
END;
```

```
/
```

```
--DBMS_REPCAT. 「ADD_PRIORITY_datatype プロシージャ」には、使用可能な
-- 異なるバージョンがいくつかあります。つまり、NUMBER、VARCHAR2 など、
-- 使用可能なデータ型ごとに 1 つのバージョンがあります。
-- 詳細は 8-65 ページの「ADD_PRIORITY_datatype プロシージャ」を
-- 参照してください。
-- 表でとり得るすべての値の優先順位を定義し終わるまで、必要に応じて何回でも
-- この API を実行してください。
```

```
BEGIN
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2 (
        GNAME => 'scott_mg',
        PGROUP => 'job_pg',
        VALUE => 'president',
        PRIORITY => 100);
END;
/

BEGIN
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2 (
        GNAME => 'scott_mg',
        PGROUP => 'job_pg',
        VALUE => 'manager',
        PRIORITY => 80);
END;
/

BEGIN
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2 (
        GNAME => 'scott_mg',
        PGROUP => 'job_pg',
        VALUE => 'salesman',
        PRIORITY => 60);
END;
/

BEGIN
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2 (
        GNAME => 'scott_mg',
        PGROUP => 'job_pg',
        VALUE => 'analyst',
        PRIORITY => 40);
END;
/

BEGIN
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2 (
        GNAME => 'scott_mg',
        PGROUP => 'job_pg',
        VALUE => 'clerk',
        PRIORITY => 20);
END;
/
```

```
-- 優先順位の値をすべて割り当てた後に、レプリケート表に
-- PRIORITY GROUP 解消方法を追加します。次の API の例で、
-- 指定された列グループの SEQUENCE_NO によって、追加された
-- 方法は 2 番目の競合解消方法であることがわかります。
```

```
BEGIN
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        SNAME => 'scott',
        ONAME => 'emp',
        COLUMN_GROUP => 'emp_cg1',
        SEQUENCE_NO => 2,
        METHOD => 'PRIORITY GROUP',
        PARAMETER_COLUMN_NAME => 'job',
        PRIORITY_GROUP => 'job_pg');
END;
/
```

```
-- 競合解消方法を定義した後、その方法を受け入れる表の
-- レプリケーション・サポートを再生成する必要があります。
```

```
BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/
```

```
-- レプリケーション・サポートを再生成した後に、「RESUME_MASTER_ACTIVITY プロシージャ」で
-- レプリケーション・アクティビティを再開します。
```

```
BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
```

## サイトの優先順位

サイトの優先順位は、優先グループを特殊化したものです。このため、サイトの優先順位に対応付けられたプロシージャは、優先グループに対応付けられたプロシージャと同じように機能します。競合は、フィールドの値の優先順位ではなく、関係するサイトの優先順位に基づいて解消されます。

たとえば、ORC2.WORLD の優先順位が ORC1.WORLD よりも高い場合に、この 2 つのサイトで競合が起これば、ORC2.WORLD の値が使用されます。

```
CONNECT repadmin/repadmin@orc1.world
```

```
BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

-- レプリケート表にサイトの値を格納するために、表に SITE 列を追加する
-- 必要があります。DDL をターゲットの表に適用するために、
-- DBMS_REPCAT. 「ALTER_MASTER_REPOBJECT プロシージャ」を使用します。
-- 単純に DDL を発行すると、レプリケート・オブジェクトが無効になる場合があります。

BEGIN
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        DDL_TEXT => 'ALTER TABLE scott.emp ADD (site VARCHAR2(20))');
END;
/

-- 新しい列をレプリケート・オブジェクトに追加した後に、影響を受ける
-- オブジェクトのためのレプリケーション・サポートを、必ず再生成して
-- ください。この作業は、レプリケーション・オブジェクトを変更した後、
-- ただちに実行する必要があります。

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/

-- SITE 列を表に追加した後に、このフィールドが列グループの一部であること、
-- さらにその列グループのためにサイト優先順位に関する競合解消のメカニズムが
-- 使用されることを確認しておく必要があります。「ADD_GROUPED_COLUMN プロシージャ」を
-- 使用して、このフィールドを既存の列グループに追加します。
-- 列グループをまだ作成していなければ、
-- DBMS_REPCAT. 「MAKE_COLUMN_GROUP プロシージャ」で新しい列グループを作成できます。

BEGIN
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        SNAME => 'scott',
        ONAME => 'emp',
        COLUMN_GROUP => 'emp_cg1',
```

```
LIST_OF_COLUMN_NAMES => 'mgr, hiredate, sal, site');
END;
/

-- サイト優先順位の値をレプリケート環境のサイトに割り当てる前に、定義した値を保持する
-- サイト優先順位グループを作成しておく必要があります。

BEGIN
    DBMS_REPCAT.DEFINE_SITE_PRIORITY (
        GNAME => 'scott_mg',
        NAME => 'site_pg');
END;
/

--DBMS_REPCAT.「ADD_SITE_PRIORITY_SITE プロシージャ」を
-- 使用して、レプリケーション環境内のサイトごとに優先順位の値を定義します。
-- レプリケーション環境内のすべてのサイトの優先順位を定義し終わるまで、
-- 必要に応じて何回でもこの API を実行します。

BEGIN
    DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (
        GNAME => 'scott_mg',
        NAME => 'site_pg',
        SITE => 'orc1.world',
        PRIORITY => 100);
END;
/

BEGIN
    DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (
        GNAME => 'scott_mg',
        NAME => 'site_pg',
        SITE => 'orc2.world',
        PRIORITY => 50);
END;
/

BEGIN
    DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (
        GNAME => 'scott_mg',
        NAME => 'site_pg',
        SITE => 'orc3.world',
        PRIORITY => 25);
END;
/
```

-- サイトに優先順位の値をすべて割り当てた後に、レプリケート表に  
-- SITE PRIORITY 解消方法を追加する必要があります。次の API の例で、  
-- 指定された列グループの SEQUENCE\_NO によって、追加された方法は  
-- 3 番目の競合解消方法であることがわかります。

```
BEGIN
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        SNAME => 'scott',
        ONAME => 'emp',
        COLUMN_GROUP => 'emp_cg1',
        SEQUENCE_NO => 3,
        METHOD => 'site priority',
        PARAMETER_COLUMN_NAME => 'site',
        PRIORITY_GROUP => 'site_pg');
END;
/
```

-- 競合解消方法を定義した後、その方法を受け入れる表の  
-- レプリケーション・サポートを再生成します。

```
BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/
```

-- レプリケーション・サポートを再生成した後に、「RESUME\_MASTER\_ACTIVITY プロシージャ」で  
-- レプリケーション・アクティビティを再開します。

```
BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
```

## 一意性競合の競合解消方法の作成

レプリケート環境では、一意性制約に関してデータの挿入を主な原因とする競合が発生する場合があります。重複する行の削除がビジネス・ルールで許可されている場合は、行の削除を Oracle の事前作成競合解消方法で定義します。

ただし、競合する値を変更して、一意性制約に違反しないようにする方が一般的です。競合する値を変更すれば、重要なデータが失われることはありません。Oracle の事前作成の一意



一意性競合解消方法を使用すると、値にサイト名や順序番号を付けることで、競合する値を一意にできます。

この他に、一意性競合解消方法の追加コンポーネントとして、通知機能があります。競合する情報は Oracle によって変更されるので、その情報を表に挿入できます。ただし、その場合、競合を通知して、その通知に基づいて競合を分析し、レコードを削除するのか、データを他のレコードにマージするのか、または競合するデータにまったく新しい値を定義するのかを決定することが必要です。

-- 次のプロシージャを実行できるのは、レプリケーション管理者のみです。

```
CONNECT repadmin/repadmin@orc1.world
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
```

```
END;
```

```
/
```

-- 一意性競合解消方法では、UNIQUE 制約がある列（複数の場合も）で

-- 発生した競合が検出され、解消されます。UNIQUE 制約を EMP 表に

-- 追加するために、「ALTER\_MASTER\_REPOBJECT プロシージャ」（8-73 ページを参照）を

-- 使用します。

```
BEGIN
```

```
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        DDL_TEXT => 'ALTER TABLE scott.emp ADD
                    (constraint emp_ename_unique UNIQUE(ename))');
```

```
END;
```

```
/
```

-- レプリケート表に UNIQUE 制約を追加した後に、影響を受ける表のための

-- レプリケーション・サポートを、必ず再生成してください。このステップは、

-- レプリケート・オブジェクトを変更した後、ただちに実行する必要があります。

```
BEGIN
```

```
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
```

```
END;
```

```
/
```

-- 次の表、conf\_report には、通知機能から受け取ったメッセージが  
-- 保存されます。

```
BEGIN
    DBMS_REPCAT.EXECUTE_DDL(
        GNAME => 'scott_mg',
        DDL_TEXT => 'CREATE TABLE scott.conf_report (
            line NUMBER(2),
            txt VARCHAR2(80),
            timestamp DATE,
            table_name VARCHAR2(30),
            table_owner VARCHAR2(30),
            conflict_type VARCHAR2(7))');
END;
/
```

```
CONNECT scott/tiger@orcl.world
```

-- 競合が検出されると、次のパッケージ notify によって、CONF\_REPORT 表に  
-- 通知が送られます。

-- このスクリプトで作成される競合解消通知のパッケージについては、  
-- [付録 B 「ユーザー定義の競合解消方法」](#) を参照してください。

```
CREATE OR REPLACE PACKAGE notify AS
    FUNCTION emp_unique_violation (ename IN OUT VARCHAR2,
        discard_new_values IN OUT BOOLEAN)
    RETURN BOOLEAN;
END notify;
/

CREATE OR REPLACE PACKAGE BODY notify AS
    TYPE message_table IS TABLE OF VARCHAR2(80) INDEX BY BINARY_INTEGER;
    PROCEDURE report_conflict(conflict_report IN MESSAGE_TABLE,
        report_length IN NUMBER,
        conflict_time IN DATE,
        conflict_table IN VARCHAR2,
        table_owner IN VARCHAR2,
        conflict_type IN VARCHAR2) IS
    BEGIN
        FOR idx IN 1..report_length LOOP
            BEGIN
                INSERT INTO scott.conf_report
                    (line, txt, timestamp, table_name, table_owner, conflict_type)
                VALUES (idx, SUBSTR(conflict_report(idx),1,80), conflict_time,
                    conflict_table, table_owner, conflict_type);
            EXCEPTION WHEN others THEN NULL;
        END LOOP;
    END;
END;
```

```

        END;
    END LOOP;
    END report_conflict;
    FUNCTION emp_unique_violation(ename IN OUT VARCHAR2,
        discard_new_values IN OUT BOOLEAN)
    RETURN BOOLEAN IS
        local_node VARCHAR2(128);
        conf_report MESSAGE_TABLE;
        conf_time DATE := SYSDATE;
    BEGIN
        BEGIN
            SELECT global_name INTO local_node FROM global_name;
        EXCEPTION WHEN others THEN local_node := '?';
        END;
        conf_report(1) := 'UNIQUENESS CONFLICT DETECTED IN TABLE EMP ON ' ||
            TO_CHAR(conf_time, 'MM-DD-YYYY HH24:MI:SS');
        conf_report(2) := ' AT NODE ' || local_node;
        conf_report(3) := 'ATTEMPTING TO RESOLVE CONFLICT USING' ||
            ' APPEND SITE NAME METHOD';
        conf_report(4) := 'ENAME: ' || ename;
        conf_report(5) := NULL;
        report_conflict(conf_report, 5, conf_time, 'EMP', 'SCOTT', 'UNIQUE');
        discard_new_values := FALSE;
        RETURN FALSE;
    END emp_unique_violation;
END notify;
/

```

```
CONNECT repadmin/repadmin@orc1.world
```

-- 次のパッケージが、レプリケーション環境にあるすべてのマスター・サイトに  
 -- レプリケートされます。このことで、通知機能がすべてのマスター・サイトで  
 -- 利用可能であることが保証されます。

```

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        GNAME => 'scott_mg',
        TYPE => 'PACKAGE',
        ONAME => 'notify',
        SNAME => 'scott');
END;
/

```

```

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        GNAME => 'scott_mg',
        TYPE => 'PACKAGE BODY',
        ONAME => 'notify',

```

```
        SNAME => 'scott');  
END;  
/  
  
-- 通知機能の構築が完了してから、競合解消方法の1つとして、通知機能を  
-- 追加します。この機能が単に競合を通知するだけであっても、  
-- 追加する必要があります。次のAPIの例で、USER FUNCTIONとして  
-- 通知機能を追加する方法を示します。
```

```
BEGIN  
    DBMS_REPCAT.ADD_UNIQUE_RESOLUTION(  
        SNAME => 'scott',  
        ONAME => 'emp',  
        CONSTRAINT_NAME => 'emp_ename_unique',  
        SEQUENCE_NO => 1,  
        METHOD => 'USER FUNCTION',  
        COMMENT => 'Notify DBA',  
        PARAMETER_COLUMN_NAME => 'ename',  
        FUNCTION_NAME => 'scott.notify.emp_unique_violation');  
END;  
/  
  
-- 通知機能を追加すると、実際の競合解消方法を表に追加できる  
-- ようになります。次のAPIの例で、APPEND SITE NAMEという  
-- 一意性競合解消方法をレプリケート表に追加する方法を  
-- 示します。
```

```
BEGIN  
    DBMS_REPCAT.ADD_UNIQUE_RESOLUTION(  
        SNAME => 'scott',  
        ONAME => 'emp',  
        CONSTRAINT_NAME => 'emp_ename_unique',  
        SEQUENCE_NO => 2,  
        METHOD => 'APPEND SITE NAME',  
        PARAMETER_COLUMN_NAME => 'ename');  
END;  
/  
  
-- 競合解消方法を定義した後、その方法を受け入れる表の  
-- レプリケーション・サポートを再生成する必要があります。
```

```
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        SNAME => 'scott',  
        ONAME => 'emp',  
        TYPE => 'TABLE',  
        MIN_COMMUNICATION => TRUE);  
END;
```

```

END;
/

-- レプリケーション・サポートを再生成した後に、「RESUME_MASTER_ACTIVITY プロシージャ」で
-- レプリケーション・アクティビティを再開します。
BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

```

## 削除競合の競合回避方法の作成

行の削除を行う場合は、比較する 2 つの値がある更新競合とは違って、1 つの値しか存在しないので、前述の更新競合解消方法を適用できません。

レプリケーション環境で行を削除する最善の方法は、削除対象の行をマークし、マークされたすべてのレコードを表から定期的にパージして競合を避けることです。物理的に行が削除されるのではなく、比較される 2 つの値が存在するので、競合が発生しても、すべてのマスター・サイトでデータを収束させることができます。ただし、他のエラーが発生していないことを前提とします。データが収束したことを確認した後、レプリケート・パージ・プロシージャを使用して、マークされた行をパージします。

データベース用のフロントエンド・アプリケーションの開発では、削除対象としてマークされた行にフィルタをかけた方が便利です。このようにすると、ユーザーにはその行が物理的に削除されたように見えます。フィルタをかけるには、単に SELECT 文で削除対象としてマークされた行をデータ・セットから除外します。たとえば、次のように SELECT 文を指定して、現在の従業員をリストします。

```
SELECT * FROM emp WHERE remove_date IS NULL;
```

次に、削除競合を回避するレプリケート表を作成する方法を示します。また、削除対象としてマークされたレコードをパージするために、プロシージャ型レプリケーションを使用する方法も示します。

```

CONNECT repadmin/repadmin@orc1.world

BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

```

```

-- レプリケート表に削除レコードのマークが格納される列を追加する必要があります。
-- 削除レコードのマークには、タイムスタンプを使用することをお勧めします。タイムスタンプに
-- よって、レコードがマーク付けされた時間が明らかになります。タイムスタンプを使用する

```

-- ので、新しい列のデータ型は DATE です。既存のレプリケート表に REMOVE\_DATE 列を追加  
-- するために、DBMS\_REPCAT.「ALTER\_MASTER\_REPOBJECT プロシージャ」を使用します。

```
BEGIN
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        DDL_TEXT => 'ALTER TABLE scott.emp ADD (remove_date DATE)');
END;
/
```

-- 新しい列をレプリケート・オブジェクトに挿入した後に、影響を受ける  
-- オブジェクトのためのレプリケーション・サポートを、必ず再生成して  
-- ください。この作業は、レプリケート・オブジェクトを変更した後、  
-- ただちに実行する必要があります。

```
BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/
```

-- 次のパッケージが、レプリケーション環境内のすべてのマスター・サイトに  
-- レプリケートされます。このパッケージによって、指定された表からすべての  
-- "マーク済" レコードがパージされます。

```
BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        GNAME => 'scott_mg',
        TYPE => 'PACKAGE',
        ONAME => 'purge',
        SNAME => 'scott',
        DDL_TEXT => 'CREATE OR REPLACE PACKAGE scott.purge AS
                    PROCEDURE remove_emp(purge_date DATE);
                    END;');
END;
/
```

```
BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        GNAME => 'scott_mg',
        TYPE => 'PACKAGE BODY',
        ONAME => 'purge',
```

```

SNAME => 'scott',
DDL_TEXT => 'CREATE OR REPLACE PACKAGE BODY scott.purge AS
            PROCEDURE remove_emp(purge_date IN DATE) IS
            BEGIN
                DBMS_REPUTIL.REPLICATION_OFF;
                LOCK TABLE scott.emp IN EXCLUSIVE MODE;
                DELETE scott.emp WHERE remove_date IS NOT NULL AND
                    remove_date < purge_date;
                DBMS_REPUTIL.REPLICATION_ON;
            EXCEPTION WHEN others THEN
                DBMS_REPUTIL.REPLICATION_ON;
            END;
            END;' );

END;
/

```

-- パッケージ（パッケージおよびパッケージの本体）を作成した後に、  
-- 各コンポーネントのレプリケーション・サポートを生成する必要があります。  
-- レプリケーション・サポートを生成すると、作成者のシノニムが作成され、  
-- レプリケート・オブジェクトとしてマスター・グループに追加されます。  
-- このシノニムのラベルは、DEFER\_PURGE.REMOVE\_EMP です。

```

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'purge',
        TYPE => 'PACKAGE',
        MIN_COMMUNICATION => TRUE);
END;
/

```

```

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'purge',
        TYPE => 'PACKAGE BODY',
        MIN_COMMUNICATION => TRUE);
END;
/

```

-- レプリケーション・サポートを再生成した後に、「RESUME\_MASTER\_ACTIVITY プロシージャ」で  
-- レプリケーション・アクティビティを再開します。

```

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

```

## 競合解消の監査

Oracle によって更新、削除または一意性の競合が検出され、正常に解消された場合は、DBA\_REPRESOLUTION\_STATISTICS データ・ディクショナリ・ビューを問い合わせることによって、競合解消に使用された方法の情報を表示できます。このビューは、競合に関連する表に対して競合解消統計収集を有効にした場合にのみ更新されます。

**関連項目：** 詳細は、9-29 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#)」を参照してください。

## 競合解消統計の収集

DBMS\_REPCAT パッケージの REGISTER\_STATISTICS プロシージャを使用して、正常に実行された、表の更新、削除および一意性の競合解消の情報を収集します。次の例では、ACCT\_REC スキーマ内の EMP 表の統計が収集されます。

```
DBMS_REPCAT.REGISTER_STATISTICS(sname => 'acct_rec',  
                                oname  => 'emp');
```

**関連項目：** 詳細は、8-124 ページの「[REGISTER\\_STATISTICS プロシージャ](#)」を参照してください。

## 競合解消統計の表示

ある表に対して REGISTER\_STATISTICS をコールすると、その表で正常に解消された各競合は DBA\_REPRESOLUTION\_STATISTICS ビューに記録されます。解消不可能な競合の情報は、オブジェクトが登録されているかどうかに関係なく、通常、DEFERROR ビューに記録されます。

**関連項目：** 詳細は、9-29 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#)」および 9-44 ページの「[DEFERROR](#)」を参照してください。

## 競合解消統計の取消し

正常に実行された、表の更新、削除および一意性の競合解消の情報収集が不要になった場合は、DBMS\_REPCAT パッケージの CANCEL\_STATISTICS プロシージャを使用します。次の例では、ACCT\_REC スキーマ内の EMP 表の統計収集が取り消されます。

```
DBMS_REPCAT.CANCEL_STATISTICS(sname => 'acct_rec',  
                               oname  => 'emp');
```



**関連項目：** 詳細は、8-80 ページの「[CANCEL\\_STATISTICS プロシージャ](#)」を参照してください。

## 統計情報の削除

正常に実行された、更新、削除および一意性の競合解消の情報を記録するように表を登録している場合は、DBMS\_REPCAT パッケージの PURGE\_STATISTICS プロシージャをコールすることによって、DBA\_REPRESOLUTION\_STATISTICS ビューからこの情報を削除できます。

次の例では、1 月 1 日～3 月 31 日の、EMP 表に対する挿入、更新および削除により解消した競合についての統計がパージされます。

```
DBMS_REPCAT.PURGE_STATISTICS(sname      => 'acct_rec',  
                              oname       => 'emp',  
                              start_date  => '01-JAN-99',  
                              end_date    => '31-MAR-99');
```

**関連項目：** 詳細は、8-121 ページの「[PURGE\\_STATISTICS プロシージャ](#)」を参照してください。



---

## API を使用したレプリケート環境の管理

この章では、レプリケーション・マネージメント API を使用して、レプリケーション環境を管理する方法を説明します。次に、項目を示します。

- マスター・サイトの管理
- スナップショット・サイトの管理
- 遅延トランザクションの管理
- エラー・キューの管理
- レプリケート・オブジェクトの変更
- エクスポートおよびインポートを使用したオフライン・インスタンス化の実行
- レプリケート表間の違いの判別
- データ・ディクショナリ・ビュー内のコメント・フィールドの更新

## マスター・サイトの管理

事業の拡大や縮小、緊急事態などにより、データ配信を変更する場合、レプリケート環境の構成も変更する必要があります。この項では、レプリケート環境のマスター・サイトを管理する方法を説明します。この項で説明する内容に従って、マスター・サイトを変更して再構成します。

### マスター定義サイトの変更

ほとんどのレプリケーション管理タスクは、マスター定義サイトのみで実行できます。DBMS\_REPCAT.RELOCATE\_MASTERDEF プロシージャを使用すると、マスター定義サイトを別のマスター・サイトに移動できます。この API は、マスター定義サイトが利用できなくなったために、新しいマスター定義サイトを指定する必要がある場合に使用します（7-3 ページの「オプション 2」を参照してください）。

#### オプション 1

すべてのマスター・サイトを使用できる場合は、次のスクリプトを実行します。

**実行者:** レプリケーション管理者

**実行場所:** いずれかのマスター・サイト

**レプリケーションの状態:** 通常

```
CONNECT repadmin/repadmin@orc1.world

BEGIN
    DBMS_REPCAT.RELOCATE_MASTERDEF (
        GNAME => 'scott_mg',
        OLD_MASTERDEF => 'orc1.world',
        NEW_MASTERDEF => 'orc2.world',
        NOTIFY_MASTERS => TRUE,
        INCLUDE_OLD_MASTERDEF => TRUE);
END;
/
```

## オプション 2

古いマスター定義サイトを利用できない場合は、次のスクリプトを実行します。

**実行者:** レプリケーション管理者

**実行場所:** いずれかのマスター・サイト

**レプリケーションの状態:** 通常

```
CONNECT repadmin/repadmin@orc3.world

BEGIN
  DBMS_REPCAT.RELOCATE_MASTERDEF (
    GNAME => 'scott_mg',
    OLD_MASTERDEF => 'orc1.world',
    NEW_MASTERDEF => 'orc2.world',
    NOTIFY_MASTERS => TRUE,
    INCLUDE_OLD_MASTERDEF => FALSE);
END;
/
```

**関連項目:** このプロシージャの使用方法の詳細は、8-125 ページの「[RELOCATE\\_MASTERDEF プロシージャ](#)」を参照してください。

## マスター・サイトの追加

レプリケート環境を拡張した場合は、[ADD\\_MASTER\\_DATABASE プロシージャ](#)を使用して、既存のマスター・グループにマスター・サイトを追加できます。このプロシージャを実行すると、既存のマスター・オブジェクトが新しいサイトにレプリケートされます。

新しいマスター・サイトを追加する前に、それがレプリケーション用に正しく設定されていることを確認します。2-4 ページの「[マスター・サイトの設定](#)」のスクリプトに従って、マスター・サイトを設定します。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイト

**レプリケーションの状態:** 停止

```
CONNECT repadmin/repadmin@orc1.world
```

レプリケーションの状態が「通常」の場合は、「停止」に変更してください。

```
BEGIN
  DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
    GNAME => 'scott_mg');
END;
/
```

```
BEGIN
  DBMS_REPCAT.ADD_MASTER_DATABASE (
```

```
        GNAME => 'scott_mg',
        MASTER => 'orc4.world',
        USE_EXISTING_OBJECTS => TRUE,
        COPY_ROWS => TRUE,
        PROPAGATION_MODE => 'ASYNCHRONOUS');
END;
/

-- 注意: DBA_REPCATLOG ビューが空になるまで、待機する必要があります。このビューには、実行が正
-- 常に終了した後で消去される、一時情報が含まれます。
-- DBA_REPCATLOG ビューを監視するために、別の SQL*Plus セッションで
-- 次の SELECT 文を実行してください。
--
--SELECT * FROM dba_repsites WHERE gname = 'scott_mg';

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
```

## マスター・サイトの削除

マスター・グループからマスター・サイトを削除する場合は、[REMOVE\\_MASTER\\_DATABASES](#) プロシージャを使用します。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイト

**レプリケーションの状態:** 停止

```
CONNECT repadmin/repadmin@orc1.world

-- レプリケーションの状態が「通常」の場合は、「停止」に変更してください。
BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
BEGIN
    DBMS_REPCAT.REMOVE_MASTER_DATABASES (
        GNAME => 'scott_mg',
        MASTER_LIST => 'orc4.world');
END;
/
```

```
-- 注意: DBA_REPCATLOG ビューが空になるまで、待機する必要があります。
--DBA_REPCATLOG ビューを監視するために、別の SQL*Plus セッションで
-- 次の SELECT 文を実行してください。
--
--SELECT * FROM dba_repcatlog WHERE gname = 'scott_mg';

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
```

## スナップショット・サイトの管理

スナップショット・レプリケーションでは、ユーザーの要求、セキュリティの構成およびフロントエンド・アプリケーションにあわせて、柔軟にデータ・セットを作成できます。次の2つの項では、レプリケーション・マネージメント API を使用して、スナップショット・サイトを管理する方法を説明します。

### グループの所有者の使用

新しいスナップショット・グループとその関連オブジェクトを定義する場合は、グループの所有者を指定します。所有者を指定しておくと、同じマスター・グループに基づいて、1つのスナップショット・サイトに複数のスナップショット・グループを作成できます。スナップショット・サイトでグループの所有者を指定すると、同じマスター・グループに基づいて、複数のスナップショット・グループを作成できます。複数のグループを作成するには、スナップショット・サイトで、異なるスキーマに基づくスナップショット・グループを作成します。

**関連項目：** グループの所有者の使用方法与複数のデータ・セットを使用する利点の詳細は、『Oracle8i レプリケーション・ガイド』の第3章「スナップショットの概念およびアーキテクチャ」、「編成メカニズム」を参照してください。

```
-- 次のプロシージャは、リモート・スナップショット・サイトでスナップショット管理者が
-- 実行する必要があります。
```

```
CONNECT snapadmin/snapadmin@snap1.world
```

```
--GNAME パラメータに指定するマスター・グループは、ターゲット・マスター・サイトで
-- レプリケーション中のマスター・グループの名前にする必要があります。
-- また、GOWNER パラメータで、同じスナップショット・サイトの同じマスター・グループに
-- 基づく複数のスナップショット・グループが作成されるようにする、
-- 追加の識別子を指定できます。
```

```
-- この例では、グループの所有者 BOB および JANE に対するスナップショット・グループを作成します。
```

-- これらの2つのスナップショット・グループは、同じマスター・グループに基づいています。

-- グループの所有者 (GOWNER) BOB に対するスナップショット・グループを作成します。

```
BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP (
        GNAME => 'scott_mg',
        MASTER => 'orcl.world',
        PROPAGATION_MODE => 'ASYNCHRONOUS',
        GOWNER => 'bob');
END;
/
```

-- グループの所有者 (GOWNER) JANE に対するスナップショット・グループを作成します。

```
BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP (
        GNAME => 'scott_mg',
        MASTER => 'orcl.world',
        PROPAGATION_MODE => 'ASYNCHRONOUS',
        GOWNER => 'jane');
END;
/
```

-- スナップショット・オブジェクトの作成時に使用した GOWNER の値は、

-- 前述のプロシージャでスナップショット・グループを作成した時に指定した

-- GOWNER の値と同じにする必要があります。スナップショット・グループを作成した後で、BOB および

-- JANE が所有する SCOTT\_MG スナップショット・グループ内で同じマスター表のスナップショットを作成

-- できます。

-- **警告:** 作成された各オブジェクトに一意な名前が付けられていることを確認

-- してください。

-- 複数のスナップショット・グループを作成するために GOWNER を使用する際に

-- オブジェクト名が重複していると、問題が発生する場合があります。オブジェクト名の

-- 競合を避けるため、次のプロシージャ (スナップショット SCOTT.BONUS\_bob の

-- 作成) にあるように、作成するオブジェクト名の後に GOWNER 値を付けます。

-- このような命名方法を使用すると、名前が競合するオブジェクトを作成なくなります。

-- BOB が所有する SCOTT\_MG スナップショット・グループ内の同じ scott.bonus 表に基づいて、

-- スナップショットを作成します。

```
BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        ONAME => 'bonus_bob',
        TYPE => 'SNAPSHOT',
        DDL_TEXT => 'CREATE SNAPSHOT scott.bonus_bob REFRESH FAST WITH
                    PRIMARY KEY FOR UPDATE AS SELECT * FROM
                    scott.bonus@orcl.world',
```



```
MIN_COMMUNICATION => TRUE,  
GOWNER => 'bob');  
END;  
/
```

--JANE が所有する SCOTT\_MG スナップショット・グループ内の同じ scott.bonus 表に基づいて、  
-- スナップショットを作成します。

```
BEGIN  
  DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (  
    GNAME => 'scott_mg',  
    SNAME => 'scott',  
    ONAME => 'bonus_jane',  
    TYPE => 'SNAPSHOT',  
    DDL_TEXT => 'CREATE SNAPSHOT scott.bonus_jane REFRESH FAST WITH  
                PRIMARY KEY FOR UPDATE AS SELECT * FROM  
                scott.bonus@orc1.world',  
    MIN_COMMUNICATION => TRUE,  
    GOWNER => 'jane');  
END;  
/
```

--BOB が所有する SCOTT\_MG スナップショット・グループ内の scott.dept 表に基づいて、  
-- スナップショットを作成します。

```
BEGIN
  DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
    GNAME => 'scott_mg',
    SNAME => 'scott',
    ONAME => 'dept_bob',
    TYPE => 'SNAPSHOT',
    ddl_text => 'CREATE SNAPSHOT scott.dept_bob REFRESH FAST WITH
                primary key for update as select * from
                scott.dept@orcl.world',
    MIN_COMMUNICATION => TRUE,
    GOWNER => 'bob');
END;
/
```

--JANE が所有する SCOTT\_MG スナップショット・グループ内の scott.emp 表に基づいて、  
-- スナップショットを作成します。

```
BEGIN
  DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
    GNAME => 'scott_mg',
    SNAME => 'scott',
    ONAME => 'emp_jane',
    TYPE => 'SNAPSHOT',
    DDL_TEXT => 'CREATE SNAPSHOT scott.emp_jane REFRESH FAST WITH
                primary key for update as select * from
                scott.emp@orcl.world',
    MIN_COMMUNICATION => TRUE,
    GOWNER => 'jane');
END;
/
```

--BOB が所有する SCOTT\_MG スナップショット・グループ内の scott.salgrade 表に基づいて、  
 -- スナップショットを作成します。

```
BEGIN
  DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
    GNAME => 'scott_mg',
    SNAME => 'scott',
    ONAME => 'salgrade_bob',
    TYPE => 'SNAPSHOT',
    DDL_TEXT => 'CREATE SNAPSHOT scott.salgrade_bob REFRESH FAST WITH
                primary key for update as select * from
                scott.salgrade@orc1.world',
    MIN_COMMUNICATION => TRUE,
    GOWNER => 'bob');
END;
/
```

-- スナップショット・グループを構築した後に、スナップショットをリフレッシュ・グループに  
 -- 追加する必要があります。スナップショットをリフレッシュ・グループに追加する方法の  
 -- 詳細は、第5章「スナップショット・グループの作成」（ステップ6）を  
 -- 参照してください。

## スナップショット・グループのマスター・サイトの変更

スナップショット・グループのマスター・サイトを他のマスター・サイトに変更するには、  
 DBMS\_REPCAT パッケージの SWITCH\_SNAPSHOT\_MASTER プロシージャをコールしま  
 す。例を次に示します。

```
BEGIN
  DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER(
    gname          => 'sales',
    master         => 'dbs2.acme.com'
    execute_as_user => 'FALSE');
END;
/
```

この例では、SALES オブジェクト・グループのマスター・サイトが DBS2 マスター・サイトに  
 変更されます。このプロシージャは、マスター・サイトを変更するスナップショット・サ  
 イトでコールする必要があります。また、新しいデータベースとして、レプリケート環境の  
 マスター・サイトを指定する必要があります。このプロシージャをコールすると、新しいマ  
 スターを使用して、ローカル・スナップショット・グループのスナップショットごとに完全  
 リフレッシュが実行されます。

切り替えられたスナップショットの古いマスター・サイトにある SYS.SLOG\$ 表のエントリは削除されません。このため、古いマスター・サイトにある、切り替えられた更新可能なスナップショットの MLOG\$ 表は際限なく増大する可能性があります。これを回避するには、DBMS\_SNAPSHOT.PURGE\_LOG をコールしてパージします。

**関連項目：** 詳細は、8-133 ページの「[SWITCH\\_SNAPSHOT\\_MASTER プロシージャ](#)」を参照してください。

## スナップショット・サイトの削除

スナップショット・サイトのレプリケーション・アクティビティを削除する必要性は、様々な場合に生じます。たとえば、データ要件が変更されたり、従業員が退職した場合などです。いずれの場合も、データベース管理者がターゲット・スナップショット・サイトに対するレプリケーション・サポートを削除する必要があります。

### 配置テンプレートを使用して作成したスナップショット・グループの削除

スナップショット・サイトで配置テンプレートをインスタンスエートして作成されたスナップショット・グループを削除する方法は、ここに示す方法とは少し異なります。リモート・スナップショット・サイトにあるスナップショット・グループを削除する場合は、ターゲット・マスター・サイトで、スナップショット・グループに対して [DROP\\_SITE\\_INSTANTIATION プロシージャ](#) を実行する必要があります。このプロシージャは、スナップショット・グループに関連するメタデータを削除します。また、そのサイトに関する配置テンプレートのデータも削除します。

[DROP\\_SITE\\_INSTANTIATION プロシージャ](#) には、パブリックとプライベートの両バージョンがあります。パブリック・バージョンでは、スナップショット・グループの所有者がスナップショット・サイトを削除できます。これに対して、プライベート・バージョンでは、スナップショット・グループの所有者のかわりにレプリケーション管理者が、スナップショット・サイトを削除します。

### パブリック

次のスクリプトをスナップショット・グループの所有者が実行します。

**実行者：** スナップショット・グループの所有者

**実行場所：** ターゲット・スナップショット・サイト用マスター・サイト

**レプリケーションの状態：** 通常

```
CONNECT scott/tiger@orcl.world

--Oracle8i Lite データベースでインスタンスiertされたスナップショット・サイトを
-- 削除する必要がある場合、詳細はOracle8i Lite のドキュメントを
-- 参照してください。

BEGIN
    DBMS_REPCAT.INSTANTIATE.DROP_SITE_INSTANTIATION(
        REFRESH_TEMPLATE_NAME => 'personnel',
        SITE_NAME => 'snap1.world');
END;
/

--DROP_SITE_INSTANTIATION プロシージャを実行した後に、
-- リモート・スナップショット・サイトに接続し、スナップショット・グループを削除する必要がある
-- あります（災害や盗難が原因でリモート・スナップショット・サイトに接続できない場合、
-- ターゲット・スナップショット・グループはリフレッシュされませんが、既存のデータは、
-- 引き続きそのスナップショット・サイトに残っています）。
CONNECT snapadmin/snapadmin@snap1.world

-- スナップショット・グループの内容を物理的に削除する場合は、DROP_CONTENTS
-- パラメータに必ず TRUE を指定してください。

BEGIN
    DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP (
        GNAME => 'scott_mg',
        DROP_CONTENTS => TRUE);
END;
/

-- スナップショット・グループを削除した後で、リフレッシュ・グループを削除する必要があります。

BEGIN
    DBMS_REFRESH.DESTROY (
        NAME => 'personnel01');
END;
/
```

### プライベート

次のスナップショットをスナップショット・グループの所有者のかわりに、レプリケーション管理者が実行します。

**実行者:** レプリケーション管理者

**実行場所:** ターゲット・スナップショット・サイト用マスター・サイト

**レプリケーションの状態:** 通常

```
CONNECT repadmin/repadmin@orc1.world
```

```
--Oracle8i Lite データベースでインスタンスiertされたスナップショット・サイトを  
-- 削除する必要がある場合、詳細はOracle8i Lite のドキュメントを  
-- 参照してください。
```

```
BEGIN  
    DBMS_REPCAT_RGT.DROP_SITE_INSTANTIATION (  
        REFRESH_TEMPLATE_NAME => 'personnel',  
        USER_NAME => 'scott',  
        SITE_NAME => 'snap1.world');  
END;  
/
```

```
--DROP_SITE_INSTANTIATION プロシージャを実行した後に、  
-- リモート・スナップショット・サイトに接続し、スナップショット・グループを削除する必要が  
-- あります（災害や盗難が原因でリモート・スナップショット・サイトに接続できない場合、  
-- ターゲット・スナップショット・グループはリフレッシュされませんが、既存のデータは、  
-- 引き続きそのスナップショット・サイトに残っています）。
```

```
CONNECT snapadmin/snapadmin@snap1.world
```

```
-- スナップショット・グループの内容を物理的に削除する場合は、DROP_CONTENTS パラメータに  
-- 必ず TRUE を指定してください。
```

```
BEGIN  
    DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP (  
        GNAME => 'scott_mg',  
        DROP_CONTENTS => TRUE);  
END;  
/
```

## スナップショット・サイトのスナップショット・オブジェクトの削除

スナップショット・サイトのレプリケーション・サポートを削除する場合の最も安全な方法は、スナップショット・サイトのレプリケート・オブジェクトまたはグループを物理的に削除することです。次の2つの項では、スナップショット・グループに接続した状態で、これらのオブジェクトとグループを削除する方法を示します。

ここに示すプロシージャは、スナップショットがターゲット・マスター・サイトに接続した状態で実行する必要があります。接続していると、マスター・サイトにあるすべての関連メタデータを確実に削除できるためです。マスター・サイトに接続できない場合は、7-14 ページの「[マスター・サイトのクリーン・アップ](#)」に示すスクリプトを実行して、関連メタデータを手動で削除します。

**スナップショット・サイトでのスナップショット・グループの削除** スナップショット・サイトからスナップショット・グループを削除する場合は、[DROP\\_SNAPSHOT\\_REPGROUP](#) プロシージャを使用します。ターゲット・マスター・サイトに接続した状態でこのプロシージャを実行すると、マスター・サイトにあるターゲット・スナップショット・グループのメタデータが削除されます。接続できない場合は、7-14 ページの「[マスター・サイトのクリーン・アップ](#)」を参照してください。

**実行者:** スナップショット管理者

**実行場所:** リモート・スナップショット・サイト

**レプリケーションの状態:** 該当せず

```
CONNECT snapadmin/snapadmin@snap1.world
```

```
-- スナップショット・グループの内容を物理的に削除する場合は、DROP_CONTENTS パラメータに  
-- 必ず TRUE を指定してください。
```

```
BEGIN  
    DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP (  
        GNAME => 'scott_mg',  
        DROP_CONTENTS => TRUE);  
END;  
/
```

**スナップショット・サイトの各スナップショットの削除** スナップショット・サイトから各スナップショットを削除する場合は、[DROP\\_SNAPSHOT\\_REPOBJECT プロシージャ](#) API を使用します。このプロシージャを実行すると、マスター・サイトにあるターゲット・スナップショットのメタデータが削除されます。ターゲット・マスター・サイトに接続した状態でこのプロシージャを実行すると、マスター・サイトにあるターゲット・スナップショット・グループのメタデータが削除されます。接続できない場合は、7-14 ページの「[マスター・サイトのクリーン・アップ](#)」を参照してください。

**実行者:** スナップショット管理者

**実行場所:** リモート・スナップショット・サイト

**レプリケーションの状態:** 該当せず

```
CONNECT snapadmin/snapadmin@snap1.world
```

```
-- スナップショットの内容を物理的に削除する場合は、DROP_CONTENTS パラメータに  
-- 必ず TRUE を指定してください。
```

```
BEGIN  
    DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT (  
        SNAME => 'scott',  
        ONAME => 'bonus',  
        TYPE => 'SNAPSHOT',  
        DROP_OBJECTS => TRUE);  
END;  
/
```

### マスター・サイトのクリーン・アップ

ターゲット・マスター・サイトに接続した状態でスナップショット・グループまたはスナップショット・オブジェクトを削除できない場合は、マスター・サイトにある関連メタデータを手動で削除する必要があります。メタデータをクリーン・アップすると、マスター表に加えられた変更をスナップショット・ログに保持する必要がなくなります。次に示すスクリプトを実行すると、スナップショット・グループまたはオブジェクトを削除した後で、マスター・サイトをクリーン・アップできます。

**スナップショット・グループを削除した後のクリーン・アップ** マスター・サイトに接続していない状態で、7-13 ページの「[スナップショット・サイトでのスナップショット・グループの削除](#)」に示したスクリプトを実行した場合は、次のスクリプトを実行してターゲット・マスター・サイトをクリーン・アップします。

**実行者:** レプリケーション管理者

**実行場所:** ターゲット・スナップショット・サイト用マスター・サイト

**レプリケーションの状態:** 通常



```
CONNECT repadmin/repadmin@orc1.world
```

```
BEGIN
```

```
  DBMS_REPCAT.UNREGISTER_SNAPSHOT_REPGROUP (  
    GNAME => 'scott_mg',  
    SNAPSITE => 'snap1.world');  
END;
```

```
/
```

-- スナップショット・グループの登録を解除した後に、ターゲット・スナップショットとして  
-- マークされたスナップショット・ログをパージする必要があります。  
-- スナップショット・レプリケーション・グループに属していた各スナップショットに対して、  
-- `PURGE_SNAPSHOT_FROM_LOG` プロシージャを実行します。

-- **注意:** なんらかの原因でスナップショット・グループの登録解除に失敗した場合でも、  
-- 次のステップを実行できます。

```
BEGIN
```

```
  DBMS_SNAPSHOT.PURGE_SNAPSHOT_FROM_LOG (  
    SNAPOWNER => 'scott',  
    SNAPNAME => 'emp',  
    SNAPSITE => 'snap1.world');  
END;
```

```
/
```

```
BEGIN
```

```
  DBMS_SNAPSHOT.PURGE_SNAPSHOT_FROM_LOG (  
    SNAPOWNER => 'scott',  
    SNAPNAME => 'dept',  
    SNAPSITE => 'snap1.world');  
END;
```

```
/
```

```
BEGIN
```

```
  DBMS_SNAPSHOT.PURGE_SNAPSHOT_FROM_LOG (  
    SNAPOWNER => 'scott',  
    SNAPNAME => 'bonus',  
    SNAPSITE => 'snap1.world');  
END;
```

```
/
```

```
BEGIN
  DBMS_SNAPSHOT.PURGE_SNAPSHOT_FROM_LOG (
    SNAPOWNER => 'scott',
    SNAPNAME => 'salgrade',
    SNAPSITE => 'snap1.world');
END;
/
```

**マスター・サイトでの各スナップショット・サポートのクリーン・アップ** マスター・サイトに接続していない状態で、7-14 ページの「[スナップショット・サイトの各スナップショットの削除](#)」に示したスクリプトを実行した場合は、次のスクリプトを実行してターゲット・マスター・サイトをクリーン・アップします。

**実行者:** レプリケーション管理者

**実行場所:** ターゲット・スナップショット・サイト用マスター・サイト

**レプリケーションの状態:** 通常

```
CONNECT repadmin/repadmin@orc1.world
```

```
BEGIN
  DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT (
    SNAPOWNER => 'scott',
    SNAPNAME => 'bonus',
    SNAPSITE => 'snap1.world');
END;
/
```

-- スナップショットの登録を解除した後に、ターゲット・スナップショットとして  
-- マークされた関連するスナップショット・ログをパージする必要があります。

-- **注意:** なんらかの原因でスナップショットの登録解除に失敗した場合でも、  
-- 次のステップを実行できます。

```
BEGIN
  DBMS_SNAPSHOT.PURGE_SNAPSHOT_FROM_LOG (
    SNAPOWNER => 'scott',
    SNAPNAME => 'bonus',
    SNAPSITE => 'snap1.world');
END;
/
```

## スナップショット・ログの管理

次の項では、スナップショット・ログの管理方法を説明します。次の内容を説明します。

- [スナップショット・ログの変更](#)
- [スナップショット・ログ領域の管理](#)
- [スナップショット・ログがあるマスター表の再編成](#)
- [スナップショット・ログの削除](#)

### スナップショット・ログの変更

スナップショット・ログを作成した後で、その記憶域パラメータおよび対応するスナップショット・サポートを変更できます。次の項では、スナップショット・ログの変更方法を説明します。次のユーザーのみが、スナップショット・ログを変更できます。

- マスター表の所有者。
- マスター表に対する SELECT 権限および MLOG\$\_master\_table\_name に対する ALTER 権限を持つユーザー。master\_table\_name はスナップショット・ログのマスター表の名前です。たとえば、マスター表が EMP の場合、そのスナップショット・ログ表名は MLOG\$\_EMP です。

**スナップショット・ログ記憶域パラメータの変更** スナップショット・ログの記憶域パラメータを変更するには、ALTER SNAPSHOT LOG 文を使用します。たとえば、次の文では、SALES スキーマ内の CUSTOMERS 表のスナップショット・ログが変更されます。

```
ALTER SNAPSHOT LOG ON sales.customers
  PCTFREE 25
  PCTUSED 40;
```

**スナップショット・ログの変更によるフィルタ列の追加** スナップショット・ログに新しいフィルタ列を追加するには、SQL 文 ALTER SNAPSHOT LOG を使用します。たとえば、次の文では、SALES スキーマ内の CUSTOMERS 表のスナップショット・ログが変更されます。

```
ALTER SNAPSHOT LOG ON sales.customers
  ADD (zip);
```

**関連項目：** フィルタ列の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

## スナップショット・ログ領域の管理

スナップショットのリフレッシュ時にスナップショット・ログ内のどの行が使用されたかが自動的に追跡され、ログが際限なく増大しないよう、不要な行がログからパージされます。複数の単純スナップショットでは同じスナップショット・ログを使用できるため、あるスナップショットのリフレッシュにすでに使用された行が、まだ別のスナップショットのリフレッシュに必要な場合があります。このため、すべてのスナップショットで使用されるまで行はログから削除されません。

たとえば、SPDB1 データベースの CUSTOMERS スナップショットがリフレッシュされたとします。この場合、マスター表および関連するスナップショット・ログを管理するサーバーでは、このスナップショットのリフレッシュ時に使用されたスナップショット・ログの行は、SPDB2 データベースの CUSTOMERS スナップショットがこの行を使用してリフレッシュされるまで、削除されません。

スナップショット・ログからの行のパージ方法によっては、複数のスナップショットが同じマスター表から作成されているときに、スナップショット・ログが際限なく増大するという状況が発生します。このような状況は、たとえば1つのマスター表が複数のスナップショットをベースとなっており、かつ次の条件のいずれかに該当する場合に発生することがあります。

- あるスナップショットが自動リフレッシュ用に構成されておらず、手動でのリフレッシュが長期間実行されていない場合
- あるスナップショットのリフレッシュ間隔が非常に長い場合（たとえば1年ごと）
- ネットワーク障害によって、マスター表をベースとする1つ以上のスナップショットの自動リフレッシュが実行できない場合
- ネットワークまたはサイト障害によって、スナップショットが削除されたことをマスターが認識できない場合

**スナップショット・ログからの行のパージ** スナップショット・ログが使用するデータベース領域を最小にするため、スナップショット・ログは常にできる限り小さくなるようにしてください。スナップショット・ログから行を削除して、新しいログ・レコード用に領域を解放するには、次のアクションのいずれかを実行します。

- ログに対応付けられたスナップショットをリフレッシュします。これによって、スナップショット・ログから行がパージされます。
- 最も古い  $n$  回目のリフレッシュが行われたスナップショットのみに必要な行を削除することにより、ログのレコードを手動でパージします。

スナップショット・ログから行を手動でパージするには、そのログを含むデータベースで、DBMS\_SNAPSHOT パッケージの PURGE\_LOG ストアド・プロシーダを実行します。たとえば、最も前にリフレッシュされたスナップショットのみに必要なエントリを CUSTOMERS 表のスナップショット・ログからパージするには、次のプロシーダを実行します。

```
BEGIN
    DBMS_SNAPSHOT.PURGE_LOG (
        master => 'sales.customers',
        num    => 1,
        flag   => 'DELETE');
END;
/
```

**関連項目：** 詳細は、8-206 ページの「[PURGE\\_LOG プロシーダ](#)」を参照してください。

スナップショット・ログの所有者または DBMS\_SNAPSHOT パッケージの EXECUTE 権限を持つユーザーのみが、PURGE\_LOG プロシーダを実行してスナップショット・ログから行をパージできます。

**スナップショット・ログの切捨て** スナップショット・ログが増大し、多数のエクステンツが割り当てられた場合、その行のログをパージしても、ログに割り当てられた領域の量は減少しません。スナップショット・ログに割り当てられた領域を削減するには、次のステップに従います。

1. マスター表に対する排他ロックを取得して、後続のプロセス中に更新が発生しないようにします。たとえば、次のような文を実行します。

```
LOCK TABLE sales.customers IN EXCLUSIVE MODE;
```

2. 2 番目のデータベース・セッションで、スナップショット・ログ (MLOG\$ ベース表) の行を一時表にコピーします。たとえば、次のような文を実行します。

```
CREATE TABLE sales.templog AS SELECT * FROM sales.mlog$_customers;
```

3. 2 番目のセッションで、SQL 文 TRUNCATE を使用してログを切り捨てます。たとえば、次のような文を実行します。

```
TRUNCATE sales.mlog$_customers;
```

4. 2 番目のセッションで、元の行を再挿入します。この作業によって、依存スナップショットの完全リフレッシュを実行する手間が省けます。たとえば、次のような文を実行します。

```
INSERT INTO sales.mlog$_customers SELECT * FROM sales.templog;  
DROP TABLE sales.templog;
```

5. 最初のセッションで、ロールバックを実行して、マスター表の排他ロックを解除します。

```
ROLLBACK;
```

---

**注意：** 新しい位置に行をコピーしてからログを切り捨てるまでの間にマスター表に対して行われた変更は、完全リフレッシュを実行するまで表示されません。

---

スナップショット・ログの所有者または DELETE ANY TABLE システム権限を持つユーザーのみが、スナップショット・ログを切り捨てることができます。

### スナップショット・ログがあるマスター表の再編成

パフォーマンスを改善し、ディスクの使用率を最適化するために、表を定期的に再編成できます。この項では、マスター表の再編成方法および関連するスナップショットの高速リフレッシュ機能の維持方法を説明します。

**再編成通知** 表を再編成するときに、スナップショット・ログの ROWID 情報を無効にする必要があります。表の再編成は、再編成によって表が切り捨てられた場合のみ自動的に検出されます。7-22 ページの「[表 T の再編成の方法 2](#)」を参照してください。

表が切り捨てられない場合は、表の再編成を通知する必要があります。表の再編成をサポートするため、DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION プロシージャと DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION プロシージャによって、表の再編成が通知されます。これらのプロシージャは、クリーン・アップ操作を実行し、高速リフレッシュ・メカニズムが必要とするログの整合性とトリガーを検証し、表のスナップショット・ログ内の ROWID 情報を無効にします。入力は、再編成されるマスター表の名前と所有者です。出力はありません。

**マスター表の切捨て** 表が切り捨てられると、そのスナップショット・ログも切り捨てられます。ただし、主キー・スナップショットの場合、スナップショット・ログを切り捨てずに、高速リフレッシュを継続できます。スナップショット・ログに格納されている情報は保持されますが、マスター表が切り捨てられるとスナップショット・ログは ROWID に関して無効になります。スナップショット・ログ内の ROWID 情報は、新しく作成されたとみなされるため、ROWID スナップショットで高速リフレッシュ用にこの情報を使用することはできません。

PRESERVE SNAPSHOT LOG オプションを指定した場合またはオプションを指定しなかった場合は、マスター表のスナップショット・ログ内の情報は保持されますが、カレント ROWID スナップショットでログを使用して高速リフレッシュを行うには、完全リフレッシュを実行する必要があります。これがデフォルトの動作です。

---

**注意：** 以前に高速リフレッシュが可能だったスナップショットのリフレッシュを有効にするには、7-21 ページの「[データベース表の再編成方法](#)」のガイドラインに従ってください。

---

PURGE SNAPSHOT LOG オプションが指定されている場合は、スナップショット・ログはマスター表とともにパージされます。

**例** 次の2つの文のいずれも、ORDERS という名前のマスター表が切り捨てられた場合にスナップショット・ログ情報を保持します。

```
TRUNCATE TABLE orders PRESERVE SNAPSHOT LOG;  
TRUNCATE TABLE orders;
```

次の文は、マスター表とともにスナップショット・ログを切り捨てます。

```
TRUNCATE TABLE orders PURGE SNAPSHOT LOG;
```

**データベース表の再編成方法** 高速リフレッシュ機能を保持したまま表を再編成する方法が4つ提供されています。これらの方法については、次の項で説明します。他の再編成方法では、後続の高速リフレッシュを使用可能にするために、最初に完全リフレッシュを実行する必要があります。

---

**注意：** マスター表の再編成中はダイレクト・ローダーを使用しないでください。ダイレクト・ローダーは列の再順序付けの原因となり、副問合せおよび LOB スナップショットで使用されるログ情報を無効にすることがあります。

---

### 表 T の再編成の方法 1

1. 表 T に対して DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION をコールします。
2. 表を T から T\_OLD に改名します。
3. SELECT \* FROM T\_OLD として表 T を作成します。
4. 新しい表 T に対して DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION をコールします。

---

---

**注意：** 表を改名すると、対応付けられた PL/SQL のトリガーも表の新しい名前に変更されます。

---

---

DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION のコールと DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。

### 表 T の再編成の方法 2

1. 表 T に対して DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION をコールします。
2. 表 T をエクスポートします。
3. PRESERVE SNAPSHOT LOG オプションを指定して表 T を切り捨てます。
4. 従来型パスを使用して表 T をインポートします。
5. 新しい表 T に対して DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION をコールします。

---

---

**注意：** 再編成中にマスター表を切り捨てるときは、切り捨てる表の DDL の PRESERVE SNAPSHOT LOG 句を使用する必要があります。

---

---

DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION のコールと DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。



### 表 T の再編成の方法 3

1. 表 T に対して DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION をコールします。
2. 表 T をエクスポートします。
3. 表を T から T\_OLD に改名します。
4. 従来型パスを使用して表 T をインポートします。
5. 新しい表 T に対して DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION をコールします。

---

**注意：** 表を改名すると、対応付けられた PL/SQL のトリガーも表の新しい名前に変更されます。

---

DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION のコールと DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。

### 表 T の再編成の方法 4

1. 表 T に対して DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION をコールします。
2. 表 T の内容を選択してフラット・ファイルに格納します。
3. 表を T から T\_OLD に改名します。
4. 表 T\_OLD と同じ形式で表 T を作成します。
5. 従来型パスを使用して SQL\*Loader を実行します。
6. 新しい表 T に対して DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION をコールします。

---

**注意：** 表を改名すると、対応付けられた PL/SQL のトリガーも表の新しい名前に変更されます。

---

DBMS\_SNAPSHOT.BEGIN\_TABLE\_REORGANIZATION のコールと DBMS\_SNAPSHOT.END\_TABLE\_REORGANIZATION のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。

### スナップショット・ログの削除

スナップショット・ログは、マスター表や既存のスナップショットに関係なく削除できます。たとえば、次の条件のいずれかに該当する場合は、スナップショット・ログを削除できます。

- マスター表のすべてのスナップショットが削除された場合
  - マスター表のすべてのスナップショットで、高速リフレッシュではなく、完全リフレッシュを使用する場合
  - マスター表が、高速リフレッシュを必要とするスナップショットをサポートしない場合
- スナップショット・ログを削除するには、SQL\*Plus で `DROP SNAPSHOT LOG SQL` 文を実行します。たとえば、次の文では、SALES スキーマ内の CUSTOMERS という名前の表のスナップショット・ログが削除されます。

```
DROP SNAPSHOT LOG ON sales.customers;
```

マスター表の所有者または `DROP ANY TABLE` システム権限を持つユーザーのみが、スナップショット・ログを削除できます。

## 遅延トランザクションの管理

通常、Oracle レプリケーションは、遅延トランザクション・キューのプッシュおよびパージを自動的に行うよう構成されます。ただし、遅延トランザクション・キューのプッシュやパージを手動で行う必要があることもあります。次の項では、スナップショット・サイトでの遅延トランザクション・キューのプッシュおよびパージの例を説明します。この項で説明するプロセスはマスター・サイトでも同様です。

### 遅延トランザクション・キューのプッシュ

スナップショットのリフレッシュ時に、遅延トランザクション・キューのトランザクションを自動的に伝播させない場合は、次のステップを実行して、更新可能なスナップショットに加えられた変更をそのマスター表に伝播します。

-- 次のプロシージャは、リモート・スナップショット・サイトでスナップショット管理者が  
-- 実行する必要があります。

```
CONNECT snapadmin/snapadmin@snap1.world
```

-- 遅延トランザクション・キューの伝播は、トランザクションの宛先に基づきます。次の  
-- SELECT 文を実行して、遅延トランザクションとその宛先を表示します（それぞれの  
-- 宛先と、宛先ごとに保留されているトランザクション数が表示されます）。

```

SELECT DISTINCT(dblink), COUNT(deferred_tran_id)
  FROM deftrandest GROUP BY dblink;

-- 遅延トランザクションの宛先として表示された各マスター・サイトに対して、
-- DBMS_DEFER_SYS.PUSH ファンクションを実行する必要があります。

DECLARE
  temp INTEGER;
BEGIN
  temp := DBMS_DEFER_SYS.PUSH (
    DESTINATION => 'orc1.world',
    STOP_ON_ERROR => FALSE,
    DELAY_SECONDS => 0,
    PARALLELISM => 0);
END;
/

-- 前述の SELECT 文で返された各宛先に対し、前述のプロシージャを繰り返し
-- 実行してください。

```

## 遅延トランザクション・キューのページ

遅延トランザクション・キュー内の、正常に伝播されたトランザクションを定期的に自動的にページしない場合は、次のステップを実行して、手動でページする必要があります。

-- 次のプロシージャは、リモート・スナップショット・サイトでスナップショット管理者が  
-- 実行する必要があります。

```

CONNECT snapadmin/snapadmin@snap1.world

DECLARE
  temp INTEGER;
BEGIN
  temp := DBMS_DEFER_SYS.PURGE (
    PURGE_METHOD => purge_method_quick);
END;
/

```

## エラー・キューの管理

レプリケーション環境の管理者は、エラー・キューを定期的に監視し、ターゲット・マスター・サイトに正常に適用されなかった遅延トランザクションがあるかどうかを確認します。

エラー・キューを調べるには、ターゲット・マスター・サイトに接続した状態で、次の SELECT 文をレプリケーション管理者として実行します。

```
SELECT * FROM deferror;
```

エラー・キューにエラーがあった場合は、エラー条件を解決し、遅延トランザクションを再実行します。遅延トランザクションを再実行するには、2 通りの方法があります。つまり、遅延トランザクションを受信したユーザーのセキュリティ・コンテキストで再実行するか、または別のセキュリティ・コンテキストで再実行します。

## 受信者としてのエラー・トランザクションの再実行

次に示すプロシージャは、遅延トランザクションを受信したユーザーのセキュリティ・コンテキストで、指定された遅延トランザクションを再実行します。このプロシージャは、エラー条件が解決した後で実行します。

**実行者:** レプリケーション管理者

**実行場所:** エラーの発生したサイト

**レプリケーションの状態:** 通常

```
CONNECT repadmin/repadmin@orc2.world
```

```
BEGIN
  DBMS_DEFER_SYS.EXECUTE_ERROR (
    DEFERRED_TRAN_ID => '128323',
    DESTINATION => 'orc2.world');
END;
/
```

## 代替ユーザーとしてのエラー・トランザクションの再実行

次に示すプロシージャは、カレント接続ユーザーのセキュリティ・コンテキストで、指定された遅延トランザクションを再実行します。このプロシージャは、エラー条件が解決した後に実行します。

**実行者:** 接続ユーザー

**実行場所:** エラーの発生したサイト

**レプリケーションの状態:** 通常

```
CONNECT scott/tiger@orc2.world

BEGIN
  DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER (
    DEFERRED_TRAN_ID => '128323',
    DESTINATION => 'orc2.world');
END;
/
```

## レプリケート・オブジェクトの変更

データベースを変更する場合、レプリケート・オブジェクトの特性も変更する必要があります。DDL を直接実行して、レプリケート・オブジェクトを変更しないでください。直接実行すると、レプリケート環境に障害が発生する原因となります。

レプリケート・オブジェクトの特性を変更するには、DBMS\_REPCAT.[ALTER\\_MASTER\\_REPOBJECT](#) プロシージャを使用します。その場合、次のスクリプト例に示すように、必要な DDL をプロシージャ・コールに含めます (DDL\_TEXT パラメータを参照してください)。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイト

**レプリケーションの状態:** 停止

```
CONNECT repadmin/repadmin@orc1.world

-- レプリケーションの状態が「通常」の場合は、「停止」に変更してください。
BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
BEGIN
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        DDL_TEXT => 'ALTER TABLE scott.emp ADD (site VARCHAR2(20))');
END;
/

-- 新しい列をレプリケート・オブジェクトに挿入した後に、影響を受ける
-- オブジェクトのためのレプリケーション・サポートを、必ず再生成してください。
-- この作業は、レプリケーション・オブジェクトを変更した後、ただちに実行する必要があります。
-- あります。

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        SNAME => 'scott',
        ONAME => 'emp',
        TYPE => 'TABLE',
        MIN_COMMUNICATION => TRUE);
END;
/

-- 注意: DBA_REPCATLOG ビューが空になるまで、待機する必要があります。
-- DBA_REPCATLOG ビューを監視するために、別の SQL*Plus セッションで
-- 次の SELECT 文を実行してください。
--
--SELECT * FROM dba_repcatlog WHERE gname = 'scott_mg';

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/
```

# エクスポートおよびインポートを使用したオフライン・インスタンスの実行

確立されたレプリケート環境が拡張されると、新しいマスター・サイトまたはスナップショット・サイトをレプリケート環境に追加した場合に、ネットワーク・トラフィックが増大します。これは、ネットワークを通じて、表またはスナップショットの内容全体が新しいレプリケート・サイトに伝播されるためです。

このようなネットワーク・トラフィックを最少にするには、レプリケート環境を拡張する場合にオフライン・インスタンス・プロシージャを使用します。オフライン・インスタンス・セッションでは、Oracle のエクスポートおよびインポート・ユーティリティを利用できます。これによって、エクスポート・ファイルを作成し、CD-ROM やテープなどの記憶メディアを使用して新しいサイトにデータを転送できます。

## マスター・サイト

次のスクリプト例は、マスター・サイトのオフライン・インスタンス・セッションを行う方法を示します。このスクリプトを実行すると、既存のマスター・グループに新しいマスター・サイトを追加する通常の方法と比べ、ネットワーク・トラフィックをかなり減らすことができます。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイトと新しいマスター・サイト

**レプリケーションの状態:** 停止と部分的な再開

```
/*****
新しいマスター・サイトのセットアップ
```

2-4 ページの「[マスター・サイトの設定](#)」に示されたステップを完了する必要があります。新しいマスター・サイトでオフライン・インスタンス・セッションを実行する前にスキーマとデータベース・リンクを適切に作成しておく必要があります。新しいマスター・サイトから既存の各マスター・サイトへのデータベース・リンクを確実に作成してください。既存の各マスター・サイトから新しいマスター・サイトへのデータベース・リンクを作成する必要もあります。

データベース・リンクを作成した後、それぞれのデータベース・リンクへのスケジュール・リンクを定義する必要もあります。  
(ステップ 9: 「[スケジュール・リンクの作成](#)」)

```
*****/
```

```

/*****
レプリケーション・アクティビティの一時停止

```

マスター・データをエクスポートしてオフライン・インスタンス・プロセスを開始する前に、既存のマスター・サイトのレプリケーション・アクティビティを一時停止する必要があります。

```

*****/

```

```

BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        GNAME => 'scott_mg');
END;
/

```

```

/*****
未処理のトランザクションがないことを確認

```

未処理の遅延トランザクションのプッシュ、エラー・トランザクションの解決、その他の管理トランザクションのプッシュを含みます。このステップは、既存の各マスター・サイトについて実行する必要があります。

```

*****/

```

-- マスター定義サイトに接続します。

```

CONNECT repadmin/repadmin@orc1.world

```

-- エラー・トランザクション・キューをチェックします。

```

SELECT * FROM deferror;

```

-- 遅延トランザクションがエラー・キューに入れられた場合は、エラー条件を  
-- 解決して、該当の遅延トランザクションを手動で再実行する必要があります。

```

BEGIN
    DBMS_DEFER_SYS.EXECUTE_ERROR (
        DEFERRED_TRAN_ID => '128323',
        DESTINATION => 'orc1.world');
END;
/

```



```
-- 未処理の管理要求をチェックします。

SELECT * FROM dba_repcatlog;

-- 管理要求が未処理のまま残っている場合、そのようなトランザクションを手動で
-- プッシュしたり、自動的に実行されるまで待機できます。管理操作には
-- 複数のステップが含まれていることがあるので、
-- DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN プロシージャ API を
-- 複数回実行することが必要な場合もあります。

BEGIN
    DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN (
        GNAME => 'scott_mg',
        ALL_SITES => TRUE);
END;
/

/*****
オフライン・インスタンスーションの開始
*****/
-- レプリケーション管理者としてマスター定義サイトに接続します。

CONNECT repadmin/repadmin@orc1.world

BEGIN
    DBMS_OFFLINE_OG.BEGIN_INSTANTIATION (
        GNAME => 'scott_mg',
        NEW_SITE => 'orc4.world');
END;
/

注意: DBA_REPCATLOG ビューが空になるまで待機する必要があります。
このビューには、実行が正常に終了した後で消去される、一時情報が含まれます。
別の SQL*Plus セッションで、
次の SELECT 文を実行して、DBA_REPCATLOG ビューを監視します。
--
SELECT * FROM dba_repcatlog WHERE gname = 'scott_mg';
*/
```

```

/*****
エクスポートするために SCOTT/TIGER として接続

```

新しいマスター・サイトに渡すためのエクスポート・ファイルを作成するには、Oracle エクスポート・ユーティリティを使用します。エクスポート・ファイルには、新しいマスター・サイトで追加されるレプリケートされたオブジェクトが含まれます。詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

```

*****/

```

```
exp scott/tiger@orc1.world
```

```

/*****
部分的レプリケーション・アクティビティの再開

```

オフライン・インスタンスには時間がかかることがあります。エクスポートが完了した後、新しいマスター・サイト以外の残りのマスター・サイトに対して DBMS\_OFFLINE\_OG.RESUME\_SUBSET\_OF\_MASTERS プロシージャを実行して、レプリケーション・アクティビティを再開できます。次の DBMS\_OFFLINE\_OG.RESUME\_SUBSET\_OF\_MASTERS プロシージャでは、新しいマスター・サイト orc4.world 以外のすべてのマスター・サイトで、レプリケーション・アクティビティが再開されます。

```

*****/

```

-- レプリケーション管理者としてマスター定義サイトに接続します。

```
CONNECT repadmin/repadmin@orc1.world
```

```
BEGIN
  DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS (
    GNAME => 'scott_mg',
    NEW_SITE => 'orc4.world');
END;
/

```

```

/*****
新しいマスター・サイトの準備

```

マスター定義サイトから新しいマスター・サイトにエクスポート・ファイルを転送した後、新しいマスター・サイトでエクスポート・ファイルにデータがインポートされるよう準備する必要があります。新しいマスター・サイトで次のプロシージャが実行されます。

```

*****/

```

```
CONNECT repadmin/repadmin@orc4.world
```

```
BEGIN
```

```
    DBMS_OFFLINE_OG.BEGIN_LOAD (
        GNAME => 'scott_mg',
        NEW_SITE => 'orc4.world');
```

```
END;
```

```
/
```

```

/*****
エクスポート・ファイルからデータのインポート

```

エクスポート・ファイルをインポートした後、マスター定義サイトから新しいマスター・サイトにデータを転送できます。

```

*****/

```

```
imp scott/tiger@orc4.world FULL=y IGNORE=y
```

```

/*****
新しいマスター・サイトでロード・プロセスの完了

```

エクスポート・ファイルをインポートした後、新しいマスター・サイトでオフライン・インスタンスエーシヨンのプロセスを実行できます。

DBMS\_OFFLINE\_OG.END\_LOAD プロシーダを実行することで、新しいサイトにおける通常のレプリケーション・アクティビティの準備が完了します。

```

*****/

```

```
CONNECT repadmin/repadmin@orc4.world
```

```
BEGIN
```

```
    DBMS_OFFLINE_OG.END_LOAD (
        GNAME => 'scott_mg',
        NEW_SITE => 'orc4.world');
```

```
END;
```

```
/
```

```

/*****
インスタンス・プロセスの完了

新しいマスター・サイトでのステップが完了すると、オフライン・インスタンス・
プロセスを完了できます。DBMS_OFFLINE_OG.END_INSTANTIATION
プロシージャによってプロセスが完了し、すべてのマスター・サイトで通常の
レプリケーション・アクティビティが再開されます。マスター定義サイトで次の
プロシージャが実行されます。
*****/
CONNECT repadmin/repadmin@orc1.world

BEGIN
  DBMS_OFFLINE_OG.END_INSTANTIATION (
    GNAME => 'scott_mg',
    NEW_SITE => 'orc4.world');
END;
/

```

## スナップショット・サイト

マスター・サイトのオフライン・インスタンスを行うのと同じ理由から、オフライン・インスタンス・プロセスを使用して、新しいスナップショット・グループをスナップショット・サイトに作成する場合があります。スナップショット・サイトではモデム接続を使用したラップトップがターゲット・コンピュータである場合が多いため、場合によってはこの方が効果的です。

次のスクリプトは、新しいスナップショット・グループに対して、オフライン・インスタンスを行うプロセスを示しています。

**実行者:** レプリケーション管理者とスナップショット管理者

**実行場所:** マスター・サイトと新しいスナップショット・サイト

**レプリケーションの状態:** 通常

```

/*****
スナップショット・サイトの設定

2-14 ページの「スナップショット・サイトの設定」に示されたステップを
完了する必要があります。スナップショットでオフライン・インスタンスを実行する前に、
スキーマとデータベース・リンクを適切に作成しておく必要があります。
*****/

```

```

/*****
スナップショット・ログの作成

```

ターゲット・マスター表にスナップショット・ログがない場合、ターゲット・マスター・サイトにスナップショット・ログを作成する必要があります。

```

*****/

```

```

CONNECT repadmin/repadmin@orc1.world

```

```

CREATE SNAPSHOT LOG ON scott.emp;
CREATE SNAPSHOT LOG ON scott.dept;
CREATE SNAPSHOT LOG ON scott.bonus;
CREATE SNAPSHOT LOG ON scott.salgrade;

```

```

/*****
テンポラリ・スナップショットの作成

```

マスター・サイトにテンポラリ・スナップショットを作成します。ここには、エクスポート・ファイルを使用して新しいスナップショット・サイトに送ったデータが含まれます。

**注意:** 第6章「競合解消」に説明されている競合解消ルーチンを追加している場合、表には列が追加されます。次の SELECT 文では追加されている列を含むように注意してください。更新可能なスナップショットでは、マスター表のすべての列を選択するために明示的に指定する必要があります (SELECT \* は不可)。

```

*****/

```

```

CREATE SNAPSHOT scott.snap_emp REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
  FROM scott.emp@orc1.world;

```

```

CREATE SNAPSHOT scott.snap_dept REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT deptno, dname, loc
  FROM scott.dept@orc1.world;

```

```

CREATE SNAPSHOT scott.snap_bonus REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT ename, job, sal, comm
  FROM scott.bonus@orc1.world;

```

```

CREATE SNAPSHOT scott.snap_salgrade REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT grade, losal, hisal
  FROM scott.salgrade@orc1.world;

```

```

/*****
エクスポートのために SCOTT/TIGER として接続

```

新しいスナップショット・サイトに転送するためのエクスポート・ファイルを作成するために Oracle エクスポート・ユーティリティを使用します。エクスポート・ファイルには、テンポラリー・スナップショットのベース表が含まれます。詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。

**注意：** 次の例は、Oracle8i データベースでのみ使用できます。  
Oracle8i 以前のバージョンのベース表は、SNAP\$ 接頭辞が付きま  
(たとえば、SNAP\$\_SNAP\_EMP)

```

*****/

```

```
exp scott/tiger@orcl.world TABLES='snap_emp','snap_dept',
'snap_bonus','snap_salgrade'
```

```

/*****
テンポラリー・スナップショットの削除

```

エクスポートの終了後、このステップの最初に作成したテンポラリー・スナップ  
ショットを削除する必要があります。

```

*****/

```

```
CONNECT scott/tiger@orcl.world
```

```
DROP SNAPSHOT snap_emp;
DROP SNAPSHOT snap_dept;
DROP SNAPSHOT snap_bonus;
DROP SNAPSHOT snap_salgrade;
```

```

/*****
必要なスキーマとデータベース・リンクの作成

```

スナップショットでオフライン・インスタンスーションを実行する前に、新しいスナップ  
ショット・サイトを含むスキーマ（マスター・サイトのマスター・オブジェクト  
を含むスキーマと同じスキーマに含まれる必要があります）と、スナップ  
ショット・サイトからマスター・サイトへのデータベース・リンクを作成します。

```

*****/

```

```
CONNECT system/manager@snap2.world
```

```
CREATE USER scott IDENTIFIED by tiger;
```

```
GRANT connect, resource, create snapshot TO scott;
```

```
CONNECT scott/tiger@snap2.world
```

```

CREATE DATABASE LINK orcl.world CONNECT TO scott IDENTIFIED by tiger;

/*****
空のスナップショット・グループの作成

新しいスナップショット・サイトで DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP API
を実行し、スナップショットの中に空のスナップショット・グループを含めます。
*****/

CONNECT snapadmin/snapadmin@snap2.world

BEGIN
    DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP (
        GNAME => 'scott_mg',
        MASTER => 'orcl.world',
        PROPAGATION_MODE => 'ASYNCHRONOUS');
END;
/

/*****
オフライン・インスタンスエーシジョンのためにスナップショット・サイトを準備

DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD API を使用して、新しいスナップショットに
必要なサポート・メカニズムを作成します。このステップでは、前のステップで
作成したスナップショット・グループに新しいスナップショットを追加します。

インポートしたスナップショットごとに DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD API
を実行する必要があります。
*****/

CONNECT system/manager@snap2.world

BEGIN
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        MASTER_SITE => 'orcl.world',
        SNAPSHOT_ONAME => 'snap_emp');
END;
/

```

```
BEGIN
  DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
    GNAME => 'scott_mg',
    SNAME => 'scott',
    MASTER_SITE => 'orc1.world',
    SNAPSHOT_ONAME => 'snap_dept');
END;
/

BEGIN
  DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
    GNAME => 'scott_mg',
    SNAME => 'scott',
    MASTER_SITE => 'orc1.world',
    SNAPSHOT_ONAME => 'snap_bonus');
END;
/

BEGIN
  DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
    GNAME => 'scott_mg',
    SNAME => 'scott',
    MASTER_SITE => 'orc1.world',
    SNAPSHOT_ONAME => 'snap_salgrade');
END;
/

/*****
新しいスナップショット・サイトにインポートするために SCOTT/TIGER
として接続

前のステップでエクスポートしたファイルをインポートするには、Oracle
インポート・ユーティリティを使用します。データをエクスポートしたときと
同じユーザーでインポートする必要があります。(たとえば、scott/tiger)
*****/

imp scott/tiger@snap2.world FULL=y IGNORE=y
```



```

/*****
オフライン・インスタンスーションの完了

```

DBMS\_OFFLINE\_SNAPSHOT.END\_LOAD API を実行して、インポートしたスナップショットのオフライン・インスタンスーションを完了します。

```

*****/

```

```

CONNECT system/manager@snap2.world

```

```

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        SNAPSHOT_ONAME => 'snap_emp');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        SNAPSHOT_ONAME => 'snap_dept');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        SNAPSHOT_ONAME => 'snap_bonus');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        GNAME => 'scott_mg',
        SNAME => 'scott',
        SNAPSHOT_ONAME => 'snap_salgrade');
END;
/

```

```
/******  
マスター・サイトで登録するためにスナップショットをリフレッシュ
```

マスター表の最新の変更を反映するのに加え、新しいスナップショット・サイトでスナップショットをリフレッシュすると、ターゲット・マスター・サイトでオフラインでインスタンスエートされたスナップショットも登録されます。

```
*****/
```

```
CONNECT scott/tiger@snap2.world
```

```
BEGIN
```

```
    DBMS_SNAPSHOT.REFRESH ('snap_emp');
```

```
END;
```

```
/
```

```
BEGIN
```

```
    DBMS_SNAPSHOT.REFRESH ('snap_dept');
```

```
END;
```

```
/
```

```
BEGIN
```

```
    DBMS_SNAPSHOT.REFRESH ('snap_bonus');
```

```
END;
```

```
/
```

```
BEGIN
```

```
    DBMS_SNAPSHOT.REFRESH ('snap_salgrade');
```

```
END;
```

```
/
```

## レプリケート表間の違いの判別

レプリケート環境を管理するとき、2つのレプリケート表の内容が同じであるかどうかを定期的にチェックできます。両方のサイトが Oracle リリース 7.3 以上のときは、DBMS\_RECTIFIER\_DIFF パッケージ内の次のプロシージャを使用すると、2つの表の違いを識別でき、また任意で調整もできます。

### DIFFERENCES

DIFFERENCES プロシージャは、1つの表の2つのレプリカを比較し、第1のレプリカにあるが第2のレプリカにはない行、および第2のレプリカにあるが第1のレプリカにはない行をすべて判別します。このプロシージャの出力は、ユーザーが作成した2つの表に格納されます。第1の表には欠落行の値が格納され、第2の表にはそれぞれの欠落行を含むサイトを示す情報が格納されます。

### RECTIFY

RECTIFY プロシージャは、DIFFERENCES プロシージャで生成された情報を使用して、2つの表を調整します。第1の表にはあって第2の表にはない行はすべて、第2の表に挿入されます。第2の表にはあって第1の表にはない行はすべて、第2の表から削除されます。

レプリケート表のコピーをすべて同一にリストアするには、次のステップに従います。

1. 表のコピーを1つ選択して参照表にします。他のすべての表のレプリカを必要に応じて更新するときに、このコピーを使用します。
2. 違いをチェックする必要があるのが、表の中のすべての行および列か、またはサブセットのみかを判別します。

たとえば、前回の違いのチェック以降に更新されていない行は、チェックしなくてもよい場合があります。すべての列をチェックする必要はありませんが、列のリストには、表の主キーを構成する（または代替識別キーとして指定した）列がすべて含まれている必要があります。

3. 表の中のどの列をチェックするかを決定した後は、比較の結果を保持するための表を2つ作成します。

まず、比較する列のデータを保持する表を作成する必要があります。たとえば、EMPLOYEE 表の EMPNO 列、SAL 列および BONUS 列を比較する場合、次のような CREATE 文を実行する必要があります。

```
CREATE TABLE missing_rows_data (
    empno    NUMBER,
    sal      NUMBER,
    bonus    NUMBER);
```

次に、行が検出された位置を示す表も作成する必要があります。この表には、次の例に示すデータ型の列を3つ入れる必要があります。

```
CREATE TABLE missing_rows_location (  
  present      VARCHAR2(128),  
  absent      VARCHAR2(128),  
  r_id         ROWID);
```

4. 比較対象の表があるオブジェクト・グループのレプリケーション・アクティビティを中断します。グループのレプリケーション・アクティビティの中断は必須ではありませんが、最初に表を停止せずに調整すると、データに不整合が生じることがあります。
5. 参照表があるサイトで、DBMS\_RECTIFIER\_DIFF.DIFFERENCES プロシージャをコールします。

たとえば、ニューヨークのサイトとサンフランシスコのサイトの EMPLOYEE 表を比較する場合、次のようなプロシージャをコールします。

```
BEGIN  
  DBMS_RECTIFIER_DIFF.DIFFERENCES(  
    sname1          => 'hr',  
    oname1          => 'employee',  
    reference_site  => 'ny.com',  
    sname2          => 'hr',  
    oname2          => 'employee',  
    comparison_site => 'sf.com',  
    where_clause    => '',  
    column_list     => 'empno,sal,bonus',  
    missing_rows_sname => 'scott',  
    missing_rows_oname1 => 'missing_rows_data',  
    missing_rows_oname2 => 'missing_rows_location',  
    missing_rows_site => 'ny.com',  
    commit_rows     => 50);  
END;  
/
```

図 7-1 に、EMPLOYEE 表の 2 つのレプリカと、これらのレプリカに対して DIFFERENCES プロシージャを実行した場合に出力される欠落行表の例を示します。

図 7-1 レプリカ間の違いの判別

NY.COMのEMPLOYEE表				
empno	ename	deptno	sal	bonus
100	Jones	20	55,000	3,500
101	Kim	20	62,000	1,000
102	Braun	20	43,500	1,500

SF.COMのEMPLOYEE表				
empno	ename	deptno	sal	bonus
100	Jones	20	55,000	3,500
101	Kim	20	62,000	2,000
102	Braun	20	43,500	1,500
103	Rama	20	48,750	2,500

MISSING_ROWS_DATA表			
empno	sal	bonus	rowid
101	62,000	1,000	000015E8.0000.0002
101	62,000	2,000	000015E8.0001.0002
103	48,750	2,500	000015E8.0002.0002

MISSING_ROWS_LOCATION表		
present	absent	r_id
ny.com	sf.com	000015E8.0000.0002
sf.com	ny.com	000015E8.0001.0002
sf.com	ny.com	000015E8.0002.0002

2つの欠落行表が、ROWID および R\_ID 列によって対応付けられている点に注目してください。

6. 次の例に示すように、DBMS\_RECTIFIER\_DIFF.RECTIFY プロシージャをコールして、参照サイトの表に等しくなるように比較サイトの表を調整します。

```
BEGIN
  DBMS_RECTIFIER_DIFF.RECTIFY(
    sname1          => 'hr',
    oname1          => 'employee',
    reference_site  => 'ny.com',
    sname2          => 'hr',
    oname2          => 'employee',
    comparison_site => 'sf.com',
    column_list     => 'empno,sal,bonus',
    missing_rows_sname => 'scott',
    missing_rows_oname1 => 'missing_rows_data',
    missing_rows_oname2 => 'missing_rows_location',
    missing_rows_site => 'ny.com',
    commit_rows     => 50);
END;
/
```

必要な挿入および削除を実行している間、その変更を伝播しないよう、RECTIFY プロシージャは、比較サイトのレプリケーションを一時的に使用禁止にします。RECTIFY は、最初に必要な DELETE をすべて実行し、次に INSERT をすべて実行します。これによって、PRIMARY KEY 制約の違反は発生しなくなります。

RECTIFY プロシージャが正常に実行されると、欠落行表は空になります。

---

---

**注意：** 比較表にその他の制約がある場合は、RECTIFY のコール時にそれらに違反しないようにする必要があります。欠落行表の情報を使用して、表を直接更新することが必要な場合もあります。その場合は、必ず欠落行表から該当の行を DELETE してください。

---

---

7. レプリケート表の残りのコピーに対して、ステップ 5 および 6 を繰り返します。このプロシージャの完了時にすべてのコピーが確実に同一になるように、必ず毎回同じ参照表を使用してください。
8. マスター・グループのレプリケーション・アクティビティを再開します。

# データ・ディクショナリ・ビュー内のコメント・フィールドの更新

DBMS\_REPCAT パッケージには、レプリケーションに関連する様々なデータ・ディクショナリ・ビュー内のコメント情報を更新できるプロシージャがいくつかあります。表 7-1 に、各ビュー用にコールするプロシージャの一覧を示します。

表 7-1 アドバンスト・レプリケーション機能のビュー内のコメントの更新

ビュー	DBMS_REPCAT プロシージャ	パラメータ情報の参照先
DBA_REPGROUP	COMMENT_ON_REPGROUP ( gname              IN VARCHAR2, Comment           IN VARCHAR2)	8-83 ページの「COMMENT_ON_REPGROUP プロシージャ」
DBA_REPOBJECT	COMMENT_ON_REPOBJECT ( sname              IN VARCHAR2, oname              IN VARCHAR2, type               IN VARCHAR2, comment            IN VARCHAR2)	8-84 ページの「COMMENT_ON_REPOBJECT プロシージャ」
DBA_REPSITES	COMMENT_ON_REPSITES ( gname              IN VARCHAR2, master              IN VARCHAR, comment            IN VARCHAR2)	8-85 ページの「COMMENT_ON_REPSITES プロシージャ」
DBA_REPCOLUMN_GROUP	COMMENT_ON_COLUMN_GROUP ( sname              IN VARCHAR2, oname              IN VARCHAR2, column_group       IN VARCHAR2, comment            IN VARCHAR2)	8-81 ページの「COMMENT_ON_COLUMN_GROUP プロシージャ」
DBA_REPPRIORITY_GROUP	COMMENT_ON_PRIORITY_GROUP ( gname              IN VARCHAR2, pgroup              IN VARCHAR2, comment            IN VARCHAR2)	8-82 ページの「COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY プロシージャ」
DBA_REPPRIORITY_GROUP (サイト優先順位グループ)	COMMENT_ON_SITE_PRIORITY ( gname              IN VARCHAR2, name               IN VARCHAR2, comment            IN VARCHAR2)	8-82 ページの「COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY プロシージャ」

表 7-1 アドバンスト・レプリケーション機能のビュー内のコメントの更新

ビュー	DBMS_REPCAT プロシージャ	パラメータ情報の参照先
DBA_ REPRESOLUTION (一意性競合)	COMMENT_ON_UNIQUE_RESOLUTION( sname                IN VARCHAR2, oname                IN VARCHAR2, constraint_name      IN VARCHAR2, sequence_no          IN NUMBER, Comment              IN VARCHAR2)	COMMENT_ON_UNIQUE_RESOLUTION プロシージャのパラメータは、8-87 ページの「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」を参照してください。
DBA_ REPRESOLUTION (更新競合)	COMMENT_ON_UPDATE_RESOLUTION( sname                IN VARCHAR2, oname                IN VARCHAR2, column_group         IN VARCHAR2, sequence_no          IN NUMBER, Comment              IN VARCHAR2)	COMMENT_ON_UNIQUE_RESOLUTION プロシージャのパラメータは、8-87 ページの「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」を参照してください。
DBA_ REPRESOLUTION (削除競合)	COMMENT_ON_DELETE_RESOLUTION( sname                IN VARCHAR2, oname                IN VARCHAR2, sequence_no          IN NUMBER, comment              IN VARCHAR2)	COMMENT_ON_UNIQUE_RESOLUTION プロシージャのパラメータは、8-87 ページの「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」を参照してください。



---

# レプリケーション・マネージメント API リファレンス

Oracle レプリケーションをすべてインストールすると、レプリケーション・マネージメント・アプリケーション・プログラミング・インタフェース（API）もインストールされます。レプリケーション・マネージメント API は、管理者が各サイトのレプリケーション機能を設定および管理するための、PL/SQL パッケージのコレクションです。また、Oracle Replication Manager でも、作業の実行に各サイトのレプリケーション・マネージメント API のプロシージャやファンクションが使用されます。

この章では、Oracle レプリケーション・マネージメント API を構成するパッケージを説明します。次の内容を説明します。

- 各パッケージのプロシージャおよびファンクション
- パッケージ化されたプロシージャやファンクションのパラメータ
- プロシージャやファンクションで発生する可能性のある例外

---

**注意：** この章で説明する PL/SQL プロシージャおよびファンクションの一部は、オーバーロードされています。すなわち、1つのパッケージ内に、同一の名前のプロシージャまたはファンクションが複数ありますが、それらの仮パラメータの個数、順序またはデータ型のグループは異なります。オーバーロードされているプロシージャまたはファンクションは、その旨を記述します。オーバーロードおよび PL/SQL の一般情報の詳細は、『Oracle8i PL/SQL ユーザーズ・ガイドおよびリファレンス』を参照してください。

---

## パッケージ

Oracle のレプリケーション・マネージメント API には次のパッケージがあります。

- [DBMS\\_DEFER パッケージ](#)
- [DBMS\\_DEFER\\_QUERY パッケージ](#)
- [DBMS\\_DEFER\\_SYS パッケージ](#)
- [DBMS\\_OFFLINE\\_OG パッケージ](#)
- [DBMS\\_OFFLINE\\_SNAPSHOT パッケージ](#)
- [DBMS\\_RECTIFIER\\_DIFF パッケージ](#)
- [DBMS\\_REFRESH パッケージ](#)
- [DBMS\\_REPCAT パッケージ](#)
- [DBMS\\_REPCAT\\_ADMIN パッケージ](#)
- [DBMS\\_REPCAT\\_INSTANTIATE パッケージ](#)
- [DBMS\\_REPCAT\\_RGT パッケージ](#)
- [DBMS\\_REPUTIL パッケージ](#)
- [DBMS\\_SNAPSHOT パッケージ](#)

## Oracle レプリケーション・マネージメント API の使用例

Oracle のレプリケーション・マネージメント API を使用するには、Enterprise Manager SQL Worksheet や SQL\*Plus などの問合せツールを使用して、プロシージャまたはファンクションをコールします。たとえば、次に示す DBMS\_REPCAT.CREATE\_MASTER\_REPOBJECT プロシージャをコールすると、ACCT レプリケーション・グループ内に新しいレプリケート表 SALES.EMP が作成されます。

```
DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    sname          => 'sales',
    oname          => 'emp',
    type           => 'table',
    use_existing_object => TRUE,
    ddl_text       => 'CREATE TABLE acct_rec.emp AS . . .',
    comment        => 'created by . . .',
    retry          => FALSE,
    copy_rows      => TRUE,
    gname          => 'acct');
```

レプリケーション・マネージメント API ファンクションをコールするには、ファンクションの戻り値を受け取る環境を用意する必要があります。たとえば、次に示す無名 PL/SQL ブロックは、DBMS\_DEFER\_SYS.DISABLED ファンクションを IF 文でコールします。

```
BEGIN
  IF DBMS_DEFER_SYS.DISABLED('inst2') THEN
    DBMS_OUTPUT.PUT_LINE('Propagation to INST2 is disabled.');
```

ELSE

```
    DBMS_OUTPUT.PUT_LINE('Propagation to INST2 is enabled.');
```

END IF;

```
END;
```

## 考慮事項

レプリケーション・マネージメント API のプロシージャおよびファンクションの多くには、いくつかの重要な考慮事項があります。次にその例を示します。

- プロシージャやファンクションのいくつかは、マルチマスター構成のマスター定義サイトからのみコールします。
- マスター・グループに対して特定の管理操作を行う場合は、レプリケーション・マネージメント API のプロシージャやファンクションをコールする前に、そのグループのレプリケーション・アクティビティを中断する必要があります。
- Oracle レプリケーション・マネージメント API のプロシージャやファンクションをコールする順序は大変重要です。レプリケーション・マネージメント・コールを発行する方法の詳細は、次の項を参照してください。

## Replication Manager および Oracle レプリケーション・マネージメント API

Oracle Replication Manager では、そのほとんどのファンクションの実行に、レプリケーション・マネージメント API が使用されます。Replication Manager には次に示す機能が備わっているため、レプリケーション・マネージメント API コールを個別に発行するより便利です。

- API コールのパラメータの入力および調整を行う GUI インタフェースが用意されています。
- 大量の関連 API コールを、適切に順序付けます。
- API コールで戻された出力を、メッセージ・ボックスとエラー・ファイルに表示します。

Replication Manager のスクリプト機能を使用すると、Oracle レプリケーション・マネージメント API の使用方法を簡単に理解できます。Replication Manager で管理セッションを開始するときに、スクリプトをオンにします。作業を終了した後、スクリプトをオフにしてスクリプト・ファイルを確認します。スクリプト・ファイルには、セッション中に発行されたすべてのレプリケーション・マネージメント API コールが保存されます。Replication Manager のスクリプト機能の詳細は、Replication Manager のヘルプを参照してください。

# DBMS\_DEFER パッケージ

## サブプログラムの概要

表 8-1 DBMS\_DEFER パッケージのサブプログラム

サブプログラム	説明
<a href="#">CALL プロシージャ</a> (8-4 ページ)	リモート・プロシージャへの遅延コールを作成します。
<a href="#">COMMIT_WORK プロシージャ</a> (8-5 ページ)	正しく定義された遅延リモート・プロシージャ・コールのチェック後に、トランザクションのコミットを実行します。
<a href="#">datatype ARG プロシージャ</a> (8-6 ページ)	遅延リモート・プロシージャ・コールに渡されるデータを提供します。
<a href="#">TRANSACTION プロシージャ</a> (8-7 ページ)	新しい遅延トランザクションの開始を示します。

## CALL プロシージャ

このプロシージャは、リモート・プロシージャへの遅延コールを作成します。

### 構文

```
DBMS_DEFER.CALL (  
  schema_name      IN  VARCHAR2,  
  package_name     IN  VARCHAR2,  
  proc_name        IN  VARCHAR2,  
  arg_count        IN  NATURAL,  
  { nodes          IN  node_list_t  
  | group_name     IN  VARCHAR2 :=''});
```

**注意：** このプロシージャはオーバーロードされています。nodes パラメータと group\_name パラメータは、同時には指定できません。

## パラメータ

表 8-2 CALL プロシージャのパラメータ

パラメータ	説明
schema_name	ストアド・プロシージャが置かれたスキーマの名前です。
package_name	ストアド・プロシージャを含むパッケージの名前です。ストアド・プロシージャはパッケージの一部になります。スタンドアロン・プロシージャへの遅延コールはサポートされていません。
proc_name	コールを遅らせる、リモート・プロシージャの名前です。
arg_count	プロシージャのパラメータ数です。各パラメータにつき 1 回、DBMS_DEFER.datatype_ARG へのコールが必要です。 <b>注意:</b> パラメータの一部がデフォルトである場合でも、プロシージャの引数をすべて数に含める必要があります。
nodes	遅延コールを伝播する、完全修飾されたデータベース名の PL/SQL 表です。表の索引付けは、1 の位置から始まり、NULL エントリが見つかったとき、または NO_DATA_FOUND 例外が発生したときに終了します。表内のデータには、大文字と小文字の区別がありません。この引数はオプションです。
group_name	内部使用に予約されています。

## 例外

表 8-3 CALL プロシージャの例外

例外	説明
ORA-23304 (malformedcall)	前のコールの形式が正しくありません。
ORA-23319	パラメータ値が不適正です。
ORA-23352	nodes、または前の DBMS_DEFER.TRANSACTION コールで指定された接続先リストに重複があります。

## COMMIT\_WORK プロシージャ

このプロシージャは、正しく定義された遅延リモート・プロシージャ・コールのチェック後に、トランザクションのコミットを実行します。

## 構文

```
DBMS_DEFER.COMMIT_WORK (  
    commit_work_comment IN VARCHAR2);
```

## パラメータ

表 8-4 COMMIT\_WORK プロシージャのパラメータ

パラメータ	説明
commit_work_ comment	SQL の "COMMIT COMMENT" 文に相当します。

## 例外

表 8-5 COMMIT\_WORK プロシージャの例外

例外	説明
ORA-23304 (malformedcall)	トランザクションの形式が正しくなかったか、またはトランザクションが正常に終了しませんでした。

## datatype\_ARG プロシージャ

このプロシージャは、遅延リモート・プロシージャ・コールに渡すデータを提供します。遅延リモート・プロシージャに渡す必要があるデータに応じて、次に示すプロシージャのいずれかをコールする必要があります。

DBMS\_DEFER.CALL を実行した後で、datatype\_ARG プロシージャを使用して、遅延リモート・プロシージャの各パラメータを指定する必要があります。遅延リモート・プロシージャ・コールのデフォルトのパラメータは使用できません。たとえば、次のようなプロシージャがあるとします。

```
CREATE OR REPLACE PACKAGE my_pack AS
    PROCEDURE my_proc(a VARCHAR2, b VARCHAR2 DEFAULT 'SALES');
END;
/
```

DBMS\_DEFER.CALL プロシージャを実行するときは、パラメータごとに個別の行を MY\_PROC プロシージャに組み込む必要があります。

```
CREATE OR REPLACE PROCEDURE load_def_tx IS
    node DBMS_DEFER.NODE_LIST_T;
BEGIN
    node(1) := 'MYCOMPUTER.WORLD';
    node(2) := NULL;
    DBMS_DEFER.TRANSACTION(node);
    DBMS_DEFER.CALL('SCOTT', 'MY_PACK', 'MY_PROC', 2);
    DBMS_DEFER.VARCHAR2_ARG('TEST');
    DBMS_DEFER.VARCHAR2_ARG('SALES'); -- 必須。省略してデフォルトを使用できません。
END;
/
```

構文

```
DBMS_DEFER.NUMBER_ARG      (arg IN NUMBER);
DBMS_DEFER.DATE_ARG        (arg IN DATE);
DBMS_DEFER.VARCHAR2_ARG    (arg IN VARCHAR2);
DBMS_DEFER.CHAR_ARG        (arg IN CHAR);
DBMS_DEFER.ROWID_ARG       (arg IN ROWID);
DBMS_DEFER.RAW_ARG         (arg IN RAW);
DBMS_DEFER.BLOB_ARG        (arg IN BLOB);
DBMS_DEFER.CLOB_ARG        (arg IN CLOB);
DBMS_DEFER.NCLOB_ARG       (arg IN NCLOB);
DBMS_DEFER.NCHAR_ARG       (arg IN NCHAR);
DBMS_DEFER.NVARCHAR2_ARG   (arg IN NVARCHAR2);
DBMS_DEFER.ANY_CLOB_ARG    (arg IN CLOB);
DBMS_DEFER.ANY_VARCHAR2_ARG (arg IN VARCHAR2);
DBMS_DEFER.ANY_CHAR_ARG    (arg IN CHAR);
```

パラメータ

表 8-6 datatype\_ARG プロシージャのパラメータ

パラメータ	説明
arg	以前コールを遅延したリモート・プロシージャに渡すパラメータ値です。

例外

表 8-7 datatype\_ARG プロシージャの例外

例外	説明
ORA-23323	引数値が長すぎます。

TRANSACTION プロシージャ

このプロシージャは、新しい遅延トランザクションの開始を示します。このコールを省略した場合、DBMS\_DEFER.CALL への最初のコールが、新しいトランザクションの開始と見なされます。

構文

```
DBMS_DEFER.TRANSACTION (
    nodes IN node_list_t);
```

**注意：** このプロシージャはオーバーロードされています。入力パラメータがない場合の動作は、入力パラメータがある場合の動作と似ています。ただし、前者では、ノード・パラメータのノードではなく、DEFDEFAULTDEST ビューの nodes を使用します。

パラメータ

表 8-8 TRANSACTION プロシージャのパラメータ

パラメータ	説明
nodes	トランザクションの遅延コールを伝播する完全修飾されたデータベース名の、PL/SQL 表です。表の索引付けは、1 の位置から始まり、NULL エントリが見つかったとき、または NO_DATA_FOUND 例外が発生したときに終了します。表内のデータには、大文字と小文字の区別がありません。

例外

表 8-9 TRANSACTION プロシージャの例外

例外	説明
ORA-23304 (malformedcall)	前のトランザクションの形式が正しくなかったか、または前のトランザクションが正常に終了しませんでした。
ORA-23319	パラメータ値が不適正です。
ORA-23352	ノード・リストに重複が存在するとき、DBMS_DEFER.CALL によって発生します。



# DBMS\_DEFER\_QUERY パッケージ

## サブプログラムの概要

表 8-10 DBMS\_DEFER\_QUERY パッケージのサブプログラム

サブプログラム	説明
<a href="#">GET_ARG_FORM ファンクシ ン</a> (8-9 ページ)	遅延コールの引数の形式を判定します。
<a href="#">GET_ARG_TYPE ファンクシ ン</a> (8-10 ページ)	遅延コールの引数のタイプを判定します。
<a href="#">GET_CALL_ARGS プロシー ジャ</a> (8-11 ページ)	指定されたコールの様々な引数のテキスト・バージョンを戻しま す。
<a href="#">GET_datatype_ARG ファン クション</a> (8-12 ページ)	遅延コールの引数値を判定します。

## GET\_ARG\_FORM ファンクション

このファンクションは、遅延コールの引数の形式を判定します。このファンクションは、遅延コール・パラメータのキャラクタ・セット ID を戻します。

**関連項目：** Replication Manager での遅延トランザクションおよびエ  
ラー・トランザクションの表示の詳細は、Replication Manager のオンラ  
イン・ヘルプを参照してください。

### 構文

```
DBMS_DEFER_QUERY.GET_ARG_FORM (
    callno           IN    NUMBER,
    arg_no           IN    NUMBER,
    deferred_tran_id IN    VARCHAR2)
RETURN NUMBER;
```

### パラメータ

表 8-11 GET\_ARG\_FORM ファンクションのパラメータ

パラメータ	説明
callno	DEFCALL ビューからのコール識別子です。
arg_no	コールの引数リスト内のパラメータの位置です。指定できるパラメー タの位置は、"1" からコール内のパラメータの数までです。

表 8-11 GET\_ARG\_FORM ファンクションのパラメータ

パラメータ	説明
deferred_tran_id	遅延トランザクション ID です。

例外

表 8-12 GET\_ARG\_FORM ファンクションの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。

戻り値

表 8-13 GET\_ARG\_Form ファンクションの戻り値

戻り値	対応するデータ型
1	CHAR, VARCHAR2, CLOB
2	NCHAR, NVARCHAR2, NCLOB

GET\_ARG\_TYPE ファンクション

このファンクションは、遅延コールの引数のタイプを判定します。遅延リモート・プロシージャ・コール（RPC）パラメータのタイプが戻されます。

**関連項目：** Replication Manager での遅延トランザクションおよびエラー・トランザクションの表示の詳細は、Replication Manager のオンライン・ヘルプを参照してください。

構文

```
DEMS_DEFER_QUERY.GET_ARG_TYPE (  
    callno          IN    NUMBER,  
    arg_no          IN    NUMBER,  
    deferred_tran_id IN    VARCHAR2)  
RETURN NUMBER;
```

## パラメータ

表 8-14 GET\_ARG\_TYPE ファンクションのパラメータ

パラメータ	説明
callno	遅延リモート・プロシージャ・コールの DEFCALL ビューから渡される ID 番号です。
arg_no	タイプを判定するコールへの引数の数値的な位置です。プロシージャへの最初の引数は、1 の位置にあります。
deferred_tran_id	遅延トランザクションの識別子です。

## 例外

表 8-15 GET\_ARG\_TYPE ファンクションの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。

## 戻り値

表 8-16 GET\_ARG\_TYPE ファンクションの戻り値

戻り値	対応するデータ型
1	VARCHAR2
2	NUMBER
11	ROWID
12	DATE
23	RAW
96	CHAR
112	CLOB
113	BLOB

## GET\_CALL\_ARGS プロシージャ

このプロシージャは、指定されたコールの様々な引数のテキスト・バージョンを戻します。テキスト・バージョンは最初の 2000 バイトに制限されます。

構文

```
DBMS_DEFER_QUERY.GET_CALL_ARGS (  
    callno      IN  NUMBER,  
    startarg    IN  NUMBER := 1,  
    argcnt      IN  NUMBER,  
    argsize     IN  NUMBER,  
    tran_id     IN  VARCHAR2,  
    date_fmt    IN  VARCHAR2,  
    types       OUT TYPE_ARG,  
    forms       OUT TYPE_ARG,  
    vals        OUT VAL_ARG);
```

パラメータ

表 8-17 GET\_CALL\_ARGS プロシージャのパラメータ

パラメータ	説明
callno	遅延 RPC の DEFCALL ビューからの ID 番号です。
startarg	記述される最初の引数の数値的な位置です。
argcnt	コール内の引数の数です。
argsize	戻される引数の最大サイズです。
tran_id	遅延トランザクションの識別子です。
date_fmt	戻される日付の形式です。
types	引数のタイプを含む配列です。
forms	引数の文字セット形式を含む配列です。
vals	テキスト形式の引数値を含む配列です。

例外

表 8-18 GET\_CALL\_ARGS プロシージャの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。

GET\_datatype\_ARG ファンクション

このファンクションは、遅延コールの引数値を判定します。

**関連項目：** Replication Manager での遅延トランザクションおよびエラー・トランザクションの表示の詳細は、Replication Manager のオンライン・ヘルプを参照してください。

## 構文

取得する引数値のタイプに応じて、該当するファンクションの構文は次のとおりです。これらのファンクションでは、指定された引数の値が戻されます。

```
DBMS_DEFER_QUERY.GET_datatype_ARG (  
    callno             IN    NUMBER,  
    arg_no             IN    NUMBER,  
    deferred_tran_id   IN    VARCHAR2 DEFAULT NULL)  
RETURN datatype;
```

*datatype* は次のとおりです。

```
{ NUMBER  
| VARCHAR2  
| CHAR  
| DATE  
| RAW  
| ROWID  
| BLOB  
| CLOB  
| NCLOB  
| NCHAR  
| NVARCHAR2 }
```

## パラメータ

**表 8-19 GET\_datatype\_ARG ファンクションのパラメータ**

パラメータ	説明
callno	遅延リモート・プロシージャ・コールの DEFCALL ビューから渡される ID 番号です。
arg_no	値を判定するコールへの引数の数値的な位置です。プロシージャへの最初の引数は、1 の位置にあります。
deferred_tran_id	遅延トランザクションの識別子です。GET_ARG_TYPE に渡された、直前のトランザクション識別子が設定されます。デフォルトでは NULL です。

例外

表 8-20 GET\_datatype\_ARG ファンクションの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。
ORA-26564	この位置にある引数は、指定されたタイプではありません。

# DBMS\_DEFER\_SYS パッケージ

## サブプログラムの概要

**表 8-21 DBMS\_DEFER\_SYS パッケージのサブプログラム (1 / 2 ページ)**

サブプログラム	説明
<a href="#">ADD_DEFAULT_DEST プロシージャ</a> (8-16 ページ)	接続先データベースを DEFDEFAULTDEST ビューに追加します。
<a href="#">DELETE_DEFAULT_DEST プロシージャ</a> (8-17 ページ)	接続先データベースを DEFDEFAULTDEST ビューから削除します。
<a href="#">DELETE_DEF_DESTINATION プロシージャ</a> (8-17 ページ)	接続先データベースを DEFSCCHEDULE ビューから削除します。
<a href="#">DELETE_ERROR プロシージャ</a> (8-18 ページ)	トランザクションを DEFERROR ビューから削除します。
<a href="#">DELETE_TRAN プロシージャ</a> (8-18 ページ)	トランザクションを DEFTRANDEST ビューから削除します。
<a href="#">DISABLED ファンクション</a> (8-19 ページ)	カレント・サイトから指定されたサイトへの、遅延トランザクション・キューの伝播が有効になっているかどうかを判定します。
<a href="#">EXCLUDE_PUSH プロシージャ</a> (8-20 ページ)	遅延トランザクション PUSH を防ぐ排他ロックを取得します。
<a href="#">EXECUTE_ERROR プロシージャ</a> (8-21 ページ)	正常に終了しなかった遅延トランザクションを、トランザクションの元の受信者のセキュリティ・コンテキストで再実行します。
<a href="#">EXECUTE_ERROR_AS_USER プロシージャ</a> (8-21 ページ)	正常に終了しなかった遅延トランザクションを、このプロシージャを実行するユーザーのセキュリティ・コンテキストで再実行します。
<a href="#">PURGE ファンクション</a> (8-22 ページ)	カレント・マスター・サイトまたはスナップショット・サイトの遅延トランザクション・キューから、プッシュされたトランザクションをバージします。
<a href="#">PUSH ファンクション</a> (8-24 ページ)	カレント・マスター・サイトまたはスナップショット・サイトの遅延リモート・プロシージャ・コール・キューを、強制的に他のマスター・サイトへプッシュします。
<a href="#">REGISTER_PROPAGATOR プロシージャ</a> (8-27 ページ)	指定されたユーザーを、ローカル・データベースのプロパゲータとして登録します。
<a href="#">SCHEDULE_PURGE プロシージャ</a> (8-27 ページ)	カレント・マスター・サイトまたはスナップショット・サイトの遅延トランザクション・キューから、プッシュされたトランザクションをバージするジョブをスケジュールします。

表 8-21 DBMS\_DEFER\_SYS パッケージのサブプログラム (2 / 2 ページ)

サブプログラム	説明
<a href="#">SCHEDULE_PUSH</a> プロシージャ (8-29 ページ)	遅延トランザクション・キューをリモート・マスターにプッシュするジョブをスケジュールします。
<a href="#">SET_DISABLED</a> プロシージャ (8-30 ページ)	カレント・サイトから、指定された接続先サイトへの遅延トランザクション・キューの伝播を、使用禁止または使用可能にします。
<a href="#">UNREGISTER_PROPAGATOR</a> プロシージャ (8-31 ページ)	プロパゲータとしてのユーザーの登録を、ローカル・データベースから削除します。
<a href="#">UNSCHEDULE_PURGE</a> プロシージャ (8-32 ページ)	スナップショット・サイトまたはマスター・サイトの遅延トランザクション・キューからの、プッシュされたトランザクションの自動パージを中止します。
<a href="#">UNSCHEDULE_PUSH</a> プロシージャ (8-32 ページ)	スナップショット・サイトまたはマスター・サイトから他のマスター・サイトへの、遅延トランザクション・キューの自動プッシュを中止します。

## ADD\_DEFAULT\_DEST プロシージャ

このプロシージャは、接続先データベースを DEFDEFAULTDEST ビューに追加します。

### 構文

```
DBMS_DEFER_SYS.ADD_DEFAULT_DEST (  
    dblink      IN   VARCHAR2);
```

### パラメータ

表 8-22 ADD\_DEFAULT\_DEST プロシージャのパラメータ

パラメータ	説明
dblink	DEFDEFAULTDEST ビューに追加するノードの、完全修飾されたデータベース名です。

### 例外

表 8-23 ADD\_DEFAULT\_DEST プロシージャの例外

例外	説明
ORA-23352	指定された dblink は、デフォルトのリストにすでに存在していません。



## DELETE\_DEFAULT\_DEST プロシージャ

このプロシージャは、接続先データベースを DEFDEFAULTDEST ビューから削除します。

### 構文

```
DBMS_DEFER_SYS.DELETE_DEFAULT_DEST (  
    dblink      IN    VARCHAR2);
```

### パラメータ

表 8-24 DELETE\_DEFAULT\_DEST プロシージャのパラメータ

パラメータ	説明
dblink	DEFDEFAULTDEST ビューから削除するノードの、完全修飾されたデータベース名です。ビューでこの dblink がみつからない場合、削除作業は実行されません。

## DELETE\_DEF\_DESTINATION プロシージャ

このプロシージャは、接続先データベースを DEFSCHEDULE ビューから削除します。

### 構文

```
DBMS_DEFER_SYS.DELETE_DEF_DESTINATION (  
    destination IN    VARCHAR2,  
    force       IN    BOOLEAN := FALSE);
```

### パラメータ

表 8-25 DELETE\_DEF\_DESTINATION プロシージャのパラメータ

パラメータ	説明
destination	DEFSCHEDULE ビューから削除する接続先の完全修飾されたデータベース名です。ビューでこの接続先がみつからない場合、削除作業は実行されません。
force	値が TRUE に設定されていると、すべての安全チェックが無視され、接続先が削除されます。

## DELETE\_ERROR プロシージャ

このプロシージャは、DEFERROR ビューからトランザクションを削除します。

### 構文

```
DBMS_DEFER_SYS.DELETE_ERROR (
    deferred_tran_id      IN   VARCHAR2,
    destination           IN   VARCHAR2);
```

### パラメータ

表 8-26 DELETE\_ERROR プロシージャのパラメータ

パラメータ	説明
deferred_tran_id	DEFERROR ビューから削除する遅延トランザクションの、DEFERROR ビューから渡される ID 番号です。このパラメータが NULL の場合、他のパラメータの要件に合致するすべてのトランザクションが削除されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFERROR ビューから渡される、完全修飾されたデータベース名です。このパラメータが NULL の場合、他のパラメータの要件に合致するすべてのトランザクションが、DEFERROR ビューから削除されます。

## DELETE\_TRAN プロシージャ

このプロシージャは、DEFTRANDEST ビューからトランザクションを削除します。他の DEFTRANDEST または DEFERROR エントリがない場合、トランザクションは DEFTRAN ビューと DEFCALL ビューからも削除されます。

### 構文

```
DBMS_DEFER_SYS.DELETE_TRAN (
    deferred_tran_id      IN   VARCHAR2,
    destination           IN   VARCHAR2);
```

## パラメータ

表 8-27 DELETE\_TRAN プロシージャのパラメータ

パラメータ	説明
deferred_tran_id	削除する遅延トランザクションの DEFTRAN ビューから渡される ID 番号です。値が NULL の場合、他のパラメータの要件に合致するすべてのトランザクションが削除されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFTRANDEST ビューから渡される、完全修飾されたデータベース名です。値が NULL の場合、他のパラメータの要件に合致するすべてのトランザクションが削除されます。

## DISABLED ファンクション

このファンクションは、カレント・サイトから指定されたサイトへの、遅延トランザクション・キューの伝播が有効になっているかどうかを判定します。指定された接続先に対して遅延リモート・プロシージャ・コール（RPC）のキューが使用禁止になっている場合、DISABLED ファンクションによって TRUE が戻されます。

### 構文

```
DBMS_DEFER_SYS.DISABLED (  
    destination IN VARCHAR2)  
RETURN BOOLEAN;
```

## パラメータ

表 8-28 DISABLED ファンクションのパラメータ

パラメータ	説明
destination	伝播ステータスをチェックするノードの、完全修飾されたデータベース名です。

### 戻り値

表 8-29 DISABLED ファンクションの戻り値

値	説明
TRUE	カレント・サイトからこのサイトへの伝播は使用禁止になっています。
FALSE	カレント・サイトからこのサイトへの伝播は有効になっています。

例外

表 8-30 DISABLED ファンクションの例外

例外	説明
NO_DATA_FOUND	指定された destination が DEFSCHEDULE ビューにありません。

EXCLUDE\_PUSH プロシージャ

このファンクションは、遅延トランザクション PUSH を妨ぐ排他ロックを取得します（シリアルまたはパラレル）。このファンクションは、ロックの取得時にコミットを実行します。ロックは RELEASE\_ON\_COMMIT => TRUE で取得されるため、次のコミット後に、遅延トランザクション・キューのプッシュを再開できます。

構文

```
DBMS_DEFER_SYS.EXCLUDE_PUSH (  
    timeout    IN    INTEGER)  
RETURN INTEGER;
```

パラメータ

表 8-31 EXCLUDE\_PUSH ファンクションのパラメータ

パラメータ	説明
timeout	タイムアウトを秒で設定します。エラーが発生したか、または PUSH が行われているために、設定された期間内にロックが取得されない場合は、値 1 が戻されます。タイムアウトの値を DBMS_LOCK.MAXWAIT に設定すると、無期限に待機します。

戻り値

表 8-32 EXCLUDE\_PUSH ファンクションの戻り値

値	説明
0	ロックは正常に取得されました。
1	タイムアウトにより、ロックは取得されませんでした。
2	デッドロックにより、ロックは取得されませんでした。
4	ロックはすでに取得されています。

## EXECUTE\_ERROR プロシージャ

このプロシージャは、正常に終了しなかった遅延トランザクションを、トランザクションの元の受信者のセキュリティ・コンテキストで再実行します。

### 構文

```
DBMS_DEFER_SYS.EXECUTE_ERROR (
    deferred_tran_id IN    VARCHAR2,
    destination      IN    VARCHAR2);
```

### パラメータ

**表 8-33 EXECUTE\_ERROR プロシージャのパラメータ**

パラメータ	説明
deferred_tran_id	再実行する遅延トランザクションの DEFERROR ビューから渡される ID 番号です。これが NULL の場合、destination へのキューに入れているすべてのトランザクションが再実行されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFERROR ビューから渡される、完全修飾されたデータベース名です。NULL は指定できません。指定したデータベース名が完全修飾されていない場合や無効の場合、エラーは発生しません。

### 例外

**表 8-34 EXECUTE\_ERROR プロシージャの例外**

例外	説明
ORA-24275	NULL パラメータと NULL 以外のパラメータの無効な組合せが使用されています。
badparam	パラメータ値が欠落しているか、または無効です（たとえば、destination が NULL）。
missinguser	無効なユーザーです。

## EXECUTE\_ERROR\_AS\_USER プロシージャ

このプロシージャは、正常に終了しなかった遅延トランザクションを再実行します。各トランザクションは、接続しているユーザーのセキュリティ・コンテキストで実行されます。

### 構文

```
DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER (
    deferred_tran_id IN    VARCHAR2,
    destination      IN    VARCHAR2);
```

パラメータ

表 8-35 EXECUTE\_ERROR\_AS\_USER プロシージャのパラメータ

パラメータ	説明
deferred_tran_id	再実行する遅延トランザクションの DEFERROR ビューから渡される ID 番号です。これが NULL の場合、destination へのキューに入れているすべてのトランザクションが再実行されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFERROR ビューから渡される、完全修飾されたデータベース名です。NULL は指定できません。

例外

表 8-36 EXECUTE\_ERROR\_AS\_USER プロシージャの例外

例外	説明
ORA-24275	NULL パラメータと NULL 以外のパラメータの無効な組合せが使用されています。
badparam	パラメータ値が欠落しているか、または無効です（たとえば、destination が NULL）。
missinguser	無効なユーザーです。

PURGE ファンクション

このファンクションは、カレント・マスター・サイトまたはスナップショット・サイトの遅延トランザクション・キューから、プッシュされたトランザクションをパージします。

構文

```
DBMS_DEFER_SYS.PURGE (
  purge_method      IN  BINARY_INTEGER := purge_method_quick,
  rollback_segment  IN  VARCHAR2       := NULL,
  startup_seconds   IN  BINARY_INTEGER := 0,
  execution_seconds IN  BINARY_INTEGER := seconds_infinity,
  delay_seconds     IN  BINARY_INTEGER := 0,
  transaction_count IN  BINARY_INTEGER := transactions_infinity,
  write_trace       IN  BOOLEAN        := NULL);
RETURN BINARY_INTEGER;
```

## パラメータ

表 8-37 PURGE ファンクションのパラメータ

パラメータ	説明
purge_method	遅延トランザクション・キューのパージ方法を制御します。purge_method_quick はコストを低く抑えられますが、purge_method_precise の方が精度が高くなります。  purge_method_quick を使用する場合、正常にプッシュされた遅延トランザクションおよび遅延プロシージャ・コールが、パージされるまで、それぞれ DEFTRAN および DEFCALL データ・ディクショナリ・ビューに保持されることがあります。詳細は、8-24 ページの「 <a href="#">使用にあたっての注意</a> 」を参照してください。
rollback_segment	パージに使用するロールバック・セグメントの名前、またはデフォルトの NULL です。
startup_seconds	同じ遅延トランザクション・キューの前のパージを待つ最大秒数です。
execution_seconds	値が >0 の場合、指定された実秒数後に、パージが完全に停止します。
delay_seconds	delay_seconds の間、遅延トランザクション・キューにパージするトランザクションがなくなると、パージが完全に停止します。
transaction_count	値が >0 の場合は、transaction_count の数だけトランザクションがパージされると、完全にシャットダウンされます。
write_trace	値が TRUE に設定されている場合、PURGE ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。

## 戻り値

表 8-38 Purge ファンクションの戻り値

値	説明
0	OK です。delay_seconds の経過後に終了しました。
1	スタート時にロック・タイムアウトにより終了しました。
2	execution_seconds の超過により終了しました。
3	transaction_count の超過により終了しました。
5	エラー発生後に終了しました。

例外

表 8-39 PURGE ファンクションの例外

例外	説明
argoutofrange	パラメータ値が有効範囲外です。
executiondisabled	ページの実行が使用禁止になっています。
defererror	内部エラーが発生しました。

使用にあたっての注意

DBMS\_DEFER\_SYS.PURGE ファンクションの `purge_method` パラメータに `purge_method_quick` を指定すると、遅延トランザクションおよび遅延プロシージャ・コールが正常にプッシュされた後、それぞれ `DEFCALL` および `DEFTRAN` データ・ディクショナリ・ビューに保持されることがあります。これは、複数のデータベース・リンクがあるレプリケーション環境で、1つのデータベース・リンクのみに対してプッシュが実行された場合に発生します。

遅延トランザクションおよび遅延プロシージャ・コールをページするには、次のいずれかを実行します。

- `purge_method` パラメータに、`purge_method_quick` ではなく `purge_method_precise` を指定します。`purge_method_precise` を使用するとコストは増大しますが、遅延トランザクションおよびプロシージャ・コールは、正常にプッシュされた後で確実にページされます。
- `purge_method` パラメータに `purge_method_quick` を指定して、遅延トランザクションをすべてのデータベース・リンクにプッシュします。遅延トランザクションおよび遅延プロシージャ・コールは、最後のデータベース・リンクに正常にプッシュされた時点でページされます。

PUSH ファンクション

このファンクションは、カレント・マスター・サイトまたはスナップショット・サイトの遅延リモート・プロシージャ・コール (RPC) ・キューを、強制的に他のマスター・サイトへプッシュ (伝播) します。シリアル伝播またはパラレル伝播のいずれかが使用されます。

構文

```
DBMS_DEFER_SYS.PUSH (  
  destination          IN  VARCHAR2,  
  parallelism          IN  BINARY_INTEGER := 0,  
  heap_size            IN  BINARY_INTEGER := 0,  
  stop_on_error        IN  BOOLEAN       := FALSE,  
  write_trace          IN  BOOLEAN       := FALSE,  
  startup_seconds      IN  BINARY_INTEGER := 0,
```



```

execution_seconds    IN  BINARY_INTEGER := seconds_infinity,
delay_seconds        IN  BINARY_INTEGER := 0,
transaction_count    IN  BINARY_INTEGER := transactions_infinity,
delivery_order_limit IN  NUMBER          := delivery_order_infinity)
RETURN BINARY_INTEGER;

```

## パラメータ

表 8-40 PUSH ファンクションのパラメータ (1 / 2 ページ)

パラメータ	説明
destination	変更を転送するマスターの完全修飾されたデータベース名です。
parallelism	0 の場合はシリアル伝播、 $n > 1$ の場合は $n$ パラレル・サーバー・プロセスのパラレル伝播、1 の場合は 1 パラレル・サーバー・プロセスのみのパラレル伝播です。
heap_size	パラレル伝播のスケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。オラクル社カスタマ・サポート・センターから指示された場合を除き、パラメータを設定しないでください。
stop_on_error	デフォルト設定の FALSE の場合、競合などのエラーが発生しても処理が続行されます。値が TRUE の場合は、接続先サイトでのエラーの発生が最初に指摘されたときに（可能な場合は完全に）シャットダウンされます。
write_trace	値が TRUE の場合、ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。
startup_seconds	同じ接続先への前のプッシュを待つ最大秒数です。
execution_seconds	<p>値が <math>&gt; 0</math> の場合は、指定された実秒数後に、プッシュが完全に停止します。transaction_count と execution_seconds が 0（デフォルト）の場合、キューにトランザクションがなくなるまで、トランザクションが実行されます。</p> <p>execution_seconds パラメータは、操作が開始可能な期間のみを制御します。リモート・サイトのトランザクションに必要な時間は含まれません。したがって、execution_seconds パラメータは、リモート・サイトへのトランザクションの伝播を正確に制御して停止するためのものではありません。正確に制御する必要がある場合は、transaction_count または delivery_order パラメータを使用します。</p>
delay_seconds	キューが空でも、指定された秒数が経過するまでプッシュが終了しません。タイトなループから PUSH がコールされるとき、実行オーバーヘッドを縮小するのに便利です。

表 8-40 PUSH ファンクションのパラメータ (2 / 2 ページ)

パラメータ	説明
transaction_count	値が > 0 の場合は、トランザクションの最大数のプッシュ後に停止します。transaction_count と execution_seconds が 0 (デフォルト) の場合、キューにプッシュする必要のあるトランザクションがなくなるまで、トランザクションが実行されます。
delivery_order_limit	delivery_order >= delivery_order_limit のトランザクションのプッシュ前に、実行が完全に停止します。

戻り値

表 8-41 PUSH ファンクションの戻り値

値	説明
0	OK です。delay_seconds の経過後に終了しました。
1	スタート時にロック・タイムアウトにより終了しました。
2	execution_seconds の超過により終了しました。
3	transaction_count の超過により終了しました。
4	delivery_order_limit の超過により終了しました。
5	エラー発生後に終了しました。

例外

表 8-42 PUSH ファンクションの例外

例外	説明
deferror incompleteparallelpush	シリアル伝播を実行するには、パラレル伝播を完全にシャットダウンする必要があります。
executiondisabled	遅延 RPC の実行が、接続先で使用禁止になっています。
crt_err_err	DEFERROR にエントリを作成中、エラーが発生しました。
deferred_rpc_qiesce	オブジェクト・グループのレプリケーション・アクティビティが中断されました。
commfailure	遅延 RPC で通信エラーが発生しました。
missingpropator	プロパゲータが存在しません。

## REGISTER\_PROPAGATOR プロシージャ

このプロシージャは、指定されたユーザーをローカル・データベースのプロパゲータとして登録します。指定されたユーザーには、CREATE SESSION、CREATE PROCEDURE、CREATE DATABASE LINK および EXECUTE ANY PROCEDURE 権限も与えられます。このため、指定されたユーザーはラッパーを作成できます。

### 構文

```
DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
    username IN VARCHAR2);
```

### パラメータ

表 8-43 REGISTER\_PROPAGATOR プロシージャのパラメータ

パラメータ	説明
username	ユーザーの名前です。

### 例外

表 8-44 REGISTER\_PROPAGATOR プロシージャの例外

例外	説明
missinguser	指定されたユーザーは存在しません。
alreadypropagator	指定されたユーザーはすでにプロパゲータとして登録されています。
duplicatepropagator	他のプロパゲータがすでに登録されています。

## SCHEDULE\_PURGE プロシージャ

このプロシージャは、カレント・マスター・サイトまたはスナップショット・サイトの遅延トランザクション・キューから、プッシュされたトランザクションをパージするジョブをスケジュールします。1つのパージ・ジョブをスケジュールする必要があります。

### 構文

```
DBMS_DEFER_SYS.SCHEDULE_PURGE (
    interval          IN  VARCHAR2,
    next_date         IN  DATE,
    reset             IN  BOOLEAN          := NULL,
    purge_method      IN  BINARY_INTEGER := NULL,
    rollback_segment  IN  VARCHAR2        := NULL,
    startup_seconds   IN  BINARY_INTEGER := NULL,
```

```
execution_seconds    IN  BINARY_INTEGER := NULL,  
delay_seconds        IN  BINARY_INTEGER := NULL,  
transaction_count    IN  BINARY_INTEGER := NULL,  
write_trace          IN  BOOLEAN         := NULL);
```

パラメータ

表 8-45 SCHEDULE\_PURGE プロシージャのパラメータ

パラメータ	説明
interval	次のパージの時間を計算するファンクションを指定できます。値は DEFSCHEDULE ビューの interval フィールドに格納され、このビューの next_date フィールドの計算に使用します。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が設定されていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、next_date に値を設定する必要があります。
next_date	サイトのキューから、プッシュされたトランザクションをパージする時間を指定できます。値は DEFSCHEDULE ビューの next_date フィールドに格納されます。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が入っていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、interval に値を設定する必要があります。
reset	値を TRUE に設定すると、LAST_TXN_COUNT、LAST_ERROR および LAST_MSG が NULL にリセットされます。
purge_method	遅延トランザクション・キューのパージ方法を制御します。purge_method_quick はコストを低く抑えられますが、purge_method_precise の方が精度が高くなります。
rollback_segment	パージに使用するロールバック・セグメントの名前、またはデフォルトの NULL です。
startup_seconds	同じ遅延トランザクション・キューの前のパージを待つ最大秒数です。
execution_seconds	値が >0 の場合、指定された実秒数後に、パージが完全に停止します。
delay_seconds	delay_seconds の間、遅延トランザクション・キューにパージするトランザクションがなくなると、パージが完全に停止します。
transaction_count	値が >0 の場合は、transaction_count の数だけトランザクションがパージされると、完全にシャットダウンされます。
write_trace	値が TRUE に設定されている場合、PURGE ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。

## SCHEDULE\_PUSH プロシージャ

このプロシージャは、遅延トランザクション・キューをリモート・マスターの接続先にプッシュするジョブをスケジュールします。このプロシージャでは COMMIT が行われます。

### 構文

```
DBMS_DEFER_SYS.SCHEDULE_PUSH (
    destination      IN  VARCHAR2,
    interval         IN  VARCHAR2,
    next_date        IN  DATE,
    reset            IN  BOOLEAN      := FALSE,
    parallelism      IN  BINARY_INTEGER := NULL,
    heap_size        IN  BINARY_INTEGER := NULL,
    stop_on_error    IN  BOOLEAN      := NULL,
    write_trace      IN  BOOLEAN      := NULL,
    startup_seconds  IN  BINARY_INTEGER := NULL,
    execution_seconds IN  BINARY_INTEGER := NULL,
    delay_seconds    IN  BINARY_INTEGER := NULL,
    transaction_count IN  BINARY_INTEGER := NULL);
```

### パラメータ

表 8-46 SCHEDULE\_PUSH プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
destination	変更を転送するマスターの完全修飾されたデータベース名です。
interval	次のプッシュの時間を計算するファンクションを指定できます。値は DEFSCHEDULE ビューの interval フィールドに格納され、このビューの next_date フィールドの計算に使用します。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が設定されていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、next_date に値を設定する必要があります。
next_date	遅延トランザクションを接続先マスター・サイトにプッシュする時間を指定できます。値は DEFSCHEDULE ビューの next_date フィールドに格納されます。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が設定されていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、interval に値を設定する必要があります。
reset	値を TRUE に設定すると、LAST_TXN_COUNT、LST_ERROR および LAST_MSG が NULL にリセットされます。

表 8-46 SCHEDULE\_PUSH プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
parallelism	0 の場合はシリアル伝播、 $n > 1$ の場合は $n$ パラレル・サーバー・プロセスの平行伝播、1 の場合は 1 パラレル・サーバー・プロセスのみの平行伝播です。
heap_size	平行伝播のスケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。オラクル社カスタマ・サポート・センターから指示された場合を除き、パラメータを設定しないでください。
stop_on_error	デフォルト設定の FALSE の場合、競合などのエラーが発生しても処理が続行されます。値が TRUE の場合は、接続先サイトでのエラーの発生が最初に指摘されたときに（可能な場合は完全に）シャットダウンされます。
write_trace	値が TRUE の場合、ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。
startup_seconds	同じ接続先への前のプッシュを待つ最大秒数です。
execution_seconds	値が $> 0$ の場合、指定された実秒数後に、実行が完全に停止します。transaction_count と execution_seconds がゼロ（デフォルト）の場合、キューにトランザクションがなくなるまで、トランザクションが実行されます。
delay_seconds	キューが空でも、指定された秒数が経過するまでプッシュが終了しません。タイトなループから PUSH がコールされる時、実行オーバーヘッドを縮小するのに便利です。
transaction_count	値が $> 0$ の場合は、トランザクションの最大数のプッシュ後に停止します。transaction_count と execution_seconds がゼロ（デフォルト）の場合、キューにプッシュする必要のあるトランザクションがなくなるまで、トランザクションが実行されます。

SET\_DISABLED プロシージャ

カレント・サイトから指定された接続先サイトへの、遅延トランザクション・キューの伝播が使用禁止あるいは使用可能にされます。disabled パラメータが TRUE の場合、指定された接続先への伝播が使用禁止になり、PUSH が呼び出されても遅延リモート・プロシージャ・コール (RPC) キューがプッシュされなくなります。SET\_DISABLED は、指定された接続先へキューをすでにプッシュしているセッションには適用されますが、DBMS\_DEFER でキューに追加されているセッションには適用されません。

disabled パラメータが FALSE の場合、指定された接続先への伝播が有効になり、キューはプッシュされませんが、PUSH の呼び出しによる指定された接続先へのキューのプッシュが可能になります。disabled パラメータが TRUE か FALSE かに関係なく、設定が他のセッションにも適用されるようにするには、COMMIT が必要です。

## 構文

```
DBMS_DEFER_SYS.SET_DISABLED (
    destination IN VARCHAR2,
    disabled    IN BOOLEAN := TRUE);
```

## パラメータ

**表 8-47 SET\_DISABLED プロシージャのパラメータ**

パラメータ	説明
destination	伝播ステータスを変更するノードの、完全修飾されたデータベース名です。
disabled	デフォルトでは、このパラメータによって、カレント・サイトから指定された接続先への遅延トランザクション・キューの伝播が使用禁止になります。値を FALSE に設定すると、伝播が有効になります。

## 例外

**表 8-48 SET\_DISABLED プロシージャの例外**

例外	説明
NO_DATA_FOUND	指定された destination の DEFSCHEDULE ビューには、エントリが見つかりませんでした。

## UNREGISTER\_PROPAGATOR プロシージャ

プロパゲータとしてのユーザーの登録を、ローカル・データベースから削除します。このプロシージャでは、次の処理が行われます。

- 指定されたプロパゲータが DEFPROPAGATOR から削除されます。
- REGISTER\_PROPAGATOR によってユーザーに与えられた権限が取り消されます。別個に与えられた同一の権限も取り消されます。
- 指定されたプロパゲータのスキーマ内に生成されたラッパーが削除されます。削除されたラッパーには、レプリケーション・カタログで削除済のマークが付けられます。

## 構文

```
DBMS_DEFER_SYS.UNREGISTER_PROPAGATOR (
    username IN VARCHAR2
    timeout  IN INTEGER DEFAULT DBMS_LOCK.MAXWAIT);
```

パラメータ

表 8-49 UNREGISTER\_PROPAGATOR プロシージャのパラメータ

パラメータ	説明
username	プロパゲータ・ユーザーの名前です。
timeout	タイムアウトを秒で設定します。プロパゲータが稼働中の場合、プロシージャはタイムアウトで指定された時間が経過するまで待機します。デフォルトは DBMS_LOCK.MAXWAIT です。

例外

表 8-50 UNREGISTER\_PROPAGATOR プロシージャの例外

パラメータ	説明
missingpropagator	指定されたユーザーはプロパゲータではありません。
propagator_inuse	プロパゲータが稼働中のため、登録を削除できません。後で実行し直してください。

UNSCCHEDULE\_PURGE プロシージャ

このプロシージャは、スナップショット・サイトまたはマスター・サイトの遅延トランザクション・キューからの、プッシュされたトランザクションの自動パージを中止します。

構文

```
DBMS_DEFER_SYS.UNSCHEDULE_PURGE (;
```

パラメータ

なし

UNSCCHEDULE\_PUSH プロシージャ

このプロシージャは、スナップショット・サイトまたはマスター・サイトから他のマスター・サイトへの、遅延トランザクション・キューの自動プッシュを中止します。

構文

```
DBMS_DEFER_SYS.UNSCHEDULE_PUSH (
    dblink IN VARCHAR2);
```



## パラメータ

**表 8-51 UNSCHEDULE\_PUSH プロシージャのパラメータ**

パラメータ	説明
dblink	遅延リモート・プロシージャ・コールの定期的な実行スケジュールを変更するマスター・データベース・サイトへの、完全修飾されたパス名です。

**表 8-52 UNSCHEDULE\_PUSH プロシージャの例外**

例外	説明
NO_DATA_FOUND	指定された dblink の DEFSCHEDULE ビューには、エントリが見つかりませんでした。

# DBMS\_OFFLINE\_OG パッケージ

## サブプログラムの概要

表 8-53 DBMS\_OFFLINE\_OG パッケージのサブプログラム

サブプログラム	説明
<a href="#">BEGIN_INSTANTIATION プロシージャ</a> (8-34 ページ)	レプリケート・マスター・グループのオフライン・インスタンスエーションを開始します。
<a href="#">BEGIN_LOAD プロシージャ</a> (8-35 ページ)	オフライン・インスタンスエーションの一環として行われる新しいマスターへのデータのインポート中、トリガーを使用禁止にします。
<a href="#">END_INSTANTIATION プロシージャ</a> (8-37 ページ)	レプリケート・マスター・グループのオフライン・インスタンスエーションを終了します。
<a href="#">END_LOAD プロシージャ</a> (8-38 ページ)	オフライン・インスタンスエーションの一環として行われる新しいマスターへのデータのインポート後、トリガーを再び有効にします。
<a href="#">RESUME_SUBSET_OF_MASTERS プロシージャ</a> (8-39 ページ)	レプリケート・マスター・グループのオフライン・インスタンスエーションが行われている間、新しいサイトを除くすべての既存のサイトで、レプリケーション・アクティビティを再開します。

## BEGIN\_INSTANTIATION プロシージャ

このプロシージャは、レプリケート・マスター・グループのオフライン・インスタンスエーションを開始します。このプロシージャは、マスター定義サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

このプロシージャを、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンスエーションに使用）や、DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャ（配置テンプレートのインスタンスエーションに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

### 構文

```
DBMS_OFFLINE_OG.BEGIN_INSTANTIATION (  
    gname      IN   VARCHAR2,  
    new_site   IN   VARCHAR2  
    fname      IN   VARCHAR2);
```

## パラメータ

**表 8-54 BEGIN\_INSTANTIATION プロシージャのパラメータ**

パラメータ	説明
gname	新しいサイトにレプリケートするオブジェクト・グループの名前です。
new_site	オブジェクト・グループをレプリケートする新しいサイトの、完全修飾されたデータベース名です。
fname	このパラメータは内部で使われます。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

## 例外

**表 8-55 BEGIN\_INSTANTIATION プロシージャの例外**

例外	説明
badargument	オブジェクト・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
dbms_repcat. nonmasterdef	このプロシージャは、マスター定義サイトからコールする必要があります。
sitealreadyexists	指定されたサイトは、すでにこのオブジェクト・グループのマスター・サイトになっています。
wrongstate	マスター定義サイトのステータスは、QUIESCED である必要があります。
dbms_repcat. missingrepgroup	gname は、レプリケート・マスター・グループとしては存在していません。
dbms_repcat.missing_ flavor	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。

## BEGIN\_LOAD プロシージャ

このプロシージャは、オフライン・インスタンスエーションの一環として行われる新しいマスター・サイトへのデータのインポート中、トリガーを使用禁止にします。このプロシージャは、新しいマスター・サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンス化に使用します。

このプロシージャを、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンス化に使用）や、DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

構文

```
DBMS_OFFLINE_OG.BEGIN_LOAD (  
    gname      IN   VARCHAR2,  
    new_site   IN   VARCHAR2);
```

パラメータ

表 8-56 BEGIN\_LOAD プロシージャのパラメータ

パラメータ	説明
gname	メンバーをインポートするオブジェクト・グループの名前です。
new_site	オブジェクト・グループ・メンバーをインポートする新しいサイトの、完全修飾されたデータベース名です。

例外

表 8-57 BEGIN\_LOAD プロシージャの例外

例外	説明
badargument	オブジェクト・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
wrongsite	このプロシージャは、新しいマスター・サイトからコールする必要があります。
unknownsite	指定されたサイトは、オブジェクト・グループで認識されません。
wrongstate	新しいマスター・サイトのステータスは、QUIESCED である必要があります。
dbms_repcat. missingrepgroup	gname は、レプリケート・マスター・グループとしては存在していません。

## END\_INSTANTIATION プロシージャ

このプロシージャは、レプリケート・マスター・グループのオフライン・インスタンスエーションを終了します。このプロシージャは、マスター定義サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

このプロシージャを、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンスエーションに使用）や、DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャ（配置テンプレートのインスタンスエーションに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

### 構文

```
DBMS_OFFLINE_OG.END_INSTANTIATION (  
    gname      IN  VARCHAR2,  
    new_site   IN  VARCHAR2);
```

### パラメータ

表 8-58 END\_INSTANTIATION プロシージャのパラメータ

パラメータ	説明
gname	新しいサイトにレプリケートするオブジェクト・グループの名前です。
new_site	オブジェクト・グループをレプリケートする新しいサイトの、完全修飾されたデータベース名です。

例外

表 8-59 END\_INSTANTIATION プロシージャの例外

例外	説明
badargument	オブジェクト・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
dbms_repcat. nonmasterdef	このプロシージャは、マスター定義サイトからコールする必要があります。
unknownsite	指定されたサイトは、オブジェクト・グループで認識されません。
wrongstate	マスター定義サイトのステータスは、QUIESCED である必要があります。
dbms_repcat. missingrepgroup	gname は、レプリケート・マスター・グループとしては存在していません。

END\_LOAD プロシージャ

このプロシージャは、オフライン・インスタンスエーションの一環として行われる新しいマスターへのデータのインポート後、トリガーを再び有効にします。このプロシージャは、新しいマスター・サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

このプロシージャを、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンスエーションに使用）や、DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャ（配置テンプレートのインスタンスエーションに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

構文

```
DBMS_OFFLINE_OG.END_LOAD (  
  gname      IN   VARCHAR2,  
  new_site   IN   VARCHAR2  
  fname      IN   VARCHAR2);
```

## パラメータ

**表 8-60 END\_LOAD プロシージャのパラメータ**

パラメータ	説明
gname	メンバーのインポートを終了したオブジェクト・グループの名前です。
new_site	オブジェクト・グループ・メンバーをインポートした新しいサイトの、完全修飾されたデータベース名です。
fname	このパラメータは内部で使います。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

## 例外

**表 8-61 END\_LOAD プロシージャの例外**

例外	説明
badargument	オブジェクト・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
wrongsite	このプロシージャは、新しいマスター・サイトからコールする必要があります。
unknownsite	指定されたサイトは、オブジェクト・グループで認識されません。
wrongstate	新しいマスター・サイトのステータスは、QUIESCED である必要があります。
dbms_repcat. missingrepgroup	gname は、レプリケート・マスター・グループとしては存在していません。
dbms_repcat.flavor_ noobject	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。
dbms_repcat.flavor_ contains	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。

## RESUME\_SUBSET\_OF\_MASTERS プロシージャ

このプロシージャは、レプリケート・マスター・グループのオフライン・インスタンスションの間、新しいサイトを除くすべての既存のサイトで、レプリケーション・アクティビティを再開します。このプロシージャは、マスター定義サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンス化に使用します。

このプロシージャを、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンス化に使用）や、DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

構文

```
DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS (  
  gname      IN  VARCHAR2,  
  new_site   IN  VARCHAR2  
  override   IN  BOOLEAN := FALSE);
```

パラメータ

表 8-62 RESUME\_SUBSET\_OF\_MASTERS プロシージャのパラメータ

パラメータ	説明
gname	新しいサイトにレプリケートするオブジェクト・グループの名前です。
new_site	オブジェクト・グループをレプリケートする新しいサイトの、完全修飾されたデータベース名です。
override	値が TRUE の場合は、保留になっている RepCat 管理要求が無視され、各マスターで通常のレプリケーション・アクティビティがリストアされます。override パラメータは、緊急の場合のみ TRUE に設定します。  値が FALSE の場合は、各マスターで通常のレプリケーション・アクティビティがリストアされます。ただし、そのマスターで gname に対する RepCat 管理要求が保留されていない場合に限られます。



例外

表 8-63 RESUME\_SUBSET\_OF\_MASTERS プロシージャの例外

例外	説明
badargument	オブジェクト・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
dbms_repcat. nonmasterdef	このプロシージャは、マスター定義サイトからコールする必要があります。
unknownsite	指定されたサイトは、オブジェクト・グループで認識されません。
wrongstate	マスター定義サイトのステータスは、QUIESCED である必要があります。
dbms_repcat. missingrepgroup	gname は、レプリケート・マスター・グループとしては存在していません。

# DBMS\_OFFLINE\_SNAPSHOT パッケージ

## サブプログラムの概要

表 8-64 DBMS\_OFFLINE\_SNAPSHOT パッケージのサブプログラム

サブプログラム	説明
<a href="#">BEGIN_LOAD プロシージャ</a> (8-42 ページ)	オフライン・インスタンスエーションの一環として行われる新しいスナップショットのインポート用に、スナップショット・サイトを用意します。
<a href="#">END_LOAD プロシージャ</a> (8-43 ページ)	スナップショットのオフライン・インスタンスエーションを終了します。

## BEGIN\_LOAD プロシージャ

このプロシージャは、オフライン・インスタンスエーションの一環として行われる新しいスナップショットのインポート用に、スナップショット・サイトを用意します。このプロシージャは、新しいスナップショットのスナップショット・サイトからコールする必要があります。

**注意：** このプロシージャは、スナップショットのオフライン・インスタンスエーションに使用します。

このプロシージャを、DBMS\_OFFLINE\_OG パッケージのプロシージャ（マスター表のオフライン・インスタンスエーションに使用）や、DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャ（配置テンプレートのインスタンスエーションに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

## 構文

```
DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (  
  gname          IN   VARCHAR2,  
  sname          IN   VARCHAR2,  
  master_site    IN   VARCHAR2,  
  snapshot_otype IN   VARCHAR2,  
  storage_c      IN   VARCHAR2 := '',  
  comment        IN   VARCHAR2 := '',  
  min_communication IN  BOOLEAN := TRUE);
```

## パラメータ

**表 8-65 BEGIN\_LOAD プロシージャのパラメータ**

パラメータ	説明
gname	オフライン・インスタンスエーションで作成するスナップショットの、オブジェクト・グループの名前です。
sname	新しいスナップショットのスキーマの名前です。
master_site	スナップショットのマスター・サイトの、完全修飾されたデータベース名です。
snapshot_ename	マスター・サイトで作成される一時スナップショットの名前です。
storage_c	スナップショット・サイトでの新しいスナップショットの作成に使用する記憶領域オプションです。
comment	ユーザーのコメントです。
min_communication	値が TRUE の場合、UPDATE 文で列が変更されるときに限り、更新トリガーで列の新しい値が送られます。また、TRUE の場合、列がキー列、または変更された列グループ内の列のときに限り、更新トリガーで列の古い値が送られます。

## 例外

**表 8-66 BEGIN\_LOAD プロシージャの例外**

例外	説明
badargument	オブジェクト・グループ、スキーマ、マスター・サイトまたはスナップショットの名前が、NULL または空の文字列です。
dbms_repcat. missingrepgroup	gname は、レプリケート・マスター・グループとしては存在していません。
missingremotesnap	指定されたマスター・サイトで指定されたスナップショットが見つかりませんでした。
dbms_repcat. missingschema	指定されたスキーマは存在しません。
snaptabmismatch	マスター・サイトとスナップショット・サイトのベース表名が一致しません。

## END\_LOAD プロシージャ

このプロシージャは、スナップショットのオフライン・インスタンスエーションを終了します。このプロシージャは、新しいスナップショットのスナップショット・サイトからコールする必要があります。

**注意：** このプロシージャは、スナップショットのオフライン・インスタンスに使用します。

このプロシージャを、DBMS\_OFFLINE\_OG パッケージのプロシージャ（マスター表のオフライン・インスタンスに使用）や、DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャ（配置テンプレートのインスタンスに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

構文

```
DBMS_OFFLINE_SNAPSHOT.END_LOAD (  
  gname          IN  VARCHAR2,  
  sname          IN  VARCHAR2,  
  snapshot_ename IN  VARCHAR2);
```

パラメータ

表 8-67 END\_LOAD プロシージャのパラメータ

パラメータ	説明
gname	オフライン・インスタンスで作成するスナップショットの、オブジェクト・グループの名前です。
sname	新しいスナップショットのスキーマの名前です。
snapshot_ename	スナップショットの名前です。

例外

表 8-68 END\_LOAD プロシージャの例外

例外	説明
badargument	オブジェクト・グループ、スキーマまたはスナップショットの名前が、NULL または空の文字列です。
dbms_repcat. missingrepgroup	gname は、レプリケート・マスター・グループとしては存在していません。
dbms_repcat. nonsnapshot	このプロシージャは、スナップショット・サイトからコールする必要があります。

# DBMS\_RECTIFIER\_DIFF パッケージ

## サブプログラムの概要

表 8-69 DBMS\_RECTIFIER\_DIFF パッケージのサブプログラム

サブプログラム	説明
<a href="#">DIFFERENCES プロシージャ</a> (8-45 ページ)	2 つの表の相違点を判定します。
<a href="#">RECTIFY プロシージャ</a> (8-48 ページ)	2 つの表の相違点を解決します。

## DIFFERENCES プロシージャ

このプロシージャは、2 つの表の相違点を判定します。

### 構文

```
DBMS_RECTIFIER_DIFF.DIFFERENCES (  
    sname1          IN  VARCHAR2,  
    oname1          IN  VARCHAR2,  
    reference_site   IN  VARCHAR2 := '',  
    sname2          IN  VARCHAR2,  
    oname2          IN  VARCHAR2,  
    comparison_site  IN  VARCHAR2 := '',  
    where_clause     IN  VARCHAR2 := '',  
    { column_list    IN  VARCHAR2 := '',  
      | array_columns IN  dbms_utility.name_array, }  
    missing_rows_sname IN  VARCHAR2,  
    missing_rows_oname1 IN VARCHAR2,  
    missing_rows_oname2 IN VARCHAR2,  
    missing_rows_site  IN  VARCHAR2 := '',  
    max_missing       IN  INTEGER,  
    commit_rows       IN  INTEGER := 500);
```

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと array\_columns パラメータは、同時には指定できません。

パラメータ

表 8-70 DIFFERENCES プロシージャのパラメータ ( 1 / 2 ページ)

パラメータ	説明
sname1	REFERENCE_SITE のスキーマの名前です。
oname1	REFERENCE_SITE の表の名前です。
reference_site	参照データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
sname2	COMPARISON_SITE のスキーマの名前です。
oname2	COMPARISON_SITE の表の名前です。
comparison_site	比較データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
where_clause	この制限を満たす行のみが比較されます。デフォルトの NULL の場合は、すべての行が比較されます。
column_list	2 つの表で比較される 1 つ以上の列名の、カンマ区切りのリストです。カンマの前後にはスペースを入れません。デフォルトの NULL の場合は、すべての列が比較されます。
array_columns	2 つの表で比較される列名の、PL/SQL 表です。索引付けは 1 の位置から始まり、配列の最後の要素は NULL になります。1 の位置が NULL の場合は、すべての列が使用されます。
missing_rows_sname	欠落行のある表が含まれるスキーマの名前です。
missing_rows_oname1	COMPARISON サイトの表から欠落している REFERENCE サイトの表の行、および REFERENCE サイトの表から欠落している COMPARISON サイトの行の情報が格納される、MISSING_ROWS_SITE の表の名前です。
missing_rows_oname2	欠落行の情報が格納される MISSING_ROWS_SITE の表の名前です。この表には 3 つの列があります。それぞれの列には、MISSING_ROWS_ONAME1 表の行の ROWID、行が存在するサイトの名前、および行が欠落しているサイトの名前が格納されます。
missing_rows_site	MISSING_ROWS_ONAME1 表と MISSING_ROWS_ONAME2 表があるサイトの名前です。デフォルトの NULL は、2 つの表がカレント・サイトにあることを示します。

表 8-70 DIFFERENCES プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
max_missing	missing_rows_onsame 表に挿入される行の最大数を示す整数値です。max_missing を超える数の行が欠落している場合は、指定された最大数の行が missing_rows_onsame に挿入され、それを超える行は欠落しているかどうかを判定せずに、ルーチンが通常どおり終了します。フラグメント間の違いが大きく、欠落行表へのエントリが多くなりすぎると、ルーチンを続行しても無駄になります。このような場合、この引数を使用すると便利です。max_missing の値が 1 より小さい場合、または NULL の場合は、例外 badnumber が発生します。
commit_rows	COMMIT が行われる前に、参照表や比較表に挿入される、または参照表や比較表から削除される行の最大数です。デフォルトでは、500 行が挿入または削除された後、COMMIT が行われます。空の文字列 ('') や NULL の場合は、1 つの表に対してすべての行が挿入または削除された後に、COMMIT が行われます。

## 例外

表 8-71 DIFFERENCES プロシージャの例外

例外	説明
nosuchsite	データベース・サイトが見つかりませんでした。
badnumber	1 より小さい値が COMMIT_ROWS パラメータに指定されています。
missingprimarykey	列のリストには、主キー（または等価の SET_COLUMNS）が含まれる必要があります。
badname	表またはスキーマの名前が、NULL または空の文字列です。
cannotbnull	パラメーターは NULL に設定できません。
notshapeequivalent	比較される表の形式が一致しません。形式とは、列数、列名、列のデータ型を意味します。
unknowncolumn	列が存在しません。
unsupportedtype	タイプがサポートされていません。
dbms_repcat. commfailure	リモート・サイトにアクセスできません。
dbms_repcat. missingobject	表が存在しません。

制限事項

MISSING\_ROWS\_DATA 表に一意キーまたは主キーの制約がある場合、エラー ORA-00001 (一意性制約に反しています) が発生します。

RECTIFY プロシージャ

このプロシージャは、2 つの表の相違点を解決します。

構文

```
DBMS_RECTIFIER_DIFF.RECTIFY (
  sname1          IN  VARCHAR2,
  oname1          IN  VARCHAR2,
  reference_site  IN  VARCHAR2 := '',
  sname2          IN  VARCHAR2,
  oname2          IN  VARCHAR2,
  comparison_site IN  VARCHAR2 := '',
  { column_list   IN  VARCHAR2 := '',
    | array_columns IN dbms_utility.name_array, }
  missing_rows_sname IN  VARCHAR2,
  missing_rows_oname1 IN  VARCHAR2,
  missing_rows_oname2 IN  VARCHAR2,
  missing_rows_site  IN  VARCHAR2 := '',
  commit_rows       IN  INTEGER := 500);
```

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと array\_columns パラメータは、同時には指定できません。

パラメータ

表 8-72 RECTIFY プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
sname1	REFERENCE_SITE のスキーマの名前です。
oname1	REFERENCE_SITE の表の名前です。
reference_site	参照データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
sname2	COMPARISON_SITE のスキーマの名前です。
oname2	COMPARISON_SITE の表の名前です。



表 8-72 RECTIFY プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
comparison_site	比較データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
column_list	2 つの表で比較される 1 つ以上の列名の、カンマ区切りのリストです。カンマの前後にはスペースを入れません。デフォルトの NULL の場合は、すべての列が比較されます。
array_columns	2 つの表で比較される列名の、PL/SQL 表です。索引付けは 1 の位置から始まり、配列の最後の要素は NULL になります。1 の位置が NULL の場合は、すべての列が使用されます。
missing_rows_sname	欠落行のある表が含まれるスキーマの名前です。
missing_rows_onsame1	COMPARISON サイトの表から欠落している REFERENCE サイトの表の行、および REFERENCE サイトの表から欠落している COMPARISON サイトの行の情報が格納される、MISSING_ROWS_SITE の表の名前です。
missing_rows_onsame2	欠落行の情報が格納される MISSING_ROWS_SITE の表の名前です。この表には 3 つの列があります。それぞれの列には、MISSING_ROWS_ONSAME1 表の行の ROWID、行が存在するサイトの名前、および行が欠落しているサイトの名前が格納されます。
missing_rows_site	MISSING_ROWS_ONSAME1 表と MISSING_ROWS_ONSAME2 表があるサイトの名前です。デフォルトの NULL は、2 つの表がカレント・サイトにあることを示します。
commit_rows	COMMIT が行われる前に、参照表や比較表に挿入される、または参照表や比較表から削除される行の最大数です。デフォルトでは、500 行が挿入または削除された後、COMMIT が行われます。空の文字列 ('') や NULL の場合は、1 つの表に対してすべての行が挿入または削除された後に、COMMIT が行われます。

例外

表 8-73 RECTIFY プロシージャの例外

例外	説明
<code>nosuchsite</code>	データベース・サイトが見つかりませんでした。
<code>badnumber</code>	1 より小さい値が <code>COMMIT_ROWS</code> パラメータに指定されています。
<code>badname</code>	表またはスキーマの名前が、 <code>NULL</code> または空の文字列です。
<code>dbms_repcat. commfailure</code>	リモート・サイトにアクセスできません。
<code>dbms_repcat. missingobject</code>	表が存在しません。

# DBMS\_REFRESH パッケージ

## サブプログラムの概要

表 8-74 DBMS\_REFRESH パッケージのサブプログラム

サブプログラム	説明
<a href="#">ADD プロシージャ</a> (8-51 ページ)	スナップショットをリフレッシュ・グループに追加します。
<a href="#">CHANGE プロシージャ</a> (8-52 ページ)	リフレッシュ・グループのリフレッシュ間隔を変更します。
<a href="#">DESTROY プロシージャ</a> (8-54 ページ)	リフレッシュ・グループからすべてのスナップショットを削除し、リフレッシュ・グループを削除します。
<a href="#">MAKE プロシージャ</a> (8-55 ページ)	リフレッシュ・グループ・メンバー、およびこのグループ・メンバーをリフレッシュする間隔を指定します。
<a href="#">REFRESH プロシージャ</a> (8-57 ページ)	リフレッシュ・グループを手動でリフレッシュします。
<a href="#">SUBTRACT プロシージャ</a> (8-58 ページ)	スナップショットをリフレッシュ・グループから削除します。

## ADD プロシージャ

このプロシージャは、スナップショットをリフレッシュ・グループに追加します。

**関連項目：** 詳細は、5-7 ページの「[リフレッシュ・グループにオブジェクトの追加](#)」および『Oracle8i レプリケーション・ガイド』の第 3 章「スナップショットの概念およびアーキテクチャ」を参照してください。

### 構文

```
DBMS_REFRESH.ADD (  
    name      IN VARCHAR2,  
    { list    IN VARCHAR2,  
      | tab   IN DBMS_UTILITY.UNCL_ARRAY, }  
    lax       IN BOOLEAN := FALSE);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

パラメータ

表 8-75 ADD プロシージャのパラメータ

パラメータ	説明
name	メンバーを追加するリフレッシュ・グループの名前です。
list	リフレッシュ・グループに追加するスナップショットの、カンマ区切りのリストです（シノニムはサポートされていません）。
tab	カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY タイプの PL/SQL 表を使用できます。この表の各要素は、スナップショットの名前になります。最初のスナップショットは 1 の位置に置かれます。最後の位置は NULL になります。
lax	スナップショットは、一度に 1 つのリフレッシュ・グループのみに所属できます。スナップショットを他のグループに移動する場合は、lax フラグを TRUE に設定する必要があります。スナップショットは自動的に移動され、移動先のグループのリフレッシュ間隔が適用されます。このフラグが FALSE に設定されている場合は、ADD のコールでエラー・メッセージが生成されます。

CHANGE プロシージャ

このプロシージャは、リフレッシュ・グループのリフレッシュ間隔を変更します。

**関連項目：** 詳細は、『Oracle8i レプリケーション・ガイド』の第 3 章「スナップショットの概念およびアーキテクチャ」を参照してください。

構文

```
DBMS_REFRESH.CHANGE (
    name                IN VARCHAR2,
    next_date            IN DATE           := NULL,
    interval             IN VARCHAR2      := NULL,
    implicit_destroy     IN BOOLEAN       := NULL,
    rollback_seg         IN VARCHAR2      := NULL,
    push_deferred_rpc    IN BOOLEAN       := NULL,
    refresh_after_errors IN BOOLEAN       := NULL,
    purge_option         IN BINARY_INTEGER := NULL,
    parallelism          IN BINARY_INTEGER := NULL,
    heap_size            IN BINARY_INTEGER := NULL);
```

パラメータ

表 8-76 CHANGE プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
name	リフレッシュ間隔を変更するリフレッシュ・グループの名前です。
next_date	次にリフレッシュを行う日付です。デフォルトでは、この日付は変更されません。
interval	このファンクションを使用して、リフレッシュ・グループ内のスナップショットを次にリフレッシュするタイミングが計算されます。 interval は、リフレッシュの直前に評価されます。したがって、リフレッシュにかかる時間より長い間隔を選択する必要があります。デフォルトでは、interval は変更されません。
implicit_destroy	implicit_destroy フラグの値をリセットできます。このフラグが設定されている場合、グループにメンバーが存在しなくなると、そのグループが自動的に削除されます。デフォルトでは、このフラグは変更されません。
rollback_seg	使用するロールバック・セグメントを変更できます。デフォルトでは、ロールバック・セグメントは変更されません。このパラメータをリセットしてデフォルトのロールバック・セグメントを使用するには、引用符を含めて NULL を指定します。引用符を含めずに NULL を指定すると、現在使用しているロールバック・セグメントが変更されません。
push_deferred_rpc	更新可能なスナップショットのみで使用します。スナップショットをリフレッシュする前に、スナップショットから対応付けられたマスターに変更をプッシュする場合は、このパラメータを TRUE に設定します。TRUE に設定しないと、変更が一時的に失われたように見えることがあります。デフォルトでは、このフラグは変更されません。
refresh_after_errors	更新可能なスナップショットのみで使用します。スナップショットのマスターで DEFERROR ビューに未解決の競合がログインされていてもリフレッシュを行う場合は、このパラメータを TRUE に設定します。デフォルトでは、このフラグは変更されません。

表 8-76 CHANGE プロシージャのパラメータ ( 2 / 2 ページ)

パラメータ	説明
purge_option	<p>パラレル伝播メカニズムを使用している場合（すなわち parallelism が 1 以上に設定されている場合）、このパラメータは次のようになります。</p> <ul style="list-style-type: none"><li>0= パージなし</li><li>1=lazy パージ（デフォルト）</li><li>2=aggressive パージ</li></ul> <p>ほとんどの場合、lazy パージが最適な設定になります。マルチマスター・レプリケーション・グループが異なるターゲット・サイトにプッシュされ、1 つ以上のレプリケーション・グループに対する更新が不定期に行われるか不定期にプッシュされる場合は、aggressive パージに設定してキューを少なくします。すべてのレプリケーション・グループが不定期に更新およびプッシュされる場合は、パージなしに設定して、随時 PUSH を実行します。PUSH を実行するときは、aggressive パージに設定してキューを削減します。</p>
parallelism	<p>0 の場合はシリアル伝播、<math>n &gt; 1</math> の場合は <math>n</math> パラレル・サーバー・プロセスのパラレル伝播、1 の場合は 1 パラレル・サーバー・プロセスのみのパラレル伝播です。</p>
heap_size	<p>パラレル伝播スケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。オラクル社カスタマ・サポート・センターから指示された場合を除き、パラメータを設定しないでください。</p>

DESTROY プロシージャ

このプロシージャは、リフレッシュ・グループからすべてのスナップショットを削除して、リフレッシュ・グループを削除します。

**関連項目：** 詳細は、『Oracle8i レプリケーション・ガイド』の第 3 章「スナップショットの概念およびアーキテクチャ」を参照してください。

構文

```
DBMS_REFRESH.DESTROY (  
    name    IN    VARCHAR2);
```

パラメータ

表 8-77 DESTROY プロシージャのパラメータ

パラメータ	説明
name	破棄するリフレッシュ・グループの名前です。

MAKE プロシージャ

このプロシージャは、リフレッシュ・グループ・メンバー、およびこのグループ・メンバーをリフレッシュする間隔を指定します。

**関連項目：** 5-6 ページの「[リフレッシュ・グループの作成](#)」および『Oracle8i レプリケーション・ガイド』の第 3 章「スナップショットの概念およびアーキテクチャ」を参照してください。

構文

```
DBMS_REFRESH.MAKE (
    name                IN      VARCHAR2
  { list                IN      VARCHAR2,
    | tab               IN      DBMS_UTILITY.UNCL_ARRAY, }
  next_date            IN      DATE,
  interval             IN      VARCHAR2,
  implicit_destroy     IN      BOOLEAN           := FALSE,
  lax                  IN      BOOLEAN           := FALSE,
  job                  IN      BINARY_INTEGER   := 0,
  rollback_seg         IN      VARCHAR2         := NULL,
  push_deferred_rpc    IN      BOOLEAN           := TRUE,
  refresh_after_errors IN      BOOLEAN           := FALSE)
  purge_option          IN      BINARY_INTEGER   := NULL,
  parallelism          IN      BINARY_INTEGER   := NULL,
  heap_size            IN      BINARY_INTEGER   := NULL);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

表 8-78 MAKE プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
name	リフレッシュ・グループの識別に使用される、一意の名前です。リフレッシュ・グループは、表と同じ命名規則に従って名前付けをします。
list	リフレッシュするスナップショットの、カンマ区切りのリストです (シノニムはサポートされていません)。リストには、それぞれ異なるスキーマに置かれ異なるマスター表を持つスナップショットを指定できます。ただし、リスト内のすべてのスナップショットがカレント・データベースにある必要があります。
tab	カンマ区切りのリストのかわりに、データ型 DBMS_UTILITY.UNCL_ARRAY を使用した PL/SQL 表で、リフレッシュするスナップショットの名前を指定できます。表にスナップショット名が <i>n</i> 個ある場合、最初のスナップショットは 1 の位置に置かれ、 <i>n</i> + 1 の位置は NULL に設定されます。
next_date	次にリフレッシュを行う日付です。
interval	<p>このファンクションを使用して、グループ内のスナップショットを次にリフレッシュするタイミングが計算されます。このフィールドは、NEXT_DATE 値とともに使用します。</p> <p>たとえば、interval に NEXT_DAY (SYSDATE+1, "MONDAY") を指定した場合、NEXT_DATE が月曜と評価されると、毎週月曜日にスナップショットがリフレッシュされます。interval は、リフレッシュの直前に評価されます。したがって、リフレッシュにかかる時間より長い間隔を選択する必要があります。</p>
implicit_destroy	リフレッシュ・グループにメンバーが存在しなくなったとき、リフレッシュ・グループを自動的に削除するには、このパラメータを TRUE に設定します。このフラグは、SUBTRACT プロシージャがコールされたときのみチェックされます。つまり、このフラグが設定されていても、空のリフレッシュ・グループを作成できます。
lax	スナップショットは、一度に 1 つのリフレッシュ・グループのみに所属できます。スナップショットを、既存のグループから新しいリフレッシュ・グループへ移動する場合は、このパラメータを TRUE に設定する必要があります。スナップショットは自動的に削除され、移動先のグループのリフレッシュ間隔が適用されます。FALSE に設定されている場合は、MAKE のコールでエラー・メッセージが生成されます。
job	インポート・ユーティリティで必要です。デフォルトの 0 を使用してください。
rollback_seg	スナップショットのリフレッシュ時に使用するロールバック・セグメントの名前です。デフォルトの NULL の場合、デフォルトのロールバック・セグメントが使用されます。



表 8-78 MAKE プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
push_deferred_rpc	更新可能なスナップショットのみで使⽤します。スナップショットをリフレッシュする前に、スナップショットから対応付けられたマスターに変更をプッシュする場合は、デフォルト値 TRUE を使⽤します。TRUE に設定しないと、変更が一時的に失われたように見えることがあります。
refresh_after_errors	更新可能なスナップショットのみで使⽤します。スナップショットのマスターで DEFERROR ビューに未解決の競合がログインされていてもリフレッシュを行う場合は、このパラメータを 0 に設定します。
purge_option	パラレル伝播メカニズムを使⽤している場合 (すなわち parallelism が 1 以上に設定されている場合)、0 はページなし、1 は lazy ページ (デフォルト)、2 は aggressive ページになります。ほとんどの場合、lazy ページが最適な設定になります。  マルチマスター・レプリケーション・グループが異なるターゲット・サイトにプッシュされ、1 つ以上のレプリケーション・グループに対する更新が不定期に行われるか不定期にプッシュされる場合は、aggressive ページを設定してキューを少なくします。すべてのレプリケーション・グループが不定期に更新およびプッシュされる場合は、ページなしに設定して、随時 push を実行します。PUSH を実行するときは、aggressive ページに設定してキューを削減します。
parallelism	0 の場合はシリアル伝播、 $n > 1$ の場合は $n$ パラレル・サーバー・プロセスのパラレル伝播、1 の場合は 1 パラレル・サーバー・プロセスのみのパラレル伝播です。
heap_size	パラレル伝播スケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。オラクル社カスタマ・サポート・センターから指示された場合を除き、パラメータを設定しないでください。

REFRESH プロシージャ

このプロシージャは、リフレッシュ・グループを手動でリフレッシュします。

**関連項目：**『Oracle8i レプリケーション・ガイド』の第 3 章「スナップショットの概念およびアーキテクチャ」を参照してください。

構文

```
DBMS_REFRESH.REFRESH (  
    name      IN      VARCHAR2);
```

表 8-79 REFRESH プロシージャのパラメータ

パラメータ	説明
name	手動でリフレッシュするリフレッシュ・グループの名前です。

SUBTRACT プロシージャ

このプロシージャは、スナップショットをリフレッシュ・グループから削除します。

**関連項目：**『Oracle8i レプリケーション・ガイド』の第 3 章「スナップショットの概念およびアーキテクチャ」を参照してください。

構文

```
DBMS_REFRESH.SUBTRACT (  
  name          IN    VARCHAR2,  
  { list        IN    VARCHAR2,  
    | tab        IN    DBMS_UTILITY.UNCL_ARRAY, }  
  lax           IN    BOOLEAN := FALSE);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

パラメータ

表 8-80 SUBTRACT プロシージャのパラメータ

パラメータ	説明
name	メンバーを削除するリフレッシュ・グループの名前です。
list	リフレッシュ・グループから削除するスナップショットの、カンマ区切りのリストです（シノニムはサポートされていません）。それぞれ異なるスキーマに置かれ異なるマスター表を持つスナップショットを指定できます。ただし、リスト内のすべてのスナップショットが、カレント・データベースに存在する必要があります。
tab	カンマ区切りのリストのかわりに、データ型 DBMS_UTILITY.UNCL_ARRAY を使用した PL/SQL 表で、リフレッシュするスナップショットの名前を指定できます。表にスナップショット名が $n$ 個ある場合、最初のスナップショットは 1 の位置に置かれ、 $n + 1$ の位置は NULL に設定されます。
lax	リフレッシュ・グループ・メンバーではないスナップショットの削除を試みたときに、エラー・メッセージが生成されるようにするには、このパラメータを FALSE に設定します。

# DBMS\_REPCAT パッケージ

## サブプログラムの概要

表 8-81 DBMS\_REPCAT パッケージのサブプログラム (1 / 5 ページ)

サブプログラム	説明
<a href="#">ADD_GROUPED_COLUMN プロシージャ</a> (8-63 ページ)	既存の列グループにメンバーを追加します。
<a href="#">ADD_MASTER_DATABASE プロシージャ</a> (8-64 ページ)	レプリケート環境に他のマスター・サイトを追加します。
<a href="#">ADD_PRIORITY_datatype プロシージャ</a> (8-65 ページ)	優先グループにメンバーを追加します。
<a href="#">ADD_SITE_PRIORITY_SITE プロシージャ</a> (8-67 ページ)	サイト優先順位グループに新しいサイトを追加します。
<a href="#">ADD_conflicttype_RESOLUTION プロシージャ</a> (8-68 ページ)	更新、削除または一意性の競合のための、解消方法を指定します。
<a href="#">ALTER_MASTER_PROPAGATION プロシージャ</a> (8-71 ページ)	指定されたマスター・サイトの、指定されたオブジェクト・グループの伝播方法を変更します。
<a href="#">ALTER_MASTER_REPOBJECT プロシージャ</a> (8-73 ページ)	レプリケート環境のオブジェクトを変更します。
<a href="#">ALTER_PRIORITY プロシージャ</a> (8-74 ページ)	指定された優先グループ・メンバーに対応付けられた優先順位レベルを変更します。
<a href="#">ALTER_PRIORITY_datatype プロシージャ</a> (8-75 ページ)	優先グループ・メンバーの値を変更します。
<a href="#">ALTER_SITE_PRIORITY プロシージャ</a> (8-77 ページ)	指定されたサイトに対応付けられた優先順位レベルを変更します。
<a href="#">ALTER_SITE_PRIORITY_SITE プロシージャ</a> (8-78 ページ)	指定された優先順位レベルに対応付けられたサイトを変更します。
<a href="#">ALTER_SNAPSHOT_PROPAGATION プロシージャ</a> (8-79 ページ)	カレント・スナップショット・サイトの、指定されたオブジェクト・グループの伝播方法を変更します。
<a href="#">CANCEL_STATISTICS プロシージャ</a> (8-80 ページ)	正常に実行された、表の更新、一意性および削除の競合解消の統計の収集を中止します。
<a href="#">COMMENT_ON_COLUMN_GROUP プロシージャ</a> (8-81 ページ)	列グループの ALL_REPCOLUMN_GROUP ビュー内のコメント・フィールドを更新します。

表 8-81 DBMS\_REPCAT パッケージのサブプログラム (2 / 5 ページ)

サブプログラム	説明
<a href="#">COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY プロシージャ</a> (8-82 ページ)	(サイト) 優先順位グループの ALL_REPPRIORITY_GROUP ビュー内のコメント・フィールドを更新します。
<a href="#">COMMENT_ON_REPGROUP プロシージャ</a> (8-83 ページ)	レプリケート・マスター・グループの ALL_REPGROUP ビュー内のコメント・フィールドを更新します。
<a href="#">COMMENT_ON_REOBJECT プロシージャ</a> (8-84 ページ)	レプリケート・オブジェクトの ALL_REOBJECT ビュー内のコメント・フィールドを更新します。
<a href="#">COMMENT_ON_REPSITES プロシージャ</a> (8-85 ページ)	レプリケート・サイトの ALL_REPSITE ビュー内のコメント・フィールドを更新します。
<a href="#">COMMENT_ON_SNAPSHOT_REPSITES プロシージャ</a> (8-86 ページ)	スナップショット・サイトの ALL_REPGROUP ビュー内の SCHEMA_COMMENT フィールドを更新します。
<a href="#">COMMENT_ON_conflicttype_RESOLUTION プロシージャ</a> (8-87 ページ)	競合解消ルーチンの ALL_REPRESOLUTION ビュー内のコメント・フィールドを更新します。
<a href="#">COMPARE_OLD_VALUES プロシージャ</a> (8-88 ページ)	更新および削除のために、各マスター・サイトのレプリケート表の非キー列の古い列値を比較します。
<a href="#">CREATE_MASTER_REPGROUP プロシージャ</a> (8-90 ページ)	空の停止したマスター・レプリケーション・オブジェクト・グループを新規に作成します。
<a href="#">CREATE_MASTER_REOBJECT プロシージャ</a> (8-91 ページ)	オブジェクトがレプリケート・オブジェクトであることを示します。
<a href="#">CREATE_SNAPSHOT_REPGROUP プロシージャ</a> (8-94 ページ)	ローカル・データベースに、空のスナップショット・レプリケーション・オブジェクト・グループを新規に作成します。
<a href="#">CREATE_SNAPSHOT_REOBJECT プロシージャ</a> (8-95 ページ)	スナップショット・サイトにレプリケート・オブジェクトを追加します。
<a href="#">DEFINE_COLUMN_GROUP プロシージャ</a> (8-97 ページ)	空の列グループを作成します。
<a href="#">DEFINE_PRIORITY_GROUP プロシージャ</a> (8-98 ページ)	レプリケート・マスター・グループの新しい優先グループを作成します。
<a href="#">DEFINE_SITE_PRIORITY プロシージャ</a> (8-99 ページ)	レプリケート・マスター・グループの新しいサイト優先順位グループを作成します。
<a href="#">DO_DEFERRED_REPCAT_ADMIN プロシージャ</a> (8-100 ページ)	カレント・マスター・サイトの指定されたレプリケート・マスター・グループ、またはすべてのマスター・サイトで、ローカルな遅延管理プロシージャを実行します。

表 8-81 DBMS\_REPCAT パッケージのサブプログラム (3 / 5 ページ)

サブプログラム	説明
<a href="#">DROP_COLUMN_GROUP プロシージャ</a> (8-101 ページ)	列グループを削除します。
<a href="#">DROP_GROUPED_COLUMN プロシージャ</a> (8-102 ページ)	列グループからメンバーを削除します。
<a href="#">DROP_MASTER_REPGROUP プロシージャ</a> (8-103 ページ)	カレント・サイトからレプリケート・マスター・グループを削除します。
<a href="#">DROP_MASTER_REPOBJECT プロシージャ</a> (8-105 ページ)	レプリケート・マスター・グループからレプリケート・オブジェクトを削除します。
<a href="#">DROP_PRIORITY プロシージャ</a> (8-106 ページ)	優先順位レベルに基づいて、優先グループ・メンバーを削除します。
<a href="#">DROP_PRIORITY_GROUP プロシージャ</a> (8-107 ページ)	指定されたレプリケート・マスター・グループの優先グループを削除します。
<a href="#">DROP_PRIORITY_datatype プロシージャ</a> (8-108 ページ)	値に基づいて、優先グループ・メンバーを削除します。
<a href="#">DROP_SITE_PRIORITY プロシージャ</a> (8-109 ページ)	指定されたレプリケート・マスター・グループのサイト優先順位グループを削除します。
<a href="#">DROP_SITE_PRIORITY_SITE プロシージャ</a> (8-110 ページ)	サイト優先順位グループから、指定された名前のサイトを削除します。
<a href="#">DROP_SNAPSHOT_REPGROUP プロシージャ</a> (8-111 ページ)	レプリケート環境からスナップショット・サイトを削除します。
<a href="#">DROP_SNAPSHOT_REPOBJECT プロシージャ</a> (8-112 ページ)	スナップショット・サイトからレプリケート・オブジェクトを削除します。
<a href="#">DROP_conflicttype_RESOLUTION プロシージャ</a> (8-113 ページ)	更新、削除または一意性の競合解消ルーチンを削除します。
<a href="#">EXECUTE_DDL プロシージャ</a> (8-114 ページ)	各マスター・サイトで実行される DDL を渡します。
<a href="#">GENERATE_REPLICATION_SUPPORT プロシージャ</a> (8-116 ページ)	レプリケーションのサポートに必要なトリガー、パッケージおよびプロシージャを生成します。
<a href="#">GENERATE_SNAPSHOT_SUPPORT プロシージャ</a> (8-117 ページ)	トリガーをアクティブにして、更新可能なスナップショットのレプリケーションやプロシージャ型レプリケーションのサポートに必要なパッケージを生成します。
<a href="#">MAKE_COLUMN_GROUP プロシージャ</a> (8-119 ページ)	1 つ以上のメンバーを持つ新しい列グループを作成します。

表 8-81 DBMS\_REPCAT パッケージのサブプログラム (4 / 5 ページ)

サブプログラム	説明
<a href="#">PURGE_MASTER_LOG プロシージャ</a> (8-120 ページ)	指定された ID 番号、ソースまたはレプリケート・マスター・グループに対応付けられた、RepCatLog 内のローカル・メッセージを削除します。
<a href="#">PURGE_STATISTICS プロシージャ</a> (8-121 ページ)	ALL_REPRESOLUTION_STATISTICS ビューから情報を削除します。
<a href="#">REFRESH_SNAPSHOT_REPGROUP プロシージャ</a> (8-122 ページ)	スナップショット・サイト・オブジェクト・グループを、対応付けられたマスター・サイトの最新データにリフレッシュします。
<a href="#">REGISTER_SNAPSHOT_REPGROUP プロシージャ</a> (8-123 ページ)	DBA_REGISTERED_SNAPSHOT_GROUPS でのスナップショットの挿入、変更または削除を行って、マスター・サイトでのスナップショットの管理を容易にします。
<a href="#">REGISTER_STATISTICS プロシージャ</a> (8-124 ページ)	正常に実行された、表の更新、削除および一意性の競合解消の統計を収集します。
<a href="#">RELOCATE_MASTERDEF プロシージャ</a> (8-125 ページ)	マスター定義サイトを、レプリケート環境の他のマスター・サイトに変更します。
<a href="#">REMOVE_MASTER_DATABASES プロシージャ</a> (8-126 ページ)	レプリケート環境から、1 つ以上のマスター・データベースを削除します。
<a href="#">REPCAT_IMPORT_CHECK プロシージャ</a> (8-128 ページ)	レプリケート・オブジェクト、またはアドバンスト・レプリケーション機能で使されたオブジェクトのエクスポートまたはインポート後に、レプリケート・マスター・グループのオブジェクトが適切なオブジェクト識別子およびステータス値を持つようにします。
<a href="#">RESUME_MASTER_ACTIVITY プロシージャ</a> (8-129 ページ)	レプリケート環境を停止した後で、通常のレプリケーション・アクティビティを再開します。
<a href="#">SEND_OLD_VALUES プロシージャ</a> (8-129 ページ)	更新および削除のために、レプリケート表の非キー列の古い列値を送信するかどうかを指定します。
<a href="#">SET_COLUMNS プロシージャ</a> (8-131 ページ)	主キーのかわりに代替列または列グループを使用して、行レベル・レプリケーションを使用する際に比較する表中の列を決定します。
<a href="#">SUSPEND_MASTER_ACTIVITY プロシージャ</a> (8-132 ページ)	オブジェクト・グループのレプリケーション・アクティビティを中断します。
<a href="#">SWITCH_SNAPSHOT_MASTER プロシージャ</a> (8-133 ページ)	スナップショット・レプリケート・マスター・グループのマスター・データベースを、他のマスター・サイトに変更します。
<a href="#">UNREGISTER_SNAPSHOT_REPGROUP プロシージャ</a> (8-134 ページ)	repcat\$_repsite でのスナップショットの挿入、変更または削除を行って、マスター・サイトでのスナップショットの管理を容易にします。

表 8-81 DBMS\_REPCAT パッケージのサブプログラム (5 / 5 ページ)

サブプログラム	説明
<a href="#">VALIDATE</a> ファンクション (8-135 ページ)	マルチマスター・レプリケーション環境のキー条件が正確であるかどうかを確認します。
<a href="#">WAIT_MASTER_LOG</a> プロシージャ (8-137 ページ)	マスター・サイトに非同期で伝播された変更の適用状況を判定します。

## ADD\_GROUPED\_COLUMN プロシージャ

このプロシージャは、既存の列グループにメンバーを追加します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.ADD_GROUPED_COLUMN (  
    sname                IN    VARCHAR2,  
    oname                IN    VARCHAR2,  
    column_group         IN    VARCHAR2,  
    list_of_column_names IN    VARCHAR2 | DBMS_REPCAT.VARCHAR2s);
```

### パラメータ

表 8-82 ADD\_GROUPED\_COLUMN プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	列グループが対応付けられているレプリケート表の名前です。
column_group	メンバーを追加する列グループの名前です。
list_of_column_names	指定された列グループに追加する列の名前です。列名は、カンマ区切りのリストと PL/SQL 表のいずれでも指定できます。PL/SQL 表のタイプは DBMS_REPCAT.VARCHAR2 にする必要があります。表のすべての列を含む列グループを作成するには、単一の値 '*' を使用します。

表 8-83 ADD\_GROUPED\_COLUMN プロシージャの例外

例外	説明
nomasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定された表は存在しません。
missinggroup	指定された列グループは存在しません。
missingcolumn	指定された列は、指定された表に存在しません。
duplicatecolumn	指定された列は、すでに他の列グループのメンバーになっています。
missingschema	指定されたスキーマは存在しません。
notquiesced	指定された表が所属するオブジェクト・グループは停止していません。

ADD\_MASTER\_DATABASE プロシージャ

このプロシージャは、レプリケート環境に他のマスター・サイトを追加します。このプロシージャは、既存のマスター・サイトで、すべてのトリガーとそれらに対応付けられたパッケージを再生成します。このプロシージャは、マスター定義サイトからコールする必要があります。

構文

```
DBMS_REPCAT.ADD_MASTER_DATABASE (  
  gname                IN   VARCHAR2,  
  master               IN   VARCHAR2,  
  use_existing_objects IN   BOOLEAN := TRUE,  
  copy_rows            IN   BOOLEAN := TRUE,  
  comment              IN   VARCHAR2 := '',  
  propagation_mode     IN   VARCHAR2 := 'ASYNCHRONOUS',  
  fname               IN   VARCHAR2 := NULL);
```

パラメータ

表 8-84 ADD\_MASTER\_DATABASE プロシージャのパラメータ

パラメータ	説明
gname	レプリケートされるオブジェクト・グループの名前です。このオブジェクト・グループは、マスター定義サイトに存在している必要があります。
master	新しいマスター・データベースの、完全修飾されたデータベース名です。



表 8-84 ADD\_MASTER\_DATABASE プロシージャのパラメータ

パラメータ	説明
use_existing_objects	新しいマスター・サイトのスキーマにすでに存在する同じタイプで同じ形式のオブジェクトを再利用する場合は、TRUE を指定します。変更の適用の詳細は、『Oracle8i レプリケーション・ガイド』の第2章「マスター・レプリケーションの概念およびアーキテクチャ」を参照してください。
copy_rows	新しいマスター・サイトの表の初期の内容を、マスター定義サイトの表の内容と一致させる場合は、TRUE を指定します。
comment	DBA_REPSITES ビューの MASTER_COMMENT フィールドに追加されます。
propagation_mode	新しいマスター・データベースに変更を送る、または新しいマスター・データベースから変更を受け取る方法です。受け付けられる値は、SYNCHRONOUS および ASYNCHRONOUS です。
fname	このパラメータは内部で使用します。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

## 例外

表 8-85 ADD\_MASTER\_DATABASE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	レプリケート・マスター・グループは中断されていません。
missingrepgroup	オブジェクト・グループは、指定されたデータベース・サイトに存在しません。
commfailure	新しいマスターにアクセスできません。
typefailure	指定された伝播モードは正しくありません。
notcompat	互換モードは 7.3.0.0 以降である必要があります。
duplrepggrp	マスター・サイトはすでに存在しています。

## ADD\_PRIORITY\_datatype プロシージャ

このプロシージャは、優先グループにメンバーを追加します。このプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、priority 列のデータ型によって異なります。priority 列の指定するすべての値について、このプロシージャを 1 回ずつコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.ADD_PRIORITY_datatype (  
  gname           IN   VARCHAR2,  
  pgroup          IN   VARCHAR2,  
  value           IN   datatype,  
  priority        IN   NUMBER);
```

*datatype* は次のとおりです。

```
{ NUMBER  
| VARCHAR2  
| CHAR  
| DATE  
| RAW  
| NCHAR  
| NVARCHAR2 }
```

パラメータ

表 8-86 ADD\_PRIORITY\_datatype プロシージャのパラメータ

パラメータ	説明
gname	優先グループを作成するレプリケート・マスター・グループです。
pgroup	優先グループの名前です。
value	優先グループ・メンバーの値です。この優先グループを使用する表に 対応付けられた priority 列の値の 1 つになります。
priority	この値の優先順位です。数値が高いほど、優先順位が高くなります。

## 例外

**表 8-87 ADD\_PRIORITY\_datatype プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicatevalue	指定された値は、優先グループにすでに存在しています。
duplicatepriority	指定された優先順位は、優先グループにすでに存在しています。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
typefailure	指定された値のデータ型は、この優先グループには不適切です。
notquiesced	指定されたレプリケート・マスター・グループは停止していません。

## ADD\_SITE\_PRIORITY\_SITE プロシージャ

このプロシージャは、サイト優先順位グループに新しいサイトを追加します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

## 構文

```
DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (
  gname      IN   VARCHAR2,
  name       IN   VARCHAR2,
  site       IN   VARCHAR2,
  priority   IN   NUMBER);
```

## パラメータ

**表 8-88 ADD\_SITE\_PRIORITY\_SITE プロシージャのパラメータ**

パラメータ	説明
gname	グループにサイトを追加するレプリケート・マスター・グループの名前です。
name	メンバーを追加するサイト優先順位グループの名前です。
site	追加するサイトのグローバル・データベース名です。
priority	追加するサイトの優先順位レベルです。数値が高いほど、優先順位レベルが高くなります。

例外

表 8-89 ADD\_SITE\_PRIORITY\_SITE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingpriority	指定されたサイト優先順位グループは存在しません。
duplicatepriority	指定された優先順位レベルは、グループの他のサイト用にすでに存在しています。
duplicatevalue	指定されたサイトは、サイト優先順位グループにすでに存在しています。
notquiesced	レプリケート・マスター・グループは停止していません。

ADD\_conflicttype\_RESOLUTION プロシージャ

これらのプロシージャは、更新、削除または一意性の競合を解消する方法を指定します。これらのプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、ルーチンが解消する競合のタイプによって異なります。

表 8-90 ADD\_conflicttype\_RESOLUTION プロシージャ

競合のタイプ	プロシージャ名
update	ADD_UPDATE_RESOLUTION
uniqueness	ADD_UNIQUE_RESOLUTION
delete	ADD_DELETE_RESOLUTION

**関連項目：** 更新競合解消方法の指定、一意性競合解消方法の選択、削除競合解消方法の割当ての詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.ADD_UPDATE_RESOLUTION (  
    sname                IN   VARCHAR2,  
    oname                IN   VARCHAR2,  
    column_group         IN   VARCHAR2,  
    sequence_no          IN   NUMBER,  
    method               IN   VARCHAR2,  
    parameter_column_name IN   VARCHAR2 | DBMS_REPCAT.VARCHAR2s,  
    priority_group        IN   VARCHAR2      := NULL,
```

```

function_name      IN   VARCHAR2      := NULL,
comment           IN   VARCHAR2      := NULL);

DBMS_REPCAT.ADD_DELETE_RESOLUTION (
    sname           IN   VARCHAR2,
    oname           IN   VARCHAR2,
    sequence_no     IN   NUMBER,
    parameter_column_name IN VARCHAR2 | DBMS_REPCAT.VARCHAR2s,
    function_name    IN   VARCHAR2,
    comment         IN   VARCHAR2      := NULL
    method          IN   VARCHAR2      := 'USER FUNCTION');

DBMS_REPCAT.ADD_UNIQUE_RESOLUTION(
    sname           IN   VARCHAR2,
    oname           IN   VARCHAR2,
    constraint_name  IN   VARCHAR2,
    sequence_no     IN   NUMBER,
    method          IN   VARCHAR2,
    parameter_column_name IN VARCHAR2 | DBMS_REPCAT.VARCHAR2s,
    function_name    IN   VARCHAR2      := NULL,
    comment         IN   VARCHAR2      := NULL);

```

## パラメータ

**表 8-91 ADD\_conflicttype\_RESOLUTION プロシージャのパラメータ (1 / 2 ページ)**

パラメータ	説明
sname	レプリケートされる表が含まれるスキーマの名前です。
oname	競合解消ルーチンを追加する表の名前です。
column_group	競合解消ルーチンを追加する列グループの名前です。列グループは、更新競合解消ルーチンのみに必要です。
constraint_name	競合解消ルーチンを追加する一意性制約または一意索引の名前です。一意索引の名前が対応付けられた一意性制約の名前と異なる場合は、一意索引の名前を使用します。制約名は、一意性競合解消ルーチンのみに必要です。
sequence_no	指定された競合解消方法を適用する順序です。

表 8-91 ADD\_conflicttype\_RESOLUTION プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
method	作成する競合解消ルーチンのタイプです。アドバンスト・レプリケーションで提供される標準のルーチンの名前を指定できます。また、独自のルーチンを記述した場合は、USER FUNCTION を選択して、独自のルーチンの名前を FUNCTION_NAME 引数として指定します。このリリースでサポートされている方法は、MINIMUM、MAXIMUM、LATEST TIMESTAMP、EARLIEST TIMESTAMP、ADDITIVE、AVERAGE、PRIORITY GROUP、SITE PRIORITY、OVERWRITE および DISCARD (更新競合用)、および APPEND SITE NAME、APPEND SEQUENCE および DISCARD (一意性競合用) です。削除の競合に使用する標準の方法はありません。
parameter_column_name	競合の解消に使用する列の名前です。標準の方法は、1 つの列で操作されます。たとえば、列グループに LATEST TIMESTAMP 方法を使用する場合は、タイムスタンプ値が含まれる列の名前を、引数として渡す必要があります。USER FUNCTION を使用する場合は、任意の数の列を使用して競合を解消できます。  この引数は、列名のカンマ区切りのリスト、または DBMS_REPCAT.VARCHAR2 タイプの PL/SQL 表を受け付けます。単一の値 '*' の場合は、競合の解消に表 (更新の競合の場合は列グループ) 内のすべての列が使用されます。'*' を指定した場合、列はアルファベット順にファンクションへ渡されます。
priority_group	更新競合解消方法の PRIORITY GROUP または SITE PRIORITY を使用する場合は、作成した優先グループの名前を渡す必要があります。  詳細は、『Oracle8i レプリケーション・ガイド』の「競合の解消」を参照してください。その他の方法を使用する場合は、この引数のデフォルト値 NULL を使用できます。この引数は、更新の競合のみに適用されます。
function_name	USER FUNCTION 方法を選択した場合、または削除競合解消ルーチンを追加する場合は、記述した競合解消ルーチンの名前を渡す必要があります。標準の方法を使用する場合は、この引数のデフォルト値 NULL を使用できます。
comment	このユーザー・コメントは、DBA_REPRESOLUTION ビューに追加されます。

## 例外

**表 8-92 ADD\_conflictype\_RESOLUTION プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションを使用する指定されたスキーマに表として存在しません。
missingschema	指定されたスキーマは存在しません。
missingcolumn	PARAMETER_COLUMN_NAME 引数の一部として指定された列は存在しません。
missinggroup	指定された列グループは存在しません。
missingprioritygroup	指定された優先グループは、表に存在しません。
invalidmethod	指定された解消方法は認識されません。
invalidparameter	PARAMETER_COLUMN_NAME 引数に指定された列数は無効です（標準のルーチンでは、1 つの列名のみが受け付けられます）。
missingfunction	指定されたユーザー・ファンクションは存在しません。
missingconstraint	一意性の競合に指定された制約は存在しません。
notquiesced	指定された表が所属するオブジェクト・グループは停止していません。
duplicateresolution	指定された競合解消方法はすでに登録されています。
duplicatesequene	指定された順序番号は指定されたオブジェクトにすでに割り当てられています。
invalidprioritygroup	指定された優先グループは存在しません。
paramtype	優先グループに割り当てられたタイプと異なります。

## ALTER\_MASTER\_PROPAGATION プロシージャ

このプロシージャは、指定されたマスター・サイトにある指定されたオブジェクト・グループの伝播方法を変更します。このオブジェクト・グループは停止する必要があります。このプロシージャは、マスター定義サイトからコールする必要があります。マスターが `dblink_list` または `dblink_table` に出現した場合、ALTER\_MASTER\_PROPAGATION ではそのデータベース・リンクが無視されます。伝播モードをマスターからマスターに変更できません。

構文

```
DBMS_REPCAT.ALTER_MASTER_PROPAGATION (
    gname                IN    VARCHAR2,
    master               IN    VARCHAR2,
    { dblink_list        IN    VARCHAR2,
      | dblink_table     IN    dbms_utility.dblink_array, }
    propagation_mode     IN    VARCHAR2 := 'asynchronous',
    comment              IN    VARCHAR2 := '' );
```

**注意：** このプロシージャはオーバーロードされています。dblink\_list パラメータと dblink\_table パラメータは、同時には指定できません。

パラメータ

表 8-93 ALTER\_MASTER\_PROPAGATION プロシージャのパラメータ

パラメータ	説明
gname	伝播モードの変更先のオブジェクト・グループの名前です。
master	伝播モードを変更するマスター・サイトの名前です。
dblink_list	伝播を変更するデータベース・リンクの、カンマ区切りのリストです。値が NULL に設定されている場合、変更されているマスター以外のすべてのマスターが使用されます。
dblink_table	伝播を変更するデータベース・リンクの、1 の位置から索引付けされている PL/SQL 表です。
propagation_mode	指定されたマスター・サイトで変更された内容を、データベース・リンクのリストで識別されるサイトへ伝播する方法を指定します。受け付けられる値は、SYNCHRONOUS および ASYNCHRONOUS です。
comment	このコメントは、DBA_REPPROP ビューに追加されます。

例外

表 8-94 ALTER\_MASTER\_PROPAGATION プロシージャの例外（1 / 2 ページ）

例外	説明
nonmasterdef	ローカル・サイトはマスター定義サイトではありません。
notquiesced	ローカル・サイトは停止していません。
typefailure	指定された伝播モードは認識されませんでした。



表 8-94 ALTER\_MASTER\_PROPAGATION プロシージャの例外 (2 / 2 ページ)

例外	説明
nonmaster	データベース・リンクのリストに、マスター・サイト以外のサイトが含まれています。

## ALTER\_MASTER\_REOBJECT プロシージャ

このプロシージャは、レプリケート環境のオブジェクトを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.ALTER_MASTER_REOBJECT (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    type           IN   VARCHAR2,
    ddl_text       IN   VARCHAR2,
    comment        IN   VARCHAR2      := '',
    retry          IN   BOOLEAN       := FALSE);
```

### パラメータ

表 8-95 ALTER\_MASTER\_REOBJECT プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
sname	変更するオブジェクトが含まれるスキーマの名前です。
oname	変更するオブジェクトの名前です。
type	変更するオブジェクトのタイプです。サポートされているタイプは、TABLE、INDEX、SYNONYM、TRIGGER、VIEW、PROCEDURE、FUNCTION、PACKAGE および PACKAGE BODY です。
ddl_text	オブジェクトの変更に使用する DDL テキストです。DDL は、適用の前に解析されることはありません。このため、変更されるオブジェクトについての適切なスキーマおよびオブジェクト名が、DDL テキストで提供されるようにする必要があります。  スキーマの指定なしで DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずそのスキーマを指定してください。
comment	値が NULL に設定されていない場合、このコメントは DBA_REPOBJECT ビューの COMMENT フィールドに追加されます。

表 8-95 ALTER\_MASTER\_REPOBJECT プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
retry	値が TRUE に設定されている場合、ALTER_MASTER_REPOBJECT では、オブジェクト・ステータスが VALID でないマスターでのみ、オブジェクトが変更されます。

例外

表 8-96 ALTER\_MASTER\_REPOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	対応付けられたオブジェクト・グループは中断されていません。
missingobject	SNAME および ONAME で識別されるオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
ddlfailure	マスター定義サイトの DDL が失敗しました。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

ALTER\_PRIORITY プロシージャ

このプロシージャは、指定された優先グループ・メンバーに対応付けられた優先順位レベルを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.ALTER_PRIORITY (  
  gname          IN   VARCHAR2,  
  pgroup         IN   VARCHAR2,  
  old_priority   IN   NUMBER,  
  new_priority   IN   NUMBER);
```

## パラメータ

**表 8-97 ALTER\_PRIORITY プロシージャのパラメータ**

パラメータ	説明
gname	優先グループが対応付けられたレプリケート・マスター・グループの名前です。
pgroup	変更する優先順位が含まれる優先グループの名前です。
old_priority	優先グループ・メンバーのカレント優先順位レベルです。
new_priority	優先グループ・メンバーに割り当てる新しい優先順位レベルです。

## 例外

**表 8-98 ALTER\_PRIORITY プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicatepriority	新しい優先順位レベルは、優先グループにすでに存在しています。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingvalue	DBMS_REPCAT.ADD_PRIORITY_datatype へのコールで値が登録されませんでした。
missingprioritygroup	指定された優先グループは存在しません。
notquiesced	指定されたレプリケート・マスター・グループは停止していません。

## ALTER\_PRIORITY\_datatype プロシージャ

このプロシージャは、優先グループ・メンバーの値を変更します。このプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、priority 列のデータ型によって異なります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

## 構文

```
DBMS_REPCAT.ALTER_PRIORITY_datatype (
    gname          IN   VARCHAR2,
    pgroup         IN   VARCHAR2,
    old_value      IN   datatype,
    new_value      IN   datatype);
```

*datatype* は次のとおりです。

```
{ NUMBER
| VARCHAR2
| CHAR
| DATE
| RAW
| NCHAR
| NVARCHAR2 }
```

パラメータ

表 8-99 ALTER\_PRIORITY\_datatype プロシージャのパラメータ

パラメータ	説明
gname	優先グループが対応付けられたレプリケート・マスター・グループの名前です。
pgroup	変更する値が含まれる優先グループの名前です。
old_value	優先グループ・メンバーのカレント値です。
new_value	優先グループ・メンバーに割り当てる新しい値です。

例外

表 8-100 ALTER\_PRIORITY\_datatype プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicatevalue	新しい値は、優先グループにすでに存在しています。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
missingvalue	古い値は存在しません。
paramtype	新しい値のデータ型は、この優先グループには不適切です。
typefailure	指定された値のデータ型は、この優先グループには不適切です。
notquiesced	指定されたレプリケート・マスター・グループは停止していません。

## ALTER\_SITE\_PRIORITY プロシージャ

このプロシージャは、指定されたサイトに対応付けられた優先順位レベルを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

### 構文

```
DBMS_REPCAT.ALTER_SITE_PRIORITY (
    gname          IN   VARCHAR2,
    name           IN   VARCHAR2,
    old_priority    IN   NUMBER,
    new_priority    IN   NUMBER);
```

### パラメータ

**表 8-101 ALTER\_SITE\_PRIORITY プロシージャのパラメータ**

パラメータ	説明
gname	サイト優先順位グループが対応付けられたレプリケート・マスター・グループの名前です。
name	メンバーを変更するサイト優先順位グループの名前です。
old_priority	優先順位レベルを変更するサイトの、カレント優先順位レベルです。
new_priority	サイトの新しい優先順位レベルです。数値が高いほど、優先順位レベルが高くなります。

### 例外

**表 8-102 ALTER\_SITE\_PRIORITY プロシージャの例外 (1 / 2 ページ)**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingpriority	古い優先順位レベルは、どのグループ・メンバーにも対応付けられていません。
duplicatepriority	新しい優先順位レベルは、グループの他のサイト用にすでに存在しています。
missingvalue	古い値は存在しません。
paramtype	新しい値のデータ型は、この優先グループには不適切です。

表 8-102 ALTER\_SITE\_PRIORITY\_SITE プロシージャの例外 ( 2 / 2 ページ)

例外	説明
notquiesced	レプリケート・マスター・グループは停止していません。

## ALTER\_SITE\_PRIORITY\_SITE プロシージャ

このプロシージャは、指定された優先順位レベルに対応付けられたサイトを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

### 構文

```
DEMS_REPCAT.ALTER_SITE_PRIORITY_SITE (  
  gname      IN   VARCHAR2,  
  name       IN   VARCHAR2,  
  old_site   IN   VARCHAR2,  
  new_site   IN   VARCHAR2);
```

### パラメータ

表 8-103 ALTER\_SITE\_PRIORITY\_SITE プロシージャのパラメータ

パラメータ	説明
gname	サイト優先順位グループが対応付けられたレプリケート・マスター・グループの名前です。
name	メンバーを変更するサイト優先順位グループの名前です。
old_site	優先順位レベルとの対応付けを解除するサイトの、カレント・グローバル・データベース名です。
new_site	カレント優先順位レベルと対応付ける、新しいグローバル・データベース名です。

## 例外

**表 8-104 ALTER\_SITE\_PRIORITY\_SITE プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingpriority	指定されたサイト優先順位グループは存在しません。
missingvalue	古いサイトはグループ・メンバーではありません。
notquiesced	レプリケート・マスター・グループは停止していません。

## ALTER\_SNAPSHOT\_PROPAGATION プロシージャ

このプロシージャは、カレント・スナップショット・サイトの指定されたオブジェクト・グループの伝播方法を変更します。このプロシージャは、スナップショット・サイトの遅延トランザクション・キューをプッシュし、スナップショット・ベース表をロックし、トリガーおよびトリガーに対応付けられたパッケージを再生成します。このプロシージャは、スナップショット・サイトからコールする必要があります。

## 構文

```
DBMS_REPCAT.ALTER_SNAPSHOT_PROPAGATION (
    gname          IN  VARCHAR2,
    propagation_mode IN  VARCHAR2,
    comment        IN  VARCHAR2  := ' ',
    gowner         IN  VARCHAR2  := 'PUBLIC');
```

## パラメータ

**表 8-105 ALTER\_SNAPSHOT\_PROPAGATION プロシージャのパラメータ**

パラメータ	説明
gname	伝播モードを変更するオブジェクト・グループの名前です。
propagation_mode	カレント・スナップショット・サイトから対応付けられたマスター・サイトへの、変更の伝播方法です。受け付けられる値は、SYNCHRONOUS および ASYNCHRONOUS です。
comment	このコメントは、DBA_REPPROP ビューに追加されます。
gowner	スナップショット・グループの所有者です。

例外

表 8-106 ALTER\_SNAPSHOT\_PROPAGATION プロシージャの例外

例外	説明
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
typefailure	伝播モードが正しく指定されませんでした。
nonsnapshot	カレント・サイトは、指定されたオブジェクト・グループのスナップショット・サイトではありません。
commfailure	マスターにコンタクトできません。
notcompat	互換モードは 7.3.0.0 以降である必要があります。
failaltersnaprop	スナップショット・グループの伝播は、同じマスターにスナップショット・サイトを共有する他のスナップショット・グループがない場合に変更できます。

CANCEL\_STATISTICS プロシージャ

このプロシージャは、正常に実行された、表の更新、一意性および削除の競合解消の統計の収集を中止します。

構文

```
DBMS_REPCAT.CANCEL_STATISTICS (  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2);
```

パラメータ

表 8-107 CANCEL\_STATISTICS プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	競合解消の統計を収集しない表の名前です。



## 例外

**表 8-108 CANCEL\_STATISTICS プロシージャの例外**

例外	説明
missingschema	指定されたスキーマは存在しません。
missingobject	指定された表は存在しません。
statnotreg	指定された表は現在、統計の収集用に登録されていません。

## COMMENT\_ON\_COLUMN\_GROUP プロシージャ

このプロシージャは、列グループの DBA\_REPCOLUMN\_GROUP ビュー内のコメント・フィールドを更新します。このコメントは、DBMS\_REPCAT.GENERATE\_REPLICATION\_SUPPORT への次のコールまで、どのマスター・サイトでも追加されません。

## 構文

```
DBMS_REPCAT.COMMENT_ON_COLUMN_GROUP (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    column_group   IN   VARCHAR2,
    comment        IN   VARCHAR2);
```

## パラメータ

**表 8-109 COMMENT\_ON\_COLUMN\_GROUP プロシージャのパラメータ**

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	列グループが対応付けられているレプリケート表の名前です。
column_group	列グループの名前です。
comment	DBA_REPCOLUMN_GROUP ビューの GROUP_COMMENT フィールドに含める、更新されたコメントのテキストです。

例外

表 8-110 COMMENT\_ON\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missinggroup	指定された列グループは存在しません。
missingobj	オブジェクトがありません。

COMMENT\_ON\_PRIORITY\_GROUP/COMMENT\_ON\_SITE\_PRIORITY プロシージャ

COMMENT\_ON\_PRIORITY\_GROUP では、優先グループの DBA\_REPPRIORITY\_GROUP ビューのコメント・フィールドが更新されます。このコメントは、GENERATE REPLICATION\_SUPPORT への次のコールまで、どのマスター・サイトでも追加されません。

COMMENT\_ON\_SITE\_PRIORITY では、サイト優先順位グループの DBA\_REPPRIORITY\_GROUP ビューのコメント・フィールドが更新されます。このプロシージャは COMMENT\_ON\_COLUMN\_GROUP プロシージャのラッパーで、便宜的に提供されるものです。このプロシージャは、マスター定義サイトからコールする必要があります。

構文

```
DBMS_REPCAT.COMMENT_ON_PRIORITY_GROUP (  
    gname      IN   VARCHAR2,  
    pgroup     IN   VARCHAR2,  
    comment    IN   VARCHAR2);  
  
DBMS_REPCAT.COMMENT_ON_SITE_PRIORITY (  
    gname      IN   VARCHAR2,  
    name       IN   VARCHAR2,  
    comment    IN   VARCHAR2);
```

パラメータ

表 8-111 COMMENT\_ON\_PRIORITY\_GROUP/COMMENT\_ON\_SITE\_PRIORITY のパラメータ

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。
pgroup/name	優先グループまたはサイト優先順位グループの名前です。
comment	DBA_REPPRIORITY_GROUP ビューの PRIORITY_COMMENT フィールドに含める、更新されたコメントのテキストです。

## 例外

**表 8-112 COMMENT\_ON\_PRIORITY\_GROUP/COMMENT\_ON\_SITE\_PRIORITY の例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。

## COMMENT\_ON\_REPGROUP プロシージャ

このプロシージャは、レプリケート・マスター・グループの DBA\_REPGROUP ビュー内のコメント・フィールドを更新します。このプロシージャは、マスター定義サイトからコールする必要があります。

## 構文

```
DBMS_REPCAT.COMMENT_ON_REPGROUP (
    gname      IN   VARCHAR2,
    comment    IN   VARCHAR2);
```

## パラメータ

**表 8-113 COMMENT\_ON\_REPGROUP プロシージャのパラメータ**

パラメータ	説明
gname	コメントを付けるオブジェクト・グループの名前です。
comment	DBA_REPGROUP ビューの SCHEMA_COMMENT フィールドに含める、更新されたコメントです。

## 例外

**表 8-114 COMMENT\_ON\_REPGROUP プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

## COMMENT\_ON\_REPOBJECT プロシージャ

このプロシージャは、マスター・グループ内のレプリケート・オブジェクトの DBA\_REPOBJECT ビュー内のコメント・フィールドを更新します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.COMMENT_ON_REPOBJECT (  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2,  
    type       IN   VARCHAR2,  
    comment    IN   VARCHAR2);
```

### パラメータ

表 8-115 COMMENT\_ON\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	コメントを付けるオブジェクトの名前です。
type	オブジェクトのタイプです。サポートされているタイプは、TABLE、INDEX、SYNONYM、TRIGGER、VIEW、PROCEDURE、FUNCTION、PACKAGE および PACKAGE BODY です。
comment	DBA_REPOBJECT ビューの OBJECT_COMMENT フィールドに含める、更新されたコメントのテキストです。

### 例外

表 8-116 COMMENT\_ON\_REPOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

## COMMENT\_ON\_REPSITES プロシージャ

オブジェクト・グループがマスター・グループの場合、このプロシージャは、マスター・サイトの DBA\_REPSITES ビュー内の MASTER\_COMMENT フィールドを更新します。オブジェクト・グループがスナップショット・グループの場合、このプロシージャは、スナップショット・サイトの DBA\_REPGROUP ビュー内の SCHEMA\_COMMENT フィールドを更新します。

このプロシージャは、マスター・サイトでもスナップショット・サイトでも実行できます。スナップショット・サイトでこのプロシージャを実行する場合は、スナップショット・グループの所有者が PUBLIC である必要があります。

**関連項目：** スナップショット・グループの所有者が PUBLIC でない場合にスナップショット・サイトの DBA\_REPGROUP ビュー内の SCHEMA\_COMMENT フィールドにコメントを配置する方法の詳細は、8-86 ページの「[COMMENT\\_ON\\_SNAPSHOT\\_REPSITES プロシージャ](#)」を参照してください。

### 構文

```
DBMS_REPCAT.COMMENT_ON_REPSITES (  
    gname          IN    VARCHAR2,  
    [ master       IN    VARCHAR2],  
    comment        IN    VARCHAR2);
```

### パラメータ

表 8-117 COMMENT\_ON\_REPSITES プロシージャのパラメータ

パラメータ	説明
gname	オブジェクト・グループの名前です。1 つのデータベースが複数のレプリケート環境でマスター・サイトになっている場合、このパラメータによって混乱が回避されます。
master	(オプション) コメントを付けるマスター・サイトの、完全修飾されたデータベース名です。マスター・サイトでこのプロシージャを実行する場合は、このパラメータが必要です。スナップショット・サイトでコメントを更新するには、このパラメータを省略します。
comment	適切なディクショナリ・ビューのコメント・フィールドに含める、更新されたコメントのテキストです。サイトがマスター・サイトの場合、DBA_REPSITES ビューの MASTER_COMMENT フィールドが更新されます。サイトがスナップショット・サイトの場合、DBA_REPGROUP ビューの SCHEMA_COMMENT フィールドが更新されます。

例外

表 8-118 COMMENT\_ON\_REPSITES プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
nonmaster	起動サイトはマスター・サイトではありません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
missingrepgroup	オブジェクト・グループは存在しません。
commfailure	1 つ以上のマスター・サイトにアクセスできません。
corrupt	レプリケーション・カタログ・ビューに不整合があります。

COMMENT\_ON\_SNAPSHOT\_REPSITES プロシージャ

このプロシージャは、指定されたスナップショット・グループの DBA\_REPGROUP データ・ディクショナリ・ビュー内の SCHEMA\_COMMENT フィールドを更新します。グループ名は、ローカルでレプリケート・スナップショット・グループとして登録する必要があります。このプロシージャは、スナップショット・サイトで実行する必要があります。

構文

```
DBMS_REPCAT.COMMENT_ON_SNAPSHOT_REPSITES (  
  gowner      IN   VARCHAR2,  
  gname       IN   VARCHAR2,  
  comment     IN   VARCHAR2);
```

パラメータ

表 8-119 COMMENT\_ON\_SNAPSHOT\_REPSITES プロシージャのパラメータ

パラメータ	説明
gowner	スナップショット・オブジェクト・グループの所有者です。
gname	スナップショット・オブジェクト・グループの名前です。
comment	DBA_REPGROUP ビューの SCHEMA_COMMENT フィールドに含める、更新されたコメントです。

表 8-120 COMMENT\_ON\_SNAPSHOT\_REPSITES プロシージャの例外

パラメータ	説明
missingrepgroup	スナップショット・オブジェクト・グループは存在しません。
nonsnapshot	接続サイトはスナップショット・サイトではありません。

## COMMENT\_ON\_conflicttype\_RESOLUTION プロシージャ

このプロシージャは、競合解消ルーチンの DBA\_REPRESOLUTION ビュー内のコメント・フィールドを更新します。コールするプロシージャは、ルーチンが解消する競合のタイプによって異なります。これらのプロシージャは、マスター定義サイトからコールする必要があります。

表 8-121 COMMENT\_ON\_conflicttype\_RESOLUTION プロシージャ

競合のタイプ	プロシージャ名
更新競合	COMMENT_ON_UPDATE_RESOLUTION
一意性競合	COMMENT_ON_UNIQUE_RESOLUTION
削除競合	COMMENT_ON_DELETE_RESOLUTION

コメントは、GENERATE\_REPLICATION\_SUPPORT への次のコールまで、どのマスター・サイトでも追加されません。

## 構文

```
DBMS_REPCAT.COMMENT_ON_UPDATE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    column_group   IN   VARCHAR2,
    sequence_no    IN   NUMBER,
    comment        IN   VARCHAR2);
```

```
DBMS_REPCAT.COMMENT_ON_UNIQUE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    constraint_name IN   VARCHAR2,
    sequence_no    IN   NUMBER,
    comment        IN   VARCHAR2);
```

```
DBMS_REPCAT.COMMENT_ON_DELETE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    sequence_no    IN   NUMBER,
    comment        IN   VARCHAR2);
```

パラメータ

表 8-122 COMMENT\_ON\_conflictttype\_RESOLUTION プロシージャのパラメータ

パラメータ	説明
sname	スキーマの名前です。
oname	競合解消ルーチンが対応付けられているレプリケート表の名前です。
column_group	更新競合解消ルーチンが対応付けられている列グループの名前です。
constraint_name	一意性競合解消ルーチンが対応付けられている一意性制約の名前です。
sequence_no	競合解消プロシージャの順序番号です。
comment	DBA_REPRESOLUTION ビューの RESOLUTION_COMMENT フィールドに含める、更新されたコメントのテキストです。

例外

表 8-123 COMMENT\_ON\_conflictttype\_RESOLUTION プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
missingresolution	指定された競合解消ルーチンは登録されていません。

COMPARE\_OLD\_VALUES プロシージャ

このプロシージャは、更新および削除のために、各マスター・サイトのレプリケート表の非キー列の古い列値を比較します。デフォルトでは、すべての列の古い値が比較されます。マスター定義サイトで DBMS\_REPCAT.COMPARE\_OLD\_VALUES を呼び出すことにより、すべてのマスター・サイトおよびスナップショット・サイトで、この動作を変更できます。

構文

```
DBMS_REPCAT.COMPARE_OLD_VALUES (
  sname          IN  VARCHAR2,
  oname          IN  VARCHAR2,
  { column_list  IN  VARCHAR2,
    | column_table IN  DBMS_REPCAT.VARCHAR2s, }
  operation      IN  VARCHAR2 := 'UPDATE',
  compare        IN  BOOLEAN := TRUE );
```



---

---

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと column\_table パラメータは、同時には指定できません。

---

---

## パラメータ

**表 8-124 COMPARE\_OLD\_VALUES プロシージャのパラメータ**

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	レプリケート表の名前です。
column_list	表の列の、カンマ区切りのリストです。各エントリの間にはスペースを挿入しません。
column_table	リストのかわりに、DBMS_REPCAT.VARCHAR2S タイプの PL/SQL 表を使用して列名を指定できます。最初の列名は 1 の位置、2 番目の列名は 2 の位置と、順に配置します。
operation	受け付けられる値は、UPDATE、DELETE、または更新と削除を意味するアスタリスク・ワイルドカード '*' です。
compare	値が TRUE に設定されている場合、指定された列の古い値は送信時に比較されます。値が FALSE に設定されている場合、指定された列の古い値は送信時に比較されません。指定されていない列や操作には影響がありません。表に対して min_communication が TRUE になると、指定された変更がマスター定義サイトで有効になります。min_communication が TRUE の状態で、マスター・サイトやスナップショット・サイトで次にレプリケーション・サポートが生成されるとき、変更が有効になります。

---

---

**注意：** operation パラメータを使用すると、行が削除されたとき、または非キー列が更新されたときに、非キー列の古い値を送信するかどうかを決定できます。古い値を渡さない場合、古い値のかわりに NULL が渡されます。その場合、古い値は更新または削除を行った時点でのターゲット・サイドの列のカレント値と見なされます。

Oracle のデフォルトの動作を変更する前に、削減されたデータ伝播の詳細を、『Oracle8i レプリケーション・ガイド』で参照してください。

---

---

例外

表 8-125 COMPARE\_OLD\_VALUES プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しません。
missingcolumn	少なくとも 1 つの列が、表に存在しません。
notquiesced	レプリケート・マスター・グループは停止していません。
typefailure	無効な操作が指定されました。

CREATE\_MASTER\_REPGROUP プロシージャ

このプロシージャは、新しい空の停止したマスター・レプリケーション・オブジェクト・グループを作成します。

構文

```
DBMS_REPCAT.CREATE_MASTER_REPGROUP (  
  gname          IN   VARCHAR2,  
  group_comment  IN   VARCHAR2      := '',  
  master_comment IN   VARCHAR2      := ''),  
  qualifier      IN   VARCHAR2      := '');
```

パラメータ

表 8-126 CREATE\_MASTER\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	作成するオブジェクト・グループの名前です。
group_comment	このコメントは、DBA_REPGROUP ビューに追加されます。
master_comment	このコメントは、DBA_REPSITES ビューに追加されます。
qualifier	オブジェクト・グループの接続修飾子です。@ 符号を必ず使用してください。接続修飾子の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

例外

表 8-127 CREATE\_MASTER\_REPGROUP プロシージャの例外

例外	説明
duplicaterepgroup	オブジェクト・グループはすでに存在しています。
norepopt	アドバンスト・レプリケーション・オプションがインストールされていません。
missingrepgroup	オブジェクト・グループの名前が指定されませんでした。
qualifiertoolong	接続修飾子が長すぎます。

CREATE\_MASTER\_REPOBJECT プロシージャ

このプロシージャは、オブジェクトをレプリケート・オブジェクトにします。

クラスタ化表のレプリケーションがサポートされていますが、クラスタ化表の場合は、`use_existing_object` パラメータを `FALSE` に設定できません。つまり、クラスタ化表は、マスター・グループに関連するすべてのマスター・サイトで、事前作成する必要があります。ただし、事前作成した表に表データを入れる必要はありません。このため、クラスタ化表の `copy_rows` パラメータを `TRUE` に設定できます。

構文

```
DBMS_REPCAT.CREATE_MASTER_REPOBJECT (  
    sname                IN    VARCHAR2,  
    oname                IN    VARCHAR2,  
    type                 IN    VARCHAR2,  
    use_existing_object  IN    BOOLEAN    := TRUE,  
    ddl_text             IN    VARCHAR2    := NULL,  
    comment              IN    VARCHAR2    := '',  
    retry                IN    BOOLEAN    := FALSE,  
    copy_rows            IN    BOOLEAN    := TRUE,  
    gname                IN    VARCHAR2    := '');
```

パラメータ

このプロシージャのパラメータを次の表に示します。

表 8-128 CREATE\_MASTER\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	レプリケートするオブジェクトが置かれたスキーマの名前です。
oname	レプリケートするオブジェクトの名前です。DDL_TEXT が NULL の場合、このオブジェクトは指定されたスキーマにすでに存在しています。一意性を確実にするため、表の名前は最長で 27 バイト、パッケージの名前は 24 バイト以内にします。
type	レプリケートするオブジェクトのタイプです。サポートされているタイプは、TABLE、INDEX、SYNONYM、TRIGGER、VIEW、PROCEDURE、FUNCTION、PACKAGE および PACKAGE BODY です。
use_existing_object	カレント・マスター・サイトで、同じタイプで同じ形式のオブジェクトを再利用する場合は、TRUE を指定します。詳細は、表 8-130 を参照してください。 <b>注意：</b> クラスタ化表の場合、このパラメータは TRUE に設定する必要があります。
ddl_text	オブジェクトがマスター定義サイトに存在していない場合は、このオブジェクトの作成に必要な DDL テキストを渡す必要があります。PL/SQL パッケージ、パッケージ本体、プロシージャおよびファンクションには、セミコロンが必要です。SQL 文の最後はセミコロンではありません。DDL は、適用の前に解析されることはありません。このため、DDL テキストには、作成されるオブジェクトについての適切なスキーマおよびオブジェクト名が含まれるようにする必要があります。  スキーマ (sname パラメータ) を指定せずに DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずスキーマを指定してください。
comment	このコメントは、DBA_REPOBJECT ビューの OBJECT_COMMENT フィールドに追加されます。
retry	前に作成できなかったオブジェクトの作成を再実行する場合は、TRUE を指定します。エラーが一時的なものであった場合、または修正された場合は、このパラメータを使用します。たとえば、リソースが不十分であったためにエラーが発生した場合などです。TRUE を指定すると、オブジェクト・ステータスが VALID でないマスター・サイトでのみ、オブジェクトが作成されます。
copy_rows	新しいレプリケート・オブジェクトの初期の内容を、マスター定義サイトのオブジェクトの内容と一致させる場合は、TRUE を指定します。詳細は、表 8-130 を参照してください。
gname	レプリケート・オブジェクトを作成するオブジェクト・グループの名前です。名前を指定しない場合は、スキーマの名前がデフォルトのオブジェクト・グループ名として使用されます。

表 8-129 CREATE\_MASTER\_REPOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	レプリケート・マスター・グループは中断されていません。
duplicateobject	指定されたオブジェクトは、レプリケート・マスター・グループにすでに存在しています。また、retry は FALSE です。名前が重複している場合も、この例外が発生します。
missingobject	SNAME および ONAME で識別されるオブジェクトは存在しません。また、適切な DDL が渡されていません。
typefailure	指定されたタイプのオブジェクトはレプリケートできません。
ddlfailure	マスター定義サイトの DDL が失敗しました。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
notcompat	7.3 以降のリリースとの互換性モードでないリモート・マスターがあります。

## オブジェクト作成

表 8-130 マスター・サイトでのオブジェクト作成

既存の オブ ジェク トの 有無	COPY_ROWS	USE_EXISTING_ OBJECTS	結果
有	TRUE	TRUE	オブジェクトが一致しない場合、duplicatedobject メッセージが生成されます。表の場合は、マスター定義サイトのデータを使用します。
有	FALSE	TRUE	オブジェクトが一致しない場合、duplicatedobject メッセージが生成されます。表の場合、データベース管理者は、内容を同一にする必要があります。
有	TRUE/FALSE	FALSE	duplicatedobject メッセージが生成されます。
無	TRUE	TRUE/FALSE	オブジェクトが作成されます。マスター定義サイトのデータを使用して、表が設定されます。
無	FALSE	TRUE/FALSE	オブジェクトが作成されます。データベース管理者は、表を設定し、すべてのサイトで表の整合性を保つようにする必要があります。

## CREATE\_SNAPSHOT\_REPGROUP プロシージャ

このプロシージャは、ローカル・データベースに、新しい空のスナップショット・グループを作成します。CREATE\_SNAPSHOT\_REPGROUP では REGISTER\_SNAPSHOT\_REPGROUP が自動的にコールされますが、登録中にエラーが発生しても無視されます。

### 構文

```
DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP (  
  gname          IN   VARCHAR2,  
  master         IN   VARCHAR2,  
  comment        IN   VARCHAR2      := ' ',  
  propagation_mode IN   VARCHAR2      := 'ASYNCHRONOUS',  
  fname         IN   VARCHAR2      := NULL,  
  gowner         IN   VARCHAR2      := 'PUBLIC');
```

### パラメータ

表 8-131 CREATE\_SNAPSHOT\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。このオブジェクト・グループは、指定されたマスター・サイトに存在している必要があります。
master	マスターとして使用するレプリケート環境のデータベースの、完全修飾されたデータベース名です。必要な場合は接続修飾子も指定できます。接続修飾子の使用方法の詳細は、『Oracle8i レプリケーション・ガイド』および『Oracle8i 分散システム』を参照してください。
comment	このコメントは、DBA_REPGROUP ビューに追加されます。
propagation_mode	オブジェクト・グループ内の、すべての更新可能なスナップショットのための伝播方法です。受け付けられる値は、SYNCHRONOUS および ASYNCHRONOUS です。
fname	このパラメータは内部で使用します。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。
gowner	スナップショット・グループの所有者です。

## 例外

**表 8-132 CREATE\_SNAPSHOT\_REPGROUP プロシージャの例外**

例外	説明
duplicaterepgroup	オブジェクト・グループは、起動サイトにすでに存在しています。
nonmaster	指定されたデータベースはマスター・サイトではありません。
commfailure	指定されたデータベースにはアクセスできません。
norepopt	アドバンスト・レプリケーション・オプションがインストールされていません。
typefailure	伝播モードが正しく指定されませんでした。
missingrepgroup	レプリケート・マスター・グループはマスター・サイトに存在しません。
invalidqualifier	マスター用に指定された接続修飾子は、オブジェクト・グループに対して無効です。
alreadymastered	ローカル・サイトに、グループ名が同じでマスター・サイトが異なる別のスナップショット・グループがあります。

## CREATE\_SNAPSHOT\_REPOBJECT プロシージャ

このプロシージャは、スナップショット・サイトにレプリケート・オブジェクトを追加します。

### 構文

```
DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    type           IN   VARCHAR2,
    ddl_text       IN   VARCHAR2  := '',
    comment        IN   VARCHAR2  := '',
    gname          IN   VARCHAR2  := '',
    gen_objs_owner IN   VARCHAR2  := '',
    min_communication IN   BOOLEAN  := TRUE ,
    generate_80_compatible IN   BOOLEAN  := TRUE ,
    gowner         IN   VARCHAR2  := 'PUBLIC');
```

パラメータ

表 8-133 CREATE\_SNAPSHOT\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	レプリケート・スナップショット・オブジェクト・グループに追加するオブジェクトの名前です。ONAME は、対応付けられたマスター・サイトに存在する必要があります。
type	レプリケートするオブジェクトのタイプです。スナップショット・サイトでサポートされるタイプは、PACKAGE、PACKAGE BODY、PROCEDURE、FUNCTION、SNAPSHOT、SYNONYM、TRIGGER および VIEW です。
ddl_text	<p>タイプ SNAPSHOT のオブジェクトの場合、オブジェクトを作成するために必要な DDL です。他のタイプの場合は、次のデフォルトを使用します。</p> <p>'' (空のストリング)</p> <p>同じ名前のスナップショットがすでに存在している場合、DDL は無視され、既存のスナップショットがレプリケート・オブジェクトとして登録されます。スナップショット用のマスター表が、このスキーマに指定されているマスター・サイトのレプリケート・マスター・グループに存在しない場合は、missingobject error. が発生します。</p> <p>スキーマを指定せずに DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずスキーマを指定してください。</p>
comment	このコメントは、DBA_REPOBJECT ビューの OBJECT_COMMENT フィールドに追加されます。
gname	オブジェクトを追加するレプリケート・マスター・グループの名前です。名前を指定しない場合は、スキーマの名前がデフォルトのグループ名として使用されます。
gen_objs_owner	トランザクションの所有者として割り当てるユーザーの名前です。
min_communication	マスター・サイトのいずれかで Oracle7 リリース 7.3 が実行されている場合は、FALSE に設定します。新旧の値の伝播を最小にするには、TRUE に設定します。デフォルトは TRUE です。競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。
generate_80_compatible	マスター・サイトのいずれかで Oracle8i リリース 8.1.5 より前の Oracle Server が実行されている場合は、TRUE に設定します。レプリケート環境が Oracle8i リリース 8.1.5 以降の場合は、FALSE に設定します。
gowner	スナップショット・グループの所有者です。



## 例外

**表 8-134 CREATE\_SNAPSHOT\_REPOBJECT プロシージャの例外**

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
nonmaster	マスターは、マスター・サイトではありません。
missingobject	指定されたオブジェクトは、マスターのレプリケート・マスター・グループに存在しません。
duplicateobject	指定されたオブジェクトは、異なる形式ですすでに存在しています。
typefailure	許可されないタイプです。
ddlfailure	DDL が失敗しました。
commfailure	マスター・サイトにアクセスできません。
missingschema	スキーマは、データベース・スキーマとして存在していません。
badsnapddl	DDL は実行されましたが、スナップショットが存在しません。
onlyonesnap	マスター表のスナップショットは1つしか作成できません。
badsnapname	スナップショットのベース表がマスター表と異なります。
missingrepgroup	レプリケート・マスター・グループは存在しません。

## DEFINE\_COLUMN\_GROUP プロシージャ

このプロシージャは、空の列グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

## 構文

```
DBMS_REPCAT.DEFINE_COLUMN_GROUP (
  sname          IN   VARCHAR2,
  oname          IN   VARCHAR2,
  column_group   IN   VARCHAR2,
  comment        IN   VARCHAR2 := NULL);
```

パラメータ

表 8-135 DEFINE\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	列グループを作成するレプリケート表の名前です。
column_group	作成する列グループの名前です。
comment	このユーザー・コメントは、DBA_REPCOLUMN_GROUP ビューに表示されます。

例外

表 8-136 DEFINE\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定された表は存在しません。
duplicategroup	指定された列グループはこの表にすでに存在しています。
notquiesced	指定された表が所属するオブジェクト・グループは停止していません。

DEFINE\_PRIORITY\_GROUP プロシージャ

このプロシージャは、レプリケート・マスター・グループの新しい優先グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.DEFINE_PRIORITY_GROUP (  
  gname          IN   VARCHAR2,  
  pgroup         IN   VARCHAR2,  
  datatype       IN   VARCHAR2,  
  fixed_length  IN   INTEGER := NULL,  
  comment        IN   VARCHAR2 := NULL);
```

パラメータ

表 8-137 DEFINE\_PRIORITY\_GROUP プロシージャのパラメータ

パラメータ	説明
gname	優先グループを作成するレプリケート・マスター・グループの名前です。
pgroup	作成する優先グループの名前です。
datatype	優先グループ・メンバーのデータ型です。サポートされているデータ型は、CHAR、VARCHAR2、NUMBER、DATE、RAW、NCHAR および NVARCHAR2 です。
fixed_length	データ型 CHAR の列の長さを指定する必要があります。他のすべてのデータ型では、デフォルトの NULL を使用できます。
comment	このユーザー・コメントは、DBA_REPPRIORITY ビューに追加されます。

例外

表 8-138 DEFINE\_PRIORITY\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
duplicateprioritygroup	指定された優先グループは、レプリケート・マスター・グループにすでに存在しています。
typefailure	指定されたデータ型はサポートされていません。
notquiesced	レプリケート・マスター・グループは停止していません。

DEFINE\_SITE\_PRIORITY プロシージャ

このプロシージャは、レプリケート・マスター・グループの新しいサイト優先順位グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.DEFINE_SITE_PRIORITY (  
    gname          IN   VARCHAR2,  
    name           IN   VARCHAR2,  
    comment        IN   VARCHAR2 := NULL);
```

パラメータ

表 8-139 DEFINE\_SITE\_PRIORITY プロシージャのパラメータ

パラメータ	説明
gname	サイト優先順位グループを作成するレプリケート・マスター・グループの名前です。
name	作成するサイト優先順位グループの名前です。
comment	このユーザー・コメントは、DBA_REPPRIORITY ビューに追加されます。

例外

表 8-140 DEFINE\_SITE\_PRIORITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
duplicate prioritygroup	指定されたサイト優先順位グループは、レプリケート・マスター・グループにすでに存在しています。
notquiesced	レプリケート・マスター・グループは停止していません。

DO\_DEFERRED\_REPCAT\_ADMIN プロシージャ

このプロシージャは、カレント・マスター・サイトの指定されたレプリケート・マスター・グループ、またはすべてのマスター・サイトに対して、ローカルな遅延管理プロシージャを実行します。すべてのマスター・サイトに対する遅延管理プロシージャは、ジョブ・キューを利用して実行されます。

DO\_DEFERRED\_REPCAT\_ADMIN では、DO\_DEFERRED\_REPCAT\_ADMIN をコールした接続ユーザーが発行した管理要求のみが実行されます。他のユーザーが発行した要求は無視されます。

## 構文

```
DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN (  
    gname          IN   VARCHAR2,  
    all_sites      IN   BOOLEAN := FALSE);
```

## パラメータ

表 8-141 DO\_DEFERRED\_REPCAT\_ADMIN プロシージャのパラメータ

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。
all_sites	値が TRUE の場合は、ジョブを使用して、各マスター・サイトでローカルな管理プロシージャが実行されます。

## 例外

表 8-142 DO\_DEFERRED\_REPCAT\_ADMIN プロシージャの例外

例外	説明
nonmaster	起動サイトはマスター・サイトではありません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。all_sites は TRUE に設定されています。

## DROP\_COLUMN\_GROUP プロシージャ

このプロシージャは、列グループを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

## 構文

```
DBMS_REPCAT.DROP_COLUMN_GROUP (  
    sname          IN   VARCHAR2,  
    oname          IN   VARCHAR2,  
    column_group IN   VARCHAR2);
```

パラメータ

表 8-143 DROP\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	列グループを削除するレプリケート表の名前です。
column_group	削除対象の列グループの名前です。

例外

表 8-144 DROP\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
referenced	指定された列グループは、競合の検出および解消に使用されていま す。
missingobject	指定された表は存在しません。
missinggroup	指定された列グループは存在しません。
notquiesced	表が所属するレプリケート・マスター・グループは停止していませ ん。

DROP\_GROUPED\_COLUMN プロシージャ

このプロシージャは、列グループからメンバーを削除します。このプロシージャは、マ  
スター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイ  
ド』を参照してください。

構文

```
DBMS_REPCAT.DROP_GROUPED_COLUMN (  
  sname          IN   VARCHAR2,  
  oname          IN   VARCHAR2,  
  column_group   IN   VARCHAR2,  
  list_of_column_names IN VARCHAR2 | DBMS_REPCAT.VARCHAR2s);
```

## パラメータ

**表 8-145 DROP\_GROUPED\_COLUMN プロシージャのパラメータ**

パラメータ	説明
aname	レプリケート表が置かれたスキーマの名前です。
oname	列グループが置かれたレプリケート表の名前です。
column_group	メンバーを削除する列グループの名前です。
list_of_column_names	指定された列グループから削除する列の名前です。列名は、カンマ区切りのリストと PL/SQL 表のいずれでも指定できます。PL/SQL 表のタイプは DBMS_REPCAT.VARCHAR2 にする必要があります。

## 例外

**表 8-146 DROP\_GROUPED\_COLUMN プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定された表は存在しません。
notquiesced	表が所属するレプリケート・マスター・グループは停止していません。

## DROP\_MASTER\_REPGROUP プロシージャ

このプロシージャは、カレント・サイトからレプリケート・マスター・グループを削除します。レプリケート・マスター・グループを、マスター定義サイトを含むすべてのマスター・サイトから削除するには、このプロシージャをマスター定義サイトでコールして、最後の引数を TRUE に設定します。

## 構文

```
DBMS_REPCAT.DROP_MASTER_REPGROUP (
    gname          IN VARCHAR2,
    drop_contents  IN BOOLEAN    := FALSE,
    all_sites      IN BOOLEAN    := FALSE);
```

パラメータ

表 8-147 DROP\_MASTER\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	カレント・マスター・サイトから削除するレプリケート・マスター・グループの名前です。
drop_contents	デフォルトでは、マスター・サイトのオブジェクト・グループを削除しても、すべてのオブジェクトはデータベースに残ります。ただし、これらのオブジェクトはレプリケートされなくなります。すなわち、オブジェクト・グループ内のレプリケート・オブジェクトは、他のマスター・サイトとの間で変更をやり取りしなくなります。値を TRUE に設定すると、レプリケート・マスター・グループ内のレプリケート・オブジェクトが、対応付けられたスキーマから削除されます。
all_sites	値が TRUE に設定され、起動サイトがマスター定義サイトの場合、このプロシージャにより要求がすべてのマスターに同期的にマルチキャストされます。この場合、マスター定義サイトではプロシージャが即座に実行されますが、他のマスター・サイトでは実行が遅れることがあります。

例外

表 8-148 DROP\_MASTER\_REPGROUP プロシージャの例外

例外	説明
nonmaster	起動サイトはマスター・サイトではありません。
nonmasterdef	起動サイトはマスター定義サイトではありません。ALL_SITES は TRUE に設定されています。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。ALL_SITES は TRUE に設定されています。
fullqueue	遅延 RPC キューに、レプリケート・マスター・グループのエントリがあります。
masternotremoved	マスターはマスター定義サイトを認識しません。ALL_SITES は TRUE に設定されています。



## DROP\_MASTER\_REOBJECT プロシージャ

このプロシージャは、レプリケート・マスター・グループからレプリケート・オブジェクトを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.DROP_MASTER_REOBJECT (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    type           IN   VARCHAR2,
    drop_objects   IN   BOOLEAN    := FALSE);
```

### パラメータ

表 8-149 DROP\_MASTER\_REOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	レプリケート・マスター・グループから削除するオブジェクトの名前です。
type	削除対象のオブジェクトのタイプです。
drop_objects	デフォルトでは、オブジェクトはスキーマには残りますが、レプリケート・マスター・グループからは削除されます。すなわち、オブジェクトに対する変更が、他のマスター・サイトやスナップショット・サイトにレプリケートされなくなります。レプリケート環境からオブジェクトを完全に削除するには、このパラメータを TRUE に設定します。パラメータを TRUE に設定すると、オブジェクトは各マスター・サイトのデータベースから削除されます。

### 例外

表 8-150 DROP\_MASTER\_REOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

## DROP\_PRIORITY プロシージャ

このプロシージャは、優先順位レベルに基づいて、優先グループのメンバーを削除します。  
このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

### 構文

```
DEMS_REPCAT.DROP_PRIORITY(  
    gname          IN   VARCHAR2,  
    pgroup         IN   VARCHAR2,  
    priority_num   IN   NUMBER);
```

### パラメータ

表 8-151 DROP\_PRIORITY プロシージャのパラメータ

パラメータ	説明
gname	優先グループが対応付けられたレプリケート・マスター・グループの名前です。
pgroup	削除対象のメンバーが含まれる優先グループの名前です。
priority_num	グループから削除する優先グループ・メンバーの優先順位レベルです。

### 例外

表 8-152 DROP\_PRIORITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
notquiesced	レプリケート・マスター・グループは停止していません。

## DROP\_PRIORITY\_GROUP プロシージャ

このプロシージャは、指定されたレプリケート・マスター・グループの優先グループを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

### 構文

```
DBMS_REPCAT.DROP_PRIORITY_GROUP (  
    gname      IN   VARCHAR2,  
    pgroup     IN   VARCHAR2);
```

### パラメータ

表 8-153 DROP\_PRIORITY\_GROUP プロシージャのパラメータ

パラメータ	説明
gname	優先グループが対応付けられたレプリケート・マスター・グループの名前です。
pgroup	削除対象の優先グループの名前です。

### 例外

表 8-154 DROP\_PRIORITY\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
referenced	指定された優先グループは、競合解消に使用されています。
notquiesced	指定されたレプリケート・マスター・グループは停止していません。

## DROP\_PRIORITY\_datatype プロシージャ

このプロシージャは、値に基づいて、優先グループのメンバーを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、priority 列のデータ型によって異なります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

### 構文

```
DBMS_REPCAT.DROP_PRIORITY_datatype (  
    gname      IN   VARCHAR2,  
    pgroup     IN   VARCHAR2,  
    value      IN   datatype);
```

datatype は次のとおりです。

```
{ NUMBER  
| VARCHAR2  
| CHAR  
| DATE  
| RAW  
| NCHAR  
| NVARCHAR2 }
```

### パラメータ

表 8-155 DROP\_PRIORITY\_datatype プロシージャのパラメータ

パラメータ	説明
gname	優先グループが対応付けられたレプリケート・マスター・グループの名前です。
pgroup	削除対象のメンバーが含まれる優先グループの名前です。
value	グループから削除する優先グループ・メンバーの値です。

## 例外

**表 8-156 DROP\_PRIORITY\_datatype プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
paramtype, typefailure	値のデータ型は、この優先グループには不適切です。
notquiesced	指定されたレプリケート・マスター・グループは停止していません。

## DROP\_SITE\_PRIORITY プロシージャ

このプロシージャは、指定されたレプリケート・マスター・グループのサイト優先順位グループを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

## 構文

```
DBMS_REPCAT.DROP_SITE_PRIORITY (
    gname      IN   VARCHAR2,
    name       IN   VARCHAR2);
```

## パラメータ

**表 8-157 DROP\_SITE\_PRIORITY プロシージャのパラメータ**

パラメータ	説明
gname	サイト優先順位グループが対応付けられたレプリケート・マスター・グループの名前です。
name	削除対象のサイト優先順位グループの名前です。

例外

表 8-158 DROP\_SITE\_PRIORITY\_SITE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
referenced	指定されたサイト優先順位グループは、競合解消に使用されていま す。
notquiesced	指定されたレプリケート・マスター・グループは停止していません。

DROP\_SITE\_PRIORITY\_SITE プロシージャ

このプロシージャは、指定された名前のサイトをサイト優先順位グループから削除します。  
このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.DROP_SITE_PRIORITY_SITE (  
  gname      IN   VARCHAR2,  
  name       IN   VARCHAR2,  
  site       IN   VARCHAR2);
```

パラメータ

表 8-159 DROP\_SITE\_PRIORITY\_SITE プロシージャのパラメータ

パラメータ	説明
gname	サイト優先順位グループが対応付けられたレプリケート・マスター・ グループの名前です。
name	メンバーを削除するサイト優先順位グループの名前です。
site	グループから削除するサイトのグローバル・データベース名です。

例外

表 8-160 DROP\_SITE\_PRIORITY\_SITE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたレプリケート・マスター・グループは存在しません。
missingpriority	指定されたサイト優先順位グループは存在しません。
notquiesced	指定されたレプリケート・マスター・グループは停止していません。

DROP\_SNAPSHOT\_REPGROUP プロシージャ

このプロシージャは、レプリケート環境からスナップショット・サイトを削除します。DROP\_SNAPSHOT\_REPGROUP では、UNREGISTER\_SNAPSHOT\_REPGROUP が自動的にコールされ、スナップショットの登録が解除されます。ただし、登録の解除中にエラーが発生しても無視されます。

構文

```
DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP (  
    gname                IN    VARCHAR2,  
    drop_contents         IN    BOOLEAN    := FALSE,  
    gowner                IN    VARCHAR2    := 'PUBLIC');
```

パラメータ

表 8-161 DROP\_SNAPSHOT\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	カレント・スナップショット・サイトから削除するレプリケート・マスター・グループの名前です。トリガーやパッケージなど、レプリケーションのサポート用に生成されたすべてのオブジェクトが削除されます。
drop_contents	デフォルトでは、スナップショット・サイトのレプリケート・マスター・グループを削除しても、すべてのオブジェクトは対応付けられたスキーマに残ります。ただし、これらのオブジェクトはレプリケートされなくなります。値を TRUE に設定すると、レプリケート・マスター・グループ内のレプリケート・オブジェクトが、対応付けられたスキーマから削除されます。
gowner	スナップショット・グループの所有者です。

例外

表 8-162 DROP\_SNAPSHOT\_REPGROUP プロシージャの例外

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
missingrepgroup	指定されたオブジェクト・グループは存在しません。

DROP\_SNAPSHOT\_REPOBJECT プロシージャ

このプロシージャは、スナップショット・サイトからレプリケート・オブジェクトを削除します。

構文

```
DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT (  
  sname          IN   VARCHAR2,  
  oname          IN   VARCHAR2,  
  type           IN   VARCHAR2,  
  drop_objects   IN   BOOLEAN  := FALSE);
```

パラメータ

表 8-163 DROP\_SNAPSHOT\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	レプリケート・マスター・グループから削除するオブジェクトの名前です。
type	削除対象のオブジェクトのタイプです。
drop_objects	デフォルトでは、オブジェクトは対応付けられたスキーマには残りますが、対応付けられたオブジェクト・グループからは削除されます。カレント・スナップショット・サイトのスキーマからオブジェクトを完全に削除するには、この引数を TRUE に設定します。パラメータを TRUE に設定すると、オブジェクトはスナップショット・サイトのデータベースから削除されます。



## 例外

**表 8-164 DROP\_SNAPSHOT\_REOBJECT プロシージャの例外**

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。

## DROP\_conflicttype\_RESOLUTION プロシージャ

このプロシージャは、更新、削除または一意性の競合解消ルーチンを削除します。これらのプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、ルーチンが解消する競合のタイプによって異なります。

## パラメータ

**表 8-165 DROP\_conflicttype\_RESOLUTION プロシージャのパラメータ**

パラメータ	説明
update	DROP_UPDATE_RESOLUTION
uniqueness	DROP_UNIQUE_RESOLUTION
delete	DROP_UNIQUE_RESOLUTION

## 構文

```
DBMS_REPCAT.DROP_UPDATE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    column_group   IN   VARCHAR2,
    sequence_no    IN   NUMBER);
```

```
DBMS_REPCAT.DROP_DELETE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    sequence_no    IN   NUMBER);
```

```
DBMS_REPCAT.DROP_UNIQUE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    constraint_name IN   VARCHAR2,
    sequence_no    IN   NUMBER);
```

パラメータ

表 8-166 DROP\_conflicttype\_RESOLUTION プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	競合解消ルーチンを削除する表の名前です。
column_group	更新競合解消ルーチンを削除する列グループの名前です。
constraint_name	一意性競合解消ルーチンを削除する一意性制約の名前です。
sequence_no	削除対象の競合解消方法に割り当てられた順序番号です。この番号でルーチンが識別されます。

例外

表 8-167 DROP\_conflicttype\_RESOLUTION プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトが指定されたスキーマ内の表として存在しないか、または指定された順序番号の競合解消ルーチンが登録されていません。
notquiesced	レプリケート・マスター・グループは停止していません。

EXECUTE\_DDL プロシージャ

このプロシージャは、いくつかの、またはすべてのマスター・サイトで実行される DDL を渡します。このプロシージャは、マスター定義サイトからのみコールできます。

構文

```
DBMS_REPCAT.EXECUTE_DDL (
  gname          IN   VARCHAR2,
  { master_list  IN   VARCHAR2      := NULL,
    | master_table IN   DBMS_UTILITY.DBLINK_ARRAY, }
  DDL_TEXT       IN   VARCHAR2);
```

**注意：** このプロシージャはオーバーロードされています。MASTER\_LIST パラメータと MASTER\_TABLE パラメータは、同時には指定できません。

## パラメータ

**表 8-168 EXECUTE\_DDL プロシージャのパラメータ**

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。
master_list	渡された DDL を実行するマスター・サイトの、カンマ区切りのリストです。各サイト名の間にはスペースを挿入しません。デフォルトの NULL の場合、マスター定義サイトを含むすべてのサイトで DDL が実行されます。
master_table	渡された DDL を実行するマスター・サイトの表です。最初のマスターは 1 の位置、2 番目のマスターは 2 の位置と、順に配置します。
ddl_text	指定された各マスター・サイトで実行する DDL です。スキーマを指定せずに DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずスキーマを指定してください。

## 例外

**表 8-169 EXECUTE\_DDL プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
nonmaster	少なくとも 1 つのサイトが、マスター・サイトではありません。
ddlfailure	マスター定義サイトの DDL が失敗しました。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

## GENERATE\_REPLICATION\_SUPPORT プロシージャ

このプロシージャは、レプリケーションのサポートに必要なトリガーおよびパッケージを生成します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
  sname                IN    VARCHAR2,
  oname                IN    VARCHAR2,
  type                 IN    VARCHAR2,
  package_prefix       IN    VARCHAR2    := NULL,
  procedure_prefix     IN    VARCHAR2    := NULL,
  distributed          IN    BOOLEAN     := TRUE,
  gen_objs_owner       IN    VARCHAR2    := NULL,
  min_communication    IN    BOOLEAN     := TRUE,
  generate_80_compatible IN    BOOLEAN     := TRUE);
```

### パラメータ

表 8-170 GENERATE\_REPLICATION\_SUPPORT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	レプリケーション・サポートを生成するオブジェクトの名前です。
type	オブジェクトのタイプです。サポートされているタイプは、TABLE、PACKAGE および PACKAGE BODY です。
package_prefix	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合は、生成されたラッパー・パッケージの名前にこの値が付加されます。デフォルトは DEFER_ です。
procedure_prefix	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合は、生成されたラッパー・プロシージャの名前にこの値が付加されます。デフォルトでは接頭辞が付加されません。
distributed	このパラメータは、必ず TRUE に設定します。
gen_objs_owner	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合、オブジェクトが作成されるスキーマです。値が NULL の場合、オブジェクトは SNAME に作成されます。
min_communication	マスター・サイトのいずれかで Oracle7 リリース 7.3 が実行されている場合は、FALSE に設定します。新旧の値の伝播を最小にするには、TRUE に設定します。デフォルトは TRUE です。詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

表 8-170 GENERATE\_REPLICATION\_SUPPORT プロシージャのパラメータ

パラメータ	説明
generate_80_compatible	マスター・サイトのいずれかで Oracle8i リリース 8.1.5 より前の Oracle Server が実行されている場合は、TRUE に設定します。レプリケート環境が Oracle8i リリース 8.1.5 以降の場合は、FALSE に設定します。

## 例外

表 8-171 GENERATE\_REPLICATION\_SUPPORT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しないか、またはラッパーの生成が必要なパッケージ（本体）として存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
notquiesced	レプリケート・マスター・グループは停止していません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
missingschema	スキーマが存在しません。
dbnotcompatible	リリース 7.3.0.0 との互換性がないマスターがあります。
notcompat	リリース 7.3.0.0 との互換性がないマスターがあります。（dbnotcompatible と同様です。）
duplicateobject	オブジェクトはすでに存在しています。

## GENERATE\_SNAPSHOT\_SUPPORT プロシージャ

このプロシージャは、トリガーをアクティブにして、更新可能なスナップショットのレプリケーション、またはプロシージャ型レプリケーションのサポートに必要なパッケージを生成します。このプロシージャは、スナップショット・サイトからコールする必要があります。

---

**注意：** CREATE\_SNAPSHOT\_REPOBJECT では、更新可能なスナップショットのスナップショット・サポートが自動的に生成されます。

---

構文

```
DBMS_REPCAT.GENERATE_SNAPSHOT_SUPPORT (  
    sname                IN VARCHAR2,  
    oname                IN VARCHAR2,  
    type                IN VARCHAR2,  
    gen_objs_owner       IN VARCHAR2 := '',  
    min_communication    IN BOOLEAN  := TRUE,  
    generate_80_compatible IN BOOLEAN := TRUE);
```

パラメータ

表 8-172 GENERATE\_SNAPSHOT\_SUPPORT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	サポートを生成するオブジェクトの名前です。
type	オブジェクトのタイプです。サポートされているタイプは、SNAPSHOT、PACKAGE および PACKAGE BODY です。
gen_objs_owner	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合、オブジェクトが作成されるスキーマです。値が NULL の場合、オブジェクトは SNAME に作成されます。
min_communication	値が TRUE の場合、UPDATE 文で列が変更されるときに限って、更新トリガーで列の新しい値が送信されます。列がキー列の場合、または変更された列グループ内の列の場合は、更新トリガーで列の古い値が送信されます。
generate_80_compatible	マスター・サイトのいずれかで Oracle8i リリース 8.1.5 より前の Oracle Server が実行されている場合は、TRUE に設定します。レプリケート環境が Oracle8i リリース 8.1.5 以降の場合は、FALSE に設定します。

例外

表 8-173 GENERATE\_SNAPSHOT\_SUPPORT プロシージャの例外 (1 / 2 ページ)

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
missingobject	指定されたオブジェクトは、行または列レベル・レプリケーションの情報が必要なレプリケート・スキーマ内のスナップショットとして存在しないか、またはラッパーの生成が必要なパッケージ（本体）として存在しません。

表 8-173 GENERATE\_SNAPSHOT\_SUPPORT プロシージャの例外 (2 / 2 ページ)

例外	説明
typfailure	指定されたタイプ・パラメータはサポートされていません。
missingschema	指定された生成オブジェクトの所有者は存在しません。
missingremoteobject	マスター・オブジェクトでは、レプリケーション・サポートがまだ生成されていません。
commfailure	マスターにアクセスできません。

MAKE\_COLUMN\_GROUP プロシージャ

このプロシージャは、1 つ以上のメンバーを持つ新しい列グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消方法の詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.MAKE_COLUMN_GROUP (  
    sname                IN   VARCHAR2,  
    oname                IN   VARCHAR2,  
    column_group         IN   VARCHAR2,  
    list_of_column_names IN   VARCHAR2 | DBMS_REPCAT.VARCHAR2s);
```

パラメータ

表 8-174 MAKE\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	新しい列グループを作成するレプリケート表の名前です。
column_group	作成する列グループに割り当てる名前です。
list_of_column_names	グループ化する列の名前です。列名は、カンマ区切りのリストと PL/SQL 表のいずれでも指定できます。PL/SQL 表のタイプは DBMS_REPCAT.VARCHAR2 にする必要があります。表のすべての列を含む列グループを作成するには、単一の値 '*' を使用します。

例外

表 8-175 MAKE\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicategroup	指定された列グループはこの表にすでに存在しています。
missingobject	指定された表は存在しません。
missingcolumn	指定された列は、指定された表に存在しません。
duplicatecolumn	指定された列は、すでに他の列グループのメンバーになっています。
notquiesced	レプリケート・マスター・グループは停止していません。

PURGE\_MASTER\_LOG プロシージャ

このプロシージャは、指定された ID 番号、ソースまたはレプリケート・マスター・グループに対応付けられた、DBA\_REPCATLOG ビュー内のローカル・メッセージを削除します。

構文

```
DBMS_REPCAT.PURGE_MASTER_LOG (  
    id      IN   BINARY_INTEGER,  
    source  IN   VARCHAR2,  
    gname   IN   VARCHAR2);
```

パラメータ

表 8-176 PURGE\_MASTER\_LOG プロシージャのパラメータ

パラメータ	説明
id	DBA_REPCATLOG ビューに表示される要求の ID 番号です。
source	要求が発行されたマスター・サイトです。
gname	要求の対象であるレプリケート・マスター・グループの名前です。

例外

表 8-177 PURGE\_MASTER\_LOG プロシージャの例外

例外	説明
nonmaster	gname が NULL でなく、起動サイトがマスター・サイトではありません。



# PURGE\_STATISTICS プロシージャ

このプロシージャは、DBA\_REPRESOLUTION\_STATISTICS ビューから情報を削除します。

## 構文

```
DBMS_REPCAT.PURGE_STATISTICS (  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2,  
    start_date IN   DATE,  
    end_date   IN   DATE);
```

## パラメータ

表 8-178 PURGE\_STATISTICS プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	競合解消の統計をパージする表の名前です。
start_date/end_date	統計をパージする日付の範囲です。START_DATE が NULL の場合は、END_DATE までのすべての統計がパージされます。END_DATE が NULL の場合は、START_DATE 以後のすべての統計がパージされます。

## 例外

表 8-179 PURGE\_STATISTICS プロシージャの例外

例外	説明
missingschema	指定されたスキーマは存在しません。
missingobject	指定された表は存在しません。
statnotreg	統計の収集に表が登録されていません。

## REFRESH\_SNAPSHOT\_REPGROUP プロシージャ

このプロシージャは、スナップショット・サイト・オブジェクト・グループを、対応付けられたマスター・サイトの最新データにリフレッシュします。

### 構文

```
DBMS_REPCAT.REFRESH_SNAPSHOT_REPGROUP (  
    gname                IN    VARCHAR2,  
    drop_missing_contents IN    BOOLEAN    := FALSE,  
    refresh_snapshots     IN    BOOLEAN    := FALSE,  
    refresh_other_objects IN    BOOLEAN    := FALSE,  
    gowner                IN    VARCHAR2    := 'PUBLIC');
```

### パラメータ

表 8-180 REFRESH\_SNAPSHOT\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。
drop_missing_contents	オブジェクトがレプリケート・マスター・グループから削除されても、スナップショット・サイトのスキーマからは自動的に削除されません。このようなオブジェクトは単に、レプリケートされなくなります。すなわち、オブジェクトに加えられた変更が、対応付けられたマスター・サイトに送信されなくなります。このような場合、スナップショットを、対応付けられたマスター表からリフレッシュできます。ただし、更新可能なスナップショットに加えられた変更は失われます。オブジェクトがオブジェクト・グループから削除されるとき、この引数を TRUE に設定すると、そのオブジェクトはスキーマから完全に削除されます。
refresh_snapshots	値を TRUE に設定すると、レプリケート・マスター・グループ内のスナップショットの内容がリフレッシュされます。
refresh_other_objects	値を TRUE に設定すると、レプリケート・マスター・グループ内の非スナップショット・オブジェクトの内容がリフレッシュされます。
gowner	スナップショット・グループの所有者です。

## 例外

**表 8-181 REFRESH\_SNAPSHOT\_REPGROUP プロシージャの例外**

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
nonmaster	指定されたマスターは、マスター・サイトではありません。
commfailure	マスターにアクセスできません。
missingrepgroup	オブジェクト・グループの名前が指定されていません。

## REGISTER\_SNAPSHOT\_REPGROUP プロシージャ

このプロシージャは、DBA\_REGISTERED\_SNAPSHOT\_GROUPS でのスナップショット・グループの挿入または変更を行って、マスター・サイトでのスナップショットの管理を容易にします。

## 構文

```
DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP (
    gname          IN   VARCHAR2,
    snapsite       IN   VARCHAR2,
    comment        IN   VARCHAR2  := NULL,
    rep_type       IN   NUMBER     := reg_unknown,
    fname          IN   VARCHAR2  := NULL,
    gowner         IN   VARCHAR2  := 'PUBLIC');
```

## パラメータ

**表 8-182 REGISTER\_SNAPSHOT\_REPGROUP プロシージャのパラメータ (1 / 2 ページ)**

パラメータ	説明
gname	登録するスナップショット・オブジェクト・グループの名前です。
snapsite	スナップショット・サイトのグローバル名です。
comment	スナップショット・サイトのコメント、または既存のコメントの更新です。
rep_type	スナップショット・グループのバージョンです。使用できる定数は、reg_unknown (デフォルト)、reg_v7_group、reg_v8_group および reg_repapi_group です。

表 8-182 REGISTER\_SNAPSHOT\_REPGROUP プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
fname	このパラメータは内部で使用します。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。
gowner	スナップショット・グループの所有者です。

例外

表 8-183 REGISTER\_SNAPSHOT\_REPGROUP プロシージャの例外

例外	説明
failregsnaprepgroup	スナップショット・グループの登録に失敗しました。
missingrepgroup	オブジェクト・グループの名前が指定されていません。
nullsitename	スナップショット・サイトが指定されていません。
nonmaster	プロシージャは、スナップショットのマスター・サイトで実行する必要があります。
duplicaterepgroup	オブジェクトはすでに存在しています。

REGISTER\_STATISTICS プロシージャ

このプロシージャは、正常に実行された、表の更新、一意性および削除の競合解消の情報を収集します。

構文

```
DBMS_REPCAT.REGISTER_STATISTICS (  
    sname IN   VARCHAR2,  
    oname IN   VARCHAR2);
```

パラメータ

表 8-184 REGISTER\_STATISTICS プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	競合解消の統計を収集する表の名前です。

例外

表 8-185 REGISTER\_STATISTICS プロシージャの例外

例外	説明
missingschema	指定されたスキーマは存在しません。
missingobject	指定された表は存在しません。

RELOCATE\_MASTERDEF プロシージャ

このプロシージャは、マスター定義サイトをレプリケート環境の他のマスター・サイトに変更します。

RELOCATE\_MASTERDEF をコールするときに、古いマスター定義サイトまたは新しいマスター定義サイトが使用可能である必要はありません。再構成を行うときには、NOTIFY\_MASTERS および INCLUDE\_OLD\_MASTERDEF を TRUE に設定して、RELOCATE\_MASTERDEF をコールします。

構文

```
DBMS_REPCAT.RELOCATE_MASTERDEF (  
  gname                IN   VARCHAR2,  
  old_masterdef         IN   VARCHAR2,  
  new_masterdef         IN   VARCHAR2,  
  notify_masters        IN   BOOLEAN    := TRUE,  
  include_old_masterdef IN   BOOLEAN    := TRUE,  
  require_flavor_change IN   BOOLEAN    := FALSE);
```

パラメータ

表 8-186 RELOCATE\_MASTERDEF プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
gname	マスター定義を再配置するオブジェクト・グループの名前です。
old_masterdef	カレント・マスター定義サイトの、完全修飾されたデータベース名です。
new_masterdef	新しいマスター定義サイトにする既存のマスター・サイトの、完全修飾されたデータベース名です。

表 8-186 RELOCATE\_MASTERDEF プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
notify_masters	TRUE の場合、すべてのマスター（INCLUDE_OLD_MASTERDEF が TRUE の場合には OLD_MASTERDEF も含まれます）に、変更が同期的にマルチキャストされます。いずれかのマスターで変更が適用されなかった場合は、すべてのマスターでその変更がロールバックされます。  マスター定義サイトのみに障害が発生している場合は、NOTIFY_MASTERS を TRUE、INCLUDE_OLD_MASTERDEF を FALSE に設定して、RELOCATE_MASTERDEF をコールします。複数のマスター・サイトとマスター定義サイトで障害が発生した場合、それぞれの操作マスターで、管理者が NOTIFY_MASTERS を FALSE に設定して RELOCATE_MASTERDEF をコールします。
include_old_masterdef	NOTIFY_MASTERS が TRUE で、INCLUDE_OLD_MASTERDEF も TRUE の場合は、古いマスター定義サイトにも変更が通知されます。
require_flavor_change	このパラメータは内部で使用します。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

例外

表 8-187 RELOCATE\_MASTERDEF プロシージャの例外

例外	説明
nonmaster	NEW_MASTERDEF または起動サイトがマスター・サイトではありません。
nonmasterdef	OLD_MASTERDEF はマスター定義サイトではありません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。NOTIFY_MASTERS は TRUE に設定されています。

REMOVE\_MASTER\_DATABASES プロシージャ

このプロシージャは、レプリケート環境から 1 つ以上のマスター・データベースを削除します。このプロシージャは、残りのマスター・サイトで、トリガーとそれらに対応付けられたパッケージを再生成します。このプロシージャは、マスター定義サイトからコールする必要があります。

## 構文

```
DBMS_REPCAT.REMOVE_MASTER_DATABASES (
    gname          IN   VARCHAR2,
    master_list     IN   VARCHAR2 |
    master_table    IN   DBMS_UTILITY.DBLINK_ARRAY);
```

---

---

**注意：** このプロシージャはオーバーロードされています。`master_list` パラメータと `master_table` パラメータは、同時には指定できません。

---

---

## パラメータ

**表 8-188 REMOVE\_MASTER\_DATABASES プロシージャのパラメータ**

パラメータ	説明
<code>gname</code>	レプリケート環境に対応付けられたオブジェクト・グループの名前です。1つのマスター・データベースが複数のレプリケート環境で使用されている場合、このパラメータによって混乱が避けられます。
<code>master_list</code>	レプリケート環境から削除する完全修飾されたマスター・データベース名の、カンマ区切りのリストです。リスト内の名前の間にはスペースを挿入しません。
<code>master_table</code>	リストのかわりに、 <code>DBMS_UTILITY.DBLINK_ARRAY</code> タイプの PL/SQL 表でデータベース名を指定できます。

## 例外

**表 8-189 REMOVE\_MASTER\_DATABASES プロシージャの例外**

例外	説明
<code>nonmasterdef</code>	起動サイトはマスター定義サイトではありません。
<code>nonmaster</code>	指定されたデータベースのうち少なくとも1つのデータベースが、マスター・サイトではありません。
<code>reconfigerror</code>	指定されたデータベースの1つが、マスター定義サイトです。
<code>commfailure</code>	残りのマスター・サイトのうち、少なくとも1つのマスター・サイトにアクセスできません。

## REPCAT\_IMPORT\_CHECK プロシージャ

このプロシージャを実行すると、レプリケート・オブジェクト、または Oracle レプリケーションで使用されたオブジェクトのエクスポートまたはインポート後に、レプリケート・マスター・グループのオブジェクトが適切なオブジェクト識別子およびステータス値を持つようになります。

### 構文

```
DBMS_REPCAT.REPCAT_IMPORT_CHECK (  
    gname      IN   VARCHAR2,  
    master     IN   BOOLEAN,  
    gowner     IN   VARCHAR2  := 'PUBLIC');
```

### パラメータ

表 8-190 REPCAT\_IMPORT\_CHECK プロシージャのパラメータ

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。両方のパラメータを省略した場合、カレント・サイトで、すべてのレプリケート・マスター・グループがチェックされます。
master	マスター・サイトをチェックする場合は TRUE に、スナップショット・サイトをチェックする場合は FALSE に設定します。
gowner	マスター・グループの所有者です。

### 例外

表 8-191 REPCAT\_IMPORT\_CHECK プロシージャの例外

例外	説明
nonmaster	MASTER が TRUE で、データベースがオブジェクト・グループのマスター・サイトではないか、またはデータベースが予期したデータベースではありません。
nonsnapshot	MASTER が FALSE で、データベースがオブジェクト・グループのスナップショット・サイトではありません。
missingobject	オブジェクト・グループ内の有効なレプリケート・オブジェクトが存在しません。
missingrepgroup	指定されたレプリケート・オブジェクト・グループは存在しません。
missingschema	指定されたレプリケート・オブジェクト・スキーマは存在しません。



## RESUME\_MASTER\_ACTIVITY プロシージャ

このプロシージャは、レプリケート環境を停止した後で、通常のレプリケーション・アクティビティを再開します。

### 構文

```
DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
    gname          IN  VARCHAR2,
    override       IN  BOOLEAN := FALSE);
```

### パラメータ

**表 8-192 RESUME\_MASTER\_ACTIVITY プロシージャのパラメータ**

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。
override	<p>値が TRUE の場合は、保留になっている RepCat 管理要求が無視され、各マスターで通常のレプリケーション・アクティビティがリストアされます。これは緊急の場合のみ使用します。</p> <p>値が FALSE の場合は、各マスターで通常のレプリケーション・アクティビティがリストアされます。ただし、そのマスターで gname に対する RepCat 管理要求が保留にされていない場合に限られます。</p>

### 例外

**表 8-193 RESUME\_MASTER\_ACTIVITY プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	レプリケート・マスター・グループが休止中でないか、または停止していません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
notallgenerated	レプリケーション・アクティビティを再開する前にレプリケーション・サポートが生成されました。

## SEND\_OLD\_VALUES プロシージャ

更新および削除のために、レプリケート表の非キー列の古い列値を送ることができます。デフォルトでは、すべての列の古い値が送られます。マスター定義サイトで DBMS\_REPCAT.SEND\_OLD\_VALUES を呼び出すことにより、すべてのマスター・サイトおよびスナップショット・サイトで、この動作を変更できます。

構文

```
DBMS_REPCAT.SEND_OLD_VALUES (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    { column_list  IN   VARCHAR2,
    | column_table IN   DBMS_REPCAT.VARCHAR2s, }
    operation      IN   VARCHAR2 := 'UPDATE',
    send           IN   BOOLEAN := TRUE );
```

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと column\_table パラメータは、同時には指定できません。

パラメータ

表 8-194 SEND\_OLD\_VALUES プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	レプリケート表の名前です。
column_list	表の列の、カンマ区切りのリストです。各エントリの間にはスペースを挿入しません。
column_table	リストのかわりに、DBMS_REPCAT.VARCHAR2S タイプの PL/SQL 表を使用して列名を指定できます。最初の列名は 1 の位置、2 番目の列名は 2 の位置と、順に配置します。
operation	受け付けられる値は、UPDATE、DELETE または更新と削除を意味するアスタリスク・ワイルドカード '*' です。
send	値が TRUE の場合は、指定された列の古い値が送られます。値が FALSE の場合は、指定された列の古い値が送られません。指定されていない列や操作には影響がありません。  表の min_communication が TRUE になると、マスター定義サイトで、指定された変更が有効になります。min_communication TRUE で、マスター・サイトやスナップショット・サイトで次にレプリケーション・サポートが生成されると、変更が有効になります。

**注意：** operation パラメータを使用すると、行が削除されたとき、または非キー列が更新されたときに、非キー列の古い値を送信するかどうかを決定できます。古い値を渡さない場合、古い値のかわりに NULL が渡されます。その場合、古い値は更新または削除を行った時点でのターゲット・サイドの列のカレント値と見なされます。

Oracle のデフォルトの動作を変更する前に、削減されたデータ伝播の詳細を、『Oracle8i レプリケーション・ガイド』で参照してください。

例外

表 8-195 SEND\_OLD\_VALUES プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しません。
missingcolumn	少なくとも 1 つの列が、表に存在しません。
notquiesced	レプリケート・マスター・グループは停止していません。
typefailure	無効な操作が指定されました。

SET\_COLUMNS プロシージャ

このプロシージャを使用すると、主キーのかわりに代替列または列グループを使用して、行レベル・レプリケーションを使用する際に比較する表中の列を決定できます。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 詳細は、『Oracle8i レプリケーション・ガイド』を参照してください。

構文

```
DBMS_REPCAT.SET_COLUMNS (
  sname          IN   VARCHAR2,
  oname          IN   VARCHAR2,
  { column_list  IN   VARCHAR2
  | column_table IN   DBMS_UTILITY.NAME_ARRAY } );
```

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと column\_table パラメータは、同時には指定できません。

パラメータ

表 8-196 SET\_COLUMNS プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	表の名前です。
column_list	主キーとして使用する表内の列の、カンマ区切りのリストです。各エントリの間にはスペースを挿入しません。
column_table	リストのかわりに、DBMS_UTILITY.NAME_ARRAY の PL/SQL 表を使用して列名を指定できます。最初の列名は 1 の位置、2 番目の列名は 2 の位置と、順に配置します。

例外

表 8-197 SET\_COLUMNS プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しません。
missingcolumn	少なくとも 1 つの列が、表に存在しません。
notquiesced	レプリケート・マスター・グループが休止中でないか、または停止していません。

SUSPEND\_MASTER\_ACTIVITY プロシージャ

このプロシージャは、マスター・グループのレプリケーション・アクティビティを中断します。このプロシージャは、マスター定義サイトからコールする必要があります。

構文

```
DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
  gname    IN    VARCHAR2);
```

## パラメータ

**表 8-198 SUSPEND\_MASTER\_ACTIVITY プロシージャのパラメータ**

パラメータ	説明
gname	アクティビティを中断するマスター・グループの名前です。

## 例外

**表 8-199 SUSPEND\_MASTER\_ACTIVITY プロシージャの例外**

例外	説明
nomasterdef	起動サイトはマスター定義サイトではありません。
notnormal	レプリケート・マスター・グループで通常の操作が行われていません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

## SWITCH\_SNAPSHOT\_MASTER プロシージャ

このプロシージャは、スナップショット・レプリケート・マスター・グループのマスター・データベースを、他のマスター・サイトに変更します。このプロシージャは、影響を受けるスナップショットの完全なリフレッシュを行い、必要に応じてトリガーおよびその対応付けられたパッケージを再生成します。このプロシージャでは、マスターの変更前に、キューが古いマスター・サイトにプッシュされることはありません。

## 構文

```
DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER (
    gname          IN   VARCHAR2,
    master         IN   VARCHAR2,
    gowner         IN   VARCHAR2 := 'PUBLIC');
```

## パラメータ

**表 8-200 SWITCH\_SNAPSHOT\_MASTER プロシージャのパラメータ (1 / 2 ページ)**

パラメータ	説明
gname	マスター・サイトを変更するスナップショット・オブジェクト・グループの名前です。
master	スナップショット・サイトに使用する新しいマスター・データベースの、完全修飾されたデータベース名です。

表 8-200 SWITCH\_SNAPSHOT\_MASTER プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
gowner	スナップショット・グループの所有者です。

例外

表 8-201 SWITCH\_SNAPSHOT\_MASTER プロシージャの例外

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
nonmaster	指定されたデータベースはマスター・サイトではありません。
commfailure	指定されたデータベースにはアクセスできません。
missingrepgroup	スナップショット・グループは存在しません。
qrytoolong	スナップショット定義問合せが 32KB を超えています。
alreadymastered	ローカル・サイトに、同じグループ名で、古いマスター・サイトをマスターに持つ別のスナップショット・グループがあります。

UNREGISTER\_SNAPSHOT\_REPGROUP プロシージャ

このプロシージャは、DBA\_REGISTERED\_SNAPSHOT\_GROUPS からスナップショット・グループを削除することによって、各マスター・サイトでのスナップショットの管理を容易にします。

構文

```
DBMS_REPCAT.UNREGISTER_SNAPSHOT_REPGROUP (  
  gname      IN   VARCHAR2,  
  snapsite   IN   VARCHAR2  
  gowner     IN   VARCHAR2  := 'PUBLIC');
```

パラメータ

表 8-202 UNREGISTER\_SNAPSHOT\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	登録を解除するスナップショット・オブジェクト・グループの名前です。
snapsite	スナップショット・サイトのグローバル名です。
gowner	スナップショット・グループの所有者です。

## VALIDATE ファンクション

このファンクションは、マルチマスター・レプリケーション環境のキー条件が正確であるかどうかを確認します。

### 構文

```
DBMS_REPCAT.VALIDATE (
    gname                IN  VARCHAR2,
    check_genflags        IN  BOOLEAN := FALSE,
    check_valid_objs      IN  BOOLEAN := FALSE,
    check_links_sched     IN  BOOLEAN := FALSE,
    check_links           IN  BOOLEAN := FALSE,
    error_table           OUT DBMS_REPCAT.VALIDATE_ERR_TABLE)
RETURN BINARY_INTEGER;
```

```
DBMS_REPCAT.VALIDATE (
    gname                IN  VARCHAR2,
    check_genflags        IN  BOOLEAN := FALSE,
    check_valid_objs      IN  BOOLEAN := FALSE,
    check_links_sched     IN  BOOLEAN := FALSE,
    check_links           IN  BOOLEAN := FALSE,
    error_msg_table       OUT DBMS_UTILITY.UNCL_ARRAY,
    error_num_table       OUT DBMS_UTILITY.NUMBER_ARRAY )
RETURN BINARY_INTEGER;
```

---

**注意：** このファンクションはオーバーロードされています。VALIDATE の戻り値は、検出されたエラーの数です。このファンクションの OUT パラメータでは、検出されたエラーが戻されます。前述の構文に示される最初のインタフェース・ファンクションで、error\_table は、レコードの配列で構成されます。それぞれのレコードには、VARCHAR2 と NUMBER があります。文字列フィールドにはエラー・メッセージが、数値フィールドには Oracle のエラー番号が入ります。

前述の構文に示される 2 番目のインタフェース・ファンクションも同様ですが、OUT 配列が 2 つあります。VARCHAR2 配列にはエラー・メッセージが、NUMBER 配列にはエラー番号が入ります。

---

パラメータ

表 8-203 VALIDATE ファンクションのパラメータ

パラメータ	説明
gname	確認するマスター・グループの名前です。
check_genflags	グループ内のすべてのオブジェクトが生成されたかどうかを確認されます。この処理は、マスター定義サイトでのみ行われます。
check_valid_objs	グループ内のオブジェクトの基になるオブジェクトが有効であるかどうかを確認されます。この処理は、マスター定義サイトでのみ行われます。マスター定義サイトでの処理が他のすべてのサイトに波及し、基になるオブジェクトが有効であるかどうかチェックされます。オブジェクトの有効性は、接続ユーザーのスキーマ内で確認されます。
check_links_sched	リンクがスケジュールされているかどうかを確認されます。この処理は各マスター・サイトで実行されます。
check_links	レプリケーションが正しく行われるように、接続ユーザー (repadmin) およびプロパゲータに正しいリンクが設定されているかどうかを確認されます。リンクがデータベースに存在し、アクセス可能であるかどうかチェックされます。この処理は各マスター・サイトで実行されます。
error_table	検出されたエラーの総数およびメッセージが戻されます。
error_msg_table	検出されたすべてのエラーのメッセージが戻されます。
error_num_table	検出されたエラーの総数が戻されます。

例外

表 8-204 VALIDATE ファンクションの例外 (1 / 2 ページ)

例外	説明
missingdblink	データベース・リンクがレプリケーション・プロパゲータのスキーマに存在しないか、またはスケジュールされていません。データベース・リンクが確実にデータベースに存在し、アクセス可能で、その実行がスケジュールされるようにしてください。
dblinkmismatch	ローカル・ノードのデータベース・リンク名が、リンクがアクセスするデータベースのグローバル名と一致しません。GLOBAL_NAMES 初期化パラメータが TRUE に設定され、リンク名がグローバル名と一致するようにしてください。



表 8-204 VALIDATE ファンクションの例外 (2 / 2 ページ)

例外	説明
dblinkuidmismatch	ローカル・ノードのレプリケーション管理ユーザーのユーザー名と、データベース・リンクに対応するノードのユーザー名が異なります。アドバンスド・レプリケーションでは、これらのユーザーの名前が同じことを前提としています。ローカル・ノードのレプリケーション管理ユーザーのユーザー ID と、データベース・リンクに対応するノードのユーザー ID を同じにしてください。
objectnotgenerated	オブジェクトが他のマスター・サイトで生成されていないか、生成がまだ終了していません。マスター定義サイトのオブジェクトに対する GENERATE_REPLICATION_SUPPORT および DO_DEFERRED_REPCAT_ADMIN のコールで、確実にオブジェクトが生成されるようにしてください。
opnotsupported	オブジェクト・グループが Oracle8 より前のノードでレプリケートされる場合は、操作がサポートされません。レプリケート・マスター・グループのすべてのノードが Oracle8 を実行するようにしてください。

## 使用にあたっての注意

VALIDATE の戻り値は、検出されたエラーの数です。このファンクションの OUT パラメータでは、検出されたエラーが戻されます。最初のインタフェース・ファンクション ERROR\_TABLE は、レコードの配列で構成されます。それぞれのレコードには、VARCHAR2 と NUMBER があります。文字列フィールドにはエラー・メッセージが、数値フィールドには Oracle のエラー番号が入ります。

2 番目のインタフェースは最初のインタフェースに似ていますが、OUT 配列が 2 つあります。VARCHAR2 配列にはエラー・メッセージが、NUMBER 配列にはエラー番号が入ります。

## WAIT\_MASTER\_LOG プロシージャ

このプロシージャは、マスター・サイトに非同期で伝播された変更が適用されたかどうかを判別します。

## 構文

```
DBMS_REPCAT.WAIT_MASTER_LOG (
    gname          IN    VARCHAR2,
    record_count    IN    NATURAL,
    timeout         IN    NATURAL,
    true_count      OUT   NATURAL);
```

パラメータ

表 8-205 WAIT\_MASTER\_LOG プロシージャのパラメータ

パラメータ	説明
gname	レプリケート・マスター・グループの名前です。
record_count	完了していないアクティビティの数がこのしきい値と同じか、このしきい値より少なくなると、プロシージャが終了します。
timeout	プロシージャが終了するまで待つ最大秒数です。
true_count (out parameter)	完了していないアクティビティの数が戻されます。

例外

表 8-206 WAIT\_MASTER\_LOG プロシージャの例外

例外	説明
nonmaster	起動サイトはマスター・サイトではありません。

# DBMS\_REPCAT\_ADMIN パッケージ

## サブプログラムの概要

表 8-207 DBMS\_REPCAT\_ADMIN パッケージのサブプログラム

サブプログラム	説明
<a href="#">GRANT_ADMIN_ANY_SCHEMA</a> プロシージャ (8-139 ページ)	レプリケーション管理者に、カレント・サイトでのレプリケート・マスター・グループの管理に必要な権限を与えます。
<a href="#">GRANT_ADMIN_SCHEMA</a> プロシージャ (8-140 ページ)	レプリケーション管理者に、カレント・サイトでのスキーマの管理に必要な権限を与えます。
<a href="#">REGISTER_USER_REPGROUP</a> プロシージャ (8-141 ページ)	マスター・サイトで、リモート・サイトに対するプロキシ・スナップショット管理者または受信者権限を割り当てます。
<a href="#">REVOKE_ADMIN_ANY_SCHEMA</a> プロシージャ (8-142 ページ)	GRANT_ADMIN_ANY_SCHEMA プロシージャでレプリケーション管理者に与えられた権限やロールを取り消します。
<a href="#">REVOKE_ADMIN_SCHEMA</a> プロシージャ (8-143 ページ)	GRANT_ADMIN_SCHEMA プロシージャでレプリケーション管理者に与えられた権限やロールを取り消します。
<a href="#">UNREGISTER_USER_REPGROUP</a> プロシージャ (8-144 ページ)	REGISTER_USER_REPGROUP プロシージャでプロキシ・スナップショット管理者または受信者に与えられた権限やロールを取り消します。

## GRANT\_ADMIN\_ANY\_SCHEMA プロシージャ

このプロシージャは、レプリケーション管理者に、カレント・サイトでのレプリケート・マスター・グループの管理に必要な権限を与えます。

### 構文

```
DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (  
    username IN VARCHAR2);
```

パラメータ

表 8-208 GRANT\_ADMIN\_ANY\_SCHEMA プロシージャのパラメータ

パラメータ	説明
username	カレント・サイトでのレプリケート・マスター・グループの管理に必要な権限とロールを与える、レプリケーション管理者の名前です。

例外

表 8-209 GRANT\_ADMIN\_ANY\_REPGROUP プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。

GRANT\_ADMIN\_SCHEMA プロシージャ

このプロシージャは、レプリケーション管理者に、カレント・サイトでのスキーマの管理に必要な権限を与えます。オブジェクト・グループが複数のスキーマにまたがらない場合は、このプロシージャを使用すると便利です。

構文

```
DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA (  
    username IN VARCHAR2);
```

パラメータ

表 8-210 GRANT\_ADMIN\_REPSchema プロシージャのパラメータ

パラメータ	説明
username	レプリケーション管理者の名前です。このユーザーには、カレント・サイトのレプリケート・マスター・グループ内での、同じ名前のスキーマの管理に必要な権限とロールが与えられます。

例外

表 8-211 GRANT\_ADMIN\_REPSchema プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。

# REGISTER\_USER\_REPGROUP プロシージャ

このプロシージャは、マスター・サイトで、リモート・サイトに対するプロキシ・スナップショット管理者または受信者権限を割り当てます。このプロシージャは、プロキシ・スナップショット管理者または受信者に、必要な権限のみを与えます。GRANT ADMIN\_SCHEMA プロシージャや GRANT ADMIN\_ANY\_SCHEMA プロシージャで与えられる強力な権限は、このプロシージャでは与えられません。

**関連項目：** トラステッド・セキュリティ・モデルとアントラステッド・セキュリティ・モデルの詳細は、[付録 A「セキュリティ・オプション」](#)を参照してください。

## 構文

```
DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (  
    username           IN    VARCHAR2,  
    privilege_type     IN    VARCHAR2,  
    {list_of_gnames    IN    VARCHAR2 |  
    table_of_gnames    IN    dbms_utility.name_array});
```

**注意：** このプロシージャはオーバーロードされています。list\_of\_gnames パラメータと table\_of\_gnames パラメータは、同時には指定できません。

## パラメータ

表 8-212 REGISTER\_USER\_REPGROUP プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
username	プロキシ・スナップショット管理者または受信者の権限を与えるユーザーの名前です。
privilege_type	割り当てる権限のタイプを指定します。privilege_type の定義には、次の値を使用します。  受信者の権限を与える場合は、RECEIVER を使用します。  プロキシ・スナップショット管理者の権限を与える場合は、PROXY_SNAPADMIN を使用します。

表 8-212 REGISTER\_USER\_REPGROUP プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
list_of_gnames	ユーザーに与えた受信者権限の適用対象にするオブジェクト・グループの、カンマ区切りのリストです。各エントリの間にはスペースを挿入しません。list_of_gnames を NULL に設定すると、このプロシージャのコール時にはまだ確認されていないオブジェクト・グループを含め、すべてのオブジェクト・グループが受信者権限の適用対象になります。list_of_gnames を NULL に設定するには、名前表記法を使用する必要があります。リスト内に無効なオブジェクト・グループがあると、リスト全体の登録が失敗します。
table_of_gnames	ユーザーに与えた受信者権限の適用対象にするオブジェクト・グループの、PL/SQL 表です。PL/SQL 表は、DBMS_UTILITY.NAME_ARRAY タイプである必要があります。この表は 1 ベースです ( 1 の位置から始まり、1 ずつ増加します)。ユーザーをすべてのオブジェクト・グループに登録するには、NULL を使用します。表内に無効なオブジェクト・グループがあると、表全体の登録が失敗します。

例外

表 8-213 REGISTER\_USER\_REPGROUP プロシージャの例外

例外	説明
nonmaster	指定されたオブジェクト・グループが存在しないか、または起動データベースがマスターではありません。
ORA-01917	指定されたユーザーは存在しません。
typefailure	指定された権限のタイプは正しくありません。

REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャ

このプロシージャは、GRANT\_ADMIN\_ANY\_SCHEMA でレプリケーション管理者に与えられた権限やロールを取り消します。

**注意：** GRANT\_ADMIN\_ANY\_SCHEMA 以外で与えられた同一の権限やロールも取り消されます。

構文

```
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_ANY_SCHEMA (
    username IN VARCHAR2);
```

# パラメータ

表 8-214 REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャのパラメータ

パラメータ	説明
username	権限を取り消すレプリケーション管理者の名前です。

# 例外

表 8-215 REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。

# REVOKE\_ADMIN\_SCHEMA プロシージャ

このプロシージャは、GRANT\_ADMIN\_SCHEMA でレプリケーション管理者に与えられた権限やロールを取り消します。

**注意：** GRANT\_ADMIN\_SCHEMA 以外で与えられた同一の権限やロールも取り消されます。

# 構文

```
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_SCHEMA (
    username IN VARCHAR2);
```

# パラメータ

表 8-216 REVOKE\_ADMIN\_SCHEMA プロシージャのパラメータ

パラメータ	説明
username	権限を取り消すレプリケーション管理者の名前です。

# 例外

表 8-217 REVOKE\_ADMIN\_SCHEMA プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。

## UNREGISTER\_USER\_REPGROUP プロシージャ

このプロシージャは、REGISTER\_USER\_REPGROUP プロシージャでプロキシ・スナップショット管理者または受信者に与えられた権限やロールを取り消します。

### 構文

```
DBMS_REPCAT_ADMIN.UNREGISTER_USER_REPGROUP (  
    username          IN   VARCHAR2,  
    privilege_type     IN   VARCHAR2,  
    {list_of_gnames   IN   VARCHAR2 |  
    table_of_gnames   IN   dbms_utility.name_array) };
```

**注意：** このプロシージャはオーバーロードされています。list\_of\_gnames パラメータと table\_of\_gnames パラメータは、同時には指定できません。

### パラメータ

表 8-218 UNREGISTER\_USER\_REPGROUP プロシージャのパラメータ

パラメータ	説明
username	登録を取り消すユーザーの名前です。
privilege_type	取り消す権限のタイプを指定します。privilege_type の定義には、次の値を使用します。  受信者の権限を取り消す場合は、RECEIVER を使用します。  プロキシ・スナップショット管理者の権限を取り消す場合は、PROXY_SNAPADMIN を使用します。
list_of_gnames	ユーザーに与えた受信者権限の適用を取り消すオブジェクト・グループの、カンマ区切りのリストです。各エントリの間にはスペースを挿入しません。list_of_gnames を NULL に設定すると、すべてのオブジェクト・グループに対するユーザーの登録が取り消されます。 list_of_gnames を NULL に設定するには、名前表記法を使用する必要があります。リスト内に無効なオブジェクト・グループがあると、リスト全体の登録の取消しが失敗します。
table_of_gnames	ユーザーに与えた受信者権限の適用を取り消すオブジェクト・グループの、PL/SQL 表です。PL/SQL 表は、DBMS_UTILITY.NAME_ARRAY タイプである必要があります。この表は 1 ベースです（1 の位置から始まり、1 ずつ増加します）。すべてのオブジェクト・グループに対するユーザーの登録を取り消すには、NULL を使用します。表内に無効なオブジェクト・グループがあると、表全体の登録の取消しが失敗します。



例外

表 8-219 UNREGISTER\_USER\_REPGROUP プロシージャの例外

例外	説明
nonmaster	指定されたオブジェクト・グループが存在しないか、または起動データベースがマスターではありません。
ORA-01917	指定されたユーザーは存在しません。
typefailure	指定された権限のタイプは正しくありません。

# DBMS\_REPCAT\_INSTANTIATE パッケージ

## サブプログラムの概要

表 8-220 DBMS\_REPCAT\_INSTANTIATE パッケージのサブプログラム

サブプログラム	説明
<a href="#">DROP_SITE_INSTANTIATION プロシージャ</a> (8-146 ページ)	ターゲット・サイトを DBA_REPCAT_TEMPLATE_SITES ビューから削除するパブリック・プロシージャです。
<a href="#">INSTANTIATE_OFFLINE ファンクション</a> (8-147 ページ)	オフラインでリモート・スナップショット・サイトにスナップショット環境を構築する際に使用するスクリプトを、マスター・サイトに生成するパブリック・ファンクションです。
<a href="#">INSTANTIATE_OFFLINE_REPAPI ファンクション</a> (8-149 ページ)	オフラインで RepAPI リモート・スナップショット・サイトにスナップショット環境を構築する際に使用するバイナリ・ファイルを、マスター・サイトに生成するパブリック・ファンクションです。
<a href="#">INSTANTIATE_ONLINE ファンクション</a> (8-151 ページ)	オンラインでリモート・スナップショット・サイトにスナップショット環境を構築する際に使用するスクリプトを、マスター・サイトに生成するパブリック・ファンクションです。

DBMS\_REPCAT\_INSTANTIATE パッケージの詳細は、次のファイルを参照してください。

ORACLE\_HOME/rdbms/admin/dbmsrint.sql

## DROP\_SITE\_INSTANTIATION プロシージャ

このプロシージャは、ターゲット・サイトでテンプレートのインスタンスエーションを削除します。このプロシージャは、マスター・サイトで関連するすべてのメタデータを削除します。また、指定されたサイトでのスナップショットのリフレッシュを使用禁止にします。このプロシージャは、最初にテンプレートをインスタンスエートしたユーザーとして実行する必要があります。テンプレートをインスタンスエートしたユーザーを確認するには、[ALL\\_REPCAT\\_TEMPLATE\\_SITES](#) ビューで問合せを行います。

### 構文

```
DBMS_REPCAT_INSTANTIATE.DROP_SITE_INSTANTIATION(  
    refresh_template_name  IN   VARCHAR2,  
    site_name              IN   VARCHAR2);
```

表 8-221 DROP\_SITE\_INSTANTIATION プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象の配置テンプレートの名前です。
site_name	指定されたテンプレートのインスタンスエーションを削除する Oracle サーバー・サイトです。SITE_NAME を指定する場合は、REPAPI_SITE_ID を指定しないでください。

INSTANTIATE\_OFFLINE ファンクション

このファンクションは、オフラインでリモート・スナップショット・サイトにスナップショット環境を構築するために使用するファイルを、マスター・サイトに生成します。生成されたファイルは、オフライン・インスタンスエーション・ファイルで、マスター・サイトに長時間接続できないリモート・スナップショット・サイトで使用します。

リモート・スナップショット・サイトがラップトップの場合、このプロシージャを使用すると便利です。Replication Manager のパッケージ・ツールを使用して、生成されたファイルやデータを 1 つのファイルにパッケージ化すると、FTP サイトに転送したり、CD-ROM やフロッピー・ディスクなどにロードできます。

**関連項目：** 詳細は、『Oracle8i レプリケーション・ガイド』および Replication Manager のオンライン・ヘルプを参照してください。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、Replication Manager などの Oracle Tools で使用します。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから適切な情報を取り出す際に使用します。

このパブリック・ファンクションを実行するユーザーは、指定されたサイトのインスタンスエートされたテンプレートの "登録" ユーザーになります。

**注意：** このファンクションは、配置テンプレートのオフライン・インスタンスエーションに使用します。

このファンクションを、DBMS\_OFFLINE\_OG パッケージのプロシージャ（マスター表のオフライン・インスタンスエーションに使用します）や、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンスエーションに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

構文

```
DBMS_REPCAT_INSTANTIATE.INSTANTIATE_OFFLINE(  
    refresh_template_name  IN  VARCHAR2,
```

```
site_name          IN   VARCHAR2,
runtime_parm_id    IN   NUMBER      := -1e-130,
next_date          IN   DATE        := SYSDATE,
interval           IN   VARCHAR2    := 'SYSDATE + 1',
use_default_gowner IN   BOOLEAN     := TRUE)
return NUMBER;
```

表 8-222 INstantiateOffline ファンクションのパラメータ

パラメータ	説明
refresh_template_name	インスタンスiertされる配置テンプレートの名前です。
site_name	配置テンプレートをインスタンスiertしているリモート・サイトの名前です。
runtime_parm_id	INSERT_RUNTIME_PARMS プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します。ID は、GET_RUNTIME_PARM_ID ファンクションで取得済です。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値です。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔です。
use_default_gowner	値が TRUE の場合、作成されたスナップショット・オブジェクト・グループはデフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたスナップショット・オブジェクト・グループはインスタンスiションを実行するユーザーが所有します。

表 8-223 INstantiateOffline ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
dupl_template_site	配置テンプレートは、このスナップショット・サイトでインスタンスiert済です。配置テンプレートは、各スナップショット・サイトにつき 1 回のみインスタンスiertできます。
not_authorized	このユーザーは、配置テンプレートをインスタンスiertする許可を受けていません。

## 戻り値

**表 8-224 INSTANTIATE\_OFFLINE ファンクションの戻り値**

戻り値	説明
<システム生成番号>	USER_REPCAT_TEMP_OUTPUT ビューから選択して、生成されたインスタンス・スクリプトを取得するときの、output_id に対するシステム生成番号を示します。

## INSTANTIATE\_OFFLINE\_REPAPI ファンクション

このファンクションは、オフラインでリモート RepAPI スナップショット・サイトにスナップショット環境を構築するために使用するファイルを、マスター・サイトに生成します。このオフライン・インスタンス・ファイルは、マスター・サイトに長時間接続できないリモート RepAPI サイトで使用します。

リモート・スナップショット・サイトが Oracle8i Lite (RepAPI を含みます) を実行するラップトップの場合、このファンクションを使用すると便利です。生成されたファイルは、FTP サイトに転送したり、CD-ROM やフロッピー・ディスクなどにロードできます。

このファンクションで生成されたファイルは、マスター・サイトの、パラメータ OFFLINE\_DIRPATH によって指定されたディレクトリに格納されます。このファイルの名前は、USER\_NAME、REFRESH\_TEMPLATE\_NAME および SITE\_ID に基づいて指定され、拡張子 .oli が付きます。たとえば、サイト 1234 の、MYTEMPLATE テンプレートのユーザー SCOTT 用のオフラインのインスタンス・ファイルの名前は次のようになります。

```
scott_mytemplate_1234.oli.
```

これは接続ユーザー用のオフライン・インスタンス・ファイルを生成するためのパブリック・ファンクションです。

---

**注意：** このファンクションは、配置テンプレートのオフライン・インスタンス・ファイルに使用します。

このファンクションを、DBMS\_OFFLINE\_OG パッケージのプロシージャ（マスター表のオフライン・インスタンス・ファイルに使用）や、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンス・ファイルに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

---

構文

```
DBMS_REPCAT_INSTANTIATE.INSTANTIATE_OFFLINE_REPAPI (
  refresh_template_name  IN   VARCHAR2,
  site_id                IN   VARCHAR2  := NULL,
  master                 IN   VARCHAR2  := NULL,
  url                    IN   VARCHAR2  := NULL,
  ssl                    IN   NUMBER    := 0,
  trace_vector            IN   NUMBER    := DBMS_REPCAT_RGT.NO_TRACE_DUMP,
  resultset_threshold    IN   NUMBER    := DBMS_REPCAT_INSTANTIATE.
                                     RESULTSET_THRESHOLD,
  lob_threshold           IN   NUMBER    := DBMS_REPCAT_INSTANTIATE.
                                     LOB_THRESHOLD);
```

表 8-225 INstantiate\_Offline\_RepAPI ファンクションのパラメータ (1 / 2 ページ)

パラメータ	説明
refresh_template_name	インスタンスiertされる配置テンプレートの名前です。
site_id	<p>このオフライン・インスタンスiエーションに割り当てられた一時名です。この一時名は、オフライン・インスタンスiエーション・ファイルの名前の一部になります。デフォルトの NULL 値が使用される場合は、この一時名に対する数値が生成されます。オフライン・インスタンスiエーション・ファイルの名前が目的のスナップショット・サイトを示すようにする場合は、名前を指定すると便利です。</p> <p>site_idは一意なものにする必要があります。すなわち、複数のオフライン・インスタンスiエーション・ファイルの site_idを同じにはできません。</p> <p>通常、この一時名は、配置プレートのインスタンスiエート時に RepAPI クライアントによって上書きされます。このパラメータは、デフォルトの NULL 値に設定することをお薦めします。</p>
master	RepAPI クライアントがサーバー用に使用するオプションの別名です。このパラメータを指定した場合は、RepAPI クライアントでサーバーを使用するときは必ずこの別名を使用する必要があります。
url	データベースへのアクセスのために公開された、マスター・サイトの URL です。このパラメータを指定した場合は、RepAPI クライアントでサーバーを使用するときは必ずこの URL を使用する必要があります。
ssl	値が 1 の場合、スナップショットでマスター・サイトとの通信にセキュア・ソケット・レイヤー (SSL) が使用されます。値が 0 の場合、SSL は使用されません。
trace_vector	デバッグのトレース・レベルです。

表 8-225 INSTANTIATE\_OFFLINE\_REPAPI ファンクションのパラメータ (2 / 2 ページ)

パラメータ	説明
resultset_threshold	スナップショットのリフレッシュ中に送信される、非 LOB 行データの最大サイズです。
lob_threshold	スナップショットのリフレッシュ中に送信される、LOB 行データの最大サイズです。

表 8-226 INSTANTIATE\_OFFLINE\_REPAPI ファンクションの例外

例外	説明
miss_refresh_template	テンプレートが存在しません。
miss_user	ユーザー名がデータベース内に存在しません。
miss_template_site	ユーザーおよびサイト用に、テンプレートがインスタンスエートされていません。

## 戻り値

表 8-227 INSTANTIATE\_OFFLINE\_REPAPI ファンクションの戻り値

戻り値	説明
0	エラーが発生しました。
1	エラーは発生しませんでした。

## INSTANTIATE\_ONLINE ファンクション

このファンクションは、オンラインでリモート・スナップショット・サイトにスナップショット環境を構築する際に使用するスクリプトを、マスター・サイトに生成します。生成されたスクリプトは、マスター・サイトに長時間接続できるリモート・スナップショット・サイトで使用します。新しいスナップショットに設定されるデータの量によっては、リモート・スナップショット・サイトでのインスタンスエーションに時間がかかります。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、Replication Manager などの Oracle Tools で使用します。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから適切な情報を取り出す際に使用します。

このパブリック・ファンクションを実行するユーザーは、指定されたサイトのインスタンスエートされたテンプレートの登録ユーザーになります。

構文

```
DBMS_REPCAT_INSTANTIATE.INSTANTIATE_ONLINE(  
  refresh_template_name  IN   VARCHAR2,  
  site_name              IN   VARCHAR2,  
  runtime_parm_id        IN   NUMBER      := -1e-130,  
  next_date              IN   DATE        := SYSDATE,  
  interval               IN   VARCHAR2    := 'SYSDATE + 1',  
  use_default_gowner     IN   BOOLEAN     := TRUE)  
  return NUMBER;
```

表 8-228 INSTANTIATE\_ONLINE ファンクションのパラメータ

パラメータ	説明
refresh_template_name	インスタンスエートされる配置テンプレートの名前です。
site_name	配置テンプレートをインスタンスエートしているリモート・サイトの名前です。
runtime_parm_id	INSERT_RUNTIME_PARMS プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します (ID は、GET_RUNTIME_PARM_ID ファンクションを使用して取得済です)。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値を指定します。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔を指定します。
use_default_gowner	値が TRUE の場合、作成されたスナップショット・オブジェクト・グループはデフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたスナップショット・オブジェクト・グループはインスタンスエーションを実行するユーザーが所有します。

表 8-229 INSTANTIATE\_ONLINE ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
dupl_template_site	配置テンプレートは、このスナップショット・サイトでインスタンスエート済です。配置テンプレートは、各スナップショット・サイトにつき 1 回のみインスタンスエートできます。
not_authorized	このユーザーは、配置テンプレートをインスタンスエートする許可を受けていません。



## 戻り値

表 8-230 INSTANTIATE\_ONLINE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT ビューから選択して、生成されたインスタンスエーション・スクリプトを取得するときの、output_id に対するシステム生成番号を示します。

# DBMS\_REPCAT\_RGT パッケージ

## サブプログラムの概要

表 8-231 DBMS\_REPCAT\_RGT パッケージのサブプログラム (1 / 3 ページ)

サブプログラム	説明
<a href="#">ALTER_REFRESH_TEMPLATE プロシージャ</a> (8-156 ページ)	データベース管理者が既存の配置テンプレートを変更します。
<a href="#">ALTER_TEMPLATE_OBJECT プロシージャ</a> (8-158 ページ)	指定された配置テンプレートに追加されたオブジェクトを変更します。
<a href="#">ALTER_TEMPLATE_PARM プロシージャ</a> (8-160 ページ)	データベース管理者が指定された配置テンプレートのパラメータを変更します。
<a href="#">ALTER_USER_AUTHORIZATION プロシージャ</a> (8-162 ページ)	DBA_REPCAT_USER_AUTHORIZATIONS ビューの内容を変更します。
<a href="#">ALTER_USER_PARM_VALUE プロシージャ</a> (8-163 ページ)	特定のユーザーに対して定義されている既存のパラメータ値を変更します。
<a href="#">COMPARE_TEMPLATES ファンクション</a> (8-165 ページ)	データベース管理者が2つの配置テンプレートの内容を比較します。
<a href="#">COPY_TEMPLATE ファンクション</a> (8-166 ページ)	データベース管理者が配置テンプレートをコピーします。
<a href="#">CREATE_OBJECT_FROM_EXISTING ファンクション</a> (8-168 ページ)	既存のデータベース・オブジェクトからテンプレート・オブジェクト定義を作成し、ターゲット配置テンプレートに追加します。
<a href="#">CREATE_REFRESH_TEMPLATE ファンクション</a> (8-169 ページ)	データベース管理者がテンプレート名、プライベートまたはパブリックのステータスおよびターゲット・リフレッシュ・グループの定義を行うことができる、配置テンプレートを作成します。
<a href="#">CREATE_TEMPLATE_OBJECT ファンクション</a> (8-171 ページ)	オブジェクト定義をターゲットの配置テンプレート・コンテナに追加します。
<a href="#">CREATE_TEMPLATE_PARM ファンクション</a> (8-173 ページ)	指定された配置テンプレートのパラメータを作成して、リモート・スナップショット・サイトでカスタム・データ・セットを作成します。
<a href="#">CREATE_USER_AUTHORIZATION ファンクション</a> (8-176 ページ)	特定のユーザーに対して、プライベート配置テンプレートをインスタンス化する許可を与えます。
<a href="#">CREATE_USER_PARM_VALUE ファンクション</a> (8-177 ページ)	特定のユーザーに対して、配置テンプレートのパラメータ値を事前定義します。

表 8-231 DBMS\_REPCAT\_RGT パッケージのサブプログラム (2 / 3 ページ)

サブプログラム	説明
<a href="#">DELETE_RUNTIME_PARS プロシージャ</a> (8-179 ページ)	INSERT_RUNTIME_PARS プロシージャを使用して定義したランタイム・パラメータ値を削除します。
<a href="#">DROP_ALL_OBJECTS プロシージャ</a> (8-179 ページ)	データベース管理者が、すべてのオブジェクトまたは特定のオブジェクト・タイプを、配置テンプレートから削除します。
<a href="#">DROP_ALL_TEMPLATE_PARS プロシージャ</a> (8-180 ページ)	データベース管理者が指定された配置テンプレートのテンプレート・パラメータを削除します。
<a href="#">DROP_ALL_TEMPLATE_SITES プロシージャ</a> (8-181 ページ)	DBA_REPCAT_TEMPLATE_SITES ビューからすべてのエントリを削除します。
<a href="#">DROP_ALL_TEMPLATES プロシージャ</a> (8-182 ページ)	プロシージャがコールされたサイトで、すべての配置テンプレートを削除します。
<a href="#">DROP_ALL_USER_AUTHORIZATIONS プロシージャ</a> (8-182 ページ)	データベース管理者が、指定された配置テンプレートに対して与えられた、すべてのユーザー認証を削除します。
<a href="#">DROP_ALL_USER_PARM_VALUES プロシージャ</a> (8-183 ページ)	指定された配置テンプレートのユーザー・パラメータ値を削除します。
<a href="#">DROP_REFRESH_TEMPLATE プロシージャ</a> (8-184 ページ)	配置テンプレートを削除します。
<a href="#">DROP_SITE_INSTANTIATION プロシージャ</a> (8-185 ページ)	DBA_REPCAT_TEMPLATE_SITES ビューからターゲット・サイトを削除します。
<a href="#">DROP_TEMPLATE_OBJECT プロシージャ</a> (8-186 ページ)	指定された配置テンプレートからテンプレート・オブジェクトを削除します。
<a href="#">DROP_TEMPLATE_PARM プロシージャ</a> (8-187 ページ)	DBA_REPCAT_TEMPLATE_PARS ビューから既存のテンプレート・パラメータを削除します。
<a href="#">DROP_USER_AUTHORIZATION プロシージャ</a> (8-188 ページ)	DBA_REPCAT_USER_AUTHORIZATIONS ビューからユーザー認証エントリを削除します。
<a href="#">DROP_USER_PARM_VALUE プロシージャ</a> (8-189 ページ)	特定の配置テンプレートに対して事前定義されたユーザー・パラメータ値を削除します。
<a href="#">GET_RUNTIME_PARM_ID ファンクション</a> (8-190 ページ)	ランタイム・パラメータ値を定義する際に使用する ID を取得します。
<a href="#">INSERT_RUNTIME_PARS プロシージャ</a> (8-190 ページ)	テンプレートのインスタンス化の前に、ランタイム・パラメータ値を定義します。
<a href="#">INSTANTIATE_OFFLINE ファンクション</a> (8-192 ページ)	オフラインでリモート・スナップショット・サイトにスナップショット環境を構築する際に使用するスクリプトを、マスター・サイトに生成します。

表 8-231 DBMS\_REPCAT\_RGT パッケージのサブプログラム (3 / 3 ページ)

サブプログラム	説明
<a href="#">INstantiateOfflineRepAPI ファンクション</a> (8-194 ページ)	オフラインで RepAPI リモート・スナップショット・サイトにスナップショット環境を構築する際に使用するバイナリ・ファイルを、マスター・サイトに生成します。
<a href="#">InstantiateOnlineファンク ション</a> (8-197 ページ)	オンラインでリモート・スナップショット・サイトにスナップショット環境を構築する際に使用するスクリプトを、マスター・サイトに生成します。
<a href="#">LockTemplateExclusiveプロ シージャ</a> (8-199 ページ)	配置テンプレートが更新または変更されている間、ユーザーによるテンプレートの読取りやインスタンス化を行わないようにします。
<a href="#">LockTemplateSharedプロ シージャ</a> (8-199 ページ)	指定された配置テンプレートを読取り専用にします。

## ALTER\_REFRESH\_TEMPLATE プロシージャ

このプロシージャを実行すると、データベース管理者が既存の配置テンプレートを変更できます。変更作業には、新しい配置テンプレート名、新しいリフレッシュ・グループまたは新しい所有者の定義、およびパブリックまたはプライベートへのステータス変更などがあります。

### 構文

```
DBMS_REPCAT_RGT.ALTER_REFRESH_TEMPLATE (  
    refresh_template_name      IN    VARCHAR2,  
    new_owner                  IN    VARCHAR2 := '-',  
    new_refresh_group_name     IN    VARCHAR2 := '-',  
    new_refresh_template_name  IN    VARCHAR2 := '-',  
    new_template_comment      IN    VARCHAR2 := '-',  
    new_public_template        IN    VARCHAR2 := '-',  
    new_last_modified          IN    DATE := to_date('1', 'J'),  
    new_modified_by            IN    NUMBER := -1e-130);
```

## パラメータ

**表 8-232 ALTER\_REFRESH\_TEMPLATE プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	変更される配置テンプレートの名前です。
new_owner	配置テンプレートの新しい所有者の名前です。カレント所有者を維持する場合は、値を指定しないでください。
new_refresh_group_name	必要な場合は、このパラメータを使用して、テンプレート・オブジェクトを追加する新しいリフレッシュ・グループの名前を指定します。カレント・リフレッシュ・グループを維持する場合は、値を指定しないでください。
new_refresh_template_name	新しい配置テンプレート名を指定するには、このパラメータを使用します。カレント配置テンプレート名を維持する場合は、値を指定しないでください。
new_template_comment	新しい配置テンプレート・コメントです。カレント・テンプレート・コメントを維持する場合は、値を指定しないでください。
new_public_template	配置テンプレートがパブリックであるかプライベートであるかを指定します。受け付けられる値は、'Y' (パブリック) および 'N' (プライベート) です。カレント値を維持する場合は、値を指定しないでください。
new_last_modified	この配置テンプレートが最後に変更された日付です。値を指定しない場合は、カレント日付が自動的に使用されます。
new_modified_by	この配置テンプレートを最後に変更したユーザーの名前です。値を指定しない場合は、カレント・ユーザー名が自動的に使用されます。

## 例外

**表 8-233 ALTER\_REFRESH\_TEMPLATE プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
bad_public_template	public_template パラメータが正しく指定されませんでした。public_template パラメータは、パブリック・テンプレートの場合には 'Y'、プライベート・テンプレートの場合には 'N' に指定します。
dupl_refresh_template	指定された名前のテンプレートは、すでに存在しています。DBA_REPCAT_REFRESH_TEMPLATES ビューを参照してください。

## ALTER\_TEMPLATE\_OBJECT プロシージャ

このプロシージャは、指定された配置テンプレートに追加されたオブジェクトを変更します。一般的な変更処理は、オブジェクト DDL の変更および他の配置テンプレートへのオブジェクトの再割当てです。

配置テンプレートに加えられた変更は、配置テンプレートをインスタンスエートする新しいサイトのみに反映されます。テンプレートをすでにインスタンスエートしたリモート・サイトで変更を適用するには、配置テンプレートを再びインスタンスエートする必要があります。

### 構文

```
DBMS_REPCAT_RGT.ALTER_TEMPLATE_OBJECT (
  refresh_template_name      IN   VARCHAR2,
  object_name                IN   VARCHAR2,
  object_type                IN   VARCHAR2,
  new_refresh_template_name  IN   VARCHAR2 := '-',
  new_object_name            IN   VARCHAR2 := '-',
  new_object_type            IN   VARCHAR2 := '-',
  new_ddl_text               IN   CLOB    := '-',
  new_master_rollback_seg    IN   VARCHAR2 := '-',
  new_flavor_id              IN   NUMBER := -1e-130);
```

### パラメータ

表 8-234 ALTER\_TEMPLATE\_OBJECT プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
refresh_template_name	変更するオブジェクトが含まれる配置テンプレートの名前です。
object_name	変更するテンプレート・オブジェクトの名前です。
object_type	変更するオブジェクトのタイプです。
new_refresh_template_name	このオブジェクトの再割当てを行う新しい配置テンプレートの名前です。カレント配置テンプレートへのオブジェクトの割当てを維持する場合は、値を指定しないでください。
new_object_name	テンプレート・オブジェクトの新しい名前です。カレント・オブジェクト名を維持する場合は、値を指定しないでください。

表 8-234 ALTER\_TEMPLATE\_OBJECT プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
new_object_type	このパラメータを指定すると、オブジェクト・タイプが変更されます。次のオブジェクト・タイプを指定できます。 <div> <div>SNAPSHOT</div> <div>PROCEDURE</div> </div> <div> <div>INDEX</div> <div>FUNCTION</div> </div> <div> <div>TABLE</div> <div>PACKAGE</div> </div> <div> <div>VIEW</div> <div>PACKAGE BODY</div> </div> <div> <div>SYNONYM</div> <div>TRIGGER</div> </div> <div> <div>SEQUENCE</div> <div>DATABASE LINK</div> </div>
new_ddl_text	指定されたオブジェクトの新しいオブジェクト DDL です。カレント・オブジェクト DDL を維持する場合は、新しい DDL テキストを指定しないでください。
new_master_rollback_seg	指定されたオブジェクトの新しいマスター・ロールバック・セグメントです。カレント・ロールバック・セグメントを維持する場合は、値を指定しないでください。
new_flavor_id	このパラメータは内部で使います。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

## 例外

表 8-235 ALTER\_TEMPLATE\_OBJECT プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_flavor_id	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。
bad_object_type	オブジェクト・タイプが正しく指定されませんでした。有効なオブジェクト・タイプは、 <a href="#">表 8-234</a> を参照してください。
miss_template_object	指定されたテンプレート・オブジェクト名が無効か、または存在しません。
dupl_template_object	new_refresh_template_name パラメータで指定された新しいテンプレート名は、すでに存在しています。

## 使用にあたっての注意

ALTER\_TEMPLATE\_OBJECT プロシージャでは CLOB が利用されています。このため、ALTER\_TEMPLATE\_OBJECT プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、ALTER\_TEMPLATE\_OBJECT プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'CREATE SNAPSHOT snap_sales AS SELECT *
        FROM sales WHERE salesperson = :salesid and region_id = :region';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_OBJECT(
        refresh_template_name => 'rgt_personnel',
        object_name => 'SNAP_SALES',
        object_type => 'SNAPSHOT',
        new_ddl_text => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

## ALTER\_TEMPLATE\_PARM プロシージャ

このプロシージャを実行すると、データベース管理者が、指定された配置テンプレートのパラメータを変更できます。変更作業には、パラメータの改名や、デフォルト値およびプロンプト文字列の再定義などがあります。

## 構文

```
DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM (
    refresh_template_name      IN   VARCHAR2,
    parameter_name             IN   VARCHAR2,
    new_refresh_template_name  IN   VARCHAR2 := '-',
    new_parameter_name         IN   VARCHAR2 := '-',
    new_default_parm_value     IN   CLOB := NULL,
    new_prompt_string          IN   VARCHAR2 := '-',
    new_user_override          IN   VARCHAR2 := '-');
```



## パラメータ

**表 8-236 ALTER\_TEMPLATE\_PARM プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	変更するパラメータを持つ配置テンプレートの名前です。
parameter_name	変更するパラメータの名前です。
new_refresh_template_name	指定されたパラメータが新しく割り当てられる配置テンプレートの名前です。あるテンプレートから他のテンプレートにパラメータを移動するときに便利です。カレント・テンプレートへの割当てを維持する場合は、値を指定しないでください。
new_parameter_name	テンプレート・パラメータの新しい名前です。カレント・パラメータ名を維持する場合は、値を指定しないでください。
new_default_parm_value	指定されたパラメータの新しいデフォルト値です。カレント・デフォルト値を維持する場合は、値を指定しないでください。
new_prompt_string	指定されたパラメータの新しいプロンプト・テキストです。カレント・プロンプト文字列を維持する場合は、値を指定しないでください。
new_user_override	インスタンスエーションの際に確認メッセージが表示された場合、ユーザーがデフォルト値を上書きできるかどうかを決定します。このパラメータにユーザー・パラメータ値が定義されていないと、確認メッセージが表示されます。ユーザーがデフォルト値を上書きする場合は 'Y' に、上書きできないようにする場合は 'N' に設定します。

## 例外

**表 8-237 ALTER\_TEMPLATE\_PARM プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_template_parm	指定されたテンプレート・パラメータが無効か、または存在しません。
dupl_template_parm	new_refresh_template_name と new_parameter_name の組合せは、すでに存在しています。

## 使用にあたっての注意

ALTER\_TEMPLATE\_PARM プロシージャでは CLOB が利用されています。このため、ALTER\_TEMPLATE\_PARM プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、ALTER\_TEMPLATE\_PARM プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        new_default_parm_value => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

ALTER\_USER\_AUTHORIZATION プロシージャ

このプロシージャは、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューの内容を変更します。具体的には、ユーザーまたは配置テンプレートの認可の割当てを変更できます。このプロシージャは、たとえばある従業員の職場が変更になり、他の配置テンプレートのスナップショット環境を必要とする場合などに役立ちます。データベース管理者が単にこの従業員に新しい配置テンプレートを割り当てることによって、このユーザーにターゲット・テンプレートをインスタンスエートする許可が与えられます。

構文

```
DBMS_REPCAT_RGT.ALTER_USER_AUTHORIZATION (
    user_name                IN   VARCHAR2,
    refresh_template_name    IN   VARCHAR2,
    new_user_name            IN   VARCHAR2 := '-',
    new_refresh_template_name IN   VARCHAR2 := '-');
```

パラメータ

表 8-238 ALTER\_USER\_AUTHORIZATION プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
user_name	認可の変更が行われるユーザーの名前です。
refresh_template_name	指定されたユーザーに現在割り当てられている配置テンプレートの名前です。
new_user_name	このテンプレートを認可する新しいユーザーを定義するには、このパラメータを使用します。カレント・ユーザーを維持する場合は、値を指定しないでください。

表 8-238 ALTER\_USER\_AUTHORIZATION プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
new_refresh_template_name	指定されたユーザー（既存のユーザーまたは新しいユーザー）が、インスタンスエートする許可を持つ配置テンプレートです。カレント配置テンプレートを維持する場合は、値を指定しないでください。

## 例外

表 8-239 ALTER\_USER\_AUTHORIZATION プロシージャの例外

例外	説明
miss_user_authorization	指定された user_name 値と refresh_template_name 値の組合せは、DBA_REPCAT_USER_AUTHORIZATIONS ビューに存在しません。
miss_user	new_user_name パラメータまたは user_name パラメータに指定されたユーザー名が無効か、または存在しません。
miss_refresh_template	new_refresh_template パラメータに指定された配置テンプレートが無効か、または存在しません。
dupl_user_authorization	指定されたユーザー名および配置テンプレート用の行がすでに存在しています。DBA_REPCAT_USER_AUTHORIZATIONS ビューを確認してください。

## ALTER\_USER\_PARM\_VALUE プロシージャ

このプロシージャは、指定されたユーザーに定義されている既存のパラメータ値を変更します。スナップショット環境で割当て表が使用されている場合に、このプロシージャを使用すると便利です。ユーザー・パラメータ値を変更すると、リモート・スナップショット・サイトのデータ・セットを簡単かつ安全に変更できます。

**関連項目：** 割当て表の使用方法的の詳細は、『Oracle8i レプリケーション・ガイド』の「配置テンプレートの設計」を参照してください。

## 構文

```
DBMS_REPCAT_RGT.ALTER_USER_PARM_VALUE(
    refresh_template_name    IN    VARCHAR2,
    parameter_name           IN    VARCHAR2,
    user_name                 IN    VARCHAR2,
    new_refresh_template_name IN    VARCHAR2 := '-',
    new_parameter_name       IN    VARCHAR2 := '-',
    new_user_name            IN    VARCHAR2 := '-',
    new_parm_value           IN    CLOB := NULL);
```

パラメータ

表 8-240 ALTER\_USER\_PARM\_VALUE プロシージャのパラメータ

パラメータ	説明
refresh_template_name	変更するユーザー・パラメータ値を持つ配置テンプレートの名前です。
parameter_name	変更するパラメータの名前です。
user_name	パラメータ値が変更されるユーザーの名前です。
new_refresh_template_name	指定されたユーザー・パラメータ値が新しく割り当てられる配置テンプレートの名前です。他のテンプレートに対してユーザーを認可するときに便利です。カレント・テンプレートへの割当てを維持する場合は、値を指定しないでください。
new_parameter_name	新しいテンプレート・パラメータの名前です。既存のパラメータに定義されたユーザー値を維持する場合は、値を指定しないでください。
new_user_name	このパラメータ値を割り当てる新しいユーザー名です。カレント・ユーザーへ割り当てられたパラメータ値を維持する場合は、値を指定しないでください。
new_parm_value	指定されたユーザー・パラメータの新しいパラメータ値です。カレント・パラメータ値を維持する場合は、値を指定しないでください。

例外

表 8-241 ALTER\_USER\_PARM\_VALUE プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_template_parm	指定されたテンプレート・パラメータが無効か、または存在しません。
miss_user	user_name パラメータまたは new_user_name パラメータに指定されたユーザー名が無効か、または存在しません。
miss_user_parm_values	指定されたユーザー・パラメータ値は存在しません。
dupl_user_parm_values	指定された新しいユーザー・パラメータはすでに存在しています。

使用にあたっての注意

ALTER\_USER\_PARM\_VALUE プロシージャでは CLOB が利用されています。このため、ALTER\_USER\_PARM\_VALUE プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、ALTER\_USER\_PARM\_VALUE プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```

DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_USER_PARM_VALUE(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        user_name => 'BOB',
        new_parm_value => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/

```

## COMPARE\_TEMPLATES ファンクション

このファンクションを実行すると、データベース管理者が2つの配置テンプレートの内容を比較できます。2つの配置テンプレートの内容の相違点は、USER\_REPCAT\_TEMP\_OUTPUT 表に格納されます。

COMPARE\_TEMPLATES ファンクションでは、USER\_REPCAT\_TEMP\_OUTPUT 表の問合せの際に WHERE 句で指定する数値が戻されます。たとえば、COMPARE\_TEMPLATES プロシージャで 10 が戻された場合は、次の SELECT 文を実行して、指定された2つのテンプレートのすべての相違点を確認します。2つのテンプレートの内容が同一の場合は、SELECT 文で行が戻されません。

```

SELECT text FROM user_repcat_temp_output
WHERE output_id = 10 ORDER BY LINE;

```

接続が解除されるか、または ROLLBACK が実行されると、USER\_REPCAT\_TEMP\_OUTPUT の内容が失われます。

## 構文

```

DBMS_REPCAT_RGT.COMPARE_TEMPLATES (
    source_template_name  IN  VARCHAR2,
    compare_template_name IN  VARCHAR2)
return NUMBER;

```

パラメータ

表 8-242 COMPARE\_TEMPLATES ファンクションのパラメータ

パラメータ	説明
source_template_name	比較する最初の配置テンプレートの名前です。
compare_template_name	比較する 2 番目の配置テンプレートの名前です。

例外

表 8-243 COMPARE\_TEMPLATES ファンクションの例外

例外	説明
miss_refresh_template	比較する配置テンプレートの名前が無効か、または存在しません。

戻り値

表 8-244 COMPARE\_TEMPLATES ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT ビューから選択して、比較するテンプレート間の違いを表示するときに、output_id 値に戻される番号を示します。

COPY\_TEMPLATE ファンクション

このファンクションを実行すると、データベース管理者が配置テンプレートをコピーできます。COPY\_TEMPLATE は、新しい配置テンプレートで既存の配置テンプレートのオブジェクトが多数使用されるときに便利です。このファンクションは、配置テンプレート、テンプレート・オブジェクト、テンプレート・パラメータおよびユーザー・パラメータ値をコピーします。データベース管理者は、このファンクションを使用して、テンプレートのユーザー認証をコピーできます。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

**注意：** DBA\_REPCAT\_TEMPLATE\_SITES ビュー内の値はコピーされません。

このファンクションを使用すると、データベース管理者が配置テンプレートを他のマスター・サイトにコピーできます。この機能は、配置テンプレートの配布や、複数のサイトでのネットワーク負荷の分担に役立ちます。

## 構文

```
DBMS_REPCAT_RGT.COPY_TEMPLATE (
    old_refresh_template_name    IN    VARCHAR2,
    new_refresh_template_name    IN    VARCHAR2,
    copy_user_authorizations     IN    VARCHAR2,
    dblink                       IN    VARCHAR2 := NULL)
return NUMBER;
```

## パラメータ

**表 8-245 COPY\_TEMPLATE ファンクションのパラメータ**

パラメータ	説明
old_refresh_template_name	コピーされる配置テンプレートの名前です。
new_refresh_template_name	新しい配置テンプレートの名前です。
copy_user_authorizations	元のテンプレートに対する認可を、新しい配置テンプレートにコピーするかどうかを指定します。このパラメータに受け付けられる値は、'Y'、'N' および NULL です。  <b>注意:</b> すべてのユーザーが、ターゲット・データベースに存在する必要があります。
dblink	配置テンプレートのコピー元を定義します。配置テンプレートを他のマスター・サイトに配布するときに便利です。コピー元を指定しない場合、配置テンプレートはローカル・マスター・サイトからコピーされます。

## 例外

**表 8-246 COPY\_TEMPLATE ファンクションの例外**

例外	説明
miss_refresh_template	コピーする配置テンプレートの名前が無効か、または存在しません。
dupl_refresh_template	指定された新しいリフレッシュ・テンプレートの名前は、すでに存在しています。
bad_copy_auth	copy_user_authorization パラメータに指定された値は無効です。有効な値は、'Y'、'N' および NULL です。

戻り値

表 8-247 COPY\_TEMPLATES ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用されるシステム生成番号です。

CREATE\_OBJECT\_FROM\_EXISTING ファンクション

このファンクションは、既存のデータベース・オブジェクトからテンプレート・オブジェクト定義を作成して、ターゲット配置テンプレートに追加します。ターゲット配置テンプレートがリモート・スナップショット・サイトでインスタンス化されるときに、元のデータベース・オブジェクトを作成した DDL が実行されます。既存のトリガーやプロシージャをテンプレートに追加する場合、このファンクションを使用すると便利です。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使

構文

```
DEMS_REPCAT_RGT.CREATE_OBJECT_FROM_EXISTING(  
    refresh_template_name  IN   VARCHAR2,  
    object_name             IN   VARCHAR2,  
    sname                   IN   VARCHAR2,  
    oname                   IN   VARCHAR2,  
    otype                   IN   VARCHAR2)  
return NUMBER;
```

パラメータ

表 8-248 CREATE\_OBJECT\_FROM\_EXISTING ファンクションのパラメータ (1 / 2 ページ)

パラメータ	説明
refresh_template_name	このオブジェクトが追加される配置テンプレートの名前です。
object_name	オプションで、配置テンプレートに追加する既存のオブジェクトの、新しい名前を指定します (既存のオブジェクトに新しい名前を定義できます)。
sname	作成するテンプレート・オブジェクトの基になるオブジェクトを含むスキーマです。
oname	作成するテンプレート・オブジェクトの基になるオブジェクトの名前です。



表 8-248 CREATE\_OBJECT\_FROM\_EXISTING ファンクションのパラメータ (2 / 2 ページ)

パラメータ	説明
otype	テンプレートに追加するデータベース・オブジェクトのタイプ (PROCEDURE、TRIGGER など) です。オブジェクト・タイプは次の数値識別子で指定します。DATABASE LINK または SNAPSHOT は、このファンクションでは使用できません。
	SEQUENCE                      PROCEDURE
	INDEX                            FUNCTION
	TABLE                           PACKAGE
	VIEW                            PACKAGE BODY
	SYNONYM                        TRIGGER

## 例外

表 8-249 CREATE\_OBJECT\_FROM\_EXISTING ファンクションの例外

例外	説明
miss_refresh_template	指定されたリフレッシュ・テンプレート名が無効か、または存在しません。既存の配置テンプレートを確認するには、DBA_REPCAT_REFRESH_TEMPLATE ビューの問合せを行います。
bad_object_type	オブジェクト・タイプが正しく指定されませんでした。
dupl_template_object	指定された配置テンプレートには、同名で同タイプのオブジェクトがすでに追加されています。
objectmissing	指定されたオブジェクトは存在しません。

## 戻り値

表 8-250 CREATE\_OBJECT\_FROM\_EXISTING ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用するシステム生成番号です。

## CREATE\_REFRESH\_TEMPLATE ファンクション

このファンクションは、テンプレート名、プライベートまたはパブリックのステータスおよびターゲット・リフレッシュ・グループの定義を行う、配置テンプレートを作成します。テンプレート・オブジェクト、ユーザー認証またはテンプレート・パラメータを作成するたびに、このファンクションで作成される配置テンプレートを参照します。このファンクション

は、DBA\_REPCAT\_REFRESH\_TEMPLATES ビューに行を追加します。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

構文

```
DBMS_REPCAT_RGT.CREATE_REFRESH_TEMPLATE (  
  owner                IN   VARCHAR2,  
  refresh_group_name   IN   VARCHAR2,  
  refresh_template_name IN   VARCHAR2,  
  template_comment     IN   VARCHAR2 := NULL,  
  public_template      IN   VARCHAR2 := NULL,  
  last_modified        IN   DATE := SYSDATE,  
  modified_by          IN   VARCHAR2 := USER,  
  creation_date        IN   DATE := SYSDATE,  
  created_by           IN   VARCHAR2 := USER)  
return NUMBER;
```

パラメータ

表 8-251 CREATE\_REFRESH\_TEMPLATE ファンクションのパラメータ (1 / 2 ページ)

パラメータ	説明
owner	このパラメータで、配置テンプレートの所有者のユーザー名を指定します。所有者を指定しない場合、テンプレートを作成したユーザーの名前が自動的に使用されます。
refresh_group_name	このテンプレートのインスタンス化時に作成される、リフレッシュ・グループの名前です。このテンプレートで作成されるすべてのオブジェクトは、指定されたリフレッシュ・グループに割り当てられます。
refresh_template_name	作成する配置テンプレートの名前です。この配置テンプレートを使用するすべてのアクティビティで、この名前が参照されます。
template_comment	このパラメータで定義されるユーザー・コメントは、DBA_REPCAT_REFRESH_TEMPLATES ビューにリストされます。
public_template	配置テンプレートがパブリックであるかプライベートであるかを指定します。受け付けられる値は、'Y' (パブリック) および 'N' (プライベート) です。
last_modified	この配置テンプレートが最後に変更された日付です。値を指定しない場合は、カレント日付が自動的に使用されます。
modified_by	この配置テンプレートを最後に変更したユーザーの名前です。値を指定しない場合は、カレント・ユーザー名が自動的に使用されます。
creation_date	この配置テンプレートが作成された日付です。値を指定しない場合は、カレント日付が自動的に使用されます。

表 8-251 CREATE\_REFRESH\_TEMPLATE ファンクションのパラメータ (2 / 2 ページ)

パラメータ	説明
created_by	この配置テンプレートを作成したユーザーの名前です。値を指定しない場合は、カレント・ユーザー名が自動的に使用されます。

## 例外

表 8-252 CREATE\_REFRESH\_TEMPLATE ファンクションの例外

例外	説明
dupl_refresh_template	指定された名前のテンプレートは、すでに存在しています。既存のテンプレートを確認するには、DBA_REPCAT_REFRESH_TEMPLATES ビューを参照してください。
bad_public_template	public_template パラメータが正しく指定されませんでした。 public_template パラメータは、パブリック・テンプレートの場合には 'Y'、プライベート・テンプレートの場合には 'N' に指定します。

## 戻り値

表 8-253 CREATE\_REFRESH\_TEMPLATE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用するシステム生成番号です。

## CREATE\_TEMPLATE\_OBJECT ファンクション

このファンクションは、オブジェクト定義をターゲットの配置テンプレート・コンテナに追加します。ターゲット配置テンプレートがリモート・スナップショット・サイトでインスタンシアートされるときに、指定されたオブジェクト DDL が実行されます。スナップショット以外にも、このファンクションを使用して、表、プロシージャおよび他のオブジェクトをテンプレートに追加できます。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

## 構文

```
DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
    refresh_template_name  IN   VARCHAR2,
    object_name            IN   VARCHAR2,
    object_type            IN   VARCHAR2,
    ddl_text               IN   CLOB,
    master_rollback_seg    IN   VARCHAR2 := NULL,
    flavor_id              IN   NUMBER := -1e-130)
return NUMBER;
```

パラメータ

表 8-254 CREATE\_TEMPLATE\_OBJECT ファンクションのパラメータ

パラメータ	説明
refresh_template_name	このオブジェクトが追加される配置テンプレートの名前です。
object_name	作成するテンプレート・オブジェクトの名前です。
object_type	テンプレートに追加するデータベース・オブジェクトのタイプ (SNAPSHOT、TRIGGER、PROCEDURE など) です。次のオブジェクト・タイプを指定できます。 <div>SNAPSHOTPROCEDURE</div> <div>INDEXFUNCTION</div> <div>TABLEPACKAGE</div> <div>VIEWPACKAGE BODY</div> <div>SYNONYMMATERIALIZED VIEW</div> <div>SEQUENCEDATABASE LINK</div> <div>TRIGGER</div>
ddl_text	テンプレートに追加するオブジェクトを作成する DDL です。DDL は、必ずセミコロンで終了するようにします。テンプレート・オブジェクト用にテンプレート・パラメータを作成する場合は、コロン (:) を使用できます。詳細は、 <a href="#">第 4 章「配置テンプレートの作成」</a> を参照してください。 <div>CREATE SNAPSHOT 文でスナップショットを追加する場合は、スナップショット問合せでマスター表の所有者のスキーマ名を指定してください。</div>
master_rollback_seg	定義されたオブジェクト DDL をリモート・スナップショット・サイトで実行する際に使用する、ロールバック・セグメントの名前を指定します。
flavor_id	このパラメータは内部で使用します。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

例外

表 8-255 CREATE\_TEMPLATE\_OBJECT ファンクションの例外 (1 / 2 ページ)

例外	説明
miss_refresh_template	指定されたリフレッシュ・テンプレート名が無効か、または存在しません。既存の配置テンプレートを確認するには、DBA_REPCAT_REFRESH_TEMPLATE ビューの問合せを行います。

表 8-255 CREATE\_TEMPLATE\_OBJECT ファンクションの例外 (2 / 2 ページ)

例外	説明
bad_object_type	オブジェクト・タイプが正しく指定されませんでした。有効なオブジェクト・タイプは、表 8-254 を参照してください。
dupl_template_object	指定された配置テンプレートには、同名で同タイプのオブジェクトがすでに追加されています。

## 戻り値

表 8-256 CREATE\_TEMPLATE\_OBJECT ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用するシステム生成番号です。

## 使用にあたっての注意

CREATE\_TEMPLATE\_OBJECT ファンクションでは、CLOB が利用されています。このため、CREATE\_TEMPLATE\_OBJECT ファンクションを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、CREATE\_TEMPLATE\_OBJECT ファンクションで DBMS\_LOB パッケージを使用する方法を示します。

```

DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'CREATE SNAPSHOT snap_sales AS SELECT *
        FROM sales WHERE salesperson = :salesid';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT(
        refresh_template_name => 'rgt_personnel',
        object_name => 'snap_sales',
        object_type => 'SNAPSHOT',
        ddl_text => templob,
        master_rollback_seg => 'RBS');
    DBMS_LOB.FREETEMPORARY(templob);
END;
/

```

## CREATE\_TEMPLATE\_PARM ファンクション

このファンクションを実行すると、指定された配置テンプレートのパラメータが作成され、リモート・スナップショット・サイトでカスタム・データ・セットを作成できます。この

ファンクションは、データベース管理者が、テンプレート・オブジェクトを追加する前にテンプレート変数のセットを定義する場合のみに必要です。CREATE\_TEMPLATE\_OBJECT ファンクションを使用してオブジェクトがテンプレートに追加されると、オブジェクト DDL のすべての変数が自動的に DBA\_REPCAT\_TEMPLATE\_PARMS ビューに追加されます。

データベース管理者は、通常、ALTER\_TEMPLATE\_PARM ファンクションを使用して、デフォルトのパラメータ値やプロンプト文字列を修正します。詳細は、8-160 ページの [ALTER\\_TEMPLATE\\_PARM プロシージャ](#) を参照してください。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使

構文

```
DBMS_REPCAT_RGT.CREATE_TEMPLATE_PARM (
    refresh_template_name IN  VARCHAR2,
    parameter_name        IN  VARCHAR2,
    default_parm_value    IN  CLOB  := NULL,
    prompt_string         IN  VARCHAR2 := NULL,
    user_override         IN  VARCHAR2 := NULL)
return NUMBER;
```

パラメータ

表 8-257 CREATE\_TEMPLATE\_PARM ファンクションのパラメータ

パラメータ	説明
refresh_template_name	パラメータを作成する配置テンプレートの名前です。
parameter_name	作成するパラメータの名前です。
default_parm_value	パラメータのデフォルト値は、このパラメータで定義します。ユーザー・パラメータ値やランタイム・パラメータ値がない場合、インスタンス化の際にこのデフォルト値が使用されます。
prompt_string	インスタンス化の際にこのテンプレート・パラメータに表示される、プロンプト・テキストです。
user_override	インスタンス化の際に確認メッセージが表示された場合、ユーザーがデフォルト値を上書きできるかどうかを決定します。このパラメータにユーザー・パラメータ値が定義されていないと、確認メッセージが表示されます。ユーザーがデフォルト値を上書きする場合は 'Y' に、上書きできないようにする場合は 'N' に設定します。

## 例外

**表 8-258 CREATE\_TEMPLATE\_PARM ファンクションの例外**

例外	説明
miss_refresh_template	指定されたりフレッシュ・テンプレート名が無効か、または存在しません。
dupl_template_parm	指定された配置テンプレートには、同名のパラメータがすでに定義されています。

## 戻り値

**表 8-259 CREATE\_TEMPLATE\_PARM ファンクションの戻り値**

戻り値	説明
<システム生成番号>	Oracle 内部で使用するシステム生成番号です。

## 使用にあたっての注意

CREATE\_TEMPLATE\_PARM ファンクションでは CLOB が利用されています。このため、CREATE\_TEMPLATE\_PARM ファンクションを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、CREATE\_TEMPLATE\_PARM ファンクションで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_PARM(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        default_parm_value => templob,
        prompt_string => 'Enter your region ID:',
        user_override => 'Y');
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

## CREATE\_USER\_AUTHORIZATION ファンクション

このファンクションは、特定のユーザーに対して、プライベート配置テンプレートをインスタンスエートする許可を与えます。プライベート配置テンプレートの使用を許可されていないユーザーは、プライベート・テンプレートをインスタンスエートできません。このファンクションは、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューに行を追加します。

ユーザーに許可を与える前に、配置テンプレートをインスタンスエートするマスター・サイトに、そのユーザーが存在するかどうかを確認します。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

### 構文

```
DBMS_REPCAT_RGT.CREATE_USER_AUTHORIZATION (  
    user_name          IN    VARCHAR2,  
    refresh_template_name IN  VARCHAR2)  
return NUMBER;
```

### パラメータ

表 8-260 CREATE\_USER\_AUTHORIZATION ファンクションのパラメータ

パラメータ	説明
user_name	指定されたテンプレートをインスタンスエートする許可を持つユーザーの名前です。複数のユーザーを設定するときは、各ユーザー名をカンマで区切ります ('john, mike, bob' など)。
refresh_template_name	指定されたユーザーにインスタンスエートする許可を与えるテンプレートの名前です。

### 例外

表 8-261 CREATE\_USER\_AUTHORIZATION ファンクションの例外

例外	説明
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_refresh_template	指定されたリフレッシュ・テンプレート名が無効か、または存在しません。
dupl_user_authorization	指定されたユーザーおよび配置テンプレートに対しては、すでに認可が作成されています。テンプレートに対する認可を確認するには、DBA_REPCAT_USER_AUTHORIZATIONS ビューを参照してください。



## 戻り値

**表 8-262 CREATE\_USER\_AUTHORIZATION ファンクションの戻り値**

戻り値	説明
<システム生成番号>	Oracle 内部で使用するシステム生成番号です。

## CREATE\_USER\_PARM\_VALUE ファンクション

このファンクションは、特定のユーザーに対して、配置テンプレートのパラメータ値を事前定義します。たとえば、ユーザー 33456 に対して、領域パラメータを WEST に事前定義する場合、このファンクションを使用します。

このファンクションで指定された値は、テンプレート・パラメータに指定されたデフォルト値に優先します。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使います。

## 構文

```
DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE (
    refresh_template_name    IN    VARCHAR2,
    parameter_name           IN    VARCHAR2,
    user_name                IN    VARCHAR2,
    parm_value               IN    CLOB := NULL)
return NUMBER;
```

## パラメータ

**表 8-263 CREATE\_USER\_PARM\_VALUE ファンクションのパラメータ**

パラメータ	説明
refresh_template_name	ユーザー・パラメータ値を作成するパラメータを含む配置テンプレートの名前を指定します。
parameter_name	ユーザー・パラメータ値を定義するテンプレート・パラメータの名前です。
user_name	ユーザー・パラメータ値を事前定義する対象のユーザーの名前を指定します。
parm_value	指定されたユーザーが開始したインスタンスエーション・プロセスで使用する、事前定義パラメータ値です。

例外

表 8-264 CREATE\_USER\_PARM\_VALUE ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
dupl_user_parm_values	指定されたユーザー、パラメータおよび配置テンプレートのパラメータ値は、すでに定義されています。既存のユーザー・パラメータ値を確認するには、DBA_REPCAT_USER_PARM_VALUES ビューの問合せを行います。
miss_template_parm	指定された配置テンプレート・パラメータ名が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。

戻り値

表 8-265 CREATE\_USER\_PARM\_VALUE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用されるシステム生成番号です。

使用にあたっての注意

CREATE\_USER\_PARM\_VALUE ファンクションでは CLOB が利用されています。このため、このファンクションを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、CREATE\_USER\_PARM\_VALUE ファンクションで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        user_name => 'BOB',
        user_parm_value => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

## DELETE\_RUNTIME\_PARMS プロシージャ

INSERT\_RUNTIME\_PARMS プロシージャを使用して定義したランタイム・パラメータ値を削除するには、配置テンプレートをインスタンスエートする前に、このプロシージャを実行します。

### 構文

```
DBMS_REPCAT_RGT.DELETE_RUNTIME_PARMS(  
    runtime_parm_id    IN    NUMBER,  
    parameter_name     IN    VARCHAR2);
```

### パラメータ

表 8-266 DELETE\_RUNTIME\_PARMS プロシージャのパラメータ

パラメータ	説明
runtime_parm_id	以前にランタイム・パラメータ値を代入した ID を指定します。この値は、GET_RUNTIME_PARM_ID ファンクションで取得済です。
parameter_name	削除対象のパラメータ値の名前を指定します。配置テンプレート・パラメータを確認するには、DBA_REPCAT_TEMPLATE_PARMS の問合せを行います。

### 例外

表 8-267 DELETE\_RUNTIME\_PARMS プロシージャの例外

例外	説明
miss_template_parm	指定された配置テンプレート・パラメータ名が無効か、または存在しません。

## DROP\_ALL\_OBJECTS プロシージャ

このプロシージャを実行すると、データベース管理者が、すべてのオブジェクトまたは特定のオブジェクト・タイプを、配置テンプレートから削除できます。

---

---

**注意：** このプロシージャを実行すると、元に戻せません。実行するときは十分注意してください。

---

---

構文

```
DBMS_REPCAT_RGT.DROP_ALL_OBJECTS (  
    refresh_template_name    IN    VARCHAR2,  
    object_type              IN    VARCHAR2 := NULL);
```

パラメータ

表 8-268 DROP\_ALL\_OBJECTS プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のオブジェクトを持つ配置テンプレートの名前です。
object_type	値を NULL に設定すると、テンプレート内のすべてのオブジェクトが削除されます。オブジェクト・タイプを指定すると、そのタイプのオブジェクトのみが削除されます。次のオブジェクト・タイプを指定できます。  SNAPSHOT                      PROCEDURE  INDEX                         FUNCTION  TABLE                         PACKAGE  VIEW                         PACKAGE BODY  SYNONYM                      MATERIALIZED VIEW  SEQUENCE                     DATABASE LINK  TRIGGER

例外

表 8-269 DROP\_ALL\_OBJECTS プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
bad_object_type	オブジェクト・タイプが正しく指定されませんでした。有効なオブジェクト・タイプは、 <a href="#">表 8-268</a> を参照してください。

DROP\_ALL\_TEMPLATE\_PARMS プロシージャ

このプロシージャを使用すると、指定された配置テンプレートのテンプレート・パラメータを削除できます。このプロシージャを使用して、テンプレート・オブジェクトが参照しないすべてのパラメータを削除するか、パラメータを参照するすべてのオブジェクトを、パラメータそのものと同時にテンプレートから削除できます。

---

---

**注意：** このプロシージャを実行すると、元に戻せません。実行するときは十分注意してください。

---

---

## 構文

```
DBMS_REPCAT_RGT.DROP_ALL_TEMPLATE_PARMS (  
    refresh_template_name    IN    VARCHAR2,  
    drop_objects              IN    VARCHAR2 := N);
```

## パラメータ

**表 8-270 DROP\_ALL\_TEMPLATE\_PARMS プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	削除対象のパラメータおよびオブジェクトがある配置テンプレートの名前です。
drop_objects	値を指定しない場合のデフォルトは 'N' になり、テンプレート・オブジェクトが参照しないすべてのパラメータが削除されます。  'Y' を指定した場合は、テンプレート・パラメータを参照するすべてのオブジェクト、およびテンプレート・パラメータそのものが削除されます。オブジェクトはデータベースからではなく、テンプレートから削除されます。

## 例外

**表 8-271 DROP\_ALL\_TEMPLATE\_PARMS プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。

## DROP\_ALL\_TEMPLATE\_SITES プロシージャ

このプロシージャは、特定の配置テンプレートをインスタンス化したサイトの記録が保存される DBA\_REPCAT\_TEMPLATE\_SITES ビューから、すべてのエントリを削除します。

---

---

**注意：** このプロシージャを実行すると、元に戻せません。実行するときは十分注意してください。また、削除されたテンプレートをインスタンス化している Oracle8i Lite サイトでは、スナップショットをリフレッシュできなくなります。

---

---

構文

```
DBMS_REPCAT_RGT.DROP_ALL_TEMPLATE_SITES (  
    refresh_template_name    IN    VARCHAR2);
```

パラメータ

表 8-272 DROP\_ALL\_TEMPLATE\_SITES プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のサイトを持つ配置テンプレートの名前です。

例外

表 8-273 DROP\_ALL\_TEMPLATE\_SITES プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。

DROP\_ALL\_TEMPLATES プロシージャ

このプロシージャは、プロシージャがコールされたサイトで、すべての配置テンプレートを削除します。

**注意：** このプロシージャを実行すると、元に戻せません。実行するとき  
は十分注意してください。

構文

```
DBMS_REPCAT_RGT.DROP_ALL_TEMPLATES;
```

パラメータ

なし

DROP\_ALL\_USER\_AUTHORIZATIONS プロシージャ

このプロシージャを実行すると、データベース管理者が、指定された配置テンプレートに対して与えられたすべてのユーザー認証を削除できます。このプロシージャを実行すると、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューから行が削除されます。

このプロシージャは、プライベート・テンプレートがパブリック・テンプレートに変換され、ユーザー認証がなくなかった後でインプリメントされることもあります。

## 構文

```
DBMS_REPCAT_RGT.DROP_ALL_USER_AUTHORIZATIONS (
    refresh_template_name IN VARCHAR2);
```

## パラメータ

**表 8-274 DROP\_ALL\_USER\_AUTHORIZATIONS プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	削除対象のユーザー認証を含む配置テンプレートの名前です。

## 例外

**表 8-275 DROP\_ALL\_USER\_AUTHORIZATIONS プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。

## DROP\_ALL\_USER\_PARM\_VALUES プロシージャ

このプロシージャは、指定された配置テンプレートのユーザー・パラメータ値を削除します。このプロシージャを使用すると、削除対象のユーザー・パラメータ値のセットを、データベース管理者が簡単に定義できます。たとえば、次のようにパラメータを定義します。

refresh\_template\_name: 指定された配置テンプレートの、すべてのユーザー・パラメータが削除されます。

refresh\_template\_name,user\_name: 指定された配置テンプレートの、すべての指定されたユーザー・パラメータが削除されます。

refresh\_template\_name,parameter\_name: 指定された配置テンプレート・パラメータの、すべてのユーザー・パラメータ値が削除されます。

refresh\_template\_name,parameter\_name,user\_name: 指定された配置テンプレート・パラメータの、指定されたユーザーの値が削除されます (DROP\_USER\_PARM の処理に相当します)。

## 構文

```
DBMS_REPCAT_RGT.DROP_ALL_USER_PARMS (
    refresh_template_name IN VARCHAR2,
    user_name              IN VARCHAR2,
    parameter_name         IN VARCHAR2);
```

パラメータ

表 8-276 DROP\_ALL\_USER\_PARMS プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のパラメータ値を持つ配置テンプレートの名前です。
user_name	パラメータ値が削除されるユーザーの名前です。
parameter_name	削除対象の値を持つテンプレート・パラメータです。

例外

表 8-277 DROP\_ALL\_USER\_PARMS プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_user_parm_values	配置テンプレート、ユーザーおよびパラメータの組合せは、DBA_REPCAT_USER_PARM_VALUES ビューに存在しません。

DROP\_REFRESH\_TEMPLATE プロシージャ

このプロシージャは、配置テンプレートを削除します。このプロシージャで配置テンプレートを削除すると、すべての関連するテンプレート・パラメータ、ユーザー認証、テンプレート・オブジェクトおよびユーザー・パラメータも削除されます。ただし、テンプレート・サイトは削除されません。

構文

```
DBMS_REPCAT_RGT.DROP_REFRESH_TEMPLATE (  
    refresh_template_name IN VARCHAR2);
```

パラメータ

表 8-278 DROP\_REFRESH\_TEMPLATE プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象の配置テンプレートの名前です。



## 例外

**表 8-279 DROP\_REFRESH\_TEMPLATE プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。配置テンプレートを確認するには、DBA_REPCAT_REFRESH_TEMPLATE ビューの問合せを行います。

## DROP\_SITE\_INSTANTIATION プロシージャ

このプロシージャは、ターゲット・サイト（RepAPI サイトも含まれます）でテンプレートのインスタンスエーションを削除します。このプロシージャは、マスター・サイトで関連するすべてのメタデータを削除します。また、指定されたサイトでのスナップショットのリフレッシュを使用禁止にします。

## 構文

```
DBMS_REPCAT_RGT.DROP_SITE_INSTANTIATION (
    refresh_template_name IN  VARCHAR2,
    user_name              IN  VARCHAR2,
    {site_name             IN  VARCHAR2,
    | repapi_site_id       IN  NUMBER  := -1e-130,}
    process_repapi_site    IN  VARCHAR2 := 'N');
```

**注意：** このプロシージャはオーバーロードされています。site\_name パラメータと repapi\_site\_id パラメータは、同時には指定できません。

**表 8-280 DROP\_SITE\_INSTANTIATION プロシージャのパラメータ** （1 / 2 ページ）

パラメータ	説明
refresh_template_name	削除対象の配置テンプレートの名前です。
user_name	リモート・スナップショット・サイトでテンプレートを最初にインスタンスエートしたユーザーの名前です。テンプレートをインスタンスエートしたユーザーを確認するには、ALL_REPCAT_TEMPLATE_SITES ビューの問合せを行います。詳細は、9-12 ページの「 <a href="#">ALL_REPCAT_TEMPLATE_SITES</a> 」を参照してください。
site_name	指定されたテンプレートのインスタンスエーションを削除する Oracle サーバー・サイトです。SITE_NAME を指定する場合は、REPAPI_SITE_ID を指定しないでください。

表 8-280 DROP\_SITE\_INSTANTIATION プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
repapi_site_id	指定されたテンプレートのインスタンスエーションを削除する RepAPI の位置です。REPAPI_SITE_ID を指定する場合は、SITE_NAME を指定しないでください。
process_repapi_site	'Y' が設定された場合、SITE_NAME は RepAPI の SITE_NAME であるとみなされます。デフォルト値は 'N' です。REPAPI_SITE_ID に NULL 以外が設定された場合、このパラメータは考慮されません。

例外

表 8-281 DROP\_SITE\_INSTANTIATION プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_user	指定されたユーザー名は存在しません。
miss_template_site	ユーザーおよびサイト用に、配置テンプレートがインスタンスエートされていません。

DROP\_TEMPLATE\_OBJECT プロシージャ

このプロシージャは、指定された配置テンプレートからテンプレート・オブジェクトを削除します。たとえば、データベース管理者が古いスナップショットを配置テンプレートから削除するときに、このプロシージャを使用できます。配置テンプレートに加えられた変更は、配置テンプレートをインスタンスエートする新しいサイトに反映されます。テンプレートをすでにインスタンスエートしたりリモート・サイトで変更を適用するには、配置テンプレートを再びインスタンスエートする必要があります。

構文

```
DBMS_REPCAT_RGT.DROP_TEMPLATE_OBJECT (  
  refresh_template_name IN   VARCHAR2,  
  object_name           IN   VARCHAR2,  
  object_type           IN   VARCHAR2);
```

## パラメータ

**表 8-282 DROP\_TEMPLATE\_OBJECT プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	オブジェクトが削除される配置テンプレートの名前です。
object_name	削除対象のテンプレート・オブジェクトの名前です。
object_type	削除対象のオブジェクトのタイプです。次のオブジェクト・タイプを指定できます。  SNAPSHOT                      PROCEDURE  INDEX                          FUNCTION  TABLE                          PACKAGE  VIEW                           PACKAGE BODY  SYNONYM                       MATERIALIZED VIEW  SEQUENCE                       DATABASE LINK  TRIGGER

## 例外

**表 8-283 DROP\_TEMPLATE\_OBJECT プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_template_object	指定されたテンプレート・オブジェクトが無効か、または存在しません。配置テンプレート・オブジェクトを確認するには、DBA_REPCAT_TEMPLATE_OBJECT ビューの問合せを行います。

## DROP\_TEMPLATE\_PARM プロシージャ

このプロシージャは、DBA\_REPCAT\_TEMPLATE\_PARS ビューから既存のテンプレート・パラメータを削除します。テンプレート・オブジェクトを削除して、特定のパラメータが不要になった場合、このプロシージャを使用すると便利です。

## 構文

```
DBMS_REPCAT_RGT.DROP_TEMPLATE_PARM (  
    refresh_template_name IN   VARCHAR2,  
    parameter_name        IN   VARCHAR2);
```

パラメータ

表 8-284 DROP\_TEMPLATE\_PARM プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のパラメータを持つ配置テンプレートの名前です。
parameter_name	削除対象のパラメータの名前です。

例外

表 8-285 DROP\_TEMPLATE\_PARM プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_template_parm	指定されたパラメータ名が無効か、または存在しません。テンプレート・パラメータを確認するには、DBA_REPCAT_TEMPLATE_PARMS ビューの問合せを行います。

DROP\_USER\_AUTHORIZATION プロシージャ

このプロシージャは、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューからユーザー認証エントリを削除します。ユーザーのテンプレート認可を削除するときに、このプロシージャを使用します。認可が削除されたユーザーは、ターゲット配置テンプレートをインスタンスエートできなくなります。

**関連項目：** 詳細は、8-182 ページの [DROP\\_ALL\\_USER\\_AUTHORIZATIONS プロシージャ](#) を参照してください。

構文

```
DBMS_REPCAT_RGT.DROP_USER_AUTHORIZATION (  
    refresh_template_name  IN   VARCHAR2,  
    user_name              IN   VARCHAR2);
```

パラメータ

表 8-286 DROP\_USER\_AUTHORIZATION プロシージャのパラメータ

パラメータ	説明
refresh_template_name	ユーザー認証が削除される配置テンプレートの名前です。
user_name	認可が削除されるユーザーの名前です。

## 例外

**表 8-287 DROP\_USER\_AUTHORIZATION プロシージャの例外**

例外	説明
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_user_ authorization	指定されたユーザーと配置テンプレートの組合せは存在しません。 ユーザーや配置テンプレートに対する認可を確認するには、DBA_ REPCAT_USER_AUTHORIZATIONS ビューの問合せを行います。
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。

## DROP\_USER\_PARM\_VALUE プロシージャ

このプロシージャは、特定の配置テンプレートに対して事前定義されたユーザー・パラメータ値を削除します。このプロシージャは、ユーザーのテンプレート認可が削除された後で実行されることがよくあります。

## 構文

```
DBMS_REPCAT_RGT.DROP_USER_PARM_VALUE (
    refresh_template_name    IN    VARCHAR2,
    parameter_name           IN    VARCHAR2,
    user_name                 IN    VARCHAR2);
```

## パラメータ

**表 8-288 DROP\_USER\_PARM\_VALUE プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	削除対象のパラメータ値を持つ配置テンプレートの名前です。
parameter_name	削除対象の事前定義値を持つパラメータの名前です。
user_name	パラメータ値が削除されるユーザーの名前です。

## 例外

**表 8-289 DROP\_USER\_PARM\_VALUE プロシージャの例外** (1 / 2 ページ)

例外	説明
miss_refresh_ template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。

表 8-289 DROP\_USER\_PARM\_VALUE プロシージャの例外 ( 2 / 2 ページ)

例外	説明
miss_user_parm_values	配置テンプレート、ユーザーおよびパラメータの組合せは、DBA_REPCAT_USER_PARM_VALUES ビューに存在しません。

## GET\_RUNTIME\_PARM\_ID ファンクション

このファンクションは、ランタイム・パラメータ値を定義する際に使用する ID を取得します。すべてのランタイム・パラメータ値はこの ID に割り当てられ、インスタンスエーションの際にも使用されます。

### 構文

```
DBMS_REPCAT_RGT.GET_RUNTIME_PARM_ID
RETURN NUMBER;
```

### パラメータ

なし

### 戻り値

表 8-290 GET\_RUNTIME\_PARM\_ID ファンクションの戻り値

戻り値	対応するデータ型
<システム生成番号>	ランタイム・パラメータ値はシステム生成番号に割り当てられ、インスタンスエーションの際にも使用されます。

## INSERT\_RUNTIME\_PARMS プロシージャ

このプロシージャは、テンプレートのインスタンスエーションの前に、ランタイム・パラメータ値を定義します。ユーザー・パラメータ値が定義されていない場合で、デフォルトのパラメータ値を使用しないとき、このプロシージャでパラメータ値を定義します。

このプロシージャを使用する前に、必ず GET\_RUNTIME\_PARM\_ID ファンクションを実行して、ランタイム・パラメータを挿入する際に使用するパラメータ ID を取得します。この ID は、ランタイム・パラメータ値の定義および配置テンプレートのインスタンスエーションに使用します。

## 構文

```
DBMS_REPCAT_RGT.INSERT_RUNTIME_PARMS (
    runtime_parm_id    IN    NUMBER,
    parameter_name     IN    VARCHAR2,
    parameter_value    IN    CLOB);
```

## パラメータ

**表 8-291 INSERT\_RUNTIME\_PARMS プロシージャのパラメータ**

パラメータ	説明
runtime_parm_id	GET_RUNTIME_PARM_ID ファンクションで取得した ID です。この ID は、配置テンプレートをインスタンスエートする際にも使用します。1 つの配置テンプレートのすべてのパラメータ値には、必ず同じ ID を使用してください。
parameter_name	ランタイム・パラメータ値を定義するテンプレート・パラメータの名前です。テンプレート・パラメータを確認するには、DBA_REPCAT_TEMPLATE_PARMS ビューの問合せを行います。
parameter_value	配置テンプレートのインスタンスエーションの際に使用する、ランタイム・パラメータ値です。

## 例外

**表 8-292 INSERT\_RUNTIME\_PARMS プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_user_parm_values	配置テンプレート、ユーザーおよびパラメータの組合せは、DBA_REPCAT_USER_PARM_VALUES ビューに存在しません。

## 使用にあたっての注意

このプロシージャでは CLOB が利用されています。このため、INSERT\_RUNTIME\_PARMS プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、INSERT\_RUNTIME\_PARMS プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```

DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.INSERT_RUNTIME_PARS(
        runtime_parm_id => 20,
        parameter_name => 'region',
        parameter_value => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/

```

## INstantiate\_Offline ファンクション

このファンクションは、オフラインでリモート・スナップショット・サイトにスナップショット環境を構築するために使用するスクリプトを、マスター・サイトに生成します。生成されたスクリプトは、マスター・サイトに長時間接続できないリモート・スナップショット・サイトで使用します。新しいスナップショットに設定されるデータの量によっては、リモート・スナップショット・サイトでのインスタンスエーションに時間がかかることがあります。このファンクションは、ユーザーのインスタンスエーションごとに個別に実行する必要があります。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、Replication Manager などの Oracle Tools で使用します。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから適切な情報を取り出す際に使用します。

---

**注意：** このファンクションは、配置テンプレートのオフライン・インスタンスエーションに使用します。また、このファンクションは、他のユーザーのためにインスタンスエーションを行うレプリケーション管理者が使用します。独自のインスタンスエーションを行うユーザーは、パブリックの INstantiate\_Offline ファンクションを使用します。詳細は、8-147 ページの [INstantiate\\_Offline ファンクション](#) を参照してください。

このファンクションを、DBMS\_OFFLINE\_OG パッケージのプロシージャ（マスター表のオフライン・インスタンスエーションに使用）や、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンスエーションに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

---



## 構文

```
DBMS_REPCAT_RGT.INSTANTIATE_OFFLINE (
    refresh_template_name    IN    VARCHAR2,
    site_name                IN    VARCHAR2,
    user_name                IN    VARCHAR2    := NULL,
    runtime_parm_id          IN    NUMBER      := -1e-130,
    next_date                IN    DATE        := SYSDATE,
    interval                 IN    VARCHAR2    := 'SYSDATE + 1',
    use_default_gowner       IN    BOOLEAN    := TRUE)
return NUMBER;
```

## パラメータ

**表 8-293 INSTANTIATE\_OFFLINE ファンクションのパラメータ**

パラメータ	説明
refresh_template_name	インスタンスiertされる配置テンプレートの名前です。
site_name	配置テンプレートをインスタンスiertしている、リモート・サイトの名前です。
user_name	配置テンプレートのインスタンスiert中の、許可を受けたユーザーの名前です。
runtime_parm_id	INSERT_RUNTIME_PARMs プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します (ID は、GET_RUNTIME_PARM_ID ファンクションで取得済です)。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値を指定します。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔を指定します。
use_default_gowner	値が TRUE の場合、作成されたスナップショット・オブジェクト・グループはデフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたスナップショット・オブジェクト・グループはインスタンスierションを実行するユーザーが所有します。

例外

表 8-294 INstantiate\_Offline ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_user	許可を受けたユーザーの名前が無効か、または存在しません。指定されたユーザーが DBA_REPCAT_USER AUTHORIZATIONS ビューにリストされているかどうかを確認してください。このビューにリストされていないユーザーは、ターゲット配置テンプレートをインスタンス化する許可を受けていません。

戻り値

表 8-295 INstantiate\_Offline ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT ビューから選択して、生成されたインスタンスエーション・スクリプトを取得するときの、output_id に対するシステム生成番号を示します。

INstantiate\_Offline\_RepAPI ファンクション

このファンクションは、オフラインでリモート RepAPI スナップショット・サイトにスナップショット環境を構築するために使用するファイルを、マスター・サイトに生成します。このオフライン・インスタンスエーション・ファイルは、マスター・サイトに長時間接続できないリモート RepAPI サイトで使

用します。リモート・スナップショット・サイトが Oracle8i Lite (RepAPI を含みます) を実行するラップトップの場合、このファンクションを使用すると便利です。生成されたファイルは、FTP サイトに転送したり、CD-ROM やフロッピー・ディスクなどにロードできます。

このファンクションで生成されたファイルは、パラメータ OFFLINE\_DIRPATH によって指定されたマスター・サイトのディレクトリに格納されます。このファイルの名前は、USER\_NAME、REFRESH\_TEMPLATE\_NAME および SITE\_ID に基づいて指定され、拡張子 .oli が付きます。たとえば、サイト 1234 の、MYTEMPLATE テンプレートのユーザー SCOTT 用のオフラインのインスタンスエーション・ファイルの名前は次のようになります。

scott\_mytemplate\_1234.oli.

**注意：** このファンクションは、配置テンプレートのオフライン・インスタンスエーションに使用します。また、このファンクションは、他のユーザーのためにインスタンスエーションを行うレプリケーション管理者が使用します。独自のインスタンスエーションを行うユーザーは、パブリックの [INSTANTIATE\\_OFFLINE\\_REPAPI ファンクション](#) を使用します。詳細は、8-149 ページの [INSTANTIATE\\_OFFLINE\\_REPAPI ファンクション](#) を参照してください。

このファンクションを、DBMS\_OFFLINE\_OG パッケージのプロシージャ（マスター表のオフライン・インスタンスエーションに使用）や、DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャ（スナップショットのオフライン・インスタンスエーションに使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

## 構文

```
DBMS_REPCAT_RGT.INSTANTIATE_OFFLINE_REPAPI (
  refresh_template_name  IN   VARCHAR2,
  site_id                IN   VARCHAR2,
  user_name               IN   VARCHAR2  := USER,
  master                 IN   VARCHAR2  := NULL,
  url                    IN   VARCHAR2  := NULL,
  ssl                    IN   NUMBER    := 0,
  trace_vector            IN   NUMBER    := DBMS_REPCAT_RGT.NO_TRACE_DUMP,
  resultset_threshold    IN   NUMBER    := DBMS_REPCAT_INSTANTIATE.
                                     RESULTSET_THRESHOLD,
  lob_threshold           IN   NUMBER    := DBMS_REPCAT_INSTANTIATE.
                                     LOB_THRESHOLD);
```

**表 8-296 INSTANTIATE\_OFFLINE\_REPAPI ファンクションのパラメータ** (1 / 2 ページ)

パラメータ	説明
refresh_template_name	インスタンスエートされる配置テンプレートの名前です。
site_id	配置テンプレートをインスタンスエートしているリモート・サイトの ID 番号です。この番号にはスナップショット・サイトのサイト識別子を使用します。このパラメータの値は一時的なもののなので、後続の操作で RepAPI クライアントによって更新される可能性があります。
user_name	生成されるインスタンスエーション・ファイルのユーザーの名前です。

表 8-296 INstantiateOfflineRepAPI ファンクションのパラメータ (2 / 2 ページ)

パラメータ	説明
master	RepAPI クライアントがサーバー用に使用するオプションの別名です。このパラメータを指定した場合は、RepAPI クライアントでサーバーを使用するときは必ずこの別名を使用する必要があります。
url	データベースへのアクセスのために公開された、マスター・サイトの URL です。このパラメータを指定した場合は、RepAPI クライアントでサーバーを使用するときは必ずこの URL を使用する必要があります。
ssl	値が 1 の場合、スナップショットでマスター・サイトとの通信にセキュア・ソケット・レイヤー (SSL) が使用されます。値が 0 の場合、SSL は使用されません。
trace_vector	デバッグのトレース・レベルです。
resultset_threshold	スナップショットのリフレッシュ中に送信される、非 LOB 行データの最大サイズです。
lob_threshold	スナップショットのリフレッシュ中に送信される、LOB 行データの最大サイズです。

表 8-297 INstantiateOfflineRepAPI ファンクションの例外

例外	説明
miss_refresh_template	テンプレートが存在しません。
miss_user	ユーザー名がデータベース内に存在しません。
miss_template_site	ユーザーおよびサイト用に、テンプレートがインスタンスエートされていません。

戻り値

表 8-298 INstantiateOfflineRepAPI ファンクションの戻り値

戻り値	説明
0	エラーが発生しました。
1	エラーは発生しませんでした。

## INstantiate\_online ファンクション

このファンクションは、オンラインでリモート・スナップショット・サイトにスナップショット環境を構築する際に使用するスクリプトを、マスター・サイトに生成します。生成されたスクリプトは、マスター・サイトに長時間接続できるリモート・スナップショット・サイトで使用します。新しいスナップショットに設定されるデータの量によっては、リモート・スナップショット・サイトでのインスタンス化に時間がかかることがあります。このファンクションは、ユーザーのインスタンス化ごとに個別に実行する必要があります。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、Replication Manager などの Oracle Tools で使用します。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから適切な情報を取り出す際に使用します。

**注意:** このファンクションは、他のユーザーのためにインスタンス化を行うレプリケーション管理者が使用します。独自のインスタンス化を行うユーザーは、8-151 ページの「[INstantiate\\_online ファンクション](#)」で説明したパブリックの [INstantiate\\_online ファンクション](#)を使用します。

### 構文

```
DBMS_REPCAT_RGT.INSTANTIATE_ONLINE(  
  refresh_template_name  IN   VARCHAR2,  
  site_name              IN   VARCHAR2 := NULL,  
  user_name              IN   VARCHAR2 := NULL,  
  runtime_parm_id        IN   NUMBER   := -1e-130,  
  next_date              IN   DATE      := SYSDATE,  
  interval               IN   VARCHAR2 := 'SYSDATE + 1',  
  use_default_gowner     IN   BOOLEAN  := TRUE)  
return NUMBER;
```

### パラメータ

表 8-299 INstantiate\_online ファンクションのパラメータ (1 / 2 ページ)

パラメータ	説明
refresh_template_name	インスタンス化される配置テンプレートの名前です。
site_name	配置テンプレートをインスタンス化している、リモート・サイトの名前です。
user_name	配置テンプレートをインスタンス化している、許可を受けたユーザーの名前です。

表 8-299 Instantiate\_Online ファンクションのパラメータ (2 / 2 ページ)

パラメータ	説明
runtime_parm_id	INSERT_RUNTIME_PARMS プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します (ID は、GET_RUNTIME_PARM_ID ファンクションで取得済です)。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値を指定します。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔を指定します。
use_default_gowner	値が TRUE の場合、作成されたスナップショット・オブジェクト・グループは、デフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたスナップショット・オブジェクト・グループは、インスタンス化を実行するユーザーが所有します。

例外

表 8-300 Instantiate\_Online ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレートの名前が無効か、または存在しません。
miss_user	許可を受けたユーザーの名前が無効か、または存在しません。指定されたユーザーが DBA_REPCAT_USER_AUTHORIZATIONS ビューにリストされているかどうかを確認してください。このビューにリストされていないユーザーは、ターゲット配置テンプレートをインスタンス化するための許可を受けていません。
bad_parms	一部のテンプレート・パラメータが、定義されたユーザー・パラメータ値やテンプレートのデフォルト値で設定されませんでした。事前定義された値の数がテンプレート・パラメータの数と一致しなかったか、または事前定義された値がターゲット・パラメータでは無効 (タイプの不一致) でした。

戻り値

表 8-301 Instantiate\_Online ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT ビューから選択して、生成されたインスタンス化・スクリプトを取得するときの、output_id に対するシステム生成番号を示します。

## LOCK\_TEMPLATE\_EXCLUSIVE プロシージャ

LOCK\_TEMPLATE\_EXCLUSIVE プロシージャを使用して、配置テンプレートの更新または変更が行われている間、テンプレートの読取りやインスタンスエーションができないようにします。

ROLLBACK または COMMIT が実行されると、ロックが解除されます。

---

---

**注意：** 配置テンプレートに変更を加える前に、このプロシージャを実行する必要があります。

---

---

### 構文

```
DBMS_REPCAT_RGT.LOCK_TEMPLATE_EXCLUSIVE();
```

### パラメータ

なし

## LOCK\_TEMPLATE\_SHARED プロシージャ

指定された配置テンプレートを読取り専用にするには、LOCK\_TEMPLATE\_SHARED プロシージャを使用します。テンプレートをインスタンスエートする前に、このプロシージャをコールする必要があります。このプロシージャを実行すると、配置テンプレートがインスタンスエートされている間、誰もそのテンプレートを変更できません。

ROLLBACK または COMMIT が実行されると、ロックが解除されます。

### 構文

```
DBMS_REPCAT_RGT.LOCK_TEMPLATE_SHARED();
```

### パラメータ

なし

# DBMS\_REPUTIL パッケージ

## サブプログラムの概要

表 8-302 DBMS\_REPUTIL パッケージのサブプログラム

サブプログラム	説明
<a href="#">REPLICATION_OFF</a> プロシージャ (8-200 ページ)	レプリケート環境内の他のサイトに変更内容をレプリケートせずに表を変更するか、またはプロシージャ型レプリケーションを使用する際に行レベル・レプリケーションを使用禁止にします。
<a href="#">REPLICATION_ON</a> プロシージャ (8-201 ページ)	レプリケーションが一時的に中断された後で、変更内容のレプリケーションを再び有効にします。
<a href="#">REPLICATION_IS_ON</a> ファンクション (8-201 ページ)	レプリケーションが実行されているかどうかを判定します。
<a href="#">FROM_REMOTE</a> ファンクション (8-201 ページ)	内部レプリケーション・パッケージ内のプロシージャの開始で TRUE を、これらのプロシージャの終了で FALSE を戻します。
<a href="#">GLOBAL_NAME</a> ファンクション (8-202 ページ)	ローカル・データベースのグローバル・データベース名を確認します (戻り値がグローバル名です)。
<a href="#">MAKE_INTERNAL_PKG</a> プロシージャ (8-202 ページ)	内部パッケージとレプリケーション・カタログ内の表を同期させます。このプロシージャは、オラクル社カスタマ・サポート・センターからの指示に従って実行します。
<a href="#">SYNC_UP_REP</a> プロシージャ (8-203 ページ)	内部トリガーとレプリケーション・カタログ内の表またはスナップショットを同期させます。このプロシージャは、オラクル社カスタマ・サポート・センターからの指示に従って実行します。

## REPLICATION\_OFF プロシージャ

このプロシージャは、レプリケート環境内の他のサイトに変更内容をレプリケートせずに表を変更するか、またはプロシージャ型レプリケーションを使用する際に行レベル・レプリケーションを使用禁止にします。通常は、このフラグを設定する前に、レプリケート環境内のすべてのマスター・グループに対するレプリケーション・アクティビティを停止します。

### 構文

```
DBMS_REPUTIL.REPLICATION_OFF();
```

### パラメータ

なし



## REPLICATION\_ON プロシージャ

このプロシージャは、レプリケーションが一時的に中断された後で、変更内容のレプリケーションを再び有効にします。

### 構文

```
DBMS_REPUTIL.REPLICATION_ON();
```

### パラメータ

なし

## REPLICATION\_IS\_ON ファンクション

このファンクションは、レプリケーションが実行されているかどうかを判定します。戻り値が TRUE の場合は、生成されたレプリケーション・トリガーが有効になっています。FALSE の場合は、レプリケート・マスター・グループのカレント・サイトで、レプリケーションが使用禁止になっています。

このファンクションの戻り値は、DBMS\_REPUTIL パッケージの REPLICATION\_ON プロシージャまたは REPLICATION\_OFF プロシージャのコールで設定されます。

### 構文

```
DBMS_REPUTIL.REPLICATION_IS_ON()  
    return BOOLEAN;
```

### パラメータ

なし

## FROM\_REMOTE ファンクション

このファンクションは、内部レプリケーション・パッケージ内のプロシージャの開始で TRUE を、これらのプロシージャの終了で FALSE を戻します。内部パッケージによる更新の結果として発生するトリガーがある場合は、このファンクションをチェックする必要があります。

### 構文

```
DBMS_REPUTIL.FROM_REMOTE()  
    return BOOLEAN;
```

### パラメータ

なし

## GLOBAL\_NAME ファンクション

このファンクションは、ローカル・データベースのグローバル・データベース名を判定します（戻り値がグローバル名です）。

### 構文

```
DBMS_REPUTIL.GLOBAL_NAME()  
    return VARCHAR2;
```

### パラメータ

なし

## MAKE\_INTERNAL\_PKG プロシージャ

このプロシージャは、内部パッケージの有無を、レプリケーション・カタログ内の表と同期させます。表にレプリケーション・サポートがある場合は、このプロシージャを実行して内部パッケージを作成します。レプリケーション・サポートがない場合は、関連する内部パッケージを破棄します。

**注意：** このプロシージャは、必ずオラクル社カスタマ・サポート・センターからの指示に従って実行します。

### 構文

```
DBMS_REPUTIL.MAKE_INTERNAL_PKG (  
    canon_sname      IN   VARCHAR2,  
    canon_onsame      IN   VARCHAR2);
```

### パラメータ

表 8-303 MAKE\_INTERNAL\_PKG プロシージャのパラメータ

パラメータ	説明
canon_sname	同期する表が含まれるスキーマです。  このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。
canon_onsame	同期する表の名前です。  このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。

## SYNC\_UP\_REP プロシージャ

このプロシージャは、内部トリガーの有無を、レプリケーション・カタログ内の表またはスナップショットと同期させます。表またはスナップショットにレプリケーション・サポートがある場合は、このプロシージャを実行して内部レプリケーション・トリガーを作成します。レプリケーション・サポートがない場合は、関連する内部トリガーを破棄します。

**注意：** このプロシージャは、必ずオラクル社カスタム・サポート・センタからの指示に従って実行します。

### 構文

```
DBMS_REPUTIL.SYNC_UP_REP (  
    canon_sname      IN    VARCHAR2,  
    canon_onsame      IN    VARCHAR2);
```

### パラメータ

表 8-304 SYNC\_UP\_REP プロシージャのパラメータ

パラメータ	説明
canon_sname	同期する表またはスナップショットが含まれるスキーマです。 このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。
canon_onsame	同期する表またはスナップショットの名前です。 このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。

# DBMS\_SNAPSHOT パッケージ

## サブプログラムの概要

表 8-305 DBMS\_SNAPSHOT パッケージのサブプログラム

サブプログラム	説明
<a href="#">BEGIN_TABLE_REORGANIZATION プロシージャ</a> (8-204 ページ)	リフレッシュに必要なスナップショット・データの保存処理を実行します。
<a href="#">END_TABLE_REORGANIZATION プロシージャ</a> (8-205 ページ)	マスター表のスナップショット・データが有効であり、マスター表が適切な状態にあるようにします。
<a href="#">I_AM_A_REFRESH ファンクション</a> (8-206 ページ)	I_AM_REFRESH パッケージの状態の値を戻します。
<a href="#">PURGE_DIRECT_LOAD_LOG プロシージャ</a> (8-206 ページ)	スナップショットで不要になった行を、ダイレクト・ローダー・ログからパージします (データ・ウェアハウスで使用)。
<a href="#">PURGE_LOG プロシージャ</a> (8-206 ページ)	スナップショット・ログから行をパージします。
<a href="#">PURGE_SNAPSHOT_FROM_LOG プロシージャ</a> (8-207 ページ)	スナップショット・ログから行をパージします。
<a href="#">REFRESH プロシージャ</a> (8-209 ページ)	同じリフレッシュ・グループのメンバーではない1つ以上のスナップショットを、同じようにリフレッシュします。
<a href="#">REFRESH_ALL_MVIEWS プロシージャ</a> (8-211 ページ)	マスター表に加えられた変更が反映されていないスナップショットを、すべてリフレッシュします。
<a href="#">REFRESH_DEPENDENT プロシージャ</a> (8-212 ページ)	指定されたマスター表またはマスター表のリストに依存する表ベースのスナップショットを、すべてリフレッシュします。
<a href="#">REGISTER_SNAPSHOT プロシージャ</a> (8-214 ページ)	スナップショットの個別管理を有効にします。
<a href="#">UNREGISTER_SNAPSHOT プロシージャ</a> (8-215 ページ)	スナップショットの個別管理を有効にします。マスター・サイトでコールすると、スナップショットが登録解除されます。

## BEGIN\_TABLE\_REORGANIZATION プロシージャ

このプロシージャは、リフレッシュに必要なスナップショット・データの保存処理を実行します。このプロシージャは、マスター表が再編成される前にコールする必要があります。

**関連項目：** 詳細は、7-20 ページの「スナップショット・ログがあるマスター表の再編成」を参照してください。

構文

```
DBMS_SNAPSHOT.BEGIN_TABLE_REORGANIZATION (  
    tabowner      IN   VARCHAR2,  
    tabname       IN   VARCHAR2);
```

パラメータ

表 8-306 BEGIN\_TABLE\_REORGANIZATION プロシージャのパラメータ

パラメータ	説明
tabowner	再編成される表の所有者です。
tabname	再編成される表の名前です。

END\_TABLE\_REORGANIZATION プロシージャ

このプロシージャは、マスター表が再編成された後にコールする必要があります。このプロシージャは、マスター表のスナップショット・データが有効であり、マスター表が適切な状態にあることを確認します。

**関連項目：** 詳細は、7-20 ページの「スナップショット・ログがあるマスター表の再編成」を参照してください。

構文

```
DBMS_SNAPSHOT.END_TABLE_REORGANIZATION (  
    tabowner      IN   VARCHAR2,  
    tabname       IN   VARCHAR2);
```

パラメータ

表 8-307 END\_TABLE\_REORGANIZATION プロシージャのパラメータ

パラメータ	説明
tabowner	再編成される表の所有者です。
tabname	再編成される表の名前です。

### I\_AM\_A\_REFRESH ファンクション

このファンクションは、I\_AM\_REFRESH パッケージ状態の値を戻します。戻り値の TRUE は、スナップショットに対するすべてのローカル・レプリケーション・トリガーが、このセッションで使用禁止にされていることを示します。これは、各レプリケーション・トリガーが最初にこの状態を確認するためです。戻り値の FALSE は、これらのトリガーが使用可能になっていることを示します。

#### 構文

```
DBMS_SNAPSHOT.I_AM_A_REFRESH()  
    RETURN BOOLEAN;
```

#### パラメータ

なし

### PURGE\_DIRECT\_LOAD\_LOG プロシージャ

このプロシージャは、スナップショット（マテリアライズド・ビュー）で不要になったエントリを、ダイレクト・ローダー・ログからパージします。このプロシージャは、通常、Oracle のデータ・ウェアハウス技術を使用する環境で実行されます。

**関連項目：** 詳細は、『Oracle8i データ・ウェアハウス』を参照してください。

#### 構文

```
DBMS_SNAPSHOT.PURGE_DIRECT_LOAD_LOG();
```

### PURGE\_LOG プロシージャ

このプロシージャは、スナップショット・ログから行をパージします。

#### 構文

```
DBMS_SNAPSHOT.PURGE_LOG (  
    master      IN   VARCHAR2,  
    num         IN   BINARY_INTEGER := 1,  
    flag        IN   VARCHAR2      := 'NOP');
```

パラメータ

表 8-308 PURGE\_LOG プロシージャのパラメータ

パラメータ	説明
master	マスター表の名前です。
num	<p>スナップショット・ログから行が削除される、最後にリフレッシュされたスナップショットの数です。たとえば、次の文では、最後にリフレッシュされたスナップショットを2つ指定し、そのリフレッシュに必要な行をログから削除します。</p> <pre>dbms_snapshot.purge_log('master_table', 2);</pre> <p>スナップショット・ログのすべての行を削除するには、次の例で示すように、スナップショットの数に高い数値を指定します。</p> <pre>dbms_snapshot.purge_log('master_table', 9999);</pre> <p>MASTER_TABLE をベースとするスナップショットの数が 9999 より小さい場合、この文によって、MASTER_TABLE に対応するスナップショット・ログが完全にパージされます。スナップショット・ログから行がパージされている単純スナップショットは、次のリフレッシュ時に完全リフレッシュする必要があります。</p>
flag	<p>DELETE を指定し、少なくとも1つのスナップショットについて、スナップショット・ログから行が削除されるようにします。この引数は、引数 num の設定を上書きできます。たとえば、次の文では、スナップショット・ログに依存行を持つ最後にリフレッシュされたスナップショットから行が削除されます。</p> <pre>dbms_snapshot.purge_log('master_table', 1, 'DELETE');</pre>

PURGE\_SNAPSHOT\_FROM\_LOG プロシージャ

このプロシージャはマスター・サイトでコールされ、スナップショットのリフレッシュに関連するデータ・ディクショナリ表の行を削除します。このデータ・ディクショナリ表は、snapshot\_id または snapowner、snapname および snapsite の組合せで識別されるスナップショットのマスター・サイトに保持されます。指定されたスナップショットが、いずれかのマスター表で一番最初にリフレッシュされたスナップショットの場合は、スナップショット・ログもパージされます。このプロシージャでは、スナップショットの登録解除は行われません。

1つのスナップショット・ログのパージでエラーが発生した場合、それ以前に正常に行われたスナップショット・ログのパージはロールバックされません。これは、スナップショット・ログのサイズを最小に保つためです。エラーが発生した場合は、すべてのスナップショット・ログがパージされるまで、このプロシージャを再実行できます。

構文

```
DBMS_SNAPSHOT.PURGE_SNAPSHOT_FROM_LOG (  
    snapshot_id    IN    BINARY_INTEGER |  
    snapowner      IN    VARCHAR2,  
    snapname       IN    VARCHAR2,  
    snapsite       IN    VARCHAR2);
```

**注意：** このプロシージャはオーバーロードされています。snapshot\_id パラメータと、snapowner、snapname および snapsite パラメータは、同時には指定できません。

パラメータ

表 8-309 PURGE\_SNAPSHOT\_FROM\_LOG プロシージャのパラメータ

パラメータ	説明
snapshot_id	ターゲット・スナップショットの ID に基づいてこのプロシージャを実行する場合は、snapshot_id パラメータでスナップショット ID を指定します。スナップショット ID を確認するには、スナップショット・ログ・サイトで DBA_SNAPSHOT_LOGS ビューの問合せを行います。  ターゲット・スナップショットが、登録済スナップショットのリスト (DBA_REGISTERED_SNAPSHOTS) にない場合は、スナップショット ID に基づいてこのプロシージャを実行すると便利です。
snapowner	snapshot_id を指定しない場合は、snapowner パラメータを使用して、ターゲット・スナップショットの所有者を入力してください。スナップショットの所有者を確認するには、スナップショット・ログ・サイトで DBA_REGISTERED_SNAPSHOTS ビューの問合せを行います。
snapname	snapshot_id を指定しない場合は、snapname パラメータを使用して、ターゲット・スナップショットの名前を入力してください。スナップショット名を確認するには、スナップショット・ログ・サイトで DBA_REGISTERED_SNAPSHOTS ビューの問合せを行います。
snapsite	snapshot_id を指定しない場合は、snapsite パラメータを使用して、ターゲット・スナップショットのサイトを入力してください。スナップショット・サイトを確認するには、スナップショット・ログ・サイトで DBA_REGISTERED_SNAPSHOTS ビューの問合せを行います。



## REFRESH プロシージャ

このプロシージャは、リストで指定されたスナップショットをリフレッシュします。

### 構文

```
DBMS_SNAPSHOT.REFRESH (
  { list          IN          VARCHAR2,
    | tab          IN OUT DBMS_UTILITY.UNCL_ARRAY, }
  method          IN          VARCHAR2      := NULL,
  rollback_seg     IN          VARCHAR2      := NULL,
  push_deferred_rpc IN          BOOLEAN      := TRUE,
  refresh_after_errors IN        BOOLEAN      := FALSE,
  purge_option      IN          BINARY_INTEGER := 1,
  parallelism       IN          BINARY_INTEGER := 0,
  heap_size         IN          BINARY_INTEGER := 0,
  atomic_refresh    IN          BOOLEAN      := TRUE);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

### パラメータ

表 8-310 REFRESH プロシージャのパラメータ (1 / 3 ページ)

パラメータ	説明
list   tab	<p>リフレッシュするスナップショットの、カンマ区切りのリストです (シノニムはサポートされていません)。それぞれ異なるスキーマに置かれ異なるマスター表を持つスナップショットを指定できます。ただし、すべてのスナップショットが、ローカル・データベース内に存在する必要があります。</p> <p>カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY タイプの PL/SQL 表を使用できます。この表の各要素は、スナップショットの名前になります。</p>

表 8-310 REFRESH プロシージャのパラメータ (2 / 3 ページ)

パラメータ	説明
method	<p>指定されたスナップショットのリフレッシュ方法を示す、リフレッシュ方法の文字列です。F または f は高速リフレッシュ、? は強制リフレッシュ、C または c は完全リフレッシュ、A または a は常時リフレッシュを示します。リフレッシュ方法を指定していないスナップショットがあるとき、すなわち指定されたリフレッシュ方法の数を上回るスナップショットを指定した場合、そのスナップショットにはデフォルトのリフレッシュ方法が適用されます。例として、SQL*Plus 内の EXECUTE 文を次に示します。</p> <pre>dbms_snapshot.refresh ('s_emp,s_dept,scott.s_salary','CF');</pre> <p>この文では、S_EMP スナップショットの完全リフレッシュ、S_DEPT スナップショットの高速リフレッシュ、SCOTT.S_SALARY スナップショットのデフォルト・リフレッシュが実行されます。</p>
rollback_seg	<p>スナップショットのリフレッシュ時に使用するスナップショット・サイト・ロールバック・セグメントの名前です。</p>
push_deferred_rpc	<p>更新可能なスナップショットのみで使用します。スナップショットをリフレッシュする前に、スナップショットから対応付けられたマスターに変更をプッシュする場合は、このパラメータを TRUE に設定します。TRUE に設定しないと、変更が一時的に失われたように見えることがあります。</p>
refresh_after_errors	<p>このパラメータが TRUE の場合、スナップショットのマスター表の DEFERROR ビューに解消されていない競合が残っていても、更新可能なスナップショットのリフレッシュが続行されます。このパラメータが TRUE で atomic_refresh が FALSE の場合、このプロシージャで 1 つのスナップショットのリフレッシュが失敗しても、他のスナップショットのリフレッシュが続行されます。</p>
purge_option	<p>パラレル伝播メカニズムを使用している場合、つまり parallelism が 1 以上に設定されている場合、0 はページなし、1 は lazy パージ、2 は aggressive パージになります。ほとんどの場合、lazy パージが最適な設定になります。マルチマスター・レプリケーション・グループが異なるターゲット・サイトにプッシュされ、1 つ以上のレプリケーション・グループに対する更新が不定期に行われ不定期にプッシュされる場合は、aggressive パージを設定してキューを少なくします。すべてのレプリケーション・グループが不定期に更新およびプッシュされる場合は、ページなし (0) に設定して、随時 PUSH を実行します。PUSH を実行するときは、aggressive パージ (2) に設定してキューを削減します。</p>
parallelism	<p>0 の場合はシリアル伝播、<math>n &gt; 1</math> の場合は <math>n</math> パラレル・サーバー・プロセスのパラレル伝播、1 の場合は 1 パラレル・サーバー・プロセスのみのパラレル伝播です。</p>

表 8-310 REFRESH プロシージャのパラメータ ( 3 / 3 ページ)

パラメータ	説明
heap_size	パラレル伝播のスケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。
atomic_refresh	<p>このパラメータが TRUE の場合、リストで指定されたすべてのスナップショットは、1 つのトランザクションでリフレッシュされます。リフレッシュされたスナップショットはすべて、1 つのポイント・イン・タイムに更新されます。いずれかのスナップショットのリフレッシュに失敗すると、すべてのスナップショットが更新されません。</p> <p>このパラメータが FALSE の場合、各スナップショットは、それぞれ別個のトランザクションでリフレッシュされます。このパラメータが FALSE の場合は、ジョブ・キュー・プロセスの数を 1 以上に設定する必要があります。</p> <p>このパラメータが FALSE で、Summary Management オプションが購入されない場合は、エラーが発生します。</p>

REFRESH\_ALL\_MVIEWS プロシージャ

このプロシージャは、次のプロパティを持つすべてのスナップショット（マテリアライズド・ビュー）をリフレッシュします。

- スナップショットが依存するマスター表への最新の変更以後、スナップショットがリフレッシュされていません。
- スナップショットと、スナップショットが依存するマスター表のすべてがローカルにあります。
- スナップショットが DBA\_MVIEWS ビューにあります。

このプロシージャは、データ・ウェアハウスでの使用を目的としています。

構文

```
DBMS_SNAPSHOT.REFRESH_ALL_MVIEWS (  
  number_of_failures OUT BINARY_INTEGER,  
  method             IN  VARCHAR2      := NULL,  
  rollback_seg       IN  VARCHAR2      := NULL,  
  refresh_after_errors IN  BOOLEAN      := FALSE,  
  atomic_refresh      IN  BOOLEAN      := TRUE);
```

パラメータ

表 8-311 REFRESH\_ALL\_MVIEWS プロシージャのパラメータ

パラメータ	説明
number_of_failures	処理中に発生した失敗の数が戻されます。
method	各スナップショットに対して実行するリフレッシュのタイプを示す、単一のリフレッシュ方法です。F または f は高速リフレッシュ、? は強制リフレッシュ、C または c は完全リフレッシュ、A または a は常時リフレッシュを示します。方法を指定しない場合、スナップショットはデフォルトのリフレッシュ方法でリフレッシュされます。
rollback_seg	スナップショットのリフレッシュ時に使用するスナップショット・サイト・ロールバック・セグメントの名前です。
refresh_after_errors	このパラメータが TRUE の場合、スナップショットのマスター表の DEFERROR ビューに解消されていない競合が残っていても、更新可能なスナップショットのリフレッシュが続行されます。このパラメータが TRUE で atomic_refresh が FALSE の場合、このプロシージャで 1 つのスナップショットのリフレッシュが失敗しても、他のスナップショットのリフレッシュが続行されます。
atomic_refresh	<p>このパラメータが TRUE の場合、リフレッシュされたスナップショットは、1 つのトランザクションでリフレッシュされます。リフレッシュされたスナップショットはすべて、1 つのポイント・イン・タイムに更新されます。いずれかのスナップショットのリフレッシュに失敗すると、すべてのスナップショットが更新されません。</p> <p>このパラメータが FALSE の場合、リフレッシュ済の各スナップショットは、それぞれ別個のトランザクションでリフレッシュされます。このパラメータが FALSE の場合は、ジョブ・キュー・プロセスの数を 1 以上に設定する必要があります。</p>

REFRESH\_DEPENDENT プロシージャ

- このプロシージャは、次のプロパティを持つすべてのスナップショット（マテリアライズド・ビュー）をリフレッシュします。
- スナップショットが、指定されたマスター表リスト内のマスター表に依存します。
  - スナップショットが依存するマスター表への最新の変更以後、スナップショットがリフレッシュされていません。
  - スナップショットと、スナップショットが依存するマスター表のすべてがローカルにあります。
  - スナップショットが DBA\_MVIEWS ビューにあります。
- このプロシージャは、データ・ウェアハウスでの使用を目的としています。

構文

```
DBMS_SNAPSHOT.REFRESH_DEPENDENT (  
    number_of_failures    OUT    BINARY_INTEGER,  
    { list                IN     VARCHAR2,  
      | tab               IN OUT DBMS_UTILITY.UNCL_ARRAY, }  
    method                IN     VARCHAR2      := NULL,  
    rollback_seg          IN     VARCHAR2      := NULL,  
    refresh_after_errors  IN     BOOLEAN       := FALSE,  
    atomic_refresh        IN     BOOLEAN       := TRUE);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

パラメータ

表 8-312 REFRESH\_DEPENDENT プロシージャのパラメータ (1 / 2 ページ)

パラメータ	説明
number_of_failures	処理中に発生した失敗の数が戻されます。
list   tab	<p>スナップショットが依存できるマスター表の、カンマ区切りのリストです (シノニムはサポートされていません)。表とその表に依存するスナップショットが、異なるスキーマに置かれていても問題はありません。ただし、すべての表およびスナップショットが、ローカル・データベース内にある必要があります。</p> <p>カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY タイプの PL/SQL 表を使用できます。この表の各要素は、表の名前になります。</p>
method	<p>依存スナップショットのリフレッシュ方法を示す、リフレッシュ方法の文字列です。特定の表に依存するすべてのスナップショットが、その表に対応付けられたリフレッシュ方法でリフレッシュされます。F または f は高速リフレッシュ、? は強制リフレッシュ、C または c は完全リフレッシュ、A または a は常時リフレッシュを示します。リフレッシュ方法を指定していない表があるとき、つまりリフレッシュ方法の数を上回る数の表を指定した場合、その表に依存するすべてのスナップショットには、デフォルトのリフレッシュ方法が適用されます。例として SQL*Plus 内の EXECUTE 文を次に示します。</p> <pre>dbms_snapshot.refresh_dependent ('emp,dept,scott.salary','CF');</pre> <p>この文では、EMP 表に依存するスナップショットの完全リフレッシュ、DEPT 表に依存するスナップショットの高速リフレッシュ、および SCOTT.SALARY 表に依存するスナップショットのデフォルト・リフレッシュが実行されます。</p>

表 8-312 REFRESH\_DEPENDENT プロシージャのパラメータ (2 / 2 ページ)

パラメータ	説明
rollback_seg	スナップショットのリフレッシュ時に使用するスナップショット・サイト・ロールバック・セグメントの名前です。
refresh_after_errors	このパラメータが TRUE の場合、スナップショットのマスター表の DEFERROR ビューに解消されていない競合が残っていても、更新可能なスナップショットのリフレッシュが続行されます。このパラメータが TRUE で atomic_refresh が FALSE の場合、このプロシージャで 1 つのスナップショットのリフレッシュが失敗しても、他のスナップショットのリフレッシュが続行されます。
atomic_refresh	<p>このパラメータが TRUE の場合、リフレッシュされたスナップショットは、1 つのトランザクションでリフレッシュされます。リフレッシュされたスナップショットはすべて、1 つのポイント・イン・タイムに更新されます。いずれかのスナップショットのリフレッシュに失敗すると、すべてのスナップショットが更新されません。</p> <p>このパラメータが FALSE の場合、リフレッシュ済の各スナップショットは、それぞれ別個のトランザクションでリフレッシュされます。このパラメータが FALSE の場合は、ジョブ・キュー・プロセスの数を 1 以上に設定する必要があります。</p> <p>このパラメータが FALSE で、Summary Management オプションが購入されない場合は、エラーが発生します。</p>

REGISTER\_SNAPSHOT プロシージャ

このプロシージャを実行すると、スナップショットの個別管理が有効になります。

構文

```
DBMS_SNAPSHOT.REGISTER_SNAPSHOT (  
    snapowner    IN    VARCHAR2,  
    snapname     IN    VARCHAR2,  
    snapsite     IN    VARCHAR2,  
    {snapshot_id IN    DATE | BINARY_INTEGER,  
    flag         IN    BINARY_INTEGER,}  
    qry_txt      IN    VARCHAR2,  
    rep_type     IN    BINARY_INTEGER := DBMS_SNAPSHOT.REG_UNKNOWN);
```

**注意：** このプロシージャはオーバーロードされています。snapshot\_id パラメータと flag パラメータは、同時には指定できません。

パラメータ

表 8-313 REGISTER\_SNAPSHOT プロシージャのパラメータ

パラメータ	説明
sowner	スナップショットの所有者です。
snapname	スナップショットの名前です。
snapsite	Oracle8 以降のマスターで登録されたスナップショットのスナップショット・サイトの名前です。二重引用符は使用できません。
snapshot_id	スナップショットの ID 番号です。Oracle8 のスナップショットは BINARY_INTEGER として指定します。Oracle8 以降のマスター・サイトに登録されているリリース 7 のスナップショットは、DATE として指定します。
flag	後続の CREATE または MOVE 文が問合せテキストに登録されているかどうかを示す、PL/SQL パッケージ変数です。
query_txt	スナップショット定義問合せの最初の 32,000 バイトです。
rep_type	スナップショットのバージョンです。使用できる定数は、reg_unknown (デフォルト)、reg_v7_group、reg_v8_group および reg_repapi_group です。

使用にあたっての注意

このプロシージャは、リモート・スナップショット・サイトからリモート・プロシージャ・コールを使用してマスター・サイトで実行されます。REGISTER\_SNAPSHOT が同じ SNAPOWNER、SNAPNAME および SNAPSITE で複数回コールされた場合は、SNAPSHOT\_ID、FLAG および QUERY\_TXT の最新の値が格納されます。問合せが VARCHAR2 の最大サイズを超えた場合、QUERY\_TXT は問合せの最初の 32000 文字になり、残りの文字は切り捨てられます。このプロシージャを手動で実行する場合、実行者がスナップショット・ビューで SNAPSHOT\_ID および FLAG の値を参照する必要があります。

スナップショットの問合せがマスター・サイトで登録されないようにする場合は、オプションを FALSE に設定して、SET\_REGISTER\_QUERY\_TEXT プロシージャをコールします。最新のオプション設定を確認するには、DDL を発行する前に、スナップショット・サイトで GET\_REG\_QUERY\_TEXT\_FLAG ファンクションをコールします。

UNREGISTER\_SNAPSHOT プロシージャ

このプロシージャを実行すると、スナップショットの個別管理が有効になります。マスター・サイトで実行され、スナップショットの登録を解除します。

構文

```
DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT (  
    snapowner      IN   VARCHAR2,  
    snapname       IN   VARCHAR2,  
    snapsite       IN   VARCHAR2);
```

パラメータ

表 8-314 UNREGISTER\_SNAPSHOT プロシージャのパラメータ

パラメータ	説明
snapowner	スナップショットの所有者です。
snapname	スナップショットの名前です。
snapsite	スナップショット・サイトの名前です。



---

## データ・ディクショナリ・ビュー

この章では、アドバンスド・レプリケーション機能の使用時に役立つデータ・ディクショナリ・ビューを説明します。ビューは、次の3つのグループにアルファベット順で分類されています。

- レプリケーション・カタログ・ビュー
- 遅延トランザクション・ビュー
- スナップショット・ビューおよびスナップショット・リフレッシュ・グループ・ビュー

## レプリケーション・カタログ・ビュー

レプリケーション機能をサイトにインストールすると、同じサイトにレプリケーション・カタログが自動的にインストールされます。このレプリケーション・カタログには、表とビューが含まれています。[図 9-1](#) に示すように、このビューは、マスター・サイトやスナップショット・サイトに対して、レプリケートされるオブジェクトのタイプ、レプリケート先、レプリケーションの実行中にエラーが発生していないかどうか、などを判定するために使用されます。

---

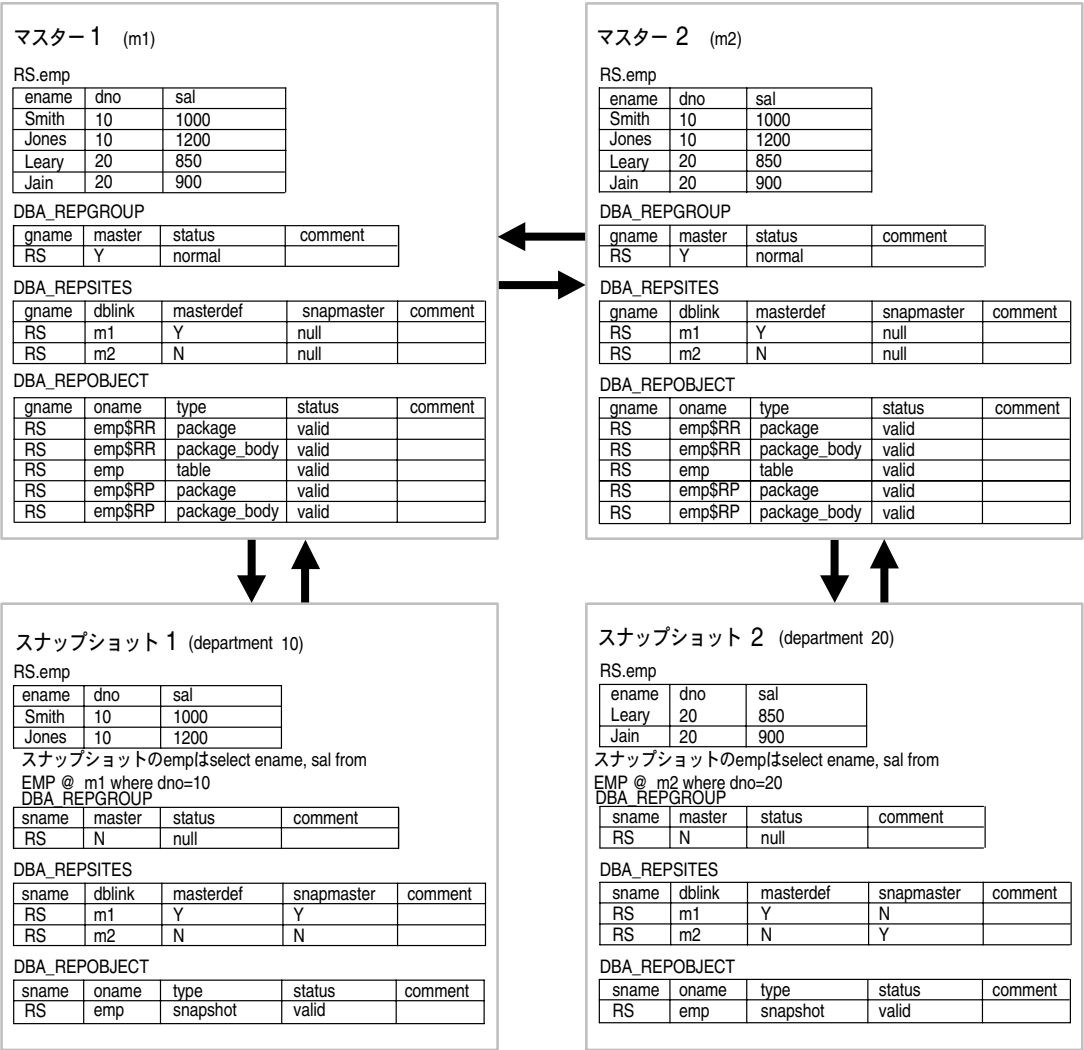
---

**注意：** レプリケーション・カタログの表は直接変更しないでください。変更が必要な場合は、DBMS\_REPCAT パッケージに用意されているプロシージャを使用してください。

---

---

図 9-1 レプリケーション・カタログ・ビュー



多くのデータ・ディクショナリ表には、対応するビューが3種類あります。

- ALL\_ ビューには、カレント・ユーザーのスキーマの情報など、カレント・ユーザーからアクセス可能な情報がすべて表示されます。ユーザーが他のスキーマのオブジェクトにアクセスする権限またはロールを付与されている場合には、それらのオブジェクトの情報も表示されます。
- DBA\_ ビューには、データベース全体の関連情報がすべて表示されます。DBA\_ ビューは、管理者のみを対象としています。SELECT\_ANY\_TABLE 権限を持つユーザーのみがアクセスできます。この権限は Oracle の初期インストール時にデータベース管理者のロールに割り当てられます。
- USER\_ ビューには、カレント・ユーザーのスキーマの情報がすべて表示されます。USER\_ ビューの問合せを行うために特別な権限は必要ありません。

1 つのデータ・ディクショナリ表に対応する ALL\_、DBA\_ および USER\_ ビューの列は、通常ほぼ同一です。このため、詳細は、ALL\_ ビューの項でまとめて説明します。DBA\_ ビューと USER\_ ビューの詳細は説明しませんが、ALL\_ ビューとの相違点があれば記述しています。

この項では、[表 9-1](#) にリストするレプリケーション・カタログ・ビューを説明します。

表 9-1 レプリケーション・カタログ・ビュー

ALL_ ビュー	DBA_ ビュー	USER_ ビュー
<a href="#">ALL_REPCATLOG</a>	<a href="#">DBA_REPCATLOG</a>	<a href="#">USER_REPCATLOG</a>
<a href="#">ALL_REPCAT_REFRESH_TEMPLATES</a>	<a href="#">DBA_REPCAT_REFRESH_TEMPLATES</a>	<a href="#">USER_REPCAT_REFRESH_TEMPLATES</a>
<a href="#">ALL_REPCAT_TEMPLATE_OBJECTS</a>	<a href="#">DBA_REPCAT_TEMPLATE_OBJECTS</a>	<a href="#">USER_REPCAT_TEMPLATE_OBJECTS</a>
<a href="#">ALL_REPCAT_TEMPLATE_PARS</a>	<a href="#">DBA_REPCAT_TEMPLATE_PARS</a>	<a href="#">USER_REPCAT_TEMPLATE_PARS</a>
<a href="#">ALL_REPCAT_TEMPLATE_SITES</a>	<a href="#">DBA_REPCAT_TEMPLATE_SITES</a>	<a href="#">USER_REPCAT_TEMPLATE_SITES</a>
<a href="#">ALL_REPCAT_USER_AUTHORIZATIONS</a>	<a href="#">DBA_REPCAT_USER_AUTHORIZATIONS</a>	<a href="#">USER_REPCAT_USER_AUTHORIZATIONS</a>
<a href="#">ALL_REPCAT_USER_PARM_VALUES</a>	<a href="#">DBA_REPCAT_USER_PARM_VALUES</a>	<a href="#">USER_REPCAT_USER_PARM_VALUES</a>
<a href="#">ALL_REPCOLUMN</a>	<a href="#">DBA_REPCOLUMN</a>	<a href="#">USER_REPCOLUMN</a>
<a href="#">ALL_REPCOLUMN_GROUP</a>	<a href="#">DBA_REPCOLUMN_GROUP</a>	<a href="#">USER_REPCOLUMN_GROUP</a>
<a href="#">ALL_REPCONFLICT</a>	<a href="#">DBA_REPCONFLICT</a>	<a href="#">USER_REPCONFLICT</a>
<a href="#">ALL_REPDDL</a>	<a href="#">DBA_REPDDL</a>	<a href="#">USER_REPDDL</a>
<a href="#">ALL_REPGENOBJECTS</a>	<a href="#">DBA_REPGENOBJECTS</a>	<a href="#">USER_REPGENOBJECTS</a>
<a href="#">ALL_REPGROUP</a>	<a href="#">DBA_REPGROUP</a>	<a href="#">USER_REPGROUP</a>
<a href="#">ALL_REPGROUP_PRIVILEGES</a>	<a href="#">DBA_REPGROUP_PRIVILEGES</a>	<a href="#">USER_REPGROUP_PRIVILEGES</a>
<a href="#">ALL_REPGROUPED_COLUMN</a>	<a href="#">DBA_REPGROUPED_COLUMN</a>	<a href="#">USER_REPGROUPED_COLUMN</a>
<a href="#">ALL_REPKEY_COLUMNS</a>	<a href="#">DBA_REPKEY_COLUMNS</a>	<a href="#">USER_REPKEY_COLUMNS</a>
<a href="#">ALL_REPOBJECT</a>	<a href="#">DBA_REPOBJECT</a>	<a href="#">USER_REPOBJECT</a>
<a href="#">ALL_REPPARAMETER_COLUMN</a>	<a href="#">DBA_REPPARAMETER_COLUMN</a>	<a href="#">USER_REPPARAMETER_COLUMN</a>
<a href="#">ALL_REPPRIORITY</a>	<a href="#">DBA_REPPRIORITY</a>	<a href="#">USER_REPPRIORITY</a>
<a href="#">ALL_REPPRIORITY_GROUP</a>	<a href="#">DBA_REPPRIORITY_GROUP</a>	<a href="#">USER_REPPRIORITY_GROUP</a>
<a href="#">ALL_REPPROP</a>	<a href="#">DBA_REPPROP</a>	<a href="#">USER_REPPROP</a>
<a href="#">ALL_REPRESOL_STATS_CONTROL</a>	<a href="#">DBA_REPRESOL_STATS_CONTROL</a>	<a href="#">USER_REPRESOL_STATS_CONTROL</a>
<a href="#">ALL_REPRESOLUTION</a>	<a href="#">DBA_REPRESOLUTION</a>	<a href="#">USER_REPRESOLUTION</a>
<a href="#">ALL_REPRESOLUTION_METHOD</a>	<a href="#">DBA_REPRESOLUTION_METHOD</a>	<a href="#">USER_REPRESOLUTION_METHOD</a>
<a href="#">ALL_REPRESOLUTION_STATISTICS</a>	<a href="#">DBA_REPRESOLUTION_STATISTICS</a>	<a href="#">USER_REPRESOLUTION_STATISTICS</a>
<a href="#">ALL_REPSITES</a>	<a href="#">DBA_REPSITES</a>	<a href="#">USER_REPSITES</a>

## ALL\_REPCATLOG

各マスター・サイトで生成された非同期管理要求およびエラー・メッセージの暫定的なステータスの情報が示されます。このビューに含まれる管理要求およびエラー・メッセージは、カレント・ユーザーからアクセス可能です。要求の処理中に生成されたすべてのエラー・メッセージは、その要求を作成したマスター・サイトの ALL\_REPCATLOG ビューに転送されます。エラーが発生せずに管理要求が正常に完了した場合、この要求に関するすべてのトレースは、ALL\_REPCATLOG ビューから削除されます。

### 関連するビュー：

- DBA\_REPCATLOG には、データベース内のすべての非同期管理要求およびエラー・メッセージのステータスが示されます。
- USER\_REPCATLOG には、カレント・ユーザーが所有するすべての非同期管理要求およびエラー・メッセージのステータスが示されます。

列	データ型	NULL	説明
ID	NUMBER		順序番号。ID 列と SOURCE 列の値により、1 つの管理要求に関連するすべてのログ・レコード（全マスター・サイトを含む）が識別されます。
SOURCE	VARCHAR2 (128)		要求が作成された場所。
USERID	VARCHAR2 (30)		要求を作成したユーザー名。
TIMESTAMP	DATE		要求が作成された時刻。
ROLE	VARCHAR2 (9)		サイトがマスター定義サイト（masterdef）か、マスター・サイト（master）かを示します。
MASTER	VARCHAR2 (128)		role 列が 'masterdef' のときにリモート・タスクを実行する場合、どのマスターがタスクを実行するかを示します。
SNAME	VARCHAR2 (30)		レプリケート・オブジェクトのスキーマ名（該当する名前がある場合）。
REQUEST	VARCHAR2 (29)		実行された DBMS_REPCAT 管理プロシージャの名前。
ONAME	VARCHAR2 (30)		レプリケート・オブジェクトの名前（該当する名前がある場合）。
TYPE	VARCHAR2 (12)		レプリケート・オブジェクトのタイプ。
STATUS	VARCHAR2 (14)		管理要求のステータス。ready、do_callback、await_callback、error のいずれかです。
MESSAGE	VARCHAR2 (200)		戻されたエラー・メッセージ。
ERRNUM	NUMBER		Oracle エラー・メッセージ番号。
GNAME	VARCHAR2 (30)		レプリケート・オブジェクト・グループの名前。

## ALL\_REPCAT\_REFRESH\_TEMPLATES

カレント・ユーザーからアクセス可能な、各配置テンプレートに関するグローバル情報が示されます。テンプレート名、テンプレートの所有者、テンプレート・オブジェクトが属するリフレッシュ・グループ、テンプレートの種類（プライベートまたはパブリック）などの情報があります。

データベース管理者は、テンプレートのコンテナにスナップショット定義を追加するときに、該当する REFRESH\_TEMPLATE\_NAME を参照します。特定のテンプレートに追加されたスナップショットは、REFRESH\_GROUP\_NAME で指定されたリフレッシュ・グループに追加されます。

パブリックとして作成された配置テンプレートは、マスター・サイトへの接続を許可されているユーザーであれば、どのユーザーも使用できます。一方、プライベートとして作成された配置テンプレートは、ALL\_REPCAT\_USER\_AUTHORIZATIONS ビューにリストされるユーザーのみが使用できます。

### 関連するビュー：

- DBA\_REPCAT\_REFRESH\_TEMPLATES には、データベース内のすべての配置テンプレートが示されます。
- USER\_REPCAT\_REFRESH\_TEMPLATES には、カレント・ユーザーが所有するすべての配置テンプレートが示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)		配置テンプレートの名前。
OWNER	VARCHAR2 (30)		配置テンプレートの所有者。
REFRESH_GROUP_NAME	VARCHAR2 (30)		インスタンスエーション・プロセス中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPLATE_COMMENT	VARCHAR2 (2000)		ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)		Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。

ALL\_REPCAT\_TEMPLATE\_OBJECTS

カレント・ユーザーからアクセス可能な、各配置テンプレートの各オブジェクト定義が示されます。REFRESH\_TEMPLATE\_NAME にターゲット・テンプレートを指定すると、各オブジェクトがテンプレートに追加されます。

DDL\_TEXT には、テンプレートをパラメータ化するための変数を含めることができます。変数を作成するには、変数名の先頭にコロン (:) を入力します (:region など)。テンプレートをパラメータ化しておく、テンプレートのインスタンスエーション・プロセスを柔軟に行うことができます (スナップショット・サイト固有のデータ・セットを定義する場合など)。

オブジェクトがテンプレートに追加されると、指定された DDL が検索されます。定義済パラメータが検索されると、それらのパラメータは、自動的に ALL\_REPCAT\_TEMPLATE\_PARMs ビューに追加されます。

関連するビュー：

- DBA\_REPCAT\_TEMPLATE\_OBJECTS には、データベース内のすべての配置テンプレートのオブジェクト定義が示されます。
- USER\_REPCAT\_TEMPLATE\_OBJECTS には、カレント・ユーザーが所有する各配置テンプレートのオブジェクト定義が示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートの名前。
OBJECT_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートのオブジェクト名。
OBJECT_TYPE	VARCHAR2 (17)		配置テンプレートのオブジェクトのタイプ。
DDL_TEXT	CLOB (4000)		配置テンプレートのオブジェクトを作成するために実行される DDL。
MASTER_ROLLBACK_SEGMENT	VARCHAR2 (30)		配置テンプレートのオブジェクトをインスタンスエートするときに使用されるロールバック・セグメントの名前。
DERIVED_FROM_SNAME	VARCHAR2 (30)		テンプレート・オブジェクトの基になったオブジェクトを含むスキーマが表示されます (該当する値がある場合)。
DERIVED_FROM_ONAME	VARCHAR2 (30)		テンプレート・オブジェクトの基になったオブジェクトの名前が表示されます (該当する値がある場合)。
FLAVOR_ID	NUMBER		配置テンプレートのオブジェクトの固有 ID。



DDL\_TEXT パラメータは CLOB として定義されているため、ALL\_REPCAT\_TEMPLATE\_OBJECTS ビューに対して単純に SELECT 文を実行するとエラーが戻されます。オブジェクト DDL を参照する必要がある場合は、次の SELECT 文を使用してください (DDL\_TEXT パラメータは除外してください)。

```
SELECT refresh_template_name, object_name, object_type, master_rollback_seg, flavor_id
FROM dba_repcat_template_objects;
```

次に、カーソルと DBMS\_LOB パッケージを使用し、ALL\_REPCAT\_TEMPLATE\_OBJECTS ビューの内容をすべて表示するためのスクリプトを示します。DDL\_TEXT 列を含む ALL\_REPCAT\_TEMPLATE\_OBJECTS ビューの内容をすべて表示するには、次のスクリプトを使用します。

```
SET SERVEROUTPUT ON

DECLARE
  CURSOR mycursor IS
    SELECT refresh_template_name, object_name, object_type, ddl_text,
           master_rollback_seg, flavor_id
    FROM dba_repcat_template_objects;
  tempstring VARCHAR2(1000);
  len NUMBER;
BEGIN
  FOR myrec IN mycursor LOOP
    len := DBMS_LOB.GETLENGTH(myrec.ddl_text);
    DBMS_LOB.READ(myrec.ddl_text, len, 1, tempstring);
    DBMS_OUTPUT.PUT_LINE(myrec.refresh_template_name||' '||
                          myrec.object_name||' '||myrec.object_type||' '||tempstring||' '||
                          myrec.master_rollback_seg||' '||myrec.flavor_id);
  END LOOP;
END;
/
```

**関連項目：** カーソルの使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド 基礎編』を参照してください。DBMS\_LOB パッケージと LOB の一般的な使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

ALL\_REPCAT\_TEMPLATE\_PARMS

カレント・ユーザーからアクセス可能な、すべてのテンプレートのオブジェクト DDL で定義されたパラメータが含まれます。オブジェクトがテンプレートに追加されると、DDL に変数があるかどうかを検査されます。パラメータが見つかると、自動的にこのビューに追加されます。

このビューでは、デフォルトのパラメータ値およびプロンプト文字列も定義できます。これらの値を定義すると、インスタンスエーション・プロセスでテンプレートが使用しやすくなります。

関連項目： 9-8 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_OBJECTS](#)」

関連するビュー：

- DBA\_REPCAT\_TEMPLATE\_PARMS には、データベース内のすべての配置テンプレートのテンプレート・パラメータが示されます。
- USER\_REPCAT\_TEMPLATE\_PARMS には、カレント・ユーザーが所有するすべての配置テンプレートのテンプレート・パラメータが示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートの名前。
OWNER	VARCHAR2 (30)	NOT NULL	配置テンプレートの所有者。
REFRESH_GROUP_NAME	VARCHAR2 (30)	NOT NULL	インスタンスエーション・プロセス中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPLATE_COMMENTS	VARCHAR2 (2000)		ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)		Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。
PARAMETER_NAME	VARCHAR2 (30)	NOT NULL	パラメータ名。
DEFAULT_PARM_VALUE	CLOB (4000)		デフォルトのパラメータ値。
PROMPT_STRING	VARCHAR2 (2000)		パラメータのプロンプト文字列。
USER_OVERRIDE	VARCHAR2 (1)		Y の場合、ユーザーによるデフォルトのパラメータ値の上書きが可能です。 N の場合、ユーザーによるデフォルトのパラメータ値の上書きが不可能です。

DEFAULT\_PARM\_VALUE は CLOB として定義されているため、ALL\_REPCAT\_TEMPLATE\_PARMS ビューに対して単純に SELECT 文を実行するとエラーが戻されます。デフォルトのパラメータ値を参照する必要がない場合は、次の SELECT 文を使用してください (DEFAULT\_PARM\_VALUE は除外してください)。

```
SELECT refresh_template_name, owner, refresh_group_name, template_comment,
       public_template, parameter_name, prompt_string, user_override
FROM dba_repcat_template_parms;
```

次に、カーソルと DBMS\_LOB パッケージを使用して、ALL\_REPCAT\_TEMPLATE\_PARMS ビューの内容をすべて表示するためのスクリプトを示します。DEFAULT\_PARM\_VALUE 列を含む ALL\_REPCAT\_TEMPLATE\_PARMS ビューの内容をすべて表示するには、次のスクリプトを使用します。

```
SET SERVEROUTPUT ON
```

```
DECLARE
  CURSOR mycursor IS
    SELECT refresh_template_name, owner, refresh_group_name,
           template_comment, public_template, parameter_name, default_parm_value,
           prompt_string, user_override
    FROM dba_repcat_template_parms;
  tempstring VARCHAR2(1000);
  len NUMBER;
BEGIN
  FOR myrec IN mycursor LOOP
    len := DBMS_LOB.GETLENGTH(myrec.default_parm_value);
    DBMS_LOB.READ(myrec.default_parm_value, len, 1, tempstring);
    DBMS_OUTPUT.PUT_LINE(myrec.refresh_template_name||' '||
      myrec.owner||' '||myrec.refresh_group_name||' '||
      myrec.template_comment||' '||myrec.public_template||' '||
      myrec.parameter_name||' '||tempstring||' '||myrec.prompt_string||' '||
      myrec.user_override);
  END LOOP;
END;
/
```

**関連項目：** カーソルの使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド 基礎編』を参照してください。DBMS\_LOB パッケージと LOB の一般的な使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

ALL\_REPCAT\_TEMPLATE\_SITES

エンタープライズ・ネットワークにおける、テンプレートのインスタンス化の  
カレント・ステータス情報が示されます。カレント・ユーザーからアクセス可能な、配置テン  
プレートのインスタンス化・サイトの情報も含まれます。具体的には、データベース  
管理者は、RepAPI サイトを含む特定のサイトで実行されるテンプレートのインストールや  
削除を監視できます。

関連するビュー：

- DBA\_REPCAT\_TEMPLATE\_SITES には、データベース内のすべてのテンプレートのリ  
モート・インスタンス化・サイトがすべて示されます。
- USER\_REPCAT\_TEMPLATE\_SITES には、カレント・ユーザーが所有するすべてのテン  
プレートのリモート・インスタンス化・サイトが示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートの名前。
REFRESH_GROUP_NAME	VARCHAR2 (30)		インスタンス化・プロセス中にテンプレ ート・オブジェクトが追加されるリフレッシュ・グ ループの名前。
TEMPLATE_OWNER	VARCHAR2 (30)		配置テンプレートの所有者とみなされるユーザーの 名前。
USER_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートをインストールしたユー ザーの名前。
SITE_NAME	VARCHAR2 (128)		配置テンプレートのターゲット・スナップショッ ト・サイト。RepAPI サイトの場合、このフィール ドは NULL になります。
REPAPI_SITE_NAME	VARCHAR2 (128)		表示されている配置テンプレートを実際にインス タインストールした RepAPI サイトの ID。RepAPI 以外 のクライアントの場合、このフィールドは NULL になります。
STATUS	VARCHAR2 (10)		ターゲット・スナップショット・サイトにおける配 置テンプレートのステータスが表示されます。 0 = インストールされていません。 1 = インストール済です。 -1 = インストールされていますがエラーが発生して います。

## ALL\_REPCAT\_USER\_AUTHORIZATIONS

カレント・ユーザーからアクセス可能なプライベート配置テンプレートに対する権限を与えられたユーザーが表示されます。このビューにリストされたユーザーは、指定されたテンプレートをインスタンスエートできます。このビューに表示されないユーザーは配置テンプレートをインスタンスエートできません。

### 関連するビュー：

- DBA\_REPCAT\_USER\_AUTHORIZATIONS には、データベース内のすべてのプライベート配置テンプレートに対する権限を与えられたユーザーがリストされます。
- USER\_REPCAT\_USER\_AUTHORIZATIONS には、カレント・ユーザーが所有するプライベート配置テンプレートに対する権限を与えられたユーザーがリストされます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	ユーザーがインスタンスエートする権限を与えられた配置テンプレートの名前。
OWNER	VARCHAR2 (30)	NOT NULL	配置テンプレートの所有者の名前。
REFRESH_GROUP_NAME	VARCHAR2 (30)	NOT NULL	インスタンスエーション・プロセス中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPLATE_COMMENT	VARCHAR2 (2000)		ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)		Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。
USER_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートをインスタンスエートする権限を与えられたユーザーの名前。

ALL\_REPCAT\_USER\_PARM\_VALUES

カレント・ユーザーからアクセス可能なすべての配置テンプレートのテンプレート・パラメータが示されます。データベース管理者には、テンプレートを配布してインスタンスエートする前に、ユーザー・パラメータの表を作成するオプション機能があります。指定されたユーザーがテンプレートをインスタンスエートする場合は、そのユーザー用に ALL\_REPCAT\_USER\_PARM\_VALUES ビューに格納された値が自動的に使用されます。

関連するビュー：

- DBA\_REPCAT\_USER\_PARM\_VALUES には、データベース内のすべての配置テンプレートのテンプレート・パラメータが示されます。
- USER\_REPCAT\_USER\_PARM\_VALUES には、カレント・ユーザーが所有するすべての配置テンプレートのテンプレート・パラメータが示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	ユーザー・パラメータ値が定義された配置テンプレートの名前。
OWNER	VARCHAR2 (30)	NOT NULL	配置テンプレートの所有者の名前。
REFRESH_GROUP_NAME	VARCHAR2 (30)	NOT NULL	インスタンスエーション・プロセス中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPATE_COMMENT	VARCHAR2 (2000)		ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)		Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。
PARAMETER_NAME	VARCHAR2 (30)	NOT NULL	ユーザー・パラメータ値が定義されたパラメータの名前。
DEFAULT_PARM_VALUE	CLOB (4000)		パラメータのデフォルト値。
PROMPT_STRING	VARCHAR2 (2000)		パラメータのプロンプト文字列。
PARM_VALUE	CLOB (4000)		指定されたユーザーに対して定義されたパラメータ値。
USER_NAME	VARCHAR2 (30)	NOT NULL	指定されたパラメータ値が定義されたユーザーの名前。

DEFAULT\_PARM\_VALUE および PARM\_VALUE は CLOB として定義されているため、ALL\_REPCAT\_USER\_PARM\_VALUES ビューに対して単純に SELECT 文を実行するとエラーが戻されます。デフォルトのパラメータ値やユーザー・パラメータ値を参照する必要がある場合は、次の SELECT 文を使用してください (DEFAULT\_PARM\_VALUE および PARM\_VALUE は除外してください)。

```
SELECT refresh_template_name, owner, refresh_group_name, template_comment,
       public_template, parameter_name, prompt_string, user_name
FROM dba_repcat_user_parm_values;
```

次に、カーソルと DBMS\_LOB パッケージを使用して、ALL\_REPCAT\_USER\_PARM\_VALUES ビューの内容をすべて表示するためのスクリプトを示します。DEFAULT\_PARM\_VALUE 列および PARM\_VALUE 列を含む ALL\_REPCAT\_TEMPLATE\_PARMS ビューの内容をすべて表示するには、次のスクリプトを使用します。

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
    CURSOR mycursor IS
```

```
        SELECT refresh_template_name, owner, refresh_group_name,
               template_comment, public_template, parameter_name, default_parm_value,
               prompt_string, parm_value, user_name
        FROM dba_repcat_user_parm_values;
```

```
    tempstring VARCHAR2(1000);
```

```
    tempstring2 varchar2(1000);
```

```
    len NUMBER;
```

```
BEGIN
```

```
    FOR myrec IN mycursor LOOP
```

```
        len := DBMS_LOB.GETLENGTH(myrec.default_parm_value);
```

```
        DBMS_LOB.READ(myrec.default_parm_value, len, 1, tempstring);
```

```
        DBMS_OUTPUT.PUT_LINE(myrec.refresh_template_name||' '||
```

```
                               myrec.owner||' '||myrec.refresh_group_name||' '||
```

```
                               myrec.template_comment||' '||myrec.public_template||' '||
```

```
                               myrec.parameter_name||' '||tempstring||' '||myrec.prompt_string||' '||
```

```
                               tempstring2||' '||myrec.user_name);
```

```
    END LOOP;
```

```
END;
```

```
/
```

**関連項目：** カーソルの使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド 基礎編』を参照してください。DBMS\_LOB パッケージと LOB の一般的な使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

# ALL\_REPCOLUMN

カレント・ユーザーからアクセス可能な表の、レプリケートされた列が表示されます。

## 関連するビュー：

- DBA\_REPCOLUMN には、データベース内のすべての表の、レプリケートされた列が表示されます。
- USER\_REPCOLUMN には、カレント・ユーザーが所有するすべての表の、レプリケートされた列が表示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	オブジェクトの所有者の名前。
ONAME	VARCHAR2 (30)	NOT NULL	オブジェクトの名前。
TYPE	VARCHAR2 (8)		オブジェクトのタイプ。
CNAME	VARCHAR2 (30)	NOT NULL	レプリケートされた列の名前。
ID	NUMBER		レプリケートされた列の ID 番号。
POS	NUMBER	NOT NULL	レプリケートされた列の順序。
COMPARE_OLD_ON_DELETE	VARCHAR2 (1)		レプリケートされた情報の削除時に、列の古い値と比較するかどうかを示します。
COMPARE_OLD_ON_UPDATE	VARCHAR2 (1)		レプリケートされた情報の更新時に、列の古い値と比較するかどうかを示します。
SEND_OLD_ON_DELETE	VARCHAR2 (1)		レプリケートされた情報の削除時に、列の古い値を送信するかどうかを示します。
SEND_OLD_ON_UPDATE	VARCHAR2 (1)		レプリケートされた情報の更新時に、列の古い値を送信するかどうかを示します。
CTYPE	VARCHAR2 (9)		列のタイプが表示されます。
DATA_LENGTH	NUMBER		列の長さがバイト単位で表示されます。
DATA_PRECISION	NUMBER		列の精度が表示されます。NUMBER 列の場合は 10 進数で表示され、FLOAT 列の場合は 2 進数で表示されます。
DATA_SCALE	NUMBER		小数点以下の位の桁数が表示されます。
NULLABLE	VARCHAR2 (1)		列に NULL 値を指定できるかどうかを示します。
CHARACTER_SET_NAME	VARCHAR2 (44)		列で使用するキャラクタ・セットの名前が表示されます（該当する値がある場合）。



## ALL\_REPCOLUMN\_GROUP

カレント・ユーザーからアクセス可能な各レプリケート表の列グループが示されます。

### 関連するビュー：

- DBA\_REPCOLUMN\_GROUP には、データベース内のすべての表の列グループが示されます。
- USER\_REPCOLUMN\_GROUP には、カレント・ユーザーが所有するすべての表の列グループが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
GROUP_NAME	VARCHAR2 (30)	NOT NULL	列グループの名前。
GROUP_COMMENT	VARCHAR2 (80)		ユーザーが入力するコメント。

**注意：** USER\_REPCOLUMN\_GROUP ビューには、SNAME 列が示されません。

## ALL\_REPCONFLICT

競合解消方法が定義されていて、カレント・ユーザーからアクセス可能な各表の名前と、その方法で解消される競合のタイプが含まれます。

### 関連するビュー：

- DBA\_REPCONFLICT には、競合解消方法が定義されているデータベース内のすべての表に対する競合解消方法が示されます。
- USER\_REPCONFLICT には、競合解消方法が定義されている、カレント・ユーザーが所有するすべての表に対する競合解消方法が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	競合解消方法が定義された表の名前。
CONFLICT_TYPE	VARCHAR2 (10)		競合解消方法を適用する競合のタイプ。削除競合、一意性競合、更新競合のいずれかです。
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	方法を適用するオブジェクト。削除競合では表の名前です。一意性競合では制約の名前です。更新競合では列グループの名前です。

**注意：** USER\_REPCONFLICT ビューには、SNAME 列が示されません。

ALL\_REPDDL

カレント・ユーザーからアクセス可能な各レプリケーション・オブジェクトの DDL が示されます。

関連するビュー：

- DBA\_REPDDL には、データベース内の各レプリケート・オブジェクトの DDL が示されます。
- USER\_REPDDL には、カレント・ユーザーが所有する各レプリケート・オブジェクトの DDL が示されます。

列	データ型	NULL	説明
LOG_ID	NUMBER		ALL_REPCATLOG レコードの ID 番号。
SOURCE	VARCHAR2 (128)		要求の作成元データベースの名前。
ROLE	VARCHAR2 (1)		Y の場合、データベースは要求に対するマスター定義サイト（masterdef）です。 N の場合、データベースはマスター・サイトです。
MASTER	VARCHAR2 (128)		この要求を処理するデータベースの名前。
LINE	NUMBER (38)		1 つの要求に含まれるレコードの順序。
TEXT	VARCHAR2 (2000)		引数または DDL テキストの一部分。

ALL\_REPGENOBJECTS

カレント・ユーザーからアクセス可能で、レプリケーションのサポート用に生成されたオブジェクトが示されます。

**関連するビュー：**

- DBA\_REPGENOBJECTS には、レプリケーションのサポート用に生成された、データベース内の各オブジェクトが示されます。
- USER\_REPGENOBJECTS には、レプリケーションのサポート用に生成された、カレント・ユーザーが所有する各オブジェクトが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)		レプリケート・スキーマの名前。
ONAME	VARCHAR2 (30)		生成されたオブジェクトの名前。
TYPE	VARCHAR2 (12)		生成されたオブジェクトのタイプ。
BASE_SNAME	VARCHAR2 (30)		ベースとなるオブジェクトの所有者。
BASE_ONAME	VARCHAR2 (30)		このオブジェクトを生成する原因となったオブジェクト。
BASE_TYPE	VARCHAR2 (12)		ベースとなるオブジェクトのタイプ。
PACKAGE_PREFIX	VARCHAR2 (30)		パッケージ・ラッパーの接頭辞。
PROCEDURE_PREFIX	VARCHAR2 (30)		パッケージ・ラッパーのプロシージャ接頭辞。
DISTRIBUTED	VARCHAR2 (1)		この列は廃止されています。
REASON	VARCHAR2 (30)		オブジェクトの生成理由。

**ALL\_REPGROUP**

レプリケートの対象となる、カレント・ユーザーからアクセス可能なオブジェクト・グループがすべて示されます。各オブジェクト・グループのメンバーは、ALL\_REPOBJECT ビューにリストされます。

**関連するビュー：**

- DBA\_REPGROUP には、レプリケートの対象となる、データベース内のオブジェクト・グループがすべて示されます。
- USER\_REPGROUP には、レプリケートの対象となる、カレント・ユーザーが所有するオブジェクト・グループがすべて示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。リリース 7.3 以降では廃止されています。
MASTER	VARCHAR2 (1)		Y の場合、カレント・サイトはマスター・サイトです。 N の場合、カレント・サイトはスナップショット・サイトです。
STATUS	VARCHAR2 (9)		マスター・サイトに対してのみ使用されます。ステータスは、通常、休止、停止のいずれかです。
SCHEMA_COMMENT	VARCHAR2 (80)		ユーザーが入力するコメント。
GNAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクト・グループの名前。
FNAME	VARCHAR2 (30)		固有の名前。
RPC_PROCESSING_DISABLED	VARCHAR2 (1)		N の場合、このサイトでの遅延 RPC の受信や適用が可能です。 Y の場合、このサイトでの遅延 RPC の受信や適用が不可能です。
OWNER	VARCHAR2 (30)	NOT NULL	レプリケーション・グループの所有者。

ALL\_REPGROUP\_PRIVILEGES

カレント・ユーザーからアクセス可能で、オブジェクト・グループ内で権限を登録されているユーザーの情報が示されます。

関連するビュー：

- DBA\_REPGROUP\_PRIVILEGES には、データベース内のすべてのオブジェクト・グループ内で権限を登録されているユーザーの情報が示されます。
- USER\_REPGROUP\_PRIVILEGES には、カレント・ユーザーが所有するオブジェクト・グループ内で権限を登録されているユーザーの情報が示されます。

列	データ型	NULL	説明
USERNAME	VARCHAR2 (30)	NOT NULL	ユーザーの名前が表示されます。
GNAME	VARCHAR2 (30)		レプリケート・オブジェクト・グループの名前が表示されます。
CREATED	DATE	NOT NULL	オブジェクト・グループの登録日が表示されます。
RECEIVER	VARCHAR2 (1)		ユーザーが受信者としての権限を持っているかどうかを示します。

列	データ型	NULL	説明
PROXY_SNAPADMIN	VARCHAR2 (1)		ユーザーが proxy_snapadmin の権限を持っているかどうかを示します。
OWNER	VARCHAR2 (30)		レプリケーション・グループの所有者。

## ALL\_REPGROUPED\_COLUMN

カレント・ユーザーからアクセス可能な各表の列グループを構成する、すべての列が示されます。

### 関連するビュー：

- DBA\_REPGROUPED\_COLUMN には、データベース内の各表の列グループを構成する、すべての列が示されます。
- USER\_REPGROUPED\_COLUMN には、カレント・ユーザーが所有する各表の列グループを構成する、すべての列が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
GROUP_NAME	VARCHAR2 (30)	NOT NULL	列グループの名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	列グループに含まれる列の名前。

---

**注意：** USER\_REPGROUPED\_COLUMN ビューには、SNAME 列が示されません。

---

## ALL\_REPKEY\_COLUMNS

カレント・ユーザーからアクセス可能な各表の 1 つ以上の主キー列が示されます。

### 関連するビュー：

- DBA\_REPKEY\_COLUMNS には、データベース内の各表の 1 つ以上の主キー列が示されます。
- USER\_REPKEY\_COLUMNS には、カレント・ユーザーが所有する各表の 1 つ以上の主キー列が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表の所有者。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
COL	VARCHAR2 (30)	NOT NULL	表内の 1 つ以上の主キー列。

ALL\_REPOBJECT

カレント・ユーザーからアクセス可能な各レプリケート・オブジェクト・グループ内のオブジェクトの情報が示されます。オブジェクトは、1 つのオブジェクト・グループのみに属します。レプリケート・オブジェクト・グループは、複数のスキーマにまたがることができます。

関連するビュー：

- DBA\_REPOBJECT には、データベースの各レプリケート・オブジェクト・グループ内のオブジェクトの情報が示されます。
- USER\_REPOBJECT には、カレント・ユーザーが所有する各レプリケート・オブジェクト・グループ内のオブジェクトの情報が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)		レプリケート・オブジェクトを含むスキーマの名前。
ONAME	VARCHAR2 (30)		レプリケート・オブジェクトの名前。
TYPE	VARCHAR2 (16)		レプリケート・オブジェクトのタイプ。タイプには、表、ビュー、パッケージ、パッケージ本体、プロシージャ、ファンクション、索引、シノニム、トリガーまたはスナップショットがあります。
STATUS	VARCHAR2 (9)		"CREATE" は、ユーザーまたは Oracle によって生成された DDL が、ローカル・データベースに対して適用され、オブジェクトがローカルに作成されることを示します。ローカル・レプリカが存在する場合、Oracle は、そのレプリカをマスター定義と比較して、両者に整合性があることを確認します。オブジェクトの作成または比較が正常に実行されると、ステータスが VALID に更新されます。正常に実行されなかった場合、ステータスは ERROR になります。オブジェクトを削除すると、ALL_REPOBJECT ビューから行が削除される前にステータスが DROPPED に更新されます。

列	データ型	NULL	説明
GENERATION_STATUS	VARCHAR2 (9)		オブジェクトでレプリケーション・パッケージを生成する必要があるかどうかを示します。
ID	NUMBER		ローカル・データベース・オブジェクトの識別子 (該当する値がある場合)。
OBJECT_COMMENT	VARCHAR2 (80)		ユーザーが入力するコメント。
GNAME	VARCHAR2 (30)		オブジェクトが属する、レプリケート・オブジェクト・グループの名前。
MIN_COMMUNICATION	VARCHAR2 (1)		Y の場合、更新されたビューに対して新しい値のみが送信されます。 N の場合、更新されたビューに対して古い値と新しい値が送信されます。
REPLICATION_TRIGGER_EXISTS	VARCHAR2 (1)		Y の場合、内部レプリケーション・トリガーが存在します。 N の場合、内部レプリケーション・トリガーは存在しません。
INTERNAL_PACKAGE_EXISTS	VARCHAR2 (1)		Y の場合、内部パッケージが存在します。 N の場合、内部パッケージは存在しません。
GROUP_OWNER	VARCHAR2 (30)		レプリケーション・グループの所有者。

## ALL\_REPPARAMETER\_COLUMN

ALL\_REPPARAMETER\_COLUMN ビューには、ALL\_REPRESOLUTION ビューに示される情報と、カレント・ユーザーからアクセス可能な各レプリケート表の競合解消に使用される列の情報が示されます。これらの列の値は、LIST\_OF\_COLUMN\_NAMES の引数として、DBMS\_REPCAT パッケージの ADD\_conflicttype\_RESOLUTION プロシージャに渡されます。

### 関連するビュー：

- DBA\_REPPARAMETER\_COLUMN には、データベース内の各レプリケート表の競合解消に使用される列の情報が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
CONFLICT_TYPE	VARCHAR2 (10)		競合解消方法を適用する競合のタイプ。削除競合、一意性競合、更新競合のいずれかです。
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	方法を適用するオブジェクト。削除競合では表の名前です。一意性競合では制約の名前です。更新競合では列グループの名前です。
SEQUENCE_NO	NUMBER	NOT NULL	競合解消方法を適用する順序。1 は、最初に適用されることを示します。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle によって提供される競合解消方法。ユーザー指定の方法を使用する場合、この値は 'user function' になります。
FUNCTION_NAME	VARCHAR2 (92)	NOT NULL	競合解消方法が 'user function' の場合は、ユーザーが入力した競合解消方法の名前になります。
PRIORITY_GROUP	VARCHAR2 (30)		競合解消方法が 'priority group' の場合は、優先グループの名前になります。
PARAMETER_TABLE_NAME	VARCHAR2 (30)	NOT NULL	パラメータ列が属する表の名前が表示されます。
PARAMETER_COLUMN_NAME	VARCHAR2 (30)	NOT NULL	競合解消方法で IN パラメータとして使用される列の名前。
PARAMETER_SEQUENCE_NO	NUMBER	NOT NULL	IN パラメータとして使用される列の順序。

**注意：** USER\_REPPARAMETER\_COLUMN ビューには、SNAME 列が示されません。

ALL\_REPPRIORITY

カレント・ユーザーからアクセス可能な各優先グループの各メンバーの値および優先順位レベルが示されます。優先グループには、レプリケート・オブジェクト・グループ内で一意の名前を付ける必要があります。また、指定された優先グループ内のそれぞれの優先順位レベルと値は、そのグループ内で一意であることが必要です。

関連するビュー：

- DBA\_REPPRIORITY には、データベース内の各優先グループの各メンバーの値および優先順位レベルが示されます。
- USER\_REPPRIORITY には、カレント・ユーザーが所有する各優先グループの各メンバーの値および優先順位レベルが示されます。



列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。リリース 7.3 以降では廃止されています。
PRIORITY_GROUP	VARCHAR2 (30)	NOT NULL	優先グループまたはサイト優先順位グループの名前。
PRIORITY	NUMBER	NOT NULL	メンバーの優先順位レベル。最も大きい数値は、優先順位レベルが最も高いことを示します。
DATA_TYPE	VARCHAR2 (9)		優先グループの値のデータ型。
FIXED_DATA_LENGTH	NUMBER (38)		CHAR データ型の値の最大長。
CHAR_VALUE	CHAR (255)		優先グループのメンバーの値 (DATA_TYPE = CHAR の場合)。
VARCHAR2_VALUE	VARCHAR2 (4000)		優先グループのメンバーの値 (DATA_TYPE = VARCHAR2 の場合)。
NUMBER_VALUE	NUMBER		優先グループのメンバーの値 (DATA_TYPE = NUMBER の場合)。
DATE_VALUE	DATE		優先グループのメンバーの値 (DATA_TYPE = DATE の場合)。
RAW_VALUE	RAW (2000)		優先グループのメンバーの値 (DATA_TYPE = RAW の場合)。
GNAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクト・グループの名前。
NCHAR_VALUE	CHAR (500)		優先グループのメンバーの値 (DATA_TYPE = NCHAR の場合)。
NVARCHAR2_VALUE	VARCHAR2 (1000)		優先グループのメンバーの値 (DATA_TYPE = NVARCHAR2 の場合)。
LARGE_CHAR_VALUE	CHAR (2000)		優先グループのメンバーの値 (255 文字以上の空白埋込み文字列の場合)。

**注意：** USER\_REPPRIORITY ビューには、SNAME 列および GNAME 列が示されません。

## ALL\_REPPRIORITY\_GROUP

カレント・ユーザーからアクセス可能な各レプリケート・オブジェクト・グループに定義された、優先グループまたはサイト優先順位グループが示されます。

### 関連するビュー：

- DBA\_REPPRIORITY\_GROUP には、データベース内の各レプリケート・オブジェクト・グループに定義された、優先グループまたはサイト優先順位グループが示されます。
- USER\_REPPRIORITY\_GROUP には、カレント・ユーザーが所有する各レプリケート・オブジェクト・グループに定義された、優先グループまたはサイト優先順位グループが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。リリース 7.3 以降では廃止されています。
PRIORITY_GROUP	VARCHAR2 (30)	NOT NULL	優先グループまたはサイト優先順位グループの名前。
DATA_TYPE	VARCHAR2 (9)		優先グループの各値のデータ型。
FIXED_DATA_LENGTH	NUMBER (38)		CHAR データ型の値の最大長。
PRIORITY_COMMENT	VARCHAR2 (80)		ユーザーが入力するコメント。
GNAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクト・グループの名前。

**注意：** USER\_REPPRIORITY ビューには、SNAME 列および GNAME 列が示されません。

## ALL\_REPPROP

各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。このビューに示されるのは、カレント・ユーザーからアクセス可能なオブジェクトです。操作の伝播は、ストアド・プロシージャやプロシージャ・ラップ・プログラムをコールしたり、表を直接操作することによって実行します。

### 関連するビュー：

- DBA\_REPPROP には、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。このビューに示されるのは、データベース内のすべてのオブジェクトです。
- USER\_REPPROP には、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。このビューに示されるのは、カレント・ユーザーが所有するオブジェクトです。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクトを含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクトの名前。
TYPE	VARCHAR2 (16)		レプリケートされているオブジェクトのタイプ。
DBLINK	VARCHAR2 (128)	NOT NULL	マスター・サイトのデータベースの完全修飾名。このデータベースに対して、変更が伝播されます。
HOW	VARCHAR2 (13)		伝播の実行方法。ローカル・マスター・サイトの場合は 'none' です。それ以外のサイトの場合は 'synchronous' または 'asynchronous' です。
PROPAGATE_COMMENT	VARCHAR2 (80)		ユーザーが入力するコメント。

## ALL\_REPRESOL\_STATS\_CONTROL

カレント・ユーザーからアクセス可能なすべてのレプリケート表に対する、競合解消の統計情報が示されます。

### 関連するビュー：

- DBA\_REPRESOL\_STATS\_CONTROL には、データベース内のすべてのレプリケート表に対する、競合解消の統計情報が示されます。
- USER\_REPRESOL\_STATS\_CONTROL には、カレント・ユーザーが所有するすべてのレプリケート表に対する、競合解消の統計情報が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	表の所有者。
ONAME	VARCHAR2 (30)	NOT NULL	表の名前。
CREATED	DATE	NOT NULL	統計収集の開始を示すタイムスタンプ。
STATUS	VARCHAR2 (9)		統計収集のステータス。ACTIVE または CANCELLED になります。
STATUS_UPDATE_DATE	DATE	NOT NULL	前回のステータス更新日を示すタイムスタンプ。
PURGED_DATE	DATE		前回の統計データのパージ日を示すタイムスタンプ。
LAST_PURGE_START_DATE	DATE		前回の統計情報のパージ期間の開始日。
LAST_PURGE_END_DATE	DATE		前回の統計情報のパージ期間の終了日。

**注意：** USER\_REPRESOL\_STATS\_CONTROL ビューには、SNAME 列が示されません。

ALL\_REPRESOLUTION

カレント・ユーザーからアクセス可能な各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。

関連するビュー：

- DBA\_REPRESOLUTION には、データベース内の各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。
- USER\_REPRESOLUTION には、カレント・ユーザーが所有する各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、行レベル・レプリケーションによってレプリケートされています。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
CONFLICT_TYPE	VARCHAR2 (10)		競合解消方法を適用する競合のタイプ。削除競合、一意性競合、更新競合のいずれかです。
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	方法を適用するオブジェクト。削除競合では表の名前です。一意性競合では制約の名前です。更新競合では列グループの名前です。
SEQUENCE_NO	NUMBER	NOT NULL	競合解消方法を適用する順序。1 は、最初に適用されることを示します。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle によって提供される競合解消方法の名前。ユーザー指定の方法を使用する場合、この値は 'user function' になります。
FUNCTION_NAME	VARCHAR2 (92)	NOT NULL	競合解消方法が 'user function' の場合は、ユーザーが入力した競合解消方法の名前になります。
PRIORITY_GROUP	VARCHAR2 (30)		競合解消方法が 'priority group' の場合は、優先グループの名前になります。
RESOLUTION_COMMENT	VARCHAR2 (80)		ユーザーが入力するコメント。

**注意：** USER\_REPRESOLUTION ビューには、SNAME 列が示されません。

ALL\_REPRESOLUTION\_METHOD

データベース内で使用可能な競合解消方法がすべてリストされます。まず、このビューには、Oracle レプリケーションが提供する標準の方法がリストされます。新しいユーザー・ファンクションを作成して、データベース内のオブジェクト用の競合解消方法として追加すると、そのファンクションがこのビューに追加されます。

関連するビュー：

- DBA\_REPRESOLUTION\_METHOD には、データベース内で使用可能な競合解消方法がすべてリストされます。
- USER\_REPRESOLUTION\_METHOD には、カレント・ユーザーが所有する使用可能な競合解消方法がすべてリストされます。

列	データ型	NULL	説明
CONFLICT_TYPE	VARCHAR2 (10)		競合解消方法を適用する競合のタイプ。更新競合、一意性競合、削除競合のいずれかです。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle が提供する競合解消方法の名前。またはユーザーが入力した方法の名前。

ALL\_REPRESOLUTION\_STATISTICS

カレント・ユーザーからアクセス可能なすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。これらの統計情報は、DBMS\_REPCAT.REGISTER\_STATISTICS プロシージャをコールした場合に限り、表に対して収集されます。

関連するビュー：

- DBA\_REPRESOLUTION\_STATISTICS には、データベース内のすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。
- USER\_REPRESOLUTION\_STATISTICS には、カレント・ユーザーが所有するすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
CONFLICT_TYPE	VARCHAR2 (10)		正常に解消された競合のタイプ。削除競合、一意性競合、更新競合のいずれかです。
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	競合解消方法を適用するオブジェクト。削除競合では表の名前です。一意性競合では制約の名前です。更新競合では列グループの名前です。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle によって提供される競合解消方法の名前。ユーザー指定の方法を使用する場合、この値は 'user function' になります。
FUNCTION_NAME	VARCHAR2 (92)		競合解消方法が 'user function' の場合は、ユーザーが入力した競合解消方法の名前になります。
PRIORITY_GROUP	VARCHAR2 (30)		競合解消方法が 'priority group' の場合は、優先グループの名前になります。
RESOLVED_DATE	DATE	NOT NULL	この行の競合が解消された日付。
PRIMARY_KEY_VALUE	VARCHAR2 (2000)	NOT NULL	行の主キーの連結表現。

**注意：** USER\_REPRESOLUTION\_STATISTICS ビューには、SNAME 列が示されません。

ALL\_REPSITES

カレント・ユーザーからアクセス可能な各レプリケート・オブジェクト・グループのメンバーがリストされます。

関連するビュー：

- DBA\_REPSITES には、データベース内の各レプリケート・オブジェクト・グループのメンバーがリストされます。
- USER\_REPSITES には、カレント・ユーザーが所有する各レプリケート・オブジェクト・グループのメンバーがリストされます。

列	データ型	NULL	説明
GNAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクト・グループの名前。
DBLINK	VARCHAR2 (128)	NOT NULL	このオブジェクト・グループのマスター・サイトへのデータベース・リンク。
MASTERDEF	VARCHAR2 (1)		マスター定義サイトとして指定する DBLINK を示します。
SNAPMASTER	VARCHAR2 (1)		リフレッシュ時に使用する DBLINK を示します。スナップショット・サイトの場合に使用されます。
MASTER_COMMENT	VARCHAR2 (80)		ユーザーが入力するコメント。
MASTER	VARCHAR2 (1)		Y の場合、サイトはレプリケート・グループのマスター・サイトです。 N の場合、サイトはレプリケート・グループのマスター・サイトではありません。
GROUP_OWNER	VARCHAR2 (30)	NOT NULL	レプリケーション・グループの所有者。

DBA\_REPSITES ビューには、次の列も含まれます。

PROP_UPDATES	NUMBER	伝播方法のコード化（マスターの場合）。
MY_DBLINK	VARCHAR2 (1)	インポート後、問題を検出するために使用されます。Y の場合、DBLINK がグローバル名です。

## DBA\_REPCATLOG

各マスター・サイトの DBA\_REPCATLOG ビューには、非同期管理要求および生成されたエラー・メッセージの暫定的なステータスの情報が示されます。要求の処理中に生成されたすべてのエラー・メッセージは、その要求を作成したマスター・サイトの DBA\_REPCATLOG ビューに転送されます。エラーが発生せずに管理要求が正常に完了した場合は、この要求に関するすべてのトレースは、DBA\_REPCATLOG ビューから削除されます。このビューの列は、9-6 ページの「[ALL\\_REPCATLOG](#)」の列と同じです。

## DBA\_REPCAT\_REFRESH\_TEMPLATES

DBA\_REPCAT\_REFRESH\_TEMPLATES ビューには、データベース内の各配置テンプレートに関するグローバル情報が示されます。テンプレート名、テンプレートの所有者、テンプレート・オブジェクトが属するリフレッシュ・グループ、テンプレートの種類（プライベートまたはパブリック）などの情報があります。

このビューの列は、ALL\_REPCAT\_REFRESH\_TEMPLATES の列と同じです。このビューおよび列の詳細は、9-7 ページの「[ALL\\_REPCAT\\_REFRESH\\_TEMPLATES](#)」を参照してください。

## DBA\_REPCAT\_TEMPLATE\_OBJECTS

DBA\_REPCAT\_TEMPLATE\_OBJECTS ビューには、データベース内のすべての配置テンプレートの各オブジェクト定義が示されます。REFRESH\_TEMPLATE\_NAME にターゲット・テンプレートを指定すると、各オブジェクトがテンプレートに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_OBJECTS の列と同じです。このビューおよび列の詳細は、9-8 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_OBJECTS](#)」を参照してください。

## DBA\_REPCAT\_TEMPLATE\_PARMS

データベース内のすべてのテンプレートのオブジェクト DDL で定義されたパラメータは、DBA\_REPCAT\_TEMPLATE\_PARMS 表に格納されます。オブジェクトがテンプレートに追加されると、DDL に変数があるかどうかを検査されます。パラメータが見つかったら、自動的にこのビューに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_PARMS の列と同じです。このビューおよび列の詳細は、9-10 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_PARMS](#)」を参照してください。

## DBA\_REPCAT\_TEMPLATE\_SITES

DBA\_REPCAT\_TEMPLATE\_SITES ビューは、エンタープライズ・ネットワークのすべてのサイトにおける、テンプレートのインスタンス化のカレント・ステータス情報をデータベース管理者に提供します。このビューには、データベース内のすべての配置テンプレートのインスタンス化・サイトの情報が示されます。具体的には、データベース管理者は、RepAPI サイトを含む特定のサイトで実行されるテンプレートのインストールや削除を監視できます。このビューの列は、9-12 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_SITES](#)」の列と同じです。

## DBA\_REPCAT\_USER\_AUTHORIZATIONS

DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューには、プライベート用に指定された、データベース内のすべてのテンプレートに対する権限を与えられたユーザーがリストされます。このビューにリストされたユーザーは、指定されたテンプレートをインスタンス化できます。このビューにリストされないユーザーはテンプレートをインスタンス化できません。このビューの列は、9-13 ページの「[ALL\\_REPCAT\\_USER\\_AUTHORIZATIONS](#)」の列と同じです。

## DBA\_REPCAT\_USER\_PARM\_VALUES

DBA\_REPCAT\_USER\_PARM\_VALUES ビューには、データベース内のすべての配置テンプレートのテンプレート・パラメータが示されます。データベース管理者は、テンプレートを配布してインスタンス化する前に、ユーザー・パラメータの表を作成するオプションの機能があります。指定されたユーザーがテンプレートをインスタンス化する場合は、そのユーザー用に DBA\_REPCAT\_USER\_PARM\_VALUES 表に格納された値が自動的に使用されます。



このビューの列は、ALL\_REPCAT\_USER\_PARM\_VALUES の列と同じです。このビューおよび列の詳細は、9-14 ページの「[ALL\\_REPCAT\\_USER\\_PARM\\_VALUES](#)」を参照してください。

## DBA\_REPCOLUMN

DBA\_REPCOLUMN ビューには、データベース内のすべての表のレプリケートされた列がリストされます。このビューの列は、9-16 ページの「[ALL\\_REPCOLUMN](#)」の列と同じです。

## DBA\_REPCOLUMN\_GROUP

DBA\_REPCOLUMN\_GROUP ビューには、データベース内の各レプリケート表の列グループがリストされます。このビューの列は、9-17 ページの「[ALL\\_REPCOLUMN\\_GROUP](#)」の列と同じです。

## DBA\_REPCONFLICT

DBA\_REPCONFLICT ビューには、競合解消方法が定義されている、データベース内の各表の名前と、競合解消方法を適用する競合のタイプが表示されます。このビューの列は、9-17 ページの「[ALL\\_REPCONFLICT](#)」の列と同じです。

## DBA\_REPDDL

DBA\_REPDDL ビューには、データベース内のレプリケーション・オブジェクトの DDL が示されます。このビューの列は、9-18 ページの「[ALL\\_REPDDL](#)」の列と同じです。

## DBA\_REPGENOBJECTS

DBA\_REPGENOBJECTS ビューには、レプリケーションのサポート用に生成された、データベース内の各オブジェクトが示されます。このビューの列は、9-18 ページの「[ALL\\_REPGENOBJECTS](#)」の列と同じです。

## DBA\_REPGROUP

DBA\_REPGROUP ビューには、レプリケートの対象となる、データベース内のオブジェクト・グループがすべて示されます。各オブジェクト・グループのメンバーは、DBA\_REPOBJECT ビューにリストされます。DBA\_REPGROUP ビューの列は、9-19 ページの「[ALL\\_REPGROUP](#)」の列と同じです。

## DBA\_REPGROUP\_PRIVILEGES

DBA\_REPGROUP\_PRIVILEGES ビューには、データベース内のすべてのオブジェクト・グループ内で権限を登録されているユーザーの情報が示されます。このビューの列は、9-20 ページの「[ALL\\_REPGROUP\\_PRIVILEGES](#)」の列と同じです。

## DBA\_REPGROUPED\_COLUMN

DBA\_REPGROUPED\_COLUMN ビューには、データベース内の各表の列グループを構成する、すべての列がリストされます。このビューの列は、9-21 ページの「[ALL\\_REPGROUPED\\_COLUMN](#)」の列と同じです。

## DBA\_REPKEY\_COLUMNS

DBA\_REPKEY\_COLUMNS ビューには、データベース内の各表の 1 つ以上の主キー列が示されます。このビューの列は、9-21 ページの「[ALL\\_REPKEY\\_COLUMNS](#)」の列と同じです。

## DBA\_REPOBJECT

DBA\_REPOBJECT ビューには、データベースの各レプリケート・オブジェクト・グループ内のオブジェクトの情報が示されます。オブジェクトは、1 つのオブジェクト・グループのみに属します。レプリケート・オブジェクト・グループは、複数のスキーマにまたがることができます。このビューの列は、9-22 ページの「[ALL\\_REPOBJECT](#)」の列と同じです。

## DBA\_REPPARAMETER\_COLUMN

DBA\_REPPARAMETER\_COLUMN ビューには、DBA\_REPRESOLUTION ビューに示される情報と、データベース内の各レプリケート表の競合解消に使用される列の情報が示されます。これらの列の値は、LIST\_OF\_COLUMN\_NAMES の引数として、DBMS\_REPCAT パッケージの `ADD_conflicttype_RESOLUTION` プロシージャに渡されます。DBA\_REPPARAMETER\_COLUMN ビューの列は、9-23 ページの「[ALL\\_REPPARAMETER\\_COLUMN](#)」の列と同じです。

## DBA\_REPPRIORITY

DBA\_REPPRIORITY ビューには、データベース内の各優先グループの各メンバーの値および優先順位レベルが示されます。優先グループには、レプリケート・オブジェクト・グループ内で一意の名前を付ける必要があります。また、指定された優先グループ内のそれぞれの優先順位レベルと値は、そのグループ内で一意であることが必要です。このビューの列は、9-24 ページの「[ALL\\_REPPRIORITY](#)」の列と同じです。

## DBA\_REPPRIORITY\_GROUP

DBA\_REPPRIORITY\_GROUP ビューには、データベース内の各レプリケート・オブジェクト・グループに定義された、優先グループまたはサイト優先順位グループが示されます。このビューの列は、9-26 ページの「[ALL\\_REPPRIORITY\\_GROUP](#)」の列と同じです。

## DBA\_REPPROP

DBA\_REPPROP ビューには、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。このビューに示されるのは、データベース内のすべてのオブジェクトです。操作の伝播は、ストアド・プロシージャやプロシージャ・ラップ・プログラムをコールしたり、表を直接操作することによって実行します。このビューの列は、9-26 ページの「[ALL\\_REPPROP](#)」の列と同じです。

## DBA\_REPRESOL\_STATS\_CONTROL

DBA\_REPRESOL\_STATS\_CONTROL ビューには、データベース内のすべてのレプリケート表に対する、競合解消の統計情報が示されます。このビューの列は、9-27 ページの「[ALL\\_REPRESOL\\_STATS\\_CONTROL](#)」の列と同じです。

## DBA\_REPRESOLUTION

DBA\_REPRESOLUTION ビューには、データベース内の各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。このビューの列は、9-28 ページの「[ALL\\_REPRESOLUTION](#)」の列と同じです。

## DBA\_REPRESOLUTION\_METHOD

DBA\_REPRESOLUTION\_METHOD ビューには、データベース内で使用可能な競合解消方法がすべてリストされます。まず、このビューには、アドバンスト・レプリケーション機能で提供する標準の方法がリストされます。新しいユーザー・ファンクションを作成して、データベース内のオブジェクト用の競合解消方法として追加すると、そのファンクションがビューに追加されます。このビューの列は、9-29 ページの「[ALL\\_REPRESOLUTION\\_METHOD](#)」の列と同じです。

## DBA\_REPRESOLUTION\_STATISTICS

DBA\_REPRESOLUTION\_STATISTICS ビューには、データベース内のすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。これらの統計情報は、DBMS\_REPCAT.REGISTER\_STATISTICS プロシージャをコールした場合に限り、表に対して収集されます。DBA\_REPRESOLUTION\_STATISTICS ビューの列は、9-29 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#)」の列と同じです。

DBA\_REPSITES

DBA\_REPSITES ビューには、データベース内の各レプリケート・オブジェクト・グループのメンバーがリストされます。

このビューには、ALL\_REPSITES ビューと USER\_REPSITES ビューには含まれていない次の列があります。

列	データ型	NULL	説明
PROP_UPDATES	NUMBER		伝播方法のコード化（マスターの場合）。
MY_DBLINK	VARCHAR2 (1)		インポート後、問題を検出するために使用されます。Y の場合、この dblink がグローバル名になります。

前述の列の他に、9-30 ページの「ALL\_REPSITES」と同じ列も表示されます。

USER\_REPCATLOG

各マスター・サイトの USER\_REPCATLOG ビューには、生成された非同期管理要求とエラー・メッセージの暫定的なステータスの情報が示されます。このビューに含まれる非同期管理要求およびエラー・メッセージは、カレント・ユーザーが所有しています。要求の処理中に生成されたすべてのエラー・メッセージは、その要求を作成したマスター・サイトの USER\_REPCATLOG ビューに転送されます。エラーが発生せずに管理要求が正常に完了した場合は、この要求に関するすべてのトレースは、USER\_REPCATLOG ビューから削除されます。このビューの列は、9-6 ページの「ALL\_REPCATLOG」の列と同じです。

USER\_REPCAT\_REFRESH\_TEMPLATES

このビューには、カレント・ユーザーが所有する各配置テンプレートに関するグローバル情報が含まれます。テンプレート名、テンプレートの所有者、テンプレート・オブジェクトが属するリフレッシュ・グループ、テンプレートの種類（プライベートまたはパブリック）などの情報があります。

このビューの列は、ALL\_REPCAT\_REFRESH\_TEMPLATES の列と同じです。このビューおよび列の詳細は、9-7 ページの「ALL\_REPCAT\_REFRESH\_TEMPLATES」を参照してください。

USER\_REPCAT\_TEMPLATE\_OBJECTS

USER\_REPCAT\_TEMPLATE\_OBJECTS ビューには、カレント・ユーザーが所有する各配置テンプレートの各オブジェクト定義が示されます。REFRESH\_TEMPLATE\_NAME にターゲット・テンプレートを指定すると、各オブジェクトがテンプレートに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_OBJECTS の列と同じです。このビューおよび列の詳細は、9-8 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_OBJECTS](#)」を参照してください。

## USER\_REPCAT\_TEMPLATE\_PARMS

カレント・ユーザーが所有するすべてのテンプレートのオブジェクト DDL で定義されたパラメータは、USER\_REPCAT\_TEMPLATE\_PARMS 表に格納されます。オブジェクトがテンプレートに追加されると、DDL に変数があるかどうかを検査され、パラメータが見つかるとう自動的にこのビューに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_PARMS の列と同じです。このビューおよび列の詳細は、9-10 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_PARMS](#)」を参照してください。

## USER\_REPCAT\_TEMPLATE\_SITES

USER\_REPCAT\_TEMPLATE\_SITES ビューは、エンタープライズ・ネットワークのサイトにおける、テンプレートのインスタンス化のカレント・ステータス情報をユーザーに提供します。カレント・ユーザーが所有している配置テンプレートのインスタンス化・サイトの情報も示されます。具体的には、ユーザーは RepAPI サイトを含む特定のサイトで実行されるテンプレートのインストールや削除を監視できます。このビューの列は、9-12 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_SITES](#)」の列と同じです。

## USER\_REPCAT\_USER\_AUTHORIZATIONS

USER\_REPCAT\_USER\_AUTHORIZATIONS ビューには、プライベート用に指定された、カレント・ユーザーが所有するすべてのテンプレートに対する権限を与えられたユーザーがリストされます。このビューにリストされたユーザーは、指定されたテンプレートをインスタンス化できます。このビューに表示されないユーザーはテンプレートをインスタンス化できません。このビューの列は、9-13 ページの「[ALL\\_REPCAT\\_USER\\_AUTHORIZATIONS](#)」の列と同じです。

## USER\_REPCAT\_USER\_PARM\_VALUES

USER\_REPCAT\_USER\_PARM\_VALUES ビューには、カレント・ユーザーが所有するすべての配置テンプレートのテンプレート・パラメータが示されます。データベース管理者は、テンプレートを配布してインスタンス化する前に、ユーザー・パラメータの表を作成するオプション機能があります。指定されたユーザーがテンプレートをインスタンス化する場合は、そのユーザー用に USER\_REPCAT\_USER\_PARM\_VALUES 表に格納された値が自動的に使用されます。

このビューの列は、ALL\_REPCAT\_USER\_PARM\_VALUES の列と同じです。このビューおよび列の詳細は、9-14 ページの「[ALL\\_REPCAT\\_USER\\_PARM\\_VALUES](#)」を参照してください。

## USER\_REPCOLUMN

USER\_REPCOLUMN ビューには、カレント・ユーザーが所有するすべての表のレプリケートされた列がリストされます。このビューの列は、9-16 ページの「[ALL\\_REPCOLUMN](#)」の列と同じです。

## USER\_REPCOLUMN\_GROUP

USER\_REPCOLUMN\_GROUP ビューには、カレント・ユーザーが所有する各レプリケート表の列グループがリストされます。このビューの列は、9-17 ページの「[ALL\\_REPCOLUMN\\_GROUP](#)」の列と同じです。

---

---

**注意：** USER\_REPCOLUMN\_GROUP ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPCOLUMN\_GROUP ビューおよび DBA\_REPCOLUMN\_GROUP ビューで使用可能です。

---

---

## USER\_REPCONFLICT

USER\_REPCONFLICT ビューには、競合解消方法が定義されている、カレント・ユーザーが所有する各表の名前と、その方法で解消される競合のタイプが表示されます。このビューの列は、9-17 ページの「[ALL\\_REPCONFLICT](#)」の列と同じです。

---

---

**注意：** USER\_REPCONFLICT ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPCONFLICT ビューおよび DBA\_REPCONFLICT ビューで使用可能です。

---

---

## USER\_REPDDL

USER\_REPDDL ビューには、カレント・ユーザーが所有する各レプリケーション・オブジェクトの DDL が示されます。このビューの列は、9-18 ページの「[ALL\\_REPDDL](#)」の列と同じです。

## USER\_REPGENOBJECTS

USER\_REPGENOBJECTS ビューには、レプリケーションのサポート用に生成された、カレント・ユーザーが所有する各オブジェクトが示されます。このビューの列は、9-18 ページの「[ALL\\_REPGENOBJECTS](#)」の列と同じです。

## USER\_REPGROUP

USER\_REPGROUP ビューには、レプリケートの対象となる、カレント・ユーザーが所有するオブジェクト・グループがすべて示されます。各オブジェクト・グループのメンバーは、

USER\_REPOBJECT ビューにリストされます。USER\_REPGROUP ビューの列は、9-19 ページの「[ALL\\_REPGROUP](#)」の列と同じです。

## USER\_REPGROUP\_PRIVILEGES

USER\_REPGROUP\_PRIVILEGES ビューには、カレント・ユーザーが所有するオブジェクト・グループ内で権限を登録されているユーザーの情報が示されます。このビューの列は、9-20 ページの「[ALL\\_REPGROUP\\_PRIVILEGES](#)」の列と同じです。

## USER\_REPGROUPED\_COLUMN

USER\_REPGROUPED\_COLUMN ビューには、各表の列グループを構成するすべての列がリストされます。このビューの列は、9-21 ページの「[ALL\\_REPGROUPED\\_COLUMN](#)」の列と同じです。

---

---

**注意：** USER\_REPGROUPED\_COLUMN ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPGROUPED\_COLUMN ビューおよび DBA\_REPGROUPED\_COLUMN ビューで使用可能です。

---

---

## USER\_REPKEY\_COLUMNS

USER\_REPKEY\_COLUMNS ビューには、カレント・ユーザーが所有する各表の 1 つ以上の主キー列が示されます。このビューの列は、9-21 ページの「[ALL\\_REPKEY\\_COLUMNS](#)」の列と同じです。

## USER\_REPOBJECT

USER\_REPOBJECT ビューには、カレント・ユーザーが所有する各レプリケート・オブジェクト・グループ内のオブジェクトの情報が示されます。オブジェクトは、1 つのオブジェクト・グループのみに属します。レプリケート・オブジェクト・グループは、複数のスキーマにまたがることができます。このビューの列は、9-22 ページの「[ALL\\_REPOBJECT](#)」の列と同じです。

## USER\_REPPARAMETER\_COLUMN

USER\_REPPARAMETER\_COLUMN ビューには、USER\_REPRESOLUTION ビューに示される情報と、カレント・ユーザーが所有する各レプリケート表の競合解消に使用される列の情報が示されます。これらの列の値は、LIST\_OF\_COLUMN\_NAMES の引数として、DBMS\_REPCAT パッケージの ADD\_conflicttype\_RESOLUTION プロシージャに渡されます。USER\_REPPARAMETER\_COLUMN ビューの列は、9-23 ページの「[ALL\\_REPPARAMETER\\_COLUMN](#)」の列と同じです。

---

---

**注意：** USER\_REPPARAMETER\_COLUMN ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPPARAMETER\_COLUMN ビューおよび DBA\_REPPARAMETER\_COLUMN ビューで使用可能です。

---

---

## USER\_REPPRIORITY

USER\_REPPRIORITY ビューには、カレント・ユーザーが所有する各優先グループの各メンバーの値および優先順位レベルが示されます。優先グループには、レプリケート・オブジェクト・グループ内で一意の名前を付ける必要があります。また、指定された優先グループ内のそれぞれの優先順位レベルと値は、そのグループ内で一意であることが必要です。このビューの列は、9-24 ページの「[ALL\\_REPPRIORITY](#)」の列と同じです。

---

---

**注意：** USER\_REPPRIORITY ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPPRIORITY ビューおよび DBA\_REPPRIORITY ビューで使用可能です。

---

---

## USER\_REPPRIORITY\_GROUP

USER\_REPPRIORITY\_GROUP ビューには、カレント・ユーザーが所有する各レプリケート・オブジェクト・グループに定義された、優先グループまたはサイト優先順位グループが示されます。このビューの列は、9-26 ページの「[ALL\\_REPPRIORITY\\_GROUP](#)」の列と同じです。

## USER\_REPPROP

USER\_REPPROP ビューには、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。このビューに示されるのは、カレント・ユーザーが所有するオブジェクトです。操作の伝播は、ストアド・プロシージャやプロシージャ・ラップ・プログラムをコールしたり、表を直接操作して実行します。このビューの列は、9-26 ページの「[ALL\\_REPPROP](#)」の列と同じです。

## USER\_REPRESOL\_STATS\_CONTROL

USER\_REPRESOL\_STATS\_CONTROL ビューには、カレント・ユーザーが所有するすべてのレプリケート表に対する、競合解消の統計情報が示されます。このビューの列は、9-27 ページの「[ALL\\_REPRESOL\\_STATS\\_CONTROL](#)」の列と同じです。

---

---

**注意：** USER\_REPRESOL\_STATS\_CONTROL ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPRESOL\_STATS\_CONTROL ビューおよび DBA\_REPRESOL\_STATS\_CONTROL ビューで使用可能です。

---

---



## USER\_REPRESOLUTION

USER\_REPRESOLUTION ビューには、カレント・ユーザーが所有する各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。このビューの列は、9-28 ページの「[ALL\\_REPRESOLUTION](#)」の列と同じです。

---

---

**注意：** USER\_REPREPRESOLUTION ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPREPRESOLUTION ビューおよび DBA\_REPREPRESOLUTION ビューで使用可能です。

---

---

## USER\_REPRESOLUTION\_METHOD

USER\_REPRESOLUTION\_METHOD ビューには、データベース内で使用可能な競合解消方法がすべてリストされます。まず、このビューには、アドバンスト・レプリケーション機能で提供する標準の方法がリストされます。新しいユーザー・ファンクションを作成してデータベース内のオブジェクト用の競合解消方法として追加すると、そのファンクションがビューに追加されます。このビューの列は、9-29 ページの「[ALL\\_REPRESOLUTION\\_METHOD](#)」の列と同じです。

## USER\_REPRESOLUTION\_STATISTICS

USER\_REPRESOLUTION\_STATISTICS ビューには、カレント・ユーザーが所有するすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。これらの統計情報は、DBMS\_REPCAT.REGISTER\_STATISTICS プロシージャをコールした場合に限り、表に対して収集されます。USER\_REPRESOLUTION\_STATISTICS ビューの列は、9-29 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#)」の列と同じです。

---

---

**注意：** USER\_REPRESOLUTION\_STATISTICS ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPRESOLUTION\_STATISTICS ビューおよび DBA\_REPRESOLUTION\_STATISTICS ビューで使用可能です。

---

---

## USER\_REPSITES

USER\_REPSITES ビューには、カレント・ユーザーが所有する各レプリケート・オブジェクト・グループのメンバーがリストされます。このビューの列は、9-30 ページの「[ALL\\_REPSITES](#)」の列と同じです。

## 遅延トランザクション・ビュー

Oracle には、遅延トランザクションを管理するためのビューがいくつか用意されています。これらのビューは、トランザクションの接続先、トランザクションを構成する遅延コール、トランザクションの処理中に発生するエラーなど、遅延トランザクションに関する情報を提供します。

- [DEFCALL](#)
- [DEFDEFAULTDEST](#)
- [DEFERRCOUNT](#)
- [DEFCALLDEST](#)
- [DEFERROR](#)
- [DEFLOB](#)
- [DEFPROPAGATOR](#)
- [DEFSCHEDULE](#)
- [DEFTRAN](#)
- [DEFTRANDEST](#)

---

---

**注意：** 表を直接変更しないでください。変更が必要な場合は、DBMS\_DEFER および DBMS\_DEFER\_SYS パッケージに用意されているプロシージャを使用してください。

---

---

## DEFCALL

すべての遅延リモート・プロシージャ・コールが記録されます。

列	データ型	NULL	説明
CALLNO	NUMBER		トランザクション処理中のコールの一意な ID。
DEFERRED_TRAN_ID	VARCHAR2 (30)		対応付けられたトランザクションの一意な ID。
SCHEMANAME	VARCHAR2 (30)		遅延コールのスキーマ名。
PACKAGENAME	VARCHAR2 (30)		遅延コールのパッケージ名。レプリケート表では、表名を示す場合があります。
PROCNAME	VARCHAR2 (30)		遅延コールのプロシージャ名。レプリケート表では、操作名を示す場合があります。
ARGCOUNT	NUMBER		遅延コールの引数の数。

## DEFCALLDEST

各遅延リモート・プロシージャ・コールの接続先がリストされます。

列	データ型	NULL	説明
CALLNO	NUMBER	NOT NULL	トランザクション処理中のコールの一意な ID。
DEFERRED_TRAN_ID	VARCHAR2 (30)	NOT NULL	DEFTRAN ビューの DEFERRED_TRAN_ID 列に対応しています。各遅延トランザクションには、1 つ以上の遅延コールが含まれます。
DBLINK	VARCHAR2 (128)	NOT NULL	接続先データベースの完全修飾名。

## DEFDEFAULTDEST

Oracle レプリケーションを使用せず、遅延トランザクションの接続先や、トランザクション処理中のコールを指定しない場合、Oracle は DEFDEFAULTDEST ビューを使用して、リモート・プロシージャ・コールを遅らせるための接続先データベースを判別します。

列	データ型	NULL	説明
DBLINK	VARCHAR2 (128)	NOT NULL	トランザクションのレプリケート先の、完全修飾されたデータベース名。

DEFERRCOUNT

接続先に対するエラー・トランザクションの情報が含まれます。

列	データ型	NULL	説明
ERRCOUNT	NUMBER		接続先でエラー原因となった既存のトランザクションの数。
DESTINATION	VARCHAR2 (128)		接続先を指定するためのデータベース・リンク。

DEFERROR

適用できなかった各トランザクションの ID が含まれます。この ID を使用して、このトランザクションに対応付けられたキューに入れられたコールを指定できます。これらのコールは、DEFCALL ビューに格納されています。DBMS\_DEFER\_QUERY パッケージのプロシージャを使用して、DEFCALL ビューにリストされているプロシージャに対する引数を指定できます。

列	データ型	NULL	説明
DEFERRED_TRAN_ID	VARCHAR2 (22)	NOT NULL	エラー原因となっているトランザクションの ID。
ORIGIN_TRAN_DB	VARCHAR2 (128)		遅延トランザクションを作成したデータベース。
ORIGIN_TRAN_ID	VARCHAR2 (22)		トランザクションの元の ID。
CALLNO	NUMBER		DEFERRED_TRAN_ID のコールの一意な ID。
DESTINATION	VARCHAR2 (128)		接続先を指定するためのデータベース・リンク。
START_TIME	DATE		元のトランザクションがエンキューされた時刻。
ERROR_NUMBER	NUMBER		Oracle エラー番号。
ERROR_MSG	VARCHAR2 (2000)		エラー・メッセージのテキスト。
RECEIVER	VARCHAR2 (30)		遅延トランザクションの元の受信者。

DEFLOB

遅延 RPC の LOB パラメータが含まれます。

列	データ型	NULL	説明
ID	RAW (16)	NOT NULL	LOB パラメータの識別子。
DEFERRED_TRAN_ID	VARCHAR2 (22)		この LOB パラメータを使用した遅延 RPC のトランザクション ID。

列	データ型	NULL	説明
BLOB_COL	BLOB (4000)		LOB パラメータ (バイナリ)。
CLOB_COL	CLOB (4000)		LOB パラメータ (キャラクタ)。
NCLOB_COL	CLOB (4000)		LOB パラメータ (各国文字キャラクタ)。

## DEFPROPAGATOR

ローカルなプロパゲータの情報が含まれます。

列	データ型	NULL	説明
USERNAME	VARCHAR2 (30)	NOT NULL	プロパゲータのユーザー名。
USERID	NUMBER	NOT NULL	プロパゲータのユーザー ID。
STATUS	VARCHAR2 (7)		プロパゲータのステータス。
CREATED	DATE	NOT NULL	プロパゲータが登録された時刻。

## DEFSCHEDULE

次にスケジュールされているジョブの実行時刻に関する情報が含まれます。

列	データ型	NULL	説明
DBLINK	VARCHAR2 (128)	NOT NULL	遅延リモート・プロシージャ・コールを定期的に行うマスター・データベース・サイトへの完全修飾パス名。
JOB	NUMBER		DBMS_DEFER_SYS.SCHEDULE_PUSH をコールして作成したジョブに割り当てられる番号。ジョブの実行内容を判別するには、USER_JOBS ビューの WHAT 列に問い合わせます。
INTERVAL	VARCHAR2 (200)		次回遅延トランザクション・キューをプッシュする時刻を計算するためのファンクション。
NEXT_DATE	DATE		次のジョブ実行日。
LAST_DATE	DATE		前回キューがプッシュされた日付、または前回この送信先に対してリモート・プロシージャ・コールのプッシュを試みた日付。
DISABLED	CHAR (1)		Y の場合、接続先への伝播は無効です。 N の場合、接続先への伝播は有効です。
LAST_TXN_COUNT	NUMBER		前回の実行でプッシュしたトランザクションの数。

列	データ型	NULL	説明
LAST_ERROR_NUMBER	NUMBER		前回のプッシュでの Oracle エラー番号。
LAST_ERROR_MESSAGE	VARCHAR2 (2000)		前回のプッシュでのエラー・メッセージ。

DEFTRAN

すべての遅延トランザクションが記録されます。

列	データ型	NULL	説明
DEFERRED_TRAN_ID	VARCHAR2 (30)		コールをエンキューしたトランザクション ID。
DELIVERY_ORDER	NUMBER		キュー内の遅延トランザクションの順序を決定する識別子。識別子は、元のトランザクションのシステム・コミット番号から派生しています。
DESTINATION_LIST	VARCHAR2 (1)		R の場合、接続先は ALL_REPSITES ビューによって決定されます。  D の場合、接続先は DEFDEFAULTDEST ビュー、あるいは TRANSACTION プロシージャまたは CALL プロシージャの NODE_LIST 引数によって決定されます。
START_TIME	DATE		元のトランザクションがエンキューされた時刻。

DEFTRANDEST

遅延トランザクションの接続先がリストされます。

列	データ型	NULL	説明
DEFERRED_TRAN_ID	VARCHAR2 (30)	NOT NULL	指定されたデータベース・リンクにレプリケートするトランザクションの ID。
DELIVERY_ORDER	NUMBER		キュー内の遅延トランザクションの順序を決定する識別子。識別子は、元のトランザクションのシステム・コミット番号から派生しています。
DBLINK	VARCHAR2 (128)	NOT NULL	接続先データベースの完全修飾名。

## スナップショット・ビューおよびスナップショット・リフレッシュ・グループ・ビュー

次のビューでは、スナップショットおよびスナップショット・リフレッシュ・グループの情報が提供されます。

表 9-2 スナップショット・ビューおよびスナップショット・リフレッシュ・グループ・ビュー

ALL_ ビュー	DBA_ ビュー	USER_ ビュー
ALL_REFRESH	DBA_REFRESH	USER_REFRESH
ALL_REFRESH_CHILDREN	DBA_REFRESH_CHILDREN	USER_REFRESH_CHILDREN
	DBA_REGISTERED_SNAPSHOT_GROUPS	
ALL_REGISTERED_SNAPSHOTS	DBA_REGISTERED_SNAPSHOTS	USER_REGISTERED_SNAPSHOTS
ALL_SNAPSHOT_LOGS	DBA_SNAPSHOT_LOGS	USER_SNAPSHOT_LOGS
	DBA_SNAPSHOT_LOG_FILTER_COLS	
ALL_SNAPSHOT_REFRESH_TIMES	DBA_SNAPSHOT_REFRESH_TIMES	USER_SNAPSHOT_REFRESH_TIMES
ALL_SNAPSHOTS	DBA_SNAPSHOTS	USER_SNAPSHOTS

## ALL\_REFRESH

カレント・ユーザーからアクセス可能なすべてのリフレッシュ・グループが示されます。

### 関連するビュー：

- DBA\_REFRESH には、データベース内のすべてのリフレッシュ・グループが示され  
ます。
- USER\_REFRESH には、カレント・ユーザーが所有するすべてのリフレッシュ・グルー  
プが示されます。

列	データ型	NULL	説明
ROWNER	VARCHAR2 (30)	NOT NULL	リフレッシュ・グループの所有者の名前。
RNAME	VARCHAR2 (30)	NOT NULL	リフレッシュ・グループの名前。
REFGROUP	NUMBER		リフレッシュ・グループの内部識別子。
IMPLICIT_DESTROY	VARCHAR2 (1)		Y または N を指定します。Y の場合、最後の項目が除外されるとリフレッシュ・グループが破棄されます。
PUSH_DEFERRED_RPC	VARCHAR2 (1)		Y または N を指定します。Y の場合、リフレッシュ前にスナップショットからマスターに変更がプッシュされます。
REFRESH_AFTER_ERRORS	VARCHAR2 (1)		Y の場合、遅延 RPC のプッシュ時にエラーが発生してもリフレッシュが続行されます。
ROLLBACK_SEG	VARCHAR2 (30)		リフレッシュ時に使用するロールバック・セグメントの名前。
JOB	NUMBER		グループの自動リフレッシュに使用されるジョブの識別子。
NEXT_DATE	DATE		ジョブが中断されていない場合の、次の自動リフレッシュ予定日。
INTERVAL	VARCHAR2 (200)		次の NEXT_DATE を計算するための日付関数。
BROKEN	VARCHAR2 (1)		Y または N を指定します。Y の場合、ジョブが中断されており、それ以降は実行されません。
PURGE_OPTION	NUMBER (38)		プッシュ後にトランザクション・キューをバージする方法。1 の場合は高速バージ、2 の場合は精密バージが実行されます。
PARALLELISM	NUMBER (38)		トランザクション伝播時の並行性レベル。
HEAP_SIZE	NUMBER (38)		ヒープのサイズ。



## ALL\_REFRESH\_CHILDREN

カレント・ユーザーからアクセス可能なリフレッシュ・グループのすべてのオブジェクトがリストされます。

### 関連するビュー：

- DBA\_REFRESH\_CHILDREN には、データベース内のすべてのリフレッシュ・グループのオブジェクトが示されます。
- USER\_REFRESH\_CHILDREN には、カレント・ユーザーが所有するすべてのリフレッシュ・グループのオブジェクトが示されます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	リフレッシュ・グループ内のオブジェクトの所有者。
NAME	VARCHAR2 (30)	NOT NULL	リフレッシュ・グループ内のオブジェクトの名前。
TYPE	VARCHAR2 (30)		リフレッシュ・グループ内のオブジェクトのタイプ。
ROWNER	VARCHAR2 (30)	NOT NULL	リフレッシュ・グループの所有者の名前。
RNAME	VARCHAR2 (30)	NOT NULL	リフレッシュ・グループの名前。
REFGROUP	NUMBER		リフレッシュ・グループの内部識別子。
IMPLICIT_DESTROY	VARCHAR2 (1)		Y または N を指定します。Y の場合、最後の項目が除外されるとリフレッシュ・グループが破棄されます。
PUSH_DEFERRED_RPC	VARCHAR2 (1)		Y または N を指定します。Y の場合、リフレッシュ前にスナップショットからマスターに変更がプッシュされます。
REFRESH_AFTER_ERRORS	VARCHAR2 (1)		Y の場合、遅延 RPC のプッシュ時にエラーが発生してもリフレッシュが続行されます。
ROLLBACK_SEG	VARCHAR2 (30)		リフレッシュ時に使用するロールバック・セグメントの名前。
JOB	NUMBER		グループの自動リフレッシュに使用されるジョブの識別子。
NEXT_DATE	DATE		ジョブが中断されていない場合の、次回の自動リフレッシュ予定日。
INTERVAL	VARCHAR2 (200)		次回の NEXT_DATE を計算するための日付関数。
BROKEN	VARCHAR2 (1)		Y または N を指定します。Y の場合、ジョブが中断されており、それ以降は実行されません。
PURGE_OPTION	NUMBER (38)		プッシュ後にトランザクション・キューをパージする方法。1 の場合は高速パージ、2 の場合は精密パージが実行されます。
PARALLELISM	NUMBER (38)		トランザクション伝播時の並行性レベル。
HEAP_SIZE	NUMBER (38)		ヒープのサイズ。

ALL\_REGISTERED\_SNAPSHOTS

カレント・ユーザーからアクセス可能な、登録済スナップショットがすべて示されます。

関連するビュー：

- DBA\_REGISTERED\_SNAPSHOTS には、データベース内の登録済スナップショットがすべて示されます。
- USER\_REGISTERED\_SNAPSHOTS には、カレント・ユーザーが所有する登録済スナップショットがすべて示されます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	スナップショットの所有者。
NAME	VARCHAR2 (30)	NOT NULL	スナップショットの名前。
SNAPSHOT_SITE	VARCHAR2 (128)	NOT NULL	スナップショット・サイトのグローバル名。
CAN_USE_LOG	VARCHAR2 (3)		スナップショット・ログを使用できる場合は YES、スナップショットが複雑すぎるため、ログが使用できない場合は NO になります。
UPDATABLE	VARCHAR2 (3)		スナップショットが更新可能かどうかを示します。更新可能な場合は YES、不可能な場合は NO になります。NO に設定された場合、スナップショットは読取り専用です。
REFRESH_METHOD	VARCHAR2 (11)		スナップショットの高速リフレッシュに ROWID と主キーのどちらを使用するかを示します。
SNAPSHOT_ID	NUMBER (38)		高速リフレッシュの実行時にマスターで使用されるスナップショットの識別子。
VERSION	VARCHAR2 (17)		スナップショットのバージョン。
QUERY_TXT	LONG		このスナップショットのインスタンスエーションで基になる問合せ。

ALL\_SNAPSHOT\_LOGS

カレント・ユーザーからアクセス可能な、すべてのスナップショット・ログがリストされます。

このビューは、スナップショット・ログを使用して、1つのスナップショットにつき1つの行を示しています。ただし、マスター・サイトにあるすべてのスナップショットに関するログは1つです。使用されているログを検索するには、SNAPSHOT\_ID と CURRENT\_SNAPSHOTS を選択せずに一意に、USER\_SNAPSHOT\_LOGS を問い合わせます。

**関連するビュー：**

- DBA\_SNAPSHOT\_LOGS には、データベース内のすべてのスナップショット・ログが示されます。
- USER\_SNAPSHOT\_LOGS には、カレント・ユーザーが所有するすべてのスナップショット・ログが示されます。

列	データ型	NULL	説明
LOG_OWNER	VARCHAR2 (30)	NOT NULL	ログの所有者。
MASTER	VARCHAR2 (30)	NOT NULL	変更が記録されるマスター表の名前。
LOG_TABLE	VARCHAR2 (30)	NOT NULL	マスター表に加えられた変更が記録される表の名前。
LOG_TRIGGER	VARCHAR2 (30)		Oracle8i Server では廃止されています。NULL に設定されます。このパラメータは、ログに行を挿入する、マスターの行の終わりのトリガーを示していました。
ROWIDS	VARCHAR2 (3)		YES の場合、ROWID 情報が記録されます。
PRIMARY_KEY	VARCHAR2 (3)		YES の場合、主キー情報が記録されます。
FILTER_COLUMNS	VARCHAR2 (3)		YES の場合、スナップショット・ログにフィルタ列が記録されます。
CURRENT_SNAPSHOTS	DATE		マスターのスナップショットが前回リフレッシュされた日付。1つのスナップショットに対して1つの日付が表示されます。
SNAPSHOT_ID	NUMBER (38)		スナップショットの一意識別子。

**ALL\_SNAPSHOT\_REFRESH\_TIMES**

カレント所有者からアクセス可能なスナップショットの、リフレッシュ時刻が示されます。

**関連するビュー：**

- DBA\_SNAPSHOT\_REFRESH\_TIMES には、データベース内のすべてのスナップショットのリフレッシュ時刻が示されます。
- USER\_SNAPSHOT\_REFRESH\_TIMES には、カレント・ユーザーが所有するすべてのスナップショットの、リフレッシュ時刻が示されます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	スナップショットの所有者。
NAME	VARCHAR2 (30)	NOT NULL	スナップショット・ビューの名前。
MASTER_OWNER	VARCHAR2 (30)		マスター表の所有者。

列	データ型	NULL	説明
MASTER	VARCHAR2 (30)		マスター表の名前。
LAST_REFRESH	DATE		前回のリフレッシュ。

ALL\_SNAPSHOTS

ユーザーからアクセス可能なすべてのスナップショットが示されます。

関連するビュー：

- DBA\_SNAPSHOTS には、データベース内のすべてのスナップショットが示されます。
- USER\_SNAPSHOTS には、カレント・ユーザーが所有するスナップショットがすべて示されます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	スナップショットの所有者。
NAME	VARCHAR2 (30)	NOT NULL	スナップショットを表示するために、ユーザーやアプリケーションによって使用されるビューの名前。  COMPATIBLE パラメータが 8.1.0 以上に設定されているデータベースの場合、ビューの名前は TABLE_NAME と同じです。COMPATIBLE パラメータが 8.1.0 未満に設定されているデータベースの場合、ビューの名前と TABLE_NAME は異なります。
TABLE_NAME	VARCHAR2 (30)	NOT NULL	スナップショットが格納されている表の名前。
MASTER_VIEW	VARCHAR2 (30)		スナップショットの所有者が所有する、リフレッシュ用のマスター表のビュー。Oracle8i では廃止されており、NULL に設定されています。
MASTER_OWNER	VARCHAR2 (30)		マスター表の所有者。
MASTER	VARCHAR2 (30)		スナップショットのコピー元であるマスター表の名前。
MASTER_LINK	VARCHAR2 (128)		マスター・サイトへのデータベース・リンク名。
CAN_USE_LOG	VARCHAR2 (3)		スナップショット・ログを使用できる場合は YES、スナップショットが複雑すぎるため、ログが使用できない場合は NO になります。
UPDATABLE	VARCHAR2 (3)		スナップショットが更新可能かどうかを示します。更新可能な場合は YES、不可能な場合は NO になります。NO に設定された場合、スナップショットは読取り専用です。

列	データ型	NULL	説明
REFRESH_METHOD	VARCHAR2 (11)		スナップショットのリフレッシュの実行に使用する値 (PRIMARY KEY/ROWID/COMPLEX)。PRIMARY KEY の場合、主キーを使用した高速リフレッシュが実行されます。ROWID の場合、ROWID を使用した高速リフレッシュが実行されます。COMPLEX の場合、高速リフレッシュは実行できず、完全リフレッシュのみを実行できます。
LAST_REFRESH	DATE		マスター・サイトでの前回のリフレッシュの日付と時刻。
ERROR	NUMBER		前回の正常なリフレッシュ実行以降、自動リフレッシュに失敗した回数。
FR_OPERATIONS	VARCHAR2 (10)		生成された高速リフレッシュ操作のステータス (REGENERATE または VALID)。
CR_OPERATIONS	VARCHAR2 (10)		生成された完全リフレッシュ操作のステータス (REGENERATE または VALID)。
TYPE	VARCHAR2 (8)		自動リフレッシュのタイプ。リフレッシュのタイプは、完全リフレッシュ、高速リフレッシュ、強制リフレッシュの3つです。
NEXT	VARCHAR2 (200)		次のリフレッシュ日を計算するための日付関数。
START_WITH	DATE		次のリフレッシュ日を計算するための日付関数。
REFRESH_GROUP	NUMBER		同じトランザクションでリフレッシュされる、特定のリフレッシュ・グループ内のすべてのスナップショットの数。
UPDATE_TRIG	VARCHAR2 (30)		この列は廃止されています。Oracle8i スナップショットの場合、NULL になります。この列は、UPDATE_LOG にデータを入れるトリガーの名前を示していました。
UPDATE_LOG	VARCHAR2 (30)		更新可能なスナップショットに加えられた変更を記録する表。
QUERY	LONG		このスナップショットのインスタンスエーションの基となる問合せ。
MASTER_ROLLBACK_SEG	VARCHAR2 (30)		マスター・サイトで使用するロールバック・セグメント。
STATUS	VARCHAR2 (7)		スナップショットの内容のステータス。
REFRESH_MODE	VARCHAR2 (8)		スナップショットのリフレッシュの方法およびタイミング。
PREBUILT	VARCHAR2 (3)		YES の場合、このスナップショットでは事前作成された表がベース表として使用されています。

DBA\_REFRESH

DBA\_REFRESH ビューには、データベース内のすべてのリフレッシュ・グループが示されます。このビューの列は、9-48 ページの「ALL\_REFRESH」の列と同じです。

DBA\_REFRESH\_CHILDREN

DBA\_REFRESH\_CHILDREN には、データベース内のすべてのリフレッシュ・グループのオブジェクトがすべてリストされます。このビューの列は、9-49 ページの「ALL\_REFRESH\_CHILDREN」の列と同じです。

DBA\_REGISTERED\_SNAPSHOT\_GROUPS

DBA\_REGISTERED\_SNAPSHOT\_GROUPS には、そのサイトのスナップショット・グループがすべてリストされます。このビューは、Oracle レプリケーション・パッケージをインストールしている場合に限り、使用可能です。

列	データ型	NULL	説明
NAME	VARCHAR2 (30)		スナップショット・レプリケーション・グループの名前。
SNAPSHOT_SITE	VARCHAR2 (128)		スナップショット・レプリケーション・グループのマスターのサイト。
GROUP_COMMENT	VARCHAR2 (80)		スナップショット・レプリケーション・グループの説明。
VERSION	VARCHAR2 (8)		スナップショット・レプリケーション・グループのバージョン。
FNAME	VARCHAR2 (30)		スナップショット・オブジェクト・グループの固有の名前。

DBA\_REGISTERED\_SNAPSHOTS

DBA\_REGISTERED\_SNAPSHOTS には、データベース内の登録済スナップショットがすべて示されます。このビューの列は、9-48 ページの「ALL\_REFRESH」の列と同じです。

DBA\_SNAPSHOT\_LOGS

DBA\_SNAPSHOT\_LOGS には、データベース内のすべてのスナップショット・ログが示されます。このビューの列は、9-50 ページの「ALL\_SNAPSHOT\_LOGS」の列と同じです。

DBA\_SNAPSHOT\_LOG\_FILTER\_COLS

DBA\_SNAPSHOT\_LOG\_FILTER\_COLS には、スナップショット・ログに記録される、主キー列以外のすべてのフィルタ列がリストされます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ログが記録されるマスター表の所有者。
NAME	VARCHAR2 (30)	NOT NULL	ログが記録されるマスター表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	ログが記録されるフィルタ列。

## DBA\_SNAPSHOT\_REFRESH\_TIMES

DBA\_SNAPSHOT\_REFRESH\_TIMES には、データベース内のすべてのスナップショットのリフレッシュ時刻がリストされます。このビューの列は、9-51 ページの「[ALL\\_SNAPSHOT\\_REFRESH\\_TIMES](#)」の列と同じです。

## DBA\_SNAPSHOTS

DBA\_SNAPSHOTS には、データベース内のすべてのスナップショットが示されます。このビューの列は、9-52 ページの「[ALL\\_SNAPSHOTS](#)」の列と同じです。

## USER\_REFRESH

USER\_REFRESH には、カレント・ユーザーが所有するすべてのリフレッシュ・グループが示されます。このビューの列は、9-48 ページの「[ALL\\_REFRESH](#)」の列と同じです。

## USER\_REFRESH\_CHILDREN

USER\_REFRESH\_CHILDREN には、カレント・ユーザーが所有するリフレッシュ・グループ内のすべてのオブジェクトがリストされます。このビューの列は、9-49 ページの「[ALL\\_REFRESH\\_CHILDREN](#)」の列と同じです。

## USER\_REGISTERED\_SNAPSHOTS

USER\_REGISTERED\_SNAPSHOTS には、カレント・ユーザーが所有するすべての登録済スナップショットが示されます。このビューの列は、9-48 ページの「[ALL\\_REFRESH](#)」の列と同じです。

## USER\_SNAPSHOTS

USER\_SNAPSHOTS には、カレント・ユーザーが所有するすべてのスナップショットが示されます。このビューの列は、9-52 ページの「[ALL\\_SNAPSHOTS](#)」の列と同じです。

## USER\_SNAPSHOT\_LOGS

USER\_SNAPSHOT\_LOGS には、カレント・ユーザーが所有するすべてのスナップショット・ログがリストされます。このビューの列は、9-50 ページの「[ALL\\_SNAPSHOT\\_LOGS](#)」の列と同じです。

## USER\_SNAPSHOT\_REFRESH\_TIMES

USER\_SNAPSHOT\_REFRESH\_TIMES には、カレント・ユーザーが所有するスナップショットのリフレッシュ時刻が示されます。このビューの列は、9-51 ページの「[ALL\\_SNAPSHOT\\_REFRESH\\_TIMES](#)」の列と同じです。



---

# セキュリティ・オプション

セキュリティ・オプションには次の2種類があります。

- マルチマスター・レプリケーションのセキュリティ設定
- スナップショット・レプリケーションのセキュリティ設定

## マルチマスター・レプリケーションのセキュリティ設定

ほとんどの場合、マルチマスター・レプリケーションのセキュリティを設定するときは、Replication Manager の「セットアップ・ウィザード」を使用するのが最も簡単です。ただし、レプリケーション・マネージメント API を使用してこれらの設定を行うことが必要な場合もあります。

レプリケーション環境を構成するには、データベース管理者が DBA 権限で接続して、必要な権限をレプリケーション管理者に付与する必要があります。

最初に、各マスター・サイトで、レプリケーション環境を構成およびメンテナンスする権限と、レプリケートされた変更を伝播および適用する権限を持つユーザー・アカウントを設定します。また、各マスター・サイトでユーザー用のリンクを定義する必要があります。

レプリケーション環境の "ユーザー" のカテゴリとして、レプリケート・オブジェクトにアクセスするエンド・ユーザーに加え、次の 3 つの特別なカテゴリがあります。


- レプリケーション環境の構成およびメンテナンスを行う、レプリケーション管理者
- 遅延トランザクションを伝播する、プロパゲータ
- これらのトランザクションを適用する、リモート・サイトの受信者

通常は、1 人のユーザーが管理者、プロパゲータおよび受信者の役割を果たします。ただし、機能ごとに別々のユーザーを設定することも可能です。1 人のグローバル・レプリケーション管理者のみを設定できれば、レプリケーション・グループが複数のスキーマにまたがっていない場合は、異なるスキーマごとに別々のレプリケーション管理者を設定できます。ただし、各データベースに登録できるプロパゲータは 1 人です。

A-4 ページの表 A-1 に、これらの特別なアカウントに割り当てる必要がある権限を示します。これらのユーザーに必要な権限のほとんどは、レプリケーション・マネージメント API へのコールを通じて付与されます。また、CONNECT や RESOURCE など一部の権限は、直接付与する必要があります。

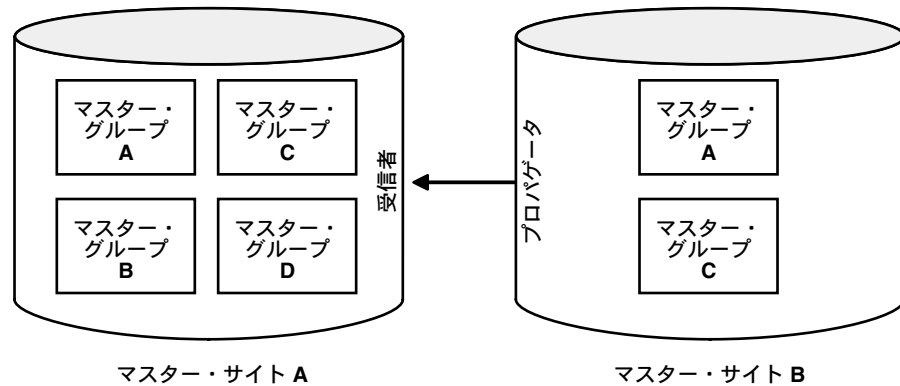
## トラステッド・セキュリティとアントラステッド・セキュリティ

ユーザーのカテゴリの他に、セキュリティ・モデルとしてトラステッドとアントラステッドのどちらをインプリメントするかを決定する必要があります。トラステッド・セキュリティ・モデルでは、受信者がすべてのローカル・マスター・グループにアクセスできます。受信者は、リモート・サイトのプロパゲータにかわってローカル・マスター・サイトでデータベース・アクティビティを実行するので、プロパゲータにも、受信者のサイトのすべてのマスター・グループへのアクセス権限があります。1 人の受信者をすべての受信トランザクションに使用することに注意してください。

たとえば、 A-1 の場合を考えます。マスター・サイト B にはマスター・グループ A と C のみが存在しますが、トラステッド・セキュリティ・モデルが使用されているので、プロパゲータは、マスター・サイト A のマスター・グループ A、B、C、D にアクセスできます。その結果、リモート・データベース管理の可動性によりデータベース管理の柔軟性が大幅に向上します。ただし、悪意のあるユーザーがリモート・サイトからマスター・サイトのデータを見たり、破損させたりする可能性も高くなります。

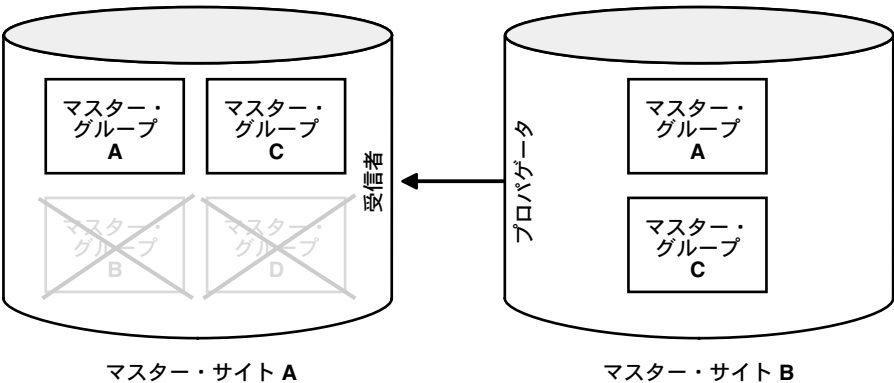
使用するセキュリティ・モデルに関係なく、Oracle8i では、オブジェクトがレプリケート環境に追加されるとき、またはレプリケート環境から削除されるときに、オブジェクトに適切な権限が自動的に付与されます。

図 A-1 トラストッド・セキュリティ：マルチマスター・レプリケーション



アントラストッド・セキュリティでは、特定のマスター・グループの使用に必要な権限のみが受信者に割り当てられます。したがって、プロパゲータは、受信者側のローカルなマスター・グループのうち、特定のグループのみにアクセスできます。図 A-2 にアントラストッド・セキュリティ・モデルを示します。マスター・サイト B には、マスター・グループ A と C のみが含まれるため、マスター・サイト A の受信者はマスター・グループ A と C に対する権限のみが付与されています。このため、プロパゲータからマスター・サイト A へのアクセスは制限されます。

図 A-2 アントラステッド・セキュリティ：マルチマスター・レプリケーション



一般にマスター・サイトは信頼できるとみなされるので、トラステッド・セキュリティ・モデルが使用されます。ただし、リモート・マスター・サイトが信頼できない場合は、アントラステッド・モデルを使用して、受信者に割り当てる権限を制限できます。サイトが信頼できないとみなされるのは、たとえばコンサルティング会社が複数の顧客の仕事を担当する場合などです。表 A-1 の受信者の項目に示す適切な API コールを使用して、ユーザーごとに適切な権限を割り当てます。

表 A-1 必要なユーザー・アカウント

ユーザー	権限
グローバル・レプリケーション管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA
スキーマレベルのレプリケーション管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA
プロパゲータ	DBMS_DEFER_SYS.REGISTER_PROPAGATOR
受信者	詳細は、8-141 ページの「 <a href="#">DBMS_REPCAT_ADMIN パッケージ REGISTER_USER_REPGROUP プロシージャ</a> 」を参照してください。  トラステッド：  DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege => 'receiver' list_of_gnames => NULL  アントラステッド：  DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege => 'receiver' list_of_gnames => 'mastergroupname'

これらのアカウントを作成し、適切な権限を割り当てた後、各サイト間でユーザー名とパスワードを含む次のプライベート・データベース・リンクを作成します。

- ローカルのレプリケーション管理者からリモートのレプリケーション管理者へのリンク
- ローカルのプロパゲータからリモートの受信者へのリンク

1つのユーザー・アカウントがレプリケーション管理者、プロパゲータおよび受信者として機能するように指定した場合は、N (N-1) 個のリンクを作成する必要があります。N はレプリケーション環境内のマスター・サイトの数です。

これらのリンクを作成した後、各ロケーションで DBMS\_DEFER\_SYS.SCHEDULE\_PUSH および DBMS\_DEFER\_SYS.SCHEDULE\_PURGE をコールして、遅延トランザクション・キューを各リモート・ロケーションに伝播する頻度、およびこのキューをパージする頻度を定義する必要があります。DBMS\_DEFER\_SYS.SCHEDULE\_PUSH は、各サイトで、1つのリモート・ロケーションに対し1回ずつ、複数回コールする必要があります。

HQ.WORLD と SALES.WORLD 間のマルチマスター・レプリケーションを設定するためのスクリプトの例を次に示します。

```
/*--- HQ のグローバル・レプリケーション管理者を作成します。---*/
connect system/manager@hq.world
create user repadmin identified by repadmin
execute dbms_repcat_admin.grant_admin_any_schema(username => 'repadmin')

/*--- Sales のグローバル・レプリケーション管理者を作成します。---*/
connect system/manager@sales.world
create user repadmin identified by repadmin
execute dbms_repcat_admin.grant_admin_any_schema(username => 'repadmin')

/*--- HQ でプロパゲータ兼受信者の役割を果たす 1 人のユーザーを作成します。---*/
connect system/manager@hq.world
create user prop_rec identified by prop_rec
/*--- プロパゲータの役割を果たすために必要な権限を付与します。---*/
execute dbms_defer_sys.register_propagator(username => 'prop_rec')
/*--- 受信者の役割を果たすために必要な権限を付与します。---*/
execute dbms_repcat_admin.register_user_repgroup(
    username => 'prop_rec',
    privilege_type => 'receiver',
    list_of_gnames => NULL)

/*--- Sales でプロパゲータ兼受信者の役割を果たす 1 人のユーザーを作成します。---*/
connect system/manager@sales.world
create user prop_rec identified by prop_rec
/*--- プロパゲータの役割を果たすために必要な権限を付与します。---*/execute
dbms_defer_sys.register_propagator(username => 'prop_rec')
/*--- 受信者の役割を果たすために必要な権限を付与します。---*/
execute dbms_repcat_admin.register_user_repgroup(
    username => 'prop_rec',
```

```
        privilege_type => 'receiver',
        list_of_gnames => NULL)

/*--- 必要な USING 句を使用した HQ から Sales へのパブリック・リンクを作成します。---*/
connect system/manager@hq.world
create public database link sales.world using sales.world

/*--- repadmin から repadmin へのプライベート・リンクを作成します。---*/
connect repadmin/repadmin@hq.world
create database link sales.world connect to repadmin identified by repadmin

/*--- HQ から Sales へのレプリケーションをスケジュールします。---*/
execute dbms_defer_sys.schedule_push(
    destination => 'sales.world',
    interval => 'sysdate + 1/24',
    next_date => sysdate,
    stop_on_error => FALSE,
    parallelism => 1)

/*--- HQ での遅延トランザクション・キューのページをスケジュールします。---*/
execute dbms_defer_sys.schedule_purge(
    next_date => sysdate,
    interval => 'sysdate + 1',
    delay_seconds => 0,
    rollback_segment => '')

/*--- スケジュール・プッシュ用のプロパゲータから受信者へのリンクを作成します。---*/
connect prop_rec/prop_rec@hq.world
create database link sales.world connect to prop_rec identified by prop_rec

/*--- 必要な USING 句を使用した Sales から HQ へのパブリック・リンクを作成します。---*/
connect system/manager@sales.world
create public database link hq.world using hq.world

/*--- repadmin から repadmin へのプライベート・リンクを作成します。---*/
connect repadmin/repadmin@sales.world
create database link hq.world connect to repadmin identified by repadmin

/*--- Sales から HQ へのレプリケーションをスケジュールします。---*/
execute dbms_defer_sys.schedule_push(
    destination => 'hq.world',
    interval => 'sysdate + 1/24',
    next_date => sysdate,
    stop_on_error => FALSE,
    parallelism => 1)

/*--- Sales での遅延トランザクション・キューのページをスケジュールします。---*/
```

```

execute dbms_defer_sys.schedule_purge(
    next_date => sysdate,
    interval => 'sysdate + 1',
    delay_seconds => 0,
    rollback_segment => '')

/*--- スケジュール・プッシュ用のプロパゲータから受信者へのリンクを作成します。---*/
connect prop_rec/prop_rec@sales.world
create database link hq.world connect to prop_rec identified by prop_rec

```

## スナップショット・レプリケーションのセキュリティ設定

ほとんどの場合、スナップショット・レプリケーションのセキュリティを設定するときは、Replication Manager の「セットアップ・ウィザード」を使用するのが最も簡単です。ただし、一部の特別なケースでは、レプリケーション・マネジメント API を使用してこれらの設定を行う必要があります。レプリケーション環境を構成するには、データベース管理者が DBA 権限で接続して、必要な権限をレプリケーション管理者に付与する必要があります。

最初に、各スナップショット・サイトで、レプリケーション環境を構成およびメンテナンスする権限と、レプリケートされた変更を伝播する権限を持つユーザー・アカウントを設定します。また、これらのユーザー用に、対応付けられたマスター・サイトへのリンクを定義する必要があります。対応付けられたマスター・サイトで追加のユーザーを作成するか、ユーザーへ追加の権限を割り当てるが必要な場合もあります。

スナップショット・サイトの "ユーザー" のカテゴリとして、レプリケート・オブジェクトにアクセスするエンド・ユーザーに加え、次の 3 つの特別なカテゴリがあります。

- レプリケーション環境の構成およびメンテナンスを行う、レプリケーション管理者
- 遅延トランザクションを伝播する、プロパゲータ
- 対応付けられたマスター表からスナップショットへ変更をプルダウンする、リフレッシュ

通常は、これらの機能を 1 人のユーザーが実行します。ただし、これらの機能をそれぞれ別のユーザーが実行する場合もあります。たとえば、スナップショット・サイト管理者がスナップショットを作成し、別のエンド・ユーザーがリフレッシュを実行する場合などです。

表 A-2 に、スナップショット・サイトの作成とメンテナンスに必要な権限を示します。

表 A-2 スナップショット・サイトに必要なユーザー・アカウント

ユーザー	権限
スナップショット・サイト・レプリケーション管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA
プロパゲータ	DBMS_DEFER_SYS.REGISTER_PROPAGATOR
リフレッシャ	CREATE ANY SNAPSHOT ALTER ANY SNAPSHOT

スナップショット・サイトに適切なユーザーを作成することに加えて、対応付けられたマスター・サイトにも追加ユーザーを作成する必要があります。A-11 ページの表 A-5 に、マスター・サイトのユーザーが新しいスナップショット・サイトをサポートするために必要な権限を示します。

トラステッド・セキュリティとアントラステッド・セキュリティ

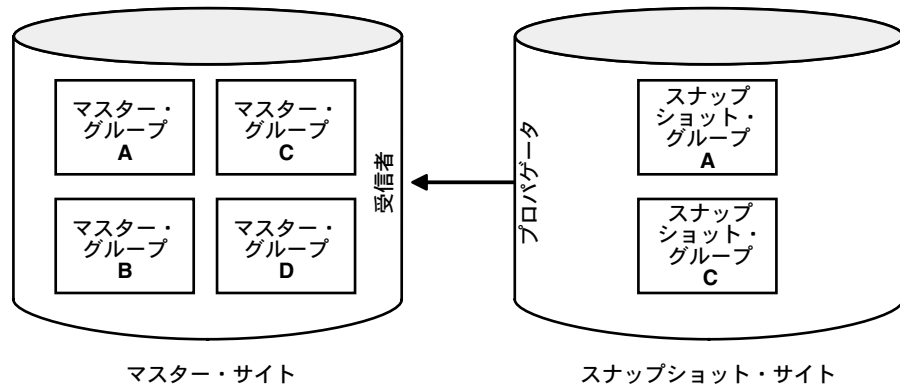
マスター・サイトのユーザーのカテゴリの他に、セキュリティ・モデルとしてトラステッドとアントラステッドのどちらをインプリメントするかを決定する必要があります。トラステッド・セキュリティ・モデルでは、受信者およびプロキシ・スナップショット管理者がすべてのローカル・マスター・グループにアクセスできます。プロキシ・スナップショット管理者と受信者はそれぞれ、リモート・スナップショット・サイトのスナップショット管理者とプロパゲータにかわって、ローカル・マスター・サイトでデータベース・アクティビティを実行するので、プロパゲータとスナップショット管理者にも、マスター・サイトのすべてのマスター・グループへのアクセス権限があります。1 人の受信者をすべての受信トランザクションに使用することに注意してください。

たとえば、図 A-3 の場合を考えます。スナップショット・サイトにはスナップショット・グループ A と C が存在しますが（マスター・サイトのマスター・グループ A と C をベースとしています）、トラステッド・セキュリティ・モデルが使用されているので、プロパゲータとスナップショット管理者はマスター・サイトのマスター・グループ A、B、C、D にアクセスできます。その結果、データベース管理者は任意のリモート・サイトで管理機能を実行し、変更内容をマスター・サイトに伝播できるため、データベース管理の柔軟性が大幅に向上します。ただし、悪意のあるユーザーがリモート・サイトからマスター・サイトのデータを見たり、破損させたりする可能性も高くなります。

使用するセキュリティ・モデルに関係なく、Oracle8i では、オブジェクトがレプリケート環境に追加されるときまたはレプリケート環境から削除されるときに、オブジェクトに適切な権限が自動的に割り当てられます。

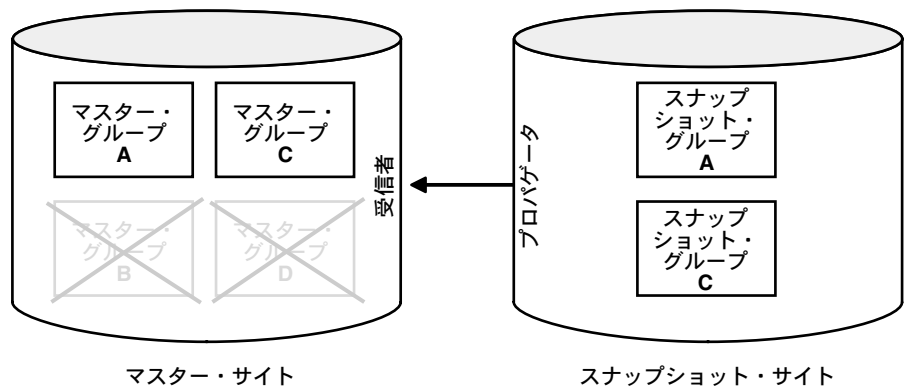


表 A-3 トラストド・セキュリティ：スナップショット・レプリケーション



アントラステッド・セキュリティでは、特定のマスター・グループの使用に必要な権限のみがプロキシ・スナップショット管理者と受信者に付与されます。したがって、プロパゲータとスナップショット管理者は、マスター・サイトのこれらの特定のマスター・グループのみにアクセスできます。図 A-4 に、スナップショット・レプリケーションのアントラステッド・セキュリティ・モデルを示します。スナップショット・サイトにはスナップショット・グループ A と C が含まれるので、マスター・グループ A と C のみへのアクセスが必要です。アントラステッド・セキュリティを使用すると、スナップショット・サイトのプロパゲータまたはスナップショット管理者は、マスター・サイトのマスター・グループ B と D にアクセスできません。

表 A-4 アントラステッド・セキュリティ：スナップショット・レプリケーション



一般にスナップショット・サイトはセキュリティ侵害を受けやすいので、アントラステッド・セキュリティ・モデルが使用されます。スナップショット・サイトでトラステッド・セキュリティ・モデルを使用する理由はほとんど考えられません。スナップショット・サイトではアントラステッド・セキュリティ・モデルを使用することをお薦めします。

トラステッド・セキュリティ・モデルを使用する理由として1つ考えられるのは、スナップショット・サイトがあらゆる観点（セキュリティ、常時ネットワーク接続、リソース）からマスター・サイトと判断される場合で、データをパーティション化する必要があるスナップショットになっている場合です。マルチマスター構成では、水平パーティション化と垂直パーティション化はサポートされていません。

表 A-5 のプロキシ・スナップショット管理者と受信者の項目に示す適切な API コールを使用して、各ユーザーに適切な権限を割り当てます。

表 A-5 マスター・サイトに必要なユーザー・アカウント

ユーザー	権限
プロキシ・スナップショット・サイト管理者	<p>詳細は、8-141 ページの「DBMS_REPCAT_ADMIN パッケージ <a href="#">REGISTER_USER_REPGROUP プロシージャ</a>」を参照してください。</p> <p>トラステッド：</p> <p>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP</p> <p>privilege =&gt; 'proxy_snapadmin'</p> <p>list_of_gnames =&gt; NULL</p> <p>アントラステッド：</p> <p>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP</p> <p>privilege =&gt; 'proxy_snapadmin'</p> <p>list_of_gnames =&gt; 'mastergroupname'</p>
受信者	<p>詳細は、8-141 ページの「DBMS_REPCAT_ADMIN パッケージ <a href="#">REGISTER_USER_REPGROUP プロシージャ</a>」を参照してください。</p> <p>トラステッド：</p> <p>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP</p> <p>privilege =&gt; 'receiver'</p> <p>list_of_gnames =&gt; NULL</p> <p>アントラステッド：</p> <p>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP</p> <p>privilege =&gt; 'receiver'</p> <p>list_of_gnames =&gt; 'mastergroupname'</p>
プロキシ・リフレッシャ	<p>トラステッド：</p> <p>CREATE SESSION 権限付与</p> <p>SELECT ANY TABLE 権限付与</p> <p>アントラステッド：</p> <p>CREATE SESSION 権限付与</p> <p>必要なマスター表およびスナップショット・ログの SELECT 権限付与</p>

スナップショット・サイトと対応付けられたマスター・サイトの両方でアカウントを作成した後、スナップショット・サイトからマスターへのプライベート・データベース・リンクを作成する必要があります（ユーザー名とパスワードを含みます）。

- スナップショット・レプリケーション管理者からプロキシ・スナップショット・レプリケーション管理者へのリンク
- プロパゲータから受信者へのリンク
- リフレッシュャからプロキシ・リフレッシュャへのリンク
- スナップショットの所有者からマスター・サイトへのリンク（リフレッシュ用）

1つのユーザー・アカウントがスナップショット管理者、プロパゲータおよびリフレッシュャとして機能するように指定した場合は、これらの機能に対し、1つのスナップショット・サイトごとにリンクを1つ作成する必要があります。マスター・サイトからスナップショット・サイトへのリンクは必要ありません。

これらのリンクを作成した後、スナップショット・サイトで `DBMS_DEFER_SYS.SCHEDULE_PUSH` および `DBMS_DEFER_SYS.SCHEDULE_PURGE` をコールして、遅延トランザクション・キューを対応付けられたマスター・サイトに伝播する頻度、およびこのキューをパージする頻度を定義する必要があります。また、スナップショット・サイトで `DMBS_REFRESH.REFRESH` をコールして、対応付けられたマスター・サイトからの変更を受信する頻度をスケジュールする必要があります。

---

## ユーザー定義の競合解消方法

この付録では、ユーザー定義の競合解消ファンクションおよび通知ファンクションの作成方法を説明します。次の内容を説明します。

- [ユーザー定義の競合解消方法](#)
- [ユーザー定義の競合通知方法](#)
- [競合解消情報の表示](#)

## ユーザー定義の競合解消方法

ユーザー独自の競合解消方法や通知方法を作成できます。ユーザー定義の競合解消方法とは、TRUE または FALSE のいずれかを返す PL/SQL ファンクションです。TRUE は、列グループへの競合する変更すべてが競合解消方法によって正常に解消されたことを示します。競合が正常に解消されない場合は、FALSE が戻されます。競合解消方法から TRUE が戻されるか、または使用可能な競合解消方法がなくなるまで、方法の評価が順番に続行されます。

競合解消方法で例外が呼び出されると、その方法の評価が停止します。それ以降の順序番号でその他の競合解消方法が用意されていても、評価は行われません。

## 競合解消方法のパラメータ

ユーザー定義の競合解消方法に必要なパラメータは、解消する競合のタイプ（一意、更新または削除）およびレプリケートされる表の列によって決定されます。すべての競合解消方法は、表の古い列値、新しい列値およびカレント列値の組合せを使用します。

- 古い値は、発信元サイトでの変更前の行の値を表します。
- 新しい値は、発信元サイトでの変更後の行の値を表します。
- カレント値は、受信サイトでの対応する行の値を表します。

---

---

**注意：** Oracle では、比較する行の判別に、主キー、すなわち SET\_COLUMNS で指定されたキーが使用されることに注意してください。

---

---

競合解消ファンクションは、DBMS\_REPCAT.ADD\_conflicttype\_RESOLUTION プロシージャの PARAMETER\_COLUMN\_NAME 引数で指定した列値をパラメータとして受け取ります。列パラメータは、PARAMETER\_COLUMN\_NAME 引数に示される順序で、またこの引数に '\*' を指定した場合はアルファベット昇順で、競合解消方法に渡されます。古い列値と新しい列値の両方が（更新競合の）パラメータとして渡される場合、列の古い値は新しい値の直前に置かれます。

---

**注意：** ユーザー定義の競合解消方法では、パラメータ列のタイプのチェックは、対応付けられたレプリケート表のレプリケーション・サポートを再生成するまで実行されません。

---

## 更新競合の解消

更新競合の場合、ユーザー定義のファンクションでは、列グループ内の各列に次の値を使用する必要があります。

- 発信元サイトからの古い列値。このパラメータのモードは IN です。この値は変更できません。
- 発信元サイトからの新しい列値。このパラメータのモードは IN OUT です。そのファンクションで競合を正常に解消できる場合は、必要に応じて新しい列値を変更する必要があります。
- 受信サイトからのカレント列値。このパラメータのモードは IN です。

列値は、古い値、新しい値、カレント値と連続して受信されます。競合解消方法に渡す最後の引数は、ブール型フラグであることが必要です。このフラグが FALSE の場合、IN OUT パラメータ（すなわち、新しい列）の値をすでに更新したこと、およびカレント列値をこの新しい値に更新する必要があることを示します。このフラグが TRUE の場合、カレント列値を変更できないことを示します。

## 一意性競合の解消

一意性競合は、INSERT または UPDATE の結果として発生する場合があります。ユーザー定義の一意性競合解消方法では、列グループの各列に、発信元サイトからの IN OUT モードの新しい列値を受け入れる必要があります。競合解消方法に渡す最後のパラメータは、ブール型フラグである必要があります。

競合解消方法で競合が解消された場合、新しい列値を変更して、カレント行への挿入またはカレント行の新しい列値による更新を可能にします。新しい列値を廃棄する場合はファンクションでブール型フラグを TRUE に設定し、それ以外の場合は FALSE に設定します。

競合解消方法では一意性競合におけるデータの収束を保証できないので、ユーザー定義の一意性競合解消方法に通知メカニズムを組み込む必要があります。

## 削除競合の解消

削除競合は、ローカル・サイトからの削除が正常に実行されても、対応付けられた行が（更新されたなどの理由で）リモート・サイトにない場合に発生します。削除競合の場合、ファンクションでは、行全体の古い列値を IN OUT モードで受け入れる必要があります。競合解消方法に渡す最後のパラメータは、ブール型フラグである必要があります。

競合解消方法で競合が解消されると、古い列値が変更され、すべての古い列値に一致するカレント行を削除できます。これらの列値を廃棄する場合はファンクションでブール型フラグを TRUE に設定し、それ以外の場合は FALSE に設定します。

ローカル・サイトで削除を実行し、リモート・サイトで更新を実行した場合、リモート・サイトで削除競合が検出されますが、ローカル・サイトでは解消不可能な更新競合が検出されます。このタイプの競合は自動的に処理できません。この競合によって NO\_DATA\_FOUND 例外が発生し、このトランザクションはエラー・トランザクションとして記録されます。

これらのタイプの更新や削除の競合を適切に処理するメカニズムを設計することは困難です。削除された行を単純にマーク付けして、プロシージャ型レプリケーションを使用してパージすることで、これらのタイプの競合を完全に回避するほうがはるかに容易です。

## 制限事項

ユーザー定義の競合解消方法で次の SQL 文を使用しないでください。これらの文を使用すると、予期しない結果が起こる可能性があります。

- データ定義言語（DDL）文（CREATE、ALTER、DROP など）
- トランザクション制御文（COMMIT、ROLLBACK など）
- セッション制御（ALTER SESSION など）
- システム制御（ALTER SYSTEM など）

## ユーザー定義の競合解消方法の例

標準的な事前作成の最大値（MAXIMUM）および加算（ADDITIVE）による競合解消方法を変形した、ユーザー定義方法の例を次に示します。標準方法と異なり、これらのユーザー定義ファンクションでは、競合解消に使用される列内の NULL 値を処理できます。



## 最大値ユーザー・ファンクション

-- 最大値による方法に類似したユーザー・ファンクションです。  
 -- cur が NULL または  $cur < new$  の場合は、new 値を使用します。  
 -- new が NULL または  $new < cur$  の場合は、cur 値を使用します。  
 -- いずれも NULL の場合は、解消は行われません。  
 -- 値が常に増加していない限り、3 つ以上のマスターのデータを収束しないでください。

```
FUNCTION max_null_loses(old          IN    NUMBER,
                        new          IN OUT NUMBER,
                        cur          IN    NUMBER,
                        ignore_discard_flag OUT  BOOLEAN)

RETURN BOOLEAN IS
BEGIN
  IF (new IS NULL AND cur IS NULL) OR new = cur THEN
    RETURN FALSE;
  END IF;
  IF new IS NULL THEN
    ignore_discard_flag := TRUE;
  ELSIF cur IS NULL THEN
    ignore_discard_flag := FALSE;
  ELSIF new < cur THEN
    ignore_discard_flag := TRUE;
  ELSE
    ignore_discard_flag := FALSE;
  END IF;
  RETURN TRUE;
END max_null_loses;
```

## 加算ユーザー・ファンクション

-- 加算による方法に類似したユーザー・ファンクションです。  
 -- old が NULL の場合、old = 0 です。  
 -- new が NULL の場合、new = 0 です。  
 -- cur が NULL の場合、cur = 0 です。  
 --  $new = cur + (new - old)$  -> 加算による方法の場合と同様です。

```
FUNCTION additive_nulls(old          IN    NUMBER,
                        new          IN OUT NUMBER,
                        cur          IN    NUMBER,
                        ignore_discard_flag OUT  BOOLEAN)

RETURN BOOLEAN IS
  old_val NUMBER := 0.0;
  new_val NUMBER := 0.0;
  cur_val NUMBER := 0.0;
BEGIN
  IF old IS NOT NULL THEN
    old_val := old;
```

```
END IF;
IF new IS NOT NULL THEN
    new_val := new;
END IF;
IF cur IS NOT NULL THEN
    cur_val := cur;
END IF;
new := cur_val + (new_val - old_val);
ignore_discard_flag := FALSE;
RETURN TRUE;
END additive_nulls;
```

## ユーザー定義の競合通知方法

競合通知方法は、競合解消のかわりまたは追加として競合を通知する、ユーザー定義のファンクションです。たとえば、独自の競合通知方法を作成して、データベース表に競合情報を記録したり、電子メール・メッセージを送信したり、管理者を呼び出したりできます。競合通知方法を作成した後、それを列グループ（または制約）に特定の順序で割り当てて、競合が発生した時点か、後続の競合解消方法を試行する前か、競合解消を試行して失敗した後に、通知を受けることができます。

レプリケート表にユーザー定義の競合通知メカニズムを構成するには、次のステップを実行する必要があります。

1. 競合通知ログを作成します。
2. パッケージ内にユーザー定義の競合通知方法を作成します。

次の項で各ステップを説明します。

## 競合通知ログの作成

ユーザー定義の競合通知方法を使用するレプリケート表を構成するときは、最初に、競合通知を記録できるデータベース表を作成します。マスター・グループ内の1つ以上の表での競合通知を記録するデータベース表を作成できます。

すべてのマスター・サイトで競合通知ログ表を作成するには、レプリケーション実行 DDL 機能を使用します。詳細は、8-114 ページの DBMS\_REPCAT パッケージ「[EXECUTE\\_DDL プロシージャ](#)」を参照してください。競合通知表のエントリは競合を検出するサイトに固有のエントリであるため、競合通知表に対するレプリケーション・サポートを生成しないでください。

## 競合通知ログ表の例

次の CREATE TABLE 文では、マスター・グループのいくつかの表から競合通知を記録するのに使用する表を作成します。

```
CREATE TABLE conf_report (  
  line          NUMBER(2),    --- メッセージ・テキストの順序付けに使用  
  txt           VARCHAR2(80), --- 競合通知メッセージ  
  timestamp     DATE,         --- 競合の発生した時間  
  table_name    VARCHAR2(30), --- 競合の発生した表  
  table_owner   VARCHAR2(30), --- 表の所有者  
  conflict_type VARCHAR2(6)    --- INSERT、DELETE または UNIQUE  
)
```

## 競合通知パッケージの作成

競合通知方法を作成するには、PL/SQL パッケージ内で方法を定義し、対応付けられたレプリケート表とともに PL/SQL パッケージをマスター・グループの一部としてレプリケートする必要があります。

競合通知方法では、競合の通知のみ、または競合の通知と解消の両方を実行できます。可能な場合は常に、Oracle の事前作成競合解消方法のいずれかを使用して競合を解消する必要があります。ユーザー定義の競合通知方法で競合の通知のみを実行する場合は、競合を解消する競合解消方法とともに、ユーザー定義の通知方法を列グループ（または制約）に割り当ててください。

---

---

**注意：** レプリケーションの競合が最終的に解消されない場合は、通知表に加えられた更新内容を含め、トランザクション全体がロールバックされます。トランザクションとは独立して通知が必要な場合は、DBMS\_PIPES パッケージまたは Oracle Office とのデータベース・インタフェースを使用する通知メカニズムを設計できます。

---

---

## 競合通知パッケージの例

次のパッケージおよびパッケージ本体では、CUSTOMERS 表の一意性競合を定義済の CONF\_REPORT 表に記録することによって、簡単な競合通知を実行します。

---

---

**注意：** この競合通知の例では、競合は解消されません。したがって、競合解消方法（廃棄や上書きなど）を用意するか、エラーが解消されずにトランザクションがロールバックされても正しく機能する通知メカニズム（電子メールの利用など）を用意する必要があります。次のユーザー定義の競合通知方法に簡単な変更を加えれば、さらに有効な処置を取ることができます。たとえば、このパッケージでは、通知メッセージを記録でき、また DBMS\_OFFICE ユーティリティ・パッケージを使用して Oracle Office の電子メール・メッセージを管理者に送信できます。

---

---

```
CREATE OR REPLACE PACKAGE notify AS
  -- CUSTOMERS 表の一意性制約違反を報告します。
  FUNCTION customers_unique_violation (
    first_name      IN OUT VARCHAR2,
    last_name       IN OUT VARCHAR2,
    discard_new_values IN OUT BOOLEAN)
    RETURN BOOLEAN;
END notify;
/

CREATE OR REPLACE PACKAGE BODY notify AS
  -- 通知メッセージを保持する PL/SQL 表を定義します。
  TYPE message_table IS TABLE OF VARCHAR2(80) INDEX BY BINARY_INTEGER;

  PROCEDURE report_conflict (
    conflict_report IN MESSAGE_TABLE,
    report_length   IN NUMBER,
    conflict_time   IN DATE,
    conflict_table  IN VARCHAR2,
    table_owner     IN VARCHAR2,
    conflict_type   IN VARCHAR2) IS
  BEGIN
    FOR idx IN 1..report_length LOOP
      BEGIN
        INSERT INTO sales.conf_report
          (line, txt, timestamp, table_name, table_owner, conflict_type)
          VALUES (idx, SUBSTR(conflict_report(idx),1,80), conflict_time,
            conflict_table, table_owner, conflict_type);
      EXCEPTION WHEN others THEN NULL;
    END;
  END LOOP;
END report_conflict;
```

```

-- CUSTOMERS 表で一意性制約違反が検出されたときに
-- 最初にコールされる競合解消方法です。

FUNCTION customers_unique_violation (
    first_name IN OUT VARCHAR2,
    last_name IN OUT VARCHAR2,
    discard_new_values IN OUT BOOLEAN)
RETURN BOOLEAN IS
    local_node VARCHAR2(128);
    conf_report MESSAGE_TABLE;
    conf_time DATE := SYSDATE;
BEGIN
    -- ローカル・サイトのグローバル・ネームを取得します。
    BEGIN
        SELECT global_name INTO local_node FROM global_name;
    EXCEPTION WHEN others THEN local_node := '?';
    END;
    -- DBA に対するメッセージを生成します。
    conf_report(1) := 'UNIQUENESS CONFLICT DETECTED IN TABLE CUSTOMERS ON ' ||
        TO_CHAR(conf_time, 'MM-DD-YYYY HH24:MI:SS');
    conf_report(2) := ' AT NODE ' || local_node;
    conf_report(3) := 'ATTEMPTING TO RESOLVE CONFLICT USING ' ||
        'APPEND SEQUENCE METHOD';
    conf_report(4) := 'FIRST NAME: ' || first_name;
    conf_report(5) := 'LAST NAME: ' || last_name;
    conf_report(6) := NULL;
    --- 競合を報告します。
    report_conflict(conf_report, 5, conf_time, 'CUSTOMERS',
        'OFF_SHORE_ACCOUNTS', 'UNIQUE');
    --- 新しい列値を廃棄しないでください。
    --- これらの値は他の競合解消方法で必要になります。
    discard_new_values := FALSE;
    --- 競合が解消されなかったことを示します。
    RETURN FALSE;
END customers_unique_violation;
END notify;
/

```

# 競合解消情報の表示

Oracle には、レプリケーション・カタログ (REPCAT)・ビューが用意されており、このビューを使用して、レプリケート環境の各表および列グループで使用している競合解消方法を判別できます。各ビューには、USER\_\*、ALL\_\* および SYS.DBA\_\* という 3 つのカテゴリがあります。使用できるビューを次に示します。

ALL_REPRESOLUTION_ METHOD	使用可能な競合解消方法がすべて表示されます。
ALL_REPCOLUMN_GROUP	データベースに定義された列グループがすべて表示されます。
ALL_REPGROUPED_COLUMN	データベース内の各列グループの列がすべて表示されます。
ALL_REPPRIORITY_GROUP	データベースに定義された優先グループおよびサイト優先順位グループがすべて表示されます。
ALL_REPPRIORITY	各優先グループまたはサイト優先順位グループの値および対応する優先順位レベルが表示されます。
ALL_REPCONFLICT	データベース内の表、列グループおよび一意性制約に関して解消方法を指定した競合のタイプ（削除、更新または一意性）が表示されます。
ALL_REPRESOLUTION	各オブジェクトで発生する競合解消方法に関する、より具体的な情報が表示されます。
ALL_REPPARAMETER_COLUMN	競合解消方法が競合の解消に使用する列が表示されます。

**関連項目：** [第 9 章「データ・ディクショナリ・ビュー」](#)

## A

ALL\_REFRESH\_CHILDREN ビュー, 9-49  
 ALL\_REFRESHES ビュー, 9-48  
 ALL\_REGISTERED\_SNAPSHOTS ビュー, 9-50  
 ALL\_REPCAT\_REFRESH\_TEMPLATES ビュー, 9-7  
 ALL\_REPCAT\_TEMPLATE\_OBJECTS ビュー, 9-8  
 ALL\_REPCAT\_TEMPLATE\_PARSMS ビュー, 9-10  
 ALL\_REPCAT\_TEMPLATE\_SITES ビュー, 9-12  
 ALL\_REPCAT\_USER\_AUTHORIZATIONS ビュー,  
 9-13  
 ALL\_REPCAT\_USER\_PARM\_VALUES ビュー, 9-14  
 ALL\_REPCATLOG ビュー, 9-6  
 ALL\_REPCOLUMN\_GROUP ビュー, 9-17  
 ALL\_REPCOLUMN ビュー, 9-16  
 ALL\_REPCONFLICT ビュー, 9-17  
 ALL\_REPDDL ビュー, 9-18  
 ALL\_REPGENOBJECTS ビュー, 9-18  
 ALL\_REPGROUP\_PRIVILEGES ビュー, 9-20  
 ALL\_REPGROUPED\_COLUMN ビュー, 9-21  
 ALL\_REPGROUP ビュー, 9-19  
 ALL\_REPKEY\_COLUMNS ビュー, 9-21  
 ALL\_REPOBJECT ビュー, 9-22  
 ALL\_REPPARAMETER\_COLUMN ビュー, 9-23  
 ALL\_REPPRIORITY\_GROUP ビュー, 9-26  
 ALL\_REPPRIORITY ビュー, 9-24  
 ALL\_REPPROP ビュー, 9-26  
 ALL\_REPRESOL\_STATS\_CONTROL ビュー, 9-27  
 ALL\_REPRESOLUTION\_METHOD ビュー, 9-29  
 ALL\_REPRESOLUTION\_STATISTICS ビュー, 9-29  
 統計収集, 6-24  
 ALL\_REPRESOLUTION ビュー, 9-28  
 ALL\_REPSITES ビュー, 9-30  
 ALL\_SNAPSHOT\_LOGS ビュー, 9-50  
 ALL\_SNAPSHOT\_REFRESH\_TIMES ビュー, 9-51

ALL\_SNAPSHOTS ビュー, 9-52  
 ALTER SNAPSHOT LOG 文, 7-17

## D

DBA\_REFRESH\_CHILDREN ビュー, 9-54  
 DBA\_REFRESH ビュー, 9-54  
 DBA\_REGISTERED\_SNAPSHOT\_GROUPS ビュー,  
 9-54  
 DBA\_REGISTERED\_SNAPSHOTS ビュー, 9-54  
 DBA\_REPCAT\_REFRESH\_TEMPLATES ビュー, 9-31  
 DBA\_REPCAT\_TEMPLATE\_OBJECTS ビュー, 9-32  
 DBA\_REPCAT\_TEMPLATE\_PARSMS ビュー, 9-32  
 DBA\_REPCAT\_TEMPLATE\_SITES ビュー, 9-32  
 DBA\_REPCAT\_USER\_AUTHORIZATIONS ビュー,  
 9-32  
 DBA\_REPCAT\_USER\_PARM\_VALUES ビュー, 9-32  
 DBA\_REPCATLOG ビュー, 9-31  
 パージ, 8-120  
 DBA\_REPCOLUMN\_GROUP ビュー, 9-33  
 更新, 8-81  
 DBA\_REPCOLUMN ビュー, 9-33  
 更新, 7-45  
 DBA\_REPCONFLICT ビュー, 9-33  
 DBA\_REPDDL ビュー, 9-33  
 DBA\_REPGENOBJECTS ビュー, 9-33  
 DBA\_REPGROUP\_PRIVILEGES ビュー, 9-34  
 DBA\_REPGROUPED\_COLUMN ビュー, 9-34  
 DBA\_REPGROUP ビュー, 9-33  
 更新, 8-83  
 DBA\_REPKEY\_COLUMNS ビュー, 9-34  
 DBA\_REPOBJECT ビュー, 9-34  
 更新, 8-84  
 DBA\_REPPARAMETER\_COLUMN ビュー, 9-34

DBA\_REPPRIORITY\_GROUP ビュー, 9-35  
     更新, 7-45, 8-82  
 DBA\_REPPRIORITY ビュー, 9-34  
 DBA\_REPPROP ビュー, 9-35  
 DBA\_REPRESOL\_STATS\_CONTROL ビュー, 9-35  
 DBA\_REPRESOLUTION\_METHOD ビュー, 9-35  
 DBA\_REPRESOLUTION\_STATISTICS ビュー, 9-35  
     ページ, 6-25, 8-121  
 DBA\_REPRESOLUTION ビュー, 9-35  
     更新, 7-45, 8-87  
 DBA\_REPSHEMA ビュー  
     更新, 7-45  
 DBA\_REPSITES ビュー, 9-36  
     更新, 8-85  
 DBA\_SNAPSHOT\_LOG\_FILTER\_COLS ビュー, 9-54  
 DBA\_SNAPSHOT\_LOGS ビュー, 9-54  
 DBA\_SNAPSHOT\_REFRESH\_TIMES ビュー, 9-55  
 DBA\_SNAPSHOTS ビュー, 9-55  
 DBMS\_DEFER\_QUERY パッケージ, 8-9  
     GET\_ARG\_FORM ファンクション, 8-9  
     GET\_ARG\_TYPE ファンクション, 8-10  
     GET\_CALL\_ARGS プロシージャ, 8-11  
     GET\_CHAR\_ARG プロシージャ, 8-12  
     GET\_datatype\_ARG ファンクション, 8-12  
     GET\_NUMBER\_ARG プロシージャ, 8-12  
     GET\_RAW\_ARG プロシージャ, 8-12  
     GET\_ROWID\_ARG プロシージャ, 8-12  
     GET\_VARCHAR2\_ARG プロシージャ, 8-12  
 DBMS\_DEFER\_SYS パッケージ, 8-15  
     ADD\_DEFAULT\_DEST プロシージャ, 8-16  
     DELETE\_DEF\_DESTINATION プロシージャ, 8-17  
     DELETE\_DEFAULT\_DEST プロシージャ, 8-17  
     DELETE\_ERROR プロシージャ, 8-18  
     DELETE\_TRAN プロシージャ, 8-18, 8-19, 8-21  
     DISABLED ファンクション, 8-19  
     EXCLUDE\_PUSH ファンクション, 8-20  
     EXECUTE\_ERROR\_AS\_USER プロシージャ, 7-27, 8-21  
     EXECUTE\_ERROR プロシージャ, 7-26, 7-30, 8-21  
     PURGE ファンクション, 7-25, 8-22  
     PUSH ファンクション, 7-25, 8-24  
     REGISTER\_PROPAGATOR プロシージャ, 2-5, 2-15, 8-27  
     SCHEDULE\_EXECUTION プロシージャ, 8-29  
     SCHEDULE\_PURGE プロシージャ, 2-6, 2-17, 8-27  
     SCHEDULE\_PUSH プロシージャ, 2-12, 2-17, 8-29  
     SET\_DISABLED プロシージャ, 8-30  
     UNREGISTER\_PROPAGATOR プロシージャ, 8-31  
     UNSCHEDULE\_PURGE プロシージャ, 8-32  
     UNSCHEDULE\_PUSH プロシージャ, 8-32  
 DBMS\_DEFER パッケージ, 8-4  
     CALL プロシージャ, 8-4  
     CHAR\_ARG プロシージャ, 8-6  
     COMMIT\_WORK プロシージャ, 8-5  
     datatype\_ARG プロシージャ, 8-6  
     DATE\_ARG プロシージャ, 8-6  
     NUMBER\_ARG プロシージャ, 8-6  
     RAW\_ARG プロシージャ, 8-6  
     ROWID\_ARG プロシージャ, 8-6  
     TRANSACTION プロシージャ, 8-7  
     VARCHAR2\_ARG プロシージャ, 8-6  
 DBMS\_OFFLINE\_OG パッケージ, 8-34  
     BEGIN\_INSTANTIATION プロシージャ, 7-31, 8-34  
     BEGIN\_LOAD プロシージャ, 7-33, 8-35  
     END\_INSTANTIATION プロシージャ, 7-34, 8-37  
     END\_LOAD プロシージャ, 7-33, 8-38  
     RESUME\_SUBSET\_OF\_MASTERS プロシージャ, 7-32, 8-39  
 DBMS\_OFFLINE\_SNAPSHOT パッケージ, 8-42  
     BEGIN\_LOAD プロシージャ, 7-37, 7-38, 8-42  
     END\_LOAD プロシージャ, 7-39, 8-43  
 DBMS\_RECTIFIER\_DIFF パッケージ, 7-41, 8-45  
     DIFFERENCES プロシージャ, 7-41, 8-45  
     RECTIFY プロシージャ, 7-41, 8-48  
 DBMS\_REFRESH パッケージ, 8-51  
     ADD プロシージャ, 5-7, 8-51  
     CHANGE プロシージャ, 8-52  
     DESTROY プロシージャ, 8-54  
     MAKE プロシージャ, 5-6, 8-55  
     REFRESH プロシージャ, 8-57  
     SUBTRACT プロシージャ, 8-58  
 DBMS\_REPCAT\_ADMIN パッケージ, 8-139  
     GRANT\_ADMIN\_ANY\_SCHEMA プロシージャ, 2-4, 2-15, 8-139  
     GRANT\_ADMIN\_SCHEMA プロシージャ, 8-140  
     REGISTER\_USER\_REPGROUP プロシージャ, 2-5, 2-6, 8-141  
     REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャ, 8-142  
     REVOKE\_ADMIN\_SCHEMA プロシージャ, 8-143



UNREGISTER\_USER\_REPGROUP プロシージャ,  
 8-144  
 DBMS\_REPCAT\_INSTANTIATE パッケージ, 8-146  
 DROP\_SITE\_INSTANTIATION プロシージャ,  
 7-10, 8-146  
 INSTANTIATE\_OFFLINE\_REPAPI ファンク  
 ション, 8-149  
 INSTANTIATE\_OFFLINE ファンクション, 8-147  
 INSTANTIATE\_ONLINE ファンクション, 8-151  
 DBMS\_REPCAT\_RGT パッケージ, 8-154  
 ALTER\_REFRESH\_TEMPLATE プロシージャ,  
 8-156  
 ALTER\_TEMPLATE\_OBJECT プロシージャ, 8-158  
 ALTER\_TEMPLATE\_PARM プロシージャ, 4-7,  
 8-160  
 ALTER\_USER\_AUTHORIZATION プロシージャ,  
 8-162  
 ALTER\_USER\_PARM\_VALUE プロシージャ,  
 8-163  
 COMPARE\_TEMPLATES ファンクション, 8-165  
 COPY\_TEMPLATE ファンクション, 8-166  
 CREATE\_OBJECT\_FROM\_EXISTING ファンク  
 ション, 8-168  
 CREATE\_REFRESH\_TEMPLATE ファンクション,  
 8-169  
 CREATE\_REFRESH\_TEMPLATE プロシージャ,  
 4-4  
 CREATE\_TEMPLATE\_OBJECT ファンクション,  
 8-171  
 CREATE\_TEMPLATE\_OBJECT プロシージャ, 4-5  
 CREATE\_TEMPLATE\_PARM ファンクション,  
 8-173  
 CREATE\_USER\_AUTHORIZATION ファンク  
 ション, 8-176  
 CREATE\_USER\_AUTHORIZATION プロシージャ,  
 4-9  
 CREATE\_USER\_PARM\_VALUE ファンクション,  
 8-177  
 CREATE\_USER\_PARM\_VALUE プロシージャ, 4-8  
 DELETE\_RUNTIME\_PARMS プロシージャ, 8-179  
 DROP\_ALL\_OBJECTS プロシージャ, 8-179  
 DROP\_ALL\_TEMPLATE\_PARMS プロシージャ,  
 8-180  
 DROP\_ALL\_TEMPLATE\_SITES プロシージャ,  
 8-181  
 DROP\_ALL\_TEMPLATES プロシージャ, 8-182  
 DROP\_ALL\_USER\_AUTHORIZATIONS プロシー  
 ジャ, 8-182  
 DROP\_ALL\_USER\_PARM\_VALUES プロシージャ,  
 8-183  
 DROP\_REFRESH\_TEMPLATE プロシージャ, 8-184  
 DROP\_SITE\_INSTANTIATION プロシージャ,  
 7-12, 8-185  
 DROP\_TEMPLATE\_OBJECT プロシージャ, 8-186  
 DROP\_TEMPLATE\_PARM プロシージャ, 8-187  
 DROP\_USER\_AUTHORIZATION プロシージャ,  
 8-188  
 DROP\_USER\_PARM\_VALUE プロシージャ, 8-189  
 GET\_RUNTIME\_PARM\_ID ファンクション, 8-190  
 INSERT\_RUNTIME\_PARMS プロシージャ, 8-190  
 INSTANTIATE\_OFFLINE\_REPAPI ファンク  
 ション, 8-194  
 INSTANTIATE\_OFFLINE ファンクション, 8-192  
 INSTANTIATE\_OFFLINE プロシージャ, 4-13  
 INSTANTIATE\_ONLINE ファンクション, 8-197  
 INSTANTIATE\_ONLINE プロシージャ, 4-13  
 LOCK\_TEMPLATE\_EXCLUSIVE プロシージャ,  
 8-199  
 LOCK\_TEMPLATE\_SHARED プロシージャ, 8-199  
 DBMS\_REPCAT パッケージ, 8-59  
 ADD\_DELETE\_RESOLUTION プロシージャ, 8-68  
 ADD\_GROUPED\_COLUMN プロシージャ, 8-63  
 ADD\_MASTER\_DATABASE プロシージャ, 3-7,  
 7-3, 8-64  
 ADD\_PRIORITY\_CHAR プロシージャ, 8-65  
 ADD\_PRIORITY\_datatype プロシージャ, 8-65  
 ADD\_PRIORITY\_DATE プロシージャ, 8-65  
 ADD\_PRIORITY\_NUMBER プロシージャ, 8-65  
 ADD\_PRIORITY\_VARCHAR2 プロシージャ, 8-65  
 ADD\_SITE\_PRIORITY\_SITE プロシージャ, 6-15,  
 8-67  
 ADD\_UNIQUENESS\_RESOLUTION プロシージャ,  
 8-68  
 ADD\_UPDATE\_RESOLUTION プロシージャ, 6-4,  
 6-5, 6-8, 6-10, 6-13, 6-16, 8-68  
 ALTER\_MASTER\_PROPAGATION プロシージャ,  
 8-71  
 ALTER\_MASTER\_REPOBJECT プロシージャ, 6-7,  
 6-14, 6-17, 6-22, 7-27, 8-73  
 ALTER\_PRIORITY\_CHAR プロシージャ, 8-75  
 ALTER\_PRIORITY\_datatype プロシージャ, 8-75  
 ALTER\_PRIORITY\_DATE プロシージャ, 8-75  
 ALTER\_PRIORITY\_NUMBER プロシージャ, 8-75

ALTER\_PRIORITY\_RAW プロシージャ, 8-75  
ALTER\_PRIORITY プロシージャ, 8-74  
ALTER\_SITE\_PRIORITY\_SITE プロシージャ, 8-78  
ALTER\_SITE\_PRIORITY プロシージャ, 8-77  
ALTER\_SNAPSHOT\_PROPAGATION プロシージャ, 8-79  
CANCEL\_STATISTICS プロシージャ, 6-24, 8-80  
COMMENT\_ON\_COLUMN\_GROUP プロシージャ, 7-45, 8-81  
COMMENT\_ON\_DELETE\_RESOLUTION プロシージャ, 7-45, 8-87  
COMMENT\_ON\_PRIORITY\_GROUP プロシージャ, 7-45, 8-82  
COMMENT\_ON\_REPGROUP プロシージャ, 7-45, 8-83  
COMMENT\_ON\_REPOBJECT プロシージャ, 8-84  
COMMENT\_ON\_REPSchema プロシージャ, 7-45  
COMMENT\_ON\_REPSITES プロシージャ, 8-85  
COMMENT\_ON\_SITE\_PRIORITY プロシージャ, 8-82  
COMMENT\_ON\_SNAPSHOT\_REPSITES プロシージャ, 8-86  
COMMENT\_ON\_UNIQUE\_RESOLUTION プロシージャ, 7-45, 8-87  
COMMENT\_ON\_UPDATE\_RESOLUTION プロシージャ, 7-45, 8-87  
COMPARE\_OLD\_VALUES プロシージャ, 8-88  
CREATE\_MASTER\_REPGROUP プロシージャ, 3-5, 8-90  
CREATE\_MASTER\_REPOBJECT プロシージャ, 6-7, 8-91  
CREATE\_SNAPSHOT\_REPGROUP プロシージャ, 5-5, 7-6, 7-37, 8-94  
CREATE\_SNAPSHOT\_REPOBJECT プロシージャ, 5-6, 7-6, 8-95  
DEFINE\_COLUMN\_GROUP プロシージャ, 8-97  
DEFINE\_PRIORITY\_GROUP プロシージャ, 8-98  
DEFINE\_SITE\_PRIORITY プロシージャ, 6-15, 8-99  
DO\_DEFERRED\_REPCAT\_ADMIN プロシージャ, 7-31, 8-100  
DROP\_COLUMN\_GROUP プロシージャ, 8-101  
DROP\_DELETE\_RESOLUTION プロシージャ, 8-113  
DROP\_GROUPED\_COLUMN プロシージャ, 8-102  
DROP\_MASTER\_REPGROUP プロシージャ, 8-103

DROP\_MASTER\_REPOBJECT プロシージャ, 8-105  
DROP\_PRIORITY\_CHAR プロシージャ, 8-108  
DROP\_PRIORITY\_datatype プロシージャ, 8-108  
DROP\_PRIORITY\_DATE プロシージャ, 8-108  
DROP\_PRIORITY\_GROUP プロシージャ, 8-107  
DROP\_PRIORITY\_NUMBER プロシージャ, 8-108  
DROP\_PRIORITY\_VARCHAR2 プロシージャ, 8-108  
DROP\_PRIORITY プロシージャ, 8-106  
DROP\_SITE\_PRIORITY\_SITE プロシージャ, 8-110  
DROP\_SITE\_PRIORITY プロシージャ, 8-109  
DROP\_SNAPSHOT\_REPGROUP プロシージャ, 7-13, 8-111  
DROP\_SNAPSHOT\_REPOBJECT プロシージャ, 7-14, 8-112  
DROP\_UNIQUE\_RESOLUTION プロシージャ, 8-113  
DROP\_UPDATE\_RESOLUTION プロシージャ, 8-113  
EXECUTE\_DDL プロシージャ, 8-114  
GENERATE\_REPLICATION\_SUPPORT プロシージャ, 3-8, 7-28, 8-116  
GENERATE\_SNAPSHOT\_SUPPORT プロシージャ, 8-117  
MAKE\_COLUMN\_GROUP プロシージャ, 6-3, 6-5, 6-8, 6-9, 6-11, 6-14, 8-119  
PURGE\_MASTER\_LOG プロシージャ, 8-120  
PURGE\_STATISTICS プロシージャ, 6-25, 8-121  
REFRESH\_SNAPSHOT\_REPGROUP プロシージャ, 8-122  
REGISTER\_SNAPSHOT\_REPGROUP プロシージャ, 8-123  
REGISTER\_STATISTICS プロシージャ, 6-24, 8-124  
RELOCATE\_MASTERDEF プロシージャ, 7-2, 8-125  
REMOVE\_MASTER\_DATABASES プロシージャ, 8-126  
REMOVE\_MASTER\_DATABASE プロシージャ, 7-4  
REPCAT\_IMPORT\_CHECK プロシージャ, 8-128  
RESUME\_MASTER\_ACTIVITY プロシージャ, 3-9, 8-129  
SEND\_OLD\_VALUES プロシージャ, 8-129  
SET\_COLUMNS プロシージャ, 8-90, 8-131  
SUSPEND\_MASTER\_ACTIVITY プロシージャ, 8-132

SWITCH\_SNAPSHOT\_MASTER プロシージャ,  
7-9, 8-133  
UNREGISTER\_SNAPSHOT\_REPGROUP プロシ  
ージャ, 7-15, 8-134  
VALIDATE プロシージャ, 8-135  
WAIT\_MASTER\_LOG プロシージャ, 8-137  
コメント・プロシージャ, 7-45  
DBMS\_REPUTIL パッケージ, 8-200  
FROM\_REMOTE ファンクション, 8-201  
GLOBAL\_NAME ファンクション, 8-202  
MAKE\_INTERNAL\_PKG プロシージャ, 8-202  
REPLICATION\_IS\_ON ファンクション, 8-201  
REPLICATION\_OFF プロシージャ, 8-200  
REPLICATION\_ON プロシージャ, 8-201  
SYNC\_UP\_REP プロシージャ, 8-203  
DBMS\_SNAPSHOT パッケージ, 8-204  
BEGIN\_TABLE\_REORGANIZATION プロシ  
ージャ, 7-20, 8-204  
END\_TABLE\_REORGANIZATION プロシージャ,  
7-20, 8-205  
I\_AM\_A\_REFRESH ファンクション, 8-206  
PURGE\_DIRECT\_LOAD\_LOG プロシージャ,  
8-206  
PURGE\_LOG プロシージャ, 7-19, 8-206  
PURGE\_SNAPSHOT\_FROM\_LOG プロシージャ,  
7-16, 8-207  
REFRESH\_ALL\_MVIEWS プロシージャ, 8-211  
REFRESH\_DEPENDENT プロシージャ, 8-212  
REFRESH プロシージャ, 8-209  
REGISTER\_SNAPSHOT プロシージャ, 8-214  
UNREGISTER\_SNAPSHOT プロシージャ, 7-16,  
8-215  
DDL, 「データ定義言語」を参照  
DEFCALLDEST ビュー, 9-43  
DEFCALL ビュー, 9-43  
DEFDEFAULTDEST ビュー, 9-43  
接続先の削除, 8-17  
接続先の追加, 8-16  
DEFERRCOUNT ビュー, 9-44  
DEFERROR ビュー, 7-26, 9-44  
トランザクションの削除, 8-18  
DEFLOB ビュー, 9-44  
DEFPROPAGATOR ビュー, 9-45  
DEFSCHEDULE ビュー, 9-45  
DEFTRANDEST ビュー, 9-46  
DEFTRAN ビュー, 9-46  
DROP SNAPSHOT LOG 文, 7-24

## G

---

GLOBAL\_NAMES 初期化パラメータ, 1-4

## J

---

JOB\_QUEUE\_INTERVAL 初期化パラメータ, 1-4  
JOB\_QUEUE\_PROCESSES 初期化パラメータ, 1-4

## P

---

PRESERVE SNAPSHOT LOG オプション  
TRUNCATE TABLE 文, 7-21  
PURGE SNAPSHOT LOG オプション  
TRUNCATE TABLE 文, 7-21

## S

---

set\_disabled, 8-30  
SNP バックグラウンド・プロセス, 1-4

## T

---

TRUNCATE TABLE 文  
PRESERVE SNAPSHOT LOG オプション, 7-21  
PURGE SNAPSHOT LOG オプション, 7-21  
TRUNCATE 文, 7-19

## U

---

USER\_REFRESH\_CHILDREN ビュー, 9-55  
USER\_REFRESH ビュー, 9-55  
USER\_REGISTERED\_SNAPSHOTS ビュー, 9-55  
USER\_REPCAT\_REFRESH\_TEMPLATES ビュー, 9-36  
USER\_REPCAT\_TEMPLATE\_OBJECTS ビュー, 9-36  
USER\_REPCAT\_TEMPLATE\_PARMS ビュー, 9-37  
USER\_REPCAT\_TEMPLATE\_SITES ビュー, 9-37  
USER\_REPCAT\_USER\_AUTHORIZATIONS ビュー,  
9-37  
USER\_REPCAT\_USER\_PARM\_VALUES ビュー, 9-37  
USER\_REPCATLOG ビュー, 9-36  
USER\_REPCOLUMN\_GROUP ビュー, 9-38  
USER\_REPCOLUMN ビュー, 9-38  
USER\_REPCONFLICT ビュー, 9-38  
USER\_REPDDL ビュー, 9-38  
USER\_REPGENOBJECTS ビュー, 9-38  
USER\_REPGROUP\_PRIVILEGES ビュー, 9-39

USER\_REPGROUPED\_COLUMN ビュー, 9-39  
USER\_REPGROUP ビュー, 9-38  
USER\_REPKEY\_COLUMNS ビュー, 9-39  
USER\_REPOBJECT ビュー, 9-39  
USER\_REPPARAMETER\_COLUMN ビュー, 9-39  
USER\_REPPRIORITY\_GROUP ビュー, 9-40  
USER\_REPPRIORITY ビュー, 9-40  
USER\_REPPROP ビュー, 9-40  
USER\_REPRESOL\_STATS\_CONTROL ビュー, 9-40  
USER\_REPRESOLUTION\_METHOD ビュー, 9-41  
USER\_REPRESOLUTION\_STATISTICS ビュー, 9-41  
USER\_REPRESOLUTION ビュー, 9-41  
USER\_REPSITES ビュー, 9-41  
USER\_SNAPSHOT\_LOGS ビュー, 9-56  
USER\_SNAPSHOT\_REFRESH\_TIMES ビュー, 9-56  
USER\_SNAPSHOTS ビュー, 9-55

## い

---

インスタンスエーション, 4-16  
    オフライン, 4-10  
    スクリプト, 4-14  
    リフレッシュ, 4-17  
インポート  
    オブジェクト・グループ  
        オフライン・インスタンスエーション, 8-35, 8-38  
    ステータス・チェック, 8-128  
    スナップショット  
        オフライン・インスタンスエーション, 8-42, 8-43

## え

---

エラー  
    エラー・キュー  
        DEFERROR ビュー, 7-26  
        管理, 7-26  
    エラー・トランザクション  
        受信者としての再実行, 7-26  
        代替ユーザーとしての再実行, 7-27

## お

---

オブジェクト  
    削除  
        スナップショット・サイト, 8-112

作成, 8-91  
    スナップショット・サイト, 8-95  
    マスター・グループ, 8-90  
    マスター・サイト, 8-91  
    スナップショット・サイトからの削除, 7-13  
    スナップショット・サイトへの追加, 8-95  
    変更, 7-27, 8-73  
    レプリケーション・サポートの生成, 8-116  
オフライン・インスタンスエーション  
    スナップショット, 8-42, 8-43  
    スナップショット・サイトの追加, 7-34  
    マスター・サイトの追加, 7-29  
    レプリケーション・サイトの追加, 7-29  
    レプリケート・オブジェクト・グループ, 8-34, 8-35, 8-37, 8-38, 8-39

## か

---

監査  
    競合解消, 6-24  
管理者  
    スナップショット・サイト  
        作成, 2-15  
        レプリケーション  
            作成, 2-4

## き

---

記憶域パラメータ  
    スナップショット・ログ  
        変更, 7-17

休止  
    マスター・グループ, 8-132

競合  
    回避  
        削除, 6-21  
    通知パッケージ  
        作成, B-7  
        サンプル, B-8  
    通知方法  
        ユーザー定義, B-6  
    通知ログ表  
        作成, B-6  
        例, B-7

競合解消, 6-1  
    DBA\_REPRESOLUTION\_STATISTICS ビュー, 6-25  
    上書きによる方法, 6-3

- 加算による方法, 6-9, 8-68
- 監査, 6-24
- 最小値による方法, 6-4
- 最大値による方法, 6-4
- サイトの優先順位, 6-13
- 準備, 6-2
- 情報
  - 表示, B-10
- 情報の表示, B-10
- タイムスタンプによる方法, 6-6
- 統計, 8-80, 8-124
  - 収集, 6-24
  - 取消し, 6-24
  - 表示, 6-24
- 廃棄による方法, 6-3
- 平均による方法, 6-9
- ユーザー定義方法, B-2
  - 一意性競合, B-3
  - 更新競合, B-3
  - 削除競合, B-4
  - 制限事項, B-4
  - パラメータ, B-2
  - 例, B-4
- 優先グループ, 6-11
- 列グループ, 6-3, 6-5, 6-8, 6-9, 6-11, 6-14

## こ

---

- コメント
  - 更新, 7-45
  - コメント・フィールド
    - ビュー内の更新, 7-45

## さ

---

- サイトの優先順位
  - 変更, 8-77
- サイト優先順位グループ
  - 削除, 8-109
  - 作成
    - 構文, 8-99
  - メンバーの削除, 8-110
  - メンバーの追加, 8-67

## し

---

- 主キー
  - レプリケート表, 3-6
- 受信者
  - 登録, 2-5
- 使用禁止
  - 伝播, 8-30
- ジョブ
  - キュー
    - ジョブの削除, 8-32
- ジョブ間隔, 1-5
- ジョブ・プロセス, 1-4

## す

---

- スケジュール・リンク
  - 作成, 2-12
- ステータス
  - 伝播, 8-19
- スナップショット
  - オフライン・インスタンスエーション, 8-42, 8-43
  - 削除, 7-14
  - サポートの生成, 8-117
  - スナップショット・ログ
    - 作成, 4-1, 5-4
  - セキュリティ, A-7
    - トラステッドとアントラステッド, A-8
  - データ・ディクショナリ・ビュー, 9-47
  - リフレッシュ, 4-17, 8-209, 8-211, 8-212
  - リフレッシュ・グループ
    - データ・ディクショナリ・ビュー, 9-47
- スナップショット・グループ
  - オブジェクトの追加, 5-6
  - グループの所有者, 7-5
  - 削除, 7-10
  - 作成, 5-4, 5-5, 8-94
- スナップショット・サイト
  - オブジェクトの削除, 7-13
  - オフライン・インスタンスエーションを使用した追加, 7-34
  - 管理, 7-5
  - 管理者
    - 作成, 2-15
  - グループの所有者
    - 使用, 7-5

- 削除, 7-10, 8-111
- スケジュール・ページ, 2-16
- スケジュール・プッシュ, 2-17
- 設定用フローチャート, 2-14
- データベース・リンク
  - 作成, 2-16, 5-4
- マスターの変更, 7-9, 8-133
- マスターへの変更の伝播, 8-29
- ユーザー
  - 作成, 2-15
- リフレッシュ
  - 作成, 2-15
- リフレッシュ, 8-122
- スナップショット・ログ
  - 管理, 7-17
    - 領域, 7-18
  - 行のページ, 7-18
  - 切捨て, 7-19
    - 必要な権限, 7-20
  - 削除, 7-24
  - 作成, 4-1, 5-4
  - ページ
    - 手動, 7-19
    - 必要な権限, 7-19
  - フィルタ列
    - 追加, 7-17
  - 変更, 7-17
    - 必要な権限, 7-17
  - マスター表
    - ページ, 8-206, 8-207
  - マスター表の切捨て, 7-21
  - マスター表の再編成, 7-20
  - 割り当てられた領域の削減, 7-19

## せ

---

- セキュリティ
  - スナップショット・レプリケーション, A-7
    - トラステッドとアントラステッド, A-8
  - トラステッドとアントラステッド, A-2, A-8
  - マルチマスター・レプリケーション, A-2
    - トラステッドとアントラステッド, A-2
  - レプリケーション, A-1

## ち

---

- 遅延トランザクション
  - DEFDEFAULTDEST ビュー
    - 接続先の削除, 8-17
    - 接続先の追加, 8-16
  - DefDefaultDest 表
    - 接続先の削除, 8-17
  - 開始, 8-7
  - キューからの削除, 8-18
  - 再実行, 8-21
  - 実行のスケジュール, 8-29
  - 遅延リモート・プロシージャ・コール (RPC)
    - 作成, 8-4
    - 即時実行, 8-24
    - 引数, 8-6
    - 引数値, 8-12
    - 引数のタイプ, 8-10
  - ビュー, 9-42
- 遅延トランザクション・キュー, 7-24
  - 管理, 7-24
    - 伝播されたトランザクションのページ, 7-25
- 遅延トランザクション・キューのページ
  - スナップショット・サイト, 2-16
  - マスター・サイト, 2-6
- 遅延トランザクションのプッシュ, 7-24
- 違い
  - 表, 8-45
    - 調整, 8-48
- 調整
  - 表, 7-41, 8-48

## つ

---

- 追加
  - リフレッシュ・グループへのオブジェクト, 5-7
- 通知パッケージ
  - 競合
    - 作成, B-7
- 通知方法
  - ユーザー定義, B-6
- 通知ログ表
  - 競合
    - 作成, B-6
    - サンプル, B-7

## て

---

- データ定義言語
  - 非同期, 8-114
  - 非同期の提供, 8-114
- データ・ディクショナリ・ビュー
  - スナップショット, 9-47
  - 遅延トランザクション, 9-42
  - レプリケーション, 9-1
- データベース・リンク, 5-4
  - 作成, 2-11, 2-16, 5-4
- 伝播
  - 使用禁止, 8-30
  - ステータス, 8-19
  - 変更
    - 伝播
      - 方法の変更, 8-71
    - 方法の変更, 8-79
- テンプレート, 「配置テンプレート」を参照
- テンプレート・ユーザーの認可, 4-9

## と

---

- 統計
  - 競合解消の監査, 6-24
  - 削除, 6-25
  - 収集, 8-124
  - パージ, 8-121
- 登録
  - ローカル・データベースのプロパゲータ, 8-27
- ドキュメント
  - 表記規則, xvi
- トラステッド・セキュリティ, A-2, A-8

## は

---

- パージ
  - DBA\_REPCATALOG 表, 8-120
  - 遅延トランザクション・キュー, 2-6, 7-25
- 配置テンプレート
  - インスタンスエーション・スクリプト, 4-14
  - インスタンスエート, 4-16
- オブジェクト
  - 削除, 8-186
  - 作成, 8-171
  - すべて削除, 8-179

- オブジェクトの追加, 4-5
- オブジェクトの変更, 8-158
- オフライン・インスタンスエーション, 4-10, 8-147, 8-149, 8-192, 8-194
- オンライン・インスタンスエーション, 8-151, 8-197
- 概要, 4-2
- 既存のオブジェクトからのオブジェクトの作成, 8-168
- サイト
  - 削除, 8-185
  - すべて削除, 8-181
- サイトのインスタンスエーションの削除, 8-146
- 削除, 8-184
- 作成, 4-2, 4-4
  - ユーザー・パラメータ値, 4-8
- 作成用フローチャート, 4-3
- スナップショット・グループの削除, 7-10
- すべて削除, 8-182
- データ・ディクショナリ・ビュー, 9-7
- テンプレートのコピー, 8-166
- テンプレートの作成, 8-169
- テンプレートの比較, 8-165
- テンプレートの変更, 8-156
- テンプレートのロック, 8-199
- パッケージ化, 4-10, 4-12
  - オフライン・インスタンスエーション, 4-12
  - オンライン・インスタンスエーション, 4-13
- パラメータ
  - 削除, 8-187
  - 作成, 4-7, 8-173
  - すべて削除, 8-180
  - ユーザー値, 4-8
- パラメータの変更, 8-160
- ファイルの配布, 4-15
- ユーザー認証
  - 削除, 8-188
  - 作成, 8-176
  - すべて削除, 8-182
- ユーザー認証の変更, 8-162
- ユーザーの認可, 4-9
- ユーザー・パラメータ値
  - 削除, 8-189
  - 作成, 8-177
  - すべて削除, 8-183
- ユーザー・パラメータ値の変更, 8-163

- ランタイム・パラメータ
  - ID の取得, 8-190
  - 削除, 8-179
  - 作成, 8-190
  - 挿入, 8-190

- 配置テンプレートのパッケージ化, 4-10

- パッケージ変数

- i\_am\_a\_refresh, 8-206

- パラメータ

- 配置テンプレート, 4-7

- ユーザー値, 4-8

## ひ

---

- 比較

- 表, 8-45

- ビュー

- スナップショット, 9-47

- 遅延トランザクション, 9-42

- リフレッシュ・グループ, 9-47

- レプリケーション・カタログ, 9-2

- 表

- コメントの更新, 7-45

- 相違, 7-41

- 調整, 7-41, 8-48

- 比較, 8-45

## ふ

---

- プッシュ

- 遅延トランザクション, 7-24

- スナップショット・サイト, 2-17

- プロキシ・スナップショット管理者

- 作成, 2-6

- プロバゲータ

- 登録, 2-5, 8-27

## へ

---

- 変更

- 伝播方法, 8-71, 8-79

## ま

---

- マスター・グループ

- オブジェクトの追加, 3-5

- 休止, 8-132

- 削除, 8-103

- 作成, 3-2, 3-5, 8-90

- 作成用フローチャート, 3-4

- レプリケーション・アクティビティの再開, 8-129

- マスター・サイト

- オフライン・インスタンスエーションを使用し  
た追加, 7-29

- 管理, 7-2

- 削除, 7-4, 8-126

- 作成, 8-64

- スケジュール・ページ, 2-6

- スケジュール・リンク, 2-12

- 設定, 2-4

- 設定用フローチャート, 2-3

- 相違の判別, 7-41

- 追加, 3-7, 7-3

- データベース・リンク, 2-11

- 変更の伝播, 8-29

- マスター定義サイトの変更, 7-2

- メタデータ

- クリーン・アップ, 7-14

- ユーザーの作成, 2-6

- マスター定義サイト

- 再配置, 8-125

- マスター表

- 切捨て, 7-21

- 再編成, 7-20

- 方法, 7-21

- マルチマスター・レプリケーション

- セキュリティ, A-2

- トラステッドとアントラステッド, A-2

## ゆ

---

- ユーザー

- スナップショット・サイト, 2-15

- 配置テンプレート用の認可, 4-9

- マスター・サイト用の作成, 2-6

- 優先グループ

- サイト優先順位グループ

- メンバーの追加, 8-67

- 削除, 8-107

- 作成, 8-98

- メンバーの削除, 8-106, 8-108

- メンバーの追加, 8-65



- メンバーの変更  
値, 8-75  
優先順位, 8-74

## り

---

- リフレッシャ  
作成, 2-15
- リフレッシュ  
スナップショット, 8-209, 8-211, 8-212  
スナップショット・サイト, 8-122
- リフレッシュ・グループ  
オブジェクトの追加, 5-7  
削除, 8-54  
作成, 8-55  
データ・ディクショナリ・ビュー, 9-47  
メンバーの削除, 8-58  
メンバーの追加, 8-51  
リフレッシュ  
手動, 8-57  
リフレッシュ間隔  
変更, 8-52
- 領域  
スナップショット・ログの削減, 7-19

## れ

---

- 列  
列グループ, 6-3, 6-5, 6-8, 6-9, 6-11, 6-14  
削除, 8-101  
作成, 8-97, 8-119  
メンバーの削除, 8-102  
メンバーの追加, 8-63
- レプリケーション  
エラー・キュー  
管理, 7-26  
オブジェクト  
配置テンプレートへの追加, 4-5  
変更, 6-6, 7-27  
マスター・グループへの追加, 3-5  
カタログ・ビュー, 9-1, 9-2  
環境の管理, 7-1  
環境の構築, 1-2  
管理者の作成, 2-4  
競合解消, 6-1  
再開, 3-9

- サイト  
設定, 2-2  
サイトの設定, 2-2  
サポートの生成, 3-8  
受信者  
登録, 2-5  
使用禁止, 8-200  
スケジュール・リンク  
作成, 2-12  
スナップショット・グループ  
作成, 5-4, 5-5  
スナップショット・サイト  
管理, 7-5  
スナップショット・ログ  
管理, 7-17  
セキュリティ, A-1  
遅延トランザクション・キュー  
管理, 7-24  
データ・ディクショナリ・ビュー, 9-1  
データベース・リンク  
作成, 2-11
- 表  
主キー, 3-6  
表間の相違の判別, 7-41  
プロパゲータ  
登録, 2-5  
マスター・グループ  
作成, 3-2  
マスター・サイト  
管理, 7-2  
追加, 3-7  
有効化, 8-201  
列グループ, 6-3, 6-5, 6-8, 6-9, 6-11, 6-14  
レプリケーション・アクティビティの再開, 8-129  
レプリケーション・カタログ  
コメント  
更新, 7-45  
レプリケーション・カタログ・ビュー, 9-1  
コメント  
更新, 7-45  
レプリケーション・サポートの生成, 3-8  
レプリケーション・マネジメント API, 8-1  
概要, 1-1  
競合解消, 6-1  
スナップショット・グループ  
作成, 5-4

- 配置テンプレート
  - インスタンシエート, 4-16
  - 作成, 4-2
  - パッケージ化, 4-10
- パッケージ, 8-2
- マスター・グループ
  - 作成, 3-2
  - 例, 8-2
- レプリケーション・サイトの設定, 2-2
- レプリケート環境の管理, 7-1
- レプリケート・オブジェクト
  - マスター・サイトからの削除, 8-105
- レプリケート・オブジェクト・グループ
  - オフライン・インスタンシエーション, 8-34, 8-35, 8-37, 8-38, 8-39