



異機種間複写ガイド

---

**SAP Replication Server® 15.7.1**

**SP200**

ドキュメント ID：DC20175-01-1571200-01

改訂：2014 年 4 月

Copyright © 2014 by SAP AG or an SAP affiliate company. All rights reserved.

このマニュアルの内容を SAP AG の明示的許可を得ずに、いかなる手段によっても、複製、転載することを禁じます。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダー独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行わないものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。その他の商標に関する情報および通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> を参照してください。

# 目次

|   |    |
|---|----|
| 表記の規則 .....                                   | 1  |
| 複製システムの概要 .....                               | 5  |
| 基本的な複製システム .....                              | 5  |
| 異機種間複製システム .....                              | 6  |
| SAP 複製システムのコンポーネント .....                      | 8  |
| プライマリデータサーバ .....                             | 8  |
| Replication Agent .....                       | 9  |
| Replication Server .....                      | 9  |
| Replication Server システムデータベース<br>(RSSD) ..... | 11 |
| データベースゲートウェイ .....                            | 13 |
| ExpressConnect for Oracle .....               | 13 |
| ExpressConnect for SAP HANA データベース ..         | 13 |
| レプリケートデータサーバ .....                            | 14 |
| ASE 以外の複製 .....                               | 15 |
| プライマリデータベース .....                             | 15 |
| レプリケートデータベース .....                            | 16 |
| 文字セット .....                                   | 17 |
| 異機種間複製の制限事項 .....                             | 17 |
| ストアドプロシージャの複製 .....                           | 17 |
| 所有者指定のオブジェクト名 .....                           | 18 |
| ラージオブジェクトの複製 .....                            | 18 |
| レプリケートデータベースの設定 .....                         | 19 |
| Replication Server の暗号化カラムのサポート ..            | 20 |
| サブスクリプションマテリアライゼーション ..                       | 20 |
| Replication Server の rs_dump コマンド .....       | 21 |
| Replication Server rs_marker コマンド .....       | 21 |
| Replication Server の rs_dumptran コマンド .....   | 22 |

|   |           |
|---|-----------|
| Replication Server の rs_subcmp ユーティリ<br>ティ .....    | 22        |
| 動的 SQL .....  | 23        |
| バルクコピー .....  | 23        |
| Replication Server rs_ticket ストアドプロシー<br>ジャ .....   | 23        |
| 複製システムの ASE 以外の設定 .....                             | 23        |
| ASE 以外のプライマリから Adaptive Server レ<br>プリケートへの複製 ..... | 23        |
| ASE サーバのプライマリから ASE 以外のサー<br>バのレプリケートへの複製 .....     | 24        |
| ASE 以外のプライマリから ASE 以外のレプリ<br>ケートへの複製 .....          | 25        |
| ASE 以外から ASE 以外への双方向複製 .....                        | 26        |
| <b>SAP の複製製品 .....</b>                              | <b>29</b> |
| Replication Server .....                            | 29        |
| Replication Server の動作 .....                        | 30        |
| パブリッシュとサブスクライブモデル .....                             | 31        |
| 複製ファンクション .....                                     | 31        |
| トランザクション管理 .....                                    | 32        |
| 他のシステムコンポーネントとの関係 .....                             | 33        |
| データベースコネクション .....                                  | 37        |
| DDL ユーザの目的 .....                                    | 40        |
| テーブルレベルの引用符付き識別子の複製 .....                           | 40        |
| データ型、データ型定義、および制限付き<br>データ型 .....                   | 42        |
| ASE 以外のデータサーバのエラークラス .....                          | 42        |
| ASE 以外のデータサーバのファンクション文<br>文字列クラス .....              | 42        |
| オブジェクトのパブリケーションとサブスク<br>リプションの制限 .....              | 45        |
| Replication Agent .....                             | 45        |

|   |           |
|---|-----------|
| Replication Agent の動作 .....   | 45        |
| DDL ユーザの処理 .....  | 48        |
| ASE 以外の Replication Agent .....                                       | 49        |
| Enterprise Connect™ Data Access .....                                 | 50        |
| ECDA の動作 .....  | 50        |
| ECDA データベースゲートウェイ .....   | 52        |
| ECDA Option for ODBC .....  | 53        |
| Mainframe Connect DirectConnect for z/OS<br>Option .....              | 53        |
| ExpressConnect for Oracle .....                                       | 54        |
| ExpressConnect for SAP HANA データベース .....                              | 55        |
| <b>プライマリデータサーバとしての IBM DB2 for z/OS .....</b>                         | <b>57</b> |
| Replication Agent for DB2 UDB .....                                   | 57        |
| 複写の干渉と影響 .....  | 57        |
| DB2 UDB プライマリデータベースのパーミッション<br>.....                                  | 58        |
| プライマリデータサーバの接続 .....  | 59        |
| Replication Server の接続 .....  | 59        |
| Replication Server システムデータベースの接続 .....                                | 59        |
| DB2 UDB プライマリデータベースの設定 .....  | 60        |
| DB2 for z/OS におけるプライマリテーブルの複写定<br>義 .....                             | 61        |
| DB2 for z/OS プライマリデータ型の変換 .....                                       | 62        |
| 文字セット .....   | 62        |
| マテリアライゼーション .....   | 63        |
| <b>プライマリデータサーバとしての IBM DB2 for Linux,<br/>UNIX, および Windows .....</b> | <b>65</b> |
| Replication Agent for UDB .....                                       | 65        |
| DB2 UDB システムの管理 .....   | 65        |
| DB2 UDB での複写の干渉と影響 .....  | 66        |
| DB2 UDB プライマリデータベースのパーミッション<br>と制限 .....                              | 66        |

|  |           |
|--|-----------|
| プライマリデータサーバの接続 .....                                 | 66        |
| Replication Server と RSSD のコネクティビティ .....            | 67        |
| Replication Agent オブジェクト .....                       | 67        |
| トランケーション用の Java プロシージャ .....                         | 68        |
| 複製オブジェクトの実際の名前の取得 .....                              | 68        |
| DB2 UDB プライマリデータベースの設定 .....                         | 69        |
| Java Runtime Environment .....                       | 69        |
| rs_source_ds 設定パラメータと rs_source_db<br>設定パラメータ .....  | 69        |
| filter_maint_userid 設定パラメータ .....                    | 69        |
| lfl_character_case 設定パラメータ .....                     | 70        |
| 複製定義での大文字と小文字および引用符付<br>き識別子の設定 .....                | 70        |
| 大文字で格納されるオブジェクト名 .....                               | 70        |
| DB2 UDB におけるプライマリテーブルの複製定義 .....                     | 71        |
| DB2 UDB プライマリデータ型の変換 .....                           | 71        |
| <b>プライマリデータサーバとしての Microsoft SQL Server</b><br>..... | <b>73</b> |
| Replication Agent for Microsoft SQL Server .....     | 73        |
| sybfilter ドライバ .....                                 | 73        |
| Microsoft SQL Server システムの管理 .....                   | 74        |
| Replication Agent のパーミッション .....                     | 74        |
| プライマリデータサーバの接続 .....                                 | 74        |
| CLASSPATH 環境変数の設定 .....                              | 74        |
| Replication Server と RSSD のコネクティビティ .....            | 75        |
| Replication Agent オブジェクト .....                       | 75        |
| テーブル、プロシージャ、マーカ、およびト<br>リガオブジェクト .....               | 76        |
| Microsoft SQL Server プライマリデータベースの設<br>定 .....        | 76        |
| rs_source_ds 設定パラメータと rs_source_db<br>設定パラメータ .....  | 77        |

|   |           |
|---|-----------|
| filter_maint_userid 設定パラメータ .....                   | 77        |
| ltl_character_case 設定パラメータ .....                    | 77        |
| Microsoft SQL Server におけるプライマリテーブル<br>の複写定義 .....   | 78        |
| Microsoft SQL Server プライマリデータ型の変換 .....             | 78        |
| Microsoft SQL Server の大きな値のデータ型の複写 .....            | 79        |
| Microsoft SQL Server の大きな値のデータ型の<br>制限 .....        | 79        |
| Microsoft SQL Server の大きな値のデータ型の<br>部分更新 .....      | 80        |
| <b>プライマリデータサーバとしての Oracle .....</b>                 | <b>81</b> |
| Replication Agent for Oracle .....                  | 81        |
| Oracle におけるプライマリテーブルの複写定<br>義 .....                 | 81        |
| Oracle システムの管理 .....                                | 82        |
| Oracle での複写の干渉と影響 .....                             | 82        |
| Oracle から SAP HANA データベースへの連続複写<br>の制限 .....        | 82        |
| Oracle プライマリデータベースのパーミッション .....                    | 83        |
| プライマリデータサーバの接続 .....                                | 83        |
| Replication Server と RSSD のコネクティビティ .....           | 83        |
| Replication Agent オブジェクト .....                      | 84        |
| Oracle プライマリデータベースの設定 .....                         | 84        |
| Java Runtime Environment .....                      | 85        |
| 必要な JDBC ドライバ .....                                 | 85        |
| rs_source_ds 設定パラメータと rs_source_db<br>設定パラメータ ..... | 85        |
| filter_maint_userid 設定パラメータ .....                   | 85        |
| ltl_character_case 設定パラメータ .....                    | 86        |
| 複写定義での大文字と小文字および引用符付<br>き識別子の設定 .....               | 86        |
| Oracle プライマリデータ型の変換 .....                           | 86        |

|   |            |
|---|------------|
| Automatic Storage Management .....                      | 87         |
| Real Application Cluster .....                          | 87         |
| <b>プライマリデータサーバとしての SAP Business Suite</b>               |            |
| データベース .....  | <b>89</b>  |
| SAP Business Suite データベースの複写 .....                      | 89         |
| クラスターテーブルの複写 .....                                      | 90         |
| SAP Business Suite データベースのマテリアライ<br>ゼーションと複写のシナリオ ..... | 91         |
| シナリオ 1 .....  | 91         |
| シナリオ 2 .....  | 94         |
| <b>レプリケートデータサーバとしての IBM DB2 for z/OS .....</b>          | <b>99</b>  |
| DB2 UDB for z/OS レプリケートデータサーバの環境<br>.....               | 99         |
| DB2 UDB for z/OS システム管理 .....                           | 99         |
| DB2 UDB for z/OS での複写の干渉と影響 .....                       | 99         |
| DB2 for z/OS レプリケートデータベースのパーミッ<br>ション .....             | 101        |
| DB2 UDB for z/OS に関するレプリケートデータベ<br>ースのコネクティビティ .....    | 101        |
| DB2 for z/OS におけるレプリケートデータベースの<br>制限事項 .....            | 102        |
| DB2 for z/OS レプリケートデータベースの設定 .....                      | 102        |
| Replication Server インストール .....                         | 103        |
| 接続プロファイル .....  | 103        |
| 追加設定 .....  | 105        |
| <b>Replicate Data Server としての Linux、UNIX、および</b>        |            |
| <b>Windows に関する IBM DB2 .....</b>                       | <b>107</b> |
| DB2 UDB レプリケートデータサーバ .....                              | 107        |
| DB2 UDB での複写の干渉と影響 .....                                | 107        |
| DB2 UDB レプリケートデータベースのパーミッシ<br>ョンと制限 .....               | 108        |



|  |            |
|--|------------|
| DB2 UDB レプリケートデータベースのコネクティビ<br>ティ .....                  | 109        |
| DB2 UDB レプリケートデータベースの設定 .....                            | 109        |
| Replication Server インストール .....                          | 110        |
| 接続プロファイル .....   | 110        |
| 追加設定 .....   | 112        |
| IBM DB2 レプリケートデータベースの並列 DSI ス<br>レッド .....               | 113        |
| 外部コミット制御 .....   | 113        |
| 内部コミット制御 .....   | 114        |
| トランザクションの逐次化メソッド .....                                   | 114        |
| レプリケートデータサーバとしての <b>Microsoft SQL<br/>Server</b> .....   | <b>119</b> |
| Microsoft SQL Server レプリケートデータサーバ .....                  | 119        |
| Microsoft SQL Server での複写の干渉と影響 .....                    | 119        |
| Microsoft SQL Server におけるレプリケートデータ<br>ベースの制限事項 .....     | 120        |
| Microsoft SQL Server レプリケートデータベースの<br>パーミッション .....      | 122        |
| Microsoft SQL Server に関するレプリケートデータ<br>ベースのコネクティビティ ..... | 122        |
| Microsoft SQL Server レプリケートデータベースの<br>設定 .....           | 123        |
| Replication Server インストール .....                          | 123        |
| 接続プロファイル .....   | 124        |
| 追加設定 .....   | 125        |
| Microsoft SQL Server レプリケートデータベースの<br>並列 DSI スレッド .....  | 127        |
| 外部および内部のコミット制御 .....                                     | 127        |
| トランザクションの逐次化メソッド .....                                   | 128        |
| レプリケートデータサーバとしての <b>Oracle</b> .....                     | <b>133</b> |
| Oracle レプリケートデータサーバ .....                                | 133        |

|  |            |
|--|------------|
| Oracle での複写の干渉と影響 .....                    | 133        |
| Oracle レプリケートデータベースのパーミッション .....          | 134        |
| Oracle に関するレプリケートデータベースのコネク<br>ティビティ ..... | 135        |
| Oracle レプリケートデータベースの設定 .....               | 135        |
| Replication Server インストール .....            | 136        |
| 接続プロファイル .....                             | 137        |
| 追加設定 .....                                 | 138        |
| Oracle レプリケートデータベースの並列 DSI スレッ<br>ド .....  | 142        |
| 外部および内部のコミット制御 .....                       | 143        |
| トランザクションの逐次化メソッド .....                     | 143        |
| <b>レプリケートデータサーバとしての SAP IQ .....</b>       | <b>145</b> |
| Real-Time Loading ソリューション .....            | 145        |
| RTL のコンパイルとバルク適用 .....                     | 147        |
| 最終的な変更のデータベース .....                        | 149        |
| RTL の処理と制限事項 .....                         | 149        |
| SAP IQ レプリケートデータサーバ .....                  | 152        |
| SAP IQ での複写の干渉と影響 .....                    | 152        |
| SAP IQ に関するレプリケートデータベースのコネク<br>ティビティ ..... | 154        |
| SAP IQ レプリケートデータベースのパーミッション<br>.....       | 155        |
| メンテナンスユーザ ID に権限を与える .....                 | 155        |
| SAP IQ レプリケートデータベースの設定 .....               | 156        |
| Replication Server インストール .....            | 156        |
| RTL を有効にする .....                           | 158        |
| RTL パフォーマンスのチューニング .....                   | 159        |
| リトライメカニズムの強化 .....                         | 162        |
| メモリ消費の制御 .....                             | 163        |
| SAP IQ への Multi-Path Replication .....     | 167        |

|  |     |
|--|-----|
| SAP IQ への代替レプリケートコネクションの<br>作成 .....             | 167 |
| 代替レプリケート SAP IQ コネクションの変更<br>または削除 .....         | 169 |
| レプリケートコネクションに関する情報の表<br>示 .....                  | 169 |
| レプリケーションロード分散統計 .....                            | 170 |
| 分散モデルの設定 .....                                   | 171 |
| 参照制約のあるテーブル .....                                | 171 |
| 複写定義の作成と変更 .....                                 | 172 |
| RTL 情報の表示 .....                                  | 173 |
| Replication Server 15.5 のシステムテーブルサポート<br>.....   | 174 |
| 混合バージョンのサポートと下位互換性 .....                         | 174 |
| SAP IQ への複写シナリオ .....                            | 175 |
| interfaces ファイルのエントリの作成 .....                    | 175 |
| テストテーブルの作成 .....                                 | 176 |
| プライマリデータベースとレプリケートデー<br>タベースへのコネクションの作成 .....    | 176 |
| RTL の有効化 .....                                   | 177 |
| 複写テスト準備のためのテーブルへのマーク<br>付け .....                 | 178 |
| 複写定義とサブスクリプションの作成 .....                          | 179 |
| RTL が機能することを検証する .....                           | 180 |
| ステージングソリューションから RTL へのマイグ<br>レート .....           | 180 |
| ステージングソリューションからのマイグ<br>レートの準備 .....              | 181 |
| Real-Time Loading ソリューションにマイグ<br>レート .....       | 182 |
| マイグレーション後のクリーンアップ .....                          | 183 |
| Replication Server と SAP IQ InfoPrimer の統合 ..... | 183 |

|   |            |
|---|------------|
| Replication Server と SAP IQ InfoPrimer の統合<br>の使用 .....                                   | 184        |
| パラメータ .....   | 190        |
| Replication Server のコンポーネント .....   | 191        |
| デフォルトのデータ型変換 .....  | 193        |
| サポートされない機能 .....  | 193        |
| <b>レプリケートデータサーバとしての SAP HANA データ<br/>ベース .....</b>  | <b>195</b> |
| SAP HANA データベースのレプリケートデータサー<br>バ .....  | 195        |
| SAP HANA データベースでの複写の干渉と影響 .....   | 195        |
| プライマリキーのデータ型としての Oracle<br>RAW および LONG RAW .....   | 196        |
| Oracle の TIMESTAMP WITH TIME ZONE デー<br>タ型のマッピング .....                                    | 197        |
| Oracle から SAP HANA データベースへの連続<br>複写の制限 .....  | 197        |
| Identity カラム .....  | 197        |
| SAP HANA データベースのレプリケートデータベー<br>スのパーミッション。 .....   | 197        |
| 異機種間複写環境における DDL 複写 .....   | 198        |
| テーブル複写定義の排除 .....   | 199        |
| ユニークキーがないテーブルの LOB カラムの複写 ...   | 199        |
| ExpressConnect for SAP HANA データベースと SAP<br>HANA データベースに対するレプリケートデータ<br>ベースのコネクティビティ ..... | 200        |
| ExpressConnect for SAP HANA データベース<br>の設定 .....   | 201        |
| ExpressConnect for SAP HANA データベース<br>のライセンス .....  | 202        |
| トレースとデバッグ .....   | 202        |
| 文字セットの変換 .....  | 203        |

|   |            |
|---|------------|
| LOB ポインタファンクション .....   | 203        |
| SAP HANA データベースのレプリケートデータベ<br>ースの設定 .....                             | 203        |
| Replication Server インストール .....                                       | 204        |
| 接続プロファイル .....  | 204        |
| 追加設定 .....  | 207        |
| 複写定義での大文字と小文字および引用符付<br>き識別子の設定 .....                                 | 208        |
| <b>異機種間における Multi-Path Replication .....</b>                          | <b>209</b> |
| 並列トランザクションストリーム .....   | 211        |
| デフォルトおよび代替接続 .....  | 211        |
| SAP IQ に関する Interfaces ファイル要件 .....                                   | 212        |
| 専用ルート .....   | 212        |
| 専用ルートの作成 .....  | 213        |
| 専用ルートを管理するコマンド .....  | 213        |
| 専用ルート情報の表示 .....  | 216        |
| 異機種間 Multi-Path Replication のシナリオ .....                               | 216        |
| Adaptive Server から SAP HANA データベース<br>への Multi-Path Replication ..... | 217        |
| Adaptive Server から Oracle への Multi-Path<br>Replication .....          | 222        |
| Adaptive Server から SAP IQ への Multi-Path<br>Replication .....          | 226        |
| Oracle から Adaptive Server への Multi-Path<br>Replication .....          | 230        |
| Oracle から SAP HANA データベースへの<br>Multi-Path Replication .....           | 234        |
| Oracle から Oracle への Multi-Path Replication<br>.....                   | 239        |
| Oracle から SAP IQ への Multi-Path Replication<br>.....                   | 244        |
| <b>Oracle に対する異機種ウォームスタンバイ .....</b>                                  | <b>249</b> |

|  |     |
|--|-----|
| Oracle に対するウォームスタンバイの動作 .....                                    | 249 |
| ウォームスタンバイアプリケーション .....  | 250 |
| ウォームスタンバイの要件と制限 .....  | 250 |
| スタンバイデータベースを管理するためのファンク<br>ション文字列 .....                          | 251 |
| ウォームスタンバイアプリケーションの複写情報 .....                                     | 251 |
| ウォームスタンバイデータベースの設定 .....   | 252 |
| 論理コネクションの作成 .....  | 253 |
| アクティブデータベースへの ReplicationAgent<br>の初期化 .....                     | 254 |
| 複写システムへのアクティブデータベースの<br>追加 .....                                 | 256 |
| スタンバイデータベースの初期化 .....  | 257 |
| スタンバイデータベースへの ReplicationAgent<br>の初期化 .....                     | 257 |
| スタンバイデータベースへのコネクションの<br>作成 .....                                 | 259 |
| アクティブデータベースとスタンバイデータ<br>ベースへのコネクションのレジューム .....                  | 259 |
| アクティブデータベースとスタンバイデータ<br>ベースへの Replication Agents のレジューム<br>..... | 260 |
| アクティブデータベースとスタンバイデータベース<br>の切り替え .....                           | 260 |
| アクティブデータベースとスタンバイデータ<br>ベースを切り替える前 .....                         | 261 |
| 内部での切り替え手順 .....   | 262 |
| アクティブデータベースとスタンバイデータ<br>ベースを切り替えたあと .....                        | 263 |
| ウォームスタンバイアプリケーションのモニタリン<br>グ .....                               | 263 |
| 複写定義およびサブスクリプション .....   | 264 |

|   |            |
|---|------------|
| 追加のウォームスタンバイデータベースの複<br>写定義 .....                 | 264        |
| ウォームスタンバイアプリケーションでのサ<br>ブスクリプション .....            | 265        |
| アップグレードの考慮事項 .....                                | 266        |
| ダウングレードの考慮事項 .....                                | 266        |
| ダウングレード後の複写のレジューム .....                           | 266        |
| <b>Oracle レプリケートデータベースの再同期 .....</b>              | <b>267</b> |
| 製品の互換性 .....                                      | 267        |
| データベースの再同期の設定 .....                               | 267        |
| Replication Server にトランザクションをス<br>キップさせる .....    | 268        |
| データベース再同期マーカを Replication<br>Server に送信する .....   | 269        |
| データベースのダンプを取得する .....                             | 270        |
| ダンプデータベースマーカを Replication<br>Server に送信する .....   | 272        |
| DSI スレッド情報をモニタする .....                            | 273        |
| 再同期するデータベースにダンプを適用する ..                           | 273        |
| レプリケートデータベースの再初期化 .....                           | 273        |
| データベース再同期化シナリオ .....                              | 277        |
| 1 つ以上のレプリケートデータベースをプライ<br>マリデータベースから直接再同期する ..... | 278        |
| サードパーティダンプユーティリティを使用<br>して再同期する .....             | 280        |
| プライマリデータベースとレプリケートデー<br>タベースを同じダンプから再同期 .....     | 282        |
| <b>データ型の変換とマッピング .....</b>                        | <b>285</b> |
| 同機種マッピング .....                                    | 285        |
| 異機種マッピング .....                                    | 286        |
| DB2 データ型 .....                                    | 287        |

|   |     |
|---|-----|
| Adaptive Server データ型から DB2 データ型へ<br>の変換 .....                 | 287 |
| DB2 データ型から Adaptive Server データ型へ<br>の変換 .....                 | 288 |
| DB2 から SAP HANA データベース .....                                  | 289 |
| DB2 データ型から Microsoft SQL Server データ<br>型への変換 .....            | 290 |
| DB2 データ型から Oracle データ型への変換 ...                                | 291 |
| DB2 に対応する Replication Server のデータ型<br>名 .....                 | 291 |
| Microsoft SQL Server データ型 .....                               | 292 |
| Adaptive Server データ型から Microsoft SQL<br>Server データ型への変換 ..... | 292 |
| Microsoft SQL Server データ型から DB2 データ<br>型への変換 .....            | 293 |
| Microsoft SQL Server から SAP HANA データ<br>ベース .....             | 293 |
| Microsoft SQL Server データ型から Oracle デー<br>タ型への変換 .....         | 296 |
| Microsoft SQL Server に対応する Replication<br>Server のデータ型名 ..... | 296 |
| Oracle データ型 .....   | 296 |
| Adaptive Server データ型から Oracle データ型<br>への変換 .....              | 297 |
| Oracle データ型から Adaptive Server データ型<br>への変換 .....              | 297 |
| Oracle データ型から DB2 データ型への変換 ...                                | 298 |
| Oracle から SAP HANA データベース .....                               | 298 |
| Oracle データ型から Microsoft SQL Server デー<br>タ型への変換 .....         | 301 |
| Oracle に対応する Replication Server のデータ<br>型名 .....              | 301 |



|  |            |
|--|------------|
| SAP HANA データベースデータ型 .....  | 302        |
| Adaptive Server から SAP HANA データベース<br>へのデータ型 .....                         | 302        |
| <b>マテリアライゼーション .....</b>   | <b>305</b> |
| マテリアライゼーションの種類 .....   | 305        |
| 異機種マテリアライゼーション .....   | 305        |
| バルクマテリアライゼーションのオプション .....   | 306        |
| プライマリデータベースからのデータのアンロード<br>.....   | 307        |
| データ型変換 .....   | 307        |
| レプリケートデータベースへのデータのロード .....  | 308        |
| アトミックバルクマテリアライゼーション .....  | 308        |
| マテリアライゼーションの準備 .....   | 308        |
| アトミックバルクマテリアライゼーションの<br>実行 .....   | 309        |
| ノンアトミックバルクマテリアライゼーション .....  | 311        |
| マテリアライゼーションの準備 .....   | 311        |
| ノンアトミックバルクマテリアライゼーショ<br>ンの実行 .....   | 312        |
| オートコレクション .....  | 314        |
| 直接ロードマテリアライゼーション .....   | 315        |
| サブスクリプションと直接ロードマテリアラ<br>イゼーション .....                                       | 317        |
| Adaptive Server から SAP HANA データベース<br>への直接ロードマテリアライゼーション .....             | 318        |
| Adaptive Server 以外のデータベースから SAP<br>HANA データベースへの直接ロードマテリ<br>アライゼーション ..... | 319        |
| 直接ロードマテリアライゼーションの設定パ<br>ラメータ .....   | 322        |
| <b>異機種データベースの調整 .....</b>  | <b>323</b> |
| SAP の rs_subcmp ユーティリティ .....  | 323        |

|  |            |
|--|------------|
| SAP Replication Server データ保証オプション .....            | 323        |
| データベース比較アプリケーション .....                             | 324        |
| <b>異機種間複写システムのトラブルシューティング .....</b>                | <b>327</b> |
| インバウンドキューの問題 .....                                 | 327        |
| インバウンドキューが更新されていない原因<br>の特定 .....                  | 327        |
| アウトバウンドキューの問題 .....                                | 329        |
| アウトバウンドキューが更新されていない原<br>因の特定 .....                 | 330        |
| レプリケートデータベースがなぜ更新されないかを<br>特定します。 .....            | 330        |
| HDS の問題と制限事項 .....                                 | 332        |
| 変換先のデータ型の範囲を超える変換元の値<br>.....                      | 332        |
| 真数値データ型の問題 .....                                   | 333        |
| Microsoft SQL Server での数値変換と identity<br>カラム ..... | 335        |
| 特定のエラーのトラブルシューティング .....                           | 335        |
| rs_lastcommit が更新されない .....                        | 336        |
| 想定したデータ型変換が行われない .....                             | 336        |
| カラム値の欠落 .....                                      | 338        |
| ログ転送言語の生成とトレース .....                               | 339        |
| <b>Oracle から Oracle への複製のリファレンス実装 .....</b>        | <b>343</b> |
| プラットフォームのサポート .....                                | 343        |
| Oracle のリファレンス実装のコンポーネント .....                     | 344        |
| リファレンス環境の前提条件の確認 .....                             | 344        |
| Oracle 用のリファレンス実装設定ファイル .....                      | 345        |
| 用語解説 .....   | 347        |
| 索引 .....   | 367        |

## 表記の規則

ここでは、SAP® マニュアルで使用しているスタイルおよび構文の表記規則について説明します。

### 表記の規則

| 構文要素              | 定義  |
|-------------------|---|
| 等幅 (固定幅)          | <ul style="list-style-type: none"> <li>SQL およびプログラムコード</li> <li>表示されたとおりに入力する必要のあるコマンド</li> <li>ファイル名</li> <li>ディレクトリ名</li> </ul>   |
| 斜体等幅              | SQL またはプログラムコードのスニペット内では、ユーザ指定の値のプレースホルダ (以下の例を参照)  |
| 斜体                | <ul style="list-style-type: none"> <li>ファイルおよび変数の名前</li> <li>他のトピックまたはマニュアルとの相互参照</li> <li>本文中では、ユーザ指定の値のプレースホルダ (以下の例を参照)</li> <li>用語解説に含まれているテキスト内の用語</li> </ul>  |
| 太字体<br>sans-serif | <ul style="list-style-type: none"> <li>コマンド、関数、ストアドプロシージャ、ユーティリティ、クラス、メソッドの名前</li> <li>用語解説のエントリ (用語解説内)</li> <li>メニューオプションのパス</li> <li>番号付きの作業または手順内では、クリックの対象となるボタン、チェックボックス、アイコンなどのユーザインタフェース (UI) 要素</li> </ul> |

必要に応じて、プレースホルダ (システムまたは設定固有の値) の説明が本文中に追加されます。次に例を示します。

次のコマンドを実行します。

```
installation directory/start.bat
```

*installation directory* はアプリケーションがインストールされた場所です。

構文の表記規則

| 構文要素 | 定義   |
|------|--|
| { }  | 中カッコで囲まれたオプションの中から必ず 1 つ以上を選択する。コマンドには中カッコは入力しない。                    |
| [ ]  | 角カッコは、オプションを選択しても省略してもよいことを意味する。コマンドには角カッコは入力しない。                    |
| ( )  | このカッコはコマンドの一部として入力する。  |
|      | 縦線はオプションのうち 1 つのみを選択できることを意味する。                                      |
| ,    | カンマは、表示されているオプションを必要な数だけ選択でき、選択したものをコマンドの一部として入力するときにカンマで区切ることを意味する。 |
| ...  | 省略記号 (...) は、直前の要素を必要な回数だけ繰り返し指定できることを意味する。省略記号はコマンドには入力しない。         |

大文字と小文字の区別

- すべてのコマンド構文およびコマンドの例は、小文字で表記しています。ただし、複写コマンド名では、大文字と小文字が区別されません。たとえば、**RA\_CONFIG**、**Ra\_Config**、**ra\_config** は、すべて同じです。
- 設定パラメータの名前では、大文字と小文字が区別されます。たとえば、**Scan\_Sleep\_Max** は、**scan\_sleep\_max** とは異なり、パラメータ名としては無効になります。
- データベースオブジェクト名は、複写コマンド内では、大文字と小文字が区別されません。ただし、複写コマンドで大文字と小文字が混在したオブジェクト名を使用する場合 (プライマリデータベースの大文字と小文字が混在したオブジェクト名と一致させる場合)、引用符でオブジェクト名を区切ります。次に例を示します。 **pdb\_get\_tables "TableName"**
- 識別子および文字データでは、使用しているソート順によっては大文字と小文字が区別されます。
  - “binary” などの大文字と小文字を区別するソート順を使用する場合には、識別子や文字データは、大文字と小文字を正しく入力してください。
  - “nocase” などの大文字と小文字を区別しないソート順を使用する場合には、識別子や文字データは、大文字と小文字をどのような組み合わせでも入力できます。

用語

SAP® Replication Server® はさまざまなコンポーネントと連携して、SAP Adaptive Server Enterprise (SAP ASE)、SAP HANA® データベース、SAP® IQ、Oracle、IBM

DB2 UDB、Microsoft SQL Server など、サポートされているデータベース間の複製を実現します。SAP Replication Server では SAP ASE を Replication Server システムデータベース (RSSD) に使用します。または、SAP® SQL Anywhere® を Embedded Replication Server システムデータベース (ERSSD) に使用します。

Replication Agent™ は、SAP ASE、SAP HANA データベース、Oracle、IBM DB2 UDB、Microsoft SQL Server 用の Replication Agent を表現するために使用される一般的な用語です。具体的な名前は、次のとおりです。

- RepAgent - SAP ASE 用の Replication Agent スレッド
- Replication Agent for Oracle
- Replication Agent for Microsoft SQL Server
- Replication Agent for UDB – Linux、Unix、Windows 用の IBM DB2
- Replication Agent for DB2 for z/OS



# 複製システムの概要

SAP® は、SAP® Adaptive Server® Enterprise (SAP® ASE) サーバから別の ASE サーバへの基本的な複製システム、および 1 つ以上のサーバが ASE 以外の場合の異機種間複製システムをサポートします。

## 基本的な複製システム

基本的な SAP 複製システムは、プライマリ SAP ASE データベース、SAP Replication Server®、レプリケート SAP ASE データベースで構成されます。

SAP ASE には、SAP 複製システムをサポートするために必要な機能すべてが備わっており、SAP Replication Server 以外の追加コンポーネントは必要ありません。

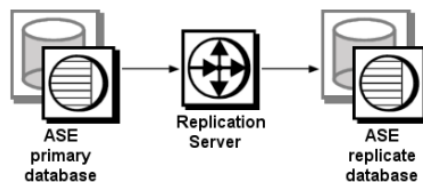
次の 3 つのコンポーネントが含まれています。

- プライマリデータベース-元になるデータ変更オペレーション(トランザクション)が実行されるデータベース。完了したトランザクションだけが複製用に取り出される。
- *SAP Replication Server* 複製されるトランザクションをプライマリデータベースから受け取り、レプリケートデータベースに配信する Open Client™ および SAP® Open Server™ で構成された製品。
- レプリケートデータベース-複製トランザクションを Replication Server から受け取り、プライマリデータの独自の「コピー」にそのトランザクションを適用するデータベース。

複製システムの管理には、SAP® Control Center などのシステム管理ツールを使用できます。

基本的な SAP 複製システムの図は、2 台の Adaptive Server と 1 台の Replication Server 間のデータの流れを示しています。

図 1 : 基本的な SAP 複製システム



## 複写システムの概要

データは、プライマリデータベースから Replication Server へ流れてから、レプリケートデータベースへと流れます。

基本的な SAP 複写システムの概念と Replication Server の機能の詳細については、『Replication Server デザインガイド』と、『Replication Server 管理ガイド 第 1 巻』の「Replication Server の概要」を参照してください。

## 異機種間複写システム

---

異機種 SAP 複写システムは、同一または異なるベンダの 2 つのデータベース間 (ASE 間以外) のデータ変更オペレーションで構成されます。

ASE から ASE レプリケーションの詳細については、『Replication Server 管理ガイド 第 1 巻』および『Replication Server 管理ガイド 第 2 巻』を参照してください。

異機種間複写には次が含まれます。

- Adaptive Server Enterprise (ASE) がプライマリデータサーバまたはレプリケートデータサーバであり、ASE 以外のデータサーバ (IBM DB2 UDB など) がもう一方のデータサーバである複写システム
- プライマリデータサーバとレプリケートデータサーバの両方が ASE 以外のデータサーバである複写システム (Oracle がプライマリデータサーバで IBM DB2 UDB がレプリケートデータサーバである場合や、Microsoft SQL Server がプライマリサーバで Microsoft SQL Server がレプリケートサーバである場合など)

ASE は、Replication Server をサポートするために強化されました。Replication Server のサポートに必要なデータサーバの要素すべて (プライマリデータベースのデータ変更の取得メカニズムと、レプリケートデータベースのシステムテーブルおよびストアドプロシージャ) が、Adaptive Server Enterprise に組み込まれているか、Replication Server または Adaptive Server ソフトウェアに付属しているユーティリティによって使用可能になります。

ASE 以外のデータサーバを含む SAP 複写システムを実装するには、以下の追加コンポーネントが必要です。

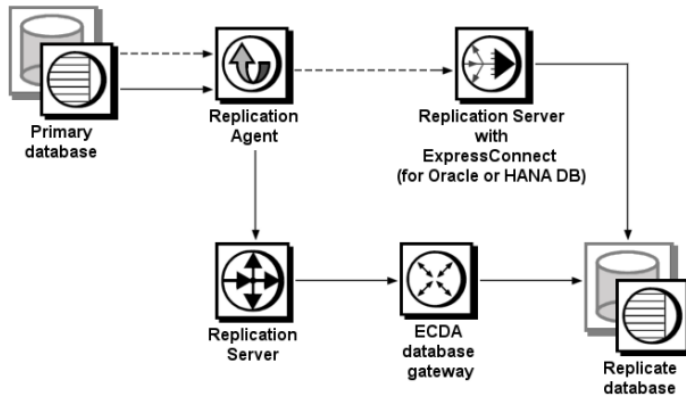
- Replication Agent™
- Enterprise Connect™ Data Access for ODBC (for Microsoft または DB2 UDB) または ExpressConnect (for Oracle または SAP HANA データベース) (複写データサーバのコネクティビティ要件が Replication Server と互換性がない場合に必要)

図は、ASE 以外のデータサーバを含む一般的な SAP 複写システムの例であり、以下を経由するデータサーバ間のデータの流れを示しています。



- Replication Agent、Replication Server、Enterprise Connect™ Data Access データベースゲートウェイ
- Replication Agent、Replication Server、ExpressConnect (for Oracle または for SAP HANA データベース)

図 2 : ASE 以外のデータサーバを含む SAP 複製システム



ECDA データベースゲートウェイを使用している場合、データは、プライマリデータベースから Replication Agent へ、Replication Agent から Replication Server へ、Replication Server から ECDA データベースゲートウェイ、そしてデータベースゲートウェイからレプリケートデータベースへと流れます。

ECDA データベースゲートウェイは、Open Client と SAP Open Server 間の接続性と ODBC をレプリケートデータサーバに提供します。そして、SQL 変換とその他のサービスを提供することによって、IBM DB2 UDB と Microsoft SQL Server をサポートします。また、Replication Server は、ASE 以外のデータサーバのデータ型サポートも提供します。

ExpressConnect (for Oracle または for SAP HANA データベース) を使用している場合、データは、プライマリデータベースから Replication Agent へ、Replication Agent から Replication Server へ、Replication Server から直接レプリケートデータベースへと流れます。

Replication Agent は、プライマリデータベースの完了済みトランザクションを読み取り、配信のために Replication Server に送信することによって、ASE 以外のプライマリデータサーバをサポートします。

## SAP 複写システムのコンポーネント

---

SAP 複写システムのコンポーネントにおける機能と役割を説明します。

複写システムのコンポーネントには、次のものが含まれます。

- プライマリデータサーバ
- Replication Agent
- Replication Server
- 次のようなレプリケートデータベースへのコネクタ
  - Enterprise Connect™ Data Access
  - ExpressConnect for Oracle
  - ExpressConnect for SAP HANA データベース
- レプリケートデータサーバ

**参照：**

- SAP の複写製品 (29 ページ)

### プライマリデータサーバ

プライマリデータサーバは、複写システムにおけるデータ変更オペレーションまたはトランザクションの送信元となる 1 つ以上のプライマリデータベースを管理します。プライマリデータサーバは、複写に必要な情報を取得するように設定されます。

すべてのプライマリデータサーバは、Replication Agent によってサポートされます。ASE の内部には、Replication Agent が含まれています。ASE 以外のサーバには、外部の Replication Agent が必要です。

**参照：**

- プライマリデータベース (15 ページ)

### サポートされているプライマリデータベースサーバ

SAP の複写テクノロジーでは、Adaptive Server Enterprise に加えて、別のリレーショナルデータベースサーバからのトランザクションの複写もアクティブにサポートしています。

サポートされているリレーショナルデータベースサーバには、次のものがあります。

- z/OS 上の IBM DB2 UDB

- UNIX/Windows 上の IBM DB2 UDB
- Microsoft SQL Server
- Oracle

これらのデータサーバのサポートされている最新バージョンについては、ASE 以外の特定のデータサーバをサポートする Replication Agent のマニュアルを参照してください。

## **Replication Agent**

*Replication Agent* は、他の (レプリケート) データベースに配信するために、データスキーマに加えられた変更とストアドプロシージャの実行を表すトランザクション情報をプライマリデータサーバから Replication Server に転送します。

Replication Agent は、プライマリデータを含むデータベースごと、または複写ストアドプロシージャが実行されるデータベースごとに必要です。

Adaptive Server Enterprise では、Replication Agent がデータベース管理システムソフトウェアに付属しています。ASE 用の Replication Agent は RepAgent と呼ばれ、Adaptive Server スレッドの 1 つです。

ASE 以外のデータサーバについては、SAP では以下の Replication Agent 製品を提供しています。

- Replication Agent for DB2 UDB - IBM z/OS プラットフォームで実行されている IBM DB2 UDB サーバに、プライマリデータサーバのサポートを提供します。
- Replication Agent - Linux、UNIX、または Microsoft Windows プラットフォームで実行されている DB2 UDB、Microsoft SQL Server、Oracle データサーバに、プライマリデータサーバのサポートを提供します。

Replication Agent はプライマリデータベースのトランザクションログを読み取り、プライマリ Replication Server はトランザクションを再構築し、データのサブスクリプションを持つレプリケートサイトに転送します。

## **Replication Server**

各プライマリサイトまたはレプリケートサイトの Replication Server は、ローカルデータサーバのデータ複写アクティビティを調整し、他のサイトの Replication Server とデータを交換します。

Replication Server は、以下の方法で、保証されたトランザクションの配信を各レプリケートサイトに提供します。

- Replication Agent を介してプライマリデータベースからトランザクションを受け取り、データのサブスクリプションを持つレプリケートデータベースサイトに配信する。

## 複写システムの概要

- 他の Replication Server からトランザクションを受け取り、ローカルのレプリケートデータベースに適用するか、データに対するサブスクリプションを持つ他の Replication Server に転送する。
- レプリケートデータベースからデータ更新の要求を受信して、プライマリデータベースに適用する。

これらのタスクの実行に必要な情報は、Replication Server システムデータベースに格納される Replication Server システムテーブルに格納されています。

Replication Server の内部要素の詳細については、『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Replication Server の内部処理」を参照してください。

### **ID サーバ**

ID サーバは、複写システム内のすべての Replication Server とデータベースを登録している Replication Server です。

ID サーバとして機能する Replication Server は、通常の Replication Server タスクに加え、複写システム内の各 Replication Server とデータベースにユニークな ID 番号を割り当てます。ID サーバはまた、複写システムのバージョン情報を管理します。この点を除けば、ID サーバはその他の Replication Server と変わらない働きをします。

新しい Replication Server や、新しいデータベースを管理する Replication Server がログインして ID 番号を取り出せるようにするには、ID サーバが次の場合に稼働している必要があります。

- Replication Server をインストールする場合
- ルートを作成する場合
- データベースコネクションを作成または削除する場合

上記の条件があるので、ID サーバは、複写システムをインストールするときに最初にインストールおよび起動する Replication Server になります。Replication Server が 1 つだけの場合や、Replication Server を初めてインストールする場合は、その Replication Server は ID サーバにもなります。既存の複写システムに Replication Server を追加する場合、その複写システムの ID サーバである Replication Server の名前を知る必要があります。

ID サーバには、Replication Server が ID サーバと接続する時に使用する、Replication Server 用のログイン名が必要です。このログイン名は、複写システムを設定および管理するときに、`rs_init` 設定プログラムによって、複写システム内のすべての Replication Server の設定ファイルに記録されます。

---

**警告！** ID サーバは、複写環境にとって重要なものなので、いったんインストールすると、移動が困難です。いったん ID サーバの名前を決定してしまうと、別の

Replication Server へは変更できません。SAP では、設定ファイルに記録した ID サーバの名前を変更する手順はサポートしていません。

---

### 複写システムドメイン

「複写システムドメイン」とは、同じ ID サーバを使用するすべての複写システムコンポーネントを指します。

企業によっては、独立した複数の複写システムを持つところもあります。ID サーバは、複写システム内のメンバ Replication Server およびデータベースを決定するので、複数の複写システムがある場合の各複写システムを ID サーバドメインともいいます。

複数の ID サーバドメインを設定するために特別な作業は必要ありません。どの Replication Server やデータベースも、1つの複写システム、つまり ID サーバドメインに属し、その ID サーバドメイン内でユニークな ID 番号を持ちます。

次の制限事項のもとで、複数の複写システムドメインを設定できます。

- 異なるドメインに属する Replication Server 間では、データを交換できません。各ドメインは、相互に通信できない独立した複写システムとして扱います。異なるドメインに属する Replication Server 間にルートを作成することはできません。
- 1つのデータベースを管理できるのは、1つのドメイン内のただ1つの Replication Server のみです。どのデータベースも、ただ1つの ID サーバのドメイン内に存在します。つまり、異なるドメインから同じデータベースへのコネクションを複数作成することはできません。

## **Replication Server システムデータベース (RSSD)**

RSSD (Replication Server システムデータベース) は、Replication Server システムテーブルを格納するデータベースです。

各 Replication Server には、1つの Replication Server に対するシステムテーブルを保持する RSSD または Embedded Replication Server システムデータベース (ERSSD) が1つ必要です。RSSD は Adaptive Server によって管理されます。ERSSD は SQL Anywhere® によって管理されます。

### システムテーブル

Replication Server システムテーブルには、Replication Server が複写データを送受信するために必要な情報が保持されます。

システムテーブルには次のような情報が保持されます。

- 複写データと関連情報の記述
- 複写定義やサブスクリプションなどの複写オブジェクトの説明

## 複写システムの概要

- Replication Server ユーザ用のセキュリティレコード
- 他の Replication Server サイトに対するルート指定情報
- ローカルデータベースのアクセスメソッド
- その他の管理情報

Replication Server のシステムテーブルは、Replication Server のインストール時に RSSD にロードされます。

システムテーブル全体のリストについては、『Replication Server リファレンスマニュアル』の「Replication Server システムテーブル」を参照してください。

システムテーブルの内容は、RCL コマンドの実行などの Replication Server アクティビティ中に変更されます。システムテーブルを変更できるのは、複写システム管理者と、**rs\_systabgroup** グループのメンバだけです。

システムテーブルに問い合わせステータス情報を調べるには、次の手順に従います。

- Replication Server のシステム情報コマンドまたはシステム管理コマンドを使用します。詳細については、『Replication Server リファレンスマニュアル』の「複写コマンド言語」の「システム情報コマンド」と『Replication Server リファレンスマニュアル』の「複写コマンド言語」の「システム管理コマンド」を参照してください。
- Adaptive Server スタアドプロシージャを使用して、複写システムについての情報を表示します。『Replication Server リファレンスマニュアル』の「Adaptive Server コマンドとシステムプロシージャ」を参照してください。

---

**警告！** RSSD テーブルは、Replication Server の内部で使用するものです。SAP 製品の保守契約を結んでいるサポートセンタからの指示がないかぎり、RSSD テーブルを直接変更しないでください。

---

### **RSSD と複写エージェントの仕様**

Replication Agent は、Replication Server がルートの送信元である場合に、RSSD の Replication Agent が必要です。

ルートの送信元である Replication Server は、その RSSD 内の情報の一部を他の Replication Server に分配します。

RSSD は、RSSD がサポートする Replication Server 専用です。ユーザデータの格納には使用しないでください。ただし、1つのデータサーバに RSSD とユーザデータベースを格納することはできます。RSSD 用のデータベースデバイス領域には、少なくとも 20MB (データ用に 10MB とログ用に 10MB) が必要です。データベースとデータベースログは別々のデバイスに置くことをおすすめします。

## データベースゲートウェイ

データベースゲートウェイによって、クライアントは特定の通信プロトコルを使用して、異なるプロトコルを使用するデータサーバに接続できるようになります。

Enterprise Connect™ Data Access (ECDA) 製品群は、Open Client および Sybase Open Server プロトコルを使用するクライアント (Replication Server など) が、データサーバのネイティブな通信プロトコルや標準の ODBC プロトコルを使用する SAP 以外のデータサーバと接続できるようにする、データベースゲートウェイサーバで構成されます。

また、Enterprise Connect Data Access 製品を使用して、ASE 以外のレプリケートデータサーバからメタデータを取得することもできます。

### 参照：

- Enterprise Connect™ Data Access (50 ページ)

## ExpressConnect for Oracle

ExpressConnect for Oracle は、Replication Server とレプリケート Oracle データサーバ間の直接通信を提供します。

ExpressConnect for Oracle では、ゲートウェイサーバのインストールと設定の必要がないため、複写システムのパフォーマンスを向上させ、複写システムの管理が煩雑にならないようにします。

### 参照：

- ExpressConnect for Oracle (54 ページ)

## ExpressConnect for SAP HANA データベース

ExpressConnect for SAP HANA® データベースは、SAP HANA データベースへの複写用に SAP Replication Server によってロードされる埋め込みライブラリです。

ExpressConnect for SAP HANA データベースは、SAP Replication Server 15.7.1 SP200 とともにインストールされ、SAP Replication Server とレプリケート SAP HANA データベースサーバ間の直接通信を可能にします。

ExpressConnect for SAP HANA データベースは、SAP Replication Server とともにインストールされます。ExpressConnect for SAP HANA データベースの専用のインストーラはありません。

『異機種間複写ガイド』の「ExpressConnect for SAP HANA データベースと SAP HANA データベースに対するレプリケートデータベースのコネクティビティ」を参照してください。

### 参照：

- ExpressConnect for SAP HANA データベース (55 ページ)

## レプリケートデータサーバ

レプリケートデータサーバは、レプリケートデータ、つまりプライマリデータベース内のデータの「コピー」であるデータを格納するデータベースを管理します。

Replication Server は、データベースユーザとしてログインすることによって、レプリケートデータサーバ内のデータを管理します。ASE 以外のデータサーバの場合は、データベースゲートウェイサーバを介して、または直接レプリケートデータサーバにログインします。

Replication Server は、必要なデータオペレーションとトランザクション処理命令が直接的 (Adaptive Server Enterprise など) または間接的 (Enterprise Connect Data Access データベースゲートウェイサーバなど) にサポートされている場合、あらゆるサーバをデータサーバとして扱うことができます。

### 参照：

- レプリケートデータベース (16 ページ)

## サポートされているレプリケートデータベースサーバ

SAP 複写テクノロジーは、別のリレーショナルデータベースサーバでのトランザクションの複写をサポートしています。

リレーショナルデータベースサーバには、次のものがあります。

- SAP HANA データベース
- z/OS 上の IBM DB2 UDB
- UNIX/Windows 上の IBM DB2 UDB
- Microsoft SQL Server
- Oracle
- SAP IQ

Microsoft SQL Server および DB2 UDB データサーバのサポートされている最新バージョンの詳細については、ASE 以外の特定のデータサーバに関連付けられた ECDA データベースゲートウェイのマニュアルを参照してください。

ExpressConnect についてサポートされている Oracle データサーババージョンについては、『ExpressConnect for Oracle Configuration Guide』を参照してください。

ExpressConnect for SAP HANA データベースについては、このマニュアルの「レプリケートデータサーバとしての SAP HANA データベース」を参照してください。



## ASE 以外の複写

---

ASE 以外のサーバを使用して複写する場合は、データサーバの種類またはメーカーに関係なく、複写システムにおけるデータサーバの役割に関する問題を考慮する必要があります。

異機種間複写システムを適切に実装する上での最大の課題は、さまざまなベンダによって提供されているデータサーバ固有の特性に対処することです。1つのデータサーバがプライマリデータサーバとレプリケートデータサーバの両方の役割を果たす場合は(双方向複写)、さらに多くの問題を考慮しなければなりません。

### プライマリデータベース

異機種間複写システムを適切に実装するには、プライマリデータベースの問題を処理する必要があります。

ASE 以外のプライマリデータベースを使用する場合は、次の事項を検討してください。

- Replication Agent の稼働条件とデータサーバへの Replication Agent の干渉と影響。たとえば、一部の Replication Agent は、複写をサポートするためにプライマリデータベースにデータベースオブジェクトを作成して使用します。
- 他の複写システムコンポーネントに必要な、データサーバのアクセスとパーミッション。データベースのプライマリ Replication Server と Replication Agent の両方に、データベースで定義された、プライマリデータベースオブジェクトにアクセスするための適切なパーミッションを持つユーザ ID とパスワードが必要です。
- データサーバと他の複写システムコンポーネント間の通信をサポートするために必要なコネクティビティ。Replication Agent は、データサーバのネイティブ通信プロトコル、ODBC プロトコル、または JDBC プロトコルを使用して、プライマリデータベースと通信します。Replication Server には、データサーバと通信するためにデータベースゲートウェイが必要な場合があります。
- 特定のデータサーバからの複写に関する固有の制限。たとえば、Replication Agent は、一部のデータサーバの設定オプションを制限します。Replication Server には、データベースによって、一部のネイティブなデータ型にサイズの制限がある場合があります。
- RSSD に格納されている複写定義が、特定のデータサーバの Replication Agent によってどのように使用されるか。たとえば、データベースオブジェクト名を識別するときに、Replication Server と Replication Agent の両方が大文字と小文字を区別しますが、大文字と小文字を区別しないデータベースもあります。
- 特定のデータサーバから別の種類のデータサーバにトランザクションを複写するときに必要となる可能性があるデータ型変換。たとえば、ほとんどすべての

## 複写システムの概要

種類のデータサーバが、それぞれに固有の方法で時間データを表します。あるデータベースの `TIMESTAMP` データ型を、別のデータベースでは `datetime` データ型に「変換」して格納する必要がある場合があります。

- 特定のデータサーバに固有の複写システム管理に関する問題。たとえば、データサーバに応じて、異なるシステム管理オプションを使用することができます。

特定のデータベースに固有のプライマリデータベースの問題の詳細については、使用しているデータベースに該当するトピックを参照してください。

## レプリケートデータベース

異機種間複写システムを適切に実装するには、レプリケートデータベースの問題を処理する必要があります。

ASE 以外のレプリケートデータベースを使用する場合は、次の事項を検討してください。

- レプリケートデータベースで `ExpressConnect` を使用できるかどうか。複写システムを簡略化し、パフォーマンスを向上させるために、レプリケートデータベースで使用可能であれば `ExpressConnect` を使用してください。
- 特定のデータベースサーバに関する ECDA データベースゲートウェイの稼働条件。レプリケートデータベースサーバおよび `Replication Server` とともに動作するように `DirectConnect™` のアクセスサービスを設定します。
- トランザクションをレプリケートデータベースに適用するために、複写システムのデータサーバで必要とされるアクセスとパーミッション。データベースのレプリケート `Replication Server` と ECDA ゲートウェイの両方に、データベースで定義された、レプリケートデータベースオブジェクトにアクセスするための適切なパーミッションを持つユーザ ID とパスワードが必要です。
- レプリケートデータサーバと他の複写システムコンポーネント間の通信をサポートするために必要なコネクティビティ。ECDA ゲートウェイと `ExpressConnect (for Oracle または SAP HANA データベース)` は、データサーバのネイティブな通信プロトコルと標準の `ODBC` プロトコルのどちらかを使用してレプリケートデータベースと通信します。ECDA ゲートウェイと `ExpressConnect (for Oracle または SAP HANA データベース)` にはコネクティビティライブラリが必要です。このライブラリは各自で用意する必要があります。ASE 以外のデータサーバと通信する場合、`Replication Server` には通常、データベースゲートウェイまたは `ExpressConnect (for Oracle または SAP HANA データベース)` が必要です。
- 特定のデータサーバへの複写に関する制限。たとえば、`Replication Server` には、データベースによって、一部のネイティブなデータ型に制限があります。
- `Replication Server` の動作をサポートするために必要なデータベースオブジェクトの介入と影響。`Replication Server` には 2 つのテーブルが必要です。また、レ

プリケートデータベースを管理するためにストアードプロシージャが必要な場合もあります。

- 特定のデータサーバに固有の複写システム管理に関する問題。たとえば、データサーバに応じて、異なるシステム管理オプションを使用することができます。

特定のデータベースに固有のレプリケートデータベースの問題の詳細については、使用しているデータベースに該当するトピックを参照してください。

## 文字セット

文字セットを設定すると、プライマリデータベースとレプリケートデータベース間のデータの不整合を引き起こす問題を回避します。

プライマリデータサーバとレプリケートデータサーバの種類が異なる異機種間複写システムでは、データサーバが同じ文字セットをすべてサポートしていない場合があります。そのような場合、複写システムコンポーネントは、(プライマリデータサーバの文字セットからレプリケートデータサーバの文字セットへの)文字セットの変換を、少なくとも1回は実行する必要があります。

プライマリデータサーバとレプリケートデータサーバの種類が同じである同機種間複写システムでも、複写システムコンポーネントが複数の種類のプラットフォームに存在している場合は、文字セットの変換が必要になることがあります。

文字セットの問題を回避するには、次のいずれかを実行する必要があります。

- 複写システムのすべてのサーバおよびプラットフォームで同じ文字セットを使用する。
- 複写システムのすべてのサーバおよびプラットフォームで、互換性のある文字セットを使用し、適切な文字セットの変換を実行するように、複写システムコンポーネントを設定する。

デフォルト文字セットの設定と上書きの詳細については、該当する Replication Agent のマニュアルを参照してください。

## 異機種間複写の制限事項

特定の関連データベースによっては、SAPの複写テクノロジーに基づき、異機種間複写システムにはいくつかの制限事項があります。

### ストアードプロシージャの複写

ストアードプロシージャの複写によって、プライマリストアードプロシージャ呼び出しに引数として渡されるパラメータ値を含めた、複写するストアードプロシージャの実行呼び出しが可能になります。

ストアードプロシージャの複写を使用できるかどうかは、プライマリデータベースおよびレプリケートデータベースの機能と、関連する Replication Agent、ECDA

データベースゲートウェイ、および ExpressConnect (for Oracle または SAP HANA データベース) のサポートによって決まります。ストアドプロシージャの複写がデータベースで使用可能かどうかを確認するには、特定の Replication Agent、ECDA、および ExpressConnect コンポーネントのマニュアルを参照してください。

### 所有者指定のオブジェクト名

ASE 以外のデータベース内のレプリケートテーブルおよびストアドプロシージャにアクセスする場合、レプリケートテーブルまたはストアドプロシージャへの参照に所有者の指定が必要なことがよくあります。

たとえば、Oracle のレプリケートデータベースにトランザクションを適用するために割り当てられている Replication Server のメンテナンスユーザが *orauser* であるとし、**insert** コマンド (table1 へのレプリケート) を実行すると、「テーブルが見つからない」というエラーが発生することがあります (テーブル table1 の所有者が *bob* である場合)。table1 を検索する場合、Oracle では bob.table1 ではなく orauser.table1 を探します。更新するレプリケートテーブルを適切に指定するには、次の作業を行います。

- 正しいレプリケートテーブルを参照するエイリアスを、Oracle のレプリケートデータベースで作成します。たとえば、“bob.table1” という完全修飾名を参照する table1 という同義オブジェクトを Oracle で作成します。
- 複写定義を作成するときに、**with replicate table named [table\_owner. [table\_name]]** 句を使用します。同じ例を引き続き使用すると、次のような句になります。

```
with replicate table named bob.table1
```

#### *複数のレプリケートデータベースがある場合の所有者の指定*

table1 が複数のレプリケートデータベース (たとえば、Oracle レプリケートテーブル bob.table1) に複写される場合は、問題が多少複雑になります。複写定義で **with replicate table named** 句を使用する方法では、1 つのレプリケートテーブル名しかサポートされません。

この問題に対処するには、必要とされるユニークなレプリケートテーブル名ごとに、1 つずつ複写定義を作成します。各サブスクリプションが正しい複写定義を参照し、各複写定義が **with replicate table named** 句を使用していることを確認してください。

### ラージオブジェクトの複写

ラージオブジェクト (LOB) データ型 (BLOB、CLOB、IMAGE、TEXT など) は、1 つのカラム内における最長の文字ストリームおよびバイナリデータストリームをサポートします。プライマリデータとレプリケートデータのどちらの場合においても、LOB データ型のサイズによって固有の問題が生じます。

### プライマリデータベースのLOB 複写の問題

プライマリデータベースでは、LOB データ型はトランザクションログ機能に影響します。

Replication Agent の場合、LOB データの変更の保持をサポートする十分なログリソースが必要です。ログに記録されるのは、LOB データの更新後のイメージのみです。LOB 複写の機能は、Replication Agent の機能によって決まります。

### レプリケートデータベースのLOB 複写の問題

SAP 以外のデータベースがレプリケートデータベースである場合、レプリケートデータベースとの通信に使用されるデータベースゲートウェイは、Adaptive Server のテキストポインタ処理をエミュレートできなければなりません。

ECDA Option for ODBC to Microsoft ゲートウェイと SAP® Mainframe Connect™ DirectConnect for z/OS Option ゲートウェイはこの機能を備えています。ECDA Option for ODBC to DB2 UDB はこの機能を備えていません。ExpressConnect (for Oracle または SAP HANA データベース) には、LOB 複写をサポートするためのテキストポインタ処理のサポートは必要ありません。ExpressConnect (for Oracle または SAP HANA データベース) はLOB 複写をサポートしますが、テキストポインタ処理は不要なためサポートしません。

Adaptive Server Enterprise では、テキストポインタを使用して text および image カラムデータの場所を特定します。このテキストポインタは、これらのラージカラム内のデータに対して実際の更新を実行するシステムファンクションに渡されます。Replication Server でも、内部で同じ方法を使用してLOB データ型に適用されます。Replication Server がテキストポインタを取得すると、レプリケートデータベースにデータを適用するデータサーバファンクションが呼び出されます。

ECDA Option for ODBC は、Microsoft SQL Server データベースへのLOB の複写をサポートしています。

## レプリケートデータベースの設定

Replication Server は、Adaptive Server データベースを設定する **rs\_init** というユーティリティを提供しています。

**rs\_init** は、Adaptive Server データベースをプライマリまたはレプリケートデータベースとして、次のように設定します。

- Replication Server データベースコネクションを作成する
- 必要なテーブルとストアードプロシージャをレプリケートデータベースに作成する
- Replication Server のメンテナンスユーザ ID を定義する

異機種間複写のサポートには、**rs\_init** に相当するユーティリティは含まれていません。代わりに、コネクションを作成するための **Replication Server** コマンドと、複写をサポートするオブジェクト (メンテナンスユーザなど) を作成するための **プライマリデータサーバコマンド** と **レプリケートデータサーバコマンド** を使用できます。これらのタスクの多くは、**Replication Server 15.2** で導入された **create connection** コマンドの “using profile” 句を使用して実現できます。

## Replication Server の暗号化カラムのサポート

**Replication Server** は、**Adaptive Server** データベース間の暗号化カラムデータの複写をサポートしています。ただし、**ASE** 以外のレプリケートデータベースへの暗号化カラムデータの複写はサポートしていません。

暗号化カラムを含む **ASE** データベースに暗号化されていないデータを複写するには、**Adaptive Server** コネクションに対する **rs\_set\_ciphertext** ファンクション文字列を無効にします。**rs\_set\_ciphertext** ファンクション文字列は、デフォルトで、すべての **ASE** コネクションに対して実行されます。これは、複写するデータが既に暗号化されていて、プライマリデータベースも同じ暗号を使用した **ASE** であると仮定されるレプリケート **ASE** データベースを示します。**rs\_set\_ciphertext** ファンクション文字列を無効にすることにより、レプリケート **ASE** は入力レプリケートデータに対して暗号化を実行できます。**ASE** が入力データを暗号化できるようにすることは、プライマリデータベースが **ASE** 以外の場合またはプライマリ **ASE** データベースが暗号化されたカラムを使用しない場合に適しています。

### rs\_set\_ciphertext ファンクション文字列

**rs\_set\_ciphertext** は、**Adaptive Server** テーブルへの暗号化カラムの複写を制御します。

ファンクション文字列 **rs\_set\_ciphertext** を変更して、**ASE** 固有のコマンド “**set ciphertext on**” の実行を無効にします。

```
alter function string rs_set_ciphertext
for some_function_string_class
output language
''
```

## サブスクリプションマテリアライゼーション

マテリアライゼーションとは、サブスクリプションを作成してアクティブ化し、プライマリデータベースからレプリケートデータベースにデータをコピーすることによって、レプリケートデータベースを初期化することです。

プライマリデータベースからデータを複写するには、プライマリデータベースの状態と一致した状態になるように、各レプリケートデータベースを設定し、データを読み込んでおく必要があります。**Replication Server** では、次の2種類のサブスクリプションマテリアライゼーションをサポートしています。

- バルクマテリアライゼーション - サブスクリプションを手動で作成してアクティビティ化し、複製システムのコントロール外のデータアンロードユーティリティとデータロードユーティリティを使用してレプリケートデータベースにデータを読み込ませます。
- 自動マテリアライゼーション - Replication Server コマンドを使用してサブスクリプションを作成し、レプリケートデータベースにデータを読み込みます。

異機種間複製ではバルクマテリアライゼーションメソッドがサポートされていませんが、個別の Replication Agent 機能によって複雑さは異なります。

サブスクリプションマテリアライゼーション全般については、『Replication Server 管理ガイド』を参照してください。また、個別の Replication Agent とそのマテリアライゼーションのサポートの詳細については、該当する Replication Agent のマニュアルを参照してください。

## Replication Server の rs\_dump コマンド

**rs\_dump** コマンドは通常、複製システム全体におけるデータベースのダンプアクティビティを調整するために使用されます。

レプリケートコネクションが **rs\_dump** トランザクションを受け取ると、Replication Server はそのコネクションに対して **rs\_dump** ファンクション文字列を実行します。コマンドが、必要なあらゆる操作を実行するように、**rs\_dump** ファンクション文字列をカスタマイズできます。

ASE 以外のプライマリデータベースの複製の場合、一部の Replication Agent には SAP 以外のプライマリデータベースから **rs\_dump** コマンドを呼び出す方法が用意されています。プライマリデータベースからの **rs\_dump** の実行がサポートされているかどうかを確認するには、該当する Replication Agent のマニュアルを参照してください。

レプリケートデータベースについては、**rs\_dump** のデフォルトファンクション文字列は用意されていません。

**rs\_dump** コマンド、その使用方法、およびファンクション文字列の変更の詳細については、『Replication Server リファレンスマニュアル』を参照してください。

## Replication Server rs\_marker コマンド

**rs\_marker** コマンドは、マテリアライゼーションプロセスを支援するプライマリデータベーストランザクションログのマーカメカニズムです。

**rs\_marker** を実行すると、**activate subscription** コマンドと **validate subscription** コマンドがプライマリ Replication Server に渡されます。ほとんどの Replication Agent は、マテリアライゼーションを支援するために **rs\_marker** の呼び出しをサポートしています。

**rs\_marker** の使用方法の詳細については、『Replication Server リファレンスマニュアル』を参照してください。特定のデータベースでの **rs\_marker** の使用方法と可用性の詳細については、該当する Replication Agent のマニュアルを参照してください。

### Replication Server の rs\_dumptran コマンド

**rs\_dumptran** コマンドは通常、複写システム全体におけるデータベーストランザクションのダンプアクティビティを調整するために使用されます。

レプリケートコネクションが **rs\_dumptran** トランザクションを受け取ると、Replication Server はそのコネクションに対して **rs\_dumptran** ファンクション文字列を実行します。コマンドが必要なあらゆる操作を実行するように、**rs\_dumptran** ファンクション文字列をカスタマイズできます。

異機種間複写では、SAP 以外のプライマリデータベースに対して **rs\_dumptran** はサポートされません。

レプリケートデータベースについては、**rs\_dumptran** のデフォルトファンクション文字列は用意されていません。

**rs\_dumptran** コマンド、その使用方法、およびファンクション文字列の変更の詳細については、『Replication Server リファレンスマニュアル』を参照してください。

### Replication Server の rs\_subcmp ユーティリティ

**rs\_subcmp** 実行プログラムは、プライマリテーブルとレプリケートテーブルを比較して差異を確認し、必要に応じて調整を行います。

プライマリデータベースおよびレプリケートデータベースへのコネクティビティが確保されている場合には、SAP 以外のデータベースをサポートするために、**rs\_subcmp** を使用できます。プライマリデータベースおよびレプリケートデータベース用に、両者の比較可能な出力を生成するカスタム **SELECT** コマンドを開発する必要もあります。その他に、そのような機能を備えたサードパーティのツールを購入するか、または独自のアプリケーションを作成する方法があります。

SAP® Replication Server® データ保証オプションを使用して、2つ以上のデータベースのローデータおよびスキーマを比較して、不一致を報告することもできます。

#### 参照：

- 異機種データベースの調整 (323 ページ)



## 動的 SQL

動的 SQL により、Replication Server データサーバインタフェース (DSI) を使用して、ターゲットユーザデータベースに動的 SQL 文を作成し、繰り返し実行できます。

動的 SQL は、UNIX、Windows、および Linux 上の Oracle、SAP HANA データベース、Microsoft SQL Server、DB2 UDB z/OS、および DB2 UDB で使用できます。SAP IQ では使用できません。

## バルクコピー

バルクコピーは、Replication Server データサーバインタフェース (DSI) に Open Client Open Server Bulk-Library インタフェースを使用して同じテーブルで大量の insert 文を複写する場合にパフォーマンスを向上させます。

ExpressConnect (for Oracle または SAP HANA データベース) を使用する場合、SAP IQ、Oracle、および SAP HANA データベースを除き、ASE 以外のデータサーバのいずれでもバルクコピーを使用することはできません。

## Replication Server rs\_ticket ストアドプロシージャ

rs\_ticket は、Replication Server のパフォーマンス、モジュールのハートビート、複写の正常性、テーブルレベルのクワイズをモニタするのに役立つ、プライマリデータベースのストアドプロシージャです。

rs\_ticket は、UNIX、Windows、および Linux 上の Oracle、SAP HANA データベース、DB2 UDB と、Microsoft SQL で使用できます。DB2 UDB z/OS では使用できません。『Replication Agent リファレンスマニュアル』を参照してください。

## 複写システムの ASE 以外の設定

Replication system の問題は、異機種または ASE 以外のデータサーバと構成が異なるために発生します。

## ASE 以外のプライマリから Adaptive Server レプリケートへの複写

最も簡単な異機種間複写シナリオは、ASE 以外のプライマリデータベースから Adaptive Server レプリケートデータベースへの単方向の複写です。

固有の要件は、ASE 以外のプライマリデータベースからトランザクションデータを抽出するために設計された Replication Agent が存在すること、およびプライマリ

## 複写システムの概要

データベースのネイティブデータ型を Adaptive Server データ型に変換するための Replication Server の異機種データ型サポート (HDS) 機能を適用することだけです。

『Replication Server 管理ガイド 第1巻』の「複写テーブルの管理」の「HDSを使用したデータ型の変換」を参照してください。

### 複写システムのコンポーネント

ASE 以外のプライマリから Adaptive Server レプリケートへの複写の設定には、次のコンポーネントが必要です。

- ASE 以外のプライマリデータサーバ。Oracle などです。
- プライマリデータサーバ用に設計された Replication Agent
- Replication Server
- Adaptive Server レプリケートデータサーバ

### 複写システムに関する問題

ASE 以外のプライマリから Adaptive Server レプリケートへの複写の設定では、ユーザ ID がプライマリデータベースにトランザクションを適用しない場合でも、プライマリデータベースの Replication Server データベースコネクションに、(Replication Agent に対してのみ検証される) プライマリデータベースに有効なユーザ ID とパスワードが必要になる場合があります。

## ASE サーバのプライマリから ASE 以外のサーバのレプリケートへの複写

簡単な異機種間複写シナリオは、Adaptive Server プライマリデータベースから ASE 以外のレプリケートデータベースへの単方向の複写です。

固有の要件は、レプリケートデータベースにトランザクションデータを適用するためのコンポーネント、および Adaptive Server データ型をレプリケートデータベースのネイティブデータ型に変換するための Replication Server の HDS 機能を適用することだけです。

HDS の詳細については、『Replication Server 管理ガイド 第1巻』の「複写テーブルの管理」で「HDSを使用したデータ型の変換」を参照してください。

### 複写システムのコンポーネント

Adaptive Server プライマリから ASE 以外のレプリケートへの複写の設定に必要なコンポーネントは次のとおりです。

- Adaptive Server プライマリデータベース
- Replication Server
- Oracle または SAP HANA データベースに複写するための ExpressConnect (for Oracle または SAP HANA データベース)、あるいはレプリケートデータサーバ

向けに設計された、関連する ECDA データベースゲートウェイ (ECDA Option for ODBC for Microsoft SQL Server など)

- ASE 以外のレプリケートデータサーバ。Microsoft SQL Server など

#### 複写システムに関する問題

Adaptive Server プライマリから ASE 以外のレプリケートへの複写の設定では、次の点を考慮してください。

- レプリケートデータベースの Replication Server データベースコネクションに、レプリケートデータベースに有効なユーザ ID とパスワード (メンテナンスユーザ) が指定されている必要があります。このユーザ ID には、レプリケートデータベースにレプリケートトランザクションを適用する権限が必要です。
- プライマリデータベースとレプリケートデータベース用の正しいプロファイルを使用して、Replication Server レプリケートデータベース接続を作成します。接続プロファイルは、レプリケートデータベースの正しいファンクション文字列クラスとエラークラスを指定します。また、クラスレベル変換の定義とレプリケートデータベースオブジェクトの作成の複写サポートを含める場合があります。

### ASE 以外のプライマリから ASE 以外のレプリケートへの複写

ASE プライマリから ASE 以外のレプリケートへのシナリオは、ASE 以外のデータサーバの組み合わせに応じて複雑さが異なります。

#### 複写システムのコンポーネント

ASE 以外のプライマリから ASE 以外のレプリケートへの複写の設定には、次のコンポーネントが必要です。

- ASE 以外のプライマリデータサーバ。Oracle などです。
- プライマリデータサーバ用に設計された Replication Agent。Replication Agent for Oracle などです。
- Replication Server。
- Oracle または SAP HANA データベースに複写するための ExpressConnect (for Oracle または SAP HANA データベース)、あるいはレプリケートデータサーバ向けに設計された、関連する ECDA データベースゲートウェイ (ECDA Option for ODBC for Microsoft SQL Server など)。
- ASE 以外のレプリケートデータサーバ。Microsoft SQL Server など

#### 複写システムに関する問題

ASE 以外のプライマリから ASE 以外のレプリケートへの複写の設定では、次の点を考慮してください。

- Replication Server プライマリデータベースコネクションに、プライマリデータベースに有効なユーザ ID とパスワードが必要になる場合があります。この

ユーザ ID には、レプリケートトランザクションを適用する権限が必要です (プライマリデータベースにトランザクションが複写されない場合でも必要です)。

- 直接ロードマテリアライゼーションを使用するには、適切な接続プロファイル (rs\_rs\_to\_oracle\_ra、rs\_rs\_to\_mssql\_ra、rs\_rs\_to\_udb\_ra、または rs\_rs\_to\_hanadb\_ra) を使用してプライマリデータベースへのコネクションを作成する必要があります。
- プライマリデータベースとレプリケートデータベース用の正しいプロファイルを使用して、Replication Server レプリケートデータベース接続を作成する必要があります。接続プロファイルは、レプリケートデータベースの正しいファンクション文字列クラスとエラークラスを指定します。また、クラスレベル変換の定義とレプリケートデータベースオブジェクトの作成の複写サポートを含める場合があります。

### **ASE 以外から ASE 以外への双方向複写**

ASE 以外から ASE 以外への双方向複写シナリオでは、複写は各データベース間で発生します。

ASE 以外の各データベースに、Replication Agent と ECDA データベースゲートウェイの両方が必要です。

#### 複写システムのコンポーネント

ASE 以外のプライマリと ASE 以外のレプリケート間の双方向複写の設定には、次のコンポーネントが必要です。

- ASE 以外のプライマリデータサーバ。UNIX、Windows、Linux 上の DB2 UDB などです。
- プライマリデータサーバ用に設計された Replication Agent。Replication Agent for Oracle、Microsoft SQL Server、DB2 UDB などです。
- Oracle または SAP HANA データベースに複写するための ExpressConnect (for Oracle または SAP HANA データベース)、あるいはレプリケートデータサーバ向けに設計された、関連する ECDA データベースゲートウェイ (ECDA Option for ODBC (for DB2 UDB) など)。
- Replication Server。
- Oracle または SAP HANA データベースに複写するための ExpressConnect (for Oracle または SAP HANA データベース)、あるいはレプリケートデータサーバ向けに設計された、関連する ECDA データベースゲートウェイ (ECDA Option for ODBC for Microsoft SQL Server など)。
- プライマリデータサーバとして動作する「レプリケート」データサーバ用に設計された Replication Agent。Replication Agent for Linux、Microsoft Windows、UNIX などです。
- ASE 以外のレプリケートデータサーバ。Microsoft SQL Server など

### 複製システムに関する問題

技術的には、Replication Server データベース接続を2つ(各データベースに「プライマリおよびレプリケート」コネクションを1つずつ)使用するだけで、双方向複製シナリオを設定できます。

---

**注意：**双方向複製の問題に関する以下の説明では、2つのデータベースを Database #1 と Database #2 と呼びます。これは、どちらのデータベースも複製システムで「プライマリ」と「レプリケート」の両方の役割を果たすためです。

---

ASE 以外のプライマリと ASE 以外のレプリケート間の双方向複製の設定では、次の点を考慮してください。

- 直接ロードマテリアライゼーションを使用するには、データベースごとに受信データ用と送信データ用の2つのコネクションを用意する必要があります。適切な接続プロファイル(rs\_rs\_to\_<replicate>\_ra。<replicate>は oracle、mssql、udb、または hanadb)を使用して、Replication Agent で使用される受信データ用のコネクションを作成する必要があります。送信データ用のコネクションを作成するには、通常の複製をサポートしている接続プロファイルを使用します。
- Database #1 用の Replication Server プライマリデータベースコネクションに、プライマリデータベースに有効なユーザ ID とパスワードが指定されている必要があります。このユーザ ID は、Database #2 用の Replication Server レプリケートデータベースコネクションで指定されているユーザ ID (メンテナンスユーザ)と同じでなければなりません。このユーザ ID には、Database #1 のレプリケートテーブルにトランザクションオペレーションを適用する権限が必要です。
- Database #1 の Replication Agent は、Database #2 内のレプリケートテーブルからトランザクションが返されないようにするために、メンテナンスユーザのトランザクションを無視するように設定されている必要があります。メンテナンスユーザのトランザクションを無視するように Replication Agent を設定する方法の詳細については、該当する Replication Agent のマニュアルを参照してください。
- Database #2 用の Replication Server プライマリデータベースコネクションに、プライマリデータベースに有効なユーザ ID とパスワードが指定されている必要があります。このユーザ ID は、Database #1 用の Replication Server レプリケートデータベースコネクションで指定されているユーザ ID (メンテナンスユーザ)と同じでなければなりません。このユーザ ID には、Database #2 のレプリケートテーブルにトランザクションオペレーションを適用する権限が必要です。
- Database #2 の Replication Agent は、Database #1 内のレプリケートテーブルからトランザクションが返されないようにするために、メンテナンスユーザのトランザクションを無視するように設定されている必要があります。メンテナンスユーザのトランザクションを無視するように Replication Agent を設定する方法

## 複写システムの概要

の詳細については、該当する Replication Agent のマニュアルを参照してください。

- レプリケートデータベース用の正しいプロファイルを使用して、Database #1 と Database #2 への Replication Server レプリケートデータベースコネクションが作成されている必要があります。接続プロファイルは、レプリケートデータベースの正しいファンクション文字列クラスとエラークラスを指定します。また、クラスレベル変換の定義とレプリケートデータベースオブジェクトの作成の複写サポートを含める場合があります。

# SAP の複写製品

SAP では、SAP の複写テクノロジーに基づいた、異機種データサーバや ASE 以外のデータサーバを含む複写システムを特にサポートする製品群を提供しています。

SAP の複写製品には次のものがあります。

- SAP Replication Server。SAP の高度な複写テクノロジーの中心部であり、特に SAP 複写システム内の ASE 以外のデータサーバをサポートするための機能が組み込まれています。
- SAP® Replication Server® Options。Replication Agent と、Enterprise Connect Data Access (ECDA) または ExpressConnect (for Oracle または SAP HANA データベース) で構成されます。
  - Replication Agent は、ASE 以外のプライマリデータベースから複写データを取得する方法を提供することで Replication Server をサポートします。このサポートは、DB2 UDB、Microsoft SQL Server、Oracle データサーバに対して提供されます。
  - ECDA データベースゲートウェイは、ASE 以外のさまざまなデータベースへのアクセスを提供し、これらのデータベースが SAP 複写システムでレプリケートデータベースとして機能できるようにすることで Replication Server をサポートします。
  - ExpressConnect (for Oracle または SAP HANA データベース) は、別のゲートウェイサーバを必要とせずに、Replication Server と Oracle または SAP HANA データベースとの間の直接通信を提供することで、Replication Server をサポートします。ExpressConnect for Oracle は Replication Server Options 15.5 以降でのみ使用できます。ExpressConnect for SAP HANA データベースは Replication Server 15.7.1 SP100 以降でのみ使用できます。
- Replication Agent for IBM DB2 UDB。メインフレーム上の IBM DB2 UDB からデータを複写します。

## Replication Server

---

Replication Server は、リモートの集中型データベースからではなく、ローカルのデータにアクセスできます。集中型データシステムと比べると、複写システムではシステムのパフォーマンスとデータ可用性が向上し、通信のオーバーヘッドが軽減されます。

Replication Server は、データを複写するための、費用効果が高く障害に強い (フォールトトレラントな) システムを提供します。Replication Server は、トランザクション (データのコピーではなく追加された変更作業) とストアドプロシージャ

の呼び出し (ストアードプロシージャの実行によって生じるオペレーションではありません) を複写するので、パフォーマンスが高い分散データ環境が実現すると同時に、システム全体で複写データのトランザクションの整合性が確保されます。

### Replication Server の動作

Replication Server は、ネットワーク上でトランザクションの整合性を維持しながら、レプリケートトランザクションを管理することにより、ネットワークを介してデータを配信します。

アプリケーション開発者とシステム管理者に、データとストアードプロシージャを複写するための、柔軟性のあるパブリッシュとサブスクライブモデルを提供します。

各プライマリサイトまたはレプリケートサイトの Replication Server は、ローカルデータサーバのデータ複写アクティビティを調整し、他のサイトの Replication Server とデータを交換します。

Replication Server は、次の処理を実行します。

- Replication Agent を介してプライマリデータベースからトランザクションを受信し、それらをデータに対するサブスクリプションを持っているサイトに分配する。
- 他の Replication Server からトランザクションを受け取り、ローカルデータベースに適用する。

Replication Server のシステムテーブルには、これらのタスクの実行に必要な情報が格納されます。システムテーブルには、複写データと次に示す複写オブジェクトの説明が格納されます。

- 複写定義およびサブスクリプション
- Replication Server ユーザ用のセキュリティレコード
- 他のサイトに対するルート指定情報
- ローカルデータベースのアクセス方法
- その他の管理情報

Replication Server のシステムテーブルは、Replication Server システムデータベース (RSSD) と呼ばれるデータベースに格納されます。

Replication Server で複写情報を管理するには、Replication Command Language (RCL) を使用します。RCL コマンドは、SQL コマンドに似ており、SAP の対話型 SQL ユーティリティ **isql** を使用して Replication Server で実行できます。RCL の詳細については、『Replication Server リファレンスマニュアル』を参照してください。



## パブリッシュとサブスクライブモデル

ユーザは、他 (レプリケートサイト) の Replication Server がサブスクリプションを作成する対象のプライマリサイトにデータがパブリッシュされます。

プライマリデータベースで実行されたトランザクションは Replication Agent によって検出され、ローカルの Replication Server に転送されます。次に、この Replication Server が、ネットワークを介して送信先サイトの Replication Server に情報を分配します。その後、これらの Replication Server により、リモートクライアントの要求に従って、レプリケートデータベースが更新されます。

プライマリデータは、Replication Server が他のデータベースに複写するデータのソースです。データをパブリッシュおよびサブスクライブするには、まず、複写定義を作成してプライマリデータのスコープとロケーションを指定します。複写定義には、テーブルの構造を記述します。データベース複写定義を使用すると、個々のテーブル、ファンクション、DDL を複写できます。テーブル複写定義には、テーブルの構造を記述し、テーブルに対して更新と削除を問い合わせるために使用するキーを宣言します。

複写定義を作成しただけでは、Replication Server がデータを複写するようにはなりません。さらに、複写定義に対するサブスクリプションを作成して、別のデータベースにデータを複写するように Replication Server に指示する必要があります。サブスクリプションは、次のような SQL の **select** 文に似ています。 **where** 句を追加して、ローカルデータベースに複写するテーブルのローを指定できます。

さまざまなオブジェクトをフィルタするために、1つのプライマリテーブルに対して複数の複写定義を作成できます。レプリケートテーブルでは、異なる複写定義のサブスクリプションを作成し、データの異なるビューを取得することができます。

複写定義またはパブリケーションのサブスクリプションを作成後に、Replication Server は、データのサブスクリプションを持つデータベースにトランザクションを複写します。

## 複写ファンクション

この場合、1つの複写ファンクションに多数の変更をカプセル化することによって、通常の方法よりもパフォーマンスを向上させることができます。

複写ファンクションはテーブル複写定義と関連付けられていないので、データを直接修正するかどうかに関係なくストアドプロシージャを実行できます。

データサーバによっては、Replication Server を使用して、ストアドプロシージャ呼び出しをデータベース間で非同期的に複写できます。

---

**注意：** Replication Server は、すべての種類のデータサーバでストアードプロシージャの複写をサポートしているわけではありません。特定のデータサーバにおけるストアードプロシージャの複写の詳細については、該当する Replication Agent のマニュアルを参照してください。

---

複写ファンクションを使用すると、別のデータベースでストアードプロシージャを実行できます。複写ファンクションを使用すると、次のことが実行できます。

- サブスクリプションを作成するサイトに、ストアードプロシージャの実行を複写する。
- データベースの実際の変更ではなく、ストアードプロシージャの名前とパラメータだけを複写することによって、パフォーマンスを向上させる。

Replication Server は、適用ファンクションと要求ファンクションの両方をサポートします。

- 「適用ファンクション」は、プライマリデータベースからレプリケートデータベースに複写されます。レプリケートサイトでファンクション複写定義のサブスクリプションを作成し、プライマリデータベースで複写対象としてマークを付けます。

---

**注意：** SAP HANA データベース用の適用ファンクションを複数作成する場合は、複写定義で指定するカラム順をすべてのレプリケート SAP HANA データベースで実行される複写ファンクション内の順序と同じにする必要があります。

---

- 要求ファンクションは、レプリケートデータベースからプライマリデータベースに複写されます。要求ファンクションのサブスクリプションはありません。レプリケートデータベースで、複写対象としてストアードプロシージャにマークを付けます。

## トランザクション管理

Replication Server では、トランザクション処理サービスの提供をデータサーバに依存しています。分散データの一貫性を保証するために、データサーバは、原子性や一貫性といったトランザクション処理規則に従う必要があります。

プライマリデータを格納するデータサーバは、分散データベースシステムで必要とされるほとんどの同時制御機能を提供します。トランザクションが、プライマリデータを持つテーブルの更新に失敗すると、Replication Server は、そのトランザクションを他のサイトには分配しません。トランザクションによってプライマリデータが更新されると、Replication Server は変更を分配し、障害が発生しないかぎり、データのサブスクリプションを作成したすべてのサイトで更新が正常に行われます。

## 他のシステムコンポーネントとの関係

Replication Server は、サーバまたはクライアントとして、複写システムの他のコンポーネントと対話します。

サーバとしての Replication Server は、以下からの接続をサポートしています。

- プライマリデータベースからのデータベースコマンドの送信に使用される Replication Agent
- 複写システム内で、メッセージ配信処理を分散させ、複写システムにスケールビリティを提供するための Replication Server
- 管理、データサーバの識別、メッセージのパブリケーションとサブスクリプションなどに使用されるユーザツールまたは管理ツール

クライアントとしての Replication Server は、以下に接続します。

- 外部 Adaptive Server Enterprise データベース上にあるかまたは内部 Embedded RSSD (ERSSD) である Replication Server システムデータベース (RSSD)
- ASE 以外のレプリケートデータベースに接続するためのデータベースゲートウェイ
- レプリケートデータベースに直接 (ExpressConnect (for Oracle または SAP HANA データベース) を使用している場合)

### Replication Server の通信プロトコル

Replication Server は、基本となる通信プロトコルとして Tabular Data Stream™ (TDS) を使用する Open Client および Open Server アプリケーションです。

Replication Server にサービスを要求するすべてのクライアントに、Open Client インタフェースが実装されている必要があります。これらのクライアントには、Replication Agent、システム管理ツール、**isql** などのユーザインタフェースツールが含まれます。

他の Replication Server やレプリケートデータサーバにメッセージを配信するクライアントとして、Replication Server は Open Client インタフェースを使用します。そのため、Replication Server がデータサーバにメッセージを送信する必要がある場合、そのデータサーバが TDS で実行されている Open Server インタフェースをサポートしているか、Replication Server とそのレプリケートデータサーバ間に Open Server/TDS ブリッジまたは「ゲートウェイ」アプリケーションがあることが必要です。

SAP IQ は、Replication Server にとって Open Server に見えるため、複写用に追加のゲートウェイソフトウェアは必要ありません。DB2 UDB と Microsoft SQL Server に複写するためのゲートウェイソフトウェアの形式は、ECDA データベースゲート

ウェイです。ECDA ゲートウェイは、Open Server/TDS からデータサーバの ODBC ドライバにブリッジします。Replication Server の設定は、使用するゲートウェイによって異なります。

ExpressConnect を使用した Oracle または SAP HANA データベースへの複写では、追加のゲートウェイを必要としません。ExpressConnect はネイティブな Oracle または SAP HANA データベースコネクティビティを使用するため、ExpressConnect を使用した Replication Server はレプリケートデータベースに直接接続できます。

### Replication Server のユーザ ID とパーミッション

Replication Server には、いくつかの異なるユーザ ID が必要です。他のコンポーネント (またはユーザ) が Replication Server にアクセスするために必要なユーザ ID もあれば、Replication Server が複写システム内の他のコンポーネントにアクセスするために必要なものもあります。

ユーザ ID は、Replication Server の **create connection** コマンドを使用して、Replication Server で定義します。

---

**注意:** 複写システムの設定によっては、下記のリストにある一部のユーザ ID が不要な場合もあります。たとえば、プライマリデータベースとレプリケートデータベースに別の Replication Server を使用している場合、プライマリ Replication Server には、レプリケートデータベースにアクセスするためのユーザ ID は必要ありません。

---

Replication Server で定義されるユーザ ID は、次のとおりです。

- Replication Agent ユーザ - プライマリ Replication Server にログインするために Replication Agent が使用します。このユーザ ID には、LTL インタフェースを介してデータベースコマンドを配信するために、**connect source** パーミッションが必要です。
- Replication Server ユーザ - Replication Server にログインし、メッセージを転送するために、他の Replication Server が使用します。このユーザ ID には、RCL インタフェースを介してデータベースコマンドを転送するために、**connect source** パーミッションが必要です。
- SysAdmin ユーザ - 管理作業を実行するために、システム管理者またはシステム管理ツールが使用します。作業によっては、このユーザ ID には **sa**、**create object**、または **primary subscribe** パーミッションが必要です。
- メンテナンスユーザ - レプリケートデータサーバにメッセージを配信するために、Replication Server が使用します。このユーザ ID には、配信されるメッセージのマップ先のコマンドをプライマリデータベースに対して実行するために必要な、レプリケートデータサーバでのパーミッションが必要です。メンテナンスユーザが実行する作業は複写されません。

- レプリケートユーザ - プライマリデータサーバにメッセージを配信するために、レプリケート Replication Server が使用します。「要求」メッセージ、つまり、プライマリデータサーバに配信するために選択されたレプリケートデータサーバからのメッセージの配信については、Replication Server は、レプリケートデータベースでコマンドを実行するユーザのユーザ ID を使用します。このユーザ ID には、配信されるメッセージのマップ先のコマンドを実行するために必要な、プライマリデータサーバでのパーミッションが必要です。
- RSI ユーザ - 他の Replication Server にログインして、配信されるメッセージを転送するために Replication Server が使用します。このユーザ ID には、レプリケート Replication Server での **connect source** パーミッションが必要です。
- RSSD ユーザ - 操作可能データを管理する Replication Server システムデータベース (RSSD) にログインするために、Replication Server が使用します。このユーザ ID には、オブジェクトの作成と削除、プロシージャの実行、テーブルの問い合わせと更新を行うための RSSD でのフルコントロールが必要です。

### **Replication Agent との関係**

Replication Server は、レプリケートデータサーバのニーズに合わせて拡張可能 (カスタマイズ可能なファンクション文字列とエラー処理、カスタムデータ型定義、データ型間の変換) ですが、プライマリデータサーバについてのサポートは限られています。

プライマリデータサーバの Replication Server インタフェースは、独自の Log Transfer Language (LTL) です。プライマリデータサーバからのトランザクションがプライマリ Replication Server に配信されるには、LTL への変換が必要です。そのため、プライマリデータサーバのサポートは、プライマリデータベースのオペレーションのために LTL への変換を行う Replication Agent が SAP から提供されているものに限られます。

プライマリ側とレプリケート側の Replication Server のインタフェースは、TDS で実行されている基本の Open Client/Open Server インタフェースによってサポートされます。

### **LTM ロケータの更新**

プライマリ Replication Server は、「ロケータ」値 (LTM ロケータ) を保持していません。これにより、プライマリ Replication Server ですべてのデータが正常に受け取られているトランザクションログ内の最後のポイントを特定します。

Replication Agent は、Replication Server コネクションにこの値を定期的に要求して、トランザクションログ内の位置を特定します。その後、これを使用して、ログから古いデータを解放または削除できる位置を特定できます。

LTM ロケータの更新を要求する頻度の決定には、パフォーマンスとの兼ね合いが必要です。Replication Server に LTM ロケータ値を頻繁に問い合わせると、複写の

速度が遅くなる可能性がある一方 (Replication Agent は、LTM ロケータ値を要求して受け取るために必要な時間、LTL コマンドの送信を停止しなければならない)、データをプライマリデータベースのトランザクションログから解放できる機会が増えます。再起動時には、Replication Agent は、Replication Server から最後の LTM ロケータ値を受け取った後にログに保持されているすべてのデータを、再送信しなければなりません。

一般に、複写のスループットのパフォーマンスを優先する場合は、十分なログリソースを確保し、ログトランケーションの頻度と LTM ロケータ値の取得の頻度を少なくしてください。ログリソースが少ない場合は、LTM ロケータ値の取得とトランケーションをより頻繁に行わなければならない可能性があります。

LTM ロケータの使用の詳細については、該当する Replication Agent のマニュアルを参照してください。

### LTL の生成

Replication Server に送信される情報のバイト数は、複写システムのパフォーマンスに直接影響します。Replication Server が受け取るデータやコマンドが多いほど、必要な作業が増え、処理に時間がかかります。

また、データが多くなると、さらに多くのネットワークリソースが必要になります。Replication Agent には、この影響を最小限に抑えるために使用できるさまざまな設定オプションがあります。

- RSSD の使用。RSSD から複写定義を読み取ることによって、Replication Agent は複写定義で指定されたカラム順序でカラムデータを送信できます。これにより、Replication Server は、カラム情報を処理する前にソートを行う必要がなくなります。さらに、カラム名がデータとともに送信されないため、必要な情報のバイト数が減少します。
- 最少カラムの送信。テーブルで更新オペレーションを行うときに、一部のカラムのみを変更すればよい場合があります。変更されたカラムのみの更新前イメージと更新後イメージを送信することによって、Replication Agent が送信する情報が少なくなります。

---

**注意：**レプリケートデータベースのデータにカスタムファンクション文字列が含まれる場合は、最少カラムを使用しないでください。

---

- バッチモード。Replication Agent は、Replication Server の限られた量の管理作業を行う LTL にトランザクションを「まとめる」必要があります。バッチモードでは、Replication Agent は同一の管理コマンドセットに複数のコマンドをまとめることができるため、ネットワークと Replication Server によって生成および処理される LTL が全体的に削減されます。

バッチモードに加えて、ほとんどの Replication Agent には “batch timeout” パラメータがあります。このパラメータを使用すると、指定された時間 Replication Agent が待機してもバッチを満たす他のトランザクションを受け取らなかった

場合に、バッチの一部が Replication Server に送信されるようにすることができます。

---

**注意：** Replication Server のユーザ定義データ型 (UDD) 変換をカラムレベルまたはクラスレベルで使用している場合は、Replication Agent のバッチモードを使用しないでください。

---

- オリジン時間。Replication Server に送信される各トランザクションには、オリジンキュー ID があります。このオリジンキュー ID に、プライマリデータベースでトランザクションがコミットされた時間が含まれている場合があります。Replication Agent からオリジン時間が送信されなかった場合、処理作業は多少減りますが、Replication Server に送信される LTL の量は同じです。

LTL の出力に影響する Replication Agent の設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### rs\_ticket

一部の Replication Agent では、**rs\_ticket** トランザクションを起動できます。

トランザクションは、Replication Server のパフォーマンス、モジュールのハートビート、複写の正常性、およびテーブルレベルのクワイズについてのデータを提供します。『Replication Server リファレンスマニュアル』を参照してください。

## データベースコネクション

Replication Server は、プライマリデータベースとレプリケートデータベースを識別する「コネクション」と、他の Replication Server を識別する「ルート」を使用して、複写システム内の他のコンポーネントを追跡します。

Replication Server は本来、Adaptive Server Enterprise データベースの複写を目的として設計されているので、Replication Server のコネクションの定義は、SAP の標準である *server name.database name* の形式に準拠しています。たとえば、ASE1 という名前の Adaptive Server のデータベース PUBS への Replication Server コネクションの名前は ASE1.PUBS となります。

ASE 以外のプライマリデータサーバに接続するために、Replication Server は、ASE 以外のプライマリデータベースの代わりに、Replication Agent からコネクションを受け付けます。レプリケートデータベースの場合、Replication Server は、ECDA データベースゲートウェイに接続します。ECDA データベースゲートウェイは、続いて ASE 以外のレプリケートデータサーバに接続します。Oracle または SAP HANA データベースの場合、Replication Server は、ExpressConnect を使用してレプリケートデータサーバに直接接続することもできます。Replication Agent、ECDA ゲートウェイ、および ExpressConnect はデータサーバではないので、これらのコンポーネントの Replication Server コネクションのプロパティは、データベースサーバコネクションの場合とは意味が異なる場合があります。

1つの Replication Server コネクションで、単方向または双方向のデータフローをサポートできます。データの受信は、Replication Agent ユーザスレッドを介した Replication Server コネクションによって行われます。データの送信は、データサーバインタフェース (DSI) スレッドを介した Replication Server コネクションによって行われます。各 Replication Server コネクションでは、送信データフローのみ (DSI スレッド経由)、または受信および送信データフロー (Replication Agent User スレッドおよび DSI スレッド経由) をサポートできます。

### **Replication Agent ユーザスレッド**

Replication Server は、複写されるすべてのデータ変更オペレーションまたはトランザクションを、プライマリデータサーバから、そのデータサーバのデータベースコネクションの Replication Agent User スレッドを介して受け取ります。

複写されるトランザクションを提供するすべてのプライマリデータベースが、有効な Replication Agent ユーザスレッドを持つ Replication Server データベースコネクションによって表されている必要があります。

プライマリデータベースが Adaptive Server 内にある場合、Replication Server はそのデータベースと直接コネクションを確立します。プライマリデータベースが ASE 以外のデータサーバにある場合は、プライマリデータベースに代わって、別の Replication Agent コンポーネントが Replication Agent ユーザスレッドコネクションを使用して Replication Server と通信します。

---

**注意：** Replication Server は、コネクションの Replication Agent ユーザスレッドには接続しません。Replication Agent ユーザスレッドとの通信を開始できるエンティティは、プライマリデータサーバと Replication Agent のみです。

---

Replication Agent ユーザスレッドでは、プライマリデータサーバまたは Replication Agent がクライアント、プライマリ Replication Server がサーバです。

### **DSI スレッド**

Replication Server コネクションの DSI スレッドは、Replication Server によって複写トランザクションが配信される場所にあります。

複写トランザクションを受け取ることが予想されるすべてのレプリケートデータベースが、有効な DSI スレッドを持つ Replication Server コネクションによって表されている必要があります。

レプリケートデータベースが Adaptive Server 内にある場合、Replication Server はそのデータベースと直接コネクションを確立します。レプリケートデータベースが SAP 以外のデータサーバにある場合は、Replication Server は次を使用して通信します。



- ECDA データベースゲートウェイ (コネクションの DSI スレッドを介して)、または、
- ExpressConnect (Oracle または SAP HANA レプリケートデータベースとのコネクションを直接確立)

---

**注意：** レプリケートデータサーバまたはデータベースゲートウェイは、コネクションの DSI スレッドには接続しません。DSI スレッドとの通信を開始できるエンティティは、Replication Server だけです。

---

DSI スレッドでは、Replication Server がクライアント、レプリケートデータサーバまたはデータベースゲートウェイがサーバです。

### メンテナンスユーザの目的

メンテナンスユーザは複写テーブルでローの挿入、削除、および更新を行い、複写ストアードプロシージャを実行します。データベース所有者 (またはシステム管理者) は、メンテナンスユーザがこれらの作業を実行するために必要なパーミッションを付与する必要があります。

複写データを更新する場合、Replication Server はレプリケートデータサーバにメンテナンスユーザとしてログインします。Adaptive Server レプリケートデータベースでは、**rs\_init** によって Replication Server メンテナンスユーザのユーザ ID が自動的に作成され、レプリケートデータベースにユーザが追加されます。

メンテナンスユーザ ID とパスワードは、レプリケートデータベースに対する Replication Server の **create connection** コマンドによって、Replication Server で自動的に定義されます。データサーバでメンテナンスユーザ ID のパスワードを変更する場合、Replication Server の **alter connection** コマンドを使用して Replication Server コネクションのパスワードを変更できます。

また、Replication Server メンテナンスユーザには、rs\_lastcommit および rs\_info システムテーブルと、それらのテーブルを使用するすべてのストアードプロシージャにアクセスするためのパーミッションが必要です。

**rs\_init** は、ユーザテーブルとストアードプロシージャに対するデータベースパーミッションをメンテナンスユーザに付与しません。複写テーブルに対するトランザクションの複写または複写ストアードプロシージャの実行の複写を行うには、事前に複写テーブルと複写ストアードプロシージャに対するデータベースパーミッションを付与しておく必要があります。データベースで複写されるテーブル、および複写のために実行されるストアードプロシージャごとに、次のコマンドを実行してください。

```
grant all on table_name to maint_user
```

また、すべてのレプリケートオブジェクトに対する必要な権限がデータベース管理者ロールにある場合、メンテナンスユーザ ID (*maint\_user*) をこのロールに割り当てることもできます。

## DDL ユーザの目的

Replication Agent for Microsoft SQL Server や Replication Agent for Oracle では、プライマリデータベースで入力された DDL コマンドをサブスクライバデータベースに複写できます。

この機能は、プライマリデータサーバとレプリケートデータサーバが同じである場合 (たとえば、Oracle と Oracle) にのみサポートされます。詳細については、『Replication Agent 管理ガイド』を参照してください。

## テーブルレベルの引用符付き識別子の複写

オブジェクトに対してテーブルレベルの `dsi_quoted_identifier` を設定することにより、引用符付きオブジェクトを正しく複写するよう Replication Server に指示できます。

ある接続でテーブル複写を実行するとします。接続レベルの `dsi_quoted_identifier` は `on` に設定されています。その後、引用符付き識別子が出現するテーブルを複写するために、同じ接続でデータベース複写を設定します。これには、接続レベルの `dsi_quoted_identifier` を `always` に設定する必要があります。テーブル名とカラム名がすべて引用符で囲まれているため、既存のテーブル複写は失敗する可能性があります。既存のテーブル複写を正常に実行するためには、そのテーブルに対してテーブルレベルの `dsi_quoted_identifier` を `on` に設定します。

テーブルレベルの `dsi_quoted_identifier` の設定は、既存の接続レベルの `dsi_quoted_identifier` の設定より優先されます。また、接続レベルの設定は、サーバレベルの `dsi_quoted_identifier` の設定より優先されます。

プライマリテーブルに引用符付き識別子がある場合に、複写が正常に実行されるようにするには、`quoted` 句を指定してテーブル複写定義を作成し、接続レベルの `dsi_quoted_identifier` を `on` に設定する必要があります。

たとえば、プライマリテーブル “test quote” に引用符付き識別子がある場合でも、複写が実行されるようにするには、`quoted` 句を指定してテーブル複写定義 `repdef1` を作成し、接続レベルの `dsi_quoted_identifier` を `on` に設定します。

```
create table "test quote" ("coll###@" int, col2 char(10), primary
key("coll###@"))
go

create replication definition repdef1
...
with all tables named "test quote" quoted
("coll###@" int quoted, col2 char(10))
primary key("coll###@")
```

```
alter connection to LON_DS.rdb
set dsi_quoted_identifier to 'on'
```

プライマリテーブルに引用符付き識別子がある場合で、テーブル複写定義に対して **create subscription** に **init replicate table** オプションを指定して使用するときは、**dsi\_quoted\_identifier** を **always** に設定し、次の条件を確認してください。

- テーブル複写定義で指定するプライマリテーブル名がプライマリデータベース内のテーブル名と同じである。
- Replication Agent パラメータ **ltd\_character\_case** が **asis** に設定されている。これにより、オブジェクト名が、複写定義で指定されているとおりに大文字と小文字が区別されて送信されます。
- テーブル複写定義で指定するカラム名がプライマリデータベース内のカラム名と同じである。
- スキーママッピングで指定する所有者名がプライマリデータベース内の所有者名と同じである。
- データベース複写の作成後にテーブル複写定義を削除する場合は、テーブル複写定義内のすべての識別子 (テーブル名、カラム名など) を引用符で囲む。

たとえば、プライマリテーブル “test quote” に引用符付き識別子がある場合でも、複写が実行されるようにするには、すべてのオブジェクト (テーブルおよびカラム) に対して **quoted** 句を指定してテーブル複写定義 *repdef1* を作成し、**dsi\_quoted\_identifier** を **always** に設定します。

```
create table "test quote" ("coll%##@" int, col2 char(10), primary
key("coll%##@"))
go

create replication definition repdef1
...
with all tables named "test quote" quoted
("coll%##@" int quoted, col2 char(10) quoted)
primary key('coll%##@')

alter connection to LON_DS.rdb
set dsi_quoted_identifier to 'always'
```

参照:

- 『Replication Server 管理ガイド 第 1 巻』の「引用符付き識別子」
- 『Replication Agent リファレンスマニュアル』>「設定パラメータ」>「設定パラメータリファレンス」>「ltd\_character\_case」

## データ型、データ型定義、および制限付きデータ型

特定のデータサーバのデータ型のデータ型定義は、データ型クラスにまとめられます。

データ型定義 (ユーザ定義データ型) の詳細については、「データ型の変換とマッピング (285 ページ)」を参照してください。

### *制限付きデータ型*

`rs_address` データ型は、カラムレベル変換やクラスレベル変換の変換元データ型および変換先データ型として使用できません。

## ASE 以外のデータサーバのエラークラス

Replication Server では、サポートされている ASE 以外のすべてのレプリケートデータサーバについてエラークラスを提供しています。

ASE 以外のエラークラスは Replication Server によって作成され、さまざまな ASE 以外のエラークラスに対してエラーアクションが定義されます。適切な接続プロファイルを使用することにより、対応するエラークラスを含む ASE 以外のデータベースへのコネクションを作成できます。

## ASE 以外のデータサーバのファンクション文字列クラス

Replication Server では、サポートされている ASE 以外のすべてのレプリケートデータサーバについて、ファンクション文字列クラスとそれに関連付けられたファンクション文字列を提供しています。

ASE 以外のデータベースの親クラスから派生したファンクション文字列クラスを作成したら、Replication Server システムデータベース (RSSD) 内の `rs_translation` テーブルを手動で更新して、派生クラスが親クラスからクラスレベル変換を継承するようにします。

『Replication Server 管理ガイド 第2巻』の「ファンクション文字列継承」を参照してください。

### 派生ファンクション文字列クラス用の `rs_translation` テーブルの更新

この例では、作成した派生クラスが親クラスからクラスレベル変換を継承するように、Replication Server システムデータベース (RSSD) 内の `rs_translation` テーブルを更新する方法を示します。

1. Replication Server で派生ファンクション文字列クラスを作成します。  
たとえば、Oracle 用の `private_class_for_oracle` 派生ファンクション文字列クラスを作成するには、次のように入力します。

```
create function string class private_class_for_oracle
set parent to rs_oracle_function_class
go
```

派生クラスを作成するときには必ず、ASE 以外のデータベースの適切な基本クラスを親クラスとして指定してください。『Replication Server 管理ガイド 第2巻』の「ファンクション文字列継承」を参照してください。

2. SQL **select** 文を使用して **rs\_classes** Replication Server システムテーブルから **private\_class\_for\_oracle** のクラス ID を取得します。

```
select classid from rs_classes where
classname='private_class_for_oracle'
go
```

次のような情報が表示されます。

```
classid
-----
0x0100006801000065

(1 row affected)
```

3. **rs\_oracle\_function\_class** 用に定義されたクラスレベル変換を取得します。

```
select * from rs_translation where classid = 0x0000000001000007
go
```

次のような情報が表示されます (わかりやすいように、4 カラムずつ 2 段に分けています)。

| prsid | classid            | type | source_dtid        |
|-------|--------------------|------|--------------------|
| 0     | 0x0000000001000007 | D    | 0x0000000000000001 |
| 0     | 0x0000000001000007 | D    | 0x000000000000000c |
| 0     | 0x0000000001000007 | D    | 0x000000000000000d |
| 0     | 0x0000000001000007 | D    | 0x000000000000000e |
| 0     | 0x0000000001000007 | D    | 0x000000000000000f |
| 0     | 0x0000000001000007 | D    | 0x0000000000000013 |
| 0     | 0x0000000001000007 | D    | 0x000000000000001b |
| 0     | 0x0000000001000007 | D    | 0x000000000000001c |

| target_dtid        | target_length | target_status | rowtype |
|--------------------|---------------|---------------|---------|
| 0x0000000000010202 | 0             | 0             | 0       |
| 0x0000000000010200 | 19            | 0             | 0       |
| 0x0000000000010200 | 19            | 0             | 0       |
| 0x0000000000010205 | 136           | 0             | 0       |
| 0x0000000000010205 | 136           | 0             | 0       |
| 0x0000000000010202 | 0             | 0             | 0       |
| 0x0000000000010201 | 9             | 0             | 0       |
| 0x0000000000010213 | 8             | 0             | 0       |

(8 row affected)

4. 手順 3 で返されたローごとに、手順 3 で取得したクラスレベル変換値を **private\_class\_for\_oracle** 派生ファンクション文字列クラスに挿入するように SQL **insert** 文を構築します。

次のテンプレートを使用します。

```
insert into rs_translation
(prsid,
classid,
type,
source_dtid,
target_dtid,
target_length,
target_status,
rowtype)
values (0,
classid from step 2,
'D',
source_dtid from output of 3,
target_dtid from output of step 3,
target_length from output of step 3,
target_status from output of 3,
0)
```

『Replication Server リファレンスマニュアル』の「rs\_translation」を参照してください。

たとえば、**private\_class\_for\_oracle** 派生ファンクション文字列クラスの最初のローの **insert** 文は次のようにする必要があります。

```
insert rs_translation values (0, 0x0100006801000065, 'D',
0x0000000000000000c, 0x00000000000010200, 19, 0, 0)
```

5. 手順 4 で構築した **insert** 文を実行します。

```
insert rs_translation values (0, 0x0100006801000065, 'D',
0x0000000000000000c, 0x00000000000010200, 19, 0, 0)
insert rs_translation values (0, 0x0100006801000065, 'D',
0x0000000000000000d, 0x00000000000010200, 19, 0, 0)
insert rs_translation values (0, 0x0100006801000065, 'D',
0x00000000000000001, 0x00000000000010202, 0, 0, 0)
insert rs_translation values (0, 0x0100006801000065, 'D',
0x000000000000000013, 0x00000000000010202, 0, 0, 0)
insert rs_translation values (0, 0x0100006801000065, 'D',
0x0000000000000000E, 0x00000000000010205, 136, 0, 0)
insert rs_translation values (0, 0x0100006801000065, 'D',
0x0000000000000000f, 0x00000000000010205, 136, 0, 0)
insert rs_translation values (0, 0x0100006801000065, 'D',
0x00000000000000001b, 0x00000000000010201, 9, 0, 0)
insert rs_translation values (0, 0x0100006801000065, 'D',
0x00000000000000001c, 0x00000000000010213, 8, 0, 0)
```

6. classid 用の rs\_translation ローがすでに RSSD に存在する場合は、Replication Server を再起動します。そうでない場合は、Replication Server を再

起動することによって、派生ファンクション文字列クラスの適切な実行を保証することができます。

`rs_translation` を更新したら、対応する ASE 以外のデータベースへのコネクションに派生ファンクション文字列クラスを使用できます。

## オブジェクトのパブリケーションとサブスクリプションの制限

SAP 複写システムでのオブジェクトのパブリケーションとサブスクリプションには制限事項があります。

制限事項を次に示します。

- ASE 以外のプライマリデータベースの複写定義でカラムを宣言するときには、プライマリデータベースのカラムのデータ型と一致する Replication Server データ型を使用します。一致する Replication Server のネイティブデータ型がない場合は、プライマリデータベースのデータ型と一致するデータ型定義を探します。「データ型の変換とマッピング (285 ページ)」を参照してください。
- カラムレベル変換に関連するカラムに基づいた述部 **where** 句を使用してサブスクリプションを作成する場合、「宣言された」(つまり変換前の)形式で述部の値を指定します。『Replication Server 管理ガイド 第 1 巻』>「複写テーブルの管理」>「HDS を使用したデータ型の変換」を参照してください。

## Replication Agent

Replication Agent は、SAP 複写システム内で ASE 以外のデータサーバをプライマリデータサーバとしてサポートすることによって Replication Server の機能を拡張します。

Replication Agent はプライマリデータに加えられた変更を検出し、RCL (Replication Control Language) のサブセットであるログ転送言語 (LTL) を使用して、プライマリデータの変更内容をプライマリ Replication Server に送信します。

## Replication Agent の動作

Replication Agent は、プライマリデータベースのトランザクションログから情報を取得し、プライマリ Replication Server に対応するようにフォーマットする Replication Server クライアントです。

最初に、Replication Agent 内で必要なプライマリテーブルとストアードプロシージャを複写用にマーク付けします。

Replication Agent は、次のように動作します。

1. Replication Server にログインします。
2. **connect source** コマンドを送信してセッションをログ転送元として特定し、トランザクション情報が転送されるデータベースを指定します。
3. Replication Server から、データベースのメンテナンスユーザ名を取得します。
4. Replication Server にデータベースのセカンダリトランザクションポイントを要求します。
5. トランザクションログから、レコードを (セカンダリトランザクションポイントに続くレコードを起点として) 取得し、情報をログ転送言語 (LTL) コマンドにフォーマットします。

### **Replication Agent コネクション**

Replication Agent は、Replication Server にデータを送信します。Replication Agent は Replication Server にログインして、Replication Server コネクションの Replication Agent ユーザスレッドに接続し、そのコネクション上で Replication Server と通信します。

次に、Replication Agent コネクションの影響を示します。

- Replication Server にログインするために Replication Agent が使用する有効なユーザ ID が、Replication Server で定義されている必要があります。
- Replication Agent ユーザ ID に、Replication Server での **connect source** パーミッションが付与されている必要があります。 **connect source** パーミッションによって、Replication Agent は Replication Agent User スレッドのみで有効なコマンドを送信できます。
- Replication Agent は、このユーザ ID とそのパスワードを記録する必要があります。
- Replication Agent は、正しい Replication Agent ユーザスレッドを特定して接続するために、Replication Server コネクション定義のサーバおよびデータベース部分を記録する必要があります。
- **user\_name** と **password** (Replication Server の **create connection** コマンドで定義) は、プライマリデータベースで有効なユーザ ID とパスワードである場合があります。

---

**注意：** Replication Agent for Oracle、Microsoft SQL Server、および IBM DB2 UDB for UNIX には有効な **user\_name** と **password** が指定され、プライマリデータベースで該当のユーザが見つからない場合は、エラーを報告する必要があります。

---

Replication Agent は、コネクションの **user\_name** がプライマリデータベースにあるかどうかを検証します。しかし、Replication Server は、DSI スレッドが使用されるかどうか (またはその時期) を認識していません。そのため、DSI ス



レッドがアクティブな場合は、ユーザ ID とパスワードが有効でなければなりません。

---

**注意：**有効なプライマリデータベースのユーザ ID の必要条件は、Replication Agent によって異なります。一部の Replication Agent は、Replication Server コネクション上の有効なユーザ ID を必要としません(チェックも行いません)。

---

### **Interfaces ファイル**

Replication Agent と Replication Server 間の対話のために必要なものは Replication Server を識別する interfaces ファイルのエントリだけです。

Replication Agent for DB2 UDB には、interfaces ファイルは必要ありません。Replication Server と RSSD のロケーションは、必要に応じて LTMCFG ファイル内に記録されます。

Replication Agent (UNIX および Windows プラットフォーム上の DB2 UDB 用、Microsoft SQL Server 用、Oracle 用)には、interfaces ファイルエントリは必要ありません。これは、設定パラメータに Replication Server のホスト名とポート番号が記録されるためです。

### **Replication Agent メンテナンスユーザの処理**

Replication Agent が Replication Server コネクションに接続する際に、Replication Agent はメンテナンスユーザ ID を要求し、プライマリデータベースにそのユーザ ID があるかどうかを検証する場合があります。

この検証では、Replication Server コネクションで定義されているメンテナンスユーザ ID が、そのコネクションが表すデータベースに対して有効でなければなりません。そのコネクションがプライマリトランザクションのみ、レプリケートトランザクションのみ、またはその両方のどれに使用されるかは関係ありません。

Replication Agent は、プライマリデータベースにログインするときにメンテナンスユーザ ID を使用しません。ユーザ ID があるかどうかを検証する以外に、Replication Agent がメンテナンスユーザ ID に対して行う唯一の参照は、メンテナンスユーザによって作成されたプライマリデータベーストランザクションをフィルタすることです。

Replication Agent は、プライマリデータベースに 1 つのトランザクションが何回も適用されないようにするために、メンテナンスユーザのトランザクションをフィルタします。双方向の複写スキームでは、同じデータベース(プライマリとレプリケートの両方の役割を果たす可能性があります)に対して双方向に複写を実行できます。プライマリトランザクションがレプリケートデータベースに適用される場合、適用するユーザ ID はレプリケートデータベースのメンテナンスユーザです。レプリケートデータベースでトランザクションをスキャンする Replication Agent

は、Replication Server メンテナンスユーザによって適用されたトランザクションを無視することで、それらのトランザクションが再度送信され、プライマリデータベースに適用されないようにします。

Replication Agent は、プライマリデータベースで定義されているユーザ ID (DB2 の場合は DB2 ログファイルにアクセスできるユーザ ID) を使用してデータベースにアクセスします。このユーザ ID は、Replication Server コネクションで定義されているメンテナンスユーザとは異なります。プライマリデータベースへのアクセスに使用される Replication Agent ユーザ ID には、複写トランザクションを適用するために定義されたメンテナンスユーザとは異なる役割と目的があります。

また、Replication Agent には、Replication Agent を管理するために使用される別のユーザ ID が定義されている場合もあります。このユーザ ID も、レプリケートトランザクションを適用する Replication Server メンテナンスユーザとは異なります。

Replication Agent では、次の 3 つのユーザを使用できます。

- Replication Agent がプライマリデータサーバにログインし、プライマリ複写オブジェクトの操作やデータベーストランザクションログの読み取りを行うために使用する、プライマリデータベースで定義されるユーザ ID。
- Replication Agent にログインし、Replication Agent コマンドの発行や Replication Agent パラメータの設定を行うことができるユーザ ID。
- プライマリデータベースで定義され、プライマリ Replication Server コネクションに記録されるメンテナンスユーザ ID。Replication Agent は、Replication Server に代わってこのユーザ ID を検証します。また、このユーザ ID によって作成されるトランザクションを無視するように、Replication Agent を設定できます。

## DDL ユーザの処理

DDL ユーザは、DDL 複写を使用できる場合にプライマリデータベースで定義されます。

このユーザ名は、Replication Agent によって送信されるすべての DDL コマンドに LTL で含められます。Replication Server の DSI スレッドは、このユーザ名を使用して DDL をレプリケートデータベースに適用します。

### 直接ロードマテリアライゼーションのサポート

一部の Replication Agent は、Replication Server がプライマリデータベースへの問い合わせに必要とする接続を提供することで、Replication Server での直接ロードマテリアライゼーションを容易にします。

「直接ロードマテリアライゼーション (315 ページ)」を参照してください。プライマリデータベースで直接ロードマテリアライゼーションがサポートされているかどうかを判断するには、このドキュメントでプライマリデータベースに関する適切な章を参照してください。

## ASE 以外の Replication Agent

SAP では、Replication Agent for DB2 UDB、Replication Agent など、ASE 以外の Replication Agent を提供しています。

### *Replication Agent for DB2 UDB*

Replication Agent for DB2 UDB は、IBM z/OS プラットフォームで実行されている DB2 UDB サーバに、プライマリデータサーバのサポートを提供します。

Replication Agent for DB2 UDB は、複写システムで次のように機能します。

- プライマリデータサーバは DB2 UDB で、IBM z/OS においてサブシステムとして実行される。トランザクションログは DB2 ログ。
- Replication Agent for DB2 UDB は、IBM z/OS では、開始されたタスクまたはジョブとして実行される。DB2 ログを読み込んで、1つ以上の DB2 サブシステム用に複写するようにマーク付けされているテーブルについて、関連するアクティブな DB2 ログエントリとアーカイブログエントリを取得する。このデータを、TCP/IP 通信プロトコルを使用して Replication Server に転送する。

DB2 データサーバは、DB2 テーブル内のローが変更されると、そのすべての変更をログに記録します。トランザクションログに書き込まれる情報には、変更前後のデータのコピーが含まれます。DB2 では、これらのレコードは“undo”レコードおよび“redo”レコードとして使用されます。制御レコードは、**commits** と **aborts** 用には書き込まれます。これらのレコードは、**commit** オペレーションと **rollback** オペレーションに変換されます。

DB2 のログは、一連のデータセットで構成されます。SAP Log Extract は、これらのデータセットを使用して DB2 データの変更内容を識別します。DB2 は、変更が行われると変更レコードをアクティブなログに書き込むので、SAP Log Extract は、変更レコードが入力された直後にログレコードを処理できます。

### *Replication Agent*

Replication Agent は、Linux、UNIX、Microsoft Windows プラットフォーム上の DB2 UDB、Microsoft SQL Server、または Oracle プライマリデータベースのデータベーストランザクションログを読み取る製品です。

Replication Agent は、Java プログラミング言語で実装されています。Replication Agent をインストールすると、Replication Agent ホストマシンとして指定されているコンピュータに Java Runtime Environment (JRE) がインストールされます。

Replication Agent は、すべての通信に JDBC (Java Database Connectivity) プロトコルを使用します。Replication Agent は、SAP JDBC ドライバ (SAP® jConnect™ for JDBC) の 1 つのインスタンスを使用して、Open Client および Open Server アプリケーション (プライマリ Replication Server を含む) へのすべてのコネクションを管

理します。プライマリデータサーバの場合、Replication Agent は、適切な JDBC ドライバを使用してプライマリデータベースに接続します。

## Enterprise Connect™ Data Access

---

ECDA (Enterprise Connect Data Access) 製品は、DB-Library™ および CT-Library アプリケーションプログラミングインタフェース (API)、JDBC および ODBC (Open Database Connectivity) プロトコルをサポートする、Open Server ベースのソフトウェアエージェントウェイです。

ECDA 製品は、メインフレームおよび ASE 以外のデータソースへのアクセスを可能にするデータベースミドルウェアアプリケーションの基本要素となります。

ECDA 製品は、以下のサービスを提供します。

- ASE 以外のデータソースへのアクセスを提供するアクセスサービス
- サーバ側のシステム管理を (DirectConnect Manager を介して) 提供する管理サービス

### ECDA の動作

すべての ECDA オプションは、ASE 以外のデータサービスへの基本的な接続ティビティを実現します。特に、アクセス管理、コピー管理、リモートシステム管理を提供します。

各 ECDA オプションは、1 つの DirectConnect サーバおよび 1 つ以上のアクセスサービスライブラリで構成されています。サーバは、サービスライブラリが動作するフレームワークを提供します。各アクセスサービスライブラリは、このサーバから DB2 UDB、Microsoft SQL Server、Oracle などの特定のターゲットデータベースのデータにアクセスします。

各アクセスサービスライブラリには、設定プロパティの特定のセットである 1 つ以上のアクセスサービスが格納されています。アクセスサービスは、Replication Server とターゲットデータベース間でデータを転送します。

DirectConnect サーバは、言語イベントや RPC などの受信クライアント接続を受信して検証し、許可します。これらのイベントは、データ型変換、ネットワークのコネクティビティ、SQL 変換など、ターゲット固有のコネクティビティ機能を実現するアクセスサービスを介して、ターゲットデータソース (レプリケートデータベース) にルート指定されます。

## **Interface ファイル**

interface ファイルには、ラベルのリストが格納されています。ラベルとは通常、それぞれ対応するホスト名とポート番号を持つサーバ名のことです。サーバ名で表されたサーバは、そのホストとポートでログイン要求を「受信」します。

Replication Server は、Open Server アプリケーションです。別の Open Server アプリケーションの場所 (ホストとポート番号) を特定する方法としては、ファイル内の場所を探す方法が推奨されます。

ECDA データベースゲートウェイと Replication Server との対話では、interface ファイルが重要です。Replication Server は Replication Server コネクションにおいてサーバ名で表されたサービスにログインしようとするので、そのサービス名が Replication Server の interfaces ファイル内になければなりません。また、その interface ファイルエントリは ECDA ゲートウェイ設定ファイルのエントリにもサービス名として存在している必要があります。

1 つの ECDA は、1 つまたは複数の異なるデータベースインスタンスのゲートウェイとして機能できます。ECDA の設定では、ECDA がアクセスする各データベースは、ユニークなサービス名として設定されます。設定されているサービス名のうち、どのサービス名に接続するかを Replication Server が認識できるように、クライアントはログイン時に渡された server name を使用し、コネクションの確立に使用する、一致するサービス名を特定します。コネクションが interfaces ファイルエントリと一致している必要があります。Microsoft SQL Server の場合は、データベース名がそのサービスに対して有効なデータベースである必要があります。サービス名の役割とその設定の詳細については、『ECDA Access Service ユーザーズガイド』を参照してください。

## **Replication Agent と ECDA が共有するコネクション**

ECDA ゲートウェイと Replication Agent は、それぞれ別のスレッドで Replication Server に接続するので、1 つの Replication Server コネクションでこれらの両方のコンポーネントをサポートできます。

同じデータベースに対して双方向で情報を複写する場合、データベースゲートウェイと Replication Agent の両方に共通のコネクションがあれば、複写システムのネットワークポロジで使用されるリソースを削減することができます。

プライマリとレプリケートの両方を兼ねるデータベースへの Replication Server コネクションを作成するには、ECDA データベースゲートウェイを正しくサポートするコネクションを定義してから、Replication Agent を適切に設定する必要があります。

- Replication Server で、**create connection** コマンドを使用して、コネクションの **server\_name** と **database\_name** を定義します。**server\_name** 値は、ECDA で設定されたサービス名と一致している必要があります。
- Replication Agent で、**rs\_source\_ds** パラメータの値をその **server\_name** に、**rs\_source\_db** パラメータの値を目的の **database\_name** に設定します。

直接ロードマテリアライゼーションを使用するには、受信データ用と送信データ用の2つのデータベース接続を用意する必要があります。適切な接続プロファイル (**rs\_rs\_to\_<replicate>\_ra**。<replicate> は oracle、mssql、udb、または hanadb) を使用して、Replication Agent で使用される受信データ用のコネクションを作成する必要があります。送信データ用のコネクションを作成するには、通常の複写をサポートしている接続プロファイルを使用します。

### ECDA データベースゲートウェイ

ECDA データベースゲートウェイは、Replication Server からのトランザクションを SAP 複写システムの ASE 以外のレプリケートデータベースに適用します。

そのために Replication Server は、Replication Server コネクションについて指定された情報を使用して ECDA ゲートウェイにログインします。Replication Server は **user\_name** と **password** を使用してサーバにログインし、コネクションで定義されたデータベースに対して **use database** コマンドを発行します。

Replication Server の場合、ECDA ゲートウェイを Adaptive Server レプリケートデータベースと区別するものは何もありません。Replication Server は、通信する DSI スレッドから同じコマンドを配信し、同じ結果を想定します。

これには、以下のような意味があります。

- レプリケートデータベースにログインするために Replication Server が使用する有効なユーザ ID が、Replication Server コネクションで定義されている必要があります。
- このユーザ ID には、レプリケートテーブルを更新するパーミッションと、レプリケートプロシージャを実行するパーミッションが付与されている必要があります。
- Replication Server には、DB2 UDB、SAP HANA データベース、Microsoft SQL Server、Oracle データベースでレプリケートデータベースに必要なテーブルとファンクションを設定するための接続プロファイルが用意されています。レプリケートデータサーバおよびゲートウェイの条件の概要については、『Replication Server デザインガイド』>「Adaptive Server 以外のデータサーバへのデータの複写」を参照してください。
- データ型の表現が、レプリケートデータベースのネイティブデータ型と一致するように変換される必要があります。Replication Server には、DB2 UDB、Microsoft SQL Server、Oracle データサーバへの複写をサポートするために必要

なファンクション文字列、ファンクション文字列クラス、基本データ型定義、変換を設定するための接続プロファイルが用意されています。

- Replication Server の **resume connection** コマンドは、指定されたコネクションの DSI スレッドを使用してアクティビティを開始しようとします。ECDA の場合、これは DirectConnect サーバにログインし、レプリケートデータベース内の RS\_LASTCOMMIT テーブルにアクセスした後、レプリケートデータベースにトランザクションを適用するプロセスです。このプロセスに失敗すると、Replication Server ログにエラーとして記録されます。

## **ECDA Option for ODBC**

ECDA Option for ODBC は、DB2 UDB、Microsoft SQL Server、ODBC アクセス可能なデータベースに対する Open Client インタフェースを Replication Server に提供します。

---

**注意：** ECDA Option for ODBC の ODBC ドライバ (ターゲットに接続するバックエンドドライバ) は、SAP からは提供されません。各自で入手してインストールと設定を行う必要があります。

---

ECDA Option for ODBC は、IBM DB2 や Microsoft SQL Server などのターゲットデータベースに合わせて各自で取得した ODBC バックエンド (サーバ側) ドライバを使用して、ASE 以外のデータソースに対するアクセスを実現します。ODBC ドライバをベンダの指示に従って ECDA Option for ODBC と同じサーバにインストールしてから、この ODBC ドライバを使用してデータベースにアクセスするように ECDA Option for ODBC を設定します。

---

**注意：** 取得した ODBC ドライバに SAP のドライバマネージャソフトウェアとの互換性があるか、または取得した ODBC ドライバにドライバマネージャが含まれていることを確認してください。

---

ODBC ドライバの機能は広範囲に渡っているため、ASE の提供ではないサードパーティ製の ODBC ドライバで処理を行う場合には、必要な条件を満たしていることを確認するために統合とテストを注意深く実施する必要があります。

## **Mainframe Connect DirectConnect for z/OS Option**

Mainframe Connect DirectConnect for z/OS Option は、メインフレーム上で実行されている DB2 に対する Open Client インタフェースを Replication Server に提供します。

## ExpressConnect for Oracle

---

ExpressConnect for Oracle (ECO) は、Oracle を複写するための Replication Server 15.5 以降によってロードされるライブラリです。

ECO は、ECDA で次の利点があります。

- 起動、モニタリング、または管理のために、個別のサーバプロセスは必要ありません。
- Replication Server と ECO は同一のプロセス内で実行されるため、これらの中で SSL は不要であり、ECDA for Oracle のグローバル設定パラメータでこれまでに扱われていた設定内容を設定するための要件もありません。
- サーバ接続は Replication Server から **create connection** コマンドと **alter connection** コマンドを使用して設定されるため、ECDA for Oracle の **connect\_string** 設定で個別に同等の設定をする必要はありません。『Replication Server リファレンスマニュアル』を参照してください。
- `text_chunksize`、`autocommit`、`array_size` など、ECDA for Oracle と同等の特定の設定も設定する必要がありません。これらの設定は Replication Server によって (Replication Agent 入力に基づく場合もあります) 自動的に決められて ECO と通信します。

ECO には、次に示すように ECDA for Oracle と類似した機能が備えられています。

- データ型変換のセットが同じ。
- SAP データと Oracle データ間での言語および charset の変換。ECO では、`map.cfg` ファイルを使用して設定されます。
- Adaptive Server プライマリデータベース内の空の文字列が Oracle レプリケートデータベースに複写されると、Oracle では 1 つまたは複数の (カラムが `varchar` データ型か `fixed char width` データ型かによる) 空白で構成される文字列値になります。

ExpressConnect for Oracle でロケーションの透過性を確立するには、`tnsnames.ora` ファイルのみが必要です。ECDA for Oracle のように `interfaces` ファイルは必要ではありません。コネクションの設定のためには、`tnsnames.ora` ファイルで定義されるサービス名を指定する必要があります。

ECO の詳細については、『ExpressConnect for Oracle Configuration Guide』を参照してください。



## ExpressConnect for SAP HANA データベース

---

ExpressConnect for SAP HANA データベースは、SAP HANA データベースへの複写用に Replication Server によってロードされるライブラリです。

ExpressConnect for SAP HANA データベースでは次のことを行います。

- Replication Server の一部として動作します
- SAP HANA データベース ODBC ドライバを使用します。
- SAP® Secure Store と標準認証の両方をサポートします。
- LOB 複製時にテキストポインタを必要とせず、テキストポインタを操作するカスタムファンクション文字列を使用しません。



# プライマリデータサーバとしての IBM DB2 for z/OS

SAP 複写システムにおける、IBM z/OS プラットフォーム上の DB2 UDB サーバに固有のプライマリデータサーバの問題と考慮事項について説明します。

## Replication Agent for DB2 UDB

---

プライマリデータサーバとしての DB2 UDB は、Replication Agent for DB2 UDB と対話します。

Replication Agent for DB2 UDB を使用する際には、次の点を考慮してください。

- Replication Agent はデータ変更オペレーションまたはトランザクションに関する情報を識別し、DB2 UDB プライマリデータベースからプライマリ Replication Server に転送します。
- Replication Agent は、プライマリ Replication Server と対話します。また、プライマリ Replication Server の RSSD と対話するように設定されている場合は、RSSD とも対話します。

## 複写の干渉と影響

---

Replication Agent DB2 ライブラリは、許可プログラム機能 (APF) によって許可されている必要があります。

SAP 複写システムにおける、DB2 UDB プライマリデータサーバのパフォーマンスと動作は、以下の影響を受ける場合があります。

- DB2 UDB トランザクションログは、次のように影響を受けます。
  - 複写には、変更される各ローの更新前イメージと更新後イメージが必要です。プライマリテーブルを複写するようマーク付けした場合、テーブルは **DATA CAPTURE CHANGES** 句によって変更されます。複写するようマーク付けされたテーブルの数が増えると、DB2 UDB アクティブログデータセットに必要な DASD 領域も増加します。
  - Replication Agent for DB2 UDB を使用すると、DB2 UDB ログに格納されるデータの量が増加します。増加の規模は、プライマリテーブルの数、種類、サイズと、複写されるトランザクションの種類によって異なります。たとえば、**update** トランザクションには更新前イメージと更新後イメージの両方が必要であり、これには、変更されていないカラムも含め、ロー内のす

すべてのカラムが含まれます。詳細については、Replication Agent for DB2 UDB のマニュアルを参照してください。

- Replication Agent をインストールすると、次の 2 つの Replication Agent システムテーブルがプライマリ DB2 UDB に作成されます。
  - LTMOBJECTS は、複写するようマーク付けされた各プライマリテーブルのローを格納します。サイズは、複写するようマーク付けされたテーブルの数によって異なります。
  - 更新された LTMMARKER を使用すると、マテリアライゼーション処理に役立ちます。
- Replication Agent for DB2 UDB で開始されたタスクは、単一の DB2 サブシステムのログ、または DB2 データ共有グループ内のすべてのログを処理できます。この動作は、LTMCFG パラメータである **DataSharingOption**、**DataSharingMember**、**Log\_identifier**、および **BSDS** によって制御されます。
- プライマリデータベースの制限事項:
  - LOB 複写がサポートされていません。
  - char と varchar の最大サイズが 32767 です。
  - DDL とストアドプロシージャの複写がサポートされていません。
- 次の DB2 UDB ユーティリティは使用しないでください。使用すると、複写の整合性が失われる可能性があります。
  - **LOAD LOG NO**
  - **RECOVER**
  - **REORG with RECOVER**
- **rs\_ticket** を Replication Agent DB2 UDB で起動することはできません。Replication Server 15.5 で、**rs\_ticket** を Replication Agent DB2 コネクションに「挿入」することはできません。『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**sysadmin issue\_ticket**」を参照してください。

## DB2 UDB プライマリデータベースのパーミッション

---

LTM for z/OS の **LTM\_process\_maint\_uid\_trans** 設定パラメータが Y でないかぎり、メンテナンスユーザによってプライマリデータベースに適用される更新は無視されます。

次の 2 つのユーザ ID を作成します。

- LTMADMIN ユーザ - 以下を行う TSO ユーザ。必要に応じて LTMADMIN という名前が付けられます。
  - Replication Agent for DB2 UDB のインストール、起動、停止
  - DB2 UDB 上の Replication Agent システムテーブルの管理

LTMADMIN ユーザには、複写するようマーク付けされたすべての DB2 UDB テーブルに対する **ALTER TABLE** 権限が必要です。このユーザ ID は、複写する

ようマーク付けされたプライマリテーブルで **ALTER TABLE DATA CAPTURE CHANGES** コマンドを発行します。

また、LTMADMIN ユーザには、DB2 UDB ログファイルに対する **TRACE**、**DISPLAY**、**MONITOR2** の各パーミッションも必要です。

- Replication Server メンテナンスユーザ - Replication Server の **create connection** コマンドで、プライマリデータベースについて指定されたユーザ ID。

## プライマリデータサーバの接続

---

IBM z/OS 環境でプライマリ DB2 UDB データサーバに接続するには、Replication Agent for DB2 UDB に、IBM z/OS に対して定義され、正しい DB2 UDB プランおよびパッケージに対する execute パーミッションが付与されている有効なユーザ ID が必要です。

Replication Agent for DB2 UDB は、このユーザ ID を使用して DB2 UDB にログインします。

Replication Agent for DB2 UDB ジョブの JCL (Job Control Language) が、正しいアカウント、ユーザ ID、DB2 UDB ログ、DB2 UDB サブシステムライブラリを使用し実行するように変更されている必要があります。

## Replication Server の接続

---

Replication Agent for DB2 UDB では、Replication Server に接続するために interface ファイルを使用しません。Replication Server に接続するために必要な情報は、LTMCFG ファイルに含まれています。

Replication Server の interface ファイルには、Replication Agent for DB2 のエントリは必要ありません。

## Replication Server システムデータベースの接続

---

Replication Agent for DB2 UDB では、Replication Server System Database (RSSD) へのアクセスは不要です。

ただし、RSSD を使用することにより、Replication Agent for DB2 UDB と Replication Server 間のデータの量を減らすことができます。

LTMCFG パラメータの **Use\_repdef** が Y の場合、Replication Agent for DB2 UDB の起動時に複写定義がロードされます。その複写定義を変更する場合は、

Replication Agent が変更を認識できるように Replication Agent を停止し、再起動します。

RSSDに接続するために必要な情報は LTMCFG ファイルで提供されます。パラメータはすべて RSSD で始まり、すべてのパラメータを入力する必要があります。ただし、これらは、**Use\_repdef** が **N** に設定されている場合は検証されません。

## DB2 UDB プライマリデータベースの設定

---

Replication Agent は、単一の DB2 サブシステム、または DB2 データ共有グループ内のすべてのログに対して実行できます。LTMCFG パラメータでは、Replication Agent for DB2 UDB の DB2 環境 (**DataSharingOption**、**DataSharing Member**、**Log-identifier**、**BSDS**) を記述します。

Replication Agent for DB2 UDB はメインフレーム z/OS のアプリケーションであり、単一の z/OS アドレス空間で同時に実行される、次の 2 つのタスクで構成されています。

- *SAP Log Extract* - プライマリテーブルのデータ変更オペレーションに関する DB2 UDB のアクティブログとアーカイブログを連続してスキャンします。
- *Replication Agent for DB2 UDB for z/OS* - SAP Log Extract から複写トランザクションを受け取り、Log Transfer Language (LTL) に変換した後、プライマリ Replication Server に送信します。

**DataSharingOption** が Multi の場合、Replication Agent for DB2 UDB は **Boot Strap Dataset** (BSDS) パラメータを参照してデータ共有グループ内の各 DB2 メンバの **BSDS** を識別し、Replication Agent for DB2 UDB の位置とグループのメンバごとの DB2 ログを表示します。

Replication Agent のインストールと設定の問題はすべて、『*Replication Agent for DB2 UDB Installation Guide*』に記載されています。ただし、異機種間複写システムでは次のようになります。

- **rs\_source\_ds** パラメータと **rs\_source\_db** パラメータの値では、大文字と小文字が区別されます。Replication Agent と Replication Server の両方のパラメータで大文字と小文字が正しく指定されていない場合は、コネクションに失敗します。
- Replication Agent for DB2 UDB for z/OS の **LTM\_process\_maint\_uid\_trans** 設定パラメータは、メンテナンスユーザによって実行されたトランザクションを Replication Agent がプライマリ Replication Server に送信するかどうかを制御します。双方向複写環境 (同一の DB2 UDB 領域が複写元と複写先となる) では、**LTM\_process\_maint\_uid\_trans** パラメータの値を **N** に設定する必要があります。

true に設定していないと、別のサイトに複写されたトランザクションが元のサイトに戻って適用され、無限ループが生成される可能性があります。

## DB2 for z/OS におけるプライマリテーブルの複写定義

---

Replication Agent for DB2 UDB for z/OS の **Use\_repdef** 設定パラメータは、複写定義で指定されたカラムのみを格納するログ転送言語 (LTL) と、DB2 UDB プライマリテーブル内のすべてのカラムを格納するログ転送言語のどちらを Replication Agent が送信するかを制御します。

**Use\_repdef** パラメータの値が **N** に設定されている場合、Replication Agent は DB2 UDB プライマリテーブル内のすべてのカラムのデータを含む LTL を送信します。

**Use\_repdef** パラメータの値が **Y** に設定されている場合は、Replication Agent は、複写定義で指定されたカラムのデータのみを含む LTL を送信します。

複写定義に必要なカラムのデータのみを送信すると、ネットワークトラフィックが軽減されるので、パフォーマンスが向上することがあります。

**Use\_repdef** の値を **Y** に設定すると、他のパラメータ (**suppress\_col\_names**) などを使用して、Replication Agent のパフォーマンスを向上させることができます。

『Replication Agent for DB2 UDB Installation Guide』を参照してください。

**LTL\_table\_col\_case** パラメータは、Replication Agent が Replication Server にテーブル名とカラム名を送信するときの大文字と小文字の区別を制御します。DB2 でのデフォルトは大文字です。ただし、このパラメータを使用して、テーブル名とカラム名を大文字または小文字に変更したり、DB2 で定義されている名前を保持したりできます。

テーブルの名前が Replication Server またはターゲットデータベースの予約語と競合する可能性があります。 **with primary table named** 句や **with replicate table named** 句を使用すると、テーブル名を予約できます。ただし、LTMOBJECTS テーブルで REPLICATE\_NAME オプションを使用すると、Replication Agent for DB2 は、LTL を Replication Server に送信する前にテーブル名を変更します。『Replication Agent for DB2 UDB User and Troubleshooting Guide』の「Replication Agent Setup」の「DB2 Source Table Considerations」の「DB2 Table Names and Reserved Keywords」を参照してください。

## DB2 for z/OS プライマリデータ型の変換

---

Replication Agent for DB2 UDB for z/OS の **Date\_in\_char**、**Time\_in\_char**、**Timestamp\_in\_char** の各設定パラメータは、Replication Agent が値を文字列として送信するか、SAP の `datetime` フォーマットに変換するかを制御します。

これらのパラメータの詳細については、『Replication Agent for DB2 UDB Users and Troubleshooting Guide』を参照してください。

---

**注意：**日付または時間に関連するユーザ定義データ型 (UDD) を複写定義で使用する場合は、プライマリデータベースにとってネイティブなフォーマットで Replication Server にデータを送信するように、Replication Agent を設定することをおすすめします。また、Replication Agent でデータ型変換を実行しないことをおすすめします。

---

通常、Replication Agent for DB2 UDB はデータ型変換を行いません。ただし、すべてのレプリケートデータサーバで同じ変換が必要な場合は、処理時間を節約するために、各レプリケートデータベースの DSI で変換を行うのではなく、Replication Agent で一度に変換を行うようおすすめします。

IBM DB2 UDB では、午前 0 時は 24.00 として表されます。このフォーマットは、他のデータサーバと互換性がない場合があります。データ型定義を変更すると、この値を自動的に 24.00 から 00.00 に変更できます。

IBM DB2 UDB では、他のデータサーバに対応していない可能性がある年の値を使用できます。IBM DB2 UDB で許容される過去の年がレプリケートデータサーバで使用できない場合は、LTMCFG パラメータの *Minimum\_year* を設定し、DB2 UDB Replication Agent で、*Minimum\_year* パラメータよりも過去の年を *Date\_conv\_default* パラメータに変更できるようにします。

### 文字セット

DB2 内のデータは、複数の文字セットを使用してコード化できます。また、Replication Agent for DB2 を使用して、文字セットが Replication Server に送信される前に、複写された文字を Replication Server の文字セットに変換することもできます。

Replication Agent DB2 の文字セットプロパティを制御するパラメータは、**codepage** と **RS\_ccsid** です。これらのパラメータの詳細については、『Replication Agent for DB2 UDB Installation Guide』の「LTM for MVS Configuration Parameters」を参照してください。



## マテリアライゼーション

---

マテリアライゼーションとは、プライマリデータベースからのデータのコピーでレプリケートデータベースを初めて移植する処理のことです。

DB2 データを使用してターゲットをマテリアライズするには、Replication Agent for DB2 UDB を使用します。DB2 のアンロードユーティリティは、データファイルと、そのデータを記述したパンチカードファイルを生成します。これらのファイルを Replication Agent for DB2 UDB のマテリアライゼーション機能への入力として使用することにより、複写のターゲットを初期化できます。

『Replication Agent for DB2 UDB User and Troubleshooting Guide』の「Replication Server Setup」の「Task 3:Materializing Replicate Tables」の「Using Replication Agent Materialization」を参照してください。



# プライマリデータサーバとしての IBM DB2 for Linux, UNIX, および Windows

SAP 複写システムにおける、UNIX、Windows、Linux プラットフォーム上の DB2 UDB サーバに固有のプライマリデータベースの問題と考慮事項について説明します。

## Replication Agent for UDB

---

プライマリデータサーバとしての DB2 UDB は、Replication Agent と対話します。DB2 UDB に対応するように設定された Replication Agent のインスタンスは、「Replication Agent for UDB」と呼ばれます。

Replication Agent for UDB はデータ変更オペレーションまたはトランザクションに関する情報を識別し、DB2 UDB プライマリデータサーバからプライマリ Replication Server に転送します。

---

**注意：** トランザクションの複写元のデータベースごとに、個別の Replication Agent for UDB インスタンスが必要です。

---

Replication Agent は、プライマリ Replication Server と対話します。また、プライマリ Replication Server の RSSD と対話するように設定されている場合は、RSSD とも対話します。

---

**注意：** Replication Agent は Java プログラムです。オペレーティングシステムによっては、Java をサポートするためのパッチが必要な場合があります。『Replication Agent 管理ガイド』と『Replication Agent リリースノート』を参照してください。

---

## DB2 UDB システムの管理

---

Replication Agent には、プライマリデータベースに関するメタデータ情報(データベース名、テーブル名、プロシージャ名、カラム名など)を返す多くのコマンドがあります。

メタ情報を返すには、メタ情報を返すように設計された特別な JDBC 呼び出しを発行するか、システムテーブルに直接問い合わせます。

## DB2 UDB での複写の干渉と影響

---

SAP 複写システムにおける、DB2 UDB プライマリデータサーバのパフォーマンスと動作は、トランザクションログの影響を受けます。

- **LOGARCHMETH1** 設定パラメータを **LOGRETAIN** または **DISK:<path>** に設定する必要があります。ここで、<path> は、ログのアーカイブ先のディレクトリです。現在の **LOGARCHMETH1** 設定を指定するには、次の UDB コマンドを使用します。

```
get db cfg for <db-alias>
```

- 複写には、変更される各ローの更新前イメージと更新後イメージが必要です。プライマリテーブルを複写するようマークを付けた場合、Replication Agent for UDB はテーブルの **DATA CAPTURE** オプションを **DATA CAPTURE CHANGES** に設定します。複写するようマーク付けされたテーブルの数が増えると、DB2 UDB トランザクションログに必要な領域も増加します。
- プライマリデータベースには、ページサイズが 8KB 以上の、テンポラリーユーザーシステムが管理するテーブル領域が必要です。

## DB2 UDB プライマリデータベースのパーミッションと制限

---

Replication Agent for UDB には、プライマリデータベース内のデータにアクセスし、新しいオブジェクトを作成するためのパーミッションを持つ、DB2 UDB ログインが必要です。

プライマリデータベーストランザクションログにアクセスするには、DB2 UDB のログインに SYSADM または DBADM の権限が必要です。

Replication Agent では、DB2 UDB のストアードプロシージャはサポートされていません。『Replication Agent プライマリデータベースガイド』を参照してください。

## プライマリデータサーバの接続

---

プライマリ DB2 データサーバに接続するには、Replication Agent for UDB でいくつかのタスクを実行する必要があります。

Replication Agent for UDB が DB2 UDB サーバと異なるホストマシンにインストールされている場合、Replication Agent ホストマシンに DB2 UDB Administration Client をインストールしてください。

プライマリデータサーバとしての IBM DB2 for Linux, UNIX, および Windows

Replication Agent for UDB ソフトウェアが DB2 UDB サーバと同じホストマシンにインストールされている場合、個別の DB2 UDB Administration Client は必要ありません。

Windows システムでは、Replication Agent for UDB のコネクティビティを設定するときは、DB2 UDB Administration Client で ODBC データソースを設定してから、その ODBC データソースについて指定されているデータベース名とデータベースエイリアスを使用します。

UNIX システムでは、ODBC を使用する代わりに、UDB のノードとプライマリデータベースをカタログします。次に、プライマリデータベースのカタログ時に指定されたデータベースエイリアスを使用して、データソース Replication Agent の設定パラメータを設定します。

コネクティビティの設定方法の詳細については、『Replication Agent インストールガイド』>「Replication Agent のインストール」>「プライマリデータベースへのコネクティビティの設定」を参照してください。

設定する必要がある Replication Agent 設定パラメータについては、『Replication Agent インストールガイド』>「インストールの計画」を参照してください。

## Replication Server と RSSD のコネクティビティ

---

Replication Agent は、TCP/IP と SAP JDBC ドライバ (Replication Agent インストールに含まれている SAP jConnect) を使用して、他の SAP サーバと通信します。

Replication Agent は、コネクティビティ情報を得るのに interfaces ファイルを使用しません。

Replication Agent がプライマリ Replication Server に接続できるようにするために設定する必要がある Replication Agent 設定パラメータについては、『Replication Agent インストールガイド』>「インストールの計画」を参照してください。

## Replication Agent オブジェクト

---

`pdb_xlog init` を使用して Replication Agent を初期化すると、複写をサポートするオブジェクトがプライマリデータベースに作成されます。

詳細については、『Replication Agent プライマリデータベースガイド』を参照してください。

Replication Agent for UDB により、SYBRAUJAR.jar と SYBTRUNCJAR.jar が次のディレクトリにインストールされます。

- Windows では、ファイルは %DB2DIR%\SQLLIB\FUNCTION\jar ¥pds\_username にインストールされます。%DB2DIR% は UDB のインストールパス、pds\_username は Replication Agent 設定パラメータ pds\_username によって指定されたプライマリデータベースユーザ名です。
- UNIX では、ファイルは \$HOME/sqlllib/function/jar/pds\_username にインストールされます。\$HOME は UDB インスタンス所有者のホームディレクトリ、pds\_username は Replication Agent 設定パラメータ pds\_username によって指定されたプライマリデータベースユーザ名です。

これらの Jar ファイルにより、UDB プライマリデータベースにいくつかの Java プロシージャが実装されます。「トランケーション用の Java プロシージャ」の表は、Replication Agent の初期化時に作成され、ログのトランケーションに使用される Java プロシージャのリストです。

---

**注意：** 複数の Replication Agent インスタンスが 1 つの UDB サーバインストール (トランザクションが複製される各データベースのインストール) に設定されている場合、各 Replication Agent インスタンスは pds\_username 設定パラメータで異なるプライマリデータベースユーザ名を指定する必要があります。

---

## トランケーション用の Java プロシージャ

Replication Agent の初期化時に作成され、ログのトランケーションに使用される Java プロシージャをリストします。

| プロシージャ                                     | データベース名                  |
|--|--------------------------|
| 現在の LSN が格納されたログファイルの名前を取得する               | prefixget_log_name_      |
| get_log_name Java クラスのバージョンを取得する           | prefixget_version_str_   |
| データベースログファイルまたはアーカイブログディレクトリのファイルをトランケートする | prefixtrunc_log_files_   |
| trunc_log_files Java クラスのバージョンを取得する        | prefixget_trunc_ver_str_ |

## 複製オブジェクトの実際の名前の取得

Replication Agent インスタンスによって生成される Replication Agent データベースオブジェクトの名前を確認します。

Replication Agent 管理ポートで、キーワードを指定せずに **pdb\_xlog** コマンドを呼び出します。

```
pdb_xlog
```

**pdb\_xlog** コマンドは、Replication Agent によってプライマリデータベースに作成されたオブジェクトのリストを返します。

## DB2 UDB プライマリデータベースの設定

異機種間複写に固有の追加の問題について説明します。

プライマリ DB2 UDB データサーバのインストールの問題と設定パラメータの詳細はすべて、『Replication Agent for DB2 UDB Installation Guide』に記載されています。

### Java Runtime Environment

Replication Agent をインストールするときに、Replication Agent for UDB と互換性がある Java Runtime Environment (JRE) がインストールされます。

Java Runtime Environment に関する特記事項については、『Replication Agent リリースノート』を確認してください。

#### Java のヒープサイズ

プライマリ DB2 UDB データサーバに十分な大きさの JVM ヒープが確保されていることを確認します。

DB2 UDB `java_heap_sz` 設定パラメータを 2048 より大きい値に設定します。

### rs\_source\_ds 設定パラメータと rs\_source\_db 設定パラメータ

Replication Agent 設定ファイル内の設定パラメータ値はすべて、大文字と小文字を区別します。

Replication Server も大文字と小文字を区別するので、`rs_source_ds` および `rs_source_db` パラメータの値を指定するときには注意が必要です。Replication Agent と Replication Server の両方のパラメータで、大文字と小文字が正しく指定されていない場合は接続できません。

### filter\_maint\_userid 設定パラメータ

Replication Agent の `filter_maint_userid` 設定パラメータは、メンテナンスユーザが実行したトランザクションを、Replication Agent がプライマリ Replication Server に転送するかどうかを制御します。

メンテナンスユーザの名前は、Replication Server の `create connection` で、プライマリデータベースに対して定義されます。

双方向複写環境 (同一のデータベースが複写元と複写先になる) では、`filter_maint_userid` パラメータの値を `true` に設定します。true に設定していないと、別のサイトに複写されたトランザクションが元のサイトに戻って適用され、無限ループが生成される可能性があります。

## ltl\_character\_case 設定パラメータ

Replication Agent の **ltl\_character\_case** 設定パラメータは、Replication Agent がプライマリ Replication Server にデータベースオブジェクト名を送信するときの大文字と小文字の区別を制御します。

たとえば、複写定義が all tables named testtab に対して作成される場合、Replication Agent が送信するテーブル名は testtab でなければなりません。そうでない場合、不一致が起こります。Replication Server は大文字と小文字を区別するので、値 TESTTAB は値 testtab と一致しません。

複写定義を作成する場合は、デフォルト文字を選択し (たとえば、すべて大文字またはすべて小文字で複写定義を作成する)、Replication Agent の **ltl\_character\_case** パラメータの値を変更することで、大文字と小文字を一致させます。

## 複写定義での大文字と小文字および引用符付き識別子の設定

レプリケート SAP HANA 上でテーブルをマテリアライズするための複写定義を手動で作成しているときに、テーブルスキーマ内で引用符付き識別子を使用する場合は、いくつかのルールに従う必要があります。

プライマリデータベースにはデフォルトでオブジェクト名が大文字で格納されるので、次のルールに従ってください。

- 引用符なしのすべてのカラム名とテーブル名に大文字を使用します。
- プライマリカラムにすでに定義されているものと同じ名前を引用符付きカラムに使用します。
- Replication Agent の **ltl\_character\_case** 設定パラメータ値を **asis** に設定します。

## 大文字で格納されるオブジェクト名

DB2 UDB では、オブジェクトの作成時に大文字と小文字の区別が割り当てられていなければ、オブジェクト名はデフォルトですべて大文字で格納されます。そのため、特に設定されていないかぎり、Replication Agent はプライマリ Replication Server に大文字でオブジェクト名を送信します。

**ltl\_character\_case** パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。



## DB2 UDB におけるプライマリテーブルの複写定義

---

Replication Agent の **use\_rssd** 設定パラメータは、複写定義で指定されたカラムのみを格納するログ転送言語 (LTL) と、プライマリテーブル内のすべてのカラムを格納する LTL のどちらを Replication Agent が送信するかを制御します。

**use\_rssd** パラメータの値が **false** の場合、Replication Agent はプライマリテーブル内のすべてのカラムのデータを含む LTL を送信します。**use\_rssd** パラメータの値が **true** の場合は、Replication Agent は、各プライマリテーブルの複写定義で指定されたカラムのデータのみを含む LTL を送信します。

複写定義で指定されたカラムのデータのみを送信すると、ネットワークトラフィックが軽減されるので、パフォーマンスが向上することがあります。

また、Replication Agent は複写定義で指定した順番で情報を送信できるので、カラム名とパラメータ名は LTL から削除されます。LTL の **minimal columns** および **structured tokens** オプションは、**use\_rssd** パラメータの値が **true** の場合に使用できます。詳細については、『Replication Agent 管理ガイド』を参照してください。

## DB2 UDB プライマリデータ型の変換

---

Replication Agent では、DB2 UDB の DATE、TIME、および TIMESTAMP カラム値を Replication Server に送信する方法を制御できます。

DB2 UDB データ型マッピングの完全なリストについては、『Replication Agent プライマリデータベースガイド』 > 「Replication Agent for UDB」 > 「DB2 UDB 固有の考慮事項」 > 「DB2 UDB データ型の互換性」を参照してください。



# プライマリデータサーバとしての Microsoft SQL Server

SAP 複写システムにおける、Microsoft SQL Server データサーバに固有のプライマリデータベースの問題について説明します。

---

**注意：** Replication Agent for Microsoft SQL Server は Microsoft Windows にインストールする必要があります。

---

## Replication Agent for Microsoft SQL Server

---

プライマリデータサーバとしての Microsoft SQL Server は、Replication Agent と対話します。Replication Agent が Microsoft Windows にインストールされ、Microsoft SQL データベースログに直接アクセスできる必要があります。

Replication Agent はデータ変更オペレーションまたはトランザクションに関する情報を識別し、Microsoft SQL Server プライマリデータベースからプライマリ Replication Server に転送します。

---

**注意：** トランザクションの複写元のデータベースごとに、個別の Replication Agent インスタンスが必要です。

---

Replication Agent は、プライマリ Replication Server と対話します。また、プライマリ Replication Server の RSSD と対話するように設定されている場合は、RSSD とも対話します。

## sybfilter ドライバ

---

Replication Agent でデータを複写するには、sybfilter ドライバを使用して Microsoft SQL Server のログファイルを読み込み可能にする必要があります。

Replication Agent は Microsoft SQL Server のログファイルを読み込むことができます。ただし、Microsoft SQL Server プロセスは排他的な読み込みパーミッションを使用してこれらのログファイルを開くので、Replication Agent を含む他のプロセスからこのファイルを読み込むことはできません。『Replication Agent プライマリデータベースガイド』を参照してください。

## Microsoft SQL Server システムの管理

---

Replication Agent には、プライマリデータベースに関するメタデータ情報 (データベース名、テーブル名、プロシージャ名、カラム名など) を返す多くのコマンドがあります。

メタ情報を返すには、メタ情報を返すように設計された特別な JDBC 呼び出しを発行するか、システムテーブルに直接問い合わせます。

## Replication Agent のパーミッション

---

Microsoft SQL Server にログインするために Replication Agent インスタンスが使用するユーザ ID は、プライマリデータベースにアクセスする必要があります。

Replication Agent for Microsoft SQL Server では、複写作業用のデータベースオブジェクトをプライマリデータベースで作成します。自動的に付与される必要なパーミッションのリストについては、『Replication Agent プライマリデータベースガイド』を参照してください。

## プライマリデータサーバの接続

---

Replication Agent では、プライマリデータベースと通信するために JDBC ドライバが必要です。Microsoft SQL Server データベース用の JDBC ドライバはサードパーティデータベースベンダから提供されています。

使用するデータベース用の JDBC ドライバがまだインストールされていない場合は、ベンダの Web サイトから適切なドライバを入手してください。Microsoft SQL Server の JDBC の最新バージョンについては、『Replication Agent リリースノート』を参照してください。

## CLASSPATH 環境変数の設定

---

CLASSPATH 環境変数を設定する方法について説明します。

1. Replication Agent があるホストマシンまたは Replication Agent がアクセスできるホストマシンに JDBC ドライバをインストールします。
2. JDBC ドライバのロケーションを CLASSPATH 環境変数に追加します。

[スタート]>[設定]>[コントロールパネル]>[システム]>[環境] を選択し、パスセパレータとしてセミコロン (;) を使用して、次のロケーションを既存の

CLASSPATH 環境変数に追加します。または、[ユーザ環境変数] パネルでパスを作成します。

```
drive:¥path_name¥driver
```

各パラメータの意味は、次のとおりです。

- *drive* は、ドライブ文字です。
- *path\_name* は、JDBC ドライバをインストールした場所です。
- *driver* は、JDBC ドライバの名前です。Microsoft SQL Server の場合、この名前は `sqljdbc.jar` です。

3. [適用] をクリックし、次に [OK] をクリックします。

設定する必要がある Replication Agent 設定パラメータについては、『Replication Agent インストールガイド』>「インストールの計画」を参照してください。

## Replication Server と RSSD のコネクティビティ

---

Replication Agent は、TCP/IP と SAP JDBC ドライバ (Replication Agent インストールに含まれている SAP jConnect) を使用して、他の SAP サーバと通信します。

Replication Agent は、コネクティビティ情報を得るのに `interfaces` ファイルを使用しません。

Replication Agent がプライマリ Replication Server に接続できるようにするために設定する必要がある Replication Agent 設定パラメータについては、『Replication Agent インストールガイド』>「インストールの計画」を参照してください。

## Replication Agent オブジェクト

---

Replication Agent は、複写作業用のオブジェクトをプライマリデータベースで作成します。

Replication Agent オブジェクトは、`pdb_xlog` コマンドを `init` キーワードを指定して呼び出すと自動的に作成されます。既存のプライマリデータベースオブジェクトには、複写のマークを付けることができます。

一般的な情報については、『Replication Agent 管理ガイド』を参照してください。

Replication Agent のデータベースオブジェクト名には、次の 2 つの変数がありません。

- *prefix* - `pdb_xlog_prefix` パラメータの 1 ~ 3 文字の文字列値を表します (デフォルトは `ra_`)。

- *xxx*-英数字のカウンタを表します。この文字列は、データベース内でユニークな名前にするためにデータベースオブジェクト名に追加されます。

**pdb\_xlog\_prefix** パラメータの値は、すべての Replication Agent オブジェクト名で使用されるプレフィクス文字列です。**pdb\_xlog\_prefix\_chars** パラメータの値は、**pdb\_xlog\_prefix** で指定するプレフィクス文字列で使用できる非英数字のリストです。使用できる文字のリストはデータベース固有です。たとえば、Microsoft SQL Server では、データベースオブジェクト名に使用できる非英数字は \$、#、@、\_ のみです。

プライマリデータベース内の Replication Agent トランザクションログコンポーネントの名前を表示するには、**pdb\_xlog** コマンドを使用します。

ログオブジェクト名を設定する方法の詳細については、『Replication Agent 管理ガイド』を参照してください。

### テーブル、プロシージャ、マーカ、およびトリガオブジェクト

Replication Agent のトリガオブジェクトと見なされるコマンドに加え、Replication Agent のオブジェクトと見なされるテーブルおよびプロシージャオブジェクト、マーカプロシージャおよびマーカ隠しテーブルは、『Replication Agent プライマリデータベースガイド』にリストされています。

insert パーミッションと delete パーミッションは、データベース名 **prefixddl\_trig\_xxx** の DDL 隠しテーブルでのみ PUBLIC に付与されます。他のテーブルに対するパーミッションは与えられません。

**sp\_SybSetLogforReplTable** プロシージャと **sp\_SybSetLogforReplProc** プロシージャは、Microsoft SQL Server の **mssqlsystemresource** システムデータベースに作成されます。これらのプロシージャの **execute** パーミッションは PUBLIC に付与されますが、Replication Agent の **pds\_username** ユーザだけがプロシージャを正常に実行できます。これは、**pds\_username** ユーザだけが **sys.syssschobjs** テーブルに対する **select** パーミッションを付与されているためです。他のプロシージャの作成時に付与されるパーミッションはありません。

## Microsoft SQL Server プライマリデータベースの設定

異機種間複写に固有の追加の問題について説明します。

Microsoft SQL Server プライマリデータサーバのインストールの問題と設定パラメータの詳細については、すべて『Replication Agent インストールガイド』に記載されています。

## rs\_source\_ds 設定パラメータと rs\_source\_db 設定パラメータ

Replication Agent 設定ファイル内の設定パラメータ値はすべて、大文字と小文字を区別します。

Replication Server も大文字と小文字を区別するので、**rs\_source\_ds** および **rs\_source\_db** パラメータの値を指定するときには注意が必要です。Replication Agent と Replication Server の両方のパラメータで、大文字と小文字が正しく指定されていない場合は接続できません。

## filter\_maint\_userid 設定パラメータ

**maint\_user** として sysadmin 権限を持つ Microsoft SQL Server ログインを使用する場合、そのログインを対応するデータベースのユーザにマップします。そうしないと、この **maint\_user** によって実行されたトランザクションを Replication Agent が正しくフィルタできません。

## ltl\_character\_case 設定パラメータ

Replication Agent の **ltl\_character\_case** 設定パラメータは、Replication Agent がプライマリ Replication Server にデータベースオブジェクト名を送信するときの大文字と小文字の区別を制御します。

たとえば、複写定義が all tables named testtab に対して作成される場合、Replication Agent が送信するテーブル名は testtab でなければなりません。そうでない場合、不一致が起こります。Replication Server は大文字と小文字を区別するので、値 TESTTAB は値 testtab と一致しません。

複写定義を作成する場合は、デフォルト文字を選択し (たとえば、すべて大文字またはすべて小文字で複写定義を作成する)、Replication Agent の **ltl\_character\_case** パラメータの値を変更することで、大文字と小文字を一致させます。

以下は、データベースの作成時に指定した照合によって異なります。Microsoft SQL Server データベースでは、オブジェクトの作成時に大文字と小文字の区別が割り当てられていなければ、オブジェクト名はデフォルトで小文字で格納されます。特に設定されていないかぎり、Replication Agent はプライマリ Replication Server に小文字でオブジェクト名を送信します。

**ltl\_character\_case** パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## Microsoft SQL Server におけるプライマリテーブルの複写定義

---

複写定義で指定されたカラムのデータのみを送信すると、ネットワークトラフィックが軽減されるので、パフォーマンスが向上することがあります。

Replication Agent の `use_rssd` 設定パラメータは、複写定義で指定されたカラムのみを格納するログ転送言語 (LTL) と、プライマリテーブル内のすべてのカラムを格納する LTL のどちらを Replication Agent が送信するかを次のように制御します。

- `use_rssd` パラメータの値が `false` の場合、Replication Agent はプライマリテーブル内のすべてのカラムのデータを含む LTL を送信します。
- `use_rssd` パラメータの値が `true` の場合は、Replication Agent は、各プライマリテーブルの複写定義で指定されたカラムのデータのみを含む LTL を送信します。

また、Replication Agent は複写定義で指定した順番で情報を送信できるので、カラム名とパラメータ名は LTL から削除されます。LTL の `minimal columns` および `structured tokens` オプションは、`use_rssd` パラメータの値が `true` の場合に使用できません。『Replication Agent 管理ガイド』を参照してください。

複写定義を変更するには、『Replication Server 管理ガイド 第1巻』の「複写テーブルの管理」の「複写定義の修正」の「複写定義の変更」の「複写定義の変更要求プロセス」を参照してください。

## Microsoft SQL Server プライマリデータ型の変換

---

Microsoft SQL Server のデータ型は、対応するいくつかの Adaptive Server のデータ型と互換性があります。

`varchar(max)`、`nvarchar(max)`、`varbinary(max)` の各 LOB データ型を Microsoft SQL Server 以外のデータベースに複写することはできません。ただし、Replication Agent パラメータの `lr_partial_updates_allowed` が `false` に設定されている場合は、LOB データ型を SAP HANA データベースに複写することができます。『Replication Agent プライマリデータベースガイド』 > 「Replication Agent for Microsoft SQL Server」 > 「Microsoft SQL Server 固有の考慮事項」 > 「SAP HANA データベースへの Microsoft SQL Server ラージオブジェクトデータ型の複写」と『Replication Agent リファレンスマニュアル』 > 「設定パラメータ」 > 「設定パラメータリファレンス」 > 「`lr_partial_updates_allowed`」を参照してください。

Microsoft SQL Server と Replication Server のデフォルトのデータ型マッピングの完全なリストについては、『Replication Agent プライマリデータベースガイド』 >



「Replication Agent for Microsoft SQL Server」 > 「Microsoft SQL Server-固有の考慮事項」 > 「Microsoft SQL Server とのデータ型互換性」を参照してください。

## Microsoft SQL Server の大きな値のデータ型の複写

Replication Server は、Microsoft SQL Server の大きな値のデータ型 varchar (max)、nvarchar (max)、および varbinary (max) の複写をサポートしています。これらのデータ型は、それぞれ 2,147,483,647 バイトまでのデータを格納できます。

Replication Server は、テーブルレベルの複写環境でユーザ定義データ型 (UDD) として Microsoft SQL Server の大きな値のデータ型をサポートしています。また、新しい Microsoft SQL Server の大きな値のデータ型のデータベースレベルの複写もサポートしています。これらの Microsoft SQL Server の大きな値のデータ型は、text、unitext、および image という Replication Server のデータ型に直接マップされます。

表 1 : Microsoft SQL Server の大きな値のデータ型から Replication Server の基本データ型へのマッピング

| 新しい Microsoft SQL Server の大きな値のデータ型 | Replication Server の基本データ型 |
|-------------------------------------|----------------------------|
| varchar (max)                       | text                       |
| nvarchar (max)                      | unitext                    |
| varbinary (max)                     | image                      |

## Microsoft SQL Server の大きな値のデータ型の制限

Microsoft SQL Server の大きな値のデータ型の複写にはいくつかの制限があります。

- 複写定義で大きな値のデータ型カラムをプライマリキーとして定義することはできません。
- 複写定義で大きな値のデータ型カラムを検索可能として定義することはできません。
- 新しい大きな値のデータ型のいずれかをパラメータとして含むストアドプロシージャを複写することはできません。
- テキストポインタを使用して、新しい大きな値のデータ型のデータを操作することはできません。

混合バージョン環境では、プライマリ Replication Server とレプリケート Replication Server のサイトバージョンが 15.2 以降、LTL バージョンが 710 以降であることが必要です。

ASE 以外のデータサーバに関するデータ型の問題については、『Replication Server 異機種間複写ガイド』を参照してください。

### **Microsoft SQL Server の大きな値のデータ型の部分更新**

部分更新トランザクションでは、完全更新のように **delete** コマンドと **replace** コマンドを発行しないで、テーブルカラムのユーザ定義位置に文字列を直接書き込みます。

部分更新では、マルチバイト文字セットの変換がサポートされません。部分更新は Microsoft SQL Server でのみサポートされます。

# プライマリデータサーバとしての Oracle

SAP 複写システムにおける、Oracle データサーバに固有のプライマリデータベースの問題と考慮事項について説明します。

## Replication Agent for Oracle

---

プライマリデータサーバとしての Oracle は、Replication Agent と対話します。Replication Agent はデータ変更オペレーション(またはトランザクション)に関する情報を識別し、Oracle プライマリデータサーバからプライマリ Replication Server に転送します。

---

**注意：** トランザクションの複写元の Oracle データベースごとに、個別の Replication Agent インスタンスが必要です。

---

Replication Agent は、プライマリ Replication Server と対話します。また、プライマリ Replication Server の RSSD と対話するように設定されている場合は、RSSD とも対話します。

---

**注意：** Replication Agent は Java プログラムです。オペレーティングシステムによっては、Java をサポートするためのパッチが必要な場合があります。詳細については、『Replication Agent 管理ガイド』および『Replication Agent リリースノート』を参照してください。

---

## Oracle におけるプライマリテーブルの複写定義

複写定義で指定されたカラムのデータのみを送信すると、ネットワークトラフィックが軽減されるので、パフォーマンスが向上することがあります。

Replication Agent の `use_rssd` 設定パラメータは、複写定義で指定されたカラムのみを格納するログ転送言語 (LTL) と、プライマリテーブル内のすべてのカラムを格納する LTL のどちらを Replication Agent が送信するかを制御します。

`use_rssd` パラメータの値が `false` の場合、Replication Agent はプライマリテーブル内のすべてのカラムのデータを含む LTL を送信します。`use_rssd` パラメータの値が `true` の場合は、Replication Agent は、各プライマリテーブルの複写定義で指定されたカラムのデータのみを含む LTL を送信します。

また、Replication Agent は複写定義で指定した順番で情報を送信できるので、カラム名とパラメータ名は LTL から削除されます。LTL の `minimal columns` および

**structured tokens** オプションは、**use\_rssd** パラメータの値が **true** の場合に使用できません。『Replication Agent 管理ガイド』を参照してください。

複写定義を変更するには、『Replication Server 管理ガイド 第1巻』の「複写テーブルの管理」の「複写定義の修正」の「複写定義の変更」の「複写定義の変更要求プロセス」を参照してください。

## Oracle システムの管理

---

Replication Agent には、プライマリデータベースに関するメタデータ情報(データベース名、テーブル名、プロシージャ名、カラム名など)を返す多くのコマンドがあります。

メタ情報を返すには、メタ情報を返すように設計された特別な JDBC 呼び出しを発行するか、Oracle システムテーブルに直接問い合わせます。

**注意：** Adaptive Server Enterprise とは異なり、Oracle は 1 つのサーバインスタンス内では複数のデータベースをサポートしません。

---

## Oracle での複写の干渉と影響

---

SAP 複写システムが組み込まれている場合、Oracle プライマリデータサーバのパフォーマンスと動作が影響を受ける場合があります。

Replication Agent は Oracle オンラインログとアーカイブ redo ログにアクセスしてトランザクション情報を取得する一方で、特定のログ設定を必要とします。必要な情報を提供および管理するには、Oracle で以下の項目を有効にします。

- redo ログのアーカイブ
- プライマリキーとユニークインデックスデータの補足ロギング

## Oracle から SAP HANA データベースへの連続複写の制限

---

カラムが NULL に設定されている場合は連続複写が失敗します。

デフォルトでは、Oracle プライマリデータベースは最低限のロギングを行うため、**UPDATE** 文ですべてのカラムが指定されるわけではありません。**UPDATE** 文で指定されていないカラムは、**where** 句を含むサブスクリプションで実行されるローマイグレーションの結果として自動的に NULL に設定されます。**where** 句で指定されたカラムの更新時に直接ロードマテリアライゼーションに成功しても、カラムが NULL に設定されるので連続複写には失敗します。

## Oracle プライマリデータベースのパーミッション

---

Replication Agent には、プライマリデータベース内のデータにアクセスし、新しいオブジェクトを作成するパーミッションを持つ Oracle ログイン ID が必要です。

このような必要なパーミッションを持つ Oracle ログイン ID のリストについては、『Replication Agent プライマリデータベースガイド』を参照してください。

## プライマリデータサーバの接続

---

Replication Agent では、Oracle プライマリデータベースに接続するために JDBC ドライバが必要です。

JDBC ドライバは、Replication Agent ホストマシンにインストールされ、CLASSPATH システム変数で参照される必要があります。Java は、CLASSPATH システム変数の内容を参照して Java クラスの検索場所を特定します。Oracle JDBC ドライバの場合、次のように、CLASSPATH 変数に完全なパスとファイル名が含まれていなければなりません。

```
drive:¥<path_name>¥ojdbc14.jar
```

サポートされる JDBC ドライバのバージョンについては、『Replication Agent リリースノート』を参照してください。

JDBC のコネクティビティを確保するには、Oracle プライマリデータサーバの TNS Listener プロセスが実行されている必要があります。

設定する必要がある Replication Agent 設定パラメータについては、『Replication Agent インストールガイド』>「インストールの計画」を参照してください。

## Replication Server と RSSD のコネクティビティ

---

Replication Agent は、TCP/IP と SAP JDBC ドライバ (Replication Agent インストールに含まれている SAP jConnect) を使用して、他の SAP サーバと通信します。

Replication Agent は、コネクティビティ情報を得るのに interfaces ファイルを使用しません。

Replication Agent がプライマリ Replication Server に接続できるようにするために設定する必要がある Replication Agent 設定パラメータについては、『Replication Agent インストールガイド』>「インストールの計画」を参照してください。

## Replication Agent オブジェクト

---

Replication Agent は、複写作業用のオブジェクトをプライマリデータベースで作成します。

Replication Agent のデータベースオブジェクト名には、次の 2 つの変数があります。

- *prefix* - **pdb\_xlog\_prefix** パラメータの 1 ～ 3 文字の文字列値を表します (デフォルトは **ra\_**)。
- *xxx* - 英数字のカウンタを表します。この文字列は、データベース内でユニークな名前にするためにデータベースオブジェクト名に追加されます。

**pdb\_xlog\_prefix** パラメータの値は、**rs\_marker** と **rs\_dump** を除き、すべての Replication Agent オブジェクト名で使用されるプレフィクス文字列です。

**pdb\_xlog\_prefix\_chars** パラメータの値は、**pdb\_xlog\_prefix** で指定するプレフィクス文字列で使用できる非英数字のリストです。使用できる文字のリストはデータベース固有です。たとえば、データベースオブジェクト名に使用できる非英数字は \$、#、\_ のみです。

プライマリデータベース内の Replication Agent トランザクションログコンポーネントの名前を表示するには、**pdb\_xlog** コマンドを使用します。

オブジェクト名を設定する方法の詳細については、『Replication Agent 管理ガイド』を参照してください。

作成されたオブジェクトの名前を調べるには、Replication Agent 管理ポートで、キーワードを指定せずに **pdb\_xlog** コマンドを呼び出します。

```
pdb_xlog
```

**pdb\_xlog** コマンドは、すべての Replication Agent オブジェクトのリストを返します。

## Oracle プライマリデータベースの設定

---

異機種間複写に固有の追加の問題について説明します。

Oracle プライマリデータサーバのインストールの問題と設定パラメータの詳細はすべて、『Replication Agent インストールガイド』に記載されています。

## Java Runtime Environment

Replication Agent をインストールするときに、Replication Agent と互換性がある Java Runtime Environment (JRE) がインストールされます。

Java Runtime Environment に関する特記事項については、『Replication Agent リリースノート』を参照してください。

## 必要な JDBC ドライバ

プライマリデータサーバへのコネクティビティを確保するために、Replication Agent には JDBC ドライバが必要です。

SAP では、Oracle データサーバ用の JDBC ドライバは提供していません。Oracle データサーバ用 JDBC ドライバの入手方法については、『Replication Agent リリースノート』を参照してください。

## rs\_source\_ds 設定パラメータと rs\_source\_db 設定パラメータ

Replication Agent 設定ファイル内の設定パラメータ値はすべて、大文字と小文字を区別します。

Replication Server も大文字と小文字を区別するので、**rs\_source\_ds** および **rs\_source\_db** パラメータの値を指定するときには注意が必要です。Replication Agent と Replication Server の両方のパラメータで、大文字と小文字が正しく指定されていない場合は接続できません。

## filter\_maint\_userid 設定パラメータ

Replication Agent の **filter\_maint\_userid** 設定パラメータは、メンテナンスユーザが実行したトランザクションを、Replication Agent がプライマリ Replication Server に転送するかどうかを制御します。

メンテナンスユーザの名前は、Replication Server の **create connection** で、プライマリデータベースに対して定義されます。

双方向複写環境 (同一のデータベースが複写元と複写先になる) では、**filter\_maint\_userid** パラメータの値を **true** に設定します。true に設定していないと、別のサイトに複写されたトランザクションが元のサイトに戻って適用され、無限ループが生成される可能性があります。

## ltl\_character\_case 設定パラメータ

Replication Agent の **ltl\_character\_case** 設定パラメータは、Replication Agent がプライマリ Replication Server にデータベースオブジェクト名を送信するときの大文字と小文字の区別を制御します。

たとえば、複写定義が all tables named testtab に対して作成される場合、Replication Agent が送信するテーブル名は testtab でなければなりません。そうでない場合、不一致が起こります。Replication Server は大文字と小文字を区別するので、値 TESTTAB は値 testtab と一致しません。

複写定義を作成する場合は、デフォルト文字を選択し (たとえば、すべて大文字またはすべて小文字で複写定義を作成する)、Replication Agent の **ltl\_character\_case** パラメータの値を変更することで、大文字と小文字を一致させます。

Oracle データベースでは、オブジェクトの作成時に大文字と小文字の区別が強制されていないければ、オブジェクト名はデフォルトですべて大文字で格納されます。特に設定されていないかぎり、Replication Agent はプライマリ Replication Server に大文字でオブジェクト名を送信します。

**ltl\_character\_case** パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## 複写定義での大文字と小文字および引用符付き識別子の設定

レプリケート SAP HANA 上でテーブルをマテリアライズするための複写定義を手動で作成しているときに、テーブルスキーマ内で引用符付き識別子を使用する場合は、いくつかのルールに従う必要があります。

プライマリデータベースにはデフォルトでオブジェクト名が大文字で格納されるので、次のルールに従ってください。

- 引用符なしのすべてのカラム名とテーブル名に大文字を使用します。
- プライマリカラムにすでに定義されているものと同じ名前を引用符付きカラムに使用します。
- Replication Agent の **ltl\_character\_case** 設定パラメータ値を **asis** に設定します。

## Oracle プライマリデータ型の変換

データ型の変換とデータ型マッピングを実行すると、Oracle データ型のデータ型マッピングの完全なリストが表示されます。

参照：

- データ型の変換とマッピング (285 ページ)



## Automatic Storage Management

---

Replication Agent for Oracle では、オンライン用の Oracle Automatic Storage Management (ASM) 機能の使用、および redo ログのアーカイブがサポートされています。

ASM は、Oracle データベース環境に対してファイルシステム管理とボリューム管理をサポートします。ASM は、Real Application Cluster (RAC) 環境と RAC 以外の環境の両方で使用できます。ASM には、RAID (Redundant Array of Independent Disks) または論理ボリュームマネージャ (LVM) と同様の利点があります。

これらのテクノロジーと同様に、ASM では一連の個別のディスクから 1 つのディスクグループを定義できます。ASM は、ディスクグループで定義されたすべてのデバイス間で負荷分散を行います。また、ストライピング機能とミラーリング機能も提供します。RAID または LVM と異なり、ASM だけが Oracle データベースで作成され、読み込まれたファイルをサポートします。ASM は汎用的なファイルシステムでは使用できず、バイナリファイルやフラットファイルを保管できません。オペレーティングシステムは ASM ファイルにアクセスできません。

Oracle ASM 用の Replication Agent サポートの詳細については、『Replication Agent プライマリデータベースガイド』を参照してください。

## Real Application Cluster

---

Replication Agent は、Oracle 10g、11g、および 12c Real Application Cluster (RAC) 環境をサポートします。

Replication Agent for Oracle インスタンスを初期化すると、Oracle データベースに対して、クラスタによってサポートされるノード数を決定するための問い合わせが実行されます。この情報を基に、Replication Agent はすべてのノードから redo ログ情報を処理するように自動的に設定されます。

---

**注意：** RAC データベースの複製は、RAC 以外のデータベースの複製と同じです。

---

Oracle RAC クラスタ内にあるすべてのノードの redo ログデータを処理するためには、Oracle ノードが各自の redo データの格納に使用する共有領域にアクセスできるロケーションから、Replication Agent を実行する必要があります。Replication Agent は、オンラインの redo ログとアーカイブされた redo ログの両方が存在する共有領域に対し、読み込みアクセスする必要があります。

必要なホスト、ポート、Oracle SID の値を **pds\_host\_name**、**pds\_port\_number**、**pds\_database\_name** の各設定パラメータに指定することで、1 つの Oracle インスタ

ンスに接続するように Replication Agent を設定できます。Oracle RAC 環境では、ノードで障害が発生したり、使用できない場合に備えて、Replication Agent はクラスタ内のすべてのノードに接続できる必要があります。

複数のノードロケーションの設定をサポートするために、Replication Agent は、指定した1つのエントリについて Oracle tnsnames.ora ファイルから必要な情報を取得することによって、すべての RAC ノードへのコネクティビティをサポートします。これにより、すべてのノードに対して個々のホスト名、ポート名、インスタンス名を設定するのではなく、tnsnames.ora ファイルの場所と使用する TNS 接続の名前のみが必要になります。

Oracle RAC 用の Replication Agent サポートの詳細については、『Replication Agent プライマリデータベースガイド』を参照してください。

# プライマリデータサーバとしての SAP Business Suite データベース

SAP 複写システムにおける、SAP Business Suite に固有のプライマリデータベースの問題と考慮事項について説明します。

## SAP Business Suite データベースの複写

---

Replication Server は、サポートされているデータベース上で稼働中の SAP Business Suite アプリケーションから SAP HANA データベースへの、データベース全体またはテーブルのリストのマテリアライゼーションおよび複写をサポートしています。

サポートされている次のデータベース上で稼働している SAP Business Suite アプリケーションを SAP HANA データベースに複写できます。

- Oracle
- IBM DB2 UDB
- Microsoft SQL Server

---

**注意：** SAP Replication Server は、異機種プライマリデータベースからのデータベースレベルのマテリアライゼーションをサポートしていません。

---

### Replication Agent for Microsoft SQL Server によるテーブルのトランケーション

**sp\_SybTruncateTable** ストアドプロシージャを使用すると、テーブルのトランケーションに関する Microsoft SQL Server の制限を無効にすることによって、SAP ERP または任意のアプリケーションからの複写対象としてマーク付けされたテーブルをトランケートできます。

**sp\_SybTruncateTable** を発行すると、マーク付けされたテーブルの複写フラグが一時的にオフになり、**truncate table** コマンドが発行された後、複写フラグがオンになります。

Replication Agent for Microsoft SQL Server を複写に使用する SAP ERP システムには、**sp\_SybTruncateTable** ストアドプロシージャを含む SAP カーネルの最新のパッチが必要です。SAP ERP アプリケーションでマーク付けされたテーブルを複写するには、**truncate table** オペレーションが必要です。

SAP カーネルのダウンロードとインストールの手順については、「SAP ノート 1972365 - Retry on error 4711」(<https://css.wdf.sap.corp/sap/support/notes/1972365>)を参照してください。

SAP HANA データベースへの複写を初めて設定する場合、複写コンポーネントのインストールについては『SAP Replication Server: SAP HANA データベースクイックスタートガイド』を、複写環境の設定、複写対象のテーブルの特定、およびマテリアライゼーションについては『Replication Management Agent 設定およびユーザーズガイド』を参照してください。

コマンドを使用して複写環境を手動で設定するには、『SAP Replication Server: SAP HANA データベースクイックスタートガイド』を参照してください。

## クラスタテーブルの複写

---

Replication Server は、クラスタテーブルを使用する SAP Business Suite アプリケーションから SAP HANA データベースへのデータベース複写をサポートしています。

SAP Business Suite では、アプリケーションレイヤで論理テーブルが定義されます。論理テーブルはクラスタ化されて、データベースの物理テーブルに格納されます。クラスタテーブルとは、複数のテーブルをマテリアライズしたナチュラルジョインの圧縮表現です。クラスタテーブルが格納されたデータベースのマテリアライゼーションと複写を実行するとき、Replication Server はデータのクラスタ化を解除してから DML を SAP HANA データベースに送信します。

---

**注意：** Replication Server は、プールテーブルの複写をサポートしていません。

---

クラスタテーブルを複写する場合は、次のことを確認してください。

- SAP HANA が 1.00.62 以降である。
- SAP カーネルが 720、721、740、または 741 である。
- ライブラリ `libSQLDBCHDB.dll` (Windows) または `libSQLDBCHDB.so` (UNIX) が Revision 1.00.72.00 以降である。
- ライブラリ `libodbcHDB.dll` (Windows) または `libodbcHDB.so` (UNIX) が Revision 1.00.72.00 以降である。

---

**注意：** クラスタテーブル `CLU4` と `VER_CLUSTR` は、複写対象としてマーク付けしないでください。詳細については、「SAP ノート 902817 - クラスタテーブル (論理項目名/物理項目名) における不整合」を参照してください。 <https://css.wdf.sap.corp/sap/support/notes/902817>

---

## SAP Business Suite データベースのマテリアライゼーションと複製のシナリオ

---

サポートされているデータベース上で稼働している SAP Business Suite アプリケーションのマテリアライゼーションと SAP HANA データベースへの複製を実行するには、Replication Server を設定する必要があります。

シナリオ 1 では、DB2 UDB のプライマリデータベースにクラスタテーブルが格納されている場合の Replication Server の設定方法について説明します。シナリオ 2 では、Oracle のプライマリデータベースにクラスタテーブルが格納されている場合の Replication Server の設定方法について説明します。プライマリデータベースが Microsoft SQL Server の場合、Replication Server の設定手順はシナリオ 2 と同じですが、手順 7 を省略する必要があります。

紹介するシナリオの前提条件と考慮事項は次のとおりです。

- 複製コンポーネントがインストールされている。
- SAP HANA のバージョンが 1.00.62 以降である。
- SAP カーネルが 720、721、740、または 741 である。
- ライブラリ `libSQLDBCHDB.dll` (Windows) または `libSQLDBCHDB.so` (UNIX) Revision 1.00.72.00 以降がダウンロードされてインストールされている。
- ライブラリ `libodbcHDB.dll` (Windows) または `libodbcHDB.so` (UNIX) Revision 1.00.72.00 以降がダウンロードされてインストールされている。
- Replication Server からプライマリデータベースへの接続がすでに存在している。
- ExpressConnect for SAP HANA データベースを使用したレプリケート SAP HANA データベースインスタンスへの接続がすでに存在している。

---

**注意：** SAP アプリケーションによって作成されたスキーマ内に手動でテーブルを作成しないでください。

---

『SAP Replication Server: SAP HANA データベースクイックスタートガイド』を参照してください。

### シナリオ 1

IBM DB2 UDB データベース上で稼働し、クラスタテーブルが格納された SAP Business Suite アプリケーションから SAP HANA データベースへの複製を設定できます。

以下の手順では、マテリアライゼーションと複製を有効にするように Replication Server を設定する方法について説明します。

1. varbinary 値での後続ゼロの複写を有効にします。

```
configure replication server
set varbinary_strip_trailing_zeros to 'off'
```

2. SAP HANA データベース <sid>adm ユーザの \$\$SAPSYSTEMNAME 環境変数の値を sapsystemname 設定パラメータに設定します。

```
alter connection to rds.rdb
set sapsystemname to 'sid_adm_value'
```

構文の説明は次のとおりです。

- *rds.rdb* は SAP HANA データベースのコネクション名です。
- *sid\_adm\_value* は SID (SAP SYSTEM NAME) の名前です。  
たとえば、SYSTEM NAME は *DD01* です。次のように入力する必要があります。  

```
alter connection to rds.rdb set sapsystemname to 'DD01'
```

3. 次のようにプライマリスキーマをレプリケートスキーマにマップします。

```
alter connection to rds.rdb
set schemamap with decluster from pds.pdb.primary_schema to
HANA_schema
```

構文の説明は次のとおりです。

- *rds.rdb* は SAP HANA データベースのコネクション名です。
  - *pds.pdb* はプライマリデータベースに対する **create connection** コマンドで使用されるコネクション名です。
  - *primary\_schema* はプライマリデータベースのスキーマ名です。
  - *HANA\_schema* は SAP HANA データベースのスキーマ名です。
4. **isql** を使用して Replication Agent に接続し、SAP データディクショナリ (DD) テーブル (DD16S、DD03L、DD02L) を複写対象としてマーク付けします。

```
pdb_setreptable primary_schema.tablename, mark, owner
go
```

*primary\_schema.tablename* は SAP テーブル名です。

---

**注意：** クラスターテーブル CLU4 と VER\_CLUSTR は、複写対象としてマーク付けしないでください。詳細については、「SAP ノート 902817 - クラスターテーブル (論理項目名/物理項目名) における不整合」を参照してください。 <https://css.wdf.sap.corp/sap/support/notes/902817>

---

5. Replication Agent で、各 SAP DD テーブルの複写定義を作成します。

```
rs_create_repdef primary_schema.tablename
go
```

*primary\_schema.tablename* は SAP DD テーブルの名前です。

6. SAP HANA データベース上の SAP DD テーブルをマテリアライズするために、レプリケートコネクションに対するサブスクリプションを作成します。

```
create subscription sub_name
for table_repdef
with replicate at rds.rdb
without holdlock
direct_load init replicate table
with create_or_truncate
user username password pass
go
```

構文の説明は次のとおりです。

- *sub\_name* はサブスクリプションの名前です。
- *table\_repdef* はサブスクリプション *sub\_name* の対象となるテーブル複写定義です。
- *rds.rdb* は前に作成したレプリケート SAP HANA データベース接続です。
- *username* は Replication Agent for UDB に接続するユーザです。
- *pass* は *username* のパスワードです。パスワードを入力する必要があります。

---

**注意：**他にも **create**、**recreate**、**truncate** などの **create subscription** オプションを使用することもできます。

---

7. **isql** を使用して Replication Agent に接続し、SAP テーブル DDNTT、DDNTF、TADIR、SVERS、DD09L、DD12L、および DD17S をマーク付けします。

```
pdb_setreptable primary_schema.tablename, mark, owner
go
```

*primary\_schema.tablename* は SAP テーブル名です。

8. Replication Agent で、複写対象としてマーク付けした各 SAP テーブルの複写定義を作成します。

```
rs_create_repdef primary_schema.tablename
go
```

*primary\_schema.tablename* は SAP テーブルの名前です。

9. SAP HANA データベース上の SAP テーブルをマテリアライズするために、レプリケートコネクションに対するサブスクリプションを作成します。

```
create subscription sub_name
for table_repdef
with replicate at rds.rdb
without holdlock
direct_load init replicate table
with create_or_truncate
user username password pass
go
```

構文の説明は次のとおりです。

- *sub\_name* はサブスクリプションの名前です。
- *table\_repdef* はサブスクリプション *sub\_name* の対象となるテーブル複写定義です。

- *rds.rdb* は前に作成したレプリケート SAP HANA データベース接続です。
- *username* は Replication Agent for UDB に接続するユーザです。
- *pass* は *username* のパスワードです。パスワードを入力する必要があります。

**注意：**他にも **create**、**recreate**、**truncate** などの **create subscription** オプションを使用することもできます。

10. **isql** を介して Replication Agent に接続し、UMGPOCNV テーブルをマークします。

```
pdb_setreptable primary_schema.UMGPOCNV, mark, owner
go
```

11. Replication Agent で、UMGPOCNV テーブルの複写定義を作成します。

```
rs_create_repdef primary_schema.UMGPOCNV
go
```

12. SAP HANA データベース上の UMGPOCNV テーブルをマテリアライズするために、レプリケートコネクションに対するサブスクリプションを作成します。

```
create subscription sub_name
for table_repdef
with replicate at rds.rdb
without holdlock
direct_load init replicate table
with create_or_truncate
user username password pass
go
```

パラメータの意味は、次のとおりです。

- *sub\_name* はサブスクリプションの名前です。
- *table\_repdef* はサブスクリプション *sub\_name* の対象となるテーブル複写定義です。
- *rds.rdb* は前に作成したレプリケート SAP HANA データベース接続です。
- *username* は Replication Agent for UDB に接続するユーザです。
- *pass* は *username* のパスワードです。パスワードを入力する必要があります。

13. その他のすべての SAP テーブルに対するマテリアライゼーションを含むテーブル複写定義とサブスクリプションを作成します。

## シナリオ 2

Oracle データベース上で稼働し、クラスタテーブルが格納された SAP Business Suite アプリケーションから SAP HANA データベースへの複写を設定できます。

以下の手順では、マテリアライゼーションと複写を有効にするように Replication Server を設定する方法について説明します。

1. **varbinary** 値での後続ゼロの複写を有効にします。



```
configure replication server
set varbinary_strip_trailing_zeros to 'off'
```

2. SAP HANA データベース <sid>adm ユーザの \$SAPSYSTEMNAME 環境変数の値を sapsystemname 設定パラメータに設定します。

```
alter connection to rds.rdb
set sapsystemname to 'sid_adm_value'
```

構文の説明は次のとおりです。

- *rds.rdb* は SAP HANA データベースのコネクション名です。
- *sid\_adm\_value* は SID (SAP SYSTEM NAME) の名前です。  
たとえば、SYSTEM NAME は *D01* です。次のように入力する必要があります。

```
alter connection to rds.rdb set sapsystemname to 'D01'
```

3. 次のようにプライマリスキーマをレプリケートスキーマにマップします。

```
alter connection to rds.rdb
set schemamap with decluster from
pds.pdb.primary_schema to HANA_schema
```

構文の説明は次のとおりです。

- *rds.rdb* は SAP HANA データベースのコネクション名です。
- *pds.pdb* はプライマリデータベースに対する **create connection** コマンドで使われるコネクション名です。
- *primary\_schema* はプライマリデータベースのスキーマ名です。
- *HANA\_schema* は SAP HANA データベースのスキーマ名です。

4. **isql** を使用して Replication Agent に接続し、SAP データディクショナリ (DD) テーブル (DD16S、DD03L、DD02L) を複写対象としてマーク付けします。

```
pdb_setreptable primary_schema.tablename, mark, owner
go
```

*primary\_schema.tablename* は SAP テーブル名です。

---

**注意：** クラスターテーブル CLU4 と VER\_CLUSTR は、複写対象としてマーク付けしないでください。詳細については、「SAP ノート 902817 - クラスターテーブル (論理項目名/物理項目名) における不整合」を参照してください。 <https://css.wdf.sap.corp/sap/support/notes/902817>

---

5. Replication Agent で、各 SAP DD テーブルの複写定義を作成します。

```
rs_create_repdef primary_schema.tablename
go
```

*primary\_schema.tablename* は SAP DD テーブルの名前です。

6. SAP HANA データベース上の SAP DD テーブルをマテリアライズするために、レプリケートコネクションに対するサブスクリプションを作成します。

```
create subscription sub_name
for table_repdef
```

```
with replicate at rds.rds
without holdlock
direct_load init replicate table
with create_or_truncate
user username password pass
go
```

構文の説明は次のとおりです。

- *sub\_name* はサブスクリプションの名前です。
- *table\_repdef* はサブスクリプション *sub\_name* の対象となるテーブル複写定義です。
- *rds.rdb* は前に作成したレプリケート SAP HANA データベース接続です。
- *username* は Replication Agent for Oracle に接続するユーザです。
- *pass* は *username* のパスワードです。パスワードを入力する必要があります。

---

**注意：**他にも **create**、**recreate**、**truncate** などの **create subscription** オプションを使用することもできます。

---

7. (Oracle データベース上で稼働している SAP Business Suite の場合のみ)  
**sap\_trim\_len** を **on** に設定して、DDIC 型の長さが 255 バイトを超える RAW や LRAW などの特定の SAP カラムの最初の 2 バイトをトリムします。

```
alter connection to data_server.database
set sap_trim_len to 'on'
```

8. **reblock\_ddntf** を **off** に設定して、Replication Server が DDNTF テーブル内の FIELDS カラムの長さ調整を無効にできるようにします。

```
alter connection to data_server.database
set reblock_ddntf to 'off'
```

9. **isql** を使用して Replication Agent に接続し、SAP テーブル DDNTT、DDNTF、TADIR、SVERS、DD09L、DD12L、および DD17S をマーク付けします。

```
pdb_setreptable primary_schema.tablename, mark, owner
go
```

*primary\_schema.tablename* は SAP テーブルの名前です。

10. Replication Agent で、複写対象としてマーク付けした各 SAP テーブルの複写定義を作成します。

```
rs_create_repdef primary_schema.tablename
go
```

*primary\_schema.tablename* は SAP テーブルの名前です。

11. SAP HANA データベース上の SAP テーブルをマテリアライズするために、レプリケートコネクションに対するサブスクリプションを作成します。

```
create subscription sub_name
for table_repdef
with replicate at rds.rdb
without holdlock
direct_load init replicate table
```

```
with create_or_truncate
user username password pass
go
```

構文の説明は次のとおりです。

- *sub\_name* はサブスクリプションの名前です。
- *table\_repdef* はサブスクリプション *sub\_name* の対象となるテーブル複写定義です。
- *rds.rdb* は前に作成したレプリケート SAP HANA データベース接続です。
- *username* は Replication Agent for Oracle に接続するユーザです。
- *pass* は *username* のパスワードです。パスワードを入力する必要があります。

---

**注意：**他にも **create**、**recreate**、**truncate** などの **create subscription** オプションを使用することもできます。

---

12. **isql** を介して Replication Agent に接続し、UMGPOCNV テーブルをマークします。

```
pdb_setreptable primary_schema.UMGPOCNV, mark, owner
go
```

13. Replication Agent で、UMGPOCNV テーブルの複写定義を作成します。

```
rs_create_repdef primary_schema.UMGPOCNV
go
```

14. SAP HANA データベース上の UMGPOCNV テーブルをマテリアライズするために、レプリケートコネクションに対するサブスクリプションを作成します。

```
create subscription sub_name
for table_repdef
with replicate at rds.rdb
without holdlock
direct_load init replicate table
with create_or_truncate
user username password pass
go
```

パラメータの意味は、次のとおりです。

- *sub\_name* はサブスクリプションの名前です。
- *table\_repdef* はサブスクリプション *sub\_name* の対象となるテーブル複写定義です。
- *rds.rdb* は前に作成したレプリケート SAP HANA データベース接続です。
- *username* は Replication Agent for UDB に接続するユーザです。
- *pass* は *username* のパスワードです。パスワードを入力する必要があります。

15. その他のすべての SAP テーブルに対するマテリアライゼーションを含むテーブル複写定義とサブスクリプションを作成します。



# レプリケートデータサーバとしての IBM DB2 for z/OS

SAP 複写システムにおける、DB2 UDB for z/OS データサーバに固有のプライマリデータベースの問題と考慮事項について説明します。

複写システムの基本的な管理については、『Replication Server 管理ガイド 第 1 巻』を参照してください。

## DB2 UDB for z/OS レプリケートデータサーバの環境

---

ゲートウェイ環境のレプリケートデータサーバとしての DB2 UDB for z/OS は、レプリケート Replication Server からのコマンドを受け入れ、それらのコマンドをレプリケート DB2 UDB データベースに適用する Mainframe Connect DirectConnect for z/OS Option データベースゲートウェイと対話します。

## DB2 UDB for z/OS システム管理

---

Replication Server バージョン 12.0 における HDS の導入によって、**create connection** コマンドの **dsi\_sql\_data\_style** パラメータは無効になりました。

**create connection** コマンドの **dsi\_sql\_data\_style** パラメータは、DB2 UDB for z/OS レプリケートデータベースの一部のデータ変換をサポートするために、以前のバージョンの Replication Server で使用されていました。Replication Server バージョン 12.0 以降では、このパラメータを使用しないでください。デフォルト設定は、"" (空白) です。

---

**注意：** このシステム管理に関する問題は、レプリケート DB2 UDB for z/OS データサーバに固有です。

---

## DB2 UDB for z/OS での複写の干渉と影響

---

レプリケート DB2 UDB への唯一の重大な干渉または影響は、Replication Server のオペレーションをサポートするために、レプリケートデータベースに 3 つのテー

ブルを作成する接続プロファイルによって作成されるデータベースオブジェクトです。

テーブルの内容は次のとおりです。

- レプリケートデータベースによって使用されるソート順と文字セットに関する情報を格納する `RS_INFO`

---

**注意：**このテーブルに対する `INSERT` 文で、使用しているデータサーバに、適切な文字セットとソート順が指定されていることを確認してください。

---

Replication Server バージョン 12.5 以降を使用する場合、レプリケートデータベースのソート順と文字セットは、`RS_INFO` テーブルに記録されなければなりません。そのためには、Replication Server の `rs_get_charset` および `rs_get_sortorder` ファンクションを使用して、レプリケートデータベース内の `RS_INFO` テーブルから文字セットとソート順を取得します。

- レプリケートデータベースに適用する複製トランザクションに関する情報を格納する `RS_LASTCOMMIT`

`RS_LASTCOMMIT` テーブルの各ローは、プライマリデータベースからレプリケートデータベースに配信されコミットされた、最後のトランザクションを示します。Replication Server は、この情報によってすべてのトランザクションが配信されたことを確認します。

Replication Server の `rs_get_lastcommit` ファンクションは、レプリケートデータベース内の最後にコミットされたトランザクションに関する情報を取得します。ASE 以外のレプリケートデータベースについては、`rs_get_lastcommit` は、データベース固有のファンクション文字列クラスで、レプリケートデータベース内の `RS_LASTCOMMIT` テーブルへのアクセスに必要なクエリによって置き換えられます。

- `RS_TICKET_HISTORY` は、Replication Server コマンド `rs_ticket` の実行結果を格納します。プライマリデータベースに対して `rs_ticket` コマンドを発行し、コマンドがプライマリデータベースからレプリケートデータベースまで移動するために要する時間を測定できます。この情報を使用して、Replication Server のパフォーマンス、モジュールのハートビート、複製の正常性、テーブルレベルのクワイズをモニタできます。`rs_ticket` の各実行結果は、レプリケートデータベース内にある `RS_TICKET_HISTORY` テーブルの 1 つのローに保存されます。`RS_TICKET_HISTORY` テーブルの各ローを問い合わせると、個々の `rs_ticket` 実行結果を取得したり、別のローの結果と比較できます。このテーブルに保存されるデータは複製をサポートするために必要ではなく、手動でトランケートして領域を再利用できます。

---

**注意：**`RS_TICKET_HISTORY` テーブルは、Replication Server 15.1 以降でのみ使用できます。

---

## DB2 for z/OS レプリケートデータベースのパーミッション

---

レプリケートデータベースにトランザクションを適用するために、Replication Server では Replication Server の **create connection** コマンドで指定するメンテナンスユーザ ID が必要となります。

メンテナンスユーザ ID は、DB2 UDB for z/OS データサーバに対して定義され、レプリケートデータベースにトランザクションを適用する権限が付与されている必要があります。メンテナンスユーザ ID には、レプリケート DB2 UDB データベースにおける、次のパーミッションが必要です。

- Replication Server の処理に使用するテーブルを作成するための **CREATE TABLE** 権限
- すべてのレプリケートテーブルに対する **UPDATE** 権限と、すべてのレプリケートストアードプロシージャに対する **EXECUTE** 権限

## DB2 UDB for z/OS に関するレプリケートデータベースのコネクティビティ

---

Replication Server データベースコネクション名は、データサーバ名 (**server\_name**) とデータベース名 (**db\_name**) の 2 つの部分から構成されています。

Mainframe Connect DirectConnect for z/OS Option データベースゲートウェイを使用している場合、**server\_name** はデータベースゲートウェイサーバの名前、**db\_name** はレプリケート DB2 UDB データベースの名前です。

レプリケート Replication Server は、Replication Server データベースコネクションで指定されたデータベースゲートウェイの **server\_name** について、interface ファイルエントリを検索します。レプリケート Replication Server は、データベースコネクションで指定された **user\_name** と **password** を使用して、レプリケートデータサーバにログインします。

Replication Server の **interfaces** ファイルにエントリを作成して、Mainframe Connect DirectConnect for z/OS Option データベースゲートウェイサーバが受信を行うホストとポートを指定してください。interfaces ファイルエントリの名前は、Replication Server データベースコネクションの **server\_name** 部分と一致している必要があります。

## DB2 for z/OS におけるレプリケートデータベースの制限事項

---

Mainframe Connect DirectConnect for z/OS Option では、LOB データ型 (BLOB と CLOB) の直接の複製がサポートされています。

また、Mainframe Connect DirectConnect for z/OS Option データベースゲートウェイによって、DB2 UDB のバイナリ値に対して ASCII から EBCDIC への変換が行われるため、Replication Server は値をバイナリ文字列として送信できません。そのため、DB2 UDB for z/OS に複製されるすべての binary または varbinary データ型を、rs\_db2\_char\_for\_bit または rs\_db2\_varchar\_for\_bit データ型にマップする必要があります。

## DB2 for z/OS レプリケートデータベースの設定

---

Replication Server の異機種データ型サポート (HDS) 機能には、レプリケート Replication Server と DB2 UDB for z/OS レプリケートデータベースで HDS 機能を設定するために使用できる設定情報が用意されています。

この設定情報は、インストールプロセスにおいて、接続プロファイルの一部として提供されます。

- Replication Server インストール
  - ファンクション文字列、エラークラス、ユーザ定義データ型の作成
- 接続プロファイル：
  - RSSD へのクラスレベルデータ型変換の適用
  - DB2 UDB for z/OS でのオブジェクトの作成
  - 接続プロパティの設定
- 追加設定
  - ECDA の設定
  - 動的 SQL の設定
  - コマンドのバッチ処理の設定

### 参照：

- RSSD へのクラスレベルデータ型変換 (104 ページ)
- DB2 UDB for z/OS でのオブジェクトおよび接続プロパティ (105 ページ)
- ECDA の設定 (105 ページ)
- 動的 SQL の設定 (105 ページ)
- コマンドのバッチ処理の設定 (105 ページ)



## Replication Server インストール

Replication Server インストールでは、複写をサポートするために必要なファンクション文字列とクラスが自動的にインストールされます。

### ファンクション文字列、エラークラス、ユーザ定義データ型

ファンクション文字列は、Replication Server のデフォルトの **rs\_db2\_function\_class** に追加されます。

ファンクション文字列は、いくつかのデフォルト Replication Server ファンクション文字列を、DB2 UDB for z/OS レプリケートデータベースと通信し、作成されたテーブルとプロシージャにアクセスするように設計されたカスタムファンクション文字列に置き換えます。

---

**注意：** ExpressConnect for Oracle および ExpressConnect for SAP HANA データベースでは、LOB データの管理に LOB ポインタが使用されません。その結果、LOB ポインタの管理に使用される Replication Server システムファンクションは、ExpressConnect for Oracle または ExpressConnect for SAP HANA データベースで使用できません。このようなファンクション (**rs\_get\_textptr**、**rs\_textptr\_init**、**rs\_writetext** など) は ExpressConnect で認識できますが、ファンクションを使用しても Replication Server から無視されます。

---

## 接続プロファイル

接続プロファイルを使用すると、プライマリデータベースとレプリケートデータベースの複写要件を満たす、事前に定義されたプロパティのセットで接続を設定できます。

### 構文

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

### パラメータ

**data\_server** - 複写システムに追加するデータベースを持つデータサーバです。

**database** - 複写システムに追加するデータベースです。

**connection\_profile** - コネクションの設定、RSSD の修正、およびレプリケートデータベースオブジェクトの作成に使用する接続プロファイルを示します。

**version** - 使用する接続プロファイルのバージョンを指定します。

*user* - データベースの Replication Server メンテナンスユーザのログイン名です。Replication Server は、複製データを管理するのにこのログイン名を使用します。ネットワークベースセキュリティを有効に設定していない場合には、ユーザ名を指定する必要があります。

*other\_create\_connection\_options* - プロファイルで指定されない接続オプションの設定 (パスワードの設定など)、またはプロファイルで指定されているオプションの上書き (Replication Server に用意されているファンクション文字列クラスを上書きするカスタムファンクション文字列クラスの指定など) を行うには、他の **create connection** オプションを使用します。**create connection** コマンドの他のオプションのリストについては、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**create connection**」を参照してください。

**display\_only** - **display\_only** は、**using profile** 句とともに使用し、実行されるコマンド、およびそのコマンドを実行するサーバの名前を表示します。**display\_only** を使用した結果については、クライアントログおよび Replication Server ログを参照してください。

### **RSSD へのクラスレベルデータ型変換**

クラスレベル変換によって、プライマリデータ型とレプリケートデータ型、およびデータの変換先のレプリケートデータ型が特定されます。

たとえば、Oracle の DATE は DB2 UDB レプリケートデータベースの TIMESTAMP に変換されます。

クラスレベル変換は、以下に示す適切な名前の接続プロファイルによって DB2 UDB for z/OS レプリケートデータベースに提供されます。

- *rs\_ase\_to\_db2* - Adaptive Server から DB2 UDB へのクラスレベル変換をインストールします。
- *rs\_udb\_to\_db2* - DB2 UDB (for UNIX および Windows) データ型を DB2 UDB for z/OS データ型に変換します。
- *rs\_msss\_to\_db2* - Microsoft SQL Server データ型を DB2 データ型に変換します。
- *rs\_oracle\_to\_db2* - Oracle データ型を DB2 UDB データ型に変換します。

使用可能なすべてのプロファイルを表示するには、**admin show\_connection\_profiles** コマンドを使用します。

**DB2 UDB for z/OS でのオブジェクトおよび接続プロパティ**

接続プロファイルによって、RS\_INFO、RS\_LASTCOMMIT、RS\_TICKET\_HISTORY の各テーブルがレプリケートデータベース内に作成されません。

また、次の接続プロパティも設定されます。

```
set error class rs_db2_error_class
set function string rs_db2_function_class
```

**追加設定**

複写をサポートするための追加設定について説明します。

設定内容は次のとおりです。

- ECDA の設定
- 動的 SQL の設定
- コマンドのバッチ処理の設定

**ECDA の設定**

ECDA および DirectConnect アクセスサービス設定ファイルのプロパティの値について説明します。

ECDA 設定ファイルには、次の設定を使用します。

```
TransactionMode=long
Allocate=connect
SQLTransformation=sybase
```

DB2 UDB for z/OS レプリケートデータベースへの複写に Mainframe Connect DirectConnect for z/OS Option データベースゲートウェイを使用している場合は、DirectConnect の db2.cfg アクセスサービス設定ファイルで、次のプロパティを設定します。

```
SQLTransformation=passthrough
TransactionMode=long
```

**動的 SQL の設定**

ECDA 12.6.1 およびそれ以降では、動的 SQL がサポートされています。

**コマンドのバッチ処理の設定**

コマンドのバッチ処理を使用すると、Replication Server は複数のコマンドを 1 つのコマンドバッチとしてデータサーバに送信できます。

セミコロン (;) で区切ることによって、言語ファンクション文字列の出力テンプレートに複数のコマンドを入力できます。データベースがコマンドバッチを使用できるように設定されている場合 (デフォルトの設定) は、Replication Server は、こ

のセミコロンをその接続の DSI コマンドセパレータ文字に置き換えてから、単一のバッチ内のファンクション文字列としてデータサーバに送信します。

セパレータ文字は、**dsi\_cmd\_separator** オプション (**alter connection** コマンド) に定義されています。データベースへの接続がバッチを使用できるように設定されていない場合は、Replication Server は、ファンクション文字列内のコマンドを一度に 1 つずつデータサーバに送信します。データベースのバッチを有効または無効にするには、**alter connection** コマンドを使用します。

コマンドのバッチ処理を使用するには、次のように入力します。

```
set batch = on
set dsi_cmd_separator = ;
set batch_begin = off
use_batch_markers = on
```

『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**alter connection**」を参照し、**batch** オプションおよび **dsi\_cmd\_separator** オプションを **alter connection** コマンドを使用して設定する方法についての情報を取得します。

# Replicate Data Server としての Linux、UNIX、および Windows に関する IBM DB2

SAP 複写システムにおける、Linux、UNIX、および Windows 用の DB2 UDB データサーバに固有のプライマリデータベースの問題と考慮事項について説明します。

複写システムの基本的な管理については、『Replication Server 管理ガイド 第 1 巻』を参照してください。

## DB2 UDB レプリケートデータサーバ

---

複写システム内のレプリケートデータサーバとしての DB2 UDB は、ECDA Option for ODBC データベースゲートウェイと対話します。

ECDA Option for ODBC は、レプリケート Replication Server からコマンドを受け入れ、それらのコマンドを DB2 UDB サーバにあるデータベースに適用します。

## DB2 UDB での複写の干渉と影響

---

DB2 UDB レプリケートデータベースへの唯一の重大な干渉または影響は、Replication Server のオペレーションをサポートするために、レプリケートデータベースに 3 つのテーブルを作成する接続プロファイルによって作成されるデータベースオブジェクトです。

テーブルの内容は次のとおりです。

- レプリケートデータベースによって使用されるソート順と文字セットに関する情報を格納する RS\_INFO

---

**注意：** RS\_INFO に対する INSERT 文で、使用している DB2 UDB サーバに、適切な文字セットとソート順が指定されていることを確認してください。

---

Replication Server バージョン 12.0 以降を使用する場合、レプリケートデータベースのソート順と文字セットは、RS\_INFO テーブルに記録されなければなりません。

Replication Server の `rs_get_charset` および `rs_get_sortorder` ファンクションは、レプリケートデータベース内の RS\_INFO テーブルから文字セットとソート順を取得します。

- レプリケートデータベースに適用する複写トランザクションに関する情報を格納する RS\_LASTCOMMIT

RS\_LASTCOMMIT テーブルの各ローは、プライマリデータベースからレプリケートデータベースに配信されコミットされた、最後のトランザクションを示します。Replication Server は、この情報によってすべてのトランザクションが配信されたことを確認します。

Replication Server の **rs\_get\_lastcommit** ファンクションは、レプリケートデータベース内の最後にコミットされたトランザクションに関する情報を取得します。ASE 以外のレプリケートデータベースについては、**rs\_get\_lastcommit** 関数は、データベース固有のファンクション文字列クラスで、レプリケートデータベース内の RS\_LASTCOMMIT テーブルへのアクセスに必要なクエリによって置き換えられます。

- RS\_TICKET\_HISTORY は、Replication Server コマンド **rs\_ticket** の実行結果を格納します。プライマリデータベースに対して **rs\_ticket** コマンドを発行し、コマンドがプライマリデータベースからレプリケートデータベースまで移動するために要する時間を測定できます。この情報を使用して、Replication Server のパフォーマンス、モジュールのハートビート、複製の正常性、テーブルレベルのクワイズをモニタします。**rs\_ticket** の各実行結果は、レプリケートデータベース内にある RS\_TICKET\_HISTORY テーブルの 1 つのローに保存されます。

RS\_TICKET\_HISTORY テーブルの各ローを問い合わせると、個々の **rs\_ticket** 実行結果を取得したり、別のローの結果と比較できます。このテーブルに保存されるデータは複製をサポートするために必要ではなく、手動でトランケートして領域を再利用できます。

---

**注意：** RS\_TICKET\_HISTORY テーブルは、Replication Server 15.1 以降でのみ使用できます。

---

## DB2 UDB レプリケートデータベースのパーミッションと制限

レプリケートデータベースにトランザクションを適用するために、Replication Server では Replication Server の **create connection** コマンドで指定するメンテナンスユーザ ID が必要となります。

メンテナンスユーザ ID は DB2 UDB サーバで定義され、レプリケートデータベースにトランザクションを適用する権限が付与されている必要があります。メンテナンスユーザ ID には、DB2 UDB レプリケートデータベースにおける、次のパーミッションが必要です。

- Replication Server の処理に使用するテーブルを作成するための **CREATE TABLE** 権限
- すべてのレプリケートテーブルに対する **UPDATE** 権限

Replication Server から ECDA Option for ODBC への直接の LOB データ型 (BLOB、CLOB、DECLOB、LONG VARCHAR、および LONG VARCHAR) の複写は、サポートされていません。

## DB2 UDB レプリケートデータベースのコネクティビティ

---

Replication Server データベースコネクション名は、データサーバ名 (**server\_name**) とデータベース名 (**db\_name**) の 2 つの部分から構成されています。**server\_name** は ECDA Option for ODBC データベースゲートウェイサーバの名前、**db\_name** は DB2 UDB レプリケートデータベースの名前です。

レプリケート Replication Server は、Replication Server データベースコネクションで指定されたデータベースゲートウェイの **server\_name** について、**interfaces** ファイルエントリを検索します。レプリケート Replication Server は、データベースコネクションで指定された **user\_name** と **password** を使用して、レプリケートデータサーバにログインします。

Replication Server の **interfaces** ファイルにエントリを作成して、ECDA Option for ODBC データベースゲートウェイサーバが受信を行うホストとポートを指定してください。**interfaces** ファイルエントリの名前は、Replication Server データベースコネクションの **server\_name** 部分と一致している必要があります。

## DB2 UDB レプリケートデータベースの設定

---

Replication Server の異機種データ型サポート (HDS) 機能には、レプリケート Replication Server と DB2 UDB レプリケートデータベースで HDS 機能を設定するために使用できる設定情報が用意されています。

この設定情報は、インストールにおいて、接続プロファイルの一部として提供されます。

- Replication Server インストール
  - ファンクション文字列、エラークラス、ユーザ定義データ型の作成
- 接続プロファイル：
  - RSSD へのクラスレベルデータ型変換の適用
  - DB2 UDB レプリケートデータベースでのオブジェクトの作成
  - 接続プロパティの設定
- 追加設定
  - ECDA の設定 (必須)
  - 動的 SQL の設定 (オプション)

- コマンドのバッチ処理の設定 (オプション)

**参照：**

- RSSD へのクラスレベルデータ型変換 (111 ページ)
- DB2 UDB レプリケートデータベースでのオブジェクトおよび接続プロパティ (111 ページ)

## **Replication Server インストール**

Replication Server インストールでは、複写をサポートするために必要なファンクション文字列とクラスが自動的にインストールされます。

### **ファンクション文字列、エラークラス、ユーザ定義データ型**

ファンクション文字列は、Replication Server のデフォルトの **rs\_udb\_function\_class** に追加されます。

ファンクション文字列は、いくつかのデフォルト Replication Server ファンクション文字列を、DB2 UDB レプリケートデータベースと通信し、作成されたテーブルとプロシージャにアクセスするように設計されたカスタムファンクション文字列に置き換えます。

エラークラスに定義されているエラーアクションを調べるには、『Replication Server リファレンスマニュアル』の「RSSD ストアドプロシージャ」の「rs\_helperror」を参照してください。

## **接続プロファイル**

接続プロファイルを使用すると、プライマリデータベースとレプリケートデータベースの複写要件を満たす、事前に定義されたプロパティのセットでコネクションを設定できます。

### **構文**

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

### **パラメータ**

**data\_server** - 複写システムに追加するデータベースを持つデータサーバです。

**database** - 複写システムに追加するデータベースです。

**connection\_profile** - コネクションの設定、RSSD の修正、およびレプリケートデータベースオブジェクトの作成に使用する接続プロファイルを示します。

**version** - 使用する接続プロファイルのバージョンを指定します。



*user* - データベースの Replication Server メンテナンスユーザのログイン名です。Replication Server は、複製データを管理するのにこのログイン名を使用します。ネットワークベースセキュリティを有効に設定していない場合には、ユーザ名を指定する必要があります。

*other\_create\_connection\_options* - プロファイルで指定されない接続オプションの設定 (パスワードの設定など)、またはプロファイルで指定されているオプションの上書き (Replication Server に用意されているファンクション文字列クラスを上書きするカスタムファンクション文字列クラスの指定など) を行うには、他の **create connection** オプションを使用します。**create connection** コマンドの他のオプションのリストについては、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**create connection**」を参照してください。

**display\_only** - **display\_only** は、**using profile** 句とともに使用し、実行されるコマンド、およびそのコマンドを実行するサーバの名前を表示します。**display\_only** を使用した結果については、クライアントログおよび Replication Server ログを参照してください。

### RSSD へのクラスレベルデータ型変換

クラスレベル変換によって、プライマリデータ型とデータの変換先のレプリケートデータ型が特定されます (たとえば、Microsoft SQL Server の binary は DB2 UDB の CHAR FOR BIT DATA に変換されます)。

以下の接続プロファイルは、DB2 UDB レプリケートデータベースにクラスレベル変換を提供します。

- *rs\_ase\_to\_udb* - Adaptive Server から DB2 UDB へのクラスレベル変換をインストールします。
- *rs\_db2\_to\_udb* - DB2 for z/OS データ型を DB2 UDB データ型に変換します。
- *rs\_msss\_to\_udb* - Microsoft SQL Server データ型を DB2 UDB データ型に変換します。
- *rs\_oracle\_to\_udb* - Oracle データ型を DB2 UDB データ型に変換します。

使用可能なすべてのプロファイルを表示するには、**admin show\_connection\_profiles** コマンドを使用します。

### DB2 UDB レプリケートデータベースでのオブジェクトおよび接続プロパティ

接続プロファイルによって、RS\_INFO、RS\_LASTCOMMIT、RS\_TICKET\_HISTORY の各テーブルがレプリケートデータベース内に作成されません。

接続プロファイルによって次の接続プロパティが設定されます。

```
set error class rs_udb_error_class
set function string rs_udb_function_class
```

## 追加設定

複写をサポートするための追加設定について説明します。

設定内容は次のとおりです。

- ECDA の設定 (必須)

ECDA 設定ファイルには、次の設定を使用します。

```
Transaction Mode = long
allocate = connect
```

```
SQL transformation = Sybase
```

- 動的 SQL の設定 (オプション)

動的 SQL は Replication Server 15.0.1 でサポートされ、DirectConnect UDB 12.6.1 ESD #2 以降を必要とします。

- コマンドのバッチ処理の設定 (オプション)

コマンドのバッチ処理を使用すると、Replication Server は複数のコマンドを 1 つのコマンドバッチとしてデータサーバに送信できます。セミコロン (;) で区切ることによって、言語ファンクション文字列の出力テンプレートに複数のコマンドを入力できます。データベースがコマンドバッチを使用できるように設定されている場合 (デフォルトの設定) は、Replication Server は、このセミコロンをその接続の DSI コマンドセパレータ文字に置き換えてから、単一のバッチ内のファンクション文字列としてデータサーバに送信します。セパレータ文字は、**dsi\_cmd\_separator** オプション (**alter connection** コマンド) に定義されています。

データベースへの接続がバッチを使用できるように設定されていない場合は、Replication Server は、ファンクション文字列内のコマンドを一度に 1 つずつデータサーバに送信します。データベースのバッチを有効または無効にするには、**alter connection** コマンドを使用します。

コマンドのバッチ処理を使用するには、次のように入力します。

```
set batch = on
```

```
set dsi_cmd_separator = ;
```

```
set batch_begin = off
```

```
use_batch_markers = on
```

**batch** オプションおよび **dsi\_cmd\_separator** オプションの詳細については、『Replication Server リファレンスマニュアル』の「alter connection」を参照してください。

## IBM DB2 レプリケートデータベースの並列 DSI スレッド

---

異機種間複写環境では、並列 DSI を使用すると、レプリケートデータベースのコミット順はプライマリデータベースと必ず同じになります。デッドロックが発生した場合、DSI によってデッドロック競合が解決され、Replication Server ではトランザクションをロールバックして再度実行します。

Replication Server は、トランザクションのコミット順序を維持し、次のいずれかの方法で、同時に並列して実行されているトランザクションでの更新の競合を検出できます。

- 内部的に、Replication Server の内部テーブルとファンクション文字列を使用する。
- 外部的に、レプリケートデータベースの `rs_threads` システムテーブルを使用する。

外部コミット制御では、次のルールに従う必要があります。

- 異なるセッションが同じローで動作する場合、セッション 1 の **update** オペレーションによってセッション 2 の **select** オペレーションはブロックされます。
- 異なるセッションが異なるローで動作する場合、セッション 1 の **update** オペレーションによってセッション 2 の **update** はブロックされません。

内部コミット制御メソッドは、依存する条件が少ないため、外部コミット制御よりも優れています。デッドロックが発生した場合、内部コミット制御では Replication Server が単一のトランザクションをロールバックできますが、外部コミット制御ではすべてのトランザクションをロールバックします。

Replication Server で、並列処理が最大限に実行され、トランザクション間の競合が最小限に抑えられるようにする方法は、他にもあります。たとえば、トランザクションの逐次化メソッドを使用すると、システムが競合を起こすことなく処理できる並列度を選択できます。

並列 DSI スレッドを使用する方法の詳細については、『Replication Server 管理ガイド 第 2 巻』の「パフォーマンスチューニング」を参照してください。

### 外部コミット制御

Replication Server では、レプリケートデータベースが IBM DB2 UDB の場合、ローレベルのロックを使用して `rs_threads` を作成します。

デフォルトでは、ローレベルのロックは “on” です。例：

```
create table rs_threads (id int,seq int)
```

```
create unique index thread_index on rs_threads(id) cluster
```

独立性レベルが 3 の場合、次のファンクション文字列を使用する必要があります。

```
select seq from rs_threads where id = ? with cs
```

構文の説明は次のとおりです。

**cs** はカーソル安定性を示し、IBM DB2 UDB のデフォルトの独立性レベルです。

### 内部コミット制御

Replication Server は、**rs\_dsi\_check\_thread\_lock** ファンクションを使用して、現在の DSI エグゼキュータスレッドが別のレプリケートデータベースのプロセスをブロックしているかどうかをチェックします。

例：

```
select count(*) as seq from table(snapshot_lock('',-1))
as T1 where TABLE_NAME!= '' AND AGENT_ID in (SELECT
AGENT_ID FROM TABLE(SNAPSHOT_APPL_INFO('',-1)) as T2
WHERE APPL_ID = (VALUES APPLICATION_ID()))
```

IBM DB2 UDB では、以下を使用して、現在のセッションのロック情報を選択します。

```
select agent_id from table(snapshot_lock('',-1)) as locktable
```

現在のセッション ID を取得するには、以下を使用します。

```
SELECT APPL.AGENT_ID FROM TABLE(SNAPSHOT_APPL_INFO('',
-1)) AS APPL WHERE APPL.APPL_ID = (VALUES APPLICATION_ID())
```

### トランザクションの逐次化メソッド

Replication Server には、並列化のレベルを指定するための 3 つの逐次化メソッドが用意されています。複写環境、そして並列スレッド間で予想される競合の度合いに応じて、逐次化メソッドを選択してください。

各逐次化メソッドでは、トランザクションが直前のトランザクションのコミットを待機しなければならなくなる前に、そのトランザクションをどの程度の量だけ開始できるようにするかを定義します。

逐次化メソッドは、次のとおりです。

- **no\_wait**
- **wait\_for\_start**
- **wait\_for\_commit**
- **wait\_after\_commit**

**alter connection** コマンドと **dsi\_serialization\_method** パラメータを使用して、データベースコネクションに逐次化メソッドを選択します。たとえば、次のコマンドを

入力して、**wait\_for\_commit** 逐次化メソッドをコネクションに選択します。このコネクションは pubs2 データベースへのもので、これは SYDNEY\_DS データサーバにあります。

```
alter connection to SYDNEY_DS.pubs2
set dsi_serialization_method to 'wait_for_commit'
```

トランザクションは次の3つの部分で構成されています。

- 先頭部分。
- トランザクション本体。**insert**、**update**、**delete** などのオペレーションで構成されます。
- トランザクションの末尾部分。コミットまたはロールバックで構成されます。

逐次化メソッドは、コミットの一貫性を保ちながら、トランザクションの先頭部分が直前のトランザクションのコミット準備の完了を待機するかどうか、またはトランザクションの先頭部分がそれよりも前に処理されるようにするかどうかを定義します。

### **no\_wait**

**no\_wait** メソッドは、直前のトランザクションのコミットを待機せずに次のトランザクションを開始するよう DSI に指示します。

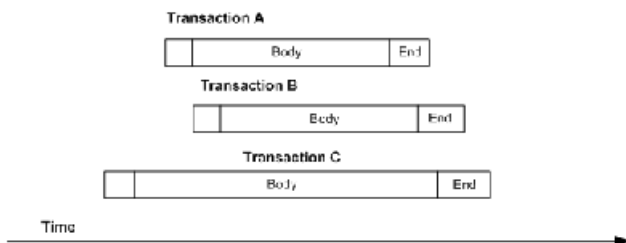
**no\_wait** を使用する場合、使用するプライマリアプリケーションが更新の競合を回避するよう設計されているか、**dsi\_partitioning\_rule** を効果的に使用して競合を減少させたり、取り除いたりしていることが前提となります。Adaptive Server は、**dsi\_isolation\_level** が **3** に設定されなければ、更新ロックを保持しません。このメソッドは、並列トランザクション間の競合がほとんどないことを前提とし、結果的には「**wait\_for\_commit** 逐次化メソッドによるスレッドのタイミング」図に示すように並列に近い形で実行されます。

---

**注意：** **dsi\_commit\_control** を “on” に設定している場合は、**dsi\_serialization\_method** は **no\_wait** にのみ設定できます。

---

図 3 : **no\_wait** 逐次化メソッドによるスレッドのタイミング



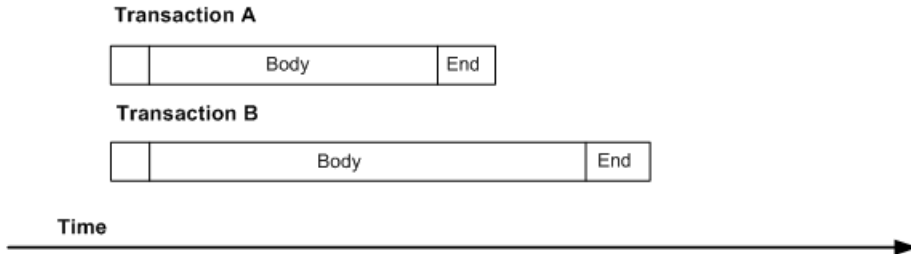
**no\_wait** を指定すると、パフォーマンスが向上する可能性が高くなりますが、競合が発生する危険性も高くなります。

**wait\_for\_start**

**wait\_for\_start** は、開始直前にコミットするようにスケジュールされているトランザクションが開始した直後に、トランザクションを開始できることを指定します。

**dsi\_serialization\_method** を **wait\_for\_start** に設定し、同時に **dsi\_commit\_control** を **off** に設定することがないようにしてください。

**図 4 : wait\_for\_start 逐次化メソッドによるスレッドのタイミング**

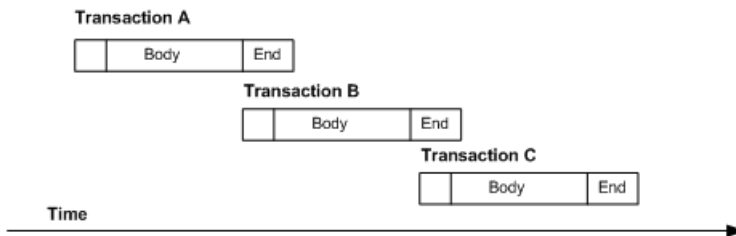


**wait\_for\_commit**

**wait\_for\_commit** メソッドでは、直前のトランザクションの処理が正常に完了してコミットの送信が開始されるまで、次のスレッドのトランザクショングループが処理のために送信されることはありません。

デフォルトの設定です。このメソッドは、並列トランザクション間にかかなりの競合があることを前提とし、図に示すように実行にずれが生じます。

**図 5 : wait\_for\_commit 逐次化メソッドによるスレッドのタイミング**

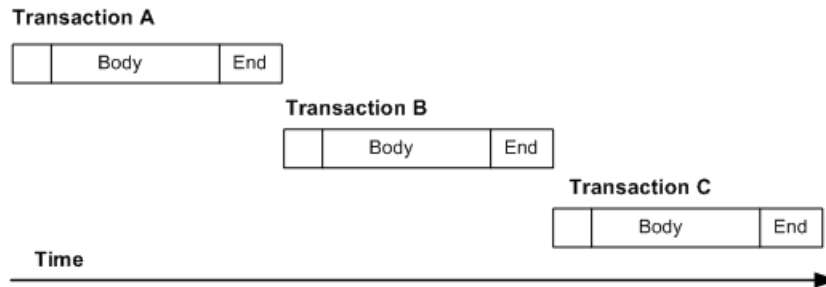


このメソッドでは、1つのトランザクションのコミットの準備ができるまで待機してから次のトランザクションを開始するように DSI に指示し、トランザクションの逐次化を維持します。最初のトランザクションは必要なロックをすでに保持しているため、最初のトランザクションのコミット中に次のトランザクションをレプリケートデータサーバに送信できます。

**wait\_after\_commit**

**wait\_after\_commit** は、直前にコミットするようにスケジュールされているトランザクションのコミットが完了するまで、トランザクションを開始できないよう指定します。

**図 6 : wait\_after\_commit 逐次化メソッドによるスレッドのタイミング**







# レプリケートデータサーバとしての Microsoft SQL Server

SAP 複写システムにおける、Microsoft SQL Server データサーバに固有のレプリケートデータベースの問題と考慮事項について説明します。

## Microsoft SQL Server レプリケートデータサーバ

---

レプリケートデータサーバとしての Microsoft SQL Server は、ECDA Option for ODBC データベースゲートウェイと対話します。

ECDA Option for ODBC サーバは、レプリケート Replication Server からコマンドを受け入れ、それらのコマンドを Microsoft SQL Server データベースに適用します。

---

**注意：** ECDA Option for ODBC では、Replication Server から Microsoft SQL Server データベースへの直接の LOB データ型 (image、ntext、および text) の複写がサポートされています。

---

## Microsoft SQL Server での複写の干渉と影響

---

Microsoft SQL Server レプリケートデータベースへの唯一の重大な干渉または影響は、Replication Server レプリケートデータベースのオペレーションをサポートするために接続プロファイルによって作成されるデータベースオブジェクトです。

接続プロファイルによって、Replication Server のオペレーションをサポートする 3 つのテーブルがレプリケートデータベース内に作成されます。

- レプリケートデータベースによって使用されるソート順と文字セットに関する情報を格納する RS\_INFO。

---

**注意：** insert 文 (RS\_INFO テーブル) で、使用している Microsoft SQL Server データサーバに、適切な文字セットとソート順が指定されていることを確認してください。

---

Replication Server バージョン 12.0 以降を使用する場合、レプリケートデータベースのソート順と文字セットは、RS\_INFO テーブルに記録されなければなりません。

Replication Server の **rs\_get\_charset** および **rs\_get\_sortorder** ファンクションは、レプリケートデータベース内の **RS\_INFO** テーブルから文字セットとソート順を取得します。

- レプリケートデータベースに適用する複製トランザクションに関する情報を格納する **RS\_LASTCOMMIT**。

**RS\_LASTCOMMIT** テーブルの各ローは、プライマリデータベースからレプリケートデータベースに配信されコミットされた、最後のトランザクションを示します。Replication Server は、この情報によってすべてのトランザクションが配信されたことを確認します。

Replication Server の **rs\_get\_lastcommit** ファンクションは、レプリケートデータベース内の最後にコミットされたトランザクションに関する情報を取得します。ASE 以外のレプリケートデータベースについては、**rs\_get\_lastcommit** 関数は、データベース固有のファンクション文字列クラスで、レプリケートデータベース内の **RS\_LASTCOMMIT** テーブルへのアクセスに必要なクエリによって置き換えられます。

- RS\_TICKET\_HISTORY** は、Replication Server コマンド **rs\_ticket** の実行結果を格納します。**rs\_ticket** コマンドは、コマンドがプライマリデータベースからレプリケートデータベースまで移動するために要する時間を測定するために、プライマリデータベースに対して発行できます。この情報を使用して、Replication Server のパフォーマンス、モジュールのハートビート、複製の正常性、テーブルレベルのクワイズをモニタできます。**rs\_ticket** の各実行結果は、レプリケートデータベース内にある **RS\_TICKET\_HISTORY** テーブルの 1 つのローに保存されます。**RS\_TICKET\_HISTORY** テーブルの各ローを問い合わせると、個々の **rs\_ticket** 実行結果を取得したり、別のローの結果と比較できます。このテーブルに保存されるデータは、手動でトランケートできます。

---

**注意：** **RS\_TICKET\_HISTORY** テーブルは、Replication Server リリース 15.1 以降でのみ使用できます。

---

## Microsoft SQL Server におけるレプリケートデータベースの制限事項

---

Microsoft SQL Server では、サーバの起動オプションによって 28 桁または 38 桁の精度がサポートされます。デフォルトの精度は 28 桁です。

しかし、Replication Server には、デフォルトの 28 桁の精度をサポートするユーザ定義データ型 (UDD) が用意されていません。

サーバの設定を超える精度で Microsoft SQL Server データベースに数値データを複製しようとする、Replication Server は次のエラーを返します。

```

E. 2007/12/14 11:14:58. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsiqmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|Data Source Name=mssql70_devdb|
SQLState=22003|Native Error=1007|Message=
[Microsoft] [ODBC SQL Server Driver][SQL
Server]The number
'999999999999999999.999999999999999999' is out
of the range for numeric representation (maximum
precision 28).
[Message Iteration=2|SQLState=22003|Native
Error=[Message=[Microsoft][ODBC SQL Server
Driver][SQL Server]The number
'0.999999999999999999999999999999999999' is out
of the range for numeric representation (maximum
precision 28).] <DCA>'

```

Microsoft SQL Server は Adaptive Server Enterprise と同じように identity カラムをサポートしているため、identity への挿入をオフおよびオンに設定する Replication Server のファンクション文字列が、Microsoft SQL Server で正常に動作します。ただし、28 桁の数値精度をサポートするには、SAP のネイティブ numeric データ型を rs\_msss\_numeric データ型に変換する必要があり、その結果、ID 特性が失われます。

Microsoft SQL Server レプリケートデータベースで 28 桁の精度をサポートするために、numeric から rs\_msss\_numeric へのデータ型変換を使用するように選択した場合、レプリケートテーブルは、そのデータを ID として受け取る数値カラムを宣言できません。

レプリケート Microsoft SQL Server テーブルが、変換されたデータを ID として受け取る数値カラムを宣言した場合、Replication Server は次のエラーを返します。

```

E. 2007/12/14 12:05:39. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsiqmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|Data Source Name=mssql70_devdb|SQL
Function=INSERT|SQLState=23000|Native
Error=544|Message=[Microsoft][ODBC SQL Server
Driver][SQL Server]Cannot insert explicit value
for identity column in table 'ase_alltypes' when
IDENTITY_INSERT is set to OFF.] <DCA>'

```

## Microsoft SQL Server レプリケートデータベースのパーミッション

---

レプリケートデータベースにトランザクションを適用するために、Replication Server では Replication Server の **create connection** コマンドで指定するメンテナンスユーザ ID が必要となります。

メンテナンスユーザ ID は Microsoft SQL Server データサーバで定義され、レプリケートデータベースにトランザクションを適用する権限が付与されている必要があります。メンテナンスユーザ ID には、Microsoft SQL Server レプリケートデータベースにおける、次のパーミッションが必要です。

- Replication Server の処理に使用するテーブルを作成するための **create table** 権限
- すべてのレプリケートテーブルに対する **update** 権限
- すべてのレプリケートストアプロシージャに対する **execute** 権限

## Microsoft SQL Server に関するレプリケートデータベースのコネクティビティ

---

Replication Server データベースコネクション名は、データサーバ名 (**server\_name**) とデータベース名 (**db\_name**) の 2 つの部分から構成されています。

**server\_name** は ECDA for ODBC データベースゲートウェイサーバの名前、**db\_name** は Microsoft SQL Server レプリケートデータベースの名前です。

レプリケート Replication Server は、Replication Server データベースコネクションで指定されたデータベースゲートウェイの **server\_name** について、**interfaces** ファイルエントリを検索します。レプリケート Replication Server は、データベースコネクションで指定された **user\_name** と **password** を使用して、レプリケートデータサーバにログインします。

Replication Server の **interfaces** ファイルにエントリを作成して、ECDA Option for ODBC データベースゲートウェイサーバが受信を行うホストとポートを指定してください。**interfaces** ファイルエントリの名前は、Replication Server データベースコネクションの **server\_name** 部分と一致している必要があります。

## Microsoft SQL Server レプリケートデータベースの設定

---

Replication Server の異機種データ型サポート (HDS) 機能には、レプリケート Replication Server と Microsoft SQL Server レプリケートデータベースで HDS 機能を設定するために使用できる設定情報が用意されています。

この設定情報は、インストールにおいて、接続プロファイルの一部として提供されます。

- Replication Server インストール
  - ファンクション文字列、エラークラス、ユーザ定義データ型の作成
- 接続プロファイル：
  - RSSD へのクラスレベルデータ型変換の適用
  - Microsoft SQL Server データベースでのオブジェクトの作成
  - 接続プロパティの設定
- 追加設定
  - ECDA の設定
  - 動的 SQL の設定
  - コマンドのバッチ処理の設定

### 参照：

- RSSD へのクラスレベルデータ型変換 (125 ページ)
- Microsoft SQL Server データベースでのオブジェクトおよび接続プロパティ (125 ページ)

## Replication Server インストール

---

Replication Server インストールでは、複写をサポートするために必要なファンクション文字列とクラスが自動的にインストールされます。

### ファンクション文字列、エラークラス、ユーザ定義データ型

ファンクション文字列は、Replication Server のデフォルトの **rs\_msss\_function\_class** に追加されます。

ファンクション文字列は、いくつかのデフォルト Replication Server ファンクション文字列を、Microsoft SQL Server と通信し、作成されたテーブルとプロシージャにアクセスするように設計されたカスタムファンクション文字列に置き換えます。

エラークラスに定義されているエラーアクションを調べるには、『Replication Server リファレンスマニュアル』の「RSSD ストアドプロシージャ」の「rs\_helperror」を参照してください。

## 接続プロファイル

接続プロファイルを使用すると、プライマリデータベースとレプリケートデータベースの複製要件を満たす、事前に定義されたプロパティのセットでコネクションを設定できます。

### 構文

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

### パラメータ

*data\_server* - 複製システムに追加するデータベースを持つデータサーバです。

*database* - 複製システムに追加するデータベースです。

*connection\_profile* - コネクションの設定、RSSD の修正、およびレプリケートデータベースオブジェクトの作成に使用する接続プロファイルを示します。

*version* - 使用する接続プロファイルのバージョンを指定します。

*user* - データベースの Replication Server メンテナンスユーザのログイン名です。Replication Server は、複製データを管理するのにこのログイン名を使用します。ネットワークベースセキュリティを有効に設定していない場合には、ユーザ名を指定する必要があります。

*other\_create\_connection\_options* - プロファイルで指定されない接続オプションの設定 (パスワードの設定など)、またはプロファイルで指定されているオプションの上書き (Replication Server に用意されているファンクション文字列クラスを上書きするカスタムファンクション文字列クラスの指定など) を行うには、他の **create connection** オプションを使用します。**create connection** コマンドの他のオプションのリストについては、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**create connection**」を参照してください。

**display\_only** - **display\_only** は、**using profile** 句とともに使用し、実行されるコマンド、およびそのコマンドを実行するサーバの名前を表示します。**display\_only** を使用した結果については、クライアントログおよび Replication Server ログを参照してください。

### RSSD へのクラスレベルデータ型変換

クラスレベル変換によって、プライマリデータ型とデータの変換先のレプリケートデータ型が特定されます (たとえば、DB2 UDB の `TIMESTAMP` は Microsoft SQL Server の `datetime` に変換されます)。

---

**注意：** これらの変換は、Replication Server のパフォーマンスに影響を与える可能性があります。使用している特定のプライマリデータベースとレプリケートデータベースに必要な変換だけを、RSSD に適用してください。

---

以下の接続プロファイルは、Microsoft SQL Server レプリケートデータベースにクラスレベル変換を提供します。

- `rs_db2_to_msss` - Adaptive Server から Microsoft SQL Server へのクラスレベル変換をインストールします。
- `rs_ase_to_msss.sql` - Adaptive Server データ型を Microsoft SQL Server データ型に変換します。
- `rs_udb_to_msss` - DB2 UDB (for UNIX および Windows) データ型を Microsoft SQL Server データ型に変換します。
- `rs_oracle_to_msss` - Oracle データ型を Microsoft SQL Server データ型に変換します。

使用可能なすべてのプロファイルを表示するには、**admin show\_connection\_profiles** コマンドを使用します。

### Microsoft SQL Server データベースでのオブジェクトおよび接続プロパティ

接続プロファイルによって、`RS_INFO`、`RS_LASTCOMMIT`、`RS_TICKET_HISTORY` の各テーブルがレプリケートデータベース内に作成されます。

接続プロファイルによって次の接続プロパティが設定されます。

```
set error class rs_msss_error_class
set function string rs_msss_function_class
```

## 追加設定

複写をサポートするための追加設定について説明します。

設定内容は次のとおりです。

- ECDA の設定

ECDA 設定ファイルには、次の設定を使用します。

```
Transaction Mode = long
allocate = connect
```

```
SQL transformation = Sybase
```

## レプリケートデータサーバとしての Microsoft SQL Server

set batch が “on” の場合、以下も指定する必要があります。

```
DelimitSqlRequests = yes
```

レプリケートテーブルに tinyint データ型が存在する場合、ECDA Microsoft SQL Server の Microsoft SQL サービスの Datatype Conversion セクションに次のパラメータを追加する必要があります。

```
TinyIntResults=tinyint
```

- 動的 SQL の設定

動的 SQL は Replication Server 15.0.1 でサポートされ、ECDA Option for ODBC 12.6.1 ESD #2 以降を必要とします。

- コマンドのバッチ処理の設定

コマンドのバッチ処理を使用すると、Replication Server は複数のコマンドを 1 つのコマンドバッチとしてデータサーバに送信できます。セミコロン (;) で区切ることによって、言語ファンクション文字列の出力テンプレートに複数のコマンドを入力できます。データベースがコマンドバッチを使用できるように設定されている場合 (デフォルトの設定) は、Replication Server は、このセミコロンをそのコネクションの DSI コマンドセパレータ文字に置き換えてから、単一のバッチ内のファンクション文字列としてデータサーバに送信します。セパレータ文字は、**dsi\_cmd\_separator** オプション (**alter connection** コマンド) に定義されています。

データベースへのコネクションがバッチを使用できるように設定されていない場合は、Replication Server は、ファンクション文字列内のコマンドを一度に 1 つずつデータサーバに送信します。データベースのバッチを有効または無効にするには、**alter connection** コマンドを使用します。

コマンドのバッチ処理を使用するには、次のように入力します。

```
batch = on
```

```
batch_begin = on or off
```

**batch\_begin** に on を使用すると、ネットワーク転送の数が減ります。

```
use_batch_markers = off
```

追加のバッチマーカは必要ありません。

set batch が “on” の場合、以下の設定も指定する必要があります。

```
dsi_cmd_seperator set = ;
```

この設定を指定しない場合、ECDA は各バッチ後のコミットを無視し、DSI コネクションの消失後にすべての複製要求がロールバックされます。

**batch** オプションおよび **dsi\_cmd\_separator** オプションの詳細については、『Replication Server リファレンスマニュアル』の「alter connection」を参照してください。



## Microsoft SQL Server レプリケートデータベースの並列 DSI スレッド

---

異機種間複写環境では、並列 DSI を使用すると、レプリケートデータベースのコミット順はプライマリデータベースと必ず同じになります。

デッドロックが発生した場合、DSI によってデッドロック競合が解決され、Replication Server ではトランザクションをロールバックして再度実行します。

Replication Server は、トランザクションのコミット順序を維持し、次のいずれかの方法で、同時に並列して実行されているトランザクションでの更新の競合を検出できます。

- 内部的に、Replication Server の内部テーブルとファンクション文字列を使用する。
- 外部的に、レプリケートデータベースの `rs_threads` システムテーブルを使用する。

外部コミット制御では、次のルールに従う必要があります。

- 異なるセッションが同じローで動作する場合、セッション 1 の **update** オペレーションによってセッション 2 の **select** オペレーションはブロックされます。
- 異なるセッションが異なるローで動作する場合、セッション 1 の **update** オペレーションによってセッション 2 の **update** はブロックされません。

内部コミット制御メソッドは、依存する条件が少ないため、外部コミット制御よりも優れています。デッドロックが発生した場合、内部コミット制御では Replication Server が単一のトランザクションをロールバックできますが、外部コミット制御ではすべてのトランザクションをロールバックします。

Replication Server で、並列処理が最大限に実行され、トランザクション間の競合が最小限に抑えられるようにする方法は、他にもあります。たとえば、トランザクションの逐次化メソッドを使用すると、システムが競合を起こすことなく処理できる並列度を選択できます。

並列 DSI スレッドを使用する方法の詳細については、『Replication Server 管理ガイド 第 2 巻』の「パフォーマンスチューニング」を参照してください。

### 外部および内部のコミット制御

Replication Server では、レプリケートデータベースが Microsoft SQL Server の場合、ローレベルのロックを使用して `rs_threads` を作成します。

デフォルトで、ローレベルのロックは“on”で、ページレベルのロックも“on”です。外部コミット制御メソッド必要なのは、ローレベルのロックのみです。ロー

レベルのロックをテーブルに適用する場合、ユニークなインデックスまたはプライマリキーをテーブルに付与する必要があります。例：

```
create table rs_threads
(id int,seq int CONSTRAINT PK_rs_threads PRIMARY KEY CLUSTERED(id
ASC)
WITH (ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = OFF))
```

独立性レベルが3の場合、以下を使用します。

```
Select seq from rs_threads with(nolock) where id =?
```

トランザクションの独立性レベルの選択の詳細については、『[Replication Server 管理ガイド 第2巻](#)』の「パフォーマンスチューニング」を参照してください。

Replication Server は、**rs\_dsi\_check\_thread\_lock** ファンクションを使用して、現在の DSI エグゼキュータスレッドが別のレプリケートデータベースのプロセスをブロックしているかどうかをチェックします。例：

```
select count(*) 'seq' from master..sysprocesses where blocked =
@@spid
```

### トランザクションの逐次化メソッド

Replication Server には、並列化のレベルを指定するための3つの逐次化メソッドが用意されています。複写環境、そして並列スレッド間で予想される競合の度合いに応じて、逐次化メソッドを選択してください。

各逐次化メソッドでは、トランザクションが直前のトランザクションのコミットを待機しなければならなくなる前に、そのトランザクションをどの程度の量だけ開始できるようにするかを定義します。

逐次化メソッドは、次のとおりです。

- **no\_wait**
- **wait\_for\_start**
- **wait\_for\_commit**
- **wait\_after\_commit**

**alter connection** コマンドと **dsi\_serialization\_method** パラメータを使用して、データベースコネクションに逐次化メソッドを選択します。たとえば、次のコマンドを入力して、**wait\_for\_commit** 逐次化メソッドをコネクションに選択します。このコネクションは pubs2 データベースへのもので、これは SYDNEY\_DS データサーバにあります。

```
alter connection to SYDNEY_DS.pubs2
set dsi_serialization_method to 'wait_for_commit'
```

トランザクションは次の3つの部分で構成されています。

- 先頭部分。

- トランザクション本体。insert、update、delete などのオペレーションで構成されます。
- トランザクションの末尾部分。コミットまたはロールバックで構成されます。

逐次化メソッドは、コミットの一貫性を保ちながら、トランザクションの先頭部分が直前のトランザクションのコミット準備の完了を待機するかどうか、またはトランザクションの先頭部分がそれよりも前に処理されるようにするかどうかを定義します。

### no\_wait

**no\_wait** メソッドは、直前のトランザクションのコミットを待機せずに次のトランザクションを開始するよう DSI に指示します。

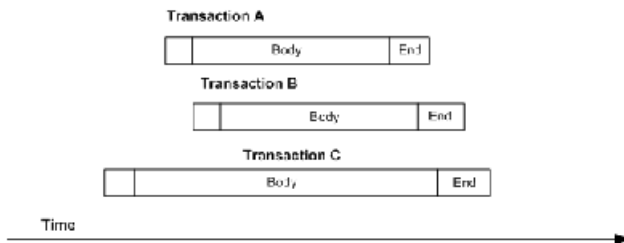
**no\_wait** を使用する場合、使用するプライマリアプリケーションが更新の競合を回避するよう設計されているか、**dsi\_partitioning\_rule** を効果的に使用して競合を減少させたり、取り除いたりしていることが前提となります。Adaptive Server は、**dsi\_isolation\_level** が 3 に設定されなければ、更新ロックを保持しません。このメソッドは、並列トランザクション間の競合がほとんどないことを前提とし、結果的には「wait\_for\_commit 逐次化メソッドによるスレッドのタイミング」図に示すように並列に近い形で実行されます。

---

**注意：** **dsi\_commit\_control** を “on” に設定している場合は、**dsi\_serialization\_method** は **no\_wait** にのみ設定できます。

---

図 7：no\_wait 逐次化メソッドによるスレッドのタイミング



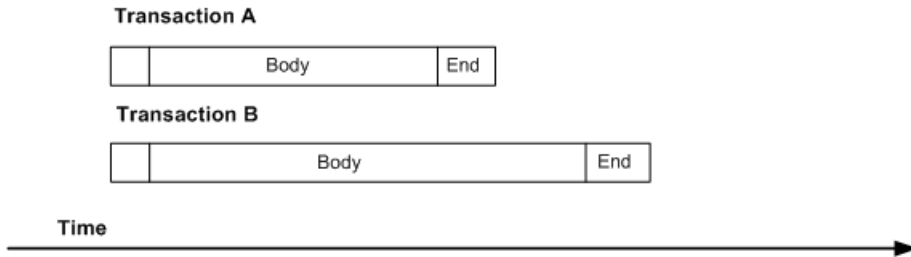
**no\_wait** を指定すると、パフォーマンスが向上する可能性が高くなりますが、競合が発生する危険性も高くなります。

### wait\_for\_start

**wait\_for\_start** は、開始直前にコミットするようにスケジュールされているトランザクションが開始した直後に、トランザクションを開始できることを指定します。

**dsi\_serialization\_method** を **wait\_for\_start** に設定し、同時に **dsi\_commit\_control** を **off** に設定することがないようにしてください。

図 8 : `wait_for_start` 逐次化メソッドによるスレッドのタイミング

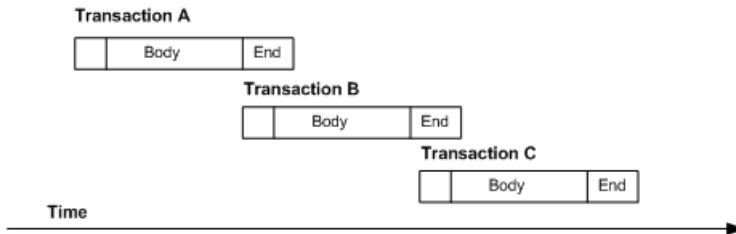


**wait\_for\_commit**

`wait_for_commit` メソッドでは、直前のトランザクションの処理が正常に完了してコミットの送信が開始されるまで、次のスレッドのトランザクショングループが処理のために送信されることはありません。

デフォルトの設定です。このメソッドは、並列トランザクション間にかかなりの競合があることを前提とし、図に示すように実行にずれが生じます。

図 9 : `wait_for_commit` 逐次化メソッドによるスレッドのタイミング

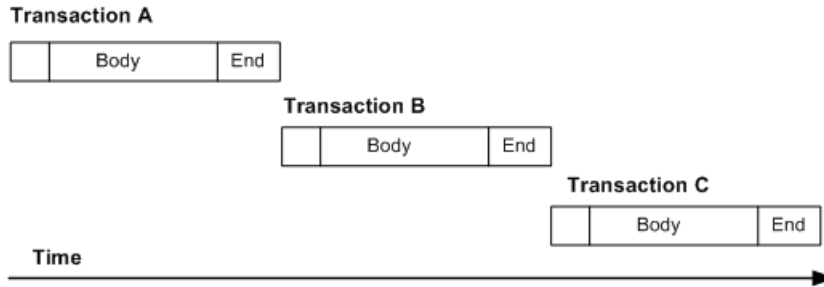


このメソッドでは、1つのトランザクションのコミットの準備ができるまで待機してから次のトランザクションを開始するように DSI に指示し、トランザクションの逐次化を維持します。最初のトランザクションは必要なロックをすでに保持しているため、最初のトランザクションのコミット中に次のトランザクションをレプリケートデータサーバに送信できます。

**wait\_after\_commit**

**wait\_after\_commit** は、直前にコミットするようにスケジュールされているトランザクションのコミットが完了するまで、トランザクションを開始できないよう指定します。

図 10 : **wait\_after\_commit** 逐次化メソッドによるスレッドのタイミング





# レプリケートデータサーバとしての Oracle

SAP 複写システムにおける、Oracle データサーバに固有のレプリケートデータベースの問題と考慮事項について説明します。

## Oracle レプリケートデータサーバ

---

ExpressConnect for Oracle を使用して Oracle データサーバに複写します。

ExpressConnect for Oracle は、Replication Server とレプリケートデータサーバ間の直接通信を可能にします。ExpressConnect for Oracle では、ゲートウェイサーバのインストールと設定の必要がないため、異機種間複写環境での Oracle データのアクセシビリティが向上します。

## Oracle での複写の干渉と影響

---

Oracle レプリケートデータベースへの唯一の重大な干渉または影響は、Replication Server のオペレーションをサポートするために、レプリケートデータベースに3つのテーブルを作成する接続プロファイルによって作成されるデータベースオブジェクトです。

作成されるテーブルの内容は次のとおりです。

- レプリケートデータベースによって使用されるソート順と文字セットに関する情報を格納する RS\_INFO。Replication Server バージョン 12.0 以降を使用する場合、レプリケートデータベースのソート順と文字セットは、RS\_INFO テーブルに記録されなければなりません。

**注意：**このテーブルに対する **INSERT** 文で、使用している Oracle データサーバに、適切な文字セットとソート順が指定されていることを確認してください。

Replication Server の **rs\_get\_charset** および **rs\_get\_sortorder** ファンクションは、レプリケートデータベース内の RS\_INFO テーブルから文字セットとソート順を取得します。

- レプリケートデータベースに適用する複写トランザクションに関する情報を格納する RS\_LASTCOMMITRS\_LASTCOMMIT テーブルの各ローは、プライマリデータベースからレプリケートデータベースに配信されコミットされた、最後のトランザクションを示します。Replication Server は、この情報によってすべてのトランザクションが配信されたことを確認します。

Replication Server の **rs\_get\_lastcommit** ファンクションは、レプリケートデータベース内の最後にコミットされたトランザクションに関する情報を取得します。ASE 以外のレプリケートデータベースについては、**rs\_get\_lastcommit** 関数は、データベース固有のファンクション文字列クラスで、レプリケートデータベース内の **RS\_LASTCOMMIT** テーブルへのアクセスに必要なクエリによって置き換えられます。

- **RS\_TICKET\_HISTORY** は、Replication Server コマンド **rs\_ticket** の実行結果を格納します。**rs\_ticket** コマンドは、コマンドがプライマリデータベースからレプリケートデータベースまで移動するために要する時間を測定するために、プライマリデータベースに対して発行できます。この情報を使用して、Replication Server のパフォーマンス、モジュールのハートビート、複製の正常性、テーブルレベルのクワイズをモニタできます。**rs\_ticket** の各実行結果は、レプリケートデータベース内にある **RS\_TICKET\_HISTORY** テーブルの 1 つのローに保存されます。**RS\_TICKET\_HISTORY** テーブルの各ローを問い合わせると、個々の **rs\_ticket** 実行結果を取得したり、別のローの結果と比較できます。データは、手動でトランケートできます。

---

**注意：** **RS\_TICKET\_HISTORY** テーブルは、Replication Server バージョン 15.1 以降でのみ使用できます。

---

## Oracle レプリケートデータベースのパーミッション

---

レプリケートデータベースにトランザクションを適用するために、Replication Server では Replication Server の **create connection** コマンドで指定するメンテナンスユーザ ID が必要となります。

メンテナンスユーザ ID は Oracle データサーバで定義され、レプリケートデータベースにトランザクションを適用する権限が付与されている必要があります。メンテナンスユーザ ID には、Oracle レプリケートデータベースにおける、次のパーミッションが必要です。

- Replication Server の処理に使用するテーブルを作成するための **CREATE TABLE** 権限。
- すべてのレプリケートテーブルに対する **UPDATE** 権限。
- すべてのレプリケートストアードプロシージャに対する **EXECUTE** 権限。



## Oracle に関するレプリケートデータベースの接続ティビティ

---

Replication Server は ExpressConnect for Oracle (ECO) を使用して Oracle レプリケートデータベースに接続します。

### *ExpressConnect for Oracle の使用方法*

Replication Server のデータベースコネクション名は、データサーバ名 (**server\_name**) およびデータベース名 (**db\_name**) の 2 つの部分で構成されています。**server\_name** は、`tnsnames.ora` ファイル内のサービス (Oracle インスタンス) の名前です。**db\_name** は、Oracle データベース (Oracle SID) のインストールおよび設定時にこのデータベースに与えられた名前です。通常、これはデフォルトで “ORCL” になっています。

ExpressConnect for Oracle は、`tnsnames.ora` ファイル内で、Replication Server データベースコネクションで指定された **server\_name** と一致するエントリを検索します。レプリケート Replication Server は、データベースコネクションで指定された **user\_name** と **password** を使用して、レプリケートデータサーバにログインします。ExpressConnect for Oracle を使用して複写する場合、Oracle データサーバに必要な `interfaces` ファイルエントリはありません。

### *Replication Server がストアードプロシージャを複写する方法の指定*

ExpressConnect for Oracle (ECO) を使用している場合は、**dsi\_proc\_as\_rpc** を on に設定します。ECO では、リモートプロシージャコール (RPC) を使用したストアードプロシージャの複写のみがサポートされています。Replication Server から Oracle データベースへのコネクションを作成するときに Oracle ECO 接続プロファイルのいずれかを使用する場合、Replication Server はデフォルトで **dsi\_proc\_as\_rpc** を on に設定します。Replication Server Options 15.5 の『Installation and Configuration Guide ExpressConnect for Oracle 15.5』の「Configuring ExpressConnect for Oracle」を参照してください。

## Oracle レプリケートデータベースの設定

---

Replication Server の異機種データ型サポート (HDS) 機能には、レプリケート Replication Server と Oracle レプリケートデータベースで HDS 機能を設定するために使用できる設定情報が用意されています。

この設定情報は、インストールにおいて、接続プロファイルの一部として提供されます。

## レプリケートデータサーバとしての Oracle

- Replication Server インストール：
  - ファンクション文字列、エラークラス、ユーザ定義データ型の作成
- 接続プロファイル：
  - RSSD へのクラスレベルデータ型変換の適用
  - Oracle レプリケートデータベースでのオブジェクトの作成
  - 接続プロパティの設定  
ExpressConnect for Oracle を使用して接続することができます。  
ExpressConnect for Oracle を使用する場合、接続プロファイルのバージョンまたはオプション名は“eco”でなければなりません。
- 追加設定
  - コマンドのバッチ処理の設定
  - 動的 SQL の設定

### 参照：

- RSSD へのクラスレベルデータ型変換 (138 ページ)
- Oracle レプリケートデータベースでのオブジェクトおよび接続プロパティ (138 ページ)
- コマンドのバッチ処理の設定 (140 ページ)
- 動的 SQL の設定 (142 ページ)

## Replication Server インストール

Replication Server インストールでは、複写をサポートするために必要なファンクション文字列とクラスが自動的にインストールされます。

### ファンクション文字列、エラークラス、ユーザ定義データ型

ファンクション文字列は、Replication Server のデフォルトの `rs_oracle_function_class` に追加されます。

ファンクション文字列は、いくつかのデフォルト Replication Server ファンクション文字列を、Oracle データサーバと通信し、テーブルとプロシージャにアクセスするように設計されたカスタムファンクション文字列に置き換えます。

---

**警告！** ExpressConnect for Oracle は、text 型と image 型を処理するためのカスタムファンクション文字列の使用をサポートしていません。

---

## 接続プロファイル

接続プロファイルを使用すると、プライマリデータベースとレプリケートデータベースの複写要件を満たす、事前に定義されたプロパティのセットで接続を設定できます。

### 構文

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]
[display_only]
```

### パラメータ

*data\_server* - 複写システムに追加するデータベースを持つデータサーバです。

*database* - 複写システムに追加するデータベースです。

*connection\_profile* - コネクションの設定、RSSD の修正、およびレプリケートデータベースオブジェクトの作成に使用する接続プロファイルを示します。

*version* - 使用する接続プロファイルのバージョンを指定します。

*user* - データベースの Replication Server メンテナンスユーザのログイン名です。Replication Server は、複写データを管理するのにこのログイン名を使用します。ネットワークベースセキュリティを有効に設定していない場合には、ユーザ名を指定する必要があります。

*other\_create\_connection\_options* - プロファイルで指定されない接続オプションの設定 (パスワードの設定など)、またはプロファイルで指定されているオプションの上書き (Replication Server に用意されているファンクション文字列クラスを上書きするカスタムファンクション文字列クラスの指定など) を行うには、他の **create connection** オプションを使用します。**create connection** コマンドの他のオプションのリストについては、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**create connection**」を参照してください。

**display\_only** - **display\_only** は、**using profile** 句とともに使用し、実行されるコマンド、およびそのコマンドを実行するサーバの名前を表示します。**display\_only** を使用した結果については、クライアントログおよび Replication Server ログを参照してください。

### RSSD へのクラスレベルデータ型変換

クラスレベル変換によって、プライマリデータ型とデータの変換先のレプリケートデータ型が特定されます (たとえば、DB2 UDB の `TIMESTAMP` は Oracle の `DATE` に変換されます)。

クラスレベル変換は、以下に示す適切な名前の接続プロファイルによって Oracle レプリケートデータベースに提供されます。

- `rs_ase_to_oracle` - Adaptive Server から Oracle へのクラスレベル変換をインストールします。
- `rs_db2_to_oracle` - DB2 UDB for z/OS データ型を Oracle データ型に変換します。
- `rs_udb_to_oracle` - DB2 UDB (for UNIX および Windows) データ型を Oracle データ型に変換します。
- `rs_msss_to_oracle` - Microsoft SQL Server データ型を Oracle データ型に変換します。

使用可能なすべてのプロファイルを表示するには、**admin show\_connection\_profiles** コマンドを使用します。

Oracle 複写環境に対して Adaptive Server Enterprise (ASE) の ExpressConnect for Oracle バージョンのプロファイルを使用したスクリプトの例：

```
create connection to oracleSID_name.oracleSID_name
using profile rs_ase_to_oracle;eco
set username rs_maint_user
set password rs_maint_user_pwd
go
```

### Oracle レプリケートデータベースでのオブジェクトおよび接続プロパティ

接続プロファイルによって、`RS_INFO`、`RS_LASTCOMMIT`、および `RS_TICKET_HISTORY` の各テーブルがレプリケートデータベース内に作成されるとともに、`RS_TRIGGERS_CONTROL` パッケージが作成されます。

接続プロファイルによって次の接続プロパティが設定されます。

```
set error class rs_oracle_error_class
set function string rs_oracle_function_class
```

## 追加設定

複写をサポートするための追加設定について説明します。

設定内容は次のとおりです。

- ExpressConnect の設定
- コマンドのバッチ処理の設定

- トリガの起動の設定
- Oracle Flashback の設定
- 動的 SQL の設定

### **ExpressConnect の設定**

Replication Server には、Oracle レプリケーションコネクションの適切なデータベース固有の動作 (データ型変換、コミット処理、**rs\_ticket** サポートなど) に必要な設定およびファンクション文字列について Replication Server コネクションに指示する Oracle 接続プロファイルが用意されています。

Oracle に対する Replication Server コネクションを作成または変更する場合、該当する Oracle 接続プロファイル (ASE から Oracle への複写のプロファイルや Oracle から Oracle への複写のプロファイルなど) を使用します。

また、Oracle でストアドプロシージャを複写する場合、追加の顧客指定のファンクション文字列が必要になる場合があります。デフォルトで、Replication Server によって ASE 構文が生成されますが、ターゲットデータベースで認識されない場合があります。この場合、この構文がターゲットデータベースで認識できるように、ファンクション文字列を追加できます。たとえば、ある文字の型とある通貨の型のパラメータを使用して **econn\_test\_basic\_proc** ファンクション呼び出しを変換する場合、次のようにファンクション文字列を作成する必要があります。

```
create function string econn_test_basic_proc.econn_test_basic_proc
for
rs_oracle_function_class with overwrite output language
'call econn_test_basic_proc(?charcolp!param?, ?moneycolp!param?)'
```

この例では、ファンクション文字列によって、キーワード **call** がすべての複写定義および **rs\_oracle\_function\_class** の **econn\_test\_basic\_proc** というファンクションの前に配置されます。Oracle で使用可能な構文を生成する別のファンクション文字列の例は次のとおりです。

```
create function string econn_test_basic_proc.econn_test_basic_proc
for
rs_oracle_function_class with overwrite output language 'begin
econn_test_basic_proc(?charcolp!param?, ?moneycolp!param?);; end;'
```

この例では、ファンクション文字列によって、同じファンクション複写定義とファンクションにキーワード **begin** が付加され、文字列 “;; end;” が追加されます。

---

**警告！** ExpressConnect for Oracle は、text 型と image 型を処理するためのカスタムファンクション文字列の使用をサポートしていません。

---

### *ExpressConnect for Oracle の配列処理*

ExpressConnect for Oracle の配列処理が、テーブルレベルの複写定義が設定されたテーブルにのみ適用されていることを確認してください。ExpressConnect for

Oracle コネクション内の配列処理を通してパフォーマンスを向上させるためには、テーブルレベルの複写定義に含まれている情報が必要です。

### コマンドのバッチ処理の設定

コマンドのバッチ処理を使用すると、Replication Server は複数のコマンドを1つのコマンドバッチとしてデータサーバに送信できます。

セミコロン (;) で区切ることによって、言語ファンクション文字列の出力テンプレートに複数のコマンドを入力できます。データベースがコマンドバッチを使用できるように設定されている場合 (デフォルトの設定) は、Replication Server は、このセミコロンをそのコネクションの DSI コマンドセパレータ文字に置き換えてから、単一のバッチ内のファンクション文字列としてデータサーバに送信します。

セパレータ文字は、**dsi\_cmd\_separator** オプション (**alter connection** コマンド) に定義されています。データベースへのコネクションがバッチを使用できるように設定されていない場合は、Replication Server は、ファンクション文字列内のコマンドを一度に1つずつデータサーバに送信します。データベースのバッチを有効または無効にするには、**alter connection** コマンドを使用します。

コマンドのバッチ処理を使用するには、次のように入力します。

```
batch = on  
batch_begin = off
```

set batch が “on” の場合、以下の設定も指定する必要があります。

```
dsi_cmd_separator set = ;
```

**rs\_begin** ファンクション文字列で使用されるプレースホルダコマンドにより、**batch\_begin** を “on” に設定すると、DSI を適切に起動できない場合があります。**batch\_begin** を “off” に設定すると、**rs\_begin** および **rs\_commit** コマンドを、コマンドのバッチとは別に送信できるようになるため、転送されたすべてのコマンドで正しい SQL が保証されます。

```
use_batch_markers = on
```

Oracle では、コマンドのバッチに BEGIN マーカと END マーカが必要です。**use\_batch\_markers** を “on” に設定すると、ファンクション文字列 **rs\_batch\_start** および **rs\_batch\_end** から、これらのマーカが自動的に追加されます。『Replication Server 管理ガイド 第2巻』の「ASE 以外のサーバのコマンドのバッチ処理」を参照してください。

### トリガの起動の設定

Replication Server では、セッションレベルまたはコネクションレベルで Oracle のトリガの実行を無効にすることができます。

トリガの起動は、Replication Server で PL/SQL コマンドがレプリケートデータベースに対して実行されるたびに制御することができます。レプリケートデータベースでのトリガ実行の制御によって、レプリケートデータベース側でのトリガ制御が存在しなかったために発生したデータの重複とデータの誤りのエラーを排除できます。

レプリケートデータベースで制御されるすべてのトリガについては、トリガアクションの開始時にトリガを作成し直し、トリガ制御文を追加します。

### トリガの起動の制御

**RS\_TRIGGER\_CONTROL** パッケージを介してトリガの起動を制御します。このパッケージは、コネクションプロファイルを通してレプリケート Oracle データベースへのコネクションが作成される際に自動的にインストールされます。

1. コネクションパラメータ **dsi\_keep\_triggers** をオフに設定し、Replication Server がレプリケートデータベースへの接続時に、**RS\_TRIGGERS\_CONTROL** 有効フラグを設定するようにします。
2. トリガ制御 PL/SQL コードをトリガアクションの最初の行に追加します。

```
if RS_TRIGGER_CONTROL.IS_ENABLED then      return;end if;
```

これは、トリガが Replication Server によって起動されることを示し、トリガによる実際のアプリケーションロジックの実行を防ぎます。

『Replication Server リファレンスマニュアル』を参照してください。

### Oracle Flashback の設定

Replication Agent では、テーブルレベルおよびトランザクションレベルでの Oracle Flashback がサポートされています。

データベースがオンライン状態のとき、履歴データの問い合わせ、変更分析の実行、および自己サービスによる修復によって、論理的な破損からリカバリするには、Oracle Flashback を使用します。Oracle のユーザは、このフラッシュバックを使用して、以前のデータの変更を元に戻し、貴重なデータの誤削除、誤ったデータの削除、誤ったテーブルの削除などのオペレータまたはユーザによるアプリケーション障害を最小限に抑えることができます。

Replication Agent では、次の 2 種類のフラッシュバックをサポートしています。

- 削除されたテーブルのフラッシュバック。これによって、Oracle を対象とする **drop table**、**flashback table to before drop**、および **purge recyclebin** などのフラッシュバック DDL コマンドが複写されます。 **purge dba\_recyclebin** を複写するに

## レプリケートデータサーバとしての Oracle

は、DCO 15.0 ESD#3 以降を使用し、**sysdba** 権限を DDL ユーザに割り当てます。

- 特定のタイムスタンプまたは SCN へのテーブルのフラッシュバック。これによって、対象の Oracle データベースに DML の変更が複写されます。

特定のタイムスタンプまたは SCN にテーブルをフラッシュバックするには、次の手順を実行します。

- **pdb\_setreptable** コマンドを使用して、特定の状態にフラッシュバックする必要があるテーブルにマークを付けます。

フラッシュバック DDL 文を複写するには、次の手順を実行します。

- プライマリデータベースとレプリケートデータベースの両方の Recycle Bin を有効にします。

```
alter system set recyclebin=on
```

- ECDA を使用する場合、ECDA Option for Oracle の **rep\_sparse\_parse** パラメータを 1 に設定します。このパラメータのデフォルト値は、ECDA Option for Oracle 15.0 ESD #3 を使用している場合、0 です。
- **pdb\_setrepddl enable** コマンドを使用して、DDL 複写を有効にします。

### 動的 SQL の設定

Oracle での動的 SQL の使用は、ExpressConnect を使用する Replication Server でサポートされます。

## Oracle レプリケートデータベースの並列 DSI スレッド

異機間複写環境では、並列 DSI を使用すると、レプリケートデータベースのコミット順はプライマリデータベースと必ず同じになります。

デッドロックが発生した場合、DSI によってデッドロック競合が解決され、Replication Server ではトランザクションをロールバックして再度実行します。

Replication Server は、トランザクションのコミット順序を維持し、次のいずれかの方法で、同時に並列して実行されているトランザクションでの更新の競合を検出できます。

- Replication Server の内部テーブルとファンクション文字列
- レプリケートデータベースの **rs\_threads** システムテーブル

Replication Server で、並列処理が最大限に実行され、トランザクション間の競合が最小限に抑えられるようにする方法は、他にもあります。たとえば、トランザクションの逐次化メソッドを使用すると、システムが競合を起こすことなく処理できる並列度を選択できます。



並列 DSI スレッドを使用する方法の詳細については、『Replication Server 管理ガイド 第 2 巻』の「パフォーマンスチューニング」を参照してください。

## 外部および内部のコミット制御

Replication Server は、Oracle がレプリケートデータベースの場合、外部コミット制御をサポートしません。

Replication Server は、**rs\_dsi\_check\_thread\_lock** ファンクションを使用して、現在の DSI エグゼキュータスレッドが別のレプリケートデータベースのプロセスをブロックしているかどうかをチェックします。次に例を示します。

```
'select count(*) as seq from DBA_BLOCKERS
where holding_session in (select sid from v$session
where audsid = userenv('SESSIONID'))';'
```

## トランザクションの逐次化メソッド

Replication Server には、並列化のレベルを指定するための 4 つの逐次化メソッドが用意されています。複製環境、そして並列スレッド間で予想される競合の度合いに応じて、逐次化メソッドを選択してください。

各逐次化メソッドでは、トランザクションが直前のトランザクションのコミットを待機しなければならなくなる前に、そのトランザクションをどの程度の量だけ開始できるようにするかを定義します。

逐次化メソッドによって割り当てられた並列度を下げずに競合の確率を下げるには、**dsi\_partitioning\_rule** パラメータを使用します。

Oracle の場合は、**wait\_after\_commit** のみを使用する必要があります。

**alter connection** コマンドと **dsi\_serialization\_method** パラメータを使用して、データベース接続に逐次化メソッドを選択します。たとえば、次のコマンドを入力して、**wait\_after\_commit** 逐次化メソッドを接続に選択します。この接続は pubs2 データベースへのもので、これは SYDNEY\_DS データサーバにあります。

```
alter connection to SYDNEY_DS.pubs2
set dsi_serialization_method to 'wait_after_commit'
```

トランザクションは次の 3 つの部分で構成されています。

- 先頭部分。
- トランザクション本体。**insert**、**update**、**delete** などのオペレーションで構成されます。
- トランザクションの末尾部分。コミットまたはロールバックで構成されます。

逐次化メソッドは、コミットの一貫性を保ちながら、トランザクションの先頭部分が直前のトランザクションのコミット準備の完了を待機するかどうか、または

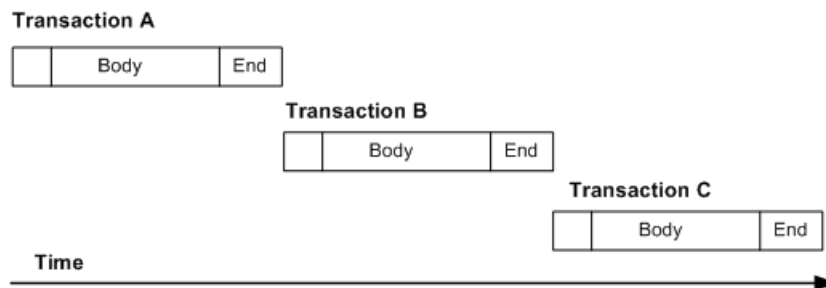
トランザクションの先頭部分がそれよりも前に処理されるようにするかどうかを定義します。

### **wait\_after\_commit**

**wait\_after\_commit** は、直前にコミットするようにスケジュールされているトランザクションのコミットが完了するまで、トランザクションを開始できないよう指定します。

Oracle など、多版型同時実行制御 (MVCC : Multiversion Concurrency Control) またはオプティミスティック同時制御 (OCC : Optimistic Concurrency Control) を使用するデータベースでは、**wait\_after\_commit** 逐次化メソッドを使用することをおすすめします。その他のデータベースでは、デフォルトのメソッドとして **wait\_for\_commit** を使用できます。

図 11 : **wait\_after\_commit** 逐次化メソッドによるスレッドのタイミング



# レプリケートデータサーバとしての SAP IQ

SAP 複写システムにおける、SAP IQ データサーバに固有のレプリケートデータベースの問題と考慮事項および SAP IQ への接続方法と複写の設定方法について説明します。

SAP IQ は、レポートおよびデータ分析に理想的なプラットフォームです。ただし、より効果的にレポートするには、リアルタイムデータが必要です。

Replication Server には、SAP IQ に直接複写するための Real-Time Loading (RTL) ソリューションが含まれており、ログされた変更をプライマリデータベースのログ順に従ってレプリケートデータベースに送信する連続複写モードの代わりに使用できます。

コンパイルとバルク適用を実行して SAP IQ レプリケートデータベースに直接複写する場合、RTL は連続複写モードよりも優れたパフォーマンスを発揮します。

## Real-Time Loading ソリューション

---

RTL はできるだけ多くのコンパイル可能なトランザクションをグループ化して、グループ内のトランザクションをまとめて最終的な変更としてコンパイルしてから、レプリケートデータベースでバルクインタフェースを使用してその変更をレプリケートデータベースに適用します。

SAP IQ レプリケートデータベースに複写する際、RTL は次を使用します。

- コンパイル - 複写データを各テーブル別、各 **insert**、**update**、**delete** オペレーション別にし、それらのオペレーションを最終的なローのオペレーションにコンパイルして整理します。
- バルク適用 - コンパイルされた最終的な結果に対して最も効率の良いバルクインタフェースを使用して、最終的な結果をバルク適用します。Replication Server は、メモリ内の最終的な変更が保管されるデータベースを使って最終的なロー変更を保管し、それをレプリケートデータベースに適用します。

RTL では、以下によって、連続複写モードやステージングソリューションなどと比べて SAP IQ への複写パフォーマンスを向上できます。

- 外部コンポーネント数の減少 - ステージング用のデータベースを使用しないので、メンテナンスコストとオーバーヘッドが削減されます。
- 遅延時間の減少 - ステージングソリューションからのオーバーヘッドがなく、SAP IQ に直接複写されます。

## レプリケートデータサーバとしての SAP IQ

- 利便性の向上 - RTL の設定には、ファンクション文字列のマップ、DSI のサスペンドと再開、ステー징データベースから SAP IQ へのデータの移植、ステーjingソリューション用のアクティビティのスケジューリングのいずれも必要ありません。
- コンパイルとバルク適用 - コンパイル処理では、ログに記録されている個々のオペレーションを送信する代わりに、グループ化された一連のオペレーションから **insert**、**update**、または **delete** の中間オペレーションを削除し、複写トランザクションの最終的なコンパイル状態のみを送信します。トランザクションプロファイルによって異なりますが、これは通常 Replication Server が SAP IQ に送信して処理させるコマンド数を少なくします。  
SAP IQ には、SQL 言語モードと比べて **insert** オペレーションのパフォーマンスを向上させるバルクインタフェースが用意されています。RTL は SAP IQ のバルクインタフェースを活用して、**insert** だけでなく **update** と **delete** のオペレーションのパフォーマンスを向上させます。  
Replication Server が大量のトランザクションを組み合わせてコンパイルし 1 つのグループにまとめるので、バルクオペレーション処理が向上し、複写スループットとパフォーマンスも向上します。グループサイズを調整して、バルク適用のためにグループ化されるデータ量を制御できます。

### ライセンス

Real-Time Loading (RTL) を使用した SAP IQ への複写は、Real-Time Loading Edition 製品エディションで実行可能です。『Replication Server インストールガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

### データベースとプラットフォームのサポート

- SAP IQ - Real-Time Loading を使用して SAP IQ バージョン 12.7 ESD #3 以降に複写できます。SAP IQ のバージョンとプラットフォームのサポートに関する最新情報については、『Replication Server リリースノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。
- Adaptive Server - Replication Server は、Adaptive Server バージョン 15.0.3 またはバージョン 15.5 以降から SAP IQ への複写をサポートしています。
- Oracle - Replication Server は、Oracle 10g および 11g から SAP IQ への複写をサポートしています。Replication Server Options 15.5 の『Replication Agent 15.5 リリースノート』の「製品の概要」の「互換性のある製品」を参照してください。

### 64 ビットサポート

64 ビットのハードウェアプラットフォームを使用すると、最適なパフォーマンスを得ることができます。『Replication Server 新機能ガイド』の「Replication Server バージョン 15.5 の新機能」の「64 ビットコンピューティングプラットフォームのサポート」を参照してください。

## RTL のコンパイルとバルク適用

RTL のコンパイルでは、複写するデータがテーブルごとに **insert**、**update**、**delete** オペレーション別に整理してまとめられ、オペレーションが最終的なローのオペレーションにコンパイルされます。

RTL は複写定義内のプライマリキーによって異なったデータローを区別します。複写定義がない場合は、text と image のカラム以外はすべて、プライマリキーとみなされます。

レプリケートテーブルに複数のユニークキーがある場合は、ユニークインデックスで指定されているすべてのカラムがテーブル複写定義内のプライマリキーに含まれている必要があります。含まれていないと、複写で重複キーエラーが発生するおそれがあります。

通常の複写環境で見られるオペレーションの組み合わせでは、同一のプライマリキーを持つテーブルとローがあると、RTL は次のオペレーションの組み合わせルールに従います。

- **insert** の後に **delete** があると結果はオペレーションなしになります。
- **delete** の後に **insert** があると結果のオペレーションは減少しません。
- **update** の後に **delete** があると結果のオペレーションは **delete** になります。
- **insert** の後に **update** があると、2つのオペレーションは1つのオペレーションに集約され、結果のオペレーションは **insert** になります。結果のオペレーションの内容は、最初のオペレーション結果に次のオペレーションの相違点を上書きした結果となります。
- **update** の後にもう1つの **update** があると、2つのオペレーションは1つのオペレーションに集約され、結果のオペレーションは **update** になる。結果のオペレーションの内容は、最初のオペレーション結果に次のオペレーションの相違点を上書きした結果となる。

オペレーションのその他の組み合わせでは、コンパイルステータスが無効になります。

### 例 1

これはログ順のローごとの変更例です。この例では、T は **create table T(k int , c int)** コマンドによって以前に作成されたテーブルです。

```
1. insert T values (1, 10)
2. update T set c = 11 where k = 1
3. delete T where k = 1
4. insert T values (1, 12)
5. delete T where k =1
6. insert T values (1, 13)
```

RTL では、1 の **insert** と 2 の **update** を組み合わせて **insert T values (1, 11)** に変換できます。変換結果の **insert** と 3 の **delete** は、相殺されるので削除できます。

4の **insert** と5の **delete** は削除できます。コンパイルされた最終的なRTLオペレーションは、6の最後の **insert** になります。

```
insert T values (1, 13)
```

### 例2

ログ順のローごとの変更のもう1つの例です。

```
1. update T set c = 14 where k = 1
2. update T set c = 15 where k = 1
3. update T set c = 16 where k = 1
```

RTLでは、1と2の **update** をまとめると2の **update** になり、2と3の **updates** をまとめると3の **update** になるので、それが  $k=1$  での最終的なロー変更になります。

Replication Server は最終的な変更を保管するインメモリデータベース内の **insert**、**delete**、および **update** テーブルを使用して、レプリケートデータベースに適用する最終的なロー変更を保管します。最終的なロー変更がレプリケートテーブル別およびオペレーションの種類別 (**insert**、**update**、または **delete**) にソートされると、バルクインタフェースに渡す準備が整います。

RTLは **insert** オペレーションをレプリケートテーブルに直接ロードします。SAP IQは **update** と **delete** のバルクオペレーションをサポートしないので、RTLは **update** と **delete** オペレーションをテンポラリワークテーブルにロードします。テンポラリワークテーブルはRTLによってIQのテンポラリストア内に作成されます。次に、RTLは **join-update** または **join-delete** オペレーションをレプリケートテーブルに対して実行して、最終的な結果を生成します。ワークテーブルは動的に作成され削除されます。

例2では、次の処理によってコンパイル結果が `update T set c = 16 where k = 1` になります。

1. RTLは `#rs_uT(k int, c int)` ワークテーブルを作成します。
2. RTLがワークテーブルに対して **insert** を実行します。

```
insert into #rs_uT(k, c) location 'idemo.db' {select * from rs_uT}
```

3. RTLが次の **join-update** を実行します。

```
update T set T.c=#rs_uT.c from T,#rs_uT where T.k=#rs_uT.k
```

RTLが大量のトランザクションを1つのグループにまとめるので、バルクオペレーション処理が向上し、複製スループットとパフォーマンスも向上します。RTLサイズを設定パラメータで調整することによって、RTLがバルク適用のためにグループにまとめるデータの量を制御できます。

RTLはロー変更を変更がログされた順序で適用しませんが、データロスはありません。

- 異なったデータローでは、ロー変更が適用される順序は結果に影響しません。
- 同じロー内では、コンパイル後、**delete** の後に **insert** を適用することによって整合性を維持します。

## 最終的な変更のデータベース

Replication Server には、最終的な変更を保管するデータベースがあります。これは、インメモリレポジトリとして機能し、トランザクションの最終的なロー変更、つまりコンパイルしたトランザクションを保管します。

各トランザクションに対して1つの最終的な変更のデータベースインスタンスがあります。最終的な変更を保管するデータベース内の各複写テーブルには最高3つの追跡テーブルがあります。最終的な変更を保管するデータベースとその中のテーブルを点検することによって RTL 複写のモニタと問題のトラブルシューティングを行うことができます。

### 参照：

- 最終的な変更のデータベースのサイズ (164 ページ)

### 最終的な変更のデータベースをモニタする

最終的な変更のデータベースインスタンスにアクセスしてモニタします。

**sysadmin cdb** コマンドを使って最終的な変更のデータベースをモニタします。

『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**sysadmin cdb**」を参照してください。

## RTL の処理と制限事項

RTL は元のコミット順を維持しながら、トランザクションの最終的なロー変更のみを適用します。それによって中間的なロー変更は省略されますが、トランザクションの整合性は保証されます。

このアプローチには次のような問題が伴います。

- **Insert** トリガが起動されません。これは、RTL プロセスが最終的な新しいローのバルクロードをテーブルに対して直接行うからです。Replication Server がコンパイルの最終結果をレプリケートデータベースに適用すると、**update** と **delete** の各トリガは引き続き起動します。ただし、Replication Server がコンパイルし最終結果には含まれないロー変更が、それらのトリガから確認できなくなります。トリガが検出できるのは、最後のローイメージのみです。Replication Server を使用して、ユーザが変更したテーブルの任意のカラムにそのユーザを関連付けるトリガロジックのあるテーブルスキーマで、`last_update_user` カラムを使用してユーザ更新を監査するとします。userA がテーブルの `colA` と `colC` を変更した後に、userB が `colB` と `colD` を変更し

た場合、トリガが起動すると、トリガロジックが検出できるのはテーブルを最後に変更したユーザのみです。したがって、トリガロジックはこれら4つのカラムを変更したユーザとして userB を関連付けます。ロー変更を個別に検出する必要がある同様のロジックを含むトリガを定義する場合、そのテーブルの RTL コンパイルを無効にする必要がある場合があります。

- RTL はロー変更を変更がログに記録された順序と同じ順序では適用しません。複写テーブルにログ順に変更を適用するには、そのテーブルに対して RTL コンパイルを無効にします。
- 複写テーブルに参照制約がある場合、その参照制約を複写定義の中で指定してください。制約エラーを避けるために、RTL は複写定義に従ってテーブルをロードします。
- RTL では、**wait\_for\_commit** メソッド以外の並列 DSI 逐次化メソッドをサポートしません。
- RTL では、カスタムファンクション文字列をサポートしていないため、コンパイルできないコマンドとして扱います。
- 以下を検出すると、Replication Server は、ログ順にローごとに行う連続複写に戻ります。
  - コンパイルできないコマンド - ストアドプロシージャ、SQL 文、システムトランザクション、Replication Server の内部マーカ、および **rs\_ticket**。
  - コンパイルできないトランザクション - コンパイルできないコマンドを含んでいるトランザクション。
  - コンパイルできないテーブル - RTL が無効にされているテーブル、カスタムファンクション文字列を持つテーブル、RTL がコンパイルできないテーブルと参照制約関係があるテーブル。
  - 実行時にコンパイルできないテーブル - これは、対象のテーブルの複写定義に **replicate minimal columns** 句を使用する場合やトランザクションによってプライマリキー値が変更された場合など、最小限にバックされた更新がトランザクションに含まれている場合に発生します。
- Replication Server は、SAPIQ ビューの RTL をサポートしていません。ビューの **dsi\_compile\_enable** を off に設定して、ビューをコンパイル不可としてマーク付けします。
- テーブル複写定義がなく、プライマリキーが指定されていないテーブルの場合、Replication Server は、text カラムまたは image カラムを除き、すべてのカラムをプライマリキーとして処理するため、テーブルへの更新をプライマリキーの更新に変換します。
- RTL はトランザクションのグループ化を停止できるパラメータ (**dsi\_partition\_rule** など) を無視します。
- RTL 処理中にエラーが発生すると、Replication Server はコンパイルをリトライします。リトライでは、コンパイルが失敗したトランザクションを特定できる



まで、トランザクショングループを小さなグループに分割していきます。特定されたトランザクションは連続複写を使って適用されます。

- パフォーマンス上の利点を実現するには、プライマリデータベースとレプリケートデータベースを同期させ、エラー発生による Replication Server への余分な処理オーバーヘッドを避けるようにします。**dsi\_command\_convert** を **i2di,u2di** に設定して、データを同期できますが、これも処理オーバーヘッドを発生させます。データベースを同期する場合は、**dsi\_command\_convert** を **none** にリセットします。
- RTL はローカウムの検証を行って複写の整合性を確認します。ローカウムの検証はコンパイルに基づいて行われます。予期されるローカウムはコンパイル後のロー数です。
- 複写定義の中に **identity** データ型のカラムがある場合、Replication Server はレプリケートデータベース内で次の SAP IQ コマンドを実行します。
  - **set temporary option identity\_insert= 'table\_name'** (ID カラムの挿入および更新前)。
  - **set temporary option identity insert= ""** (ID カラムの挿入および更新後)。
- プライマリデータサーバ上のマイクロ秒から SAP IQ レプリケートデータサーバ上のミリ秒に変換するときに、Replication Server は、TIMESTAMP データ型の **time** カラム、**datetime** カラム、および **smalldatetime** カラム内のマイクロ秒の最後の 3 桁をゼロにするために複写が失敗します。『SAP IQ リファレンス: ビルディングブロック、テーブル、およびプロシージャ』の「他の Sybase データベースとの互換性」の「データ型」の「date、time、datetime、timestamp データ型」の「ASE の日時値と時刻値との互換性」を参照してください。

**SAP IQ INSERT ... LOCATION** 文を使用してレプリケートテーブルを手動でマテリアライズすると、SAP IQ は **TIMESTAMP** カラム内のマイクロ秒の最後の 3 桁を 000、333、または 666 という値で埋めます。次に例を示します。

```
insert test_datetime_iq4 location 'zeus.primaryDB4'
  { select c1,c2,c3,c4,c5 from test_datetime_iq4 }
```

c2 は **datetime**、c4 は **time** です。

**INSERT ... LOCATION** を使用してテーブルをマテリアライズした後も RTL が引き続き SAP IQ に複写するようにするには、次のように、TIMESTAMP データ型の明示的な変換を強制することによって SAP IQ テーブルを手動でマテリアライズします。

```
insert test_datetime_iq4 location 'zeus.primaryDB4'
  { select c1,convert(varchar,c2,109),
    c3,convert(varchar,c4,20),c5
    from test_datetime_iq4 }
```

- デフォルトでは、Oracle は最小限のロギングを行います。したがって、データベース複写定義を使用している場合、テーブル複写定義を作成するか、完全ロギングを有効にして、**update** コマンドが正しく動作するようにします。テーブ

## レプリケートデータサーバとしての SAP IQ

ル複写定義を作成することにした場合、Replication Agent または Replication Server で次のようにしてその定義を作成できます。

- Replication Agent for Oracle - 複写マークの付いたテーブルが存在する場合に Replication Server で複写定義を自動的に作成するには、テーブルに複写マークを付ける前に **pdb\_auto\_create\_repdefs** を **true** に設定するか、テーブルにマークを付けてから **rs\_create\_repdef** を実行します。詳細については、Replication Server Options の『Replication Agent リファレンスマニュアル』を参照してください。
- Replication Server - Replication Server で直接複写定義を作成するには、**create replication definition** に **send standby** 句を指定して実行します。『Replication Server リファレンスマニュアル』を参照してください。

### 参照：

- 参照制約のあるテーブル (171 ページ)
- RTL パフォーマンスのチューニング (159 ページ)

## SAP IQ レプリケートデータサーバ

---

レプリケート Replication Server は、SAP IQ レプリケートデータベースにログインして、複写トランザクションを適用することによって、レプリケート SAP IQ データサーバと直接データをやり取りします。

## SAP IQ での複写の干渉と影響

---

SAP IQ レプリケートデータベースに対する重大な干渉または影響は、接続プロファイルによって SAP IQ レプリケートデータベースに作成されるシステムテーブルと、RTL バルク適用を格納するために SAP IQ レプリケートデータベースに作成されるテンポラリテーブルのみです。

### システムテーブル

接続プロファイルによって3つのテーブルが SAP IQ レプリケートデータベースに作成されます。

- **rs\_threads** - Replication Serverはこのテーブルの情報を使ってデッドロックを検出し、並列 DSI スレッド間でトランザクションの逐次化を実行します。このテーブルのエントリは、トランザクションが開始されたときと、コネクションに対して2つ以上の DSI スレッドが定義されたときに更新されます。
- **rs\_lastcommit** - レプリケートデータベースに適用される複写トランザクションに関する情報を格納します。**rs\_lastcommit** テーブル内の各ローは、

プライマリデータベースからレプリケートデータベースに配信されコミットされた、最後のトランザクションを示します。Replication Server は、この情報によってすべてのトランザクションが配信されたことを確認します。

Replication Server の **rs\_get\_lastcommit** 関数は、レプリケートデータベース内の最後にコミットされたトランザクションに関する情報を取得します。ASE 以外のレプリケートデータベースについては、**rs\_get\_lastcommit** 関数は、データベース固有のファンクション文字列クラスで、レプリケートデータベース内の **rs\_lastcommit** テーブルへのアクセスに必要なクエリによって置き換えられます。

- **rs\_ticket\_history** - Replication Server コマンド **rs\_ticket** の実行結果を格納します。プライマリデータベースに対して **rs\_ticket** コマンドを発行し、コマンドがプライマリデータベースからレプリケートデータベースまで移動するために要する時間を測定できます。この情報を使用して、Replication Server のパフォーマンス、モジュールのハートビート、複製の正常性、テーブルレベルのクワイズをモニタできます。**rs\_ticket** の各実行結果は、レプリケートデータベース内にある **rs\_ticket\_history** テーブルの 1 つのローに保存されます。**rs\_ticket\_history** テーブルの各ローを問い合わせると、個々の **rs\_ticket** 実行結果を取得したり、別のローの結果と比較できます。必要に応じて、**rs\_ticket\_history** テーブル内のデータを手動でトランケートします。

#### ワークテーブル

RTL バルク適用をサポートするために、RTL によって SAP IQ データベースの IQ テンポラリストア内にテンポラリーワークテーブルが作成されます。このワークテーブルは動的に作成され削除されます。

テンポラリーテーブルに必要な SAP IQ 内の領域は SAP IQ に複製する予定のデータ量に依存します。テンポラリーワークテーブルのために SAP IQ のテンポラリーデータベースの領域を調整するには、SAP IQ の **alter dbspace** コマンドを使用します。詳細については、該当するバージョンの SAP IQ マニュアルを参照してください。

SAP IQ 15.0 以降の例:

```
ALTER DBSPACE dbspace-name ADD FILE FileHist3
  \History1\data/file3' SIZE 500MB
```

## SAP IQ に関するレプリケートデータベースのコネクティビティ

---

SAPIQ をレプリケートデータサーバとして使用する場合、データベースゲートウェイを使用する必要はありません。レプリケート Replication Server は SAP IQ レプリケートデータサーバに直接接続します。

Replication Server のデータベース接続名は、データサーバ名 (**server\_name**) とデータベース名 (**db\_name**) で構成されています。レプリケート Replication Server は、データベース接続で指定された SAP IQ データベースゲートウェイ **server\_name** を含むエントリを `interfaces` ファイルで検索します。

`dsedit` を使用して、SAP IQ レプリケートデータサーバが受信を行うホストとポートを指定するエントリを Replication Server の `interfaces` ファイルに作成します。`interfaces` ファイルエントリの名前は、Replication Server データベース接続の **server\_name** 部分と一致している必要があります。Replication Server を再起動すると、Replication Server の `interfaces` ファイル内の新規エントリが有効になります。『Replication Server 設定ガイド』>「`rs_init` による Replication Server の設定とデータベースの追加」>「新しい Replication Server の設定」>「`interfaces` ファイルの編集」を参照してください。

SAPIQ レプリケートサーバの `interfaces` ファイルにレプリケート Replication Server のエントリを作成すると、Replication Server が **INSERT ... LOCATION** 文を SAP IQ に送信したときに、SAPIQ が Replication Server に接続してデータを取得できるようになります。

SAPIQ 15.0 以降では、「`set option public.STRING_RTRUNCATION = "OFF"`」を入力すると、Replication Server が **INSERT ... LOCATION** を使用して SAP IQ にデータを挿入するときの「Right truncation of string data」エラーを回避できます。

Replication Server は、データベース接続で指定された **user\_name** と **password** を使用して、SAPIQ レプリケートデータサーバにログインします。SAPIQ レプリケートデータベースの場合、**user\_name** と **password** はメンテナンスユーザの ID とパスワードです。

## SAP IQ レプリケートデータベースのパーミッション

レプリケートデータベースにトランザクションを適用するために、Replication Server と Sybase IQ ではメンテナンスユーザ ID が必要となります。

複写を開始する前に、SAP IQ データサーバでメンテナンスユーザ ID を定義し、レプリケートデータベースにトランザクションを適用するための権限をその ID に付与する必要があります。メンテナンスユーザ ID には、SAP IQ レプリケートデータベースで次のパーミッションが必要です。

- **RESOURCE** 権限 (ワークテーブルとテンポラリインデックスの作成に必要)
- **EXECUTE** パーミッション (`sp_iqwho` ストアドプロシージャの実行に必要)
- **GRANT ALL** パーミッション (すべての複写テーブルに対して必要)
- **UPDATE** 権限 (すべての複写テーブルに対して必要) および **EXECUTE** 権限 (すべての複写ストアドプロシージャに対して必要)

### メンテナンスユーザ ID に権限を与える

単純なセットアップで開始する場合、または SAP IQ への複写をテストする場合は、DBA と RESOURCE の権限を与えます。

1. SAP IQ の `rssetup.sql` サンプルスクリプトを使用して、適切な権限を持つ SAP IQ メンテナンスユーザを作成します。

**警告！** メンテナンスユーザ ID が既に存在する場合、このスクリプトはパスワードをデフォルトパスワードにリセットします。

```
grant connect to dbmaint identified by dbmaint
grant DBA to dbmaint
grant membership in group rs_systabgroup to dbmaint

-- Create a user for REPSRV to extract -- materialization data,
etc.
-- Give sa user access to any replicated tables
-- Give sa user access to REPSRV schema
grant connect to sa identified by sysadmin
grant DBA to sa
grant membership in group rs_systabgroup to sa

-- Allow sa and dbmaint to reference replicated tables created by
DBA
grant group to DBA
grant membership in group DBA to dbmaint
grant membership in group DBA to sa
go
```

## レプリケートデータサーバとしての SAP IQ

このスクリプトは、SAP IQ インストールディレクトリ内の `scripts` ディレクトリにあります。たとえば、UNIX プラットフォームでは、次のディレクトリにあります。

- SAP IQ 15.0 より前のバージョン - `/$ASDIR/scripts`
- SAP IQ 15.0 以降のバージョン - `/$IQDIR15/scripts`

ディレクトリの位置については、『SAP IQ インストールおよび設定ガイド』を参照してください。

2. SAP IQ データベースに Transact-SQL® (IQ DBA 用) との互換性があることを確認します。

『SAP IQ リファレンス: 文とオプション』 > 「データベースオプション」 > 「Transact-SQL 互換性オプション」と『SAP IQ リファレンス: ビルディングブロック、テーブル、およびプロシージャ』 > 『他の Sybase データベースとの互換性』を参照してください。

3. 複製に関係するすべてのテーブルとストアードプロシージャに対する適切なパーミッションを与えます。

## SAP IQ レプリケートデータベースの設定

---

SAP IQ サーバの設定に関する問題について説明します。

### Replication Server インストール

---

Replication Server は必要な接続プロファイルを自動的にインストールすることによって、SAP IQ への複製をサポートするファンクション文字列とクラスを提供します。

#### 接続プロファイル

接続プロファイルを使用すると、ファンクション文字列クラスとエラークラスを設定し、ユーザ定義データ型 (UDD) と SAP IQ 変換をインストールして、SAP IQ レプリケートデータベース内に複製に必要なテーブルを作成することによって、事前に定義されたプロパティのセットでコネクションを設定できます。

`rs_ase_to_iq` や `rs_oracle_to_iq` などの接続プロファイルは、Replication Server のインストールパッケージの一部であり、Replication Server のインストール時に登録されます。次に接続プロファイルを説明します。

- ファンクション文字列、エラークラス、ユーザ定義データ型をカスタマイズします。ファンクション文字列は、いくつかのデフォルト Replication Server ファンクション文字列を、SAP IQ データサーバと通信し、テーブルとプロシージャにアクセスするように設計されたカスタムファンクション文字列に置き換えます。これらのファンクション文字列は、Replication Server のデフォルトの

`rs_iq_function_class` に追加されます。RTL はカスタムファンクション文字列をコンパイルできないコマンドとして扱います。

- クラスレベルデータ型変換をカスタマイズします。クラスレベル変換によって、プライマリデータ型とデータの変換先のレプリケートデータ型が特定されます。クラスレベル変換は、以下に示す適切な名前の接続プロファイルによって SAP IQ レプリケートデータベースに提供されます。
  - `rs_ase_to_iq` - Adaptive Server から SAP IQ へのクラスレベル変換をインストールします。
  - `rs_oracle_to_iq` - Oracle データ型を SAP IQ データ型に変換します。
- 使用可能なすべてのプロファイルを表示するには、`admin show_connection_profiles` コマンドを使用します。
- SAPIQ レプリケートデータベースに `rs_threads`、`rs_lastcommit`、および `rs_ticket_history` の各テーブルを作成します。
- SAP IQ への接続を設定するために、次のようにしてデフォルトのファンクション文字列クラスとエラークラスの接続プロパティを設定します。

```
set error class rs_iq_error_class
set function string rs_iq_function_class
```

## SAP IQ への接続の作成

レプリケート SAP IQ データベースへの接続を設定します。

1. `create connection` を `using profile` 句および該当する接続プロファイルを指定して使用し、レプリケート SAP IQ データサーバおよびデータベースを指定します。たとえば、Oracle プライマリデータサーバからの接続を作成するには、次のようにします。

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

SAP IQ データベースへの複写パスを複数作成して、複写負荷を分散できます。各パスにユニークなメンテナンスユーザ ID を使用します。

2. `admin who` を使用して、Replication Server が SAP IQ に正常に接続することを確認します。

参照：

- SAP IQ への Multi-Path Replication (167 ページ)

## SAP IQ データベースオプションの設定

`rs_session_setting` 関数と `create function string` コマンドを組み合わせることで、SAPIQ レプリケートデータベースに接続している間の SAPIQ パラメータの

## レプリケートデータサーバとしての SAP IQ

値を設定できます。たとえば、パフォーマンスを最適化するパラメータ値を設定できます。

1. 新しいファンクション文字列 (**my\_iq\_fclass** という名前) を作成して、**rs\_iq\_function\_class** を親クラスとして設定します。

```
create function string class my_iq_fclass
set parent to rs_iq_function_class
go
```

2. **my\_iq\_fclass** ファンクション文字列クラスの **rs\_session\_setting** ファンクション文字列を作成して、設定する SAP IQ パラメータおよび値を含めます。たとえば、**LOAD\_MEMORY\_MB**、**MINIMIZE\_STORAGE**、および **JOIN\_PREFERENCE** SAP IQ データベースオプションの値を設定して、パフォーマンスを最適化します。

```
create function string rs_session_setting
for my_iq_fclass
output language
'set temporary option Load_Memory_MB=''200''
set temporary option Minimize_Storage=''on''
set temporary option join_preference=5'
go
```

『Replication Server リファレンスマニュアル』の「Replication Server システム関数」の「**rs\_session\_setting**」を参照してください。

3. **my\_iq\_fclass** ファンクション文字列クラスを使用するように、IQSRVR データサーバにある iqdb データベースへのコネクションを変更します。

```
alter connection to IQSRVR.iqdb
set function string class to my_iq_fclass
go
```

## RTL を有効にする

該当するパーミッションを付与し、レプリケート SAP IQ データベースに接続したら、SAP IQ への複写について RTL を有効して設定できます。

接続の RTL を有効にするには、**dsi\_compile\_enable** を使用します。

**dsi\_compile\_enable** を off に設定した場合、Replication Server はログ順、ローごとの連続複写モードを使用します。たとえば、テーブル上のすべてのオペレーションをログ順に複写する必要があるトリガがテーブルにあるためコンパイルを使用できない場合のように、最終ロー変更を複写すると問題が発生する場合、問題のテーブルで **dsi\_compile\_enable** を off に設定します。

**dsi\_compile\_enable** を on に設定すると、Replication Server は **dsi\_cmd\_prefetch** と **dsi\_num\_large\_xact\_threads** を無効にします。

---

**留意：** SAP IQ への Real-Time Loading (RTL) 複写を有効にする前に、**dsi\_bulk\_copy** と **dynamic\_sql** を off に設定する必要があります。

---



特定のデータベースにのみ影響するように、RTL をデータベースレベルで有効にして設定するには、次のように入力します。

```
alter connection to IQ_data_server.iq_database
set dsi_compile_enable to 'on'
go
```

RTL をサーバまたはテーブルレベルで有効にして設定することもできます。

- サーバレベル - Replication Server へのすべてのデータベースコネクションに影響します。

```
configure replication server
set dsi_compile_enable to 'on'
```

- テーブルレベル - 指定した複製テーブルのみに影響します。テーブルレベルとデータベースレベルの両方でパラメータを指定している場合は、テーブルレベルのパラメータがデータベースレベルのパラメータよりも優先されます。テーブルレベルでパラメータを指定しなければ、データベースレベルのパラメータの設定が適用されます。テーブルにパラメータを設定するには、**alter connection** と **for replicate table named** 句を使用します。次に例を示します。

```
alter connection to IQ_data_server.iq_database
for replicate table named dbo.table_name
set dsi_compile_enable to 'on'
```

**for replicate table name** 句の使用によってテーブルレベルのコネクション設定を変更できます。設定の変更は指定したテーブルのすべてのサブスクリプションからの複製データと複製定義に適用されます。

---

**注意：** テーブルレベルの設定には、**alter connection** しか使用できません。これは Replication Server が **for** 句を **create connection** に対してサポートしていないためです。

---

**dsi\_compile\_enable** を実行した後、レプリケート SAP IQ データベースへの接続をサスペンドして再開します。

## RTL パフォーマンスのチューニング

Replication Server は、いくつかのパラメータの推奨デフォルト値を自動的に設定します。これらのパラメータの値を変更すると、複製パフォーマンスを微調整できます。

変更するパラメータごとに個別の **alter connection** コマンドを実行する必要があります。 **alter connection** を入力した後は、2つ以上のパラメータを入力しないでください。

パラメータの詳細な説明については、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**alter connection**」を参照してください。

### ***dsi\_bulk\_threshold***

**dsi\_bulk\_threshold** は、特定のコマンドタイプのテーブルでコンパイルが行われた後の最終的なロー変更コマンド数を指定します。その数に達したら、それがトリガーになって、Replication Server はそのテーブルの同じコマンドタイプにバルクコピーインを使用します。デフォルトの最終的なロー変更コマンド数は 20 です。

デフォルトの最終的なロー変更コマンド数は 20 です。

例：

```
alter connection to IQSRVR.iqdb
set dsi_bulk_threshold to '15'
go
```

### ***dsi\_cdb\_max\_size***

**dsi\_cdb\_max\_size** は、RTL 処理中に Replication Server が生成できる最終的な変更を保管するデータベースの最大サイズ (メガバイト単位) を指定します。

デフォルトは 1024MB です。

例：

```
alter connection to IQSRVR.iqdb
set dsi_cdb_max_size to '2048'
go
```

Replication Server は、SAP IQ への Real-Time Loading (RTL) にフルインクリメンタルコンパイルを使用します。フルインクリメンタルコンパイルでは、最終的な変更を保管するデータベースのインスタンス内でコンパイルされたトランザクションセグメントのコマンド数が、**dsi\_compile\_max\_cmds** のスレッシュホールドを超える場合、または最終的な変更を保管するデータベースのインスタンスサイズが **dsi\_cdb\_max\_size** のスレッシュホールドを超える場合、Replication Server は、最終的な変更を保管するデータベースのインスタンスに対して、そのトランザクションをレプリケートデータベースに送信し、インスタンスが消費したメモリを解放するように指示します。

### ***dsi\_compile\_max\_cmds***

**dsi\_compile\_max\_cmds** は、Replication Server が 1 つのコンパイルされたトランザクションにコンパイルできるトランザクションとコマンドのグループの最大サイズをコマンド数で指定します。RTL がコンパイルしている現在のグループで最大グループサイズに達すると、RTL は新しいグループを開始します。Replication Server は最終的な変更を保管するデータベースのインスタンスを作成して、コンパイルされたトランザクションを格納します。Replication Server は、

**dsi\_compile\_max\_cmds** によってグループに許可されるコマンドの最大数に対応させるため、最終的な変更を保管するデータベースのサイズを増やします。

Replication Server がコンパイル中の現在のグループの最大グループサイズに達する

と、Replication Server は、コンパイルされたトランザクションをレプリケートデータベースのワークテーブルに転送し、最終的な変更を保管する特定のデータベースによって消費されたメモリを開放し、新しいグループを開始して、そのグループのための最終的な変更を保管する新しいデータベースのインスタンスを作成します。

読み込むデータがなくなると、グループが最大コマンド数に達していても、RTL は現在のトランザクションのセットを現在のグループにグループ化する処理を終了します。

デフォルトは 10,000 コマンドです。

例：

```
alter connection to IQSRVR.iqdb
set dsi_compile_max_cmds to '50000'
go
```

### ***dsi\_compile\_retry\_threshold***

**dsi\_compile\_retry\_threshold** は、グループ内のコマンド数に対するスレッシュホールド値を指定します。失敗したトランザクションを含むグループ内のコマンド数が **dsi\_compile\_retry\_threshold** の値より小さい場合、Replication Server は RTL モードでそのグループのリトライ処理を行わないので、処理時間を節約してパフォーマンスを向上できます。代わりに、Replication Server は連続複写モードに切り替わります。連続複写モードでは、プライマリデータベースのログ順に従って変更がレプリケートデータベースに送信されます。

デフォルトは 100 コマンドです。

**dsi\_compile\_retry\_threshold** を設定するときに、データベースコネクションをサスペンドしてレジュームする必要はありません。このパラメータはコマンドを実行するとすぐに有効になります。

例：

```
alter connection to IQSRVR.iqdb
set dsi_compile_retry_threshold to '200'
go
```

『Replication Server 管理ガイド 第 2 巻』の「エラーと例外の処理」の「データサーバのエラー処理」>「ローカウントの検証」>「ローカウントの検証を制御する」を参照してください。

### ***dsi\_command\_convert***

**dsi\_command\_convert** は、複写コマンドの変換方法を指定します。

変換の種類は次のオペレーションの組み合わせによって指定されます。

- **d** - delete
- **i** - insert

## レプリケートデータサーバとしての SAP IQ

- **u** - update
- **t** - truncate
- **none** - オペレーションなし

**dsi\_command\_convert** に対するオペレーションの組み合わせには、**i2none**、**u2none**、**d2none**、**i2di**、**t2none**、**u2di** があります。変換前のオペレーションは "2" の前に、変換後のオペレーションは "2" の後ろにあります。例：

- **d2none - delete** コマンドを複写しない。このオプションでは、**rs\_delete** ファンクション文字列をカスタマイズする必要はありません (**delete operations** オペレーションを複写しない場合)。
- **i2di,u2di - insert** と **update** の両方を **delete** とその後の **insert** に変換する。これはオートコレクションと同等のオペレーションです。**dsi\_row\_count\_validation** を off にすることによってローカウムの検証を無効にする場合、複写時に重複キーエラーを避け、データベースの自動同期ができるようにするために、**dsi\_command\_convert** を **i2di,u2di** に設定するようおすすめします。
- **t2none - truncate table** コマンドを複写しない。

**dsi\_command\_convert** のデフォルトは **none** です。これは、コマンドの変換がないことを意味します。

例：

```
alter connection to IQSRVR.iqdb
set dsi_command_convert to 'i2di,u2di'
go
```

参照：

- メモリ消費の制御 (163 ページ)

## リトライメカニズムの強化

リトライメカニズムの強化を使用すると、Replication Server によるコンパイルおよび一括適用の回数が減らされるため、レプリケーションのパフォーマンスが向上します。

RTL はできるだけ多くのコンパイル可能なトランザクションをグループ化して、グループ内のトランザクションをまとめた最終的な変更としてコンパイルしてから、レプリケートデータベースでバルクインタフェースを使用してその変更をレプリケートデータベースに適用しようとします。RTL の処理結果から発生するレプリケーションのトランザクションが失敗すると、RTL はリトライメカニズムを呼び出します。グループ内のトランザクションが失敗すると、RTL はそのグループを同じサイズの 2 つのグループに分割し、コンパイルとバルク適用を各グループに対して試みます。リトライメカニズムは失敗したトランザクションを特定し、Replication Server がエラーアクションのマッピングを実行できるようにします。ま

た、DSIが停止する場合もあるので、失敗したトランザクションの前にあるすべてのトランザクションが適用されます。

RTL 内の最終的な変更を保管するデータベースは、トランザクションの最終的なロー変更、つまりコンパイルしたトランザクションを保管するインメモリレポジトリとして機能します。最終的な変更を保管するデータベースの内容は、複数のプライマリトランザクションからのコマンドを集約したものであり、RTL ではログ順に適用されません。したがって、リトライメカニズムがないと失敗したトランザクションを特定する方法がありません。グループ内のトランザクションが失敗したら、リトライメカニズムはそのグループを分割してコンパイルとバルク適用を繰り返します。このような連続したリトライプロセスはパフォーマンスの低下の原因となります。

リトライメカニズムの強化によって、RTL でトランザクションが失敗したグループが検出された場合にグループが3等分され、失敗したトランザクションを含むグループの特定がより効率的に行われるようになりました。

さらに **dsi\_compile\_retry\_threshold** パラメータを使用してグループ内のコマンド数にスレッシュド値を指定できます。失敗したトランザクションを含むグループ内のコマンド数が **dsi\_compile\_retry\_threshold** の値より小さい場合、Replication Server は RTL モードでそのグループのリトライ処理を行わないので、処理時間を節約してパフォーマンスを向上できます。代わりに、Replication Server は連続複写モードに切り替わります。連続複写モードでは、プライマリデータベースのログ順に従って変更がレプリケートデータベースに送信されます。

## メモリ消費の制御

RTL は、フルインクリメンタルコンパイルを使用してメモリ消費を制御するため、最終的な変更を保管するデータベースのサイズを制御してメモリの消費量を削減できます。

### RTL の SQT メモリ消費の制御

RTL でトランザクションプロファイリング中に DSI SQT キャッシュでパックされていないコマンドが消費する最大メモリ量を制御します。

SQT スレッドは、RTL のトランザクションプロファイリング処理によってアンパックされたコマンドが使用し、DSI SQT キャッシュによって参照されるメモリをモニタします。

Replication Server が RTL を使用して複写を行っている場合、DSI スレッドで使用される最大メモリ量は **dsi\_sqt\_max\_cache\_size**、**sqt\_max\_prs\_size**、および **dsi\_cdb\_max\_size** の合計です。**dsi\_sqt\_max\_cache\_size**、**sqt\_max\_prs\_size**、および **dsi\_cdb\_max\_size** に小さい値を設定するとメモリ消費は低減しますが、レプリケーションパフォーマンスは低下します。最適なメモリ消費とパフォーマンスを

実現するには、複製環境をチューニングします。パラメータの設定については、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」を参照してください。

### 最終的な変更のデータベースのサイズ

最終的な変更を保管するデータベースのメモリ消費を低減します。それには、最終的な変更を保管するデータベースのサイズがスレッシュホールドサイズに達したらそのデータベースをトリガして、データをレプリケートデータベースにフラッシュします。

メモリの消費量は、最終的な変更が保管されるデータベースなどの Replication Server データ構造および構造が格納されるデータを参照します。最終的な変更が保管されるデータベースは、インメモリデータ構造です。最終的な変更が保存されるデータベースのメモリ消費は、Replication Server が多数のカラムを持つテーブルや、大きな text および image データ型値を持つテーブルに適用されたコマンドをコンパイルすると、劇的に増加することがあります。たとえば、100 のカラムを持つテーブルで 1,000,000 のローをコンパイルすると、10 のカラムを持つテーブルで同数のローをコンパイルするのと比べて約 10 倍のメモリが消費されます。他の処理やモジュールに必要なメモリが不十分な場合、複製パフォーマンスは低下します。

Replication Server は、**dsi\_cdb\_max\_size** および **dsi\_compile\_max\_cmds** に設定した値を使用して、メモリの消費量を制御します。**dsi\_cdb\_max\_size** を使用して、Replication Server が生成できる最終的な変更を保管するデータベースの最大サイズを制御できます。このサイズが設定されたスレッシュホールドに達すると、Replication Server は最終的な変更を保管するデータベースに作成中のコンパイル済みトランザクションに新しいコマンドとトランザクションをコンパイルする処理を停止し、コンパイル済みグループをレプリケートデータベースに一括して適用し、最終的な変更を保管するデータベースをクリアし、そのデータベースが消費していたメモリを解放します。

Replication Server が生成する最終的な変更を保管するデータベースのインスタンス数は、**dsi\_cdb\_max\_size** および **memory\_limit** で設定する値によって異なります。RTL を使用する複製システムの予測メモリ要件は、複製接続の数に **dsi\_cdb\_max\_size** を掛けた値になります。

### フルインクリメンタルコンパイル

フルインクリメンタルコンパイルにより、Real-Time Loading (RTL) のレプリケーションパフォーマンスが向上しますが、これは多くのコマンドを含む大規模なコンパイル可能なトランザクションの処理中のメモリ消費が低減された結果です。

フルインクリメンタルコンパイルでは、**insert**、**delete**、または **update** の混合オペレーションを含む大規模なトランザクションをコンパイルできます。Replication

Server では、フルインクリメンタルコンパイルを使用して大規模なコンパイル可能なトランザクションをレプリケートデータベースに適用します。その際、最終的な変更を保管する複数のインメモリデータベースインスタンスを使用します。フルインクリメンタルコンパイルは、大規模なトランザクションをセグメントのシーケンスに分割します。各セグメントはコマンドグループで構成されます。

Replication Server は各セグメントをコンパイルし、1つのセグメントを格納するために最終的な変更を保管する専用データベースを作成します。Replication Server は、最終的な変更を保管するデータベースインスタンスにセグメントをレプリケートデータベースに送信して適用するよう指示します。この後、Replication Server は、最終的な変更を保管するデータベースインスタンスを閉じ、そのデータベースが消費していたメモリを解放します。Replication Server は、次のトランザクションセグメントに対して別の最終的な変更を保管するデータベースインスタンスを作成し、最終的な変更を保管するデータベースインスタンスをすべてのセグメントに対して順序どおりに引き続き作成してから閉じます。

このため、最終的な変更を保管する大規模なデータベースインスタンスに対して単一のメモリの大部分を消費して大規模トランザクションを保持するのではなく、フルインクリメンタルコンパイルによってメモリ要件は、トランザクションのセグメントのみを含んだ単一の最終的な変更を保管する小さなデータベースインスタンスが消費するメモリに低減されます。フルインクリメンタルコンパイルは、使用した最終的な変更を保管するデータベースインスタンスの数でメモリ要件を除算します。たとえば、フルインクリメンタルコンパイルが最終的な変更を保管する 10 のデータベースインスタンスを使用して大規模なトランザクションを適用する場合、メモリ要件はフルインクリメンタルコンパイルを伴わない要件の約 10 分の 1 です。

### メモリ制御パラメータおよび Replication Server の処理

Replication Server のアクションは、メモリ制御パラメータに対して設定する値によって異なります。

#### **dsi\_cdb\_max\_size の異なる値への設定**

以下は、Replication Server が 2 つのテーブルで 100,000 の更新によってトランザクションを適用しているのを示しています。テーブル 1 には約 4GB のメモリを必要とする 100 のカラムがあり、テーブル 2 にはメモリの約 10 分の 1 の 400MB を必要とする 10 のカラムがあります。

| dsi_cdb_max_size<br>値 (MB) | テーブル名  | 複写処理への影響   |
|----------------------------|--------|--|
| 1024 (デフォルト)               | テーブル 1 | 前提条件: Replication Server の <b>memory_limit</b> を 1GB の最終的な変更を保管するデータベースを作成できる大きさの値に設定する。<br><br>Replication Server は 1GB の最終的な変更を保管するデータベースのインスタンスを 4 つ使用して、トランザクションを適用する。     |
| 1024 (デフォルト)               | テーブル 2 | 前提条件: Replication Server の <b>memory_limit</b> を 400MB の最終的な変更を保管するデータベースを作成できる大きさの値に設定する。   |
| 4096                       | テーブル 1 | 前提条件: Replication Server の <b>memory_limit</b> を 4GB の最終的な変更を保管するデータベースを作成できる大きさの値に設定する。<br><br>Replication Server は、4GB の最終的な変更を保管するデータベースのインスタンスを 1 つ使用して、トランザクションを適用する。     |
| 4096                       | テーブル 2 | 前提条件: Replication Server の <b>memory_limit</b> を 400MB の最終的な変更を保管するデータベースを作成できる大きさの値に設定する。<br><br>Replication Server は、400MB の最終的な変更を保管するデータベースのインスタンスを 1 つ使用して、トランザクションを適用する。 |

### RTL のインクリメンタル解析

メモリ消費をさらに低減するには、**dsi\_incremental\_parsing** を on に設定して、DSI スケジューラスレッドによるインクリメンタル解析を有効にします。

『Replication Server 管理ガイド 第 2 巻』の「インクリメンタル解析」を参照してください。



## SAP IQ への Multi-Path Replication

---

複写のスループットとパフォーマンスを向上させ、遅延と競合を軽減するために、Replication Server からレプリケート SAP IQ データベースへのコネクションを複数作成します。

Adaptive Server または Oracle プライマリデータベースから Replication Server への複数のコネクションと、Replication Server からレプリケート SAP IQ データベースへの複数のコネクションを使用して、エンドツーエンドの複数の複写パスを作成できます。

### データベースのサポート

- プライマリデータベース
  - Adaptive Server 15.7 以降。
  - Oracle 10g と 11g。Replication Server Options の『Replication Agent リリースノート』の「製品の概要」の「製品の互換性」を参照してください。
- レプリケートデータベース - SAP IQ バージョン 15.1 以降。『Replication Server リリースノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。

### ライセンス

Multi-Path Replication™ は、Advanced Services Option の一部としてライセンスされます。RTL を使用した SAP IQ への複写は、Real-Time Loading Edition (RTLE) で実行可能です。『Replication Server インストールガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

### 参照：

- 異機種間における Multi-Path Replication (209 ページ)
- Adaptive Server から SAP IQ への Multi-Path Replication (226 ページ)
- Oracle から SAP IQ への Multi-Path Replication (244 ページ)

## SAP IQ への代替レプリケートコネクションの作成

---

**create alternate connection** を **using profile** 句と一緒に使用して、Replication Server からレプリケート SAP IQ データベースへの代替コネクションを作成します。

### 前提条件

代替コネクションを作成する前に、レプリケートデータベースへのデフォルトのコネクションを作成します。

### 手順

コネクションプロファイルとコネクションプロファイルのバージョン、およびデフォルトのコネクションと各代替コネクションの一意のメンテナンスユーザ名を指定する必要があります。

SAP IQ レプリケートデータベースへの代替コネクションを作成します。

```
create alternate connection to dataserver.database
named conn_server.conn_db
using profile connection_profile;version
set username [to] user
set password [to] pwd
```

構文の説明は次のとおりです。

- *dataserver* および *database* - レプリケートデータサーバとデータベースです。
- *conn\_server.conn\_db* - データサーバ名とコネクション名で構成される代替レプリケートコネクション。
  - 各レプリケートコネクション名は、複写システム内でユニークにしてください。
  - *conn\_server* が *dataserver* と異なる場合は、interface ファイルに *conn\_server* のエントリが必要です。
  - *conn\_server* が *dataserver* と同じ場合は、*conn\_db* が *database* と異なるようにしてください。
- *connection\_profile* - レプリケートデータベースの正しいファンクション文字列クラスとエラークラスを指定します。また、クラスレベル変換の定義とレプリケートデータベースオブジェクトの作成のサポートを含める場合があります。
  - *rs\_ase\_to\_iq* - Adaptive Server から SAP IQ への複写
  - *rs\_oracle\_to\_iq* - Oracle から SAP IQ への複写

---

**注意：** 作成する SAP IQ データベースへの各代替コネクションに対し、接続プロファイルを指定する必要があります。

---

- *version* - 使用する接続プロファイルのバージョン。

---

**注意：** 作成する SAP IQ データベースへの各代替コネクションに対し、接続プロファイルのバージョンを指定する必要があります。

---

- *user* - SAP IQ データベースへの各コネクションの Replication Server メンテナンスユーザのログイン名。Replication Server は、複写データを管理するのにこのログイン名を使用します。ネットワークベースのセキュリティを有効に設定していない場合には、ユーザ名を指定する必要があります。

---

**注意：** 作成する SAP IQ データベースへの各代替コネクションに対し、異なるメンテナンスユーザ名を使用する必要があります。異なる Replication Server から SAP IQ データベースへのコネクションを作成する場合も、必ずユニークなメンテナンスユーザ名を使用してください。ユニークなユーザ名を使用しないと、データの重複エラーが発生する場合があります。複写システムは、別

の Replication Server からコネクションを作成するために同じユーザ名を使用しているかどうかを検出できません。

たとえば、プライマリデータベースが Adaptive Server で、dbmaint2 が IQSRVR.iqdb\_conn2 のメンテナンスユーザである場合、IQSRVR SAP IQ データサーバ内の iqdb レプリケートデータベースへの IQSRVR.iqdb\_conn2 という代替レプリケートコネクションを作成するには、次のように入力します。

```
create alternate connection to IQSRVR.iqdb
named IQSRVR.iqdb_conn2
using profile rs_ase_to_iq;standard
set username to dbmaint2
set password to dbmaint2pwd
go
```

## 代替レプリケート SAP IQ コネクションの変更または削除

**alter connection** コマンドと **drop connection** コマンドを使用して、SAP IQ へのデフォルトのレプリケートコネクションまたは代替レプリケートコネクションを変更または削除します。

コマンドで指定するデータサーバ名とデータベース名は、デフォルトまたは代替レプリケートコネクション名にすることができます。

代替またはデフォルトのレプリケートコネクションを設定するときに、**alter connection** で使用可能な設定パラメータを使用することができます。

たとえば、**dsi\_bulk\_threshold** を IQSRVR.iqdb\_conn2 代替レプリケートコネクションに対して 15 に設定する場合、次のように入力します。

```
alter connection to IQSRVR.iqdb_conn2
set dsi_bulk_threshold to '15'
go
```

## レプリケートコネクションに関する情報の表示

**replicate** パラメータを **admin show\_connections** と一緒に使用して、すべてのレプリケートコネクションに関する情報を表示します。

たとえば、IQSRVR データサーバでレプリケートデータベースを制御する Replication Server で、次のように入力します。

```
admin show_connections, 'replicate'
```

次のような情報が表示されます。

| Connection Name   | Server | Database | User      |
|-------------------|--------|----------|-----------|
| IQSRVR.iqdb       | IQSRVR | iqdb     | db_maint  |
| IQSRVR.iqdb_conn2 | IQSRVR | iqdb     | db_maint2 |
| IQSRVR.iqdb_conn3 | IQSRVR | iqdb     | db_maint3 |

## レプリケートデータサーバとしての SAP IQ

IQSRVR.iqdb は、コネクション名がデータサーバとデータベース名の組み合わせに一致するため、Replication Server と IQSRVR データサーバの iqdb データベースの間のデフォルトのコネクションになります。

IQSRVR.iqdb\_conn2 および IQSRVR.iqdb\_conn3 は、コネクション名がデータサーバとデータベース名の組み合わせに一致しないため、Replication Server と IQSRVR データサーバの iqdb データベースの間の代替コネクションになります。

## レプリケーションロード分散統計

プライマリデータベースでサポートされている分散モードを使用して、使用可能なレプリケートパス上のレプリケートロードを分散します。

### 分散モード

オブジェクトバインドモードによる Adaptive Server 分散では、特定のパスにオブジェクトをバインドすることにより、複数のパスにテーブルやストアドプロシージャなどのオブジェクトを分散できます。プライマリコネクションとレプリケートコネクション名が一致する場合、オブジェクトは、プライマリデータサーバからレプリケートデータサーバへのエンドツーエンドの複写パスに続きます。

『Replication Server 管理ガイド 第2巻』>「パフォーマンスチューニング」>「Multi-Path Replication」>「複数のプライマリ複写パス」>「複写パスへのオブジェクトのバインド」を参照してください。

コネクションモード別 Adaptive Server 分散では、Adaptive Server RepAgent がさまざまなクライアントプロセスから発生したトランザクションを使用可能な複写パスに割り当てます。時間の経過とともに、使用可能なパス全体でデータ分散のバランスが取れていく傾向があります。使用可能な RepAgent パスがさらにあり、クライアントプロセスの数が多い場合、複写パフォーマンスが向上し、複写負荷分散はより均一化します。『Replication Server 管理ガイド 第2巻』>「パフォーマンスチューニング」>「Multi-Path Replication」>「並列トランザクションストリーム」>「Multi-Path Replication の分散モード」を参照してください。

### SAP IQ マルチプレックスノード

各ノードへのコネクションを作成し、関連する interfaces ファイルエントリを作成することにより、レプリケートコネクションを SAP IQ マルチプレックスの異なるノードに割り付けて複写の負荷を分散することができます。『SAP IQ 管理: マルチプレックス』を参照してください。

## 分散モデルの設定

複数のプライマリ複写パスを使用して複写の分散モデルを設定します。

### 前提条件

プライマリ Adaptive Server から Replication Server へのデフォルトのコネクションと代替コネクションを作成し、マルチスレッド RepAgent を有効にします。分散モデルをカラムフィルタ別分散に設定する場合は、複数のスキャナを有効にします。

### 手順

分散モデルを変更して、新しいバインドを追加した場合や RepAgent が既存のバインドを新しい分散モデルと関連付けることができなくなった場合、RepAgent では、新しい分散モデルで一部のバインドが無視されることを示す警告が表示されます。ただし、RepAgent では非アクティブなバインドも保持されます。非アクティブなバインドのタイプと対応する分散モデルに戻すと、RepAgent は以前の非アクティブなバインドを再度使用するようになります。たとえば、オブジェクトバインド別分散からカラムフィルタ別分散に変更すると、RepAgent は変更前に設定していたテーブルとストアドプロシージャのバインドをすべて無視します。

#### 1. 分散モデルを設定します。

```
sp_config_rep_agent database, 'multipath distribution model',
{'object' | 'connection' | 'filter'}
```

構文の説明は次のとおりです。

- **multipath distribution model - sp\_config\_rep\_agent** の分散モデルパラメータです。
- **object** - モデルをオブジェクトバインド別分散に設定します (デフォルト)。
- **connection** - モデルをコネクション別分散に設定します。
- **connection** - モデルをフィルタ別分散に設定します。

#### 2. Replication Server をクワイス状態にし、RepAgent を再起動します。

『Replication Server 管理ガイド 第 1 巻』>「複写システムの管理」>「Replication Server のクワイス」>「複写システムのクワイス」を参照してください。

## 参照制約のあるテーブル

参照制約 (外部キーその他の検査制約など) のあるテーブルの指定には複写定義を使用できます。それによって RTL にそれらのテーブルの存在が通知されます。

通常は、参照元のテーブルには同じプライマリデータベース内の参照先テーブルに対する参照制約が含まれています。RTL では複数のプライマリデータベースからの参照先テーブルをサポートするよう参照制約が拡張されています。

各プライマリデータベースに対する複写定義内で参照元テーブルを指定できます。ただし、複数の参照制約が互いに競合する場合は、Replication Server によってランダムにテーブルが1つ選択されます。

参照：

- RTL の処理と制限事項 (149 ページ)

### 複写定義の作成と変更

参照制約のあるテーブルの指定には、**references** パラメータを指定して **create replication definition** コマンドを使用します。

```
create replication definition
...
(column_name [as replicate_column_name]
...
[map to published_datatype]] [quoted]
[references [table_owner.]table_name [(column_name)]] ...)
....]
```

参照元のテーブルの追加と変更には、**references** パラメータを指定して **alter replication definition** コマンドを使用します。参照を削除するには、**null** オプションを使用します。

```
alter replication definition
.....
add column name [as replicate_column_name]
[map to published_datatype] [quoted]
[references [table_owner.]table_name [(column_name)]]
...
| alter columns with column_name references
{[table_owner.]table_name [(column_name)] | NULL}
[, column_name references {[table_owner.]table_name [(column_name)]
| NULL}
...
...
```

**alter replication definition** と **create replication definition** の両方に **reference** 句を指定すると、Replication Server の動作は次のようになります。

- **reference** 句をカラムプロパティとして扱います。各カラムはテーブルを1つだけ参照できます。
- **reference** 句内の **column\_name** パラメータに指定したカラム名を処理しません。
- 循環参照になる参照制約を許可しません。たとえば、元の参照先テーブルは元の参照元テーブルへの参照制約を持つことはできません。

複写プロセスでは、RTL は次のようにロードします。

- 参照先テーブルへの挿入の後で複写定義で指定した参照元テーブルに挿入します。
- 複写定義で指定したテーブルでの削除の後で参照先テーブルで削除します。

場合によっては、両方のテーブルでの更新が競合によって失敗することがあります。RTL が複写処理のリトライをしないようにして、パフォーマンスの低下を防ぐには、以下を行います。

- 更新を削除と挿入に変換するように、**dsi\_command\_convert** を "u2di" に設定してレプリケーションの更新を停止します。
- **dsi\_compile\_enable** を off にして、影響を受けたテーブルがコンパイルされるのを避けます。

カスタムファンクション文字列を持つテーブルと、コンパイルできない既存テーブルへの参照制約を持つテーブルは RTL でコンパイルできません。これらのテーブルにマークを付けることによって、RTL は参照制約エラーによって発生するトランザクションのリトライを避け、複写処理を最適化できます。

## RTL 情報の表示

---

設定パラメータプロパティとテーブル参照の情報を表示できます。

### 設定パラメータプロパティの表示

**admin config** を使用して、例に示されているようなデータベースレベルとテーブルレベルの設定パラメータを表示します。

- データベースレベル:
  - NY\_DS データサーバ(NY\_DS.nydb1)の nydb1 データベースへの接続に使用するデータベースレベルの設定パラメータをすべて表示するには、次のように入力します。  

```
admin config, "connection", NY_DS, nydb1
```
  - To verify that **dsi\_compile\_enable** is on for the connection to NY\_DS.nydb1, enter:  

```
admin config, "connection", NY_DS, nydb1, dsi_compile_enable
```
  - **dsi\_compile\_enable** など、名前の一部に "enable" があるデータベースレベルの設定パラメータをすべて表示するには、次のように入力します。  

```
admin config, "connection", NY_DS, nydb1, "enable"
```
- テーブルレベル:
 

**dsi\_command\_convert** を使用して NY\_DS データサーバの nydb1 データベースにある tb1 テーブルで **d2none** を設定した後、すべての設定パラメータを表示するには、次のように入力します。

```
admin config, "table", NY_DS, nydb1
```

『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**admin config**」を参照してください。

#### テーブル参照の表示

テーブル参照の情報と RTL の情報を表示するには、**rs\_helprep** を使用します。これは、Replication Server システムデータベース (RSSD) 上で実行できます。

**create replication definition** を使用して作成した **authors\_repdef** 複写定義に関する情報を表示するには、次のように入力します。

```
rs_helprep authors_repdef
```

『Replication Server リファレンスマニュアル』の「RSSD ストアドプロシージャ」の「**rs\_helprep**」を参照してください。

## Replication Server 15.5 のシステムテーブルサポート

Replication Server では **rs\_tbcconfig** テーブルをテーブルレベルの設定パラメータの保管に使用し、**rs\_columns** テーブルの **ref\_objowner** カラムと **ref\_objname** カラムを参照制約のサポートに使用します。

テーブルの詳細については、『Replication Server リファレンスマニュアル』の「Replication Server システムテーブル」を参照してください。

## 混合バージョンのサポートと下位互換性

RTL では、複写定義で指定されている参照制約を複写できるのは、アウトバウンドルートバージョンが 15.5 以降の場合のみです。

アウトバウンドルートバージョンが 15.5 より古くても RTL は機能します。しかし、バージョン 15.5 以降の場合は、参照制約情報を Replication Server で使用できません。

連続複写モードはサポートされているすべてのバージョンの Replication Server のデフォルト複写モードです。RTL を使用できるのは Replication Server 15.5 以降のみです。



## SAP IQ への複写シナリオ

---

RTL を使用した SAP IQ への複写をセットアップし、その複写作業をテストするシナリオについて説明します。

Adaptive Server データベース管理者 (ASE DBA)、Oracle データベース管理者 (Oracle DBA)、SAP IQ データベース管理者 (IQ DBA)、複写システム管理者 (RSA) は、Adaptive Server、Oracle、Replication Server、SAP IQ を複写用に準備して、SAP IQ データベースへのコネクションをセットアップする必要があります。

このシナリオで、*dbo* は、ASE\_DS プライマリ Adaptive Server または ORA\_DS プライマリ Oracle サーバの *pdb1* データベース内の *testtab* テーブルのテーブル所有者です。c1、c2、c3 は *testtab* 内のカラムで、データ型はそれぞれ *int*、*int*、*char(10)* です。IQSRVR は、*iqdb* データベースを含むレプリケート SAP IQ データサーバです。

### 参照：

- Adaptive Server から SAP IQ への Multi-Path Replication (226 ページ)
- Oracle から SAP IQ への Multi-Path Replication (244 ページ)

## interfaces ファイルのエントリの作成

---

レプリケート Replication Server と SAP IQ データサーバの *interfaces* ファイルに、互いのエントリを作成します。

1. SAP IQ データサーバの *interfaces* ファイル (Windows では *sql.ini* ファイル) に、レプリケート Replication Server のエントリを作成します。

---

**注意：**SAP IQ データサーバの *interfaces* ファイルが、SAP IQ が使用している SAP \$SYBASE ディレクトリ (Windows では %SYBASE% ディレクトリ) にない場合、このファイルを作成します。

---

2. レプリケート Replication Server の *interfaces* ファイルに、SAP IQ データサーバのエントリを作成します。

さまざまな SAP IQ マルチプレックスノードへの接続を作成している場合、影響を受ける各ノードのエントリをレプリケート Replication Server の *interfaces* ファイルに作成します。

## テストテーブルの作成

プライマリデータベースとレプリケートデータベース内にテストテーブルを作成し、複写作業をテストするためにメンテナンスユーザにそのテーブルに対するパーミッションを与えます。

1. データサーバにあるプライマリデータベース `pdb1` 内に、次の3つのカラムを持つ `testtab` という名前のテーブルを作成します。 `c1integer`、`c2integer`、および `c3char(10)`。  
たとえば、`Adaptive Server` で次を実行します。

```
use pdb1
go
create table dbo.testtab(c1 int primary key, c2 int,
c3 char(10))
go
```

Oracle のマニュアルを参照して Oracle データベースにテーブルを作成してください。

2. SAP IQ IQSRVR データサーバにあるレプリケートデータベース `iqdb` に次のように入力します。

```
use iqdb
go
create table dbo.testtab(c1 int primary key, c2 int,
c3 char(10))
go
grant all on dbo.testtab to public
go
```

## プライマリデータベースとレプリケートデータベースへの接続の作成

プライマリデータベース接続とレプリケートデータベース接続を作成します。

1. プライマリデータベースへの接続を作成します。
  - `Adaptive Server - Replication Server` の `rs_init` コーティリティを使用します。  
『`Replication Server 設定ガイド`』の「`rs_init` による `Replication Server` の設定とデータベースの追加」を参照してください。
  - Oracle - 詳細については、『`異機種間複写ガイド`』と、`Replication Server Options` 製品のマニュアルを参照してください。
2. SAP IQ レプリケートデータベースへの接続を作成します。

『`Replication Server リファレンスマニュアル`』の「`Replication Server コマンド`」の「`create connection using profile`」を参照してください。

---

**注意：** SAP IQ への接続の作成には、**rs\_init** は使用できません。

---

この例では、IQSRVR データサーバ内の iqdb データベースとデフォルト dbmaint の SAP IQ メンテナンスユーザを使用します。

- Adaptive Server の場合

```
create connection to IQSRVR.iqdb
using profile rs_ase_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

- Oracle の場合

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

コマンドが正常に実行されると、次のような出力が表示されます。

```
Connection to 'IQSRVR.iqdb' is created.
```

3. 接続が機能していることを確認します。

```
admin who
go
```

接続が機能していれば、次のような出力が表示されます。

```
SpidNameStateInfo
-----
63DSI EXECAwaiting Command103(1)IQSRVR.iqdb
62DSIAwaiting Message103 IQSRVR.iqdb
35SQMAwaiting Message103:0 IQSRVR.iqdb
```

## RTL の有効化

データベースレベルで RTL を有効にします。

### 前提条件

SAP IQ への Real-Time Loading (RTL) 複写を有効にする前に、**dsi\_bulk\_copy** と **dynamic\_sql** を **off** に設定します。

### 手順

1. 指定したデータベースのみに影響するように、データベースレベルで RTL を有効にして設定するには、次のように入力します。

```
alter connection to IQSRVR.iqdb
set dsi_compile_enable to 'on'
go
```

2. レプリケート SAP IQ データベースへの接続をサスペンドして再開し、接続の変更を有効にします。

## レプリケートデータサーバとしての SAP IQ

```
suspend connection to IQSRVR.iqdb
go
resume connection to IQSRVR.iqdb
go
```

### 複写テスト準備のためのテーブルへのマーク付け

プライマリデータベース内で SAPIQ データベースへ複写するテーブルにマークを付けます。

下記の例では、dbo は pdb1 プライマリデータベース内の testtab テーブル所有者です。c1、c2、c3 は testtab 内のカラムで、データ型はそれぞれ int、int、char(10) です。

1. testtab にデータローを挿入して複写をテストし、挿入が正常に行われることを確認します。

たとえば、Adaptive Server で次を実行します。

```
insert into testtab values(1,1,'testrow 1')
insert into testtab values(2,2,'testrow 2')
insert into testtab values(3,3,'testrow 3')
go
```

挿入が正常に実行されると、次のような出力が表示されます。

```
(1 row affected)
(1 row affected)
(1 row affected)
```

2. 複写対象の testtab にマークを付けます。

- Adaptive Server - **sp\_setrepdefmode** システムプロシージャを使用します。

- Adaptive Server 15.0.3 以降の場合、次を実行します。

```
sp_setrepdefmode testtab,'owner_on'
go
```

- Adaptive Server 15.0.3 より前の場合、次を実行します。

```
sp_setreptable testtab,'true', 'owner_on'
go
```

- Oracle - **pdb\_setreptable** Replication Agent コマンドを使用して、次を実行します。

```
pdb_setreptable pdb_table, mark, owner
```

使用に関する詳細については、Replication Server Options のマニュアルの『Replication Agent 管理ガイド』の「設定」の「プライマリデータベースオブジェクトのマーク付け」の「プライマリデータベースのテーブルへのマーク付け」を参照してください。

## 複写定義とサブスクリプションの作成

RTL を有効にして設定したら、SAP IQ への複写用にマークされているテーブルの複写定義とサブスクリプションを作成します。

1. `repdef_testtab` 複写定義を作成します。RTL をサポートする複写定義に必要なすべての参照制約句を追加します。

- Adaptive Server の場合:

```
create replication definition repdef_testtab
with primary at ASE_DS.pdb1
with primary table named 'testtab'
with replicate table named dbo.'testtab'
(c1 int, c2 int, c3 char(10))
primary key(c1)
go
```

- Oracle の場合:

```
create replication definition repdef_testtab
with primary at ORA_DS.pdb1
with primary table named 'TESTTAB'
with replicate table named dbo.'testtab'
(C1 as c1 int, C2 as c2 int, C3 as c3 char(10))
primary key(C1)
go
```

**注意：** デフォルトでは Oracle のオブジェクト名の文字設定はすべて大文字です。例に見られるように、複写定義でオブジェクト名を大文字から小文字に変換できます。Replication Agent for Oracle の設定パラメータ

`lcl_character_case` を使用して変換することもできます。詳細については、Replication Server Options のマニュアルの『Replication Agent リファレンスマニュアル』の「設定パラメータ」の「設定パラメータリファレンス」の「`lcl_character_case`」を参照してください。

2. 各テーブルとストアードプロシージャの複写定義に一致するサブスクリプションを作成します。

```
create subscription sub_testtab for repdef_testtab
with replicate at IQSRVR.iqdb
go
```

3. SAP IQ にログインして次を実行し、`testtab` がマテリアライズされたことを確認します。

```
select * from dbo.testtab
go
```

マテリアライゼーションが正常に行われると、次のような出力が表示されます。

```
c1c2c3
-----
```

## レプリケートデータサーバとしての SAP IQ

```
11testrow 1
22testrow 2
33testrow 3
(3 rows affected)
```

### 参照：

- 参照制約のあるテーブル (171 ページ)

## RTL が機能することを検証する

RTL が機能することを確認する方法について説明します。

1. プライマリデータサーバにログインして、testtab に新しいローを挿入するなどのオペレーションを実行します。  
たとえば、Adaptive Server で次を実行します。

```
insert into testtab values(4,4,'testrow 4')
insert into testtab values(5,5,'testrow 5')
insert into testtab values(6,6,'testrow 6')
go
```

次のような出力が表示されます。

```
(1 row affected)
(1 row affected)
(1 row affected)
```

2. SAPIQ にログインし、testtab への変更が SAPIQ データベースに複写されたことを確認します。

```
select * from dbo.testtab
go
```

複写が正常に実行されると、次のような出力が表示されます。

```
c1c2c3
-----
11testrow 1
22testrow 2
33testrow 3
44testrow 4
55testrow 5
66testrow 6
(6 rows affected)
```

## ステージングソリューションから RTL へのマイグレート

SAPIQ への複写に現在ステージングソリューションを使用している場合、Real-Time Loading ソリューションにマイグレートします。

このシナリオでは、pdb がプライマリデータベース、PRS がプライマリ Replication Server、RRS がレプリケート Replication Server、staging\_db がステー

ジングデータベース、iqdb がレプリケート SAP IQ データベースという複写構成を想定しています。このシナリオでのデータフローは、次のとおりです。

```
pdb -----> PRS -----> RRS -----> staging_db -----> iqdb
```

## ステージングソリューションからのマイグレートの準備

ステージングソリューションからマイグレートする前に、いくつかのタスクを実行する必要があります。

1. プライマリ Replication Server とレプリケート Replication Server の両方をバージョン 15.5 以降にアップグレードする必要があります。  
詳細については、『Replication Server インストールガイド』と『Replication Server 設定ガイド』を参照してください。
2. pdb に流れ込むトランザクションがないこと、およびマイグレート中は複写システムがクワイス状態にあることを確認します。
  - a) Replication Server で実行して、すべてのプライマリデータベースとシステムデータベースの Replication Agent を停止します。  

```
suspend log transfer from all
```
  - b) Adaptive Server を RSSD として使用している場合は、次のコマンドで RSSD の RepAgent を停止します。  

```
sp_stop_rep_agent rssid_name
```
  - c) 次を実行して、Replication Server のキューが排出され、Replication Server がクワイスされていることを確認します。  

```
admin quiesce_check
```

  
Replication Server がまだクワイスされていない場合は、**admin quiesce\_force\_rsi** で再試行します。Replication Server がクワイス状態になっていなければ、データが失われる可能性があります。
3. pdb と iqdb が同期していることを確認します。  
データベース間の再同期は、ステージングデータベースにすべてのデータが複写されてから、iqdb にステージングデータベースのデータをロードすることによって行います。データベース間の再同期を行わない場合は、iqdb のパーティションとマテリアライズを行う必要があります。
4. SAPIQ サーバがレプリケート Replication Server に接続してデータを抽出できるように、SAPIQ の interfaces ファイルにそのレプリケート Replication Server のエントリを追加します。

## Real-Time Loading ソリューションにマイグレート

ステージングソリューションから RTL にマイグレートします。

1. レプリケート SAP IQ データサーバでメンテナンスユーザを作成します。または、既存のメンテナンスユーザを使用することもできます。
2. 関連する接続プロファイルと手順 1 で作成したメンテナンスユーザ (*dbmaint* など) を使用して、レプリケート Replication Server からレプリケート SAP IQ データベースへのコネクションを作成します。

- Adaptive Server の場合：

```
create connection to IQSRVR.iqdb
using profile rs_ase_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

- Oracle の場合：

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

3. プライマリデータベースで、*dbo* が所有するテーブルが **owner\_on** としてマーク付けされていない場合、*dbo* は SAP IQ 内に存在しないので、SAP IQ がテーブルを検出できるようにそのテーブルの **owner\_on** を有効にする必要があります。

- Adaptive Server

- Adaptive Server 15.0.3 以降の場合、次を実行します。

```
sp_setrepdefmode testtab, 'owner_on'
go
```

- Adaptive Server 15.0.3. より前の場合、次を実行します。

```
sp_setreptable testtab, 'true', 'owner_on'
go
```

- Oracle

```
pdb_setreptable testtab, mark, owner
go
```

4. Adaptive Server の **owner\_on** または Oracle の **owner** を有効にしたので、複写定義を作成し直して所有者情報を含めます。
5. テーブル間に参照制約がある場合、Replication Server がその参照制約の存在を計算に入れてバルク適用を適切な順序で実行できるように、参照制約を定義して複写定義を変更する必要があります。
6. レプリケートデータベースへのコネクションを RTL で有効にします。

```
alter connection to iqserver_name.rdb
set dsi_compile_enable to 'on'
```



コネクションをサスペンドしてレジュームしたら、接続の変更が有効になります。

7. 各テーブルにサブスクリプションを作成します。プライマリデータベースとレプリケートデータベースが同期している場合は、**without materialization** 句をサブスクリプションに含めます。それ以外の場合は、マテリアライゼーション時にオートコレクションを有効にする必要があります。

これで、プライマリデータサーバから SAP IQ に直接複写できるようになりました。

#### 参照：

- 参照制約のあるテーブル (171 ページ)

### マイグレーション後のクリーンアップ

RTL を使用して複写を有効にして設定した後、ステージングソリューションのシステムをクリーンアップします。

1. ステージングデータベースのサブスクリプションを削除します。
2. 使用しない複写定義を削除します。
3. レプリケート Replication Server からステージングデータベースへのコネクションを削除します。
4. 環境を終了すると、ステージングデータベースから SAP IQ にデータが抽出されます。

## Replication Server と SAP IQ InfoPrimer の統合

Replication Server と SAP IQ InfoPrimer を統合することで、レプリケート SAP IQ データベースとは異なるスキーマを持つプライマリ Adaptive Server データベース間の複写をサポートします。

SAP IQ InfoPrimer には、データを変換して SAP IQ データベースにロードする効果的な機能が搭載されていますが、その抽出機能にはレプリケート SAP IQ データベースのデータを最新の状態に維持するために必要な Replication Server のリアルタイムモニタリングがありません。Replication Server Real-Time Loading (RTL) 機能では、バルクオペレーション処理とコンパイルされたオペレーションを使用して、高パフォーマンスの複写を達成しますが、Replication Server には SAP IQ InfoPrimer のデータ変換機能とロード機能がありません。Replication Server と SAP IQ InfoPrimer を統合することで、ソースとは異なるスキーマを持つレプリケート SAP IQ データベースで Adaptive Server データのほぼリアルタイムのコピーを維持できます。Replication Server と SAP IQ InfoPrimer の統合ソリューションは、初期データマテリアライゼーションと進行中のデータ処理という 2 つの部分で機能します。

### マテリアライゼーション

Replication Server と SAP IQ InfoPrimer の統合ソリューションは、Adaptive Server プライマリデータベースからレプリケート SAP IQ データベースへのデータのノンアトミックバルクマテリアライゼーションを実行します。このマテリアライゼーションは、Replication Server バルクマテリアライゼーションオプションに基づいており、必要に応じてオートコレクションを使用します。

SAP IQ InfoPrimer は、レプリケート SAP IQ データベースにステージングテーブルを作成し、各プライマリデータベーステーブルに対してマテリアライゼーション処理のデータ抽出手順を実行します。これらのステージングテーブルに対して変換ストアドプロシージャが実行され、結果はベーステーブルに書き込まれます。ベーステーブルは、エンドユーザテーブルとも呼ばれ、ビジネス分析に使用されます。

### 進行中のデータ処理

Replication Server は、指定されたテーブルに対してマテリアライゼーションフェーズで作成された同じステージングテーブルと変換ストアドプロシージャを使用します。可能な場合、Replication Server はオペレーションをコンパイルしてステージングテーブルにロードします。その後、Replication Server は変換ストアドプロシージャを実行して、ベーステーブルを更新します。こうして、Replication Server はレプリケート SAP IQ データベースにほぼリアルタイムのデータコピーを維持します。

### ライセンス

Replication Server と SAP IQ InfoPrimer の統合には特別なライセンスの要件が適用されます。『Replication Server 新機能ガイド』の「Replication Server バージョン 15.6 ESD #1 の新機能」の「ライセンス」を参照してください。

## **Replication Server と SAP IQ InfoPrimer の統合の使用**

SAP IQ InfoPrimer で Replication Server マテリアライゼーションメソッドを使用して、データを SAP IQ にマテリアライズし、プライマリデータに対して行われた更新を処理するよう Replication Server を設定します。

### 1. マテリアライゼーション前:

- SAP IQ InfoPrimer で EL (抽出、ロード) プロジェクトを作成し、[Replication Server でマテリアライゼーション] を選択します。  
EL プロジェクトエディタの [RepServer] タブで、プライマリ Replication Server とレプリケート Replication Server (プライマリと異なる場合のみ) の接続情報も指定する必要があります。SAP IQ InfoPrimer によって、[処理] タブにコマンドが追加されます。このコマンドは、変更も削除もしないでください。

各ソーステーブルに対して、SAP IQ InfoPrimer は必要なステージングテーブル定義を作成します。EL プロジェクトエディタの [テーブル] タブにある [不足している送信先テーブルを作成] アイコンを選択して、これらのステージングテーブルをレプリケート SAP IQ データベースに生成します。

---

**注意：**再マテリアライズを試みる場合は、rs\_status テーブルをクリアする必要があります。

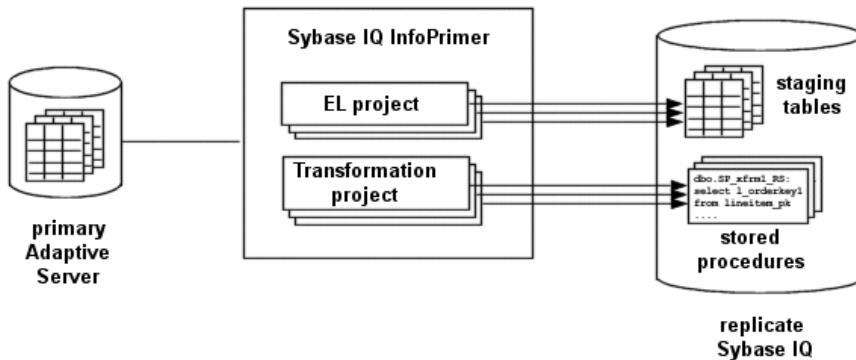
---

- SQL 変換プロジェクトを作成し、レプリケート SAP IQ データベースで生成されたステージングテーブル (挿入、更新、および削除) の各セットの変換をモデル化します。この SQL 変換プロジェクトを使用して、変換の各セットをレプリケート SAP IQ データベースでストアドプロシージャとして展開します。

---

**注意：**これらの変換ストアドプロシージャでは、オペレーションの処理が完了すると、対応するステージングテーブルがトランケートされます。

---



- Replication Server インスタンスで、**stage\_operations** 接続パラメータを使用して、EL プロジェクトで指定されたテーブルのステージオペレーションへのレプリケートデータベース接続を設定します。

---

**注意：**stage\_operations が on に設定されている場合、Replication Server は **dsi\_compile\_enable** の設定を無視し、その接続の RTL を有効にします。オペレーションはコンパイルされ、**dsi\_compile\_enable** が on に設定されると、ステージングされます。

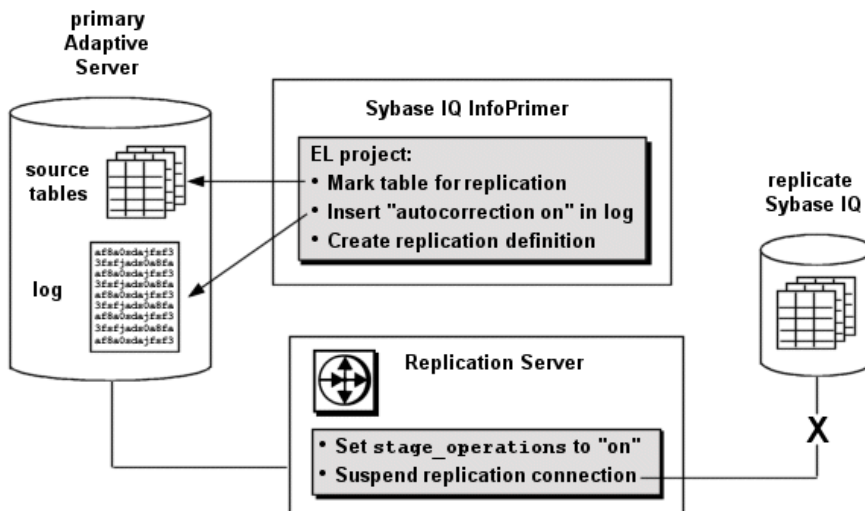
---

SAP IQ InfoPrimer で、EL プロジェクトを実行します。指定された各プライマリテーブルについて、EL プロジェクトは以下を実行します。

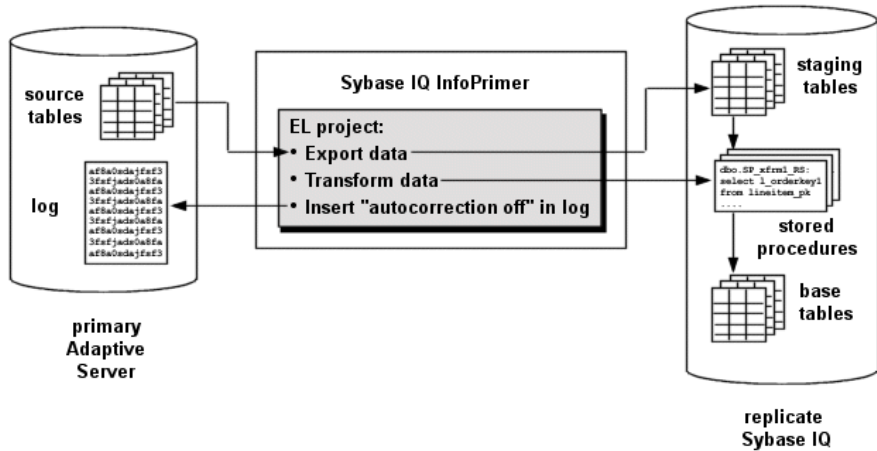
- テーブルをレプリケートするようマーク付けします。

## レプリケートデータサーバとしての SAP IQ

- b) プライマリデータベースログに autocorrection on レコードを挿入します。その結果、Replication Server レプリケートデータベースコネクションがサスペンドされます。
- c) RSSD でテーブル複写定義を作成します。

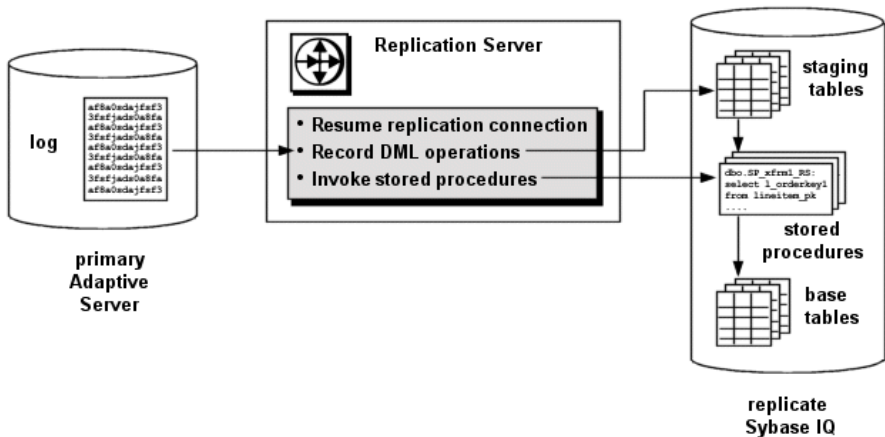


3. SAPIQ InfoPrimer EL プロジェクトは、各テーブルのプライマリデータをレプリケート SAP IQ 上にある対応するステージングテーブルにエクスポートし、変換ストアドプロシージャを実行し、autocorrection off レコードをプライマリデータベースログに挿入します。



4. Replication Server のレプリケートデータベース接続が再開され、マーク付けされたプライマリデータベーステーブルにさらに加えられる変更があれば、Replication Server がレプリケート SAP IQ データベースでステージングテーブルと変換ストアドプロシージャを使用して、それらの変更を処理します。

**注意：** SAP IQ InfoPrimer は、データの移行と、ステージングテーブルおよび変換ストアドプロシージャの作成にのみ使用されます。SAP IQ InfoPrimer は、複製には関与しません。



### ベーステーブル

ベーステーブルには、レプリケート SAP IQ データベースでの最終形式でデータが格納されます。

ベーステーブルデータの発生元は、次のとおりです。

- SQL 変換 - オペレーションをステージングするよう Replication Server のレプリケートデータベース接続が設定されると、ステージテーブルに対して実行される変換ストアドプロシージャの結果がベーステーブルに書き込まれます。
- レプリケーション - あるテーブルがステージングから除外されると、Replication Server はそのステージングテーブルをバイパスしデータをベーステーブルに直接レプリケートします。

### ステージングテーブル

プライマリテーブルのためにログされるオペレーションをステージングするよう Replication Server のレプリケートデータベース接続を設定すると、これらのオペレーションは可能な場合はコンパイルされ、レプリケート SAP IQ データベースにあるステージテーブルに書き込まれます。

ステージングする各テーブルに対して、3つのステージングテーブルが存在し、各ステージングテーブルは DELETE、INSERT、および UPDATE の各オペレーションに対応します。

- *owner\_table\_name\_DELETE\_RS*
- *owner\_table\_name\_INSERT\_RS*
- *owner\_table\_name\_UPDATE\_RS*

ここで、*owner*と *table\_name*は対応するプライマリデータベーステーブルの所有者と名前です。これらのテーブルの名前は、EL プロジェクトによって生成され、変更できません。

---

**注意：**EL プロジェクトの [テーブル] タブには、insert ステージングテーブルのみ表示されます。しかし、[テーブル作成] ウィンドウには、指定されたプライマリデータベーステーブルに対応する 3つのステージングテーブルがすべて表示されます。

---

SAP IQ InfoPrimer の EL プロジェクトでどのプライマリデータベーステーブルがステージングされるのかを識別する必要があります。ステージングからレプリケートテーブルを選択的に除外することもできます。ステージングから除外されたテーブルには、対応するステージングテーブルを作成する必要がないため、データはレプリケート SAP IQ データベースでプライマリテーブルからレプリケートテーブルに複写されます。

テーブルをステージングするようレプリケートデータベース接続を設定しても、レプリケート SAP IQ データベースにステージングテーブルが存在しない場合は、

レプリケートデータベース接続はサスペンドされます。複写定義に identity カラムとして宣言されるカラムが含まれる場合でも、これらのカラムは対応するステージングテーブルでは identity カラムとして宣言されません。

#### テーブルのコンパイル

コンパイルできないテーブルでは、コンパイルは実行されません。テーブルがコンパイルできないと見なされるのは、RTL が無効か、ファンクション文字列が変更されているか、または最小カラムレプリケーションが有効の場合です。コンパイルできないテーブルに対するオペレーションは、順番リストに取得され、コンパイル完了後に対応するレプリケートテーブルに適用されます。

---

**注意：** Replication Server がステージングされたオペレーションをコミットした後、変換ストアドプロシージャが対応するステージングテーブルをトランケートします。したがって、Replication Server の `rs_subcmp` ユーティリティを使用してステージングテーブルを確定化しないでください。

---

#### insert ステージングテーブルの構造

対応する複写定義によって適用される変更とフィルタリングを除き、insert ステージングテーブルにはプライマリテーブルと同じ数のカラムと同じカラム名が格納されます。

#### delete ステージングテーブルの構造

delete ステージングテーブルには、対応する複写定義で指定されたプライマリキーカラムのみが格納されます。

複写定義でプライマリキーが指定されない場合、delete ステージングテーブルには、以下を除くパブリッシュ済みカラムがすべて格納されます。

- 概数値カラム
- 暗号化カラム
- Java カラム
- LOB カラム

---

**注意：** テーブル複写定義でプライマリキーを指定して、処理を単純化しパフォーマンスを向上させることをおすすめします。

---

#### update ステージングテーブルの構造

update ステージングテーブルには、対応する複写定義で指定されたプライマリキーカラムごとに2つのカラムが格納されます。1つは変更前のカラムデータ用、もう1つは変更後のカラムデータ用です。

update ステージングテーブルには、複写定義で指定された各非プライマリキーカラムのカラムも格納されます。これらの非プライマリキーカラムのデータに変更が加えられたかどうかを追跡するため、update ステージングテーブルには1個ま

## レプリケートデータサーバとしての SAP IQ

たは複数のビットマップカラムが格納されます。各ビットマップカラムは `int` 型であるため、32 個の非プライマリキーカラムを追跡できます。値 1 はダーティビットであり、データがそのビット位置に対応するカラムで変更されたことを示します。

---

**注意：** `update` ステージングテーブルの変更前カラムとビットマップカラムは、SAP IQ InfoPrimer の SQL 変換プロジェクトには表示されません。

---

### 変換ストアードプロシージャ

ステージングされるどのプライマリデータベーステーブルにも、対応する変換ストアードプロシージャがレプリケート SAP IQ データベースに存在します。

Replication Server は、ステージングテーブルに対してこれらのストアードプロシージャを実行し、結果はベーステーブルに書き込まれます。

SAP IQ InfoPrimer の SQL 変換プロジェクトでこれらのストアードプロシージャによって実行される変換を指定し、それらのストアードプロシージャをレプリケート SAP IQ データベースに展開する必要があります。

レプリケート SAP IQ データベースに存在しないストアードプロシージャを使用しようとするか、またはストアードプロシージャの実行に失敗すると、レプリケートデータベース接続がサスペンドされます。

---

**注意：** SQL 変換プロジェクトに関与するテーブルをすべて確認できるように、ストアードプロシージャをレプリケート SAP IQ データベースに展開する準備が整うまでは、SQL 変換プロジェクトのプロジェクトプロパティでスキーマを選択しないでください。

---

## パラメータ

Replication Server では、**`stage_operations`** パラメータと **`dsi_stage_all_ops`** パラメータを使用して、テーブルステージングを制御します。

### stage\_operations

**`create connection`** コマンドまたは **`alter connection`** コマンドの **`stage_operations`** パラメータを設定すると、Replication Server は指定したコネクションのステージングテーブルにオペレーションを書き込みます。

レプリケートデータベースコネクションのステージングを設定できます。例：

```
create connection to SYDNEY_IQ_RS.iq_db
using profile rs_ase_to_iq;standard
set username pubs2_maint
set password pubs2_maint_pw
set stage_operations to "on"
```



個々のテーブルについてステージングの有効または無効を指定するには、特定のレプリケートテーブルに関して **alter connection** コマンドの **stage\_operations** パラメータを使用します。例：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set stage_operations to "off"
```

この場合、Replication Server は lineitem\_5 テーブルのオペレーションをステージングしませんが、代わりにオペレーションを通常どおり複製します。

**注意：** **stage\_operations** パラメータは、SAP IQ レプリケートへの接続にのみ設定できます(ここで、**dsi\_dataserver\_make** パラメータは iq に設定されます)。SAP IQ 接続プロファイルを使用して接続を作成すると、**dsi\_dataserver\_make** 接続パラメータが適切に設定されます。

### *dsi\_compile\_enable*

**stage\_operations** が on に設定されている場合、Replication Server は **dsi\_compile\_enable** の設定を無視し、その接続の RTL を有効にします。オペレーションはコンパイルされ、**dsi\_compile\_enable** が on に設定されると、ステージングされます。

### **dsi\_stage\_all\_ops**

**alter connection** コマンドの **dsi\_stage\_all\_ops** パラメータを使用して、指定したテーブルのオペレーションコンパイルを回避します。

緩やかに変化する次元 (SCD) のテーブルなどのように、テーブル履歴を保存する必要がある場合、**dsi\_stage\_all\_ops** を on に設定します。例：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set dsi_stage_all_ops to "on"
```

## **Replication Server のコンポーネント**

Replication Server では、SAP IQ InfoPrimer との統合をサポートするために追加のコンポーネントが必要です。

### **rs\_status** テーブル

**rs\_status** テーブルは、マテリアライゼーションの進行状況に関する情報を格納します。

| カラム    | データ型          | 説明                 |
|--------|---------------|--------------------|
| schema | varchar (255) | マテリアライズされるテーブルの所有者 |

| カラム       | データ型          | 説明   |
|-----------|---------------|--|
| tablename | varchar (255) | マテリアライズされるテーブルの名前  |
| action    | varchar (1)   | <ul style="list-style-type: none"> <li>• I - 初回ロード</li> <li>• A - オートコレクションフェーズ</li> <li>• R - レプリケーション</li> </ul> |
| starttime | timestamp     | アクションが開始された時間  |
| endtime   | timestamp     | アクションが完了した時間   |
| status    | varchar (1)   | <ul style="list-style-type: none"> <li>• P - アクションが進行中</li> <li>• X - 実行が完了した</li> <li>• E - 実行エラー</li> </ul>      |
| pid       | int           | 予約済み   |

たとえば、my\_table のオートコレクションが進行中の場合、rs\_status には次のようなローが含まれます。

```

schema tablename action starttime                endtime status pid
-----
sys      my_table  A      2011-07-11 19:11:25.531                P

```

my\_table のオートコレクションが完了すると、rs\_status には次のようなローが含まれます。

```

schema tablename action starttime
-----
sys      my_table  A      2011-07-11 19:11:25.531

endtime                status pid
-----
2011-07-11 19:12:14.326 X

```

rs\_status データの自動クリーンアップはありません。テーブルの再マテリアライズを試みる前に、rs\_status から対応するローを削除する必要があります。

```
delete rs_status where tablename=tablename and schema=owner
```

### オートコレクション関数

Replication Server では、rs\_autoc\_on、rs\_autoc\_off、および rs\_autoc\_ignore 関数を使用して、rs\_status テーブルを更新します。

『Replication Server リファレンスマニュアル』の「Replication Server システム関数」を参照してください。

### システム変数

**rs\_autoc\_on** 関数と **rs\_autoc\_off** 関数は、rs\_status テーブルの更新時に 2 つのシステム変数を使用します。

- *rs\_deliver\_as\_name* - オートコレクションの影響を受けるレプリケートテーブルの名前を指定します。
- *rs\_repl\_objowner* - オートコレクションの影響を受けるレプリケートテーブルの所有者を指定します。

## デフォルトのデータ型変換

SAP IQ では、すべての Adaptive Server データ型をネイティブ形式でサポートしているため、Adaptive Server から SAP IQ へのデータ型変換は必要ありません。

## サポートされない機能

Replication Server と SAP IQ InfoPrimer の統合は、特定の機能とプラットフォームに限定されています。

Replication Server と SAP IQ InfoPrimer の統合では、以下はサポートされていません。

- SAP IQ 以外のレプリケートデータベース
- Adaptive Server 以外のプライマリデータベース
- 複写ストアプロシージャ
- カスタムファンクション文字列
- RTL によって提供されたもの以外の事前ステージングオペレーション変換
- レプリケート SAP IQ データベースで変換ストアプロシージャによって実行された変換後に行われる変換



# レプリケートデータサーバとしての SAP HANA データベース

SAP 複写システムにおける、SAP HANA データベースデータサーバに固有のレプリケートデータベースの問題と考慮事項について説明します。

## SAP HANA データベースのレプリケートデータサーバ

---

ExpressConnect for SAP HANA データベースを使用して、SAP HANA データベースデータサーバに複写できます。

ExpressConnect for SAP HANA データベースは、Replication Server と SAP HANA データベース間の直接通信を可能にします。

## SAP HANA データベースでの複写の干渉と影響

---

Replication Server は、レプリケート SAP HANA データベースインスタンスにオブジェクトを作成して、Replication Server の操作をサポートします。

作成されるオブジェクトは次のとおりです。

- `rs_info`。レプリケートデータベースによって使用されるソート順と文字セットに関する情報を格納します。レプリケートデータベースのソート順と文字セットは、`rs_info` テーブルに記録する必要があります。

---

**注意：** このテーブルに対する **INSERT** 文で、使用している SAP HANA データベースデータサーバに、適切な文字セットとソート順が指定されていることを確認してください。接続プロファイルで `rs_info` が作成されると、テーブルエントリはデフォルトで UTF-8 に設定されます。Unicode データを複写するには、UTF-8 を使用するように Replication Server を設定する必要があります。

---

Replication Server の `rs_get_charset` および `rs_get_sortorder` ファンクションは、レプリケートデータベース内の `rs_info` テーブルから文字セットとソート順を取得します。

- `rs_lastcommit`。レプリケートデータベースに適用される複写トランザクションに関する情報を格納するテーブルです。`rs_lastcommit` 内の各ローは、プライマリデータベースからレプリケートデータベースに配信されコミットされた、最後のトランザクションを示します。Replication Server は、この情報によってすべてのトランザクションが配信されたことを確認します。

Replication Server の **rs\_get\_lastcommit** ファンクションは、レプリケートデータベース内の最後にコミットされたトランザクションに関する情報を取得します。ASE以外のレプリケートデータベースについては、**rs\_get\_lastcommit** ファンクションは、データベース固有のファンクション文字列クラスで、レプリケートデータベース内の **rs\_lastcommit** テーブルへのアクセスに必要なクエリによって置き換えられます。

- **rs\_ticket\_history**。Replication Server コマンド **rs\_ticket** の実行結果を格納します。Replication Server は **rs\_ticket\_seq** という名前のシーケンスを作成します。Replication Server はこのシーケンスに基づいて、**rs\_ticket\_history** に一意のロー ID を設定します。プライマリデータベースに対して **rs\_ticket** コマンドを発行し、コマンドがプライマリデータベースからレプリケートデータベースまで移動するために要する時間を測定できます。この情報を使用して、Replication Server のパフォーマンス、モジュールのハートビート、複製の正常性、テーブルレベルのクワイズをモニタできます。**rs\_ticket** の各実行結果は、レプリケートデータベース内にある **rs\_ticket\_history** テーブルの 1 つのローに保存されます。**rs\_ticket\_history** テーブルの各ローを問い合わせると、個々の **rs\_ticket** 実行結果を取得したり、別のローの結果と比較できます。データは、手動でトランケートできます。

---

**注意：** **rs\_ticket\_history** テーブルは、Replication Server バージョン 15.1 以降でのみ使用できます。

---

- **rs\_status**。直接ロードマテリアライゼーションの進行状況に関する情報を格納します。『Replication Server リファレンスマニュアル』 > 「Replication Server システムテーブル」 > 「rs\_status」を参照してください。
- **rs\_ticket\_seq**。 **rs\_ticket\_history** に対して一意のロー ID を設定するために、Replication Server が使用します。
- **rs\_update\_lastcommit**。 **rs\_lastcommit** テーブルを更新するために使用されるストアードプロシージャです。
- **rs\_version**。複製システムのバージョン番号情報を格納する Replication Server のシステムテーブルです。

### プライマリキーのデータ型としての Oracle RAW および LONG RAW

プライマリキーまたはプライマリキーの一部として、Oracle RAW または LONG RAW データ型のカラムを使用しないでください。

## Oracle の TIMESTAMP WITH TIME ZONE データ型のマッピング

Oracle の `TIMESTAMP WITH TIME ZONE` データ型を SAP HANA データベースに複製するには、`map to char` 句を使用してテーブル複製定義を明示的に作成する必要があります。

## Oracle から SAP HANA データベースへの連続複製の制限

カラムが `NULL` に設定されている場合は連続複製が失敗します。

デフォルトでは、Oracle プライマリデータベースは最低限のロギングを行うため、**UPDATE** 文ですべてのカラムが指定されるわけではありません。**UPDATE** 文で指定されていないカラムは、**where** 句を含むサブスクリプションで実行されるローマイングレーションの結果として自動的に `NULL` に設定されます。**where** 句で指定されたカラムの更新時に直接ロードマテリアライゼーションに成功しても、カラムが `NULL` に設定されるので連続複製には失敗します。

## Identity カラム

SAP HANA データベースには `identity` 型はありません。

プライマリのすべての `identity` カラムが必ず SAP HANA データベースの `numeric` 型にマップするようにします。「データ型の変換とマッピング」を参照してください。

## SAP HANA データベースのレプリケートデータベースのパフォーマンス

レプリケートデータベースにトランザクションを適用するために、Replication Server では Replication Server の **create connection** コマンドで指定するメンテナンスユーザ ID が必要となります。

メンテナンスユーザ ID は SAP HANA データベースデータサーバで定義され、レプリケートデータベースにトランザクションを適用する権限が付与されている必要があります。メンテナンスユーザ ID には、次のようなスキーマ権限が必要です。

- **CREATE ANY** - スキーマに、テーブル、ビュー、シーケンス、同義語、SQL スクリプト関数、またはデータベースプロシージャを作成することをユーザに許可します。
- **DELETE**、**DROP**、**EXECUTE**、**INDEX**、**INSERT**、**SELECT**、および **UPDATE** - 指定されたスキーマに格納されたすべてのオブジェクトに対して付与されます。

## 異機種間複写環境における DDL 複写

---

レプリケートが SAP HANA データベースである異機種間複写環境において、Replication Server は複数のデータ定義言語 (DDL) コマンドの複写をサポートしています。

Replication Server では、プライマリデータベースが Oracle、Microsoft SQL Server、または IBM DB2 UDB の場合、SAP HANA データベースへの次の DDL コマンドの複写をサポートしています (SAP ASE はプライマリデータベースとしてサポートされていません)。

- **create table**
- **alter table (add column または drop column)**
- **rename table**
- **drop table**

データベース複写定義が **auto\_update\_table\_list** パラメータまたは **auto\_extend\_table\_list** パラメータを指定して定義されている場合、Replication Server はデータベース複写定義を自動的に変更して、データベースレベルで複写対象としてマーク付けされたデータベースで発生したすべての DDL の変更を反映します。たとえば、あるデータベースが複写対象としてマーク付けされている場合、そのデータベースにテーブルが追加されると、その新しく追加されたテーブルに対して自動的に複写が確立されます。

次の場合、作成、変更、名称変更、または削除されたテーブルは、レプリケートテーブル所有者を除き、テーブル複写定義では正規化されません。

- レプリケートテーブル所有者がテーブル複写定義で定義されており、かつ
- レプリケートサイトがテーブル複写定義をサブスクライブしている。

---

**注意：** 異機種間複写環境では、プライマリデータ型が対応するレプリケートデータ型にマッピングされます。

---

参照:

- 『異機種間複写ガイド』の「データ型の変換とマッピング」
- 『リファレンスマニュアル』の「create database replication definition」



## テーブル複写定義の排除

---

異機種間環境でデータベース複写定義とサブスクリプションを使用する場合、テーブル複写定義を使用する必要はありません。

テーブル数の多いデータベースを含む複写環境ですべてのテーブル複写定義を管理するには多くの手間がかかります。DDL 複写では、テーブル複写定義との調整なしに DDL でテーブル定義を変更すると、データに不整合が生じるおそれがあります。データベース複写定義とサブスクリプションを使用すれば、データの不整合を回避しながら、テーブルごとにテーブル複写定義を使用する必要性をなくすことができます。

## ユニークキーがないテーブルの LOB カラムの複写

---

サポートされているデータベースから SAP HANA データベースへの複写時に、Replication Server は、ユニークキーがないテーブルの LOB カラムの複写をサポートしています。

ユニークキーがなく、LOB カラムがあるテーブルの場合、LOB カラムのステータスを **replicate\_if\_changed** に設定する必要があります。

Replication Agents for Oracle、Microsoft SQL Server、または IBM DB2 UDB の場合、デフォルトですべての LOB カラムのステータスが **replicate\_if\_changed** に設定されています。

SAP Adaptive Server Replication Agent Thread の場合、デフォルトでは **sp\_setreptable** によって複写ステータスが **always\_replicate** に設定されています。 **sp\_setrepcol** を使用して、LOB カラムの複写ステータスを **replicate\_if\_changed** に設定する必要があります。

参照:

- 『リファレンスマニュアル』の「sp\_setrepcol」
- 『管理ガイド 第1巻』の「ユニークキーがないテーブルの複写」

## ExpressConnect for SAP HANA データベースと SAP HANA データベースに対するレプリケートデータベースのコネクティビティ

---

Replication Server では、ExpressConnect for SAP HANA データベースを使用して SAP HANA レプリケートデータベースに接続できます。

### *ExpressConnect for HANA データベースの使用*

Replication Server データベースから SAP HANA データベースへの接続には次の 2 種類があります。

- セキュア。この接続ではデータベース接続で指定された **hdbuserstore** キーを使用します。
- 標準。この接続では、ログインにユーザ ID、パスワード、およびホスト名が必要です。標準接続では、`interfaces` ファイルにエントリが含まれていることが必要です。

---

**注意：** ExpressConnect for SAP HANA データベースでは、Red Hat Enterprise Linux 6 と SuSE Linux Enterprise Server (SLES) 11 のみがサポートされています。

---

SAP HANA データサーバの `interfaces` ファイルエントリの一般的なフォーマットは次のとおりです。

```
[dataservername]
master tcp ether hostname port
query tcp ether hostname port
```

ここで、`hostname` と `port` は SAP HANA データサーバのホストとポート番号、`dataservername` はホストとポート番号を識別するためのラベルです。`interfaces` ファイルは、TLI (トランスポートレイヤインタフェース) ではなく標準フォーマットでなければなりません。

各 SAP HANA データベースは一意的インスタンス番号とともにインストールされます。ポート番号は、インスタンス番号の前と後にそれぞれ 3 と 15 が付加されたものです。ここで、

```
3in15
```

`in` は 2 桁のインスタンス番号です。たとえば、インスタンス番号 1 の SAP HANA データベースのポート番号は 30115 になります。

## ExpressConnect for SAP HANA データベースの設定

ExpressConnect for SAP HANA データベースで、SAP HANA データベースと Replication Server 間の接続を設定します。

### 前提条件

UNIX および Linux システムでは、SAP HANA データベースの ODBC ドライバに、\$HOME 環境変数によって参照されるディレクトリに対する書き込み許可があることを確認します。

ExpressConnect for SAP HANA データベースには、必要な SAP HANA データベース ODBC ドライバが付属していません。Replication Server のインストール後、ExpressConnect for SAP HANA データベースを使用する前に、これらのライブラリを SAP Support MarketPlace からダウンロードしてインストールしてください。『Replication Server リリースノート』>「インストールに関する特別な指示」>「ExpressConnect for SAP HANA データベース用 ODBC ライブラリのインストール」を参照してください。

### 手順

1. Replication Server から接続するために使用する SAP HANA データベースのメンテナンスマスター ID とパスワードを決定します。「SAP HANA データベースのレプリケートデータベースのパーミッション」を参照してください。

---

**注意：** SAP HANA データベースへの接続を作成する前に、このメンテナンスマスター ID が SAP HANA データベースにログインできることと、ユーザー ID に SAP HANA データベースにテーブルを作成するパーミッションがあることを確認します。

---

2. **isql** を使用して、メンテナンスマスター ID とパスワードの組み合わせを使用した Replication Server から SAP HANA データベースへの接続を作成します。たとえば、標準接続と Oracle プライマリデータベースの場合は、次を実行します。

```
create connection to
<hana_server>.<hana_db>
using profile rs_oracle_to_hana;ech
set username <userid>
set password <password>
```

各パラメータの意味は、次のとおりです。

- *hana\_server* は、interfaces ファイル内のレプリケート SAP HANA データベースデータサーバを特定する大文字と小文字が区別される名前（標準接続）、または **hdbuserstore** キー（セキュア接続）です。
- *hana\_db* は、SAP HANA データベースインスタンスの ID です。この値は Replication Server のみで使用され、接続には使用されません。

---

**注意：** SAP HANA データベースは、Adaptive Server のように複数のデータベースをサポートしません。

---

セキュア接続と Oracle プライマリデータベースの場合は、次を実行します。

```
create connection to
<hana_server>.<hana_db>
using_profile rs_oracle_to_hana;ech
set username <userid>
set password <password>
set dsi_connector_sec_mech to "hdbuserstore"
```

ここで **dsi\_connector\_sec\_mech** では、DSI コネクタのセキュリティメカニズムを指定します。

『Replication Server リファレンスマニュアル』の「**create connection**」を参照してください。

3. 必要に応じて rs\_info テーブルを更新します。

```
update rs_info
set rsval = 'utf8'
where rskey = 'charset_name'
set rsval = 'bin_utf8'
where rskey = 'sortorder_name'
```

## **ExpressConnect for SAP HANA データベースのライセンス**

ExpressConnect for SAP HANA データベースのライセンスは、Replication Server によって配布されます。

『Replication Server インストールガイド』>「インストールの計画」>「SPDC または SMP からのライセンスの取得」を参照してください。

## **トレースとデバッグ**

Replication Server のトレースオプションを有効にすると、コネクタレベルと接続レベルの診断情報を収集できます。

ExpressConnect for SAP HANA データベースの実行に関する診断情報は、コネクタレベルと接続レベルの両方のオペレーションと、さまざまな診断条件に関して利用できます。すべての条件がコネクタレベルと接続レベルの両方のトレースに対応しているわけではありません。ExpressConnect for SAP HANA データベース実行プログラムの診断バージョンの使用が必要な場合もあります。

ExpressConnect for SAP HANA データベースのトレースを有効にする場合は、『ExpressConnect for Oracle Configuration Guide』>「Configuring ExpressConnect for Oracle」>「Trace and Debug」を参照してください。

## 文字セットの変換

ExpressConnect for SAP HANA データベースでは、Replication Server の **rs\_charset**、**unicode\_format**、および **dsi\_charset\_convert** パラメータのデフォルト値に基づいて文字セットが変換されます。

**rs\_charset** が `iso_1` に設定され、**unicode\_format** が `string` に設定され、**dsi\_charset\_convert** が `on` に設定されている場合、ExpressConnect for SAP HANA データベースは、変換のないネイティブフォームですべてのデータを渡します。Unicode リテラルはサポートされていません。

Unicode を複写するには、**rs\_charset** を `utf8` に設定します。このように設定すると、Unicode リテラルがサポートされます。

---

**注意：** Unicode カラムタイプ以外の Unicode 複写では、**rs\_charset** を `utf8` に設定する必要があります。

---

**rs\_charset** がサポート対象の文字セット、**unicode\_format** が `string` (デフォルト)、**dsi\_charset\_convert** が `on` に設定されている場合、ExpressConnect for SAP HANA データベースではすべての SQL 言語と文字のパラメータデータを UTF-16 に変換します。

## LOB ポインタファンクション

ExpressConnect for Oracle および ExpressConnect for SAP HANA データベースでは、LOB データの管理に LOB ポインタが使用されません。その結果、LOB ポインタの管理に使用される Replication Server システムファンクションは、ExpressConnect for Oracle または ExpressConnect for SAP HANA データベースで使用できません。

このようなファンクション (**rs\_get\_textptr**、**rs\_textptr\_init**、**rs\_writetext** など) は ExpressConnect で認識できますが、ファンクションを使用しても Replication Server から無視されます。

## SAP HANA データベースのレプリケートデータベースの設定

Replication Server の異機種データ型サポート (HDS) 機能には、レプリケート Replication Server と SAP HANA データベースのレプリケートデータベースで HDS 機能を設定するために使用できる設定情報が用意されています。

この設定情報は、インストールにおいて、接続プロファイルの一部として提供されます。

- Replication Server インストール:
  - ファンクション文字列、エラークラス、ユーザ定義データ型の作成

## レプリケートデータサーバとしての SAP HANA データベース

- 接続プロファイル:
  - RSSD へのクラスレベルデータ型変換の適用
  - SAP HANA データベースのレプリケートデータベースでのオブジェクトの作成
  - 接続プロパティの設定  
ExpressConnect for SAP HANA データベースを使用して接続できます。接続プロファイルのバージョンまたはオプション名は“ech”でなければなりません。
- 追加設定
  - ExpressConnect の設定
  - コマンドのバッチ処理の設定
  - 動的 SQL の設定

## Replication Server インストール

Replication Server インストールでは、SAP HANA データベースへの複写をサポートするために必要なファンクション文字列とクラスが自動的にインストールされます。

### ファンクション文字列、エラークラス、ユーザ定義データ型

ファンクション文字列は、Replication Server のデフォルトの `rs_hanadb_function_class` に追加されます。

ファンクション文字列クラスは、いくつかのデフォルト Replication Server ファンクション文字列を、SAP HANA データベースデータサーバと通信し、複写関連のテーブルとプロシージャにアクセスするように設計されたカスタムファンクション文字列に置き換えます。

---

**警告！** ExpressConnect for SAP HANA データベースでは、`text` 型と `image` 型の処理にデフォルトの Replication Server ファンクション文字列を使用する LOB カラムの複写をサポートしています。ExpressConnect for SAP HANA データベースでは、`text` 型と `image` 型の処理にカスタムファンクション文字列はサポートしていません。デフォルトの Replication Server の LOB の処理を、`rs_writetext`、`rs_textptr_init` のカスタムファンクション文字列か他の LOB 関連ファンクション文字列で上書きすると、`text` 型と `image` 型の処理は失敗します。

---

## 接続プロファイル

接続プロファイルを使用すると、事前に定義されたプロパティのセットで接続を設定できます。

構文

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
set password [to] password
[other_create_connection_options]
[display_only]
```

### パラメータ

*data\_server* - 次のいずれかです。

- SAP HANA データベースのホストおよびポート番号のリストを表示する *interfaces* ファイルエントリへの参照
- SAP HANA データベースの SAP Secure User Store エントリのキー名

*database* - 複写システムに追加するデータベースです。

*connection\_profile* - 接続の設定、RSSD の修正、およびレプリケートデータベースオブジェクトの作成に使用する接続プロファイルを示します。

*version* - 使用する接続プロファイルのバージョンを指定します。

*user* - データベースの Replication Server メンテナンスユーザのログイン名です。Replication Server は、複写データを管理するのにこのログイン名を使用します。ネットワークベースのセキュリティを有効に設定していない場合には、ユーザ名を指定する必要があります。

*other\_create\_connection\_options* - プロファイルで指定されない接続オプションの設定、またはプロファイルで指定されているオプションの上書き (Replication Server に用意されているファンクション文字列クラスを上書きするカスタムファンクション文字列クラスの指定など) を行うには、他の **create connection** オプションを使用します。 **create connection** コマンドの他のオプションのリストについては、『Replication Server リファレンスマニュアル』 > 「Replication Server コマンド」 > 「**create connection**」を参照してください。

*display\_only* - **display\_only** は、**using profile** 句とともに使用し、実行されるコマンド、およびそのコマンドを実行するサーバの名前を表示します。 **display\_only** を使用した結果については、クライアントログおよび Replication Server ログを参照してください。

### RSSD へのクラスレベルデータ型変換

クラスレベル変換によって、プライマリデータ型とデータの変換先のレプリケートデータ型が特定されます。

クラスレベル変換は、以下に示す適切な名前前の接続プロファイルによって SAP HANA データベースのレプリケートデータベースに提供されます。

## レプリケートデータサーバとしての SAP HANA データベース

- `rs_ase_to_hanadb` - Adaptive Server データ型を SAP HANA データベースデータ型に変換します。
- `rs_msss_to_hanadb` - Microsoft SQL Server データ型を SAP HANA データベースデータ型に変換します。
- `rs_oracle_to_hanadb` - Oracle データ型を SAP HANA データベースデータ型に変換します。
- `rs_udb_to_hanadb` - DB2 UDB データ型を SAP HANA データベースデータ型に変換します。

SAP HANA データベース複製環境に対して Adaptive Server Enterprise (ASE) の ExpressConnect for SAP HANA データベースバージョンのプロファイルを使用したスクリプトの例:

```
create connection to hana_server.hana_db
using profile rs_ase_to_hana;ech
set username rs_maint_user
set password rs_maint_user_pwd
go
```

### SAP HANA データベースのレプリケートデータベースでのオブジェクトおよび接続プロパティ

接続プロファイルによって、`rs_ticket_seq` シーケンスと、レプリケートデータベース内の `rs_info`、`rs_lastcommit`、`rs_status` および `rs_ticket_history` テーブルが作成されます。

接続プロファイルによって次の接続プロパティが設定されます。

```
set error class rs_hanadb_error_class
set function string rs_hanadb_function_class
```

### SAP Secure User Store

ExpressConnect for SAP HANA データベースなどの ASE 以外および IQ 以外のコネクタで、SAP Secure User Store を使用します。

SAP Secure User Store のログインをサポートするには、**alter connection** コマンドまたは **create connection** コマンドで `dsi_connector_sec_mech` パラメータを使用します。SAP Secure User Store のログインを使用するときは、SAP HANA データベースインスタンスへの接続に `interfaces` ファイルのエントリは必要ありません。

`dsi_connector_sec_mech` パラメータはネットワークベースのセキュリティパラメータではありません。このパラメータは、接続でのみ有効です。

ExpressConnect for SAP HANA データベースで `dsi_connector_sec_mech` を使用するには、まず **hdbuserstore** ユーティリティを使用して、暗号化された資格のセキュアなユーザストアを作成する必要があります。たとえば次のように使用します。

```
hdbuserstore set hanads myhost:30215 myuser mYpA5Sw0rD
```



ここで "hanads" はセキュアなユーザストアに問い合わせを行うためのキーとして使用するラベル、"myhost:30215" は接続環境のホスト名とポート番号、"myuser" はユーザ ID、"mYpA5Sw0rD" はパスワードです。

---

**注意：**セキュアなストアは、Replication Server を開始し実行しているオペレーティングシステムのユーザ ID と同じ ID で作成する必要があります。それ以外の ID を使用すると、Replication Server はセキュアなユーザストアにアクセスできません。

---

セキュアなユーザストアを作成したら、暗号化された資格で SAP HANA データベースインスタンスへの接続を作成できます。たとえば、ASE から SAP HANA データベースへの複写を行うために SAP HANA データベースインスタンスに接続するには、次を実行します。

```
create connection to hanads.hanadb
using profile rs_ase_to_hanadb;ech
set username "foo"
set password "bar"
set dsi_connector_sec_mech to "hdbuserstore"
go
```

ここで、ユーザ ID の "foo" とパスワードは "bar" は、**create connection** コマンドの構文を満たすためだけに指定される未使用の値です。

---

**注意：**Replication Server を開始したオペレーティングシステムユーザが、セキュアなユーザストアも所有している必要があります。

---

**dsi\_connector\_sec\_mech** を使用するように、SAP HANA データベースインスタンスに対する既存の接続を変更できます。次に例を示します。

```
alter connection to hanads.hanadb
set dsi_connector_sec_mech to "hdbuserstore"
go
```

**alter connection** コマンドの実行後に、接続の保留と再開を実行する必要があります。

```
suspend connection to hanads.hanadb
go
resume connection to hanads.hanadb
go
```

## 追加設定

複写をサポートするための追加設定について説明します。

設定内容は次のとおりです。

- コマンドのバッチ処理の設定
- 動的 SQL の設定

- HVAR の設定

#### コマンドバッチ

SAP HANA データベースでは、コマンドバッチはサポートされません。SAP HANA データベース接続では、コマンドバッチを有効にしないでください。

#### 動的 SQL

**dynamic\_sql** 設定パラメータはデフォルトで on に設定されています。この設定は、すべての SAP HANA データベース接続プロファイルでを使用することをおすすめします。

#### HVAR

Advanced Service Option に対する Replication Server ライセンスを持っている場合は、SAP HANA データベースへの複写で Replication Server の HVAR (High-Volume Adaptive Replication) を使用できます。『Replication Server 管理ガイド 第 2 巻』 > 「パフォーマンスチューニング」 > 「Advanced Services Option」 > 「Adaptive Server への High-Volume Adaptive Replication」を参照してください。

### 複写定義での大文字と小文字および引用符付き識別子の設定

プライマリデータベースが Oracle または IBM UDB for Linux、UNIX、および Windows の場合、レプリケート SAP HANA 上でテーブルをマテリアライズするための複写定義を手動で作成しているときに、テーブルスキーマ内で引用符付き識別子を使用する場合は、いくつかのルールに従う必要があります。

プライマリデータベースにはデフォルトでオブジェクト名が大文字で格納されるので、次のルールに従ってください。

- 引用符なしのすべてのカラム名とテーブル名に大文字を使用します。
- プライマリカラムにすでに定義されているものと同じ名前を引用符付きカラムに使用します。
- Replication Agent の **lrl\_character\_case** 設定パラメータ値を **asis** に設定します。

## 異機種間における Multi-Path Replication

複数のレプリケーションパスを使用して、レプリケーションのスループットとパフォーマンスを向上させ、競合を低減します。

シングルパスレプリケーション環境では、トランザクションはプライマリデータベースからレプリケートデータベースに連続して複写されるため、プライマリデータベースのトランザクションのコミット順が確保され、このためレプリケートデータベースとプライマリデータベースの一貫性が保たれます。トランザクションをレプリケートデータベースに適用する逐次モードは、複数のアプリケーションがプライマリデータベースで並列してそれぞれのトランザクションが実行されていたり、複数のプライマリデータベースから到着するトランザクションがある場合でも変更されません。

同じプライマリデータベースから生じたすべてのトランザクションを直列化することなく、テーブルのサブセット内でデータの一貫性を維持できる複写環境があります。この環境の典型例は、異なるデータのセットにアクセスする異なるアプリケーションで1つのプライマリデータベースが変更される場合です。特定のアプリケーション内で変更されたテーブルのサブセット内の異なるデータのセットは、引き続き連続して複写されます。異なるテーブルのサブセットのデータは並列して複写することができます。

Multi-Path Replication™ では、さまざまなストリームを介したデータの複写をサポートすると同時に、パス内でのデータ整合性を維持しますが、さまざまなパス間でのコミット順には従いません。

レプリケーションパスは、Replication Server とプライマリデータベースまたはレプリケートデータベースの間のコンポーネントとモジュールをすべて含んでいます。Multi-Path Replication では、プライマリデータベースから1つまたは複数の Replication Server への複数の Replication Agent コネクションのために、複数のプライマリレプリケーションパスを作成できます。また、1つまたは複数の Replication Server からレプリケートデータベースへのコネクションのために、複数のレプリケートパスを作成できます。Multi-Path Replication は、ウォームスタンバイ環境と Multi-Site Availability (MSA) 環境で設定できます。トランザクションを Replication Server 間の専用ルートで伝達して、共有ルート上での輻輳を回避できます。また、プライマリデータベースから Replication Server を経由してレプリケートデータベースに至るエンドツーエンドのレプリケーションパスをオブジェクト (テーブルやストアドプロシージャなど) 専用にすることができます。

Adaptive Server を Multi-Path Replication システムのプライマリデータベースまたはレプリケートデータベースとして設定するには、『Replication Server 管理ガイド

## 異機種間における Multi-Path Replication

第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」を参照してください。

### ライセンス

Multi-Path Replication は、Advanced Services Option の一部としてライセンスされます。『Replication Server インストールガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

### システムの稼働条件

**表 2 : 異機種間における Multi-Path Replication システムでサポートされているプライマリデータベースとレプリケートデータベースのペア**

| プライマリデータベース     | レプリケートデータベース    |
|-----------------|-----------------|
| Adaptive Server | Adaptive Server |
| Adaptive Server | SAP HANA データベース |
| Adaptive Server | Oracle          |
| Adaptive Server | SAP IQ          |
| Oracle          | Adaptive Server |
| Oracle          | SAP HANA データベース |
| Oracle          | Oracle          |
| Oracle          | SAP IQ          |

**表 3 : サポートされている Multi-Path Replication データベースバージョン**

| データベース          | サポートされているバージョン  |
|-----------------|---|
| Adaptive Server | 15.7 以降   |
| SAP HANA データベース | 1.0 SP5   |
| Oracle          | Oracle 10g と 11g。Replication Server Options の『Replication Agent リリースノート』の「製品の概要」の「製品の互換性」を参照してください。 |
| SAP IQ          | 15.1 以降。『Replication Server リリースノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。                 |

## 並列トランザクションストリーム

---

Multi-Path Replication は、トランザクションが並行ストリームに分割され、異なるストリーム全体で順番にコミットされない限り、複写パフォーマンスを向上させることができます。

トランザクションを並列レプリケーションパスに分割して輻輳を削減することにより、複写のパフォーマンスを向上させることができます。トランザクション属性または派生データ値などの並列化ルールに従って、トランザクションを分割できます。たとえば、次の方法を使用できます。

- テーブルまたはストアードプロシージャなどの特定オブジェクトにパスを割り当てます。オブジェクトをパスにバインドする場合、Replication Agent はパスを介してそのオブジェクトに実行する複写可能なアクションを、複数のレプリケーションパス設定で定義する Replication Servers に送信します。Adaptive Server の RepAgent および Replication Agent for Oracle では、この複写分散モードがサポートされています。
- プライマリデータベースでクライアント接続のセッション ID 別にトランザクションを分割します。Adaptive Server の RepAgent は、クライアント接続によるトランザクションの分散をサポートしています。
- 各パスへの Replication Server を使用します。
- オブジェクトをパスにバインドするか、Replication Server 間に専用ルートを作成して、優先度の高い複写に対し専用パスまたは輻輳の少ないパスを割り当てます。

## デフォルトおよび代替接続

---

Multi-Path Replication では、接続には、デフォルトおよび 1 つ以上の代替接続が含まれます。

SAP Replication Agent からのデータを受け入れる接続はプライマリ接続で、データをデータベースに適用する接続はレプリケート接続です。デフォルトの接続または代替接続は、プライマリ接続でもレプリケート接続でもかまいません。

デフォルトの接続は、データベースを複写ドメインに追加する際に、SAP Replication Server から特定のプライマリデータベースまたはレプリケートデータベースに作成する接続です。データサーバが SAP ASE であるか、サポートされている SAP ASE 以外のサーバであるかに応じて、**rs\_init**、**create connection**、または **create connection ... using profile** を使用して、デフォルトの接続を作成できます。

デフォルトのコネクションは、データサーバ名およびデータベース名を *dataserver.database* の形式で、コネクション名として使用します。ここで *dataserver* および *database* は、それぞれ実際のデータサーバ名およびデータベース名です。

必須のデフォルトのコネクションを作成後、複数の代替コネクションを作成できます。各代替コネクションには、それぞれユニークな名前が必要です。

代替コネクションを作成後、コネクションのプロパティを変更、またはコネクションを削除できます。すべてのコネクションのステータスを表示し、そのコネクションのサブスクリプションを作成することもできます。

代替コネクションを作成する場合、ユーザ ID が有効なユーザであることが必要です。SAP IQ レプリケートデータベースへのコネクションを作成する場合、デフォルトのコネクションおよび各代替コネクションに対し、接続プロファイル、接続プロファイルのバージョン、およびユニークなメンテナンスユーザ名を指定する必要があります。

## SAP IQ に関する Interfaces ファイル要件

---

SAP IQ データサーバの *interfaces* ファイル (Windows では *sql.ini* ファイル) に、Replication Server のエントリを作成します。単一の SAP IQ サーバにレプリケートする Replication Server が複数ある場合、すべての Replication Server を *interfaces* ファイルに追加する必要があります。

複数の SAP IQ マルチプレックスノードへのコネクションを作成する場合は、関連する各ノードのエントリをレプリケート Replication Server の *interfaces* ファイルに作成してください。

## 専用ルート

---

専用ルートは、特定のプライマリコネクションのトランザクションのみを分配します。レプリケート Replication Server に専用ルートを作成して、優先順位の高いトランザクションを複製するか、特定のプライマリコネクションのために輻輳の少ないパスを維持できます。

共有ルートは、プライマリ Replication Server から発生するすべてのプライマリコネクションにトランザクションを分配する、プライマリ Replication Server とレプリケート Replication Server の間にあります。共有ルートは特定のコネクションにバインドしません。専用ルートにバインドされていないコネクションは、使用可能な任意の有効共有ルートを使用します。

専用ルートを作成できるのは、以下の条件が満たされた場合だけです。

- プライマリ Replication Server から送信先 Replication Server への共有ルートが存在し、この共有ルートが直接ルートである。Replication Server 間に間接ルートしかない場合、専用ルートは作成できません。
- 共有ルートが有効であり、サスペンドされていない。
- 共有ルートのバージョンが 1570 またはそれ以降である。

## 専用ルートの作成

**create route** および **with primary at** 句を使用して、専用ルートを作成します。

たとえば、NY\_DS.pdb1 プライマリコネクションのために、RS\_NY プライマリ Replication Server と RS\_LON レプリケート Replication Server との間の専用ルートを作成するには、RS\_NY で次のように入力します。

```
create route to RS_LON
with primary at NY_DS.pdb1
go
```

特定のコネクションのために専用ルートを作成すると、そのコネクションから送信先 Replication Server へのトランザクションはすべて、その専用ルートを通るようになります。

## 専用ルートを管理するコマンド

**create route**、**drop route**、**resume route**、および **suspend route** を使用して、専用ルートを管理およびモニタします。

コマンドに **with primary at *dataserver.database*** 句を含めると、専用ルートを指定できます。ここで、*dataserver.database* は、プライマリコネクションの名前です。

『Replication Server リファレンスマニュアル』の「Replication Server コマンド」で、「**create route**」、「**drop route**」、「**suspend route**」、および「**resume route**」を参照してください。

## 異機種間における Multi-Path Replication

| コマンド                | 構文  | コマンドおよびパラメータの変更   |
|---------------------|---|---|
| <b>create route</b> | <pre>create route to dest_replication_server {   with primary at dataserver.   database     set next site [to] thru_replication_server     [set username [to] user]   [set password [to] passwd]   [set route_param to 'value'     [set route_param to 'value']... ]   [set security_param to 'value'     [set security_param to 'value']... ]}</pre> | <p>専用ルートの作成時にユーザ ID を使用する場合、ユーザ ID は有効なユーザである必要があります。</p> |



| コマンド                 | 構文  | コマンドおよびパラメータの変更   |
|----------------------|---|---|
| <b>drop route</b>    | <pre>drop route to dest_replica- tion_server [with primary at dataserver. database] [with nowait]</pre> | <p>専用ルートを削除してから、共有ルートを削除してください。</p> <p>専用ルートが削除されると、指定プライマリコネクションから送信先 Replication Server へのトランザクションは、共有ルートを通るようになります。</p> <hr/> <p><b>警告！ with nowait</b> 句は、最後の手段としてのみ使用してください。</p> <p>この句を使用すると、ルートのアウトバウンドキューにトランザクションが含まれている場合でも、ルートが強制的に削除されます。その結果、Replication Server はプライマリコネクションからトランザクションを破棄する可能性があります。この句は、専用ルートが送信先 Replication Server と通信できない場合でも、専用ルートを削除するように、Replication Server に指示します。</p> <p>句を使用する場合、以前の送信先サイトで <b>sysadmin purge_route_at_replicate</b> を使用して、送信先のシステムテーブルからサブスクリプションおよびルート情報を削除します。</p> <hr/> <p>『Replication Server 管理ガイド 第1巻』の「ルートの管理」の「ルートの削除」の「<b>drop route</b> コマンド」を参照してください。</p> |
| <b>suspend route</b> | <pre>suspend route to dest_repli- cation_server [with primary at dataserver. database]</pre>            |   |

| コマンド                | 構文  | コマンドおよびパラメータの変更 |
|---------------------|---|-----------------|
| <b>resume route</b> | <pre>resume route to dest_replica- tion_server [with primary at dataserver. database] [skip transaction with large message]</pre> |                 |

## 専用ルート情報の表示

**admin who** を使用して、Replication Server 間の専用ルートの情報を表示します。

次の例では、NY\_DS.pdb1 プライマリコネクションのために、RS\_NY プライマリ Replication Server と RS\_LON レプリケート Replication Server との間に専用ルートがあります。2つの Replication Server に **admin who** と入力すると、次のように表示されます。

- RS\_LON では次のように表示されます。

```
Spid Name      State          Info
45 SQT         Awaiting Wakeup 103:1 DIST NY_DS.pdb1
13 SQM         Awaiting Message 103:1 NY_DS.pdb1
32 REP AGENT   Awaiting Command NY_DS.pdb1
16 RSI         Awaiting Wakeup  RS_LON
11 SQM         Awaiting Message 16777318:0 RS_LON
55 RSI         Awaiting Wakeup  RS_LON(103) /* Dedicated RSI
thread */
53 SQM         Awaiting Message 16777318:103 RS_LON(103) /
*Dedicated RSI outbound queue */
```

- RS\_NY では次のように表示されます。

```
Spid Name      State          Info
37 RSI USER    Awaiting Command  RS_NY(103) /*Dedicated RSI user
*/
32 RSI USER    Awaiting Command  RS_NY
```

『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「**admin who**」を参照してください。

## 異機種間 Multi-Path Replication のシナリオ

シナリオを使用して、Adaptive Server、Oracle、および SAPIQ データベース間に複数の複写パスを構築します。

## Adaptive Server から SAP HANA データベースへの Multi-Path Replication

Adaptive Server プライマリから SAP HANA データベースレプリケートまでのエンドツーエンドの複写のために、2つのプライマリとレプリケートのパスを持つ Multi-Path Replication システムを設定します。

### 前提条件

このシナリオでは、次がインストールされ設定済みであることを前提としています。

- プライマリ Adaptive Server データベースおよび RepAgent スレッド
- Replication Server
- ExpressConnect for SAP HANA データベース
- レプリケート SAP HANA データベース

### 手順

1. プライマリ Adaptive Server データベースで、2つの複写パスを介して複写するテーブルまたはストアドプロシージャを2セット作成または選択します。これらのトランザクションセットは、並列複写パスに分割可能でなければなりません。
2. プライマリ Adaptive Server データベースへのデフォルト接続を作成します。

**create connection** または **rs\_init** を使用して、デフォルト接続を作成します。

『Replication Server 設定ガイド』>「**rs\_init** による Replication Server の設定とデータベースの追加」を参照してください。

3. マルチスレッド Adaptive Server RepAgent と Adaptive Server RepAgent の複数パスを有効にします。

また、メモリを設定して Adaptive Server RepAgent に使用可能なバッファを送信することもできます。『Replication Server 管理ガイド 第2巻』>「パフォーマンスチューニング」>「Multi-Path Replication」>「複数のプライマリ複写パス」>「マルチスレッド RepAgent および RepAgent の複数のパスを有効にする」を参照してください。

- a) マルチスレッド RepAgent を有効にします。  
プライマリ Adaptive Server で次を入力します。

```
sp_config_rep_agent pdb, 'multithread rep agent', 'true'
```

ここで、*pdb* はプライマリ Adaptive Server のデータベースです。

- b) RepAgent への複写パス数を設定します。  
たとえば、2つのパスを有効にするには、次を入力します。

```
sp_config_rep_agent pdb, 'max number of replication paths', '2'
```

4. プライマリデータベースからの代替複写パスを作成します。

『Replication Server 管理ガイド: 第 2 巻』 > 「パフォーマンスチューニング」 > 「Multi-Path Replication」 > 「複数のプライマリ複写パス」 > 「複数のプライマリ複写パスの作成」を参照してください。

- a) プライマリデータベースと Replication Server の間に `pdb_conn2` という名前の代替複写パスを作成します。  
プライマリデータベースで、次を入力します。

```
sp_replication_path "pdb", 'add',
"pdb_conn2", "PRS",
"muser", "mpwd"
```

各パラメータの意味は、次のとおりです。

- *PRS* は、Replication Server です。
- *muser* は、メンテナンスユーザです。
- *mpwd* は、メンテナンスユーザのパスワードです。

(オプション) 複数の Replication Server の物理パスにバインドされているデータやオブジェクトバウンドの分散に使用できる論理パスを作成します。

『Replication Server 管理ガイド 第 2 巻』 > 「パフォーマンスチューニング」 > 「Multi-Path Replication」 > 「複数のプライマリ複写パス」 > 「論理プライマリ複写パスの作成」を参照してください。

- b) Replication Server に対応する代替プライマリ接続を作成し、代替パス名 `pdb_conn2` を持つ同じ Adaptive Server データサーバ名を使用します。  
Replication Server で次を入力します。

```
create alternate connection to pds.pdb
named pds.pdb_conn2
set error class to rs_sqlserver_error_class
set function string class to rs_sqlserver_function_class
set username to muser
set password to mpwd
with primary only
```

ここで *pds* はプライマリ Adaptive Server です。

複写システムには、2つのプライマリ複写パス (デフォルトと `pdb_conn2`) が含まれます。

5. `admin show_connections, 'primary'` を使用して、作成したプライマリ接続を表示します。
6. レプリケート SAP HANA データベースインスタンスで、レプリケート SAP HANA データベースインスタンスのメンテナンスユーザを作成します。
- ```
CREATE USER muser PASSWORD mpwd;
```
7. 適切な権限をメンテナンスユーザに付与します。

- **CREATE ANY**
  - **DELETE**
  - **DROP**
  - (すべてのレプリケートストアプロシージャに対する) **EXECUTE**
  - **INDEX**
  - **SELECT**
  - (すべてのレプリケートテーブルに対する) **UPDATE**
8. 標準接続で SAP HANA データベースに接続しており、SAP Secure User Store を使用していない場合は、レプリケート SAPHANA データサーバを識別するエントリーを `interfaces` ファイルを追加し、Replication Server を再起動します。

```
[dataservername]
master tcp ether hostname port
query tcp ether hostname port
```

ここで `hostname` と `port` は SAP HANA データサーバのホストとポート番号、`dataservername` はホストとポート番号の識別に使用するラベルです。

SAP Secure User Store を使用する場合は、次のように暗号化された資格のユーザストアを作成します。

```
hdbuserstore set rds myhost:xxxxx my_securestore_user
my_securestore_pwd
```

ここで

- `rds` はセキュアなストアのエントリーのキーです。
  - `myhost.xxxxx` は、接続環境のホスト名とポート番号です。
  - `my_securestore_user` と `my_securestore_pwd` は、SAP Secure User Store の資格です。
9. ExpressConnect for SAP HANA データベースを使用して、レプリケート SAP HANA データベースインスタンスに対する接続を作成します。

標準接続の場合は、次のように指定します。

```
create connection to rds.rdb
using profile rs_ase_to_hanadb;ech
set username muser
set password mpwd
go
```

SAP Secure User Store の場合は次のように指定します。

```
create connection to rds.rdb
using profile rs_ase_to_hanadb;ech
set username auser
set password apwd
set dsi_connector_sec_mech to "hdbuserstore"
go
```

各パラメータの意味は、次のとおりです。

- *rds*は、レプリケート SAP HANA データサーバです。標準接続の場合、これは *interfaces* ファイルのエントリと一致する必要があります。SAP Secure User Store 接続の場合、これは、暗号化された資格のユーザストアを **hdbuserstore** ユーティリティを使用して作成する際にキーとして使用したものと一致する必要があります。
- *rdb*はブレースホルダです。値を指定する必要がありますが、その値は使用されません。
- *muser*は、レプリケート SAP HANA データベースインスタンスのメンテナンスユーザです。
- *mpwd*は、レプリケート SAP HANA データベースのメンテナンスユーザのパスワードです。
- *auser* と *apwd*は、**create connection** コマンドの構文を満たすためだけに指定される未使用の値です。

---

**注意：**セキュアなストアは、Replication Server を開始し実行しているオペレーティングシステムのユーザ ID と同じ ID で作成する必要があります。それ以外の ID を使用すると、Replication Server はセキュアなユーザストアにアクセスできません。

---

**10.** ExpressConnect for SAP HANA データベースを介して、レプリケート SAP HANA データベースへの代替接続を作成します。

標準接続の場合は、次のように指定します。

```
create alternate connection to rds.conn2
using profile rs_ase_to_hanadb;ech
set username muser
set password mpwd
go
```

SAP Secure User Store の場合は次のように指定します。

```
create connection to rds.conn2
using profile rs_ase_to_hanadb;ech
set username auser
set password apwd
set dsi_connector_sec_mech to "hdbuserstore"
go
```

**11.** プライマリデータベースで、*ptab1* テーブルおよび *ptab2* テーブルに複写対象としてマーク付けします。

```
sp_setreptable ptab1, 'true'
go
sp_setreptable ptab2, 'true'
go
```

**12.** プライマリデータベースで分散モードをオブジェクトのバインド別分散に設定します。

```
sp_config_rep_agent pdb, 'multipath distribution model', 'object'
```

13. Replication Server をクワイス状態にし、RepAgent を再起動します。

『Replication Server 管理ガイド 第 1 巻』>「複写システムの管理」>「Replication Server のクワイス」>「複写システムのクワイス」を参照してください。

14. ptab2 テーブルを pdb\_conn2 代替接続にバインドします。

```
sp_replication_path pdb, 'bind', "table", "dbo.ptab2",
"pdb_conn2"
```

オブジェクトを代替複写パスにバインドすることしかできません。ptab1 テーブルのような代替複写パスにバインドしないすべてのオブジェクトは、代わりにデフォルトパスを使用します。

15. プライマリ Adaptive Server データベースに 2 つの複写定義を作成します。

たとえば、ptab1 テーブルの ptab1\_repdef 複写定義を作成するには、次を実行します。

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

ptab2 テーブルの ptab2\_repdef 複写定義を作成するには、次を実行します。

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

---

**注意：**これらの複写定義には、デフォルトのプライマリ接続名を使用する必要があります。

---

プライマリ接続と代替プライマリ接続が異なる Replication Server に存在する場合、それぞれの Replication Server に複写定義を作成します。

16. デフォルトの複写接続に対し、サブスクリプションを作成します。

たとえば、ptab1\_repdef 複写定義の ptab1\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at rds.rdb
without materialization
go
```

17. 代替複写接続に対し、サブスクリプションを作成します。

たとえば、ptab2\_repdef 複写定義の ptab2\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab2_sub
for ptab2_repdef
```

```
with replicate at rds.conn2
without materialization
go
```

### Adaptive Server から Oracle への Multi-Path Replication

Adaptive Server プライマリから Oracle レプリケートまでのエンドツーエンドの複写のために、2つのプライマリとレプリケートのパスを持つ Multi-Path Replication システムを設定します。

#### 前提条件

このシナリオでは、次がインストールされ設定済みであることを前提としています。

- プライマリ Adaptive Server データベースおよび RepAgent スレッド
- Replication Server
- ExpressConnect for Oracle
- レプリケート Oracle データベース

#### 手順

1. プライマリ Adaptive Server データベースで、2つの複写パスを介して複写するテーブルまたはストアドプロシージャを2セット作成または選択します。これらのトランザクションセットは、並列複写パスに分割可能でなければなりません。
2. プライマリ Adaptive Server データベースへのデフォルト接続を作成します。

**create connection** または **rs\_init** を使用して、デフォルト接続を作成します。『Replication Server 設定ガイド』>「**rs\_init** による Replication Server の設定とデータベースの追加」を参照してください。

3. マルチスレッド Adaptive Server RepAgent と Adaptive Server RepAgent の複数パスを有効にします。

また、メモリを設定して Adaptive Server RepAgent に使用可能なバッファを送信することもできます。『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」で「複数のプライマリレプリケーションパス」の「マルチスレッド RepAgent および RepAgent の複数のパスを有効にする」を参照してください。

- a) マルチスレッド RepAgent を有効にします。  
プライマリ Adaptive Server で次を入力します。

```
sp_config_rep_agent pdb, 'multithread rep agent', 'true'
```

ここで、*pdb* はプライマリ Adaptive Server のデータベースです。

- b) RepAgent へのレプリケーションパスの数を設定します。



たとえば、2つのパスを有効にするには、次を入力します。

```
sp_config_rep_agent pdb, 'max number of replication paths', '2'
```

#### 4. プライマリデータベースからの代替レプリケーションパスを作成します。

『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」の「複数のプライマリレプリケーションパス」の「複数のプライマリレプリケーションパスの作成」を参照してください。

- a) プライマリデータベースと Replication Server の間に `pdb_conn2` という名前の代替レプリケーションパスを作成します。  
プライマリデータベースで、次を入力します。

```
sp_replication_path "pdb", 'add',
"pdb_conn2", "PRS",
"muser", "mpwd"
```

説明は次のとおりです。

- *PRS* は、Replication Server です。
- *muser* は、メンテナンスユーザです。
- *mpwd* は、メンテナンスユーザのパスワードです。

(オプション) 複数の Replication Server の物理パスにバインドされているデータやオブジェクトバウンドの分散に使用できる論理パスを作成します。

『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」の「複数のプライマリレプリケーションパス」の「論理プライマリレプリケーションパスの作成」を参照してください。

- b) Replication Server に対応する代替プライマリコネクションを作成し、代替パス名 `pdb_conn2` を持つ同じ Adaptive Server データサーバ名を使用します。  
Replication Server で次を入力します。

```
create alternate connection to pds.pdb
named pds.pdb_conn2
set error class to rs_sqlserver_error_class
set function string class to rs_sqlserver_function_class
set username to muser
set password to mpwd
with primary only
```

ここで *pds* はプライマリ Adaptive Server です。

複写システムには、2つのプライマリレプリケーションパス (デフォルトと `pdb_conn2`) が含まれます。

5. `admin show_connections, 'primary'` を使用して、作成したプライマリコネクションを表示します。
6. レプリケート Oracle データベースの `tnsnames.ora` ファイルを Replication Server の `RS_installation_directory¥REP-15_5¥connector¥oraoci`

¥network¥admin ディレクトリにコピーして、Replication Server を再起動します。

7. ExpressConnect for Oracle を介して、レプリケート Oracle データベースへの接続を作成します。

```
create connection to tns_alias_name.rdb
using profile rs_oracle_to_oracle;eco
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_connector_type to 'oci'
set batch to 'off'
go
```

各パラメータの意味は、次のとおりです。

- *tns\_alias\_name* は、レプリケート Oracle データベースの *tnsnames.ora* ファイルで定義されたレプリケート Oracle データベースのエイリアス名です。
- *rdb* は、*tnsnames.ora* ファイルの上記の *tns\_alias\_name* と対になるレプリケート Oracle System ID (SID) です。デフォルト値は *ORCL* です。
- *muser* は、レプリケート Oracle データベースのメンテナンスユーザです。
- *mpwd* は、レプリケート Oracle のメンテナンスユーザのパスワードです。

デフォルト接続の命名の詳細については、『ExpressConnect for Oracle Configuration Guide』を参照してください。

8. ExpressConnect for Oracle を介して、レプリケート Oracle データベースへの代替接続を作成します。

```
create alternate connection to tns_alias_name.rdb
named tns_alias_name.conn2
set error_class rs_oracle_error_class
set function_string_class rs_oracle_function_class
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_dataserver_type to 'oci'
set batch to 'off'
set dsi_proc_as_rpc to 'on'
go
```

9. プライマリデータベースで、*ptab1* テーブルおよび *ptab2* テーブルに複写対象としてマーク付けします。

```
sp_setreptable ptab1,'true'
go
sp_setreptable ptab2,'true'
go
```

10. プライマリデータベースで分散モードをオブジェクトのバインド別分散に設定します。

```
sp_config_rep_agent pdb, 'multipath distribution model', 'object'
```

11. Replication Server をクワイス状態にし、RepAgent を再起動します。

『Replication Server 管理ガイド 第 1 巻』の「複写システムの管理」にある「Replication Server のクワイス」の「複写システムのクワイス」を参照してください。

12. ptab2 テーブルを pdb\_conn2 代替接続にバインドします。

```
sp_replication_path pdb, 'bind', "table", "dbo.ptab2",
"pdb_conn2"
```

オブジェクトを代替レプリケーションパスにバインドすることしかできません。ptab1 テーブルのような代替レプリケーションパスにバインドしないすべてのオブジェクトは、代わりにデフォルトパスを使用します。

13. プライマリ Adaptive Server データベースに 2 つの複写定義を作成します。

たとえば、ptab1 テーブルの ptab1\_repdef 複写定義を作成するには、次を実行します。

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

ptab2 テーブルの ptab2\_repdef 複写定義を作成するには、次を実行します。

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

---

**注意：**これらの複写定義には、デフォルトのプライマリ接続名を使用する必要があります。

---

プライマリ接続と代替プライマリ接続が異なる Replication Server に存在する場合、それぞれの Replication Server に複写定義を作成します。

14. デフォルトの複写接続に対し、サブスクリプションを作成します。

たとえば、ptab1\_repdef 複写定義の ptab1\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at tns_alias_name.rdb
without materialization
go
```

15. 代替複写接続に対し、サブスクリプションを作成します。

たとえば、ptab2\_repdef 複写定義の ptab2\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at tns_alias_name.conn2
without materialization
go
```

### 参照：

- レプリケートデータサーバとしての Oracle (133 ページ)

## Adaptive Server から SAP IQ への Multi-Path Replication

Adaptive Server プライマリデータベースから SAP IQ レプリケートデータベースへのエンドツーエンドの複写のために、プライマリとレプリケートの2つのコネクションを持つ Multi-Path Replication システムを設定します。

このシナリオの複写システムは、ASE\_DS プライマリ Adaptive Server の pdb データベース、iqdb データベースが格納された IQSRVR レプリケート SAP IQ データサーバ、PRS プライマリ Replication Server、testtab1 テーブル、testtab2 テーブルで構成されます。

1. pdb プライマリデータベースで、2つのレプリケーションパスを介して複写するテーブルまたはストアドプロシージャを2セット作成または選択します。
2. pdb プライマリ Adaptive Server データベースへのデフォルトのコネクションを作成します。

**create connection** または **rs\_init** を使用して、デフォルトのコネクションを作成します。『Replication Server 設定ガイド』の「**rs\_init** による Replication Server の設定とデータベースの追加」を参照してください。

3. マルチスレッド Adaptive Server RepAgent と Adaptive Server RepAgent の複数パスを有効にします。

また、メモリを設定して Adaptive Server RepAgent に使用可能なバッファを送信することもできます。『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」で「複数のプライマリレプリケーションパス」の「マルチスレッド RepAgent および RepAgent の複数のパスを有効にする」を参照してください。

- a) マルチスレッド RepAgent を有効にします。  
プライマリ Adaptive Server で、次を入力します。

```
sp_config_rep_agent pdb, 'multithread rep agent', 'true'
```

- b) RepAgent へのレプリケーションパスの数を設定します。  
たとえば、2つのパスを有効にするには、次を入力します。

```
sp_config_rep_agent pdb, 'max number of replication paths', '2'
```

4. pdb プライマリデータベースから PRS Replication Server への代替レプリケーションパスを作成します。

『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」の「複数のプライマリレプリケーションパス」の「複数のプライマリレプリケーションパスの作成」を参照してください。

- a) pdb\_conn2 という名前の代替レプリケーションパスを作成します。場所は pdb と PRS Replication Server の間になります。  
プライマリ Adaptive Server で、次を入力します。

```
sp_replication_path "pdb", 'add',
"pdb_conn2", "PRS",
"dbmaint2", "dbmaint2pwd"
```

(オプション) 複数の Replication Server の物理パスにバインドされているデータやオブジェクトバウンドの分散に使用できる論理パスを作成します。

『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」の「複数のプライマリレプリケーションパス」の「論理プライマリレプリケーションパスの作成」を参照してください。

- b) Replication Server に対応する代替プライマリコネクションを作成し、代替パス名 pdb\_conn2 を持つ同じ Adaptive Server データサーバ名を使用します。  
Replication Server で次のように入力します。

```
create alternate connection to ASE_DS.pdb
named ASE_DS.pdb_conn2
set error_class to rs_sqlserver_error_class
set function_string_class to rs_sqlserver_function_class
set username to dbmaint2
set password to dbmaint2pwd
with primary only
```

複写システムには、2つのプライマリレプリケーションパスがあります。デフォルトは、ASE\_DS.pdb と ASE\_DS.pdb\_conn2 です。

5. **admin show\_connections, 'primary'** を使用して、作成したプライマリコネクションを表示します。
6. メンテナンスユーザが dbmaint1 の SAP IQ データベースへのデフォルトのレプリケートコネクションを作成します。

コネクションプロファイルとコネクションプロファイルのバージョン、およびデフォルトのコネクションと各代替コネクションの一意のメンテナンスユーザ名を指定する必要があります。

```
create connection to IQSRVR.iqdb
using profile rs_ase_to_iq;standard
set username to dbmaint1
set password to dbmaint1
go
```

7. コネクションが **admin who** で実行していることを確認します。
8. メンテナンスユーザが dbmaint2 で、IQSRVR.pdb\_conn2 という名前の、iqdb SAP IQ データベースへの代替レプリケートコネクションを作成します。

## 異機種間における Multi-Path Replication

```
create alternate connection to IQSRVR.iqdb
named IQSRVR.pdb_conn2
using profile rs_ase_to_iq;standard
set username to dbmaint2
set password to dbmaint2pwd
go
```

(オプション) 使用可能な SAP IQ マルチプレックスノードへの代替コネクションを作成します。コネクション名がユニークであることを確認します。

たとえば、IQSRVR2 SAP IQ ノードの iqdb2 データベースへの pdb\_conn3 代替コネクションを作成するには、次のように入力します。

```
create alternate connection to IQSRVR2.pdb_conn3
named IQSRVR2.iqdb2_conn1
using profile rs_ase_to_iq;standard
set username to dbmaint3
set password to dbmaint3pwd
go
```

9. **admin show\_connections, 'replicate'** を使用して作成したレプリケートコネクションを表示します。
10. SAP IQ へのデフォルトおよび代替レプリケートコネクションの RTL を有効にします。
  - a) デフォルトのコネクションの RTL を有効にします。

```
alter connection to IQSRVR.iqdb
set dsi_compile_enable to 'on'
go
```

- b) IQSRVR.pdb\_conn2 代替コネクションの RTL を有効にします。

```
alter connection to IQSRVR.pdb_conn2
set dsi_compile_enable to 'on'
go
```

- c) レプリケート SAP IQ データベースへのコネクションをサスペンドして再開します。

```
suspend connection to IQSRVR.iqdb
go
suspend connection to IQSRVR.pdb_conn2
go
resume connection to IQSRVR.iqdb
go
resume connection to IQSRVR.pdb_conn2
go
```

**configure replication server** を使用して RTL を有効にする場合や、すべてのコネクションを同時にサスペンドしてレジュームする場合は、Replication Server を再起動します。

11. RTL の処理を設定し、RTL パフォーマンスを調整するため、オプションとしてパラメータ値を設定します。
12. プライマリデータベースで、レプリケーションのため、testtab1 テーブルおよび testtab2 テーブルにマーク付けします。

```
sp_setreptable testtab1, 'true'
go
sp_setreptable testtab2, 'true'
go
```

13. プライマリデータベースで分散モードをオブジェクトのバインド別分散に設定します。

```
sp_config_rep_agent pdb, 'multipath distribution model', 'object'
```

14. Replication Server をクワイス状態にし、RepAgent を再起動します。

『Replication Server 管理ガイド 第1巻』の「複写システムの管理」にある「Replication Server のクワイス」の「複写システムのクワイス」を参照してください。

15. testtab2 テーブルを ASE\_DS.pdb\_conn2 代替コネクシオンにバインドします。

```
sp_replication_path pdb, 'bind', "table", "dbo.testtab2",
"pdb_conn2"
```

オブジェクトを代替レプリケーションパスにバインドすることしかできません。testtab1 テーブルなど、代替レプリケーションパスへバインドしないオプションはすべて、代わりにデフォルトのパスを使用します。

16. マーク付けされたテーブルの複写定義を作成します。RTL をサポートする複写定義に必要なすべての参照制約句を追加します。

testtab1 テーブルの **repdef\_testtab1** 複写定義を作成するには、次を入力します。

```
create replication definition repdef_testtab1
with primary at ASE_DS.pdb1
with primary table named 'testtab1'
with replicate table named dbo.'testtab1'
(c1 int, c2 int, c3 char(10))
primary key(c1)
go
```

testtab2 テーブルの **repdef\_testtab2** 複写定義を作成するには、次を入力します。

```
create replication definition repdef_testtab2
with primary at ASE_DS.pdb1
with primary table named 'testtab2'
with replicate table named dbo.'testtab2'
(c1 int, c2 int, c3 char(10))
primary key(c1)
go
```

すべての複写定義は、デフォルトのプライマリコネクシオンを参照します。

17. 各テーブルの複写定義に一致し、コネクシオンを指定するサブスクリプションを作成します。

- a) **repdef\_testtab1** 複写定義の **sub\_testtab1** サブスクリプションを作成し、レプリケートトランザクションへのデフォルトのコネクシオンを指定します。

## 異機種間における Multi-Path Replication

```
create subscription sub_testtab1 for repdef_testtab1
with replicate at IQSRVR.iqdb
without materialization
go
```

- b) **repdef\_testtab2** 複写定義の **sub\_testtab2** サブスクリプションを作成して、レプリケートトランザクションへの IQSRVR.iqdb\_conn2 代替コネクションを指定します。

```
create subscription sub_testtab2 for repdef_testtab2
with replicate at IQSRVR.pdb_conn2
without materialization
go
```

『Replication Server 管理ガイド 第2巻』の「パフォーマンスチューニング」の「Multi-Path Replication」の「複写定義およびサブスクリプション」で「コネクション間でのサブスクリプションの移動」を参照してください。

### 参照：

- レプリケートデータサーバとしての SAP IQ (145 ページ)
- RTL パフォーマンスのチューニング (159 ページ)

## Oracle から Adaptive Server への Multi-Path Replication

Oracle プライマリから Adaptive Server 複写へのエンドツーエンドの複写のために、2つのプライマリとレプリケートのコネクションを持つ Multi-Path Replication システムを設定します。

### 前提条件

このシナリオでは、次がインストールされ設定済みであることを前提としています。

- LogMiner のあるプライマリ Oracle データベース
- 2つの Replication Agent インスタンス
- Replication Server
- レプリケート Adaptive Server データベース

### 手順

1. プライマリ Oracle データベースで、2つの複写パスを介して複写するテーブルを2セット作成または選択します。
2. **rs\_init** を使用して、レプリケートデータベースを複写システムに追加します。
3. プライマリ Oracle データベースへのデフォルト接続を作成します。

```
create connection to pds.pdb
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
```



```
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

構文の説明は次のとおりです。

- *pds* は、Replication Agent で指定された **rs\_source\_ds** パラメータの値です。
- *pdb* は、Replication Agent で指定された **rs\_source\_db** の値です。
- *muser* は、プライマリ Oracle データベースのメンテナンスユーザです。
- *mpwd* は、メンテナンスユーザのパスワードです。

4. プライマリ Oracle データベースへの代替接続を作成します。

```
create alternate connection to pds.pdb
named pds.conn2
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with primary only
go
```

5. レプリケート Adaptive Server データベースへの接続を作成します。

```
create connection to rds.rdb
set error class rs_sqlserver_error_class
set function string class rs_sqlserver_function_class
set username muser
set password mpwd
go
```

構文の説明は次のとおりです。

- *rds* は、レプリケート Adaptive Server のデータサーバの名前です。
- *rdb* は、レプリケート Adaptive Server のデータベースです。
- *muser* は、レプリケート Adaptive Server のデータベースのメンテナンスユーザです。
- *mpwd* は、メンテナンスユーザのパスワードです。

6. レプリケート Adaptive Server のデータベースへの代替コネクションを作成します。

```
create alternate connection to rds.rdb
named rds.conn2
set error class rs_sqlserver_error_class
set function string class rs_sqlserver_function_class
set username muser
set password mpwd
go
```

7. create object パーミッションを Replication Server の **rs\_username** に付与します。

```
grant create object to rs_username
go
```

ここで、`rs_username` は、Replication Agent が Replication Server へのアクセスに使用するユーザログイン名です。

8. Replication Agent の単一インスタンスでは、`ra_admin_owner`、`ra_admin_prefix`、`ra_admin_instance_prefix`、`rs_source_ds`、および `rs_source_db` パラメータを設定します。

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri1
ra_config rs_source_ds, pds
ra_config rs_source_db, pdb
```

パラメータの説明は次のとおりです。

- `ra_user_1` は、Replication Agent インスタンスが使用する目的で、プライマリデータベース内での共有オブジェクトおよびインスタンスオブジェクトの作成に使用されるユーザ名です。このユーザ名は、プライマリデータベース上で定義済みであることが必要です。
  - `ra_` は、プライマリデータベースで共有オブジェクトを識別するために使用されるプレフィクスです。このプレフィクスは、3 文字以下です。
  - `ri1` は、レプリケーショングループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
  - `rs_source_ds` の値は `rs_source_db` の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用するコネクション名を形成します。
9. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

10. Replication Agent の他のインスタンスでは、`ra_admin_owner`、`ra_admin_prefix`、`ra_admin_instance_prefix`、`rs_source_ds`、および `rs_source_db` パラメータを設定します。

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri2
ra_config rs_source_ds, pds
ra_config rs_source_db, conn2
```

パラメータの説明は次のとおりです。

- `ri2` は、レプリケーショングループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
- `ra_admin_owner` の値と `ra_admin_prefix` の値は、レプリケーショングループ内の他のすべての Replication Agent インスタンスと同じです。
- `rs_source_ds` の値は `rs_source_db` の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用するコネクション名を形成します。

11. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

12. 手順 1 の 2 つのテーブルに複写対象としてマークを付けます。ri1 で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab1, mark
go
```

ri2 で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab2, mark
go
```

ここで、ptab1 および ptab2 は、複写されるプライマリデータベーステーブルです。

13. プライマリ Oracle データベースに対し、2 つの複写定義を作成します。たとえば、ptab1\_repdef 複写定義を ptab1 テーブルに作成するには、次を入力します。

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

ptab2\_repdef 複写定義を ptab2 テーブルに作成するには、次を入力します。

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

---

**注意：**これらの複写定義には、デフォルトのプライマリコネクション名を使用する必要があります。

---

プライマリ接続と代替プライマリ接続が異なる Replication Server に存在する場合、それぞれの Replication Server に複写定義を作成します。

14. Replication Agent のインスタンスを再開します。

```
resume
```

Replication Agent インスタンスの再開に失敗した場合、LogMiner がインストールされ設定されているかを確認します。『Replication Agent プライマリデータベースガイド』の「Replication Agent for Oracle」の「Oracle 固有の考慮事項」の「Oracle トランザクションおよびオペレーションのトラブルシューティング」の「ra\_dumptran と ra\_helpop を使用するための Replication Agent と Oracle の設定」を参照してください。

## 異機種間における Multi-Path Replication

15. デフォルトの複製接続に対し、サブスクリプションを作成します。  
たとえば、ptab1\_repdef 複製定義の ptab1\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at rds.rdb
without materialization
go
```

16. 代替複製接続に対し、サブスクリプションを作成します。  
たとえば、ptab2\_repdef 複製定義の ptab2\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at rds.conn2
without materialization
go
```

## Oracle から SAP HANA データベースへの Multi-Path Replication

Oracle プライマリから SAP HANA データベースレプリケートへのエンドツーエンドの複製のために、2つのプライマリとレプリケートの接続を持つ Multi-Path Replication システムを設定します。

### 前提条件

このシナリオでは、次がインストールされ設定済みであることを前提としています。

- LogMiner のあるプライマリ Oracle データベース
- 2つの Replication Agent インスタンス
- Replication Server
- ExpressConnect for SAP HANA データベース
- レプリケート SAP HANA データベース

### 手順

1. プライマリ Oracle データベースで、2つの複製パスを介して複製するテーブルを2つ作成または選択します。
2. プライマリ Oracle データベースへのデフォルト接続を作成します。

```
create connection to pds.pdb
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

各パラメータの意味は、次のとおりです。

- *pds* は、Replication Agent で指定された **rs\_source\_ds** パラメータの値です。
- *pdb* は、Replication Agent で指定された **rs\_source\_db** の値です。
- *muser* は、プライマリ Oracle データベースのメンテナンスユーザです。
- *mpwd* は、メンテナンスユーザのパスワードです。

### 3. プライマリ Oracle データベースへの代替接続を作成します。

```
create alternate connection to pds.pdb
named pds.conn2
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with primary only
go
```

### 4. レプリケート SAP HANA データベースインスタンスで、レプリケート SAP HANA データベースインスタンスのメンテナンスユーザを作成します。

```
CREATE USER muser PASSWORD mpwd;
```

### 5. 適切な権限をメンテナンスユーザに付与します。

- **CREATE ANY**
- **DELETE**
- **DROP**
- (すべてのレプリケートストアプロシージャに対する) **EXECUTE**
- **INDEX**
- **SELECT**
- (すべてのレプリケートテーブルに対する) **UPDATE**

### 6. 標準接続で SAP HANA データベースに接続しており、SAP Secure User Store を使用していない場合は、レプリケート SAPHANA データサーバを識別するエントリーを *interfaces* ファイルを追加し、Replication Server を再起動します。

```
[dataservername]
master tcp ether hostname port
query tcp ether hostname port
```

ここで *hostname* と *port* は SAP HANA データサーバのホストとポート番号、*dataservername* はホストとポート番号の識別に使用するラベルです。

SAP Secure User Store を使用する場合は、次のように暗号化された資格のユーザストアを作成します。

```
hdbuserstore set rds myhost:xxxxx my_securestore_user
my_securestore_pwd
```

ここで

- *rds* はセキュアなストアのエントリーのキーです。

- *myhost.xxxxx* は、接続環境のホスト名とポート番号です。
  - *my\_securestore\_user* と *my\_securestore\_pwd* は、SAP Secure User Store の資格です。
7. ExpressConnect for SAP HANA データベースを使用して、レプリケート SAP HANA データベースインスタンスに対する接続を作成します。

標準接続の場合は、次のように指定します。

```
create connection to rds.rdb
using profile rs_ase_to_hanadb;ech
set username muser
set password mpwd
go
```

SAP Secure User Store の場合は次のように指定します。

```
create connection to rds.rdb
using profile rs_ase_to_hanadb;ech
set username auser
set password apwd
set dsi_connector_sec_mech to "hdbuserstore"
go
```

各パラメータの意味は、次のとおりです。

- *rds* は、レプリケート SAP HANA データサーバです。標準接続の場合、これは *interfaces* ファイルのエントリと一致する必要があります。SAP Secure User Store 接続の場合、これは、暗号化された資格のユーザストアを **hdbuserstore** ユーティリティを使用して作成する際にキーとして使用したものと一致する必要があります。
- *rdb* はプレースホルダです。値を指定する必要がありますが、その値は使用されません。
- *muser* は、レプリケート SAP HANA データベースインスタンスのメンテナンスユーザです。
- *mpwd* は、レプリケート SAP HANA データベースのメンテナンスユーザのパスワードです。
- *auser* と *apwd* は、**create connection** コマンドの構文を満たすためだけに指定される未使用の値です。

---

**注意：**セキュアなストアは、Replication Server を開始し実行しているオペレーティングシステムのユーザ ID と同じ ID で作成する必要があります。それ以外の ID を使用すると、Replication Server はセキュアなユーザストアにアクセスできません。

---

8. ExpressConnect for SAP HANA データベースを介して、レプリケート SAP HANA データベースへの代替接続を作成します。

標準接続の場合は、次のように指定します。

```
create alternate connection to rds.conn2
using profile rs_ase_to_hanadb;ech
```

```
set username muser
set password mpwd
go
```

SAP Secure User Store の場合は次のように指定します。

```
create connection to rds.conn2
using profile rs_ase_to_hanadb;ech
set username auser
set password apwd
set dsi_connector_sec_mech to "hdbuserstore"
go
```

9. create object パーミッションを Replication Server の **rs\_username** に付与します。

```
grant create object to rs_username
go
```

ここで、**rs\_username** は、Replication Agent が Replication Server へのアクセスに使用するユーザログイン名です。

10. Replication Agent の単一インスタンスでは、**ra\_admin\_owner**、**ra\_admin\_prefix**、**ra\_admin\_instance\_prefix**、**rs\_source\_ds**、および **rs\_source\_db** パラメータを設定します。

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri1
ra_config rs_source_ds, pds
ra_config rs_source_db, pdb
```

ここで、

- **ra\_user\_1** は、Replication Agent インスタンスが使用する目的で、プライマリデータベース内での共有オブジェクトおよびインスタンスオブジェクトの作成に使用されるユーザ名です。このユーザ名は、プライマリデータサーバで定義済みであることが必要です。
- **ra\_** は、プライマリデータベースで共有オブジェクトを識別するために使用されるプレフィクスです。このプレフィクスは、3文字以下です。
- **ri1** は、複製グループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
- **rs\_source\_ds** の値は **rs\_source\_db** の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用する接続名を形成します。

11. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

12. Replication Agent の他のインスタンスでは、**ra\_admin\_owner**、**ra\_admin\_prefix**、**ra\_admin\_instance\_prefix**、**rs\_source\_ds**、および **rs\_source\_db** パラメータを設定します。

## 異機種間における Multi-Path Replication

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri2
ra_config rs_source_ds, pds
ra_config rs_source_db, conn2
```

ここで、

- **ri2** は、複製グループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
- **ra\_admin\_owner** の値と **ra\_admin\_prefix** の値は、複製グループ内の他のすべての Replication Agent インスタンスと同じです。
- **rs\_source\_ds** の値は **rs\_source\_db** の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用する接続名を形成します。

### 13. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

### 14. 手順 1 の 2 つのテーブルに複製対象としてマークを付けます。ri1 で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab1, mark
go
```

ri2 で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab2, mark
go
```

ここで、ptab1 および ptab2 は、複製されるプライマリデータベーステーブルです。

### 15. プライマリ Oracle データベースに対し、2 つの複製定義を作成します。

たとえば、ptab1 テーブルの ptab1\_repdef 複製定義を作成するには、次を実行します。

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

ptab2 テーブルの ptab2\_repdef 複製定義を作成するには、次を実行します。

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

---

**注意：**これらの複製定義には、デフォルトのプライマリ接続名を使用する必要があります。

---



プライマリ接続と代替プライマリ接続が異なる Replication Server に存在する場合、それぞれの Replication Server に複製定義を作成します。

16. Replication Agent のインスタンスを再開します。

```
resume
```

Replication Agent インスタンスの再開に失敗した場合、LogMiner がインストールされ設定されていることを確認します。『Replication Agent プライマリデータベースガイド』>「Replication Agent for Oracle」>「Oracle 固有の考慮事項」>「Oracle トランザクションおよびオペレーションのトラブルシューティング」>「ra\_dumptran と ra\_helpop を使用するための Replication Agent と Oracle の設定」を参照してください。

17. デフォルトの複製接続に対し、サブスクリプションを作成します。  
たとえば、ptab1\_repdef 複製定義の ptab1\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at rds.rdb
without materialization
go
```

18. 代替複製接続に対し、サブスクリプションを作成します。  
たとえば、ptab2\_repdef 複製定義の ptab2\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at rds.conn2
without materialization
go
```

## Oracle から Oracle への Multi-Path Replication

Oracle プライマリから Oracle レプリケートへのエンドツーエンドの複製のために、2つのプライマリとレプリケートのコネクションを持つ Multi-Path Replication システムを設定します。

### 前提条件

このシナリオでは、次がインストールされ設定済みであることを前提としています。

- LogMiner のあるプライマリ Oracle データベース
- 2つの Replication Agent インスタンス
- Replication Server
- ExpressConnect for Oracle
- レプリケート Oracle データベース

### 手順

1. プライマリ Oracle データベースで、2つのレプリケーションパスを介して複写するテーブルを2つ作成または選択します。
2. プライマリ Oracle データベースへのデフォルト接続を作成します。

```
create connection to pds.pdb
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

各パラメータの意味は、次のとおりです。

- *pds* は、Replication Agent で指定された **rs\_source\_ds** パラメータの値です。
- *pdb* は、Replication Agent で指定された **rs\_source\_db** の値です。
- *muser* は、プライマリ Oracle データベースのメンテナンスユーザです。
- *mpwd* は、メンテナンスユーザのパスワードです。

3. プライマリ Oracle データベースへの代替接続を作成します。

```
create alternate connection to pds.pdb
named pds.conn2
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with primary only
go
```

4. レプリケート Oracle データベースの `tnsnames.ora` ファイルを Replication Server の `RS_installation_directory\REP-15_5\connector\oraoci\network\admin` ディレクトリにコピーして、Replication Server を再起動します。
5. ExpressConnect for Oracle を介して、レプリケート Oracle データベースへの接続を作成します。

```
create connection to tns_alias_name.rdb
using profile rs_oracle_to_oracle;eco
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_connector_type to 'oci'
set batch to 'off'
go
```

各パラメータの意味は、次のとおりです。

- *tns\_alias\_name* は、レプリケート Oracle データベースの *tnsnames.ora* ファイルで定義されたレプリケート Oracle データベースのエイリアス名です。
  - *rdb* は、*tnsnames.ora* ファイルの上記の *tns\_alias\_name* と対になるレプリケート Oracle System ID (SID) です。デフォルト値は *ORCL* です。
  - *muser* は、レプリケート Oracle データベースのメンテナンスユーザです。
  - *mpwd* は、レプリケート Oracle のメンテナンスユーザのパスワードです。
- デフォルト接続の命名の詳細については、『ExpressConnect for Oracle Configuration Guide』を参照してください。

6. ExpressConnect for Oracle を介して、レプリケート Oracle データベースへの代替接続を作成します。

```
create alternate connection to tns_alias_name.rdb
named tns_alias_name.conn2
set error_class rs_oracle_error_class
set function_string_class rs_oracle_function_class
set username muser
set password mpwd
set dsi_dataserver_make to 'ora'
set dsi_dataserver_type to 'oci'
set batch to 'off'
set dsi_proc_as_rpc to 'on'
go
```

7. create object パーミッションを Replication Server の *rs\_username* に付与します。

```
grant create object to rs_username
go
```

ここで、*rs\_username* は、Replication Agent が Replication Server へのアクセスに使用するユーザログイン名です。

8. Replication Agent の単一インスタンスでは、*ra\_admin\_owner*、*ra\_admin\_prefix*、*ra\_admin\_instance\_prefix*、*rs\_source\_ds*、および *rs\_source\_db* パラメータを設定します。

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ril
ra_config rs_source_ds, pds
ra_config rs_source_db, pdb
```

パラメータの説明は次のとおりです。

- *ra\_user\_1* は、Replication Agent インスタンスが使用する目的で、プライマリデータベース内での共有オブジェクトおよびインスタンスオブジェクトの作成に使用されるユーザ名です。このユーザ名は、プライマリデータサーバで定義済みであることが必要です。

- `ra_` は、プライマリデータベースで共有オブジェクトを識別するために使用されるプレフィクスです。このプレフィクスは、3文字以下です。
- `ri1` は、レプリケーショングループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
- `rs_source_ds` の値は `rs_source_db` の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用するコネクション名を形成します。

### 9. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

### 10. Replication Agent の他のインスタンスでは、`ra_admin_owner`、`ra_admin_prefix`、`ra_admin_instance_prefix`、`rs_source_ds`、および `rs_source_db` パラメータを設定します。

```
ra_config ra_admin_owner, ra_user_1  
ra_config ra_admin_prefix, ra_  
ra_config ra_admin_instance_prefix, ri2  
ra_config rs_source_ds, pds  
ra_config rs_source_db, conn2
```

パラメータの説明は次のとおりです。

- `ri2` は、レプリケーショングループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
- `ra_admin_owner` の値と `ra_admin_prefix` の値は、レプリケーショングループ内の他のすべての Replication Agent インスタンスと同じです。
- `rs_source_ds` の値は `rs_source_db` の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用するコネクション名を形成します。

### 11. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

### 12. 手順 1 の 2 つのテーブルに複写対象としてマークを付けます。 `ri1` で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab1, mark  
go
```

`ri2` で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab2, mark  
go
```

ここで、`ptab1` および `ptab2` は、複写されるプライマリデータベーステーブルです。

### 13. プライマリ Oracle データベースに対し、2 つの複写定義を作成します。たとえば、`ptab1` テーブルの `ptab1_repdef` 複写定義を作成するには、次を実行します。

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

ptab2 テーブルの ptab2\_repdef 複写定義を作成するには、次を実行します。

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

---

**注意：**これらの複写定義には、デフォルトのプライマリコネクション名を使用する必要があります。

---

プライマリ接続と代替プライマリ接続が異なる Replication Server に存在する場合、それぞれの Replication Server に複写定義を作成します。

#### 14. Replication Agent のインスタンスを再開します。

```
resume
```

Replication Agent インスタンスの再開に失敗した場合、LogMiner がインストールされ設定されているかを確認します。『Replication Agent プライマリデータベースガイド』の「Replication Agent for Oracle」の「Oracle 固有の考慮事項」の「Oracle トランザクションおよびオペレーションのトラブルシューティング」の「ra\_dumptran と ra\_helpop を使用するための Replication Agent と Oracle の設定」を参照してください。

#### 15. デフォルトの複写接続に対し、サブスクリプションを作成します。

たとえば、ptab1\_repdef 複写定義の ptab1\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at tns_alias_name.rdb
without materialization
go
```

#### 16. 代替複写接続に対し、サブスクリプションを作成します。

たとえば、ptab2\_repdef 複写定義の ptab2\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at tns_alias_name.conn2
without materialization
go
```

#### 参照：

- レプリケートデータサーバとしての Oracle (133 ページ)

## Oracle から SAP IQ への Multi-Path Replication

Oracle プライマリから SAP IQ レプリケートへのエンドツーエンドの複写のために、2つのプライマリとレプリケートの接続を持つ Multi-Path Replication システムを設定します。

### 前提条件

このシナリオでは、次がインストールされ設定済みであることを前提としています。

- LogMiner のあるプライマリ Oracle データベース
- 2つの Replication Agent インスタンス
- Replication Server
- レプリケート SAP IQ データベース

### 手順

1. プライマリ Oracle データベースで、2つのレプリケーションパスを介して複写するテーブルを2つ作成または選択します。
2. プライマリ Oracle データベースへのデフォルト接続を作成します。

```
create connection to pds.pdb
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

各パラメータの意味は、次のとおりです。

- *pds* は、Replication Agent で指定された **rs\_source\_ds** パラメータの値です。
  - *pdb* は、Replication Agent で指定された **rs\_source\_db** の値です。
  - *muser* は、プライマリ Oracle データベースのメンテナンスユーザです。
  - *mpwd* は、メンテナンスユーザのパスワードです。
3. プライマリ Oracle データベースへの代替接続を作成します。

```
create alternate connection to pds.pdb
named pds.conn2
set error class rs_sqlserver_error_class
set function string class rs_oracle_function_class
set username muser
set password mpwd
with primary only
go
```

4. 接続プロファイルを使用して、レプリケート SAP IQ データベースへの接続を作成します。

```
create connection to rds.rdb
using profile rs_oracle_to_iq;standard
set username muser
set password mpwd
go
```

各パラメータの意味は、次のとおりです。

- *rds* は、レプリケート SAP IQ サーバの名前です。
- *rdb* は、レプリケート SAP IQ データベースです。
- *muser* は、レプリケート SAP IQ データベースのメンテナンスユーザです。
- *mpwd* は、レプリケート SAP IQ のメンテナンスユーザのパスワードです。

5. レプリケート SAP IQ データベースへの代替接続を作成します。

```
create alternate connection to rds.rdb
named rds.conn2
using profile rs_oracle_to_iq;standard
set username muser2
set password mpwd
go
```

- 作成する SAP IQ データベースへの各代替接続に対し、異なるメンテナンスユーザ名を使用する必要があります。異なる Replication Server から SAP IQ データベースへの接続を作成する場合でも、ユニークなメンテナンスユーザ名を使用していることを確認します。
- 代替接続でのレプリケート SAP IQ サーバに対して、以前使用した名前とは異なる名前を使用する場合、Replication Server の interfaces ファイルへの異なる名前のエントリが必要です。

6. create object パーミッションを Replication Server の **rs\_username** に付与します。

```
grant create object to rs_username
go
```

ここで、*rs\_username* は、Replication Agent が Replication Server へのアクセスに使用するユーザログイン名です。

7. Replication Agent の単一インスタンスでは、**ra\_admin\_owner**、**ra\_admin\_prefix**、**ra\_admin\_instance\_prefix**、**rs\_source\_ds**、および **rs\_source\_db** パラメータを設定します。

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri1
ra_config rs_source_ds, pds
ra_config rs_source_db, pdb
```

パラメータの説明は次のとおりです。

- `ra_user_1` は、Replication Agent インスタンスが使用する目的で、プライマリデータベース内での共有オブジェクトおよびインスタンスオブジェクトの作成に使用されるユーザ名です。このユーザ名は、プライマリデータサーバで定義済みであることが必要です。
- `ra_` は、プライマリデータベースで共有オブジェクトを識別するために使用されるプレフィクスです。このプレフィクスは、3文字以下です。
- `ri1` は、レプリケーショングループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
- `rs_source_ds` の値は `rs_source_db` の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用するコネクション名を形成します。

### 8. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

### 9. Replication Agent の他のインスタンスでは、`ra_admin_owner`、`ra_admin_prefix`、`ra_admin_instance_prefix`、`rs_source_ds`、および `rs_source_db` パラメータを設定します。

```
ra_config ra_admin_owner, ra_user_1
ra_config ra_admin_prefix, ra_
ra_config ra_admin_instance_prefix, ri2
ra_config rs_source_ds, pds
ra_config rs_source_db, conn2
```

パラメータの説明は次のとおりです。

- `ri2` は、レプリケーショングループ内でこの Replication Agent インスタンスを一意に識別するプレフィクスです。
- `ra_admin_owner` の値と `ra_admin_prefix` の値は、レプリケーショングループ内の他のすべての Replication Agent インスタンスと同じです。
- `rs_source_ds` の値は `rs_source_db` の値と組み合わせられると、この Replication Agent インスタンスが Replication Server への接続に使用するコネクション名を形成します。

### 10. Replication Agent インスタンスを初期化します。

```
ra_admin init
```

### 11. 手順 1 の 2 つのテーブルに複写対象としてマークを付けます。 `ri1` で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab1, mark
go
```

`ri2` で識別される Replication Agent インスタンス上では、次のようにします。

```
pdb_setreptable ptab2, mark
go
```



ここで、ptab1 および ptab2 は、複写されるプライマリデータベーステーブルです。

12. プライマリ Oracle データベースに対し、2つの複写定義を作成します。

たとえば、ptab1 テーブルの ptab1\_repdef 複写定義を作成するには、次を実行します。

```
create replication definition ptab1_repdef
with primary at pds.pdb
with all tables named 'ptab1'
...
go
```

ptab2 テーブルの ptab2\_repdef 複写定義を作成するには、次を実行します。

```
create replication definition ptab2_repdef
with primary at pds.pdb
with all tables named 'ptab2'
...
go
```

---

**注意：**これらの複写定義には、デフォルトのプライマリコネクション名を使用する必要があります。

---

プライマリ接続と代替プライマリ接続が異なる Replication Server に存在する場合、それぞれの Replication Server に複写定義を作成します。

13. Replication Agent のインスタンスを再開します。

```
resume
```

Replication Agent インスタンスの再開に失敗した場合、LogMiner がインストールされ設定されているかを確認します。『Replication Agent プライマリデータベースガイド』の「Replication Agent for Oracle」の「Oracle 固有の考慮事項」の「Oracle トランザクションおよびオペレーションのトラブルシューティング」の「ra\_dumptran と ra\_helpop を使用するための Replication Agent と Oracle の設定」を参照してください。

14. デフォルトの複写接続に対し、サブスクリプションを作成します。

たとえば、ptab1\_repdef 複写定義の ptab1\_sub サブスクリプションを作成するには、次を実行します。

```
create subscription ptab1_sub
for ptab1_repdef
with replicate at rds.rdb
without materialization
go
```

15. 代替複写接続に対し、サブスクリプションを作成します。

たとえば、ptab2\_repdef 複写定義の ptab2\_sub サブスクリプションを作成するには、次を実行します。

## 異機種間における Multi-Path Replication

```
create subscription ptab2_sub
for ptab2_repdef
with replicate at rds.conn2
without materialization
go
```

### 参照：

- レプリケートデータサーバとしての SAP IQ (145 ページ)

## Oracle に対する異機種ウォームスタンバイ

ウォームスタンバイアプリケーションは、データベースとそのバックアップコピーとして機能するデータベースで構成される 1 組のデータベースです。クライアントアプリケーションは「アクティブデータベース」を更新し、Replication Server はアクティブデータベースのコピーとして「スタンバイデータベース」を管理します。

アクティブデータベースで障害が発生した場合、またはアクティブデータベースやデータサーバをメンテナンスする必要がある場合は、スタンバイデータベースに切り替えると、クライアントアプリケーションがほとんど中断されることなく処理を再開できます。

スタンバイデータベースが常にアクティブデータベースとの一貫性を保つように、Replication Server では、アクティブデータベースのトランザクションログから取得したトランザクション情報が再生成されます。スタンバイデータベースへのデータの複写には、サブスクリプションは必要ありません。

### Oracle に対するウォームスタンバイの動作

---

ウォームスタンバイアプリケーションでは、Replication Server から 1 つの論理データベースへのコネクションとして、アクティブデータベースとスタンバイデータベースが複写システムに示されます。

このウォームスタンバイアプリケーションでは、次の処理が実行されます。

- クライアントアプリケーションが、アクティブデータベースでトランザクションを実行する。
- アクティブデータベースの Replication Agent が、トランザクションログからトランザクションを取得して Replication Server に転送する。
- Replication Server が、スタンバイデータベースでトランザクションを実行する。
- 場合によっては、Replication Server は送信先データベースやリモート Replication Server にもトランザクションをコピーする。

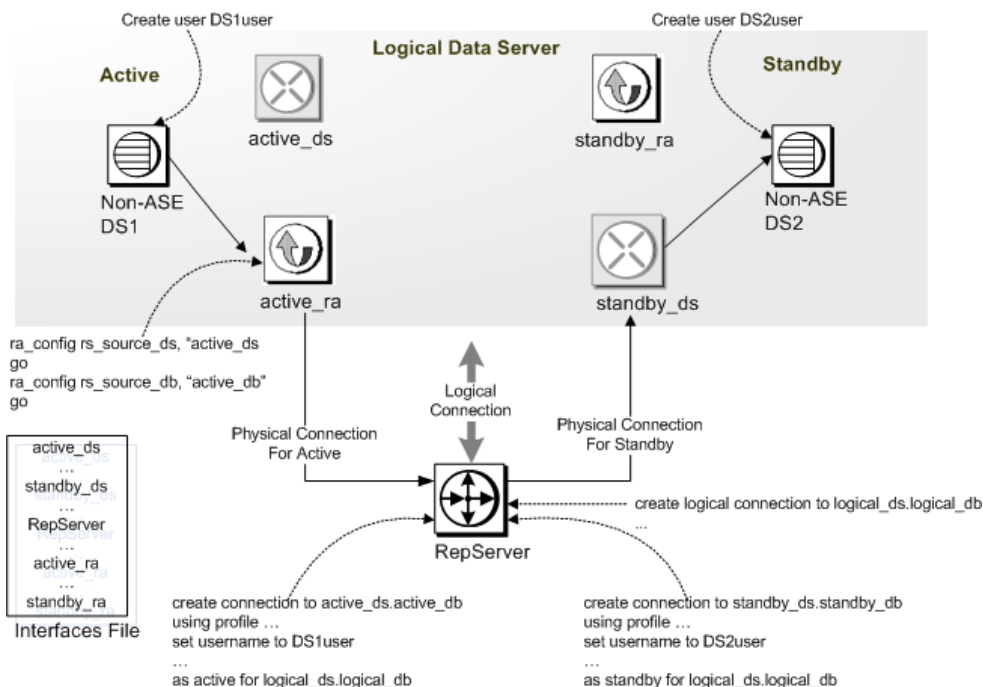
多くの Replication Server アプリケーションでは、次のことが当てはまります。

- プライマリデータベースは、複写定義とサブスクリプションを使用して他のデータベースにコピーされるデータの送信元である。
- 送信先データベースは、プライマリ (送信元) データベースからデータを受信する。

データベースコネクションの詳細については、『Replication Server 管理ガイド 第 2 巻』の「ウォームスタンバイアプリケーションの管理」を参照してください。

## ウォームスタンバイアプリケーション

ウォームスタンバイアプリケーションの一例における通常のオペレーションを示します。



## ウォームスタンバイの要件と制限

Replication Server のすべてのウォームスタンバイアプリケーションに対して適用されるいくつかの要件と制限を理解します。

- 1つの Replication Server で、アクティブデータベースとスタンバイデータベースの両方を管理します。アクティブデータベースとスタンバイデータベースは、いずれも、同じベンダで作成されたものでなければなりません。
- Replication Server は、クライアントアプリケーションをスタンバイデータベースに切り替えません。
- アクティブデータベースとスタンバイデータベースのためのデータサーバは、それぞれ別のマシンで実行します。アクティブデータベースとスタンバイデータベースを、同じデータサーバまたはハードウェアリソース上で管理すると、ウォームスタンバイ機能の利点が損なわれます。

- Replication Server は、異なるプラットフォーム間のウォームスタンバイ複写をサポートしていません。
- SAP では、アクティブデータベースとスタンバイデータベース内のテーブルにプライマリーキーを定義することをおすすめします。
- ウォームスタンバイではデータ消失が発生しないとは限りません。データが Replication Server によって正常に受信され、永続化される前、またはデータがレプリケートデータベースに書き込まれる前に、プライマリデータベース上でデータ破損が発生する可能性があります。
- 異機種ウォームスタンバイでは Oracle のすべてのデータ型と機能がサポートされるわけではありません。サポートされていない機能またはデータ型をデータベースで使用すると、プライマリデータとレプリケートデータの整合性が失われる可能性があります。サポートされていない機能とデータ型については、『Replication Agent プライマリデータベースガイド』>「Replication Agent for Oracle」>「Oracle 固有の考慮事項」を参照してください。

## スタンバイデータベースを管理するためのファンクション文字列

---

Replication Server は、スタンバイデータベースへの接続であるスタンバイ DSI に対して、システム提供ファンクション文字列クラス `rs_oracle_function_class` を使用します。

ファンクション文字列クラスには、次が含まれます。

- **rs\_marker** - スタンバイ接続のために複写を有効化する必要がある、アクティブデータベースのトランザクションログにおけるポイントをマーク付けします。マーカより後のものがすべて複製されている間、マーカより前のものはすべて複製されません。
- **rs\_repl\_off** - 現在のセッションの複写を無効にします。
- **rs\_triggers\_reset** - 現在のセッションにトリガを起動するレプリケートデータベースのトリガをすべて無効にします。
- **rs\_trunc\_set** - Replication Agent for Oracle によって使用されるトランケーションポイントをトランザクションログの末尾に移動します。

## ウォームスタンバイアプリケーションの複写情報

---

Replication Agent では、いくつかの方法で Oracle スタンバイデータベースへの複写を有効にできます。

Replication Server がスタンバイデータベースにコピーする情報のレベルとタイプは選択するメソッドによって異なりますが、次のいずれかを選択します。

- **pdb\_setrepddl** - データベースに格納されているシステムテーブルを変更する DDL コマンドとプロシージャの複写ができるようにします。DDL コマンドを使用すると、テーブルやビューなどのデータベースオブジェクトを作成、変更、削除できます。サポートされる DDL システムプロシージャは、データベースオブジェクトに関する情報に影響を与え、DDL ユーザによってスタンバイデータベースで実行されます。
- **pdb\_setreptable** - すべてのユーザテーブルまたは指定されたテーブルに複写のマークを付けます。

オプションで、スタンバイデータベースに複写するユーザストアドプロシージャを Replication Agent に指示するために **pdb\_setrepproc** を使用することもできます。

Replication Agent for Oracle の設定パラメータに関する詳細については、『Replication Agent リファレンスマニュアル』の「設定パラメータ」を参照してください。

## ウォームスタンバイデータベースの設定

---

ウォームスタンバイアプリケーションのためのデータベースの設定

### 前提条件

データベースを設定する前に、次の作業を実行します。

- アクティブデータベースとスタンバイデータベースの両方を管理する Replication Server をインストールします。1つの Replication Server で、アクティブデータベースとスタンバイデータベースの両方を管理します。
- Oracle に接続するには、ECDA または ExpressConnect for Oracle を設定します。ECDA を使用する場合、スタンバイサイトとプライマリサイトのそれぞれで1つのコピーを実行し、Oracle データベースと通信するように設定する必要があります。
- Replication Agent を設定し、アクティブデータベースとスタンバイデータベースの両方に対して管理モードで実行していることを確認します。
- アクティブデータベースとスタンバイデータベースの両方に、DDL ユーザ名を定義し、両方の Replication Agent で構成されていることを確認します。**ddl\_username** パラメータは、スタンバイデータベースまたはターゲットデータベースに DDL コマンドを複製するために、LTL に含まれるデータベースのユーザ名です。このユーザには、ターゲットデータベースで複写されたすべての DDL コマンドを実行するパーミッションが必要です。DDL ユーザは、メンテナンసుユーザとは異なる必要があります。また、DDL のユーザは、アクティブデータベースで実行したユーザとして DDL コマンドを実行するには、

**alter session** パーミッションも必要です。**ddl\_password** パラメータは、データベースユーザ名に対応するパスワードです。

## 手順

1. アクティブデータベースとスタンバイデータベースの両方によって使用されるために、1つの論理コネクションを作成します。
2. Replication Server **create connection** コマンドを使用して、複製システムにアクティブなデータベースを追加します。アクティブデータベースは、すでに複製システムに追加されている場合は追加する必要はありません。
3. Replication Server **create connection** コマンドを使用して、複製システムにスタンバイデータベースを追加します。

## 論理コネクションの作成

ウォームスタンバイアプリケーションの論理コネクションを作成します。

1. **sa** パーミッションを持つログイン名を使用して、ウォームスタンバイデータベースを管理する Replication Server にログインします。
2. **create logical connection** コマンドを実行します。

*logical\_ds.logical\_db*

```
create logical connection to logical_ds.logical_db
```

データサーバ名とデータベース名には、有効なオブジェクト名を指定できます。一般的な値には、論理コネクションを物理 Oracle 実装に関連付けるために、Oracle システム識別子 (SID) が含まれることもあります。

## 論理コネクション名の指定

*logical\_ds.logical\_db* を使用して、論理コネクションに名前を付けます。

論理コネクションに名前を付けるメソッドは、次のように2つあります。

- アクティブデータベースが複製システムに追加されていない場合、論理コネクションには、アクティブデータベースとは異なる名前を使用します。論理コネクションと物理コネクションにユニークな名前を使用すると、アクティブデータベースを簡単に切り替えられます。
- アクティブデータベースがすでに複製システムに追加されている場合、論理コネクション名には、アクティブデータベースの *data\_server* 名と *database* 名を使用します。論理コネクションは、この物理データベースを参照する既存の複製定義とサブスクリプションをすべて継承します。

ウォームスタンバイアプリケーションに対して複製定義またはサブスクリプションを作成する場合は、物理コネクションではなく論理コネクションを指定します。論理コネクションを指定すると、Replication Server は現在のアクティブデータベースを参照できます。

## アクティブデータベースへの ReplicationAgent の初期化

isql を使用して、Replication Agent for Oracle (RAO) のインスタンスを開始し、接続します。

1. 次を実行して、ソース Oracle データベースのアーカイブログファイルを設定します。

```
ra_config pdb_include_archives, true
go
ra_config pdb_archive_path, <path-to-oracle-archive-directory>
go
```

2. 次を実行して、Replication Agent のプライマリデータベースへの接続を設定します。

```
ra_config pds_host_name, <the host name of the source oracle>
go
ra_config pds_port_number <the port number of the source oracle>
go
ra_config pds_database_name,<the source oracle database name>
go
ra_config pds_username, <the oracle user for Replication Agent>
go
ra_config pds_password, <password>
go
test_connection PDS
go
```

接続が正常に確立されると、次のような出力が表示されます。

```
Type      Connection
```

```
-----  -
```

```
PDS      succeeded
```

3. 次を実行して、Replication Agent の Replication Server への接続を設定します。

```
ra_config rs_host_name, <the host name of the Replication Server>
go
ra_config rs_port_number, <the port number of the
Replication Server>
go
ra_config rs_username, <the Replication Server user for
Replication Agent>
go
ra_config rs_password, <password>
go
ra_config rs_source_ds ', ' <the DCON server name>
go
ra_config rs_source_db ', ' <the source oracle database name>
go
```



**注意：** ExpressConnect for Oracle を使用している場合、DirectConnect サーバ名を Oracle インスタンスの名前に置き換えます。例：

```
ra_config rs_source_ds, 'ordb'
go
rs_config rs_source_db, 'ordb'
go
```

4. 次を実行して、Replication Agent の ERSSD への接続を設定します。

```
ra_config rssid_host_name <the host name of the ERSSD>
go
ra_config rssid_port_number, <the port number of the ERSSD>
go
ra_config rssid_username, <the ERSSD user for
Replication Agent>
go
ra_config rssid_password, <password>
go
ra_config rssid_database_name, <the database name of the ERSSD>
go
test_connection RS
go
```

接続が正常に確立されると、次のような出力が表示されます。

```
Type      Connection
```

```
-----
```

```
RS          succeeded
```

5. Replication Server の文字セットが Replication Agent のものと同じではない場合、次を実行して、Replication Server の文字セットを更新します。

```
ra_config rs_charset, <the charset of the Replication Server>
```

6. 次を実行して、複製のマークが付けられた各テーブルに複製定義を作成します。

```
ra_config pdb_auto_create_repdefs, true
go
```

7. 次を実行して、ユーザテーブルの自動的なマーク付けを設定します。

```
ra_config pdb_automark_tables, true
go
```

8. 次を実行して、Replication Agent トランザクションログを初期化します。

```
pdb_xlog init
```

9. アクティブデータベースに対して DDL 複製を有効にします。次のように入力します。

```
pdb_setrepddl enable
```

---

**注意：**DDL 複写が有効であっても、一部の DDL コマンドはフィルタされます。DDL 複写を有効にする場合、**ddl\_password** および **ddl\_username** も設定する必要があります。

---

10. 次を実行して、Replication Agent が初期化される前に、作成されたテーブルに対して複写定義を作成します。

```
rs_create_repdef all
go
```

---

**注意：**複写システムにすでに追加しているデータベースをアクティブデータベースとして指定する場合、論理コネクションを作成するときにアクティブデータベースの RepAgent はサスペンドされます。

- 次を実行して、Replication Agent をレジュームします。

```
resume
go
```

---

## 複写システムへのアクティブデータベースの追加

アクティブデータベースへのコネクションを作成します。

1. 適切なパーミッションを持つログイン名を使って Replication Server にログインします。
2. 次のコマンドを実行します。

```
create connection to active_ds.active_db
using profile ...
set username to ...
set password to ...
with log transfer on
as active for logical_ds.logical_db
```

ExpressConnect for Oracle を使用している場合は、次を実行します。

```
create connection to ordb.ordb/*oracle data server name. database
name*/
using profile rs_oracle_to_oracle;eco
set username to ...
set password to ...
with log transfer on
as active for logical_ds.logical_db
go
```

または、ECDA を使用している場合は、次を実行します。

```
create connection to dco2active.ordb/*dco instance name.database
name*/
using profile rs_oracle_to_oracle;ecda
set username to ...
set password to ...
with log transfer on
```

```
as active for logical_ds.logical_db
go
```

## スタンバイデータベースの初期化

ダンプとロードの手法を使用して、アクティブデータベースのデータでスタンバイデータベースを初期化します。

アクティブデータベースからのデータのダンプ、およびスタンバイデータベースへのロードの方法については、Oracle のマニュアルを参照してください。

## スタンバイデータベースへの ReplicationAgent の初期化

**isql** を使用して、Replication Agent for Oracle (RAO) のインスタンスを開始し、接続します。

1. 次を実行して、スタンバイ Oracle データベースのアーカイブログファイルのパスを設定します。

```
ra_config pdb_include_archives, true
go
ra_config pdb_archive_path, <path-to-oracle-archive-directory>
go
```

2. 次を実行して、Replication Agent のスタンバイデータベースへの接続を設定します。

```
ra_config pds_host_name, <the host name of the standby oracle>
go
ra_config pds_port_number <the port number of the standby oracle>
go
ra_config pds_database_name, <the standby oracle database name>
go
ra_config pds_username, <the oracle user for Replication Agent>
go
ra_config pds_password, <password>
go
test_connection PDS
go
```

接続が正常に確立されると、次のような出力が表示されます。

```
Type      Connection
```

```
-----
```

```
PDS      succeeded
```

3. 次を実行して、Replication Agent の Replication Server への接続を設定します。

```
ra_config rs_host_name, <the host name of the Replication Server>
go
ra_config rs_port_number, <the port number of the
Replication Server>
go
```

## Oracle に対する異機種ウォームスタンバイ

```
ra_config rs_username, <the Replication Server user for
Replication Agent>
go
ra_config rs_password, <password>
go
ra_config rs_source_ds ', '<the DCON server name>
go
ra_config rs_source_db ', '<the standby oracle database name>
go
```

**注意：** ExpressConnect for Oracle を使用している場合、 DirectConnect サーバ名を Oracle インスタンスの名前に置き換えます。例：

```
ra_config rs_source_ds, 'ordb'
go
rs_config rs_source_db, 'ordb'
go
```

4. 次を実行して、Replication Agent の ERSSD への接続を設定します。

```
ra_config rssid_host_name <the host name of the ERSSD>
go
ra_config rssid_port_number, <the port number of the ERSSD>
go
ra_config rssid_username, <the ERSSD user for
Replication Agent>
go
ra_config rssid_password, <password>
go
ra_config rssid_database_name, <the database name of the ERSSD>
go
test_connection RS
go
```

接続が正常に確立されると、次のような出力が表示されます。

```
Type      Connection
```

```
-----
```

```
RS          succeeded
```

5. Replication Server の文字セットが Replication Agent のものと同じではない場合、次を実行して、Replication Server の文字セットを更新します。

```
ra_config rs_charset, <the charset of the Replication Server>
```

6. 次を実行して、複写のマークが付けられた各テーブルに複写定義を作成します。

```
ra_config pdb_auto_create_repdefs, true
go
```

7. 次を実行して、ユーザテーブルの自動的なマーク付けを設定します。

```
ra_config pdb_automark_tables, true
go
```

8. 次を実行して、Replication Agent トランザクションログを初期化します。

```
pdb_xlog init
```

9. アクティブデータベースに対して DDL 複写を有効にします。

```
pdb_setrepddl enable
```

**注意：**DDL 複写が有効であっても、一部の DDL コマンドはフィルタされます。DDL 複写を有効にする場合、**ddl\_password** および **ddl\_username** も設定する必要があります。

10. スタンバイモードで動作するように、Replication Agent を設定します。スタンバイモードで動作するように、**ra\_standby** 設定パラメータを“true”に設定します。

```
ra_config ra_standby, 'true'
go
```

## スタンバイデータベースへの接続の作成

スタンバイデータベース接続の作成

1. 適切なパーミッションを持つログイン名を使って Replication Server にログインします。
2. 次のコマンドを実行します。

```
create connection to standby_ds.standby_db
using profile ...
set username to ...
set password to ...
with log transfer on
as standby for logical_ds.logical_db
```

## アクティブデータベースとスタンバイデータベースへの接続のレジューム

アクティブデータベースとスタンバイデータベースへの接続をレジュームします。

スタンバイデータベースの初期化中、Replication Server は、アクティブデータベースへの接続をサスペンドします。

1. 次のコマンドを Replication Server で実行します。

```
resume connection to active_ds.active_db
```

2. アクティブデータベースとスタンバイデータベースへの接続をレジュームした後、次を実行して、ウォームスタンバイのステータスを確認します。

```
admin logical_status [,logical_ds,logical_db]
go
```

## アクティブデータベースとスタンバイデータベースへの Replication Agents のレジューム

アクティブデータベースとスタンバイデータベースの Replication Agents をレジュームして、複製するためにトランザクションのデータベースログのスキャンを開始します。

各 Replication Agent では、次を実行します。

```
resume  
go
```

## アクティブデータベースとスタンバイデータベースの切り替え

アクティブデータベースで障害が発生した場合またはアクティブデータベースでメンテナンスを実行する場合、スタンバイデータベースに切り替えます。

1. Replication Server で、次のように入力します。

```
switch active for logical_ds.logical_db  
to standby_ds.standby_db
```

切り替え時の Replication Server の動作については、「内部での切り替え手順」を参照してください。

2. 切り替えの進行状況をモニタするには、次を実行します。

```
admin logical_status, logical_ds, logical_db
```

Operation in Progress 出力カラムと State of Operation in Progress 出力カラムに、切り替えステータスが表示されます。

3. アクティブデータベースの切り替えが完了すると、次を実行して、Replication Server のアクティブデータベースへの接続をレジュームします。

```
resume connection to active_ds.active_db
```

4. まだサスペンドされていない場合は、元のアクティブサイトで Replication Agent をサスペンドします。次を実行して、スタンバイモードに設定します。

```
ra_config ra_standby,true
```

5. 元のスタンバイモードで Replication Agent は自動的にサスペンドされ、スタンバイモードから複製モードに変更されます。確認するには、次を実行します。

```
ra_config ra_standby
```

戻り値を false に設定する必要があります。

6. アクティブサイトとスタンバイサイトで両方の Replication Agent をレジュームします。

---

**注意：** Replication Server が切り替えの途中で停止した場合は、Replication Server を再起動すると、切り替えがレジュームします。スタンバイサイトで Replication Agent をレジュームすると、自動的に Replication Agent システムデータベース (RASD) は更新されます。

---

**参照：**

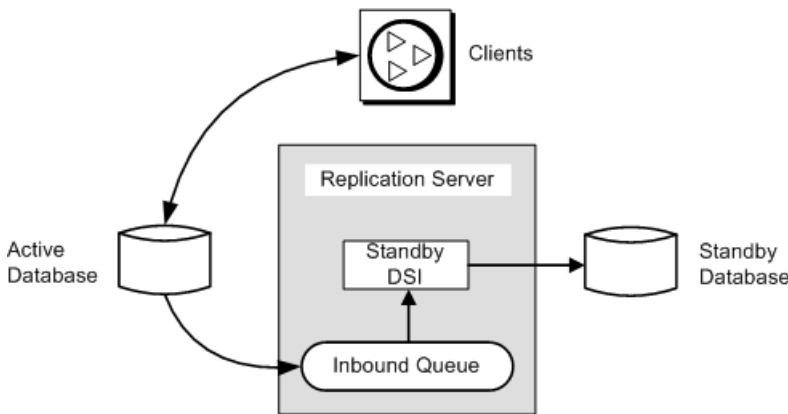
- 内部での切り替え手順 (262 ページ)

## アクティブデータベースとスタンバイデータベースを切り替える前

ウォームスタンバイアプリケーション自体のアクティビティを介する場合以外は、複製システムに関与しないデータベースに対するウォームスタンバイアプリケーションを示します。

アクティブデータベースとスタンバイデータベースを切り替える前の、ウォームスタンバイアプリケーションの通常のオペレーションを示しています。

図 12：ウォームスタンバイアプリケーション - 切り替え前



図：ウォームスタンバイアプリケーション - 切り替え前には、次を示す内部詳細が含まれます。

- Replication Server は、アクティブデータベースから受信したトランザクションを、インバウンドメッセージキューに書き込みます。
- このインバウンドキューはスタンバイデータベースの DSI スレッドによって読み取られ、このスレッドによってスタンバイデータベースでトランザクションが実行されます。

アクティブデータベースから受信したメッセージは、スタンバイ DSI スレッドがそのメッセージを読み取ってスタンバイデータベースに適用するまで、インバウンドキューからトランケートできません。

この例では、トランザクションは、アクティブデータベースからスタンバイデータベースに単純に複製されます。論理データベース自体では、次のことは行われません。

- レプリケートデータベースまたはリモート Replication Server に複製されるプライマリデータの保持
- 別の Replication Server からの複製トランザクションを受信します。

### 内部での切り替え手順

内部での切り替え手順について説明します。

アクティブデータベースとスタンバイデータベースを切り替えると、Replication Server は次を実行します。

1. アクティブおよびスタンバイのコネクションに対し、**suspend log transfer** コマンドを発行します。
2. インバウンドキューに残されているすべてのメッセージを読み取り、スタンバイデータベースに適用します。サブスクリプションデータまたは複製ストアドプロシージャについては、アウトバウンドキューに適用します。  
切り替えが完了する前に、インバウンドキュー内のコミットされたトランザクションがすべて処理されます。
3. スタンバイ DSI をサスペンドします。
4. 新しいアクティブデータベースのトランザクションログにマーカを設定します。Replication Server はこのマーカを使用して、新しいスタンバイデータベースや任意のレプリケートデータベースに適用するトランザクションを決定します。
5. 新しいアクティブデータベースに対するコネクションを停止します。インバウンドキューをパージし、rs\_oqid テーブルへの最後の oqid をフラッシュし、rs\_locator テーブルをリセットします。Replication Server セグメントのフラグをリセットし、新しいスタンバイ DSI をサスペンドします。
6. アクティブデータベースとスタンバイデータベースの両方の **ptype** パラメータを更新します。Replication Server は、古いアクティブデータベースを対象とするサブスクリプションに有効のマークを、新しいアクティブデータベースのサブスクリプションに無効のマークを付けます。
7. 新しいアクティブデータベースのコネクションをレジュームし、そのデータベースに対するログ転送をレジュームして新しいメッセージを受信できるようにします。アクティブデータベースとスタンバイデータベースの両方の DSI をレジュームします。

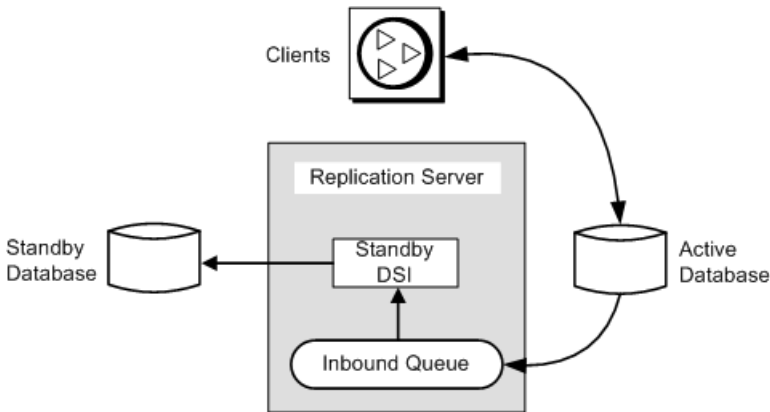


## アクティブデータベースとスタンバイデータベースを切り替えたあと

アクティブデータベースからスタンバイデータベースに切り替えたあとの、関連するプロセス、およびウォームスタンバイ環境でのコンポーネントのステータスについて説明します。

アクティブデータベースとスタンバイデータベースのロールを切り替えると、複製システムは次の図に示すように変更されます。

図 13: ウォームスタンバイアプリケーションの例 - 切り替え後



- 以前のスタンバイデータベースが新しいアクティブデータベースになります。クライアントアプリケーションは、新しいアクティブデータベースに切り替えられています。
- この例では、以前のアクティブデータベースが新しいスタンバイデータベースになります。以前のアクティブデータベースに対するメッセージは、新しいアクティブデータベースに適用するためにキューイングされます。

**注意:** 切り替え後、前のアクティブデータベースの Replication Agent が停止し、新しいアクティブデータベースの Replication Agent が起動されます。

## ウォームスタンバイアプリケーションのモニタリング

Replication Server ログファイルまたは **admin** コマンドを使用して、ウォームスタンバイアプリケーションをモニタできます。

*複製を使用するウォームスタンバイアプリケーション*

複製に関与するウォームスタンバイアプリケーションの場合、論理データベースは、複製システムのプライマリデータベースまたはレプリケートデータベースとして機能します。

複写を使用するウォームスタンバイアプリケーションの詳細については、『Replication Server 管理ガイド 第2巻』の「ウォームスタンバイアプリケーションの管理」を参照してください。

## 複写定義およびサブスクリプション

---

Replication Agent は、初期化中に自動的に複写定義を作成 (`pdb_xlog init` コマンドの実行) し、Oracle ウォームスタンバイのために `pdb_auto_create_repdefs` 設定パラメータを `true` に設定します。

Replication Agent が論理コネクションを (RSSD に問い合わせることで) 検出する場合、Replication Agent が作成した複写定義は Oracle ウォームスタンバイ環境をサポートするようにカスタマイズされます。

ウォームスタンバイ環境を非ウォームスタンバイデータベースに複写する特定のシナリオでは、複写予定の各テーブルまたはストアドプロシージャに対して、2 番目の複写定義を作成する必要があります。

### 追加のウォームスタンバイデータベースの複写定義

---

ウォームスタンバイプライマリ環境からウォームスタンバイ環境外のレプリケートデータベースに複写する場合、複写される各テーブルに対して新しい複写定義を作成することができます。

Replication Agent が初期化時に自動的に作成した複写定義には、次のような属性があります。

- Oracle データ型の Replication Server ユーザ定義データ型 (UDD) へのマッピングが提供されます。
- デフォルトで、`clob/blob` カラムを持つテーブルは `always_replicate` 句で定義されます。`auto_create_repdefs` が “on” に設定されている場合、`clob/blob` カラムは `replicate_if_change` 句で定義されます。

---

**注意：** `always_replicate` および `replicate_if_change` は、複写定義を作成するための句です。

---

- `send standby replication definition columns` 句で、複写定義は作成されます。

次のケースの場合、追加の複写定義を作成できます。

- Oracle データ型から Replication Server UDD へのマッピングを提供する。
- `clob/blob` カラムを持つテーブルのために、`replicate_if_change` を使用する。
- データベースレベルのサブスクリプションが非ウォームスタンバイデータベースへのサブスクリプション作成のために使用される場合、`send standby all columns` 句を含める。

- プライマリとレプリケートファンクションの所有者を指定する。ユーザプロセスのターゲット (スタンバイデータベース) ファンクションの所有者情報を指定するために、ファンクション文字列をカスタマイズする。

たとえば、ユーザテーブル TB1 が COL5 カラムの 1 つで、Oracle のデータ型 date として定義される場合、正常にカラムをスタンバイデータベースに複写するには、ユーザは次のように複写定義を作成する必要があります。

```
create replication definition repl
with primary at ordb.pdb
with all tables named 'USER1'.'TB1'
(
"COL1" int,
"COL2" int,
"COL3" int,
"COL4" char(255),
"COL5" rs_oracle_datetime,
)
primary key( "COL1","COL2","COL3")
searchable columns( "COL1","COL2","COL3","COL5")
send standby replication definition columns
replicate minimal columns
go
```

この例では、**create replication definition** コマンドの **send standby replication definition columns** 句は、この複写定義がサブスクリプションを作成するデータベース、およびスタンバイデータベースのために使用できることを指定します。

#### 参照：

- データ型の変換とマッピング (285 ページ)

## ウォームスタンバイアプリケーションでのサブスクリプション

**create subscription** コマンドと **define subscription** コマンドは、物理名ではなく、論理データベース名と論理データサーバ名を使用します。

アクティブデータベースからスタンバイデータベースへの複写にサブスクリプションは使用されませんが、次のことは可能です。

- 論理プライマリデータベースのデータに対してサブスクリプションを作成する。
- 他のデータベースのデータを論理レプリケートデータベースに複写するためにサブスクリプションを作成する。

複写を使用するウォームスタンバイアプリケーションの詳細については、『**Replication Server 管理ガイド 第2巻**』の「ウォームスタンバイアプリケーションの管理」を参照してください。

## アップグレードの考慮事項

---

Oracle ウォームスタンバイ機能にとって、アップグレード後に実行する必要があるという、特別な指示は不要です。

Replication Server のバージョンをアップグレードする情報については、ご使用のプラットフォームの『Replication Server 設定ガイド』の「Replication Server のアップグレードまたはダウングレード」を参照してください。

## ダウングレードの考慮事項

---

ダウングレードした後、ファンクション文字列のマッピングやデータ変換が実行できないように、Oracle スタンバイコネクションは中断されます。RSSD をダウングレードする前に、スタンバイコネクションを削除しなかった場合は、ダウングレード後に削除することができます。

`rs_init` による RSSD のダウングレードが完了した後、Oracle データサーバーへのコネクション (Oracle へのコネクションが `create connection` と `using profile` 句を使用して作成される場合) はダウンする可能性があります。理由は、Replication Server 15.5 で提供されている `wait_after_commit` 設定パラメータが使用できなくなったため、複製処理をレジュームする必要があるからです。

## ダウングレード後の複製のレジューム

---

RSSD をダウングレードした後の複製のレジュームについて説明します。

複製をレジュームするには、次を実行します。

1. 次のコマンドを実行します。

```
alter connection to data_server.database
set dsi_serialization_method to 'wait_for_commit'
go
```

2. アクティブデータベースへのログ転送をレジュームします。
3. Replication Agent for Oracle をレジュームします。

上記の手順を完了したら、アクティブデータベースから、ダウングレードの前に作成された複製定義とサブスクリプションを持つ他のレプリケートデータベースへの複製を開始することができます。

## Oracle レプリケートデータベースの再同期

Replication Server を使用すると、レプリケートデータベースを再同期してマテリアライズできます。また、プライマリデータベースのクワイズを強いることなく、データの損失や整合性を失うリスクなしで複写をレジュームできます。

データベース再同期化は、信頼されたソースから取得したデータダンプを同期先のデータベースに適用することをベースとしています。

データベースの再同期には、この機能をサポートしているデータベース用の Replication Agent のバージョンが必要です。Replication Agent のための特定のコマンドについては、Replication Agent のマニュアルを参照してください。

### 製品の互換性

Oracle データベースの再同期をサポートしている Oracle、Replication Agent for Oracle、および ExpressConnect for Oracle のバージョンを使用します。

Replication Server Options のマニュアルを参照してください。

表 4 : Oracle データベースの再同期の製品互換性

| データベースサーバのバージョン           | Replication Agent のバージョン | ExpressConnect のバージョン          |
|---------------------------|--------------------------|--------------------------------|
| Oracle Server 10g、11g、12c | 15.5                     | ExpressConnect 15.5 for Oracle |

### データベースの再同期の設定

Oracle データベースの再同期を設定します。

1. Replication Agent をサスペンドして複写プロセスを停止します。
2. Replication Server を再同期モードにします。再同期モードになると、Replication Server はトランザクションをスキップします。さらに、プライマリデータベースまたは信頼されたソースから取得したダンプを使ってレプリケートデータベースにデータを再移植する準備として、複写キューから複写データをパージします。
3. プライマリデータベースからダンプを取得します。
4. Replication Agent を再開して、データベース再同期マーカを Replication Server に送り、再同期処理が進行中であることを示します。

Replication Server がプライマリデータベースダンプが完了したことを示すダンプマーカを検出すると、Replication Server はトランザクションのスキップを停止し、どのトランザクションをレプリケートデータベースに適用するかを判定できるようになります。

5. ダンプをレプリケートデータベースに適用します。
6. レプリケートデータベースを再初期化します。
7. 複写をレジュームします。

データベースの再同期では、Replication Server と Replication Agent for Oracle の両方からのコマンドとパラメータを使用する必要があります。

### 参照：

- データベース再同期化シナリオ (277 ページ)

## Replication Server にトランザクションをスキップさせる

`resume connection` コマンドに `skip to resync` パラメータを指定して、Replication Agent から送られたダンプデータベースマーカを受け取り認識するまで、指定されたレプリケートデータベースで DSI アウトバウンドキュー内のトランザクションをスキップするよう Replication Server に指示します。

レプリケートデータベース内のデータはダンプの内容によって置き換えられることになっているので、Replication Server はアウトバウンドキュー内のレコードの処理をスキップします。

『Replication Server リファレンスマニュアル』の「Replication Server コマンド」の「`resume connection`」を参照してください。

次に示すコマンドを実行します。

```
resume connection to data_server.database  
[skip [n] transaction | execute transaction | skip to resync marker]
```

**警告！** `resume connection` を `skip to resync marker` オプションを付けて間違った接続で実行すると、レプリケートデータベースのデータが非同期となります。

`skip to resync marker` を設定すると、Replication Server は Replication Server ログ内またはデータベース例外ログ内でスキップされたトランザクションをログに記録しません。`skip [n] transaction` を設定すると、Replication Server はスキップされたトランザクションをログに記録します。

## データベース再同期マーカを Replication Server に送信する

Replication Agent for Oracle を設定または Replication Agent for Oracle に指示して、再同期処理が進行中であることを示すデータベース再同期マーカを Replication Server に送信することができます。

再同期モードで Replication Agent を再開すると、Replication Agent は再同期データベースマーカを最初のメッセージとして Replication Server へ送信してから、SQL データ定義言語 (DDL: data definition language) またはデータ操作言語 (DML: data manipulation language) のトランザクションを送信します。同じプライマリデータベースの複数のレプリケートデータベースはそれぞれに DSI アウトバウンドキューがあるので、すべて同じ再同期マーカを受け取ります。

**skip to resync marker** パラメータでレジュームする各 DSI に対して、DSI が再同期マーカを受け取ったことが、DSI アウトバウンドキューによって Replication Server システムログに記録されます。また、その時点からダンプデータベースマーカを受け取るまで、DSI がコミットされたトランザクションを拒否することも記録されます。

Replication Agent for Oracle では、データベース再同期マーカの送信で各オプションをサポートするために、**resync** または **resync, init** のパラメータを含む **resume** コマンドを使用します。『Replication Agent リファレンスマニュアル』の「Command Reference」の「**resume**」を参照してください。

### 再同期マーカの送信

Replication Agent は自動的にトランケーションポイントが変更になっているかどうかを判断することができます。

次のような場合、**resume resync** を使用して、再同期マーカを送信することができます。

- トランケーションポイントに変更がなく、Replication Agent が最後に処理したところからトランザクションログの処理を続けることになっている場合。各アウトバウンド DSI スレッドとキューはデータベース再同期マーカを受け取り処理します。再同期マーカを受け取ったとき、DSI は skip to resync マーカ要求に従って Replication Server システムログにレポートを送ります。その後、ダンプデータベースマーカを待つ間、DSI はコミットされたトランザクションを拒否します。この時点で、このメッセージと、ダンプデータベースマーカを待つ動作の変更によって、レプリケートデータベースにダンプを適用できるようになります。
- プライマリデータベースのトランケーションポイントが移動されている場合。トランケーションポイントを手動で変更すると、移動は起きます。この状況下では、resync marker を送信する前に、Replication Agent レポジトリを初期化する Replication Agent で **ra\_init force** を実行します。この初期化では、ト

ランケーションポイントを移動し、一部のトランザクションログレコードの処理をスキップするので、Replication Agent は消失の可能性があるデータベース内の変更すべてを追跡します。

トランケーションポイントが変更されると、Replication Server のインバウンドキュー内にあるオープントランザクションは、新しいトランケーションポイントから送られたアクティビティと一致しないため、パージされる必要があります。変更されたトランケーションポイントが以前のオリジンキュー ID (OQID) を持つレコードを送信する可能性があるため、Replication Server は重複検出をリセットします。以前のデータがキューからパージされると、Replication Server は Replication Agent からのどの新しいアクティビティも重複アクティビティとして扱いません。したがって、新しいアクティビティが拒否されることはありません。マーカを受け取るまで、Replication Server はアウトバウンドキューコマンドを拒否し続けるので、パージオプションは DSI の処理を変更しません。

### init コマンドによる再同期マーカの送信

再同期マーカを **init** コマンド付きで送信するには、**resume resync, init** を使用します。これによって、インバウンドキュー内のすべてのオープントランザクションをパージして重複の検出をリセットし、アウトバウンド DSI をサスペンドするよう、Replication Server に指示できます。

このオプションはプライマリデータベースにレプリケートデータベースと同じダンプを再ロードするときに使用します。プライマリデータベースから取得したダンプはないので、Replication Agent はダンプデータベースマーカを送りません。再同期マーカの後に来るダンプデータベースマーカを待つ代わりに、**init** オプションは Replication Server が再同期マーカを処理したらすぐに DSI コネクションをサスペンドします。

DSI がサスペンドされたら、それ以降 DSI を通るすべてのアクティビティは新しいトランザクションのみになります。プライマリで使用したダンプをレプリケートデータベースに再ロードしたら、DSI をレジュームできます。

### 参照：

- プライマリデータベースとレプリケートデータベースを同じダンプから再同期 (282 ページ)

## データベースのダンプを取得する

**dump** ユーティリティを使用して、データベースのダンプを取得します。

ダンプが完了した後、管理者として、ダンプが行われたときにプライマリデータベースから取得された情報を基に、希望するダンプポイントを判断する必要があります。**dump** ユーティリティはダンプポイントを提供する可能性があります。あとの項のシナリオでは、Oracle RMAN ユーティリティを使用しています。



Oracle では、**backup database plus archivelog** を使用してプライマリデータベースをダンプし、**restore database** と **recover database** を使用してレプリケートデータベースにダンプを適用します。1つの RMAN バックアップセットからダンプポイントを取得するには、**list backup** Oracle コマンドを使用します。**list backup** からの出力の例は次のとおりです。

```
RMAN>list backup;
List of Backup Sets
=====
```

| BS Key | Size    | Device Type | Elapsed Time | Completion Time |
|--------|---------|-------------|--------------|-----------------|
| 8      | 125.58M | DISK        | 00:00:04     | 16-MAY-11       |

```

BP Key: 8      Status: AVAILABLE Compressed: NO Tag:
TAG20110516T125049
Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
Obmcflp9_1_1

List of Archived Logs in backup set 8
-----
Thrd Seq      Low SCN      Low Time     Next SCN     Next Time
-----
1      1            1018110     14-MAY-11   1058201     15-MAY-11
1      2            1058201     15-MAY-11   1103370     15-MAY-11
1      3            1103370     15-MAY-11   1142662     16-MAY-11
1      4            1142662     16-MAY-11   1148674     16-MAY-11
1      5            1148674     16-MAY-11   1150375     16-MAY-11
1      6            1150375     16-MAY-11   1150477     16-MAY-11

```

| BS Key | Type | LV | Size  | Device Type | Elapsed Time | Completion Time |
|--------|------|----|-------|-------------|--------------|-----------------|
| 9      | Full |    | 1.08G | DISK        | 00:00:15     | 16-MAY-11       |

```

BP Key: 9      Status: AVAILABLE Compressed: NO Tag:
TAG20110516T125054
Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
0cmcflpe_1_1

List of Datafiles in backup set 9
File LV Type Ckp SCN      Ckp Time Name
-----
1      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
system01.dbf
2      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
sysaux01.dbf
3      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
undotbs01.dbf
4      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
users01.dbf
5      Full 1150485     16-MAY-11 /work2/oracle11.1/oradata/or11sh1/
example01.dbf

```

| BS Key | Type | LV | Size  | Device Type | Elapsed Time | Completion Time |
|--------|------|----|-------|-------------|--------------|-----------------|
| 10     | Full |    | 9.36M | DISK        | 00:00:04     | 16-MAY-11       |

```

BP Key: 10     Status: AVAILABLE Compressed: NO Tag:

```

## Oracle レプリケートデータベースの再同期

```
TAG20110516T125054
  Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
0dmcflq1_1_1
  SPFILE Included: Modification time: 14-MAY-11
  SPFILE db_unique_name: OR11SH1
  Control File Included: Ckp SCN: 1150507          Ckp time: 16-MAY-11

BS Key   Size          Device Type Elapsed Time Completion Time
-----
11       18.50K        DISK          00:00:04      16-MAY-11
  BP Key: 11      Status: AVAILABLE Compressed: NO  Tag:
TAG20110516T125118
  Piece Name: /ralinuxsh5/oracle/product/11.1.0/db_2/dbs/
0emcflq6_1_1

List of Archived Logs in backup set 11
Thrd Seq      Low SCN      Low Time     Next SCN     Next Time
-----
1     7           1150477     16-MAY-11  1150513     16-MAY-11
1     8           1150513     16-MAY-11  1150568     16-MAY-11
```

必要なダンプポイントは、最後の Archived Logs バックアップセット内の「Next SCN」カラムの最大値から 1 を引いた値です。この例では、最後のセットがセット 11 で、セット 11 内の「Next SCN」の最大値が 1150568 です。したがって、この例のダンプポイントは 1150567 です。

RMAN ユーティリティと **list backup** コマンドに関する情報については、Oracle のマニュアルを参照してください。

### ダンプデータベースマーカを Replication Server に送信する

データサーバインタフェーススレッド (DSI) は、**skip to resync marker** を使用してレジュームされた後、再同期モードになっていて、再同期モードで Replication Agent を再起動すると、再同期データベースマーカの後に受け取ったダンプデータベースマーカは DSI をサスペンドし、DSI コネクションの既存の再同期状態を削除します。

同じプライマリデータベースの複数のレプリケートデータベースは、すべて同じダンプデータベースマーカを受け取ります。In Replication Agent では、**oracle scn** パラメータを設定した **lr\_dump\_marker** コマンドを使用して、ダンプデータベースマーカは Replication Agent 設定に基づいて送信されます。Replication Agent のマニュアルを参照してください。

---

**注意：** **init** 付き **resync** を使用して Replication Agent を再起動すると、DSI は、再起動データベースマーカを受け取った後、すぐにサスペンドされます。DSI は、ダンプマーカを待たずに、サスペンドされます。

---

レプリケートデータベースにダンプを適用したら、手動で DSI をレジュームできます。DSI はコミットされたトランザクションを拒否しなくなります。また、ダ

ンプデータベースマーカが示すダンプポイントの後にコミットされたトランザクションはすべて複写されます。

## DSI スレッド情報をモニタする

データベースの再同期中に DSI についての情報を提供するには、**admin who** コマンドを使用します。

| 状態               | 説明                                                                             |
|------------------|--------------------------------------------------------------------------------|
| SkipUntil Resync | <b>skip to resync</b> を実行した後 DSI がレジュームする。このステータスは DSI がデータベース再同期マーカを受け取るまで続く。 |
| SkipUntil Dump   | DSI がデータベース再同期マーカを受け取った。このステータスは DSI がその後のダンプデータベースマーカを受け取るまで続く。               |

## 再同期するデータベースにダンプを適用する

プライマリデータベースのダンプをレプリケートデータベースに適用できるのは、メッセージがシステムログに表示された後だけです。

メッセージは次のとおりです。

- Replication Server が **resync database** マーカを受け取ると、次のようなメッセージが出力されます。  

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```
- Replication Server が **init** マーカ付きの再同期データベースを受け取る時のメッセージ。  

```
DSI for data_server.database received and processed
Resync Database Marker. DSI is now suspended. Resume after
database has been reloaded.
```
- Replication Server がダンプデータベースマーカを受け取る時のメッセージ。  

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after database has been
reloaded.
```

再同期するデータベースにダンプをロードする方法の詳細については、Oracle のマニュアルを参照してください。

## レプリケートデータベースの再初期化

プライマリデータベースまたはダンプソースからレプリケートデータベースにダンプを適用したら、レプリケートデータベースを再初期化して、ダンプが削除したユーザ、テーブル、およびパーミッションをリストアします。

## Oracle レプリケートデータベースの再同期

1. メンテナンスユーザおよび DDL ユーザがプライマリデータベースに存在しない場合、プライマリデータベースからダンプを適用した後、レプリケートデータベースにユーザを追加します。
2. **hds\_oracle\_new\_setup\_for\_replicate.sql** スクリプトをレプリケートデータベースで実行して、関連する Replication Server システムテーブルをレプリケートデータベースに追加します。このスクリプトは、UNIX の場合は \$SYBASE/scripts ディレクトリ、Windows の場合は %SYBASE%\scripts ディレクトリにあります。  
スクリプトは、レプリケートデータベースに関連する値を挿入し、必要なパーミッションを付与します。

### Oracle レプリケート初期化スクリプト

レプリケート Oracle でメンテナンスユーザ ID を使用して **hds\_oracle\_rep\_init.sql** スクリプトを適用します。

```
drop table rs_info;
commit;

create table rs_info (rskey varchar2 (20), rsval varchar2 (20));
commit;

insert into rs_info values ('charset_name', 'iso_1');
commit;

insert into rs_info values ('sortorder_name', 'bin_iso_1');
commit;

drop public synonym rs_lastcommit;
commit;

drop table rs_lastcommit;
commit;

create table rs_lastcommit(
    origin number(8),origin_qid char(72 byte),
    secondary_qid char(72 byte),
    origin_time date,
    dest_commit_time date,
    conn_id number default 0);
commit;

grant all on rs_lastcommit to public;
commit;

create public synonym rs_lastcommit for rs_lastcommit;
commit;

CREATE OR REPLACE PROCEDURE RS_UPDATE_SEQUENCE
(
    SequenceName VARCHAR2,
    SequenceValue NUMBER,
```

```

Increment NUMBER
)
AS
CurrentID NUMBER;
LastID NUMBER;
SeqCursor INTEGER;
SQLStmt VARCHAR2(1024);
Result NUMBER;
BEGIN
SQLStmt := 'SELECT ' || SequenceName || '.NEXTVAL FROM DUAL';
SeqCursor := DBMS_SQL.OPEN_CURSOR;DBMS_SQL.PARSE(SeqCursor,
SQLStmt,
DBMS_SQL.NATIVE);
DBMS_SQL.DEFINE_COLUMN(SeqCursor,
1,
LastID);
Result := DBMS_SQL.EXECUTE_AND_FETCH(SeqCursor);
DBMS_SQL.COLUMN_VALUE(SeqCursor,1,CurrentID);
LOOP
IF ( Increment < 0 ) THEN
EXIT WHEN CurrentID <= SequenceValue;
EXIT WHEN CurrentID > LastID;
ELSE
EXIT WHEN CurrentID >= SequenceValue;
EXIT WHEN CurrentID < LastID;
END IF;
LastID := CurrentID;
Result := DBMS_SQL.EXECUTE_AND_FETCH(SeqCursor);
DBMS_SQL.COLUMN_VALUE(SeqCursor,1,CurrentID);
END LOOP;
DBMS_SQL.CLOSE_CURSOR(SeqCursor);
END;
/

grant all on RS_UPDATE_SEQUENCE to public;
commit;

DROP sequence rs_ticket_seq;

CREATE sequence rs_ticket_seq;

Drop table rs_ticket_history;

CREATE TABLE rs_ticket_history (
cnt numeric(8,0),
h1 varchar(10),
h2 varchar(10),
h3 varchar(10),
h4 varchar(50),
pdb varchar(30),
prs varchar(30),
rrs varchar(30),
rdb varchar(30),
pdb_t varchar(128),
exec_t varchar(128),
dist_t varchar(128),

```

## Oracle レプリケートデータベースの再同期

```
        rsi_t varchar(128),
        dsi_t varchar(128),
        rdb_t varchar(128) default current_date,
        exec_b numeric(22,0),
        rsi_b numeric(22,0),
        dsi_tnx numeric(22,0),
        dsi_cmd numeric(22,0),
        ticket varchar(1024),
        conn_id number NULL);

create unique index rs_ticket_idx on rs_ticket_history(cnt);

create or replace trigger rs_ticket_tri
before insert on rs_ticket_history for each row
begin
    if :new.cnt is null then
        select rs_ticket_seq.nextval into :new.cnt from dual;
    end if;
end rs_ticket_tri;
/

grant all on rs_ticket_history to public;
commit;

create or replace package RS_TRIGGER_CONTROL
as
    procedure ENABLE;
    function IS_ENABLED
        return boolean;
    end;
/

create or replace package body RS_TRIGGER_CONTROL
as
    fire_trigger boolean := TRUE;
    function IS_ENABLED return boolean as
    begin
        if fire_trigger then
            return FALSE;
        else
            return TRUE;
        end if;
    end IS_ENABLED;

    procedure ENABLE
    as
    begin
        fire_trigger := FALSE;
    end ENABLE;

    procedure DISABLE
    as
    begin
        fire_trigger := TRUE;
    end DISABLE;
end RS_TRIGGER_CONTROL;
```

```

/
grant all on rs_trigger_control to public;

drop public SYNONYM rs_trigger_control;

create public SYNONYM rs_trigger_control for rs_trigger_control;
commit;

CREATE OR REPLACE PROCEDURE rs_send_repserver_cmd(rs_api IN
VARCHAR2)
AS
  cmd VARCHAR2 (4000);
  plsql_block VARCHAR2(50);
  BEGIN
    if (instr(lower(rs_api), 'rs_rcl') > 0) then
      DBMS_OUTPUT.put_line('The Replication Server command
should not contain the keyword ''rs_rcl''');
      return;
    end if;
    cmd := 'rs_rcl ' || replace(rs_api, ' ', ' ') || ' '
rs_rcl';
    if ('rs_rcl' != substr(cmd, length(cmd) - 5)) then
      DBMS_OUTPUT.put_line('The Replication Server command is
too long. ');
      DBMS_OUTPUT.put_line('Please split it into two or more
commands');
      return;
    end if;
    plsql_block := 'begin rs_marker(:a); end;';
    EXECUTE IMMEDIATE plsql_block USING cmd;
  END;
/

drop public SYNONYM rs_send_repserver_cmd;

create public SYNONYM rs_send_repserver_cmd for
rs_send_repserver_cmd;
commit;

```

## データベース再同期化シナリオ

データベースの再同期手順はそのシナリオによって異なります。再同期手順を完了すると、プライマリデータベースとレプリケートデータベースはトランザクションの一貫性が保たれた状態になります。

手順の実行には次の要件があります。

- 複写システム管理者であること。
- 正常に稼動する複写環境が存在すること。
- プライマリデータベースからレプリケートデータベースへデータをコピーするためのメソッドやプロセスがあること。

コマンドの構文については、次を参照してください。

- Replication Agent for Oracle - 『Replication Agent リファレンスマニュアル』を参照してください。
- Replication Server - 『Replication Server リファレンスマニュアル』を参照してください。

### 1つ以上のレプリケートデータベースをプライマリデータベースから直接再同期する

1つ以上のレプリケートデータベースを1つのプライマリデータベースから再同期します。

この手順では、多少の違いはありますが、次のことを実行できます。

- プライマリデータベースとレプリケートデータベース間の複製の遅延時間が、複製によるデータベースの回復が不可能で、複製データに基づくレポートの作成が実用的でなくなった場合に、レプリケートデータベースにデータを再移植します。
- プライマリデータベースから信頼されたデータをレプリケートデータベースに再移植します。
- プライマリデータベースが複数のレプリケートデータベースのソースになっている場合に、再同期を調整します。
- プライマリサイトが一对のウォームスタンバイデータベースで構成されている論理データベースであり、それに1つまたは複数のレプリケートデータベースを再同期する場合に、再同期を調整します。ウォームスタンバイのペアでは、アクティブデータベースがプライマリデータベースとして、スタンバイがレプリケートデータベースとして機能します。したがって、1つまたは複数のレプリケートデータベースからはアクティブデータベース(プライマリサイトのウォームスタンバイペアの1つ)がプライマリデータベースに見えます。

### プライマリデータベースから直接再同期する

レプリケートデータベースをプライマリデータベースから直接再同期します。

1. トランケーションポイントを移動するには、次のコマンドを実行します。

```
ra_locator move truncpt
ra_admin refresh
isql -USAMPLE_RS_RSSD_prim -PSAMPLE_RS_RSSD_prim_ps
-SSAMPLE_RS_ERSSD -DSAMPLE_RS_ERSSD
rs_zeroltm NY, NYora92
go
```

トランケーションポイントを変更しない場合は、手順2に進みます。

2. Replication Agent による複製処理を停止します。Replication Agent で次のコマンドを実行します。



```
suspend
```

- レプリケートデータベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to dataserwer.database
```

- レプリケートデータベースのアウトバウンドキューからデータを削除し、プライマリデータベースの Replication Agent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to  
resync marker
```

- データベースのマニュアルの指示に従ってプライマリデータベースの内容のダンプを取得します。
- Recovery Manager (RMAN) for Oracle を使用する場合は、Oracle の **list backup** コマンドを使用して RMAN バックアップの最後の System Change Number (SCN) を取得します。次に、Replication Agent で、**lr\_dump\_marker** の値としてこの SCN を設定します。

```
lr_dump_marker oracle scn
```

- 再同期モードで Replication Agent を起動し、再同期マーカを Replication Server に送信します。

```
resume resync  
go
```

- Replication Server システムログで、次のメッセージを検索して、DSI が Replication Agent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed  
Resync Database Marker. Waiting for Dump Marker.
```

DSI がレプリケートデータベースの再同期マーカを処理すると、レプリケートデータベースにダンプを適用できます。

---

**注意：** 複数のデータベースを再同期する場合は、再同期する各データベースの DSI コネクションが再同期マーカを受け入れていることを確認します。

---

- データベースのマニュアルの指示に従ってプライマリデータベースのダンプをレプリケートデータベースに適用します。

- Replication Server システムログで次のメッセージを検索して、Replication Server がダンプデータベースマーカを処理していることを確認します。

```
DSI for data_server.database received and processed  
Dump Marker. DSI is now suspended. Resume after database has been  
reloaded.
```

Replication Server がダンプマーカを受け取ると、DSI コネクションが自動的にサスペンドされます。

11. メンテナンスユーザおよび DDL ユーザがプライマリデータベースに存在しない場合は、プライマリデータベースからダンプを適用した後、これらのユーザをレプリケートデータベースに追加します。
12. `hds_oracle_new_setup_for_replicate.sql` スクリプトをレプリケートデータベースで実行して、`rs_info` テーブルと `rs_lastcommit` テーブルをレプリケートデータベースに追加します。  
スクリプトは、レプリケートデータベースに関連する値を挿入し、必要なパーミッションを付与します。
13. レプリケートデータベースにダンプを適用したら、次のコマンドを使用して DSI をレジュームします。

```
resume connection to data_server.database
```

### サードパーティダンプユーティリティを使用して再同期する

ディスクスナップショットなどのサードパーティの **dump** ユーティリティを使って、プライマリデータベースをダンプした後、再同期を調整します。

サードパーティツールでは、プライマリデータベースとのやり取りをネイティブのデータベースダンプユーティリティほど密接には行うことができません。

Replication Agent がダンプデータベースマーカの生成に使用できるような記録をサードパーティツールがプライマリデータベースのトランザクションログに書き込まない場合は、独自のダンプデータベースマーカを生成して再同期処理を完了できるようにします。詳細については、サードパーティツールのマニュアルを参照してください。

1. Replication Agent による複写プロセスを停止します。トランケーションポイントを変更しないでください。Replication Agent で次のコマンドを実行します。

```
suspend
```

2. レプリケートデータベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to dataserver.database
```

3. レプリケートデータベースのアウトバウンドキューからデータを削除し、プライマリデータベースの Replication Agent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to  
resync marker
```

4. トランケーションポイントが移動されていない場合は、手順 5 に進んでください。それ以外の場合は、Replication Agent レポジトリを再初期化してからプライマリデータベースのコンテンツのダンプを取得します。Replication Agent で次のコマンドを実行します。

```
ra_init force
go
```

5. サードパーティユーティリティを使用してプライマリデータベースのコンテンツのダンプを取得します。
6. サードパーティユーティリティからダンプまたは情報を取得したら、プライマリデータベースからの情報に基づいてダンプポイントを決定します。サードパーティユーティリティを使用する場合、ユーザはダンプポイントを決定する責任があります。たとえば、ディスク複写ツールを使用する場合、プライマリデータベースでアクティビティを一時的に停止してディスクスナップショットから実行中のトランザクションを消去し、ダンプデータベースマーカとして「トランザクションログの末尾」ポイントを使用できます。
7. 手順5で取得したダンプの位置の末尾にマークを付けるには、Replication Agent のプライマリデータベースでストアドプロシージャを実行します。

```
lr_dump_marker oracle scn
```

8. 再同期モードで Replication Agent を再起動し、再同期マーカを Replication Server に送信します。

```
resume resync
go
```

Replication Agent は、手順6で取得して手順7で設定したダンプ位置の末尾に基づいてダンプデータベースマーカを一度に自動的に生成し、ダンプデータベースマーカを Replication Server に送信します。

9. Replication Server システムログで次のメッセージを検索して、DSI が Replication Agent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

10. データベースとサードパーティユーティリティのマニュアルの指示に従って、サードパーティツールからプライマリデータベースのダンプをレプリケートデータベースに適用します。
11. Replication Server システムログで次のメッセージを検索して、Replication Server がダンプデータベースマーカを処理していることを確認します。

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after
database has been reloaded.
```

Replication Server がダンプマーカを受け取ると、DSI コネクションが自動的にサスペンドされます。

12. メンテナンスユーザおよび DDL ユーザがプライマリデータベースに存在しない場合は、プライマリデータベースからダンプを適用した後、これらのユーザをレプリケートデータベースに追加します。
13. hds\_oracle\_new\_setup\_for\_replicate.sql スクリプトをレプリケートデータベースで実行して、rs\_info テーブルと rs\_lastcommit テーブルをレプリケートデータベースに追加します。

スクリプトは、レプリケートデータベースに関連する値を挿入し、必要なパーミッションを付与します。

14. レプリケートデータベースにダンプを適用したら、次のコマンドを使用して DSI をレジュームします。

```
resume connection to data_server.database
```

### プライマリデータベースとレプリケートデータベースを同じダンプから再同期

再同期を調整して、プライマリデータベースとレプリケートデータベースを同じダンプまたはデータのコピーから再ロードします。プライマリデータベースからダンプを取得しないので、ダンプデータベースマークは必要ありません。

1. Replication Agent による複製プロセスを停止します。トランケーションポイントを変更しないでください。Replication Agent で次のコマンドを実行します。

```
suspend
```

2. レプリケートデータベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to data_server.database
```

3. レプリケートデータベースのアウトバウンドキューからデータを削除し、プライマリデータベースの Replication Agent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to  
resync marker
```

4. プライマリデータベースに外部ソースからのデータダンプを適用します。

5. プライマリデータベースのトランザクションログの最後までトランケーションポイントを移動します。Replication Agent で次のコマンドを実行します。

```
pdb_xlog move_truncpt  
go
```

6. プライマリデータベースからの最新のシステムデータに基づいて Replication Agent レポジトリを再初期化します。

```
ra_init force  
go
```

7. **init** オプションを指定して、再同期モードで起動するよう Replication Agent に指示します。Replication Agent で次のコマンドを実行します。

```
resume resync, init
```

8. Replication Server システムログで次のメッセージを検索して、DSI が Replication Agent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed  
Resync Database Marker. DSI is now suspended. Resume  
after database has been reloaded.
```

Replication Server が **init** マーカを使用して再同期データベースを受信して処理すると、DSI コネクションはサスペンドします。

9. レプリケートデータベースに外部ソースからのデータダンプを適用します。
10. メンテナンスユーザおよび DDL ユーザがプライマリデータベースに存在しない場合は、プライマリデータベースからダンプを適用した後、これらのユーザをレプリケートデータベースに追加します。
11. `hds_oracle_new_setup_for_replicate.sql` スクリプトをレプリケートデータベースで実行して、`rs_info` テーブルと `rs_lastcommit` テーブルをレプリケートデータベースに追加します。  
スクリプトは、レプリケートデータベースに関連する値を挿入し、必要なパーミッションを付与します。
12. レプリケートデータベースにダンプを適用したら、レプリケートデータベースで DSI をレジュームして Replication Server がプライマリデータベースからのトランザクションを適用できるようにします。

```
resume connection to data_server.database
```



## データ型の変換とマッピング

Replication Server には、サポートされる ASE 以外の各データサーバについて、あるデータ型から別のデータ型へのデフォルトのマッピングを定義するクラスレベル変換が用意されています。

次のデータ型の変換が用意されています。

- Adaptive Server データ型に直接対応しない、ASE 以外のデータ型
- ASE 以外のデータ型に直接対応しない、Adaptive Server データ型
- サポートされる ASE 以外の別のデータサーバのデータ型に直接対応しない、ASE 以外のデータ型

---

**注意：**別のデータサーバのデータ型に直接対応するデータ型には、クラスレベル変換は用意されていません。

---

### 同機種マッピング

---

Replication Server は、同じタイプとバージョンのデータサーバ間でデータ型の変換とマッピングをサポートします。

#### デフォルトのマッピング

Replication Server は、同じタイプとバージョンのデータサーバ間でサポート対象のすべてのデータ型の複写をサポートしています。デフォルトでは、Replication Server はプライマリデータベースでキャプチャされたすべてのデータ型のデータを、レプリケートデータベースの同じデータ型に複写します。

#### ユーザ定義マッピング

同じタイプとバージョンのデータサーバ間の複写については、次のように、デフォルトの Replication Server マッピングを無効にして、プライマリデータベースのデータ型からレプリケートデータベースの別のデータ型への複写を試みることができます。

- 複写定義内のカラムのパブリッシュデータ型を変更する、または
- カスタムファンクション文字列を使用する

---

**注意：**デフォルトのデータ型マッピングの動作を無効にする前に、これらのカスタマイズ機能とビジネス目標について十分理解しておいてください。

---

パブリッシュデータ型を変更する方法の1つとして、複写定義の **map to** 句を使用します。『Replication Server 管理ガイド 第1巻』>「データ型変換の使用開始にあたって」>「カラムレベル変換の作成」を参照してください。

## 異機種マッピング

---

Replication Server は、異なるタイプまたはバージョンのデータサーバ間でデータ型の変換とマッピングをサポートします。

Replication Server は、デフォルトでは、レプリケートデータベースのデータ型の長さ、小数点以下桁数、範囲、解釈と一致するデータ型をプライマリデータベースでサポートします。レプリケートデータベースのデータ型と一致しないプライマリデータ型はサポートされますが、カスタマイズが必要です。

### デフォルトのマッピング

Oracle の CHAR と SAP HANA データベース char のマッピングなどでプライマリとレプリケートのデータ型に不一致がある場合や、Oracle の CHAR と SAP HANA データベースの varchar のマッピングなどでレプリケートのデータ型がプライマリの制限を超過する場合でも、Replication Server は特別な調整を行いません。

Adaptive Server の datetime と SAP HANA データベースの timestamp のマッピングなどで、プライマリとレプリケートのデータ型の不一致がセマンティックまたは小数点以下桁数のみで発生する場合、Replication Server はデフォルトのクラスレベルの変換を実行し、プライマリのデータ型値をレプリケート データベースのデータ型値と一致させます。

### カスタムマッピング

異なるタイプまたはバージョンのデータサーバ間の複製の場合、Replication Server は、レプリケートデータベースと一致しないプライマリデータ型のデフォルトマッピングを行いません。次のような場合には、カスタムマッピングを実行する必要があります。

- レプリケートデータ型の長さ、精度、または範囲はプライマリデータ型のものより小さいため、データがトランケートされる場合があります。
- レプリケートデータサーバには、プライマリデータ型と同じ用途のデータ型がありません。たとえば、プライマリの間隔、シーケンス、XML 型、空間型に相当するデータ型がレプリケートデータサーバにはありません。

次のように Replication Server マッピングをカスタマイズして、プライマリデータベースのデータ型をレプリケートデータベースの別のデータ型に複製を試みることができます。

- 複製定義内のカラムのパブリッシュデータ型を変更する、または
- カスタムファンクション文字列を使用する

---

**注意：** マッピング動作をカスタマイズする前に、これらのカスタマイズ機能とビジネス目標を十分に理解しておいてください。

---



パブリッシュデータ型を変更する方法の1つとして、複写定義の **map to** 句を使用します。たとえば、INTERVAL 型の col2 という Oracle カラムを varchar 型の SAP HANA データベースカラムにマップするには、次を実行します。

```
col2 rs_oracle_interval map to varchar
```

このマッピングの予想結果をプレビューするには、Replication Server の **admin translate** コマンドを使用します。

```
1> admin translate, '+000000005 01:01:01.000000000',
rs_oracle_interval, 'varchar(40)'
2> go
```

| Delimiter Prefix | Translated Value              | Delimiter Postfix |
|------------------|-------------------------------|-------------------|
| '                | +000000005 01:01:01.000000000 | '                 |

『Replication Server 管理ガイド 第1巻』>「データ型変換の使用開始にあたって」>「カラムレベル変換の作成」を参照してください。

## DB2 データ型

データ型変換に関するこの情報は、メインフレーム環境 (IBM z/OS など)、あるいは UNIX および Microsoft Windows 環境の DB2 UDB に適用されます。

### Adaptive Server データ型から DB2 データ型への変換

Adaptive Server データ型から DB2 データ型へのクラスレベル変換を示します。

| Adaptive Server データ型 | DB2 データ型                 |
|----------------------|--------------------------|
| bigdatetime          | TIMESTAMP                |
| bigint               | BIGINT                   |
| bigtime              | TIMESTAMP                |
| binary               | CHAR FOR BIT DATA        |
| bit                  | TINYINT                  |
| date                 | DATE (UNIX と Windows のみ) |
| datetime             | TIMESTAMP                |
| decimal              | DECIMAL                  |
| int                  | NUMERIC                  |
| money                | NUMERIC                  |

| Adaptive Server データ型 | DB2 データ型                 |
|----------------------|--------------------------|
| numeric              | NUMERIC                  |
| real                 | REAL (UNIX と Windows のみ) |
| smalldatetime        | TIMESTAMP                |
| smallint             | NUMERIC                  |
| smallmoney           | NUMERIC                  |
| time                 | TIME (UNIX と Windows のみ) |
| tinyint              | NUMERIC                  |
| unsigned bigint      | DECIMAL (20,0)           |
| unsigned int         | BIGINT                   |
| unsigned smallint    | INTEGER                  |
| unsigned tinyint     | SMALLINT                 |
| unitext              | DBCLOB                   |
| varbinary            | VARCHAR FOR BIT DATA     |

### **DB2 データ型から Adaptive Server データ型への変換**

DB2 データ型から Adaptive Server データ型へのクラスレベル変換を示します。

| DB2 データ型                         | Adaptive Server データ型 |
|----------------------------------|----------------------|
| CHAR FOR BIT DATA                | binary               |
| DATE                             | datetime             |
| DECFLOAT UDB (UNIX と Windows のみ) | float                |
| DOUBLE (UNIX と Windows のみ)       | float                |
| REAL (UNIX と Windows のみ)         | real                 |
| TIME                             | datetime             |
| TIMESTAMP                        | datetime             |
| VARCHAR FOR BIT DATA             | varbinary            |

## DB2 から SAP HANA データベース

DB2 UDB から SAP HANA データベースへのデータ型マッピングの一覧を示します。

SAP HANA データベースデータ型で表すことができる値の範囲の現在の情報については、『SAP HANA Reference』 > 「SQL Reference Manual」 > 「Data Types」を参照してください。

表 5 : DB2 UDB から SAP HANA データベースへのデータ型マッピング

| DB2 UDB のデータ型             | 複写定義のデータ型                          | SAP HANA データ型        |
|---------------------------|------------------------------------|----------------------|
| BIGINT                    | rs_udb_bigint                      | bigint               |
| BLOB                      | rs_udb_blob                        | blob                 |
| CHAR                      | char                               | varchar<br>(下記の注を参照) |
| CHAR FOR BIT DATA         | rs_udb_char_for_bit (推奨)<br>binary | varbinary            |
| CLOB                      | text                               | clob                 |
| DATE                      | rs_udb_date                        | date                 |
| DBCLOB                    | unitext                            | nclob                |
| DECFLOAT (16)             | rs_udb_decfloat                    | double               |
| DECFLOAT (32)             | rs_udb_decfloat                    | decimal              |
| DECIMAL                   | decimal                            | decimal              |
| DOUBLE                    | rs_udb_double                      | double               |
| GRAPHIC                   | unichar                            | nvarchar             |
| INTEGER                   | rs_udb_bigint                      | integer              |
| LONG VARCHAR              | text                               | clob                 |
| LONG VARCHAR FOR BIT DATA | image                              | blob                 |
| LONG VARGRAPHIC           | unitext                            | nclob                |

| DB2 UDB のデータ型        | 複写定義のデータ型                               | SAP HANA データ型        |
|----------------------|-----------------------------------------|----------------------|
| REAL                 | rs_udb_real                             | real                 |
| SMALLINT             | smallint                                | smallint             |
| TIME                 | rs_udb_time                             | time                 |
| TIMESTAMP (12)       | rs_udb_timestamp12                      | timestamp            |
| VARCHAR              | varchar                                 | varchar<br>(下記の注を参照) |
| VARCHAR FOR BIT DATA | rs_udb_varchar_for_bit<br><br>varbinary | varbinary            |
| VARGRAPHIC           | univarchar                              | nvarchar             |

**注意：**プライマリデータベースが、拡張 ASCII 文字 (128 ~ 255 の 1 バイトの ASCII 文字値) を受け付けるシングルバイト文字セット (iso\_1 など) に対応するように設定されている場合、SAP HANA データベースでは char または varchar カラムでシングルバイトの拡張文字をダブルバイト文字に変換することがあります。この変換により、SAP HANA データベースの記憶領域サイズがプライマリデータベースの元のカラム長または記憶領域を超過する可能性があります。プライマリデータベースとアプリケーションが、シングルバイト文字セットでの拡張 ASCII 文字の使用をサポートしている場合は、SAP HANA データベースのテーブル定義で対応するカラムのサイズを増やす必要があります。この変更により、2 バイトの記憶領域への拡張文字の変換にカラムが対応できるようになります。

## DB2 データ型から Microsoft SQL Server データ型への変換

DB2 データ型から Microsoft SQL Server データ型へのクラスレベル変換を示します。

| DB2 データ型                         | Microsoft SQL Server データ型 |
|----------------------------------|---------------------------|
| CHAR FOR BIT DATA                | binary                    |
| DATE                             | datetime                  |
| DECFLOAT UDB (UNIX と Windows のみ) | float                     |
| DOUBLE (UNIX と Windows のみ)       | float                     |
| REAL (UNIX と Windows のみ)         | real                      |

| DB2 データ型             | Microsoft SQL Server データ型 |
|----------------------|---------------------------|
| TIME                 | datetime                  |
| TIMESTAMP            | datetime                  |
| VARCHAR FOR BIT DATA | varbinary                 |

## DB2 データ型から Oracle データ型への変換

DB2 データ型から Oracle データ型へのクラスレベル変換を示します。

| DB2 データ型                         | Oracle データ型  |
|----------------------------------|--------------|
| CHAR FOR BIT DATA                | RAW          |
| DATE                             | DATE         |
| DECFLOAT UDB (UNIX と Windows のみ) | FLOAT        |
| DOUBLE (UNIX と Windows のみ)       | FLOAT        |
| REAL (UNIX と Windows のみ)         | REAL         |
| TIME                             | DATE (時刻を含む) |
| TIMESTAMP                        | DATE (時刻を含む) |
| VARCHAR FOR BIT DATA             | RAW          |

## DB2 に対応する Replication Server のデータ型名

z/OS プラットフォーム上の DB2 データサーバの DB2 データ型を表す、Replication Server のユーザ定義データ型 (UDD) 名を示します。

表 6 : DB2 for z/OS データ型に対応する Replication Server UDD 名

| DB2 for z/OS データ型    | Replication Server の名前                        |
|----------------------|-----------------------------------------------|
| CHAR FOR BIT DATA    | <i>rs_db2_char_for_bit</i>                    |
| DATE                 | <i>rs_db2_date</i>                            |
| DECIMAL              | <i>rs_db2_decimal</i> 、 <i>rs_db2_numeric</i> |
| TIME                 | <i>rs_db2_time</i>                            |
| TIMESTAMP            | <i>rs_db2_timestamp</i>                       |
| VARCHAR FOR BIT DATA | <i>rs_db2_varchar_for_bit</i>                 |

## データ型の変換とマッピング

UNIX および Microsoft Windows プラットフォーム上の DB2 データサーバの DB2 データ型を表す、Replication Server UDD の名前を示します。

**表 7 : DB2 for UNIX and Windows データ型に対応する Replication Server UDD 名**

| DB2 for UNIX/Windows データ型   | Replication Server の名前        |
|-----------------------------|-------------------------------|
| <i>CHAR FOR BIT DATA</i>    | <i>rs_udb_char_for_bit</i>    |
| <i>DATE</i>                 | <i>rs_udb_date</i>            |
| <i>DECFLOAT</i>             | <i>rs_udb_decfloat</i>        |
| <i>DOUBLE</i>               | <i>rs_udb_double</i>          |
| <i>INTEGER</i>              | <i>rs_udb_bigint</i>          |
| <i>REAL</i>                 | <i>rs_udb_real</i>            |
| <i>TIME</i>                 | <i>rs_udb_time</i>            |
| <i>TIMESTAMP</i>            | <i>rs_udb_timestamp</i>       |
| <i>VARCHAR FOR BIT DATA</i> | <i>rs_udb_varchar_for_bit</i> |

## Microsoft SQL Server データ型

Microsoft SQL Server データ型のクラスレベル変換 (デフォルトのデータ型マッピング) と、Microsoft SQL Server データ型に対応する Replication Server のデータ型名について説明します。

### **Adaptive Server データ型から Microsoft SQL Server データ型への変換**

符号なしデータ型の、Adaptive Server データ型から Microsoft SQL Server データ型へのクラスレベル変換を示します。

Microsoft SQL Server データ型は Adaptive Server データ型と直接互換性があり、変換を必要としないため、Adaptive Server データ型から Microsoft SQL Server データ型への (または Microsoft SQL Server データ型から Adaptive Server データ型への) その他のクラスレベル変換は用意されていません。

表 8 : Adaptive Server データ型から Microsoft SQL Server データ型へのクラスレベル変換

| Adaptive Server データ型 | Microsoft SQL Server データ型 |
|----------------------|---------------------------|
| unsigned bigint      | DECIMAL (20,0)            |
| unsigned int         | BIGINT                    |
| unsigned smallint    | INT                       |
| unsigned tinyint     | SMALLINT                  |
| unitext              | NTEXT                     |

### Microsoft SQL Server データ型から DB2 データ型への変換

Microsoft SQL Server データ型から DB2 データ型へのクラスレベル変換を示します。

表 9 : Microsoft SQL Server データ型から DB2 データ型へのクラスレベル変換

| Microsoft SQL Server データ型 | DB2 データ型             |
|---------------------------|----------------------|
| binary                    | CHAR FOR BIT DATA    |
| bit                       | TINYINT              |
| datetime                  | TIMESTAMP            |
| decimal                   | DECIMAL              |
| money                     | NUMERIC              |
| numeric                   | NUMERIC              |
| smalldatetime             | TIMESTAMP            |
| smallmoney                | NUMERIC              |
| varbinary                 | VARCHAR FOR BIT DATA |

### Microsoft SQL Server から SAP HANA データベース

Microsoft SQL Server から SAP HANA データベースへのデータ型マッピングの一覧を示します。

SAP HANA データベースデータ型で表すことができる値の範囲の現在の情報については、『SAP HANA Reference』 > 「SQL Reference Manual」 > 「Data Types」を参照してください。

表 10 : Microsoft SQL Server から SAP HANA データベースへのデータ型マッピング

| Microsoft SQL Server データ型         | 複写定義のデータ型            | SAP HANA データベースデータ型                  |
|-----------------------------------|----------------------|--------------------------------------|
| bigdatetime                       | bigdatetime          | timestamp                            |
| bigint                            | rs_msss_bigint       | bigint                               |
| bigtime                           | bigtime              | デフォルトなし                              |
| binary                            | binary               | varbinary                            |
| bit                               | tinyint              | tinyint                              |
| char                              | char                 | varchar<br>(下記の注を参照)                 |
| date                              | date                 | date                                 |
| datetime                          | rs_msss_datetime     | timestamp                            |
| datetime2                         | rs_msss_datetime2    | timestamp                            |
| decimal                           | decimal<br>varchar   | decimal (34 桁まで)<br>varchar (34 桁以上) |
| double precision                  | float                | double                               |
| float                             | float                | float                                |
| identity                          | numeric              | numeric                              |
| integer                           | rs_msss_bigint       | bigint                               |
| image                             | image                | blob                                 |
| longsysname varchar(255) Not Null | varchar(255)         | varchar(255)                         |
| money                             | decimal              | decimal(19,4)                        |
| nchar                             | unichar              | nvarchar                             |
| nvarchar(max)                     | rs_msss_nvarchar_max | nclob                                |
| numeric                           | numeric              | decimal                              |
| nvarchar                          | varchar              | nvarchar                             |
| real                              | real                 | real                                 |



| Microsoft SQL Server<br>データ型    | 複写定義のデータ型                 | SAP HANA データベース<br>データ型 |
|---------------------------------|---------------------------|-------------------------|
| smalldatetime                   | rs_msss_datetime          | seconddate              |
| smallint                        | smallint                  | smallint                |
| smallmoney                      | decimal                   | decimal(10,4)           |
| sysname varchar(30)<br>Not Null | varchar(30)               | varchar(30)             |
| text                            | text                      | clob                    |
| timestamp                       | timestamp                 | varbinary(8)            |
| tinyint                         | tinyint                   | tinyint                 |
| time                            | rs_msss_time              | timestamp               |
| unichar                         | unichar                   | nchar                   |
| uniqueidentifier                | char(36)                  | varchar(36)             |
| unitext                         | unitext                   | clob                    |
| univarchar                      | univarchar                | nvarchar                |
| unsigned bigint                 |                           | decimal                 |
| unsigned int                    | unsigned int              | bigint                  |
| unsigned smallint               | unsigned smallint         | integer                 |
| unsigned bigint                 | unsigned bigint           | decimal                 |
| varbinary(max)                  | rs_msss_varbinary_<br>max | blob                    |
| varchar(max)                    | rs_msss_varchar_max       | clob                    |

**注意：**プライマリデータベースが、拡張ASCII文字(128～255の1バイトのASCII文字値)を受け付けるシングルバイト文字セット(ISO-1 など)に対応するように設定されている場合、SAP HANA データベースでは char または varchar カラムでシングルバイトの拡張文字をダブルバイト文字に変換することがあります。この変換により、SAP HANA データベースの記憶領域サイズがプライマリデータベースの元のカラム長または記憶領域を超過する可能性があります。プライマリデータベースとアプリケーションが、シングルバイト文字セットでの拡張 ASCII 文字の使用をサポートしている場合は、SAP HANA データベースのテーブル定義で対応するカラムのサイズを増やす必要があります。この変更により、2バイトの記憶領域への拡張文字の変換にカラムが対応できるようになります。

## Microsoft SQL Server データ型から Oracle データ型への変換

Microsoft SQL Server データ型から Oracle データ型へのクラスレベル変換を示します。

表 11 : Microsoft SQL Server データ型から Oracle データ型へのクラスレベル変換

| Microsoft SQL Server データ型 | Oracle データ型  |
|---------------------------|--------------|
| binary                    | RAW          |
| datetime                  | DATE (時刻を含む) |
| money                     | DECIMAL      |
| smalldatetime             | DATE         |
| smallmoney                | DECIMAL      |
| varbinary                 | RAW          |

## Microsoft SQL Server に対応する Replication Server のデータ型名

Microsoft SQL Server データ型を表す Replication Server のユーザ定義データ型 (UDD) 名を示します。

Microsoft SQL Server のデータ型はすべて、対応する Adaptive Server のデータ型と互換性があります。ユーザ定義データ型定義がある Microsoft SQL Server データ型は 1 つだけです。

表 12 : Microsoft SQL Server データ型に対応する Replication Server UDD 名

| Microsoft SQL Server データ型 | Replication Server の名前 |
|---------------------------|------------------------|
| integer                   | <i>rs_msss_bigint</i>  |

## Oracle データ型

Oracle データ型のクラスレベル変換 (デフォルトのデータ型マッピング) と、Oracle データ型に対応する Replication Server のデータ型について説明します。

**Adaptive Server データ型から Oracle データ型への変換**

Adaptive Server データ型から Oracle データ型へのクラスレベル変換を示します。

**表 13 : Adaptive Server データ型から Oracle データ型へのクラスレベル変換**

| Adaptive Server データ型 | Oracle データ型   |
|----------------------|---------------|
| bigdatetime          | TIMESTAMP (9) |
| bigint               | NUMBER        |
| bigtime              | TIMESTAMP (9) |
| binary               | RAW           |
| date                 | DATE          |
| datetime             | DATE (時刻を含む)  |
| money                | DECIMAL       |
| smalldatetime        | DATE          |
| smallmoney           | DECIMAL       |
| time                 | DATE (時刻を含む)  |
| unsigned tinyint     | SMALLINT      |
| unsigned smallint    | INTEGER       |
| unsigned int         | NUMBER        |
| unsigned bigint      | NUMBER        |
| unitext              | NCLOB         |
| varbinary            | RAW           |

**Oracle データ型から Adaptive Server データ型への変換**

Oracle データ型から Adaptive Server データ型へのクラスレベル変換を示します。

**表 14 : Oracle データ型から Adaptive Server データ型へのクラスレベル変換**

| Oracle データ型 | Adaptive Server データ型 |
|-------------|----------------------|
| RAW         | varbinary            |
| DATE        | datetime             |

| Oracle データ型   | Adaptive Server データ型 |
|---------------|----------------------|
| TIMESTAMP (9) | datetime             |

## Oracle データ型から DB2 データ型への変換

Oracle データ型から DB2 データ型へのクラスレベル変換を示します。

表 15 : Oracle データ型から DB2 データ型へのクラスレベル変換

| Oracle データ型   | DB2 データ型                      |
|---------------|-------------------------------|
| RAW           | CHAR FOR BIT DATA             |
| DATE          | DATE                          |
| DATE (時刻を含む)  | TIMESTAMP                     |
| FLOAT         | DOUBLE (UNIX と Windows のみ)    |
| INTEGER       | INTEGER (UNIX と Windows のみ)   |
| TIMESTAMP (9) | TIMESTAMP (UNIX と Windows のみ) |

## Oracle から SAP HANA データベース

Oracle から SAP HANA データベースへのデータ型マッピングの一覧を示します。

SAP HANA データベースデータ型で表すことができる値の範囲の現在の情報については、『SAP HANA Reference』 > 「SQL Reference Manual」 > 「Data Types」を参照してください。

---

**注意：** プライマリキーまたはプライマリキーの一部として、Oracle RAW または LONG RAW データ型のカラムを使用しないでください。

---

表 16 : Oracle から SAP HANA データベースへのデータ型マッピング

| Oracle データ型   | 複写定義のデータ型         | SAP HANA データベースデータ型 |
|---------------|-------------------|---------------------|
| ANYDATA       | opaque            | デフォルトなし             |
| BFILE         | image             | blob                |
| BINARY_DOUBLE | rs_oracle_decimal | double              |
| BINARY_FLOAT  | rs_oracle_float   | real                |
| BLOB          | image             | blob                |

| Oracle データ型 | 複写定義のデータ型                | SAP HANA データベース<br>データ型                                                                                                                                                                                                                                |
|-------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHAR        | char                     | varchar<br>(下記の注を参照)                                                                                                                                                                                                                                   |
| CLOB        | text<br>unitext (マルチバイト) | clob<br>nclob (マルチバイト)                                                                                                                                                                                                                                 |
| DATE        | rs_oracle_datetime       | timestamp                                                                                                                                                                                                                                              |
| FLOAT       | rs_oracle_float          | float                                                                                                                                                                                                                                                  |
| INTEGER     | rs_oracle_decimal        | decimal (34 桁まで)<br>varchar (34 桁以上)                                                                                                                                                                                                                   |
| INTERVAL    | rs_oracle_interval       | デフォルトなし                                                                                                                                                                                                                                                |
| LONG        | text                     | clob                                                                                                                                                                                                                                                   |
| LONG RAW    | image                    | blob                                                                                                                                                                                                                                                   |
| NCHAR       | unichar                  | nvarchar                                                                                                                                                                                                                                               |
| NCLOB       | unitext                  | nclob                                                                                                                                                                                                                                                  |
| NESTEDTAB   | opaque                   | デフォルトなし                                                                                                                                                                                                                                                |
| NUMBER      | rs_oracle_decimal        | double<br><br><b>注意：</b> Oracle の NUMBER が精度なしで指定されている場合、Oracle のカラムで使用できる最大値は SAP HANA データベースの decimal を超過します。SAP HANA データベースの double は、指定可能な Oracle 値をサポートする唯一の SAP HANA データベースデータ型です。NUMBER の精度が認識または指定されている場合は、SAP HANA データベースの decimal を使用してください。 |
| NUMBER (1)  | rs_oracle_decimal        | tinyint                                                                                                                                                                                                                                                |

## データ型の変換とマッピング

| Oracle データ型                       | 複写定義のデータ型                    | SAP HANA データベース<br>データ型                                                                                                                             |
|-----------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| NUMBER (2)<br>NUMBER (4)          | rs_oracle_decimal            | smallint                                                                                                                                            |
| NUMBER (5)<br>NUMBER (9)          | rs_oracle_decimal            | integer                                                                                                                                             |
| NUMBER (10)<br>NUMBER (18)        | rs_oracle_decimal            | bigint                                                                                                                                              |
| NUMBER (p, s)                     | rs_oracle_decimal<br>varchar | decimal (34 桁まで)<br>varchar (34 桁以上)                                                                                                                |
| NVARCHAR                          | univarchar                   | nvarchar                                                                                                                                            |
| NVARCHAR2                         | univarchar                   | nvarchar                                                                                                                                            |
| オブジェクト型                           | opaque                       | デフォルトなし                                                                                                                                             |
| RAW                               | rs_oracle_binary             | varbinary                                                                                                                                           |
| REF                               | rs_oracle_binary             | デフォルトなし                                                                                                                                             |
| ROWID                             | rs_oracle_rowid              | varchar (20)                                                                                                                                        |
| TIMESTAMP (n)                     | rs_oracle_timestamp9         | timestamp                                                                                                                                           |
| TIMESTAMP WITH<br>LOCAL TIME ZONE | rs_oracle_timestamp9         | デフォルトなし<br><br>Oracle の TIMESTAMP WITH<br>TIME ZONE データ型を SAP<br>HANA データベースに複写する<br>には、 <b>map to char</b> 句を使用し<br>てテーブル複写定義を明示的に<br>作成する必要があります。 |
| TIMESTAMP WITH<br>TIME ZONE       | rs_oracle_time-<br>stamptz   | デフォルトなし                                                                                                                                             |
| VARCHAR2                          | varchar                      | varchar<br><br>(下記の注を参照)                                                                                                                            |
| VARRAY                            | opaque                       | デフォルトなし                                                                                                                                             |

**注意：**プライマリデータベースが、拡張 ASCII 文字 (128 ~ 255 の 1 バイトの ASCII 文字値) を受け付けるシングルバイト文字セット (iso\_1 など) に対応するように設

定されている場合、SAP HANA データベースでは char または varchar カラムでシングルバイトの拡張文字をダブルバイト文字に変換することがあります。この変換により、SAP HANA データベースの記憶領域サイズがプライマリデータベースの元のカラム長または記憶領域を超過する可能性があります。プライマリデータベースとアプリケーションが、シングルバイト文字セットでの拡張 ASCII 文字の使用をサポートしている場合は、SAP HANA データベースのテーブル定義で対応するカラムのサイズを増やす必要があります。この変更により、2 バイトの記憶領域への拡張文字の変換にカラムが対応できるようになります。

#### 固定長カラムマッピング

SAP HANA データベースでは、サブスクリプションの **where** 句で指定された固定長の CHAR カラム値を埋めないため、値の不一致が発生することがあります。たとえば文字列 'abc' が、サブスクリプションの **where** 句で char(5) として定義された固定長カラムに渡されると、SAP HANA データベースでは、'abc ' のまま残っている文字列をスペースで埋めないため、不一致が発生します。したがって、固定長 CHAR カラムと SAP HANA データベースの varchar データ型をマップする必要があります。

## Oracle データ型から Microsoft SQL Server データ型への変換

Oracle データ型から Microsoft SQL Server データ型へのクラスレベル変換を示します。

表 17 : Oracle データ型から Microsoft SQL Server データ型へのクラスレベル変換

| Oracle データ型   | Microsoft SQL Server データ型 |
|---------------|---------------------------|
| RAW           | varbinary                 |
| DATE          | datetime                  |
| TIMESTAMP (9) | datetime                  |

## Oracle に対応する Replication Server のデータ型名

Oracle データ型を表す Replication Server のユーザ定義データ型 (UDD) 名を示します。

表 18 : Oracle データ型に対応する Replication Server UDD 名

| Oracle データ型 | Replication Server の名前 |
|-------------|------------------------|
| RAW         | rs_oracle_binary       |

| Oracle データ型                   | Replication Server の名前      |
|-------------------------------|-----------------------------|
| DATE                          | <i>rs_oracle_datetime</i>   |
| ROWID                         | <i>rs_oracle_rowid</i>      |
| INTEGER                       | <i>rs_oracle_int</i>        |
| INTERVAL                      | <i>rs_oracle_interval</i>   |
| BINARY_FLOAT                  | <i>rs_oracle_float</i>      |
| NUMBER                        | <i>rs_oracle_decimal</i>    |
| TIMESTAMP (n)                 | <i>rs_oracle_timestamp9</i> |
| TIMESTAMP (n) (ローカルタイムゾーンを含む) | <i>rs_oracle_timestampz</i> |
| UDD オブジェクトタイプ                 | <i>opaque</i>               |

## SAP HANA データベースデータ型

SAPHANA データベースデータ型のクラスレベル変換(デフォルトのデータ型マッピング)と、SAPHANA データベースデータ型に対応する Replication Server データ型の名前について説明します。

### Adaptive Server から SAP HANA データベースへのデータ型

Adaptive Server から SAP HANA データベースへのデータ型マッピングの一覧を示します。

SAP HANA データベースデータ型で表すことができる値の範囲の現在の情報については、『SAP HANA Reference』 > 「SQL Reference Manual」 > 「Data Types」を参照してください。

表 19 : Adaptive Server から SAP HANA データベースへのデータ型マッピング

| Adaptive Server データ型 | 複写定義のデータ型   | SAP HANA データベースデータ型 |
|----------------------|-------------|---------------------|
| bigdatetime          | bigdatetime | timestamp           |
| bigint               | bigint      | bigint              |
| bigtime              | bigtime     | デフォルトなし             |
| binary               | binary      | varbinary           |



| Adaptive Server データ型              | 複写定義のデータ型          | SAP HANA データベースデータ型                |
|-----------------------------------|--------------------|------------------------------------|
| bit                               | tinyint            | tinyint                            |
| char                              | char               | char<br>(下記の注を参照)                  |
| date                              | date               | date                               |
| datetime                          | datetime           | timestamp                          |
| decimal                           | decimal<br>varchar | decimal (34桁まで)<br>varchar (34桁以上) |
| double precision                  | float              | double                             |
| float                             | float              | float                              |
| identity                          | numeric            | numeric                            |
| int                               | integer            | integer                            |
| image                             | image              | blob                               |
| longsysname varchar(255) Not Null | varchar(255)       | varchar(255)                       |
| money                             | decimal            | decimal(19,4)                      |
| nchar                             | char               | nchar                              |
| numeric                           | numeric            | decimal                            |
| nvarchar                          | varchar            | nvarchar                           |
| real                              | real               | real                               |
| smalldatetime                     | smalldatetime      | timestamp                          |
| smallint                          | smallint           | smallint                           |
| smallmoney                        | decimal            | decimal(10,4)                      |
| sysname varchar(30) Not Null      | varchar(30)        | varchar(30)                        |
| text                              | text               | clob                               |
| timestamp                         | timestamp          | デフォルトなし                            |
| tinyint                           | tinyint            | tinyint                            |

| Adaptive Server データ型 | 複写定義のデータ型         | SAP HANA データベースデータ型  |
|----------------------|-------------------|----------------------|
| time                 | time              | time                 |
| unichar              | unichar           | nchar                |
| univarchar           | univarchar        | nvarchar             |
| unsigned bigint      |                   | decimal              |
| unsigned int         | unsigned int      | bigint               |
| unsigned smallint    | unsigned smallint | integer              |
| unsigned bigint      | unsigned bigint   | decimal              |
| unitext              | unitext           | nclob                |
| varbinary            | varbinary         | varbinary            |
| varchar              | varchar           | varchar<br>(下記の注を参照) |

**注意：**プライマリデータベースが、拡張ASCII文字(128～255の1バイトのASCII文字値)を受け付けるシングルバイト文字セット(iso\_1など)に対応するように設定されている場合、SAP HANA データベースでは char または varchar カラムでシングルバイトの拡張文字をダブルバイト文字に変換することがあります。この変換により、SAP HANA データベースの記憶領域サイズがプライマリデータベースの元のカラム長または記憶領域を超過する可能性があります。プライマリデータベースとアプリケーションが、シングルバイト文字セットでの拡張 ASCII 文字の使用をサポートしている場合は、SAP HANA データベースのテーブル定義で対応するカラムのサイズを増やす必要があります。この変更により、2バイトの記憶領域への拡張文字の変換にカラムが対応できるようになります。

## マテリアライゼーション

異機種データサーバや ASE 以外のデータサーバを含む複写システムを実装するときに考慮する必要がある、サブスクリプションマテリアライゼーションの問題、および ASE 以外のデータベースでプライマリテーブルのサブスクリプションをマテリアライズする方法についても説明します。

マテリアライゼーションとは、サブスクリプションを作成してアクティブ化し、プライマリデータベースからレプリケートデータベースにデータをコピーすることによって、レプリケートデータベースを初期化することです。

プライマリデータベースからデータを複写する前に、レプリケートオブジェクト（テーブルなど）がプライマリデータベースの状態と一致した状態になるように、各レプリケートデータベースを設定してデータを読み込んでおく必要があります。

### マテリアライゼーションの種類

---

Replication Server では、2 種類のサブスクリプションマテリアライゼーションがサポートされています。

これらのタイプには次のようなものがあります。

- バルクマテリアライゼーション - サブスクリプションを手動で作成してアクティブ化し、複写システムのコントロール外のデータアンロードユーティリティとデータロードユーティリティを使用してレプリケートデータベースにデータを読み込みます。
- 自動マテリアライゼーション - Replication Server コマンドを使用してサブスクリプションを作成し、レプリケートデータベースにデータを読み込みます。

サブスクリプションマテリアライゼーションメソッドの詳細については、『Replication Server 管理ガイド 第 1 巻』の「サブスクリプションの管理」を参照してください。

### 異機種マテリアライゼーション

---

適用できる場合、ASE 以外のデータサーバでプライマリデータのサブスクリプションをマテリアライズするには、バルクマテリアライゼーションまたは自動マテリアライゼーションを使用できます。

バルクマテリアライゼーションメソッドの場合、以下のアクティビティを調整し、手動で実行する必要があります。

- サブスクリプションの定義、アクティブ化、および検証を行う (またはマテリアライゼーションを伴わないサブスクリプションを作成する)。
- プライマリデータベースでサブスクリプションデータをアンロードする。
- アンロードしたデータをレプリケートデータベースサイトに移動する。
- プライマリデータをレプリケートデータベーステーブルにロードする。
- レプリケートデータベースが複写トランザクションを受け取ることができるように、レプリケート Replication Server からレプリケートデータサーバへのデータベースコネクションをレジュームする。
- Replication Agent インスタンスで複写をレジュームする。

## バルクマテリアライゼーションのオプション

---

ASE 以外のデータベースでのプライマリデータのサブスクリプションには、2 つのバルクマテリアライゼーションのオプションがあります。

これらのオプションには次のようなものがあります。

- アトミックバルクマテリアライゼーション
  - プライマリテーブルに対する更新を停止し、プライマリデータベースからサブスクリプションデータをダンプする。
  - レプリケート Replication Server で、サブスクリプションを定義する。
  - プライマリデータベースで、**rs\_marker** ファンクションを使用して、**with suspension** オプションを使用しているサブスクリプションを有効にする。このファンクションの適用に関する詳細については、『Replication Server リファレンスマニュアル』の「Replication Server システムファンクション」の「rs\_marker」を参照してください。
  - レプリケートテーブルにサブスクリプションデータをロードする。
  - レプリケート Replication Server からレプリケートデータベースへのデータベースコネクションをレジュームする。
  - レプリケート Replication Server で、サブスクリプションを検証する。
- ノンアトミックバルクマテリアライゼーション
  - レプリケート Replication Server では、**set autocorrection** コマンドを使用する。
  - レプリケート Replication Server で、サブスクリプションを定義する。
  - プライマリデータベースで、**rs\_marker** ストアドプロシージャを使用して、**with suspension** オプションを使用しているサブスクリプションを有効にする。
  - プライマリデータベースからサブスクリプションデータをダンプする。

- プライマリデータベースで、**rs\_marker** ストアドプロシージャを使用してサブスクリプションを検証する。
- レプリケートテーブルにサブスクリプションデータをロードする。
- レプリケート Replication Server からレプリケートデータベースへのデータベースコネクションをレジュームする。
- すべての Replication Server でサブスクリプションが有効になったら、オートコレクションをオフにする。

## プライマリデータベースからのデータのアンロード

---

サブスクリプションマテリアライゼーションプロセスには、プライマリテーブルからサブスクリプションデータをアンロードして、レプリケートテーブルにロードできるようにするプロセスが含まれます。「サブスクリプションデータ」とは、サブスクリプションによって要求されるプライマリテーブル内のデータのことで

す。

データアンロードユーティリティは、通常、データサーバソフトウェアに付属しています。OEM で提供される任意のデータアンロードユーティリティを使用することも、データベースアンロードユーティリティを使用することもできます。

---

**注意：** プライマリデータベースからサブスクリプションデータをアンロードした後、レプリケートデータベースにデータをロードする前に、アンロードしたデータに対してデータ型変換を実行しなければならない場合があります。

---

参照：

- データ型変換 (307 ページ)

## データ型変換

---

アンロードユーティリティを使用しておらず、自動マテリアライゼーションを使用している場合は、Replication Server が変換を実行します。

Replication Server の異機種データ型サポート (HDS) 機能を使用して、複写データに対してカラムレベル変換またはクラスレベル変換を実行する場合は、マテリアライゼーションのためにプライマリデータベースからアンロードしたサブスクリプションデータに対してデータ型変換を実行する必要があります。

## レプリケートデータベースへのデータのロード

---

サブスクリプションマテリアライゼーションプロセスには、プライマリテーブルからレプリケートテーブルにサブスクリプションデータをロードするプロセスが含まれます。

**注意：**プライマリデータベースからサブスクリプションデータをアンロードした後、レプリケートデータベースにデータをロードする前に、アンロードしたデータに対してデータ型変換を実行しなければならない場合があります。

---

レプリケートデータベースのデータサーバとして Adaptive Server Enterprise を使用している場合は、ASE の **bcp** ユーティリティを使用して、レプリケートデータベースにサブスクリプションデータをロードします。

レプリケートデータベースのデータサーバとして ASE 以外のデータサーバを使用している場合は、任意のロードユーティリティを使用してレプリケートデータベースにサブスクリプションデータをロードできます。

『Adaptive Server Enterprise ユーティリティガイド』の「ユーティリティコマンドリファレンス」の「**bcp**」を参照してください。

## アトミックバルクマテリアライゼーション

---

アトミックバルクマテリアライゼーションは、テーブルのコピーが作成される間、プライマリテーブルを更新するすべてのアプリケーションをサスペンドできることを前提とします。コピーはその後、レプリケートサイトにロードされます。

プライマリデータへの更新を（少なくとも一時的に）サスペンドできる場合は、このアトミックバルクマテリアライゼーションを使用してプライマリデータベースからデータを取得できます。

### マテリアライゼーションの準備

---

アトミックバルクマテリアライゼーションの手順を開始する前に、確認する必要があります。いくつかあります。

次のようなことを確認する必要があります。

- プライマリテーブルが存在し、格納データがある。
- プライマリテーブル（またはプライマリテーブルにある複製対象のカラム）に対して、所有権または **select** 権限を持つユーザ ID にアクセスできる。
- レプリケートテーブルが存在し、適切なカラム、データ型を格納している。
- 複製システムにおいて、すべての Replication Server を適切に設定している。

- プライマリ Replication Server で、複製定義を正しく作成している。
- Replication Agent for a DB2 UDB、Microsoft SQL Server、または Oracle プライマリデータベースを使用している場合は、次の点を確認する。
  - プライマリデータベースでいくつかのオブジェクトも作成する Replication Agent を適切に初期化している。
  - プライマリデータベースのプライマリテーブルに、マーク付けされ有効化された複製がある。
  - Replication Agent インスタンスを開始し、そのインスタンスを複製状態にしている。

## アトミックバルクマテリアライゼーションの実行

アトミックバルクマテリアライゼーションの実行について説明します。

1. `isql` を使用して、システム管理者 (**sa**) としてレプリケート Replication Server にログインします。

```
isql -Usa -Psa_password -SRRS_servername
```

構文の説明は次のとおりです。

- `sa` は、システム管理者のユーザ ID。
- `sa_password` は、システム管理者のユーザ ID のパスワード。
- `RRS_servername` は、レプリケート Replication Server のサーバ名。

2. レプリケート Replication Server で、サブスクリプションを定義します。

```
1> define subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> [where search_conditions]
5> go
```

`dataserver.database` は、レプリケートデータベースに対して使用する Replication Server コネクション名と一致している必要があります。

3. プライマリ Replication Server とレプリケート Replication Server で、サブスクリプションをチェックします。サブスクリプションのステータスが **DEFINED** であることを確認するには、次のように入力します。

```
1> check subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

4. プライマリトランザクションのアクティビティが行われないように、プライマリテーブルをロックします。これにより、マテリアライゼーション実行の間、プライマリテーブルに対する更新を避けることができます。
5. サイトの推奨されるデータベースアンロード方法を使用して、プライマリサイトでサブスクリプションデータをアンロードし、プライマリテーブルからデータを選択またはダンプします。

---

**注意：**プライマリテーブルからサブスクリプションデータをアンロードする場合は、複写定義で指定されているカラムと、サブスクリプションで指定されているローのみを選択するようにしてください。

---

6. サブスクリプションデータに必要なデータ型変換を実行します。

このデータの複写定義でカラムレベル変換が指定されている場合は、複写定義で指定されたデータ型変換を実行します。

サブスクリプションに対してクラスレベル変換が指定されている場合は、サブスクリプションに対して指定されたデータ型変換を実行します。

7. レプリケート Replication Server でサブスクリプションをアクティブにします。

```
1> activate subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> with suspension
5> go
```

8. プライマリ Replication Server とレプリケート Replication Server で、サブスクリプションがアクティブになるまで待機します。プライマリ Replication Server とレプリケート Replication Server の両方で、**check subscription** を実行して、サブスクリプションのステータスが **ACTIVE** であることを確認します。

レプリケート Replication Server で、サブスクリプションのステータスが **ACTIVE** である場合は、レプリケートデータベースに対するデータベースコネクションがサスペンドされています。

9. プライマリテーブルを読み込みおよび書き込みアクセスに戻します (ロックの解除)。

10. **bcpl**、またはサイトで推奨されるデータベースユーティリティを使用して、サブスクリプションデータをレプリケートデータベースにロードします。

11. レプリケート Replication Server から、レプリケートデータベースに対するデータベースコネクションをレジュームします。

```
1> resume connection
2> to dataserver.database
3> go
```

12. 次を実行して、レプリケート Replication Server でサブスクリプションを確定化します。

```
1> validate subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

13. プライマリ Replication Server とレプリケート Replication Server の両方でサブスクリプションが有効になるまで待機した後、**check subscription** を実行して、ステータスが **VALID** であることを確認します。



この手順を完了すると、サブスクリプションが作成され、レプリケートデータはプライマリデータと一致しています。複写は継続中です。

Replication Server の設定とマテリアライゼーションメソッドの詳細については、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」および『Replication Server 管理ガイド 第 1 巻』を参照してください。

#### 参照：

- Replicate Data Server としての Linux、UNIX、および Windows に関する IBM DB2 (107 ページ)

## ノンアトミックバルクマテリアライゼーション

---

ノンアトミックバルクマテリアライゼーションは、テーブルのコピーが作成される間、プライマリテーブルを更新するアプリケーションをサスペンドできないことを前提とします。

そのため、ノンアトミックマテリアライゼーションでは、レプリケートデータベースがプライマリデータベースと同期されるように、Replication Server の「オートコレクション」機能を使用する必要があります。

---

**注意：**プライマリテーブルの複写定義に **replicate minimal columns** 機能が設定されている場合、ノンアトミックマテリアライゼーションを使用することはできません。

---

## マテリアライゼーションの準備

---

アトミックではないバルクマテリアライゼーションの手順を開始する前に、確認する必要があることがいくつかあります。

次のことを確認します。

- プライマリテーブルが存在し、格納データがある。
- プライマリテーブル (またはプライマリテーブルにある複写対象のカラム) に対して、所有権または **select** 権限を持つユーザ ID にアクセスできる。
- レプリケートテーブルが存在し、適切なカラムを格納している。
- 複写システムにおいて、すべての Replication Server を適切に設定している。
- プライマリ Replication Server で複写定義を正しく作成している。
- Replication Agent for a DB2 UDB、Microsoft SQL Server、または Oracle プライマリデータベースを使用している場合は、次の点を確認する。
  - プライマリデータベースでいくつかのオブジェクトも作成する Replication Agent を適切に初期化している。

- プライマリデータベースのプライマリテーブルに、マーク付けされ有効化された複写がある。
- Replication Agent インスタンスを開始し、そのインスタンスを複写状態にしている。

### ノンアトミックバルクマテリアライゼーションの実行

アトミックではないバルクマテリアライゼーションの実行について説明します。

1. **isql** を使用して、システム管理者 (**sa**) としてレプリケート Replication Server にログインします。

```
isql -Usa -Psa_password -SRRS_servername
```

構文の説明は次のとおりです。

- *sa* は、システム管理者のユーザ ID。
- *sa\_password* は、システム管理者のユーザ ID のパスワード。
- *RRS\_servername* は、レプリケート Replication Server のサーバ名。

2. レプリケート Replication Server でオートコレクション機能をオンにします。

```
1> set autocorrection on
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

3. レプリケート Replication Server で **with suspension** オプションを使用して、サブスクリプションを定義します。

```
1> define subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> with suspension
5> go
```

*dataserver.database* は、レプリケートデータベースに対して使用する Replication Server コネクション名と一致している必要があります。

4. プライマリデータベースで、**rs\_marker** ストアドプロシージャを呼び出して、サブスクリプションをアクティブにします。
5. プライマリ Replication Server とレプリケート Replication Server で、サブスクリプションをチェックします。次を実行して、サブスクリプションのステータスが **ACTIVE** であることを確認します。

```
1> check subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

レプリケート Replication Server で、サブスクリプションのステータスが **ACTIVE** である場合は、レプリケートデータベースに対するデータベースコネクションがサスペンドされています。

6. サイトの推奨されるデータベースアンロード方法を使用して、プライマリサイトでサブスクリプションデータをアンロードし、プライマリテーブルからデータを選択またはダンプします。

---

**注意：** プライマリテーブルからサブスクリプションデータをアンロードする場合は、複写定義で指定されているカラムと、サブスクリプションで指定されているローのみを選択するようにしてください。

---

7. サブスクリプションデータに必要なデータ型変換を実行します。

このデータの複写定義でカラムレベル変換が指定されている場合は、複写定義で指定されたデータ型変換を実行します。

サブスクリプションに対してクラスレベル変換が指定されている場合は、サブスクリプションに対して指定されたデータ型変換を実行します。

8. プライマリデータベースで、**rs\_marker** ストアドプロシージャを呼び出して、サブスクリプションを検証します。
9. プライマリ Replication Server とレプリケート Replication Server の両方でサブスクリプションが有効になるまで待機した後、**check subscription** を実行して、ステータスが **VALID** であることを確認します。
10. **bcp** ユーティリティ、またはサイトで推奨されるデータベースロードユーティリティを使用して、サブスクリプションデータをレプリケートデータベースにロードします。
11. レプリケート Replication Server から、レプリケートデータベースに対するデータベースコネクションをレジュームします。

```
1> resume connection
2> to dataserver.database
3> go
```

12. プライマリ Replication Server とレプリケート Replication Server の両方でサブスクリプションが有効になるまで待機した後、**check subscription** コマンドを実行して、ステータスが **VALID** であることを確認します。

レプリケート Replication Server でサブスクリプションのステータスが **VALID** の場合、次を実行すると、レプリケートデータベースはプライマリデータベースと同期化されるので、オートコレクションをオフにすることができます。

```
1> set autocorrection off
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

13. この手順を完了すると、サブスクリプションが作成され、レプリケートデータはプライマリデータと一致しています。複写は継続中です。

## マテリアライゼーション

- 複製コマンド言語 (RCL) コマンドの詳細については、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」を参照してください。
- Replication Server の設定とマテリアライゼーションメソッドの詳細については、『Replication Server 管理ガイド 第 1 巻』を参照してください。

も参照してください。

### 参照：

- Replicate Data Server としての Linux、UNIX、および Windows に関する IBM DB2 (107 ページ)

## オートコレクション

Replication Server は **autocorrection** コマンドを複製定義に設定し、複製テーブルの重複しているローによって発生する障害を回避します。

**set autocorrection** コマンドは、各 **update** または **insert** オペレーションを **delete**、さらに **insert** に変換することによってマテリアライゼーション中に発生することのある不一致を修正します。

Replication Agents の API である **ra\_set\_autocorrection** を使って、マーク付けされたテーブルにオートコレクションを設定すると、**update** 文で変更されたカラムだけを送信するのではなく、Replication Agent はすべてのカラムを Replication Server に送信します。When Replication Agent が 1 つの特定のマーク付けテーブルまたはすべてのテーブルに対してオートコレクションを設定すると、該当する変更はプライマリデータベースに適用されます。

オートコレクションをサポートするプライマリデータベースは、次のとおりです。

- MS SQL Server
- IBM DB2
- Oracle - オートコレクション機能は、Oracle の redo ログレコードへの制限のために、LOB、LONG、LONG RAW およびユーザ定義型カラムを使用できません。
- ASE

『Replication Agent 15.5 Primary Database Guide』の「Replication Agent for Microsoft SQL」の「Replication Server **set autocorrection** Command」および『Replication Agent 15.5 Reference Manual』の「Command Reference」の「**ra\_set\_autocorrection**」を参照します。

## 直接ロードマテリアライゼーション

---

直接ロードマテリアライゼーションを使用すると、各種のプライマリデータベースとレプリケートデータベースの間でデータをマテリアライズできます。

直接ロードマテリアライゼーションは、その他の自動マテリアライゼーションメソッドとは異なります。

- マテリアライゼーションキューは、直接ロードマテリアライゼーションでは使用されません。データは、プライマリテーブルからレプリケートテーブルに直接ロードされます。
- 直接ロードマテリアライゼーション時に、その他のテーブルへの複写はサスペンドされません。マテリアライズされるプライマリテーブルに対する DML 操作はキャッチアップキューに格納され、初期マテリアライゼーションフェーズ後にレプリケートテーブルに適用されます。マテリアライズされないプライマリテーブルに対する DML 操作は、DSI が受信したときにレプリケートテーブルに複写されます。直接ロードマテリアライゼーションを使用すると、複数のテーブルを同時にマテリアライズできます。キャッチアップキューの DML 操作がレプリケートテーブルに適用されると、insert 操作のそれぞれが、delete 操作とそれに続く insert 操作に変換されます。更新でプライマリキーが変更されると、マテリアライゼーションは失敗します。

---

**注意：**（プライマリ IBM DB2 UDB または Oracle データベースのみ）**direct\_load** オプションでサブスクリプションが作成され、マテリアライゼーション中にプライマリテーブルに対する更新が予定される場合は、**ra\_set\_autocorrection** を有効にする必要があります。Replication Agent コマンドを実行すると、Replication Agent からダウンストリーム Replication Server に対してすべてのカラムの値を送信できます。

---

『Replication Agent リファレンスマニュアル』>「コマンドリファレンス」>「Replication Agent コマンドテーブル」>「ra\_set\_autocorrection」を参照してください。

- サブスクリプションのマテリアライゼーションがエラーにより停止しても、他のテーブルへの通常の複写はサスペンドされません。
- 複数の並列スレッドを使用して1つのプライマリテーブルからそれに対応するレプリケートテーブルにデータをロードできます。**max\_mat\_load\_threads** を使用して、この複数スレッド動作を調整できます。
- **direct\_load** を使用してサブスクリプションがマテリアライズされる場合は、設定パラメータ **dsi\_check\_unique\_key** を使用して、ユニークキーなしの条件をチェックする際の SAP Replication Server の動作を制御できます。この設定パラメータは、**configuration replication server** によりサーバレベルで使用すること

も、**alter connection** により接続レベルで使用することもできます。指定可能な値は、on または off で、デフォルト値は on です。

**dsi\_check\_unique\_key** が on の場合は、ユニークキーのないプライマリテーブルでは、サブスクリプションが有効 (VALID) になるまでアクティビティをいっさい実行できません。アクティビティを実行すると、キャッチアップフェーズでサブスクリプションが失敗する可能性があり、この場合はサブスクリプションがアボートされ、エラーがマークされます。サブスクリプションがアボートされたら、それを削除して作成しなおす必要があります。サブスクリプションを再作成する前に (テーブルのトランケートと同様に) レプリケートテーブルをクリーンアップする必要があります。

**dsi\_check\_unique\_key** を off に設定すると、サブスクリプションは正常に作成されますが、データの不整合が発生する可能性があります。サブスクリプションの確認後にデータが正しいことを保証するため、この設定パラメータは、次の状況でのみ off にしてください。

- ユニークキーのないプライマリテーブルに delete 文を指定できない場合
- ユニークキーのないプライマリテーブルに重複するローの挿入文を指定できない場合
- (プライマリ DB2 UDB 9.7 以降または Oracle データベースのみ) **create subscription** コマンドで **num\_of\_selects**/**selects** を指定することにより、複数のスレッドを有効にしてプライマリテーブルからデータを選択できます。『リファレンスマニュアル』の「**create subscription**」を参照してください。

ここで説明するアトミックおよびノンアトミックマテリアライゼーションメソッドは、SAP ASE プライマリでのみサポートされます。マテリアライゼーションの各種タイプおよびメソッドの説明については、『異機種間複写ガイド』の「マテリアライゼーションの種類」および『管理ガイド 第1巻』の「サブスクリプションマテリアライゼーションメソッド」を参照してください。

直接ロードマテリアライゼーションを使用してマテリアライズできるデータは次のとおりです。

- SAP ASE から SAP ASE
- SAP ASE から SAP HANA データベース
- Microsoft SQL Server から SAP HANA データベース
- Oracle から SAP HANA データベース
- DB2 UDB から SAP HANA データベース

### **create subscription** の制約と制限

- **direct\_load** オプションを使用すると、同じレプリケートテーブルに対して同時に他のサブスクリプションを作成したり定義したりできません。

- **direct\_load** オプションはテーブル複写定義へのサブスクリプション専用であり、**without holdlock** とともに使用します。 **without materialization** や **incrementally** とは併用できません。
- **user** オプションと **password** オプションは、**direct\_load** とのみ一緒に使用できません。
- **direct\_load** オプションは、物理データベース接続に対してのみ使用できます。代替接続や論理接続には使用できません。これはプライマリ接続 (複写定義で指定されている接続) および複写接続 (サブスクリプションで指定されている接続) の両方に当てはまります。
- プライマリデータベースのメンテナンスユーザを **user** オプションと **password** オプションで使用してサブスクリプションを作成することはできません。
- プライマリデータベースが SPA ASE でない場合は、アトミックマテリアライゼーションを使用できません。SAP ASE 以外のプライマリデータベースについては、サポートされている自動マテリアライゼーションのオプションは直接ロードのみです。レプリケートデータベースが SAP ASE でない場合、**with purge** オプションでサブスクリプションを削除することはできません。
- この **direct\_load** オプションが使用できるのは、レプリケート SAP Replication Server のサイトバージョンとそのプライマリ SAP Replication Server に対するルートバージョンが 1571100 以降の場合のみです。
- **direct\_load** オプションを使用して作成されたサブスクリプションでは、ローフィルタリング、名前マッピング、カスタムファンクション文字列、およびデータ型マッピングを使用できます。
- **direct\_load** オプションを使用してサブスクリプションを作成し、さらに同時に発生するサブスクリプション要求の数が **num\_concurrent\_subs** を超える場合は、Replication Server により、サブスクリプションに PENDING のマークが付きます。Replication Server は、同時サブスクリプション要求数が **num\_concurrent\_subs** を下回るまでサブスクリプションを処理しません。

## サブスクリプションと直接ロードマテリアライゼーション

直接ロードマテリアライゼーションは、テーブル複写定義に対するサブスクリプションでのみ使用します。

直接ロードマテリアライゼーションを使用するには、**create subscription** コマンドの **direct\_load** オプションを使用して、テーブル複写定義に対するサブスクリプションを作成します。『Replication Server リファレンスマニュアル』> 「Replication Server コマンド」> 「create subscription」を参照してください。

**direct\_load** オプションで作成したサブスクリプションを削除する場合:

- サブスクリプションがまだ有効になっていない場合は、**without purge** オプションを指定して **drop subscription** コマンドを使用します。これによってマテリア

ライゼーションスレッドの停止、テーブルに対する保留中の DML 操作のスキップ、サブスクリプションの削除が直ちに行われます。

- 削除されるサブスクリプションがまだ有効になっていない場合は、**drop subscription** を完了するためにレプリケート Replication Server DSI が実行されている必要があります。これは、削除されるサブスクリプションに関連するすべてのコマンドを Replication Server が消去する必要があるためです。

**without purge** オプションで作成したサブスクリプションを削除する場合は、ユーザがレプリケートテーブルを削除する必要があります。

## Adaptive Server から SAP HANA データベースへの直接ロードマテリアライゼーション

この手順は、直接ロードマテリアライゼーションの設定、および Adaptive Server プライマリから SAP HANA データベースレプリケートへの複製の有効化に使用します。

1. **isql** を次のように使用して、レプリケート Replication Server にログインします。

```
isql -Uiuser -Pipassword -SRRS_servername
```

各パラメータの意味は、次のとおりです。

- *iuser* はサブスクリプション作成者です。作成者には **create object** パーミッションが必要です。
- *ipassword* は *iuser* のパスワードです。
- *RRS\_servername* は、レプリケート Replication Server のサーバ名です。

2. レプリケート Replication Server でサブスクリプションを作成します。

```
1> create subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> [where search_conditions]
5> without holdlock
6> direct_load

8> [user cusername password cpass]
9> go
```

*dataserver.database* は、レプリケートデータベースに対して使用する Replication Server 接続名と一致している必要があります。ここで *cusername* を指定すると、この名前がプライマリデータベースへのログインに使用され、プライマリテーブルから選択されます。ここで *cusername* を選択しない場合は、**isql** で指定した名前である *iuser* が使用されます。プライマリデータベースへのログインに使用した名前にプライマリデータベースで必要とされるパーミッションが付与されていることを確認してください。



### 3. サブスクリプションマテリアライゼーションの進行状態をモニタします。

```
1> check subscription subscription_name for repdef_name
2> with replicate at replicate_dataserver.replicate_database
3> go
```

マテリアライゼーションが完了すると、Replication Server は次のようなメッセージを返します。

```
Subscription <subscription_name> has been MATERIALIZED at the
replicate.
```

サブスクリプションプロセスが完了すると、Replication Server は次のようなメッセージを返します。

```
Subscription <subscription_name> is VALID at the replicate.
```

サブスクリプションプロセスにエラーが発生すると、**check subscription** コマンドは次のようなメッセージを返します。

```
Subscription <subscription_name> encountered ERROR.
```

サブスクリプションでエラーが発生したときは、レプリケートおよびプライマリ Replication Server のログファイルを確認して原因を診断し、パージなしのサブスクリプションの削除、レプリケートテーブルからのデータの削除、エラーの修正、サブスクリプションの再作成を行います。『Replication Server トラブルシューティングガイド』>「サブスクリプションの問題」>「check subscription」>「マテリアライゼーションステータス」と『Replication Server トラブルシューティングガイド』>「サブスクリプションの問題」>「マテリアライゼーションの問題」を参照してください。

Replication Server の設定とマテリアライゼーションメソッドの詳細については、『Replication Server リファレンスマニュアル』>「Replication Server コマンド」および『Replication Server 管理ガイド 第1巻』を参照してください。

## Adaptive Server 以外のデータベースから SAP HANA データベースへの直接ロードマテリアライゼーション

この手順は、直接ロードマテリアライゼーションの設定、Microsoft SQL Server、Oracle、または DB2 UDB プライマリから SAP HANA データベースレプリケートへのデータのマテリアライズに使用します。

Replication Agent は、Microsoft SQL Server、Oracle、または DB2 UDB プライマリからの直接ロードマテリアライゼーションで使用されます。Replication Server が Replication Agent にログインし、Replication Agent が Replication Server の代わりにキューを発行します。キューの結果は Replication Server に返されます。

1. **dsedit** を使用して Replication Server の `sql.ini` (Windows) ファイルまたは `interfaces` (UNIX または Linux) ファイルを更新して、Replication Agent の場所のエントリを含めます。
2. プライマリデータベースを管理する Replication Server で、プライマリデータベースへの接続を作成します。

```
create connection to pds.pdb
using profile profile_name;standard
set username muser
set password mpwd
with log transfer on,
dsi_suspended
go
```

各パラメータの意味は、次のとおりです。

- *pds* は、Replication Agent で指定された **rs\_source\_ds** パラメータの値です。
- *pdb* は、Replication Agent で指定された **rs\_source\_db** の値です。
- *profile\_name* は Microsoft SQL Server では **rs\_rs\_to\_msss\_ra**、Oracle では **rs\_rs\_to\_oracle\_ra**、DB2 UDB では **rs\_rs\_to\_udb\_ra** です。
- *muser* は、プライマリデータベースのメンテナンスユーザです。
- *mpwd* は、メンテナンスユーザのパスワードです。

これは、一度だけ必要になります。

これらの接続プロファイル (**rs\_rs\_to\_msss\_ra**、**rs\_rs\_to\_oracle\_ra**、または **rs\_rs\_to\_udb\_ra**) のいずれも使用しない既存のプライマリデータベース接続が存在する場合は、正しいファンクション文字列クラス (**rs\_msss\_ra\_function\_class**、**rs\_oracle\_ra\_function\_class**、または **rs\_udb\_ra\_function\_class**) と **ra\_ra\_error\_class** エラークラスを使用するように、その接続を変更する必要があります。

3. **isql** を次のように使用して、レプリケート Replication Server にログインします。

```
isql -Uiuser -Pipassword -SRRS_servername
```

各パラメータの意味は、次のとおりです。

- *iuser* はサブスクリプション作成者です。作成者には **create object** パーミッションが必要です。
  - *ipassword* は *user* のパスワードです。
  - *RRS\_servername* は、レプリケート Replication Server のサーバ名です。
4. (オプション) 次の手順のユーザ ID とパスワードを使用して、Replication Agent に対する **isql** セッションをオープンします。これにより、次の手順のユーザ ID に適切な Replication Agent のパーミッションと正しいパスワードが設定されていることを確認できます。
  5. レプリケート Replication Server でサブスクリプションを作成します。

```
1> create subscription subscription_name
2> for replication_definition
```

```

3> with replicate at dataserver.database
4> [where search_conditions]
5> without holdlock
6> direct_load

8> [user cusername password cpass]
9> go

```

`dataserver.database` は、レプリケートデータベースに対して使用する Replication Server 接続名と一致している必要があります。

ここで `cusername` を指定すると、このユーザが Replication Agent への接続に使用されます。このユーザは Replication Agent の管理ユーザでなければなりません。ここで `cusername` を選択しない場合は、`isql` で指定した名前である `iuser` が使用されます。このユーザが Replication Agent の管理ユーザであることを確認します。

#### 6. サブスクリプションマテリアライゼーションの進行状態をモニタします。

```

1> check subscription subscription_name for repdef_name
2> with replicate at replicate_dataserver.replicate_database
3> go

```

マテリアライゼーションが完了すると、Replication Server は次のようなメッセージを返します。

```
Subscription <subscription_name> has been MATERIALIZED at the replicate.
```

サブスクリプションプロセスが完了すると、Replication Server は次のようなメッセージを返します。

```
Subscription <subscription_name> is VALID at the replicate.
```

サブスクリプションプロセスにエラーが発生すると、`check subscription` コマンドは次のようなメッセージを返します。

```
Subscription <subscription_name> encountered ERROR.
```

サブスクリプションでエラーが発生したときは、レプリケートおよびプライマリ Replication Server のログファイルを確認して原因を診断し、パージなしのサブスクリプションの削除、レプリケートテーブルからのデータの削除、エラーの修正、サブスクリプションの再作成を行います。『Replication Server トラブルシューティングガイド』>「サブスクリプションの問題」>「check subscription」>「マテリアライゼーションステータス」と『Replication Server トラブルシューティングガイド』>「サブスクリプションの問題」>「マテリアライゼーションの問題」を参照してください。

この手順を完了すると、サブスクリプションが作成され、レプリケートデータはプライマリデータと一致し、このテーブルの複写は有効になります。

Replication Server の設定とマテリアライゼーションメソッドの詳細については、『Replication Server リファレンスマニュアル』の「Replication Server コマンド」および『Replication Server 管理ガイド 第 1 巻』を参照してください。

## 直接ロードマテリアライゼーションの設定パラメータ

直接ロードマテリアライゼーションでは次の設定パラメータを使用します。

Replication Server のインストール後、**rs\_init** によってデフォルトの設定パラメータが設定されます。

**configure replication server** を使用してこれらのパラメータを変更する方法については、『Replication Server 管理ガイド 第 1 巻』>「複製システムの管理」>「Replication Server 設定パラメータの設定」>「Replication Server パラメータの変更」を参照してください。

表 20 : 直接ロードマテリアライゼーションの設定パラメータ

| 設定パラメータ                     | 説明                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>mat_load_tran_size</b>   | <p>直接ロードマテリアライゼーションでプライマリデータをレプリケートテーブルに初期コピーするときの最適なトランザクションサイズまたはバッチサイズを指定する。</p> <p>有効な値: 10 ~ 2147483646<br/>デフォルト値: 10000</p>                                                                                                                                                                                                                   |
| <b>max_mat_load_threads</b> | <p>マテリアライズされるテーブルごとの最大ロードスレッド数を指定する。</p> <p>有効な値: 1 ~ 20<br/>デフォルト値: 5</p> <p>Replication Server は、テーブルごとに 1 個のロードスレッドで直接ロードマテリアライゼーションを開始し、必要に応じてこのパラメータで指定された数までスレッドを追加生成する。</p> <p><b>max_mat_load_threads</b> は、ローカルの Replication Server とデータベースの接続パラメータである。</p> <p>このパラメータと <b>num_concurrent_subs</b> の値によって直接ロードマテリアライゼーションのリソース使用が制御される。</p> |

# 異機種データベースの調整

異機種間複写システムにある異なるデータベースのデータの比較と調整に関する問題について説明します。

## SAP の rs\_subcmp ユーティリティ

---

**rs\_subcmp** ユーティリティを使用すると、Adaptive Server データベース内のプライマリテーブルとレプリケートテーブルを比較し、差異を調整できます。SAP では、**rs\_subcmp** 実行プログラムを Replication Server に付属して提供しています。

他のデータベースベンダでも、自社のデータベースに対して同じ機能を実行できる同様の「比較」ユーティリティを提供している場合がありますが、異なる種類の ASE 以外のデータサーバ (Oracle データベース内のテーブルと Microsoft SQL Server データベース内のテーブルの比較など) をサポートする同等のユーティリティはありません。

ASE 以外のデータベースのサポートについては、そのような機能を備えたサードパーティのツールを購入するか、独自のアプリケーションを作成できます。

## SAP Replication Server データ保証オプション

---

SAP® Replication Server® データ保証オプションは、2つ以上のデータベースのローデータおよびスキーマを比較して、不一致を報告します。

SAP Replication Server データ保証オプションは、スケーラブルで大容量の設定可能なデータ比較製品です。「wait and retry」方式を使用してダウンタイムを排除することで、複写中でも比較ジョブを実行できます。

各比較ジョブでは、どのデータをどのように比較するかを指定した多数の設定を使用してデータの不一致を確認できます。SAP Replication Server データ保証オプションには、比較とレポート作成のあらゆるジョブを実行するためのコマンドラインツール (CLT) が搭載されています。ユーザはジョブのモニタおよびアポートを実行したり、詳細な比較レポートを作成したりできます。

SAP® Replication Server® データ保証オプションを使用すれば、大きいテーブルを複数のパーティションに分割してデータを並列で比較することができます。異機種比較環境では、SAP® Adaptive Server® Enterprise (SAP® ASE) と SAP® IQ、SAP HANA®、IBM DB2 Universal Database (UDB)、Microsoft SQL Server、または Oracle のデータベースを任意に組み合わせてローデータを比較することもできます。

SAP Replication Server データ保証オプションは、SySAM ライセンスマネージャを通じてライセンスされ、複数のプラットフォームで使用できます。SySAMの詳細については、使用しているプラットフォームの『インストールガイド』、または SySAM Web サイト (<http://www.sybase.com/sysam>) を参照してください。

## データベース比較アプリケーション

---

**rs\_subcmp** ユーティリティと同じ機能を実行するカスタムアプリケーションを開発できます。アプリケーションの複雑さは、データサーバの種類の数、比較するテーブルの複雑さ、関連するデータ変換の量などによって異なります。

以下に、異機種間複写環境内でデータベース比較アプリケーションが適切に動作するうえで対処する必要がある主要な問題を列挙し説明します。

- **コネクティビティ** - アプリケーションは、プライマリデータベースとレプリケートデータベースの両方と通信できなければなりません。複数のデータベースベンダが関連している場合、ODBC および JDBC プロトコルが、共通のインタフェースと機能を提供できます。
- **ソート順** - データベースごとにデフォルトのソート順が異なる可能性があります。アプリケーションは、比較のパフォーマンスを向上させるためにソート順を強制しなければならない場合があります。
- **文字セット** - プライマリデータベースとレプリケートデータベースに、異なる文字セットで文字データが格納されている可能性があります。カスタムアプリケーションは、これらの変換をサポートしなければならない場合があります。
- **オブジェクトの識別** - プライマリテーブルとレプリケートテーブルの名前が一致しない可能性があります。つまり、まったく同じスキーマ名やカラム名でない可能性があります。比較アプリケーションは、参照されるロケーション、データベース、テーブル名、カラム名に関する非常に明示的な命令を受け入れなければならない場合があります。
- **サブセットの比較** - アプリケーションは、テーブルの一部のみを比較しなければならない可能性があります。プライマリテーブルとレプリケートテーブルの両方に、**where** 句タイプの **select** を指定できることが重要な場合があります。
- **遅延時間** - 複写システムには、必ず遅延時間(プライマリトランザクションがレプリケートテーブルで発生するまでにかかる時間)があります。比較アプリケーションには、「そこにはない」ローと「現時点ではそこにはない」ローを区別する許容性を組み込む必要があります。
- **データ変換** - アプリケーションは、Replication Server がクラスレベル変換をサポートできるように、異なるデータベース間の精度とフォーマットに関する差異を処理できなければなりません。処理を単純にするには、データ型に基づいて特定のカラムが比較プロセスから除外されるようにします(たとえば、異なるデータベースベンダの DATE データ型を比較しない)。

- ラージオブジェクト (LOB) データ - ラージオブジェクト (例、LOB、CLOB、TEXT、または IMAGE) データ型では、サイズが原因となって追加処理の問題が生じます。引き続き「一致しない」可能性が高い場合は、パフォーマンスを向上させるために、比較に使用するバイト数を制限します。

『Replication Server 管理ガイド』および『Replication Server リファレンスマニュアル』の「実行プログラム」の「rs\_subcmp」に、**rs\_subcmp** についての詳細が記載されています。





# 異機種間複写システムのトラブルシューティング

異機種データサーバや ASE 以外のデータサーバを含む SAP 複写システムの一般的な問題を解決します。

ステーブルキューのダンプ、データサーバインタフェース (DSI) および Replication Server インタフェース (RSI) に関する障害のデバッグ、サブスクリプションに関する問題の診断と修正など、Replication Server の一般的なトラブルシューティング作業については、『Replication Server トラブルシューティングガイド』で説明しています。

ASE 以外のプライマリデータベースとレプリケートデータベースについては、Replication Agent と ECDA ゲートウェイのマニュアルに、データベースごとのトラブルシューティング情報が記載されています。

## インバウンドキューの問題

---

「インバウンドキュー」は、Replication Server が (Replication Agent または別の Replication Server を介して) プライマリデータベースから受け取ったデータを格納する場所です。

プライマリ Replication Server で Replication Server の **admin who,sqm** コマンドを発行した場合に、結果が以下のようなであれば、プライマリデータベースの Replication Server インバウンドキューは更新されていないと判断できます。

- 該当する接続の Replication Server インバウンドキューに書き込まれるブロックの数が増えない。
- 検出された重複メッセージの数が増えない。

## インバウンドキューが更新されていない原因の特定

インバウンドキューが更新されていない原因を特定する方法について説明します。

1. Replication Server 接続の Replication Agent ユーザスレッドのステータスを確認します。

該当する Replication Server データベース接続の Replication Agent ユーザスレッドのステータスを確認するには、プライマリ Replication Server で **admin who** コマンドを発行します。

- 接続の Replication Agent ユーザスレッドがない場合、その接続は **with log transfer on** 句を使用して作成されていません。必要に応じ

て、Replication Server データベースコネクションを変更して、ログ転送処理をオンにします。

- Replication Agent ユーザスレッドのステータスが Down (停止) の場合、Replication Agent は Replication Server にアクティブに接続されていません。Down ステータスは、送信する作業がある場合にのみ Replication Server に接続し、一定期間、作業が行われなければ切断する Replication Agent にとって、典型的な状態です。

### 2. 必要な Replication Agent が実行中であることを確認します。

必要な Replication Agent がアクティブであり、Replication Agent の `rs_source_ds` および `rs_source_db` 設定パラメータが、該当する Replication Server コネクション名と一致していることを確認してください。

Replication Agent が実行中であることを確認するためのその他のテストについては、該当する Replication Agent のマニュアルを参照してください。

### 3. 必要なテーブルまたはプロシージャが、複写するようマーク付けされていることを確認します。

複写のステータスを確認するために使用可能な Replication Agent のコマンドについては、Replication Agent のマニュアルを参照してください。

Replication Agent には、マーク付けの他に、複写の「有効/無効」の設定があります。この場合、マーク付けされたオブジェクトが複写に対して有効であることも確認してください。

### 4. Replication Agent が新しいレコードをスキャンしていることを確認します。

データベースオブジェクトが複写するようマーク付けされている場合、Replication Agent のログスキャンプロセスによって、追加情報がスキャンされていることが記録されます。

新しいレコードがスキャンされていることを確認するには、次の手順に従います。

- Replication Agent でトレースを開始します。
- 複写するようマーク付けされたプライマリデータベースオブジェクトを更新または実行します。
- スキャンが行われていることを確認します。

スキャンプロセスを検証するために使用できるトレースフラグを確認するには、該当する Replication Agent のマニュアルを参照してください。

## アウトバウンドキューの問題

---

「アウトバウンドキュー」は、Replication Server がレプリケートサイト (レプリケートデータベースまたは別の Replication Server) に送信する必要があるデータを格納する場所です。

レプリケート Replication Server で Replication Server の **admin who, sqm** コマンドを発行した場合に、結果が以下のようなであれば、レプリケートデータベースの Replication Server アウトバウンドキューは更新されていないと判断できます。

- 該当するコネクションの Replication Server アウトバウンドキューに書き込まれるブロックの数が増加しない。
- 検出された重複メッセージの数が増加しない。

インバウンドキューとアウトバウンドキュー間の問題は、多くの場合、名前の付け方に起因します。

プライマリ Replication Server インバウンドキューがデータを受け取ることができても、そのデータを複写定義に適用できない場合は、複写定義の名前が、Replication Agent によって作成された Log Transfer Language (LTL) で示された名前と一致していないことが原因です。デフォルトの大文字と小文字の規則が異なる、ASE 以外の異なるデータベースタイプを使用している場合、この可能性はさらに高くなります。

Replication Server の複写コマンドの処理では、大文字と小文字を区別します。ASE 以外のデータサーバを含む複写システムでは、Replication Agent によって生成された LTL が、Replication Server コネクション名および複写定義オブジェクト名と一致していることを確認してください。

一部の Replication Agent は、Replication Server (Adaptive Server、DB2 UDB など) と通信するときに、常に小文字の名前を使用します。ただし、最もよい方法は、どちらかの文字 (大文字または小文字) を選択し、すべての Replication Server コネクション、複写定義、およびサブスクリプション名で一貫してそれを使用することです。

大文字と小文字の区別は手動で確認します。**rs\_helprep** コマンドを使用すると、複写定義の名前を確認できます。その後、Replication Agent で LTL トレースをオンにし、LTL トレースで提供された名前が、複写定義で指定された名前のスペルおよび大文字と小文字の設定と一致しているかどうかを確認できます。

大文字と小文字の区別が正しくないと思われる場合は、Replication Agent のマニュアルを参照して、デフォルトの文字設定と可能な設定の変更を確認してください。名前のスペルが正しくない場合は、複写定義を削除した後、作成し直してください。

## アウトバウンドキューが更新されていない原因の特定

アウトバウンドキューが更新されていない原因を特定する方法について説明します。

1. アクティブな Replication Server ルートがあることを確認します。

プライマリ Replication Server とレプリケート Replication Server 間のルートの検証方法については、『Replication Server トラブルシューティングガイド』の「ルートの問題」を参照してください。

2. Replication Server コネクションの DSI スレッドが停止していないことを確認します。

Replication Server コネクションの DSI スレッドのステータスを確認するには、レプリケート Replication Server で **admin who** コマンドを発行します。

DSI スレッドのステータスが Down (停止) の場合、Replication Server はレプリケートデータベース (または ECDA ゲートウェイ) に接続されていません。

Replication Server ログでエラーを確認し、コネクションをレジュームしてください。

3. レプリケート Replication Server ログで該当する DSI スレッドの “Loss Detected” メッセージを参照して、DSI スレッドコネクションが “Loss Detected” モードでないことを確認します。

Replication Server がロスを検出した場合、DSI スレッドコネクションでは、それ以上メッセージが受け入れられません。

このエラーのリカバリについては、『Replication Server 管理ガイド』を参照してください。

4. プライマリ複写定義を確認します。

## レプリケートデータベースがなぜ更新されないかを特定します。

レプリケートトランザクションがレプリケートデータベースで適用されない原因を特定する方法を説明します。

Replication Server アウトバウンドキューが更新されているにもかかわらず、レプリケートデータベースでトランザクションデータが適用されていない場合、原因を特定するには、以下の手順に従います。

1. サブスクリプションに **where** 句が含まれているかどうかを調べます。

サブスクリプション定義で、該当するトランザクションデータが **where** 句を渡すことを確認します。**rs\_helpsub** ストアドプロシージャを使用して、サブスクリプションのテキストを一覧表示します。

2. HDS のインストールを確認します。

ASE 以外のデータサーバとの複写をサポートするために Replication Server HDS を使用している場合は、HDS の接続プロファイルが正しく適用されていることを確認します。

3. `rs_lastcommit` テーブルが正しく設定されていることを確認します。

ASE 以外のデータサーバとの複写をサポートするために Replication Server HDS を使用している場合は、HDS の接続プロファイルが正しく適用されていることを確認します。

4. レプリケート Replication Server ログでエラーを確認します。

5. レプリケートデータベースログでエラーを確認します。

6. レプリケートオブジェクトへの手動でのアクセスを確認します。

Replication Server コネクションのメンテナンスユーザ ID を使用して、レプリケートデータベース (または ECDA ゲートウェイ) にログインし、レプリケートテーブルまたはプロシージャに対する **update** 権限があることを確認します。

7. レプリケートデータベースに送信されるコマンドを確認します。

Replication Server からレプリケートデータベースに送信されるコマンドを表示するには、DSI バッファダンプトレースフラグを使用します。

レプリケートデータベースが ASE や IQ などの SAP データベースである場合、または ECDA ゲートウェイを使用している場合、**DSI\_BUF\_DUMP** トレースフラグを使用します。

```
trace "on", dsi, dsi_buf_dump go
```

レプリケートデータベースが通信に ExpressConnect を使用している場合は、以下を使用します。

```
alter connection to replicate_data_server.replicate database set trace to 'econn, dsi_buf_dump, on' go
```

---

**注意：** どちらのトレースでも出力が Replication Server のエラーログに書き込まれます。

---

何らかのエラーが原因で DSI が停止している場合は、トレースで出力が生成されるように DSI を再開する必要があります。

---

**注意：** トレースは、DSI スレッドが起動している間しか機能しません。

---

DSI を実行できない場合、レプリケートデータベースにデータを送信するには、DSI エラーの原因を解決する必要があります。

DSI がアクティブな状態のときに、プライマリデータベースで新しいトランザクションアクティビティを作成します。その後、トランザクションがトレース出力に表示された場合、コマンドは正常にレプリケートデータベースに配信されています。

コマンドが表示されない場合は、環境のレプリケーションパス内で事前に問題が発生しています。

---

**注意：** `DSI_BUF_DUMP` トレースフラグは、すべての Replication Server で使用できます。これに対して、類似した `DSI_CMD_DUMP` トレースフラグは、Replication Server の診断バージョンでしか使用できません。Replication Server トレースフラグについての詳細は、『Replication Server トラブルシューティングガイド』を参照してください。

---

8. ECDA ゲートウェイでトレースをオンにして、受信しているコマンドを確認します。

たとえば、ECDA Option for Oracle 設定ファイル内の次のパラメータによって、ECDA は `DCO.log` ファイルに追加情報を書き込みます。

- `network_tracing = 1`
- `traces = 1,2,3,4,5,6,10`

特定のトレースの使用範囲と構文については、該当する ECDA のマニュアルを参照してください。

### 参照：

- 想定したデータ型変換が行われない (336 ページ)
- `rs_lastcommit` が更新されない (336 ページ)

## HDS の問題と制限事項

---

この項では、Replication Server の HDS 機能に関する既知の問題と制限事項について説明します。

### 変換先のデータ型の範囲を超える変換元の値

SAP が提供しているデータ型変換では、変換元の値が変換先のデータ型の範囲を超える変換を行おうとするスレッドは停止されます。

これは以下のエラーメッセージです。

```
E. 2007/12/14 11:14:54. ERROR #32055 DSI EXEC(135(1)
snickers_dco.ora805) -
/nrm/nrm.c(7023)
Class Level translation for column/parameter
'datetimcol' failed.
Source DTID is 'datetime'.
```

```
Target DTID is 'rs_oracle_datetime'.
Function String Class ID 'rs_oracle_function_class'.
Value length is '21'; Maximum target length is '20';
The value is '99991231 23:59:59:010'
```

これらは通常、変換元／変換先データ型の組み合わせにも変換される値にも問題がないように見えるため、診断が最も困難な変換の問題です。

このような問題を診断するには、変換されるすべての変換元データ型のデータ型値の範囲を理解しておく必要があります。たとえば、上記のエラーを診断するには、Oracle の DATE 値の上限が 12/31/9999 であることを理解していなければなりません。

データ型変換については、次のようなオプションもあります。

- データ型定義の最大値を使用する。
- データ型定義の最小値を使用する。
- データ型定義のデフォルト値を使用する。

## 真数値データ型の問題

複写される値がデータ型定義でサポートされている値の範囲 (最大値または最小値) に達すると、真数値データ型に関する問題が生じることがあります。

Microsoft SQL Server では、サーバの起動方法によって 28 桁または 38 桁の精度がサポートされます。デフォルトでは、Microsoft SQL Server は 28 桁の精度をサポートします。

ただし、SAP では、Microsoft のデフォルトである 28 桁の精度をサポートするデータ型定義を提供していません。Replication Server のネイティブな numeric データ型は 72 桁までの精度をサポートするので、38 桁の精度をサポートするデータ型定義は必要はありません。

数値が Microsoft SQL Server レプリケートデータベースの数値精度を超えると、Replication Server は次のエラーを返します。

```
E. 2007/12/14 11:14:58. ERROR #1028 DSI EXEC (134 (1)
dcm_gabeat70_devdb.devdb)
- dsiqmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|Data Source
Name=mssql170_devdb|SQLState=22003|Native
Error=1007|Message=[Microsoft][ODBC SQL Server
Driver][SQL Server]The number
'99999999999999999999.99999999999999999999' is out of
the range for numeric representation (maximum
precision 28).[Message Iteration=2|SQLState=22003|
Native Error=|Message=[Microsoft][ODBC SQL Server
```





```
E. 2007/12/14 12:29:16. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsigmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1|Data Source Name=mssql70_devdb|SQL
Function=INSERT|SQLState=22003|Native Error=
8115|Message=[Microsoft][ODBC SQL Server Driver]
[SQL Server]Arithmetic overflow error converting
numeric to data type numeric.[Message Iteration=
2|SQLState=01000|Native Error=|Message=
[Microsoft][SQL Server]The statement has been
terminated.] <DCA>'
```

## Microsoft SQL Server での数値変換と identity カラム

Microsoft SQL Server では、Adaptive Server と同じように identity カラムをサポートしているため、identity の挿入をオフおよびオンに設定する Replication Server のファンクション文字列が機能します。

しかし、Microsoft SQL Server データベースの 28 桁の精度をサポートするには、numeric データ型が rs\_mssql\_numeric データ型に変換されなければなりません。その結果、ID 特性が失われます。この問題を回避するには、Microsoft SQL Server レプリケートテーブルが、変換された numeric カラムを identity として宣言しないようにしてください。

変換された numeric データ型を Microsoft SQL Server で identity カラムに複写しようとする、次のようなエラーが表示されます。

```
E. 2007/12/14 12:05:39. ERROR #1028 DSI EXEC(134(1)
dcm_gabeat70_devdb.devdb)
- dsigmint.c(2888)
Message from server: Message: 30291, State 0,
Severity 19 --
'[VENDORLIB] Vendor Library Error: [[Message
Iteration=1 |Data Source Name=mssql70_devdb|SQL
Function=INSERT|SQLState=23000|Native Error=544
|Message=[Microsoft][ODBC SQL Server Driver][SQL
Server]Cannot insert explicit value for identity
column in table 'ase alltypes' when IDENTITY_INSERT
is set to OFF.] <DCA>'
```

## 特定のエラーのトラブルシューティング

異機種データサーバや ASE 以外のデータサーバを含む SAP 複写システムで発生する可能性がある特定のエラーのトラブルシューティングについて説明します。

## rs\_lastcommit が更新されない

ASE 以外のレプリケートデータベースに複写する場合、レプリケート Replication Server は、コネクションがレジュームされると、すぐに rs\_lastcommit テーブルを更新します。

### rs\_lastcommit の更新エラーのトラブルシューティング

rs\_lastcommit テーブルの更新中にレプリケート Replication Server エラーログに構文エラーが表示された場合は問題を特定します。

1. テーブルがレプリケートデータベースに存在することを確認します。
2. アクセス権限を確認します。

そのデータベースコネクションについて **create connection** コマンドで指定された Replication Server のメンテナンスユーザ ID とパスワードを使用して、レプリケートデータベースにログインします。

このユーザ ID を使用して rs\_lastcommit table テーブルを更新できることを確認します。ダミーエントリをエラーの発生なしに挿入および削除できなければなりません。

3. 実際のコマンドをトレースします。

レプリケート Replication Server (**DSI\_BUF\_DUMP** トレース) または ECDA ゲートウェイでトレースをオンにし、Replication Server コネクションをレジュームします。

レプリケートデータベース接続で ExpressConnect for Oracle または ExpressConnect for SAP HANA データベースを使用している場合、ExpressConnect でトレースをオンにします。

```
alter connection to <rds.rdb> set trace to 'econn,dsi_buf_dump,on'  
go
```

エラーが発生している文を特定し、必要に応じて修正します。

## 想定したデータ型変換が行われない

データ型変換エラーの最も一般的な原因は、必要なユーザ定義データ型 (UDD) と変換のインストールが完全でないことです。

### UDD と変換のインストールの確認

UDD と変換のインストールを確認する方法を説明します。

1. Replication Server を再起動します。Replication Server は、起動時にすべてのファンクション文字列情報をキャッシュします。

RSSD に格納されているファンクション文字列に対するそれ以降の変更は、Replication Server が再起動されるまで有効になりません。

2. クラスレベル変換が、レプリケート Replication Server に適用されていることを確認します。

Replication Server の接続プロファイルには、レプリケート Replication Server の RSSD にクラスレベル変換を適用するために必要な、ASE 以外のプライマリデータベースと ASE 以外のレプリケートデータベースの特定の組み合わせ用の SQL 文が含まれています。

---

**注意：** 接続プロファイルは、ASE 以外のレプリケートデータベースに必要です。たとえば、ASE から Oracle に複写する場合は、rs\_lastcommit テーブルに対する Replication Server の更新が適切に変換され、レプリケートデータベースに適用されるように、変換用の rs\_ase\_to\_oracle 接続プロファイルを適用する必要があります。

---

これらの接続プロファイルは、再実行してもエラーにはなりません。接続プロファイルのコピーが、RSSD のデータベース名に対する正しい **use** 文で更新されていることを確認します。

3. レプリケートデータベースの Replication Server コネクションが、適切なファンクション文字列クラスと関連付けられていることを確認します。

クラスレベル変換を利用するには、レプリケート Replication Server コネクションで、ASE 以外の正しいファンクション文字列クラスが使用されていることが必要です。

Replication Server の **rs\_helpdb** コマンドを使用すると、データベースコネクションに定義されているファンクション文字列クラスを調べることができます。

レプリケートデータベースのファンクション文字列クラスは、次のとおりです。

- Adaptive Server Enterprise - **rs\_sqlserver\_function\_class**
- IBM z/OS プラットフォーム上の DB2 UDB - **rs\_db2\_function\_class**
- UNIX および Windows プラットフォーム上の DB2 UDB - **rs\_udb\_function\_class**
- SAP HANA データベース - **rs\_hanadb\_function\_class**
- Microsoft SQL Server - **rs\_mssql\_function\_class**
- Oracle - **rs\_oracle\_function\_class**
- SAP IQ - **rs\_iq\_function\_class**

アクティブなファンクション文字列クラスのリストを表示するには、Replication Server の **admin show\_function\_classes** コマンドを使用します。

既存のデータベース接続のファンクション文字列クラスを変更するには、Replication Server の **alter connection** コマンドを使用します。

4. ASE 以外のファンクション文字列クラスが、適切なファンクション文字列で更新されていることを確認します。

Replication Server の接続プロファイル `rs_XXX_XXX` には、レプリケート Replication Server の RSSD にファンクション文字列を適用するために必要な、ASE 以外の特定のレプリケートデータベース用の SQL 文が含まれています。

各ファンクション文字列に対して、この接続プロファイルは、**delete** に続いて **insert** を発行します。これらの接続プロファイルは、再実行してもエラーにはなりません。

接続プロファイルのコピーが、RSSD のデータベース名に対する正しい **use** 文で更新されていることを確認します。

5. Replication Server **admin translate** コマンドを使用します。

**admin translate** コマンドを使用すると、特定の変換の結果を確認できます。このコマンドを使用して、想定した変換結果が変換エンジンによって提供されることを確認してください。

異機種データ型サポート (HDS) および **admin translate** コマンドの詳細については、『Replication Server 管理ガイド 第 1 巻』の「複写テーブルの管理」の「HDS を使用したデータ型の変換」を参照してください。

## カラム値の欠落

Replication Server のさまざまな設定が原因となってカラム値が欠落することがあります。

次の場合、レプリケートテーブルで一部のカラム値が欠落することがあります。

- **where** を含むサブスクリプションでローマイグレーションが発生した場合。『Replication Server 管理ガイド 第 1 巻』の「ローマイグレーション」を参照してください。
- 各複写定義に対してオートコレクションを有効にした場合。『Replication Server 管理ガイド 第 1 巻』の「ノンアトミックマテリアライゼーション」と「ノンアトミックマテリアライゼーションのオートコレクション」を参照してください。
- **dsi\_command\_convert** を **u2di** に設定してコマンドを変換した場合。『Replication Server 管理ガイド 第 2 巻』の「コマンド変換」を参照してください。
- **rs\_update** システム関数を **insert** オペレーションを実行するようにカスタマイズした場合。『Replication Server 管理ガイド 第 2 巻』の「データベースオペレーションのカスタマイズ」を参照してください。

カラム値が欠落しないようにするには、**ra\_set\_autocorrection** Replication Agent コマンドを有効にして、Replication Agent から下流の Replication Server にすべてのカラムの値を送信できるようにします。

```
ra_set_autocorrection tablename, enable
```

---

**注意：** **ra\_set\_autocorrection** を有効にすると、複写パフォーマンスに影響が生じる可能性があります。また、有効にしても、レプリケートデータベース内のカラムで LOB 値が欠落するのを防ぐことができない場合があります。

---

『Replication Agent リファレンスマニュアル』の「**ra\_set\_autocorrection**」を参照してください。

## ログ転送言語の生成とトレース

この項では、プライマリ Replication Server に送信されるログ転送言語 (LTL) コマンドをトレースする方法と、Replication Agent のその他の重要なトレースについて説明します。

### **Replication Agent for DB2 UDB for z/OS**

この項で説明する設定パラメータを使用すると、Replication Agent for DB2 UDB for z/OS では通常提供されない追加情報を取得できます。

各ログレコードのログレコード識別子と、DB2 API から受け取った追加のメッセージを印刷するには、**Logtrace = Y** を LTMCFG に入力します。

---

**注意：** 通常、これらのパラメータを使用すると、パフォーマンスに何らかの影響があります。『Replication Agent for DB2 UDB Installation Guide』でパラメータの説明をよく読んでから、使用してください。

---

- Replication Agent のユーザ Exit に渡された情報のデバッグに使用するために、追加のトレースが必要な場合は、**API\_com\_test** 設定パラメータの値を **Y** に設定します。このトレースは、ユーザ Exit が使用されない場合にも使用できます。
- **LTL\_test\_only** 設定パラメータは、LTM for z/OS が Replication Server に接続し、複写のためにトランザクションオペレーションを送信するかどうかを制御します。**LTL\_test\_only** パラメータの値が **Y** の場合、通常は Replication Server に送信される LTL が、代わりに LTLOUT ファイルに書き込まれます。

---

**注意：** **LTL\_test\_only** パラメータの値が **Y** の場合、Replication Agent for DB2 UDB は Replication Server に接続されません。

---

- **trace=LTLebcdic** 設定パラメータは、Replication Server に渡される EBCDIC LTL を LTLOUT に書き込みます。ASCII データを含むテーブルを複写する場合は、ASCII 文字が LTLOUT データセットに書き込まれるように、**trace = LTLASCII** を

設定します。このトレースをオンにするには、これらのパラメータの値を **Y** に設定してください。

- **Use\_repdef** 設定パラメータを使用すると、LTM for z/OS が複写定義で指定されたカラムのみを格納する LTL を Replication Server に送信するように設定できます。

**use\_repdef** パラメータの値を **N** に設定すると、LTL トレースで提供される情報量が増える可能性があります。

- **suppress\_col\_names** 設定パラメータは、LTM for z/OS が、Replication Server に送信される LTL で、カラム名を非表示にするかどうかを決定します。  
LTL の出力をトレースしている場合、**suppress\_col\_names** の値を **N** に設定すると、生成された LTL でカラム名が表示されます。

### **Replication Agent**

この項で説明するトレースフラグおよび設定パラメータを使用すると、Microsoft SQL Server、Oracle、UDB 用の Replication Agent では通常提供されない追加情報を取得できます。

---

**注意：** 通常、これらのトレースフラグおよびパラメータを使用すると、パフォーマンスに何らかの影響があります。フラグまたはパラメータを使用する前に、『Replication Agent 管理ガイド』の説明をよく読んでください。

---

#### トレースフラグ

通常のトレース出力は、Replication Agent インスタンスのログファイルに送信されます。ただし、**LTITRACELTL** トレースポイントからの出力は、別の LTL 出力ログファイル (**LTITRACELTL.log**) に送信されます。

Replication Agent の問題をトラブルシューティングするには、以下のトレースフラグが特に便利です。

- **LRTRACE** - Log Reader コンポーネントの通常の実行をトレースします。
- **LTITRACE** - Log Transfer Interface コンポーネントの通常の実行をトレースします。
- **LTITRACELTL** - **LTITRACELTL.log** ファイルでの LTL 文のトレースを有効にします。
- **RACONTRC** - コネクションおよびクエリの実行をトレースします。
- **RACONTRCSQL** - プライマリデータベースに送信される SQL 文をトレースします。

#### 設定パラメータ

Replication Agent の以下の設定パラメータの設定は、トレース情報に影響します。

- **compress\_ltl\_syntax - false** に設定すると、LTL コマンドのより詳細な説明が表示されます。
- **connect\_to\_rs - false** に設定すると、Replication Server への実際の接続や情報の送信を伴わない LTL が生成されます。
- **log\_trace\_verbose - true** に設定されている場合、トレースされるコンポーネントのより詳細な説明を提供します。
- **use\_rssd - false** に設定されていると、複写定義情報の変更を伴わない LTL コマンド全体の生成を提供します。
- **column\_compression - false** に設定すると、**update** オペレーションを目的として生成された LTL 内のすべてのカラム情報 (更新後イメージ内のすべてのカラム) が送信されます。

Replication Agent のトレースフラグと設定の詳細については、『Replication Agent 管理ガイド』を参照してください。





# Oracle から Oracle への複製のリファレンス実装

Replication Server には、ご使用の環境で使用可能な製品を使って、Oracle から Oracle への複製のリファレンス実装をすばやくセットアップするためのツールセットが含まれています。

リファレンスとして複製環境を実装すると、Replication Server のさまざまな機能を試すことができます。ツールセットを使用すると、次の手順を実行できます。

1. Replication Server、プライマリデータベース、レプリケートデータベースを構築します。
2. データベース複製環境を設定します。
3. プライマリデータベースで単純なトランザクションを実行し、データベースレベルの複製を使用して変更を複製します。
4. 手順 3 の複製処理から統計とモニタカウンタを収集します。
5. リファレンス複製環境をクリーンアップします。

リファレンス実装ツールセットはスクリプトで構成されており、`$$SYBASE/REP-15_5/REFIMP-01_0` にあります。

---

**注意：** リファレンス実装は、単一の Replication Server、プライマリデータベースサーバ、レプリケートデータベースサーバのみを提供します。複数の複製システムコンポーネント用のリファレンス環境トポロジは設定できません。

---

## プラットフォームのサポート

---

リファレンス環境は、Linux on IBM p-Series (Linux on Power) 64 ビット版を除く、Replication Server がサポートしているすべてのプラットフォームに実装できます。

ただし、リファレンス環境を Replication Server がサポートする Microsoft Windows プラットフォームでセットアップするには、Cygwin を使ってリファレンス実装スクリプトを実行する必要があります。Cygwin Web サイト：<http://www.cygwin.com/> を参照してください。

## Oracle のリファレンス実装のコンポーネント

---

リファレンス環境を実装するには、複写環境でサポートされているバージョンのコンポーネントが必要です。

Oracle のリファレンス実装では、次の製品コンポーネントバージョンがサポートされています。

表 21 : Oracle のリファレンス実装でサポートされている製品コンポーネントのバージョン

| Replication Server    | Oracle        | Replication Agent for Oracle | ExpressConnect for Oracle |
|-----------------------|---------------|------------------------------|---------------------------|
| 15.5、15.6、15.7、15.7.1 | 11g Release 2 | 15.7.1 SP100                 | 15.7.1 SP100              |

たとえば、Oracle のリファレンス実装環境を構築するには、Replication Server 15.7.1、Oracle 11g Release 2、Replication Agent 15.7.1 SP100 for Oracle、および ExpressConnect for Oracle 15.7.1 SP100 が必要です。

## リファレンス環境の前提条件の確認

---

リファレンス環境にはいくつかの前提条件があります。

リファレンス実装の手順では、SAP Replication Server リリースディレクトリにある `interfaces` ファイルを使用します。リファレンス実装の手順を実行する前にこのファイルが存在する場合、この手順の実行時にファイル名拡張子が追加されて既存のファイルがバックアップされます。

Oracle の場合、リファレンス実装の手順で既存のファイル (`tnsname.ora`、`listener.ora`) の名前が変更され、Oracle リファレンス実装用に新しいファイルが作成されます。

1. Oracle の場合、Oracle リリースディレクトリでの `execute` パーミッションがあることを確認してください。たとえば、インスタンスを手動で作成できるかどうかを確認します。
2. SAP Replication Server リリースディレクトリにある `SYBASE.sh` ファイルの環境変数設定が正しいことを確認します。正しいことを確認できない場合は、このファイルを削除するか、名前を変更します。

- ご使用の bash シェルで、UNIX の **grep**、**kill**、**awk**、および **ps** の各コマンドを使用できることを確認します。

## Oracle 用のリファレンス実装設定ファイル

---

SAP では、サポートされている UNIX および Microsoft Windows プラットフォーム上での Oracle から Oracle への複写用の設定ファイルテンプレートを提供しており、ご使用の環境用の設定ファイルを作成できます。

このファイルは、`$SYBASE/REP-15_5/REFIMP-01_0/config` にあります。

表 22 : リファレンス実装の設定ファイル

| プライマリとレプリケートのデータサーバ、プラットフォーム | 設定ファイル                           |
|------------------------------|----------------------------------|
| Oracle から Oracle、UNIX        | <code>ora_unix_refimp.cfg</code> |
| Oracle から Oracle、Windows     | <code>ora_win_refimp.cfg</code>  |

リファレンス環境の要件、手順、設定ファイルのサンプル構築、リファレンス環境の実装で作成されるオブジェクトの詳細については、『Replication Server 管理ガイド 第 2 巻』の「リファレンス複写環境の実装」を参照してください。



## 用語解説

複写システムで使用される用語を解説します。

- **アクティブデータベース** – ウォームスタンバイアプリケーションのスタンバイデータベースに複写されるデータベースです。「ウォームスタンバイアプリケーション」も参照してください。
- **アプリケーションプログラミングインタフェース (API)** – ユーザまたはプログラムが相互に通信するために使用する、事前に定義されたインタフェースです。Open Client™ および SAP Open Server は、クライアント/サーバアーキテクチャで通信を行う API の 1 つです。Replication Command Language (RCL) は、SAP Replication Server の API です。
- **適用ファンクション** – ファンクション複写定義に対応する複写ファンクションであり、SAP Replication Server によってプライマリデータベースからサブスクライブ元のレプリケートデータベースに配信されます。「**複写ファンクションの配信**」、「**要求ファンクション**」、「**ファンクション複写定義**」も参照してください。
- **アークル** – テーブルまたはストアドプロシージャの複写定義を拡張したもので、パブリケーションの要素となります。アークルには、レプリケートデータベースが受信するローのサブセットを指定した **where** 句が含まれている場合もあれば、含まれていない場合もあります。
- **非同期プロシージャ配信** – プライマリデータベースまたはレプリケートデータベースで複写するように指定されたストアドプロシージャを実行できる Replication Server システムの一部です。
- **非同期コマンド** – クライアントが SAP Replication Server に送信するコマンドです。クライアントは、完了ステータスの受信を待たずに、他のオペレーションを継続できます。SAP Replication Server のコマンドの多くは、複写システム内で非同期コマンドとして動作します。
- **アトミックマテリアライゼーション** – マテリアライゼーションメソッドの 1 つです。select オペレーションを holdlock を指定して使用し、1 つのアトミックオペレーションでネットワークを介して、プライマリデータベースからレプリケートデータベースへサブスクリプションデータをコピーします。データの転送が完了するまで、プライマリデータへの変更は行えません。「**ノンアトミックマテリアライゼーション**」、「**バルクマテリアライゼーション**」、「**非マテリアライゼーション**」も参照してください。
- **オートコレクション** – オートコレクションは、複写定義に適用する機能で、レプリケートテーブルに、消失ローや重複したローが発生して障害が起こることを防ぎます。**set autocorrection** コマンドを使用して設定します。オートコレクションを有効にすると、SAP Replication Server は各更新オペレーションまたは

挿入オペレーションを削除と挿入の連続オペレーションに変換します。オートコレクションは、サブスクリプションがノンアトミックマテリアライゼーションを使用している複写定義についてのみ有効にしてください。

- **基本クラス** – 親クラスからファンクション文字列を継承しないファンクション文字列クラスです。「ファンクション文字列クラス」も参照してください。
- **ビットマップサブスクリプション** – ビットマップの比較に基づいてローを複写するサブスクリプションの種類です。int データ型のカラムを作成し、複写定義を作成するときには、カラムを rs\_address データ型として指定します。
- **バルクコピーイン** – SAP ASE 12.0 以降で、大量の insert 文を同じテーブルで複写するとき SAP Replication Server のパフォーマンスを向上させる機能です。SAP Replication Server は、Open Client™ Open Server™ Bulk-Library を使用して、レプリケートデータベースにトランザクションを送信する SAP Replication Server モジュールであるデータサーバインタフェース (DSI) にバルクコピーインを実装します。

バルクコピーインにより、サブスクリプションマテリアライゼーションのパフォーマンスも向上します。dsi\_bulk\_copy を on にすると、各トランザクションの insert コマンドの数が dsi\_bulk\_threshold を超えた場合に、SAP Replication Server は、バルクコピーインを使用してサブスクリプションをマテリアライズします。

- **バルクマテリアライゼーション** – マテリアライゼーションのメソッドの1つです。これは、複写システム以外でレプリケートデータベースのサブスクリプションのデータを初期化します。バルクマテリアライゼーションは、テーブル複写定義とファンクション複写定義のどちらのサブスクリプションにも使用できます。たとえば、磁気テープ、フロッピーディスク、CD-ROM、または光磁気ディスクなどのメディアを使用して、プライマリデータベースからデータを転送できます。バルクマテリアライゼーションでは、define subscription から始まる一連のコマンドを使用します。「アトミックマテリアライゼーション」、「ノンアトミックマテリアライゼーション」、「非マテリアライゼーション」も参照してください。
- **集中型データベースシステム** – 中央サイトに設置された1つのデータベース管理システムでデータを一元管理するデータベースシステムです。
- **クラス** – 「エラークラス」と「ファンクション文字列クラス」を参照してください。
- **クラスツリー** – 派生クラスと親クラスの複数のレベルから構成されるファンクション文字列クラスのセットです。これは、同じ基本クラスから派生します。「ファンクション文字列クラス」も参照してください。
- **クライアント** – クライアント/サーバアーキテクチャにおいて、サーバに接続されたプログラムです。ユーザが実行するフロントエンドアプリケーションプログラムの場合もあれば、システムの拡張機能として実行されるユーティリティプログラムの場合もあります。

- **Client/Server Interfaces (C/SI)** – クライアント/サーバアーキテクチャで実行するプログラムのための SAP のインタフェース標準です。
- **同時実行性** – 複数のクライアントが、データまたはリソースを共有できることを示します。データベース管理システムにおける同時実行性は、あるクライアントが使用中のデータを別のクライアントが変更しようとするときに発生する競合からクライアントを保護するシステムに依存します。
- **接続** – SAP Replication Server からデータベースへの接続です。「データサーバインタフェース (DSI)」と「論理接続」も参照してください。
- **接続プロファイル** – データベース接続を確立するために必要な情報です。
- **コーディネートダンプ** – 複写システムでファンクション `rs_dumpdb` または `rs_dumptran` を実行することによって、複数のサイト間で同期がとられているデータベースダンプセット、またはトランザクションダンプセットです。
- **データベース** – 相互に関連するデータテーブルとその他のオブジェクトが、特定の目的に合わせて編成、表現されたものです。
- **データベース世代番号** – データベースおよびデータベースを管理する SAP Replication Server の RSSD に格納されます。データベース世代番号は、各ログレコードのオリジンキュー ID (*qid*) の最初の部分です。オリジンキュー ID は、SAP Replication Server が重複したレコードを処理していないことを示します。リカバリオペレーション時には、データベースの再ロード後に送信されたレコードが SAP Replication Server に無視されないようにするために、データベース世代番号を増やす必要がある場合があります。
- **データベース複写定義** – サブスクリプションを作成できる対象のデータベースオブジェクト (テーブル、トランザクション、ファンクション、システムストアプロシージャ、DDL) を集めて記述したものです。

テーブル複写定義とファンクション複写定義も作成できます。「テーブル複写定義」と「ファンクション複写定義」も参照してください。
- **データベースサーバ** – SAP ASE などのサーバアプリケーションです。クライアントにデータベース管理サービスを提供します。
- **データ定義言語 (DDL)** – Transact-SQL などのクエリ言語のコマンドセットであり、データとデータベース内でのデータの間を記述します。Transact-SQL の DDL コマンドには、**create**、**drop**、**alter** キーワードを使用するものがあります。
- **データ操作言語 (DML)** – Transact-SQL などのクエリ言語のコマンドセットであり、データの操作を行います。Transact-SQL の DML コマンドには、**select**、**insert**、**update**、**delete** があります。
- **データサーバ** – SAP クライアント/サーバインタフェースに準拠のクライアントインタフェースによって、データベースのレプリケートテーブルの物理表現を管理するのに必要な機能を提供するサーバです。データサーバは、通常、

データベースサーバと同じものですが、SAP Replication Server に必要なインタフェースと機能を備えたデータリポジトリの場合もあります。

- **データサーバインタフェース (DSI)** – SAP Replication Server とデータベース間の接続に対応している SAP Replication Server スレッドです。DSI スレッドは、DSI アウトバウンドキューからレプリケートデータサーバへトランザクションを送信します。DSI スレッドは1つのスケジューラスレッドと1つまたは複数のエグゼキュータスレッドで構成されます。スケジューラスレッドは、トランザクションをコミット順にグループ分けしてから、それらをエグゼキュータスレッドにディスパッチします。エグゼキュータスレッドは、ファンクションをファンクション文字列にマッピングし、レプリケートデータベースのトランザクションを実行します。DSI スレッドは、データベースへの Open Client 接続を使用します。「アウトバウンドキュー」と「接続」も参照してください。
- **データソース** – リレーショナルデータサーバまたはノンリレーショナルデータサーバなどのデータベース管理システム (DBMS) 製品、その DBMS にあるデータベース、および複製システムの他のコンポーネントから DBMS にアクセスするための通信方法の組み合わせからなる概念をデータソースといいます。「データベース」と「データサーバ」も参照してください。
- **意思決定支援アプリケーション** – アドホッククエリ、レポート、計算などの処理が行え、少数データの更新トランザクションを特徴とするデータベースクライアントアプリケーションです。
- **宣言したデータ型** – Replication Agent から SAP Replication Server に配信された値のデータ型です。
  - Replication Agent が datetime などの基本 SAP Replication Server データ型を SAP Replication Server に配信する場合、宣言したデータ型は基本データ型です。
  - 上記以外の場合、宣言したデータ型は、プライマリデータベースの元のデータ型に対する UDD でなければなりません。
- **デフォルトのファンクション文字列** – システム提供クラス rs\_sqlserver\_function\_class と rs\_default\_function\_class、およびこれらのクラスからファンクション文字列を直接的または間接的に継承するクラスに対してデフォルトで提供されるファンクション文字列です。「ファンクション文字列」も参照してください。
- **マテリアライゼーション解除** – サブスクリプションが削除されたときに、適宜実行される処理です。これによって、他のサブスクリプションに使用されていない特定のローが、レプリケートデータベースから削除されます。
- **派生クラス** – 親クラスからファンクション文字列を継承するファンクション文字列クラスです。「ファンクション文字列クラス」と「親クラス」も参照してください。



- **直接ルート** – 中間 SAP Replication Server を使用せずに、送信元 SAP Replication Server から送信先 Replication Server へ直接メッセージを送信するために使用するルートです。「**間接ルート**」と「**ルート**」も参照してください。
- **ディスクパーティション** – 「**パーティション**」を参照してください。
- **分散データベースシステム** – データをネットワーク上にある複数のデータベースに格納するデータベースシステムです。
- **ディストリビュータ** – インバウンドキューにある各トランザクションの送信先を決定するために使用する SAP Replication Server スレッド (DIST) です。
- **ダンプマーカ** – ダンプの実行時に SAP ASE がデータベーストランザクションログに書き込むメッセージです。ウォームスタンバイアプリケーションでは、アクティブデータベースのデータでスタンバイデータベースを初期化するとき、SAP Replication Server がダンプマーカを使用して、トランザクションストリームの中からトランザクションをスタンバイデータベースに適用するかを決定するように指定できます。「**ウォームスタンバイアプリケーション**」も参照してください。
- **Embedded Replication Server システムデータベース (ERSSD)** – SAP Replication Server システムテーブルを格納する SAP SQL Anywhere データベースです。SAP Replication Server システムテーブルを ERSSD と SAP ASE RSSD のどちらに格納するかを選択できます。「**Replication Server システムデータベース (RSSD)**」も参照してください。
- **Enterprise Connect™ Data Access (ECDA)** – LAN ベースの ASE 以外のデータソースやメインフレームのデータソースなど、異機種データベース環境にあるデータへのアクセスを可能にするソフトウェアアプリケーションと接続ツールを統合したセットです。
- **エラーアクション** – データサーバのエラーに対する SAP Replication Server の応答処理です。SAP Replication Server のエラーアクションの種類としては、**ignore**、**warn**、**retry\_log**、**log**、**retry\_stop**、**stop\_replication** があります。エラーアクションは、特定のデータサーバエラーに割り当てられます。
- **エラークラス** – 指定したデータベースで使用するデータサーバのエラーアクションの集まりに名前を付けたものです。
- **例外ログ** – データサーバ上で失敗したトランザクションの情報を保存する SAP Replication Server の 3 つのシステムテーブルがセットになったものです。例外ログに記録されたトランザクションは、ユーザまたはインテリジェントアプリケーションが処理しなければなりません。例外ログのクエリを行うには、**rs\_helpexception** ストアドプロシージャを使用します。
- **ExpressConnect for HANA データベース** – SAP Replication Server と SAP HANA データベース間の直接通信に使用できるライブラリのセットです。
- **ExpressConnect for Oracle** – SAP Replication Server と Oracle データベース間の直接通信に使用できるライブラリのセットです。

- **フェールオーバ** – SAP フェールオーバを使用すると、バージョン 12.0 以降の 2 つの SAP ASE をコンパニオンとして設定できます。プライマリコンパニオンに障害が発生した場合、そのサーバのデバイス、データベース、接続をセカンダリコンパニオンが引き継ぐことができます。

SAP ASE における SAP フェールオーバの動作の詳細については、『高可用性システムにおける SAP フェールオーバの使用』を参照してください。これは、SAP ASE のマニュアルセットの一部です。

- **フォールトトレランス** – 1 つまたは複数のコンポーネントで障害が発生した場合でも、システムが正常に処理を継続できるシステムの機能です。
- **ファンクション** – insert、delete、select、begin transaction などのデータサーバのオペレーションを表す SAP Replication Server オブジェクトです。SAP Replication Server は、これらのオペレーションをファンクションとして他の SAP Replication Server に配信します。各ファンクションは、ファンクション名とデータパラメータのセットから構成されます。ファンクションを送信先データベースで実行するために、SAP Replication Server はファンクション文字列を使用して、そのファンクションを特定タイプのデータベース用のコマンドまたはコマンドセットに変換します。「**ユーザ定義ファンクション**」と「**複写ファンクションの配信**」も参照してください。
- **ファンクション複写定義** – 複写ファンクションの配信で使用される、複写ファンクションの記述です。ファンクション複写定義は SAP Replication Server によって管理され、複写されるパラメータや、影響を受けるデータのプライマリバージョンがあるロケーションに関する情報などからなります。「**複写ファンクションの配信**」も参照してください。
- **ファンクションのスコープ** – ファンクションの適用範囲です。ファンクションは、複写定義スコープまたはファンクション文字列クラススコープを持ちます。複写定義スコープを持つファンクションは、特定の複写定義に指定され、他の複写定義には適用できません。ファンクション文字列クラススコープを持つファンクションは、ファンクション文字列クラスに対して 1 回だけ定義され、そのクラスでのみ使用できます。
- **ファンクション文字列** – SAP Replication Server が、ファンクションとそのパラメータをデータサーバの API にマップするために使用する文字列です。ファンクション文字列により、SAP Replication Server は、プライマリデータベースとレプリケートデータベースでタイプが異なる異機種間複写を、異なる SQL 拡張機能とコマンド機能でサポートできます。
- **ファンクション文字列クラス** – 指定したデータベース接続で使用される、名前付きのファンクション文字列のコレクションです。ファンクション文字列クラスには、SAP Replication Server によって提供されるものとユーザが作成したものがあります。ファンクション文字列クラスは、ファンクション文字列継承によってファンクション文字列定義を共有できます。システムで提供されるファンクション文字列クラスには、rs\_sqlserver\_function\_class、

rs\_default\_function\_class、rs\_db2\_function\_class の3つがあります。「基本クラス」、「クラスツリー」、「派生クラス」、「ファンクション文字列継承」、「親クラス」も参照してください。

- **ファンクション文字列の継承** – クラス間でファンクション文字列定義を共有する機能です。この機能によって、派生クラスは親クラスからファンクション文字列を継承します。「派生クラス」、「ファンクション文字列クラス」、「親クラス」も参照してください。
- **ファンクション文字列変数** – 実行時に代入される値を表すために、ファンクション文字列内で使用する識別子です。ファンクション文字列内の変数は、疑問符 (?) で囲まれています。この変数は、カラムの値、ファンクションのパラメータ、システム定義の変数、ユーザ定義の変数を表します。
- **ファンクションサブスクリプション** – ファンクション複写定義に対するサブスクリプションです (適用ファンクションおよび要求ファンクションの配信で使用されます)。
- **ゲートウェイ** – 異なるネットワークアーキテクチャを持つ複数のコンピュータシステム間での通信を可能にする接続ソフトウェアです。
- **世代番号** – 「データベース生成番号」を参照してください。
- **異機種データサーバ** – 同じ分散データベースシステム内で使用される複数ベンダのデータサーバです。
- **ハイバネーションモード** – SAP Replication Server の状態です。この状態では、admin と sysadmin コマンドを除くすべてのデータ定義言語 (DDL) コマンドは拒否され、すべてのルートとコネクション、およびデータサーバインタフェース (DSI)、SAP Replication Server インタフェース (RSI) などのほとんどのサービススレッドがサスペンドされます。また、RSI と Replication Agent ユーザはログオフされログオンできません。これは、ルートのアップグレード中に使用され、SAP Replication Server が問題をデバッグするためにオン状態になることがあります。
- **High-Performance Analytic Appliance (HANA)** – SAP® インメモリオンライントランザクション処理/オンライン分析処理ソリューション。
- **High-Performance Analytic Appliance データベース (SAP HANA データベース)** – SAP インメモリデータベース。
- **高可用性 (HA)** – ダウン時間が非常に少ないことです。HA を提供するコンピュータシステムは、通常、99.999% の可用性 (予定外のダウン時間が、年間約5分) を実現しています。
- **High Volume Adaptive Replication (HVAR)** – 最終的な結果とそれ以降のレプリケートデータベースへの最終的な結果のバルク適用を生成する、insert、delete、update の各オペレーションのグループのコンパイルです。
- **ホットスタンバイアプリケーション** – クライアントアプリケーションを中断したり、トランザクションを失ったりすることなく、スタンバイデータベースを

アクティブに切り替えられるデータベースアプリケーションです。「ウォームスタンバイアプリケーション」も参照してください。

- **ID サーバ** – 複写システムのいずれか 1 つの SAP Replication Server が、ID サーバとなります。ID サーバは、SAP Replication Server の通常の作業の他に、複写システムにあるすべての SAP Replication Server とデータベースにユニークな ID 番号を割り当て、複写システムのバージョン情報を管理します。
- **インバウンドキュー** – Replication Agent から SAP Replication Server へのメッセージをスプールするために使用されるステابلキューです。
- **間接ルート** – 送信元 SAP Replication Server から送信先 SAP Replication Server へ、1 つ以上の中間 SAP Replication Server を経由してメッセージを送るために使用するルートです。「直接ルート」と「ルート」も参照してください。
- **interfaces ファイル** – SAP クライアント/サーバアーキテクチャ上のサーバプログラムが使用する、ネットワークのアクセス情報を定義するエントリのあるファイルです。サーバプログラムには、SAP ASE、ゲートウェイ、SAP Replication Server、Replication Agent があります。クライアントとサーバは、interfaces ファイルにあるエントリを使用して、ネットワーク上で相互に接続できます。
- **遅延時間** – プライマリデータベースで最初に適用されたデータ修正オペレーションが、レプリケートデータベースに分配されるまでに要する時間の単位です。この時間には、Replication Agent での処理時間、SAP Replication Server での処理時間、ネットワークのオーバーヘッドなどが含まれます。
- **ローカルエリアネットワーク (LAN)** – コンピュータとプリンタや端末などのデバイスを、データやデバイスの共有のためにケーブルで接続したシステムです。
- **ロケータ値** – SAP Replication Server の RSSD の rs\_locator テーブルに格納されている値です。この値によって、複写中に SAP Replication Server によって受信および確認された、直前の各サイトからの最新のログトランザクションレコードが特定されます。
- **論理コネクション** – SAP Replication Server が、ウォームスタンバイアプリケーションのアクティブデータベースとスタンバイデータベースとのコネクションにマップするデータベース接続です。「コネクション」と「ウォームスタンバイアプリケーション」も参照してください。
- **ログイン名** – ユーザまたは SAP Replication Server などのシステムコンポーネントがデータサーバ、SAP Replication Server、または Replication Agent にログインするために使用する名前です。
- **ログ転送言語 (LTL)** – Replication Command Language (RCL) のサブセットです。プライマリデータベースのトランザクションログから取得した情報は、RepAgent などの Replication Agent によって、LTL コマンドを使用して、SAP Replication Server に送信されます。

- **Log Transfer Manager (LTM)** – SAP SQL Server 用の Replication Agent プログラムです。「*Replication Agent*」と「*RepAgent* スレッド」も参照してください。
- **メンテナンスユーザ** – SAP Replication Server がレプリケートデータを管理するために使用するデータサーバのログイン名です。ほとんどのアプリケーションでは、メンテナンスユーザのトランザクションは複写されません。
- **マテリアライゼーション** – プライマリデータベースからレプリケートデータベースへ、サブスクリプションによって指定されたデータをコピーする処理です。これによって、レプリケートテーブルが初期化されます。レプリケートデータはネットワークを介して転送するか、またはサブスクリプションが大量のデータを扱う場合は、メディアからロードできます。「*アトミックマテリアライゼーション*」、「*バルクマテリアライゼーション*」、「*非マテリアライゼーション*」、「*ノンアトミックマテリアライゼーション*」も参照してください。
- **マテリアライゼーションキュー** – マテリアライゼーションまたはマテリアライゼーション解除されているサブスクリプションに関連したメッセージをスプールするために使用されるステابلキューです。
- **消失ロー** – プライマリテーブルには存在するが、そのテーブルの複写コピーには存在しないローです。
- **混合バージョンシステム** – ソフトウェアバージョンとサイトバージョンの違いによって異なる機能を持った、ソフトウェアバージョンの異なる SAP Replication Server がある複写システムです。混合バージョンサポートは、システムバージョンが 11.0.2 以降の場合のみ使用できます。

たとえば、SAP Replication Server バージョン 11.5 以降とバージョン 11.0.2 がある複写システムは、混合バージョンシステムです。バージョン 11.0.2 以降の SAP Replication Server では、特定の新機能の使用がシステムバージョンによって制限されていますが、それより前のバージョンではこの機能がサポートされていないため、バージョン 11.0.2 より前の SAP Replication Server を持った複写システムは、混合バージョンシステムではありません。「*サイトバージョン*」と「*システムバージョン*」も参照してください。

- **カラム数の増加** – 250 を超え、最大 1024 までの複写定義内のカラム数の増加のことです。カラム数の増加は、SAP Replication Server バージョン 12.5 以降でサポートされています。
- **Multi-Site Availability (MSA)** – テーブル、ファンクション、トランザクション、システムストアドプロシージャ、データ定義言語 (DDL) 文などのデータベースオブジェクトをプライマリデータベースからレプリケートデータベースへ複写する方法です。「*データベース複写定義*」も参照してください。
- **Multi-Path Replication™** – 送信元データベースからターゲットデータベースへのデータの並列パスを有効にすることによってパフォーマンスを向上させる SAP Replication Server の機能です。Multi-Path Replication™ は、ウォームスタンバイ環境と Multi-Site Availability (MSA) 環境で設定できます。これらの複数のパスではデータが個別に処理され、それらのパス間のトランザクションの一貫

性を必要とせずにデータセットを並列処理できる場合に適用されます。パス内でのデータ整合性を維持しますが、さまざまなパス間でのコミット順には従いません。

- **ネームスペース** – オブジェクト名がユニークでなければならない範囲 (スコープ) です。
- **ノンアトミックマテリアライゼーション** – マテリアライゼーションのメソッドの1つです。これは、`holdlock` を使わずに1つのオペレーションで、ネットワークを介してプライマリデータベースからレプリケートデータベースへサブスクリプションデータをコピーします。データの転送中もプライマリテーブルを変更できるので、レプリケートデータベースとプライマリデータベース間で一時的に不一致が生じる可能性があります。データは、レプリケートデータベースのトランザクションログが満杯にならないように、トランザクションごとに10ローずつ挿入する方法を使用して適用されます。ノンアトミックマテリアライゼーションは、`create subscription` コマンドのオプションのメソッドです。「オートコレクション」、「アトミックマテリアライゼーション」、「非マテリアライゼーション」、「バルクマテリアライゼーション」も参照してください。
- **ネットワークベースセキュリティ** – ネットワーク上でのデータの安全な転送です。SAP Replication Server は、ユーザの認証、統一化ログイン、SAP Replication Server 間の安全なメッセージ転送などのサードパーティのセキュリティメカニズムをサポートします。
- **非マテリアライゼーション** – マテリアライゼーションのメソッドの1つです。サブスクリプションデータがレプリケートサイトにすでに存在する場合、サブスクリプションを作成できます。`without materialization` 句を指定して `create subscription` コマンドを使用してください。このメソッドを使用して、テーブル複写定義のサブスクリプションを作成することもできます。「アトミックマテリアライゼーション」と「バルクマテリアライゼーション」も参照してください。
- **オンライントランザクション処理 (OLTP) アプリケーション** – データ修正 (挿入、削除、更新) を伴うさまざまなトランザクションを頻繁に実行するデータベースクライアントアプリケーションです。
- **オリジンキュー ID (qid)** – qid は、Replication Agent によって形成され、SAP Replication Server に渡された各ログレコードをユニークに識別します。日付、タイムスタンプ、およびデータベース世代番号が含まれます。「データベース世代番号」も参照してください。
- **孤立したロー** – レプリケートデータベースにあって、プライマリデータベースにはないテーブルローです。
- **アウトバウンドキュー** – メッセージをスプールするのに使用するステーブルキューです。DSI アウトバウンドキューは、レプリケートデータベースへの

メッセージをスプールします。RSI アウトバウンドキューは、レプリケート SAP Replication Server へのメッセージをスプールします。

- **並列 DSI** – 単一のデータサーバインタフェース (DSI) スレッドではなく、並列で機能する複数の DSI スレッドを使用して、レプリケートデータサーバにトランザクションが適用されるようにデータベース接続を設定する方法です。「コネクション」と「データサーバインタフェース (DSI)」も参照してください。
- **パラメータ** – プロシージャの実行時に提供される値を表す識別子です。ファンクション文字列で使用するパラメータ名は @ 記号で始まります。プロシージャをファンクション文字列から呼び出すと、SAP Replication Server はパラメータ値をそのまま変更しないでデータサーバへ渡します。「サーチャブルパラメータ」も参照してください。
- **親クラス** – 派生クラスがファンクション文字列を継承する、ファンクション文字列クラスです。「ファンクション文字列クラス」と「派生クラス」も参照してください。
- **パーティション** – SAP Replication Server が、ステーブルキューを格納するために使用するローディスクパーティションまたはオペレーティングシステムファイルです。オペレーティングシステムファイルはテスト環境でのみ使用してください。
- **物理コネクション** – SAP Replication Server からデータベースへの接続です。
- **プライマリデータ** – 複写システム内で最も信頼できるデータセットのバージョンです。プライマリデータは、データサーバによって管理されます。このデータサーバは、データのサブスクリプションがあるすべての SAP Replication Server で認識されています。
- **プライマリデータベース** – 複写システムによって別のデータベースに複写されるデータが格納されたデータベースです。
- **プライマリフラグメント** – 一連のローのプライマリバージョンを保持するテーブルの水平方向セグメントです。
- **プライマリキー** – 各ローをユニークに識別するテーブルカラムのセットです。
- **プライマリサイト** – 通常のビジネスオペレーションをサポートするために、プライマリデータサーバおよびプライマリデータベースが展開されるロケーションまたは環境です。アクティブサイトまたはメインサイトとも呼ばれます。「エラークラス」と「ファンクション文字列クラス」を参照してください。
- **プリンシパルユーザ** – アプリケーションを開始するユーザです。ネットワークベースのセキュリティを使用する場合、SAP Replication Server はプリンシパルユーザとしてリモートサーバにログインします。
- **プロファイル** – SAP Replication Server が接続するサーバに関する事前定義済みのプロパティセットにより接続を設定できます。

- **射影** – テーブルの垂直方向のスライスです。テーブルカラムのサブセットを表します。
- **パブリケーション** – 同じプライマリデータベースからのアーティクルのグループです。パブリケーションを使用すると、関連するテーブルカストアドプロシージャまたはその両方の複写定義を収集して、グループとしてそれらをサブスクライブできます。送信元 SAP Replication Server のパブリケーション内でアーティクルとして複写定義を収集し、送信先 SAP Replication Server でパブリケーションサブスクリプションを使用してそれらをサブスクライブできます。「アーティクル」と「パブリケーションサブスクリプション」も参照してください。
- **パブリケーションサブスクリプション** – パブリケーションへのサブスクリプションです。「アーティクル」と「パブリケーション」も参照してください。
- **パブリッシュデータ型** – レプリケートデータサーバにおけるカラムレベル変換後(続いてクラスレベル変換をする場合はその前)のカラムのデータ型です。パブリッシュデータ型は、SAP Replication Server 基本データ型か、ターゲットデータサーバのデータ型に対する UDD のどちらかでなければなりません。パブリッシュデータ型が複写定義から省略された場合、デフォルトで宣言したデータ型になります。
- **クエリ** – データベース管理システムで、指定した基準を満たすデータを取得するための要求です。SQL データベース言語では、クエリを指定するときに **select** コマンドを使用します。
- **クワイス状態** – ログスキャンが停止して、すべてのスキャン済みレコードが複写システムの送信先に送信された状態です。一部の Replication Agent のコマンドおよび SAP Replication Server のコマンドでは、複写システムを最初にクワイスする必要があります。
- **引用符付き識別子** – スペースや英数字以外の特殊文字が含まれるオブジェクト名、アルファベット以外の文字で始まるオブジェクト名、予約語に相当するオブジェクト名は、正しく解析されるように引用符 (一重または二重) で囲む必要があります。
- **Real-Time Loading (RTL)** – SAP IQ データベースへの High Volume Adaptive Replication (HVAR)。HVAR の変更を SAP® IQ レプリケートデータベースに適用するには、関連するコマンドとプロセスを使用します。「*High Volume Adaptive Replication*」を参照してください。
- **リモートプロシージャコール (RPC)** – リモートサーバに常駐しているプロシージャを実行するための要求です。プロシージャを実行するサーバには、SAP ASE、SAP Replication Server、または SAP Open Server を使用して構築されたサーバなどがあります。プロシージャの実行要求は、これらのサーバやクライアントアプリケーションから発行できます。RPC 要求のフォーマットは、SAP Client/Server Interfaces の一部です。



- **RepAgent スレッド** – SAP ASE データベース用の Replication Agent です。Replication Agent は SAP ASE のスレッドです。プライマリデータベースから SAP Replication Server にトランザクションログ情報を転送して、他のデータベースに分配します。
- **レプリケートデータベース** – 複写システムによって別のデータベース (プライマリデータベース) から複写されたデータが格納されたデータベースです。レプリケートデータベースは、複写システムで複写されたデータを受信するデータベースです。「プライマリデータベース」と比較してください。
- **複写ファンクションの配信** – ファンクション複写定義に対応するストアプロシージャを送信元データベースから送信先データベースに複写する方法です。「適用ファンクション」、「要求ファンクション」、「ファンクション複写定義」も参照してください。
- **複写ストアプロシージャ** – `sp_setrepproc` システムプロシージャを使用して、複写するようにマーク付けされた SAP ASE ストアドプロシージャです。複写ストアプロシージャは、ファンクション複写定義またはテーブル複写定義に関連付けることができます。「複写ファンクションの配信」と「非同期プロシージャ配信」も参照してください。
- **複写テーブル** – 複数サイトのデータベースで、SAP Replication Server が一部または全部を管理するテーブルです。これらのテーブルのうち、`sp_setreptable` システムプロシージャを使用して複写するようにマーク付けされた 1 つのバージョンがプライマリバージョンで、それ以外のすべてのバージョンは複写コピーです。
- **Replication Agent** – プライマリデータへの修正を表すトランザクションログ情報を、他のデータベースに分配するために、データベースサーバから SAP Replication Server に転送するプログラムまたはモジュールです。RepAgent は、SAP ASE データベース用の Replication Agent です。
- **複写コマンド言語 (RCL)** – SAP Replication Server の情報を管理するために使用するコマンドです。
- **複写定義** – サブスクリプションを作成するためのテーブルの定義です。複写定義は SAP Replication Server によって管理され、この中で複写されるカラムとテーブルのプライマリバージョンがあるロケーションが指定されています。  
ファンクションの複写定義も作成できます。複写定義がテーブルに関するものかファンクションに関するものかを区別するために、「テーブル複写定義」という用語を使用することもあります。「ファンクション複写定義」も参照してください。
- **Replication Management Agent (RMA)** – サポートされている任意のデータベースから SAP HANA データベースへの複写を容易に設定および管理するために使用できる分散管理エージェントです。
- **Replication Server インタフェース (RSI)** – 送信先 SAP Replication Server にログインし、送信元 SAP Replication Server の RSI アウトバウンドステابلキュー

から送信先 SAP Replication Server へコマンドを転送するスレッドです。プライマリまたは中間 SAP Replication Server からコマンドを受け取る送信先 SAP Replication Server ごとに、1つの RSI スレッドが存在します。「アウトバウンドキュー」と「ルート」も参照してください。

- **複写システム管理者** – Replication Server の定型作業を管理するシステム管理者です。
- **Replication Server システムデータベース (RSSD)** – SAP Replication Server のシステムテーブルを格納する SAP ASE データベースです。ユーザは、SAP Replication Server システムテーブルを SAP ASE に格納するか、SAP Replication Server によりホストされている SAP SQL Anywhere データベースに埋め込むかを選択できます。「*Embedded Replication Server* システムデータベース (ERSSD)」も参照してください。
- **Replication Server システム Adaptive Server** – SAP Replication Server のシステムテーブルを格納するデータベースを持つ SAP ASE です。
- **複写システム** – 複数のデータベースにデータを複写することで、リモートユーザがそれぞれのローカルデータにアクセスできるようにするデータ処理システムです。複写システムは SAP Replication Server を基にして構成され、Replication Agent やデータサーバのような他のコンポーネントも含まれています。
- **複写システムドメイン** – 同じ ID サーバを使用する複写システムのすべてのコンポーネントです。
- **要求ファンクション** – ファンクション複写定義に対応する複写ファンクションであり、SAP Replication Server によってプライマリデータベースからレプリケートデータベースに配信されます。要求ファンクションがストアードプロシージャにパラメータ値を渡し、そのストアードプロシージャがレプリケートデータベースで実行されます。ストアードプロシージャはレプリケートサイトでプライマリサイトと同じユーザによって実行されます。「*複写ファンクションの配信*」、「*要求ファンクション*」、「*ファンクション複写定義*」も参照してください。
- **再同期マーカ** – Replication Agent を再同期モードで再開すると、Replication Agent は、再同期処理が進行中であることを示すデータベース再同期マーカを SAP Replication Server に送信します。Replication Agent は最初のメッセージとして再同期マーカを送信してから、SQL データ定義言語 (DDL: data definition language) またはデータ操作言語 (DML: data manipulation language) のトランザクションを送信します。
- **ルート** – 送信元 Replication Server から送信先 Replication Server への一方向のメッセージストリームです。ルートは、データ修正コマンド (RSSD に対するものを含む) と複写ファンクションまたはストアードプロシージャを Replication Server 間でやりとりします。「*直接ルート*」と「*間接ルート*」も参照してください。

- **ルートバージョン** – ルートの送信元と送信先の SAP Replication Server のサイトバージョン番号のうち、低い方の番号です。サポートされている SAP Replication Server のバージョンでは、レプリケートサイトに送信するデータを決定するのにルートバージョン番号を使用します。「**サイトバージョン**」も参照してください。
- **ローマイグレーション** – テーブルのプライマリバージョン内のローでカラム値が変更されたとき、テーブルのレプリケートバージョン内の対応するローも、サブスクリプションの **where** 句内の値の比較に基づいて挿入または削除されるプロセスです。
- **SAP Adaptive Server Enterprise (SAP ASE)** – SAP バージョン 11.5 およびそれ以降のリレーショナルデータベースサーバです。SAP Replication Server の設定中に RSSD オプションを選択すると、SAP ASE は RSSD データベースの SAP Replication Server システムテーブルを管理します。
- **SAP Replication Server** – SAP のサーバプログラムです。通常、LAN 上で複製データを管理し、同じ LAN または WAN 上にある別の SAP Replication Server から受け取ったデータのトランザクションを処理します。
- **スキーマ** – データベースの構造体です。DDL コマンドとシステムプロシージャは、データベースに格納されているシステムテーブルを変更します。SAP Replication Server バージョン 11.5 以降と SAP ASE バージョン 11.5 以降を使用している場合には、サポートされている DDL コマンドとシステムプロシージャは、スタンバイデータベースに複製できます。
- **サーチャブルカラム** – 複製するローをサイトで制限するために、サブスクリプションまたはアートの **where** 句で指定できるレプリケートテーブル内のカラムです。
- **サーチャブルパラメータ** – サブスクリプションの **where** 句で指定できる複製ストアドプロシージャのパラメータです。このパラメータを使用して、ストアドプロシージャを複製するかどうかを決定します。「**パラメータ**」も参照してください。
- **セカンダリトランケーションポイント** – プライマリデータを保存している SAP ASE データベースには、トランザクションログ内で SAP ASE がどこまで処理を完了したかを示すアクティブなトランケーションポイントがあります。これをプライマリトランケーションポイントといいます。
- **サイト** – 最低でも SAP Replication Server、データサーバ、データベースで構成され、場合によっては Replication Agent も含まれるインストレーション環境で、通常は地理的に離れた場所にあります。各サイトのコンポーネントは、WAN を介して複製システムにある他のサイトのコンポーネントに接続されます。「**プライマリサイト**」も参照してください。
- **サイトバージョン** – 個々の SAP Replication Server のバージョン番号です。サイトバージョンが一度あるレベルに設定されると、SAP Replication Server でそのレベル特有の機能が有効になり、レベルをダウングレードすることはできません。

ん。「ソフトウェアバージョン」、「ルートバージョン」、「システムバージョン」も参照してください。

- **ソフトウェアバージョン** – 個々の SAP Replication Server のソフトウェアリリースのバージョン番号です。「サイトバージョン」と「システムバージョン」も参照してください。
- **SQL Server** – 11.5 より前の SAP リレーショナルデータベースサーバです。
- **SQL 文の複写** – このプロセスでは、SAP Replication Server は、個々のローの変更ではなく、プライマリデータを変更した SQL 文をトランザクションログから受け取ります。SAP Replication Server は、SQL 文をレプリケートサイトに適用します。RepAgent は、SQL データ操作言語 (DML) と個々のローの変更の両方を送信します。設定に応じて、SAP Replication Server が、個々のローの変更によるログの複写または SQL 文の複写のどちらかを選択します。
- **ステابلキューマネージャ (SQM)** – ステابلキューを管理するスレッドです。インバウンドキュー、アウトバウンドキューのいずれの場合でも、SAP Replication Server がアクセスするステابلキューに対して、それぞれ 1 つのステابلキューマネージャ (SQM) スレッドがあります。
- **ステابلキュートランザクション (SQT) インタフェース** – コミット順にトランザクションコマンドを再構築するスレッドです。ステابلキュートランザクション (SQT) インタフェーススレッドは、インバウンドステابلキューを読み取って、トランザクションをコミット順に配列し、それらをディストリビュータ (DIST) スレッドと DSI スレッドのうち、SQT によるトランザクションの並び替えを要求した方に送信します。
- **ステابلキュー** – SAP Replication Server が、ルートまたはデータベース接続用のメッセージを格納するための蓄積転送キューです。ステابلキューに書き込まれたメッセージは、送信先の SAP Replication Server またはデータベースに配信されるまで、このキューに格納されます。SAP Replication Server は、割り当てられたディスクパーティションを使用してステابلキューを構築します。「インバウンドキュー」、「アウトバウンドキュー」、「マテリアライゼーションキュー」も参照してください。
- **スタンドアロンモード** – リカバリ処理を開始するために使用する SAP Replication Server のモードです。
- **スタンバイデータベース** – ウォームスタンバイアプリケーションでは、アクティブデータベースからデータ変更を受信し、そのデータベースのバックアップとして機能するデータベースのことです。「ウォームスタンバイアプリケーション」も参照してください。
- **ストアドプロシージャ** – SAP ASE データベースに名前付きで格納されている SQL 文とオプションのフロー制御文の集まりです。SAP ASE が提供するストアドプロシージャは、システムプロシージャと呼ばれます。SAP Replication

Server ソフトウェアには、RSSD に問い合わせるストアブプロシージャがいくつか組み込まれています。

- **サブスクリプション** – 指定したサイトのレプリケートデータベースにあるテーブルの複製コピー、またはテーブルからのローのセットを管理するために、SAP Replication Server に対して行う要求のことです。ストアブプロシージャを複製するために、ファンクション複製定義をサブスクライブすることもできます。
- **サブスクリプションマテリアライゼーション解除** – サブスクリプションが削除されたときに、適宜実行される処理です。これによって、他のサブスクリプションに使用されていない特定のローが、レプリケートデータベースから削除されます。
- **サブスクリプションマテリアライゼーション** – プライマリデータベースからレプリケートデータベースへ、サブスクリプションによって指定されたデータをコピーする処理です。これによって、レプリケートテーブルが初期化されます。レプリケートデータはネットワークを介して転送するか、またはサブスクリプションが大量のデータを扱う場合は、最初にメディアからロードできます。
- **サブスクリプションマイグレーション** – テーブルのプライマリバージョン内のローでカラム値が変更されたとき、テーブルのレプリケートバージョン内の対応するローも、サブスクリプションの `where` 句内の値の比較に基づいて挿入または削除されるプロセスです。
- **SAP Control Center for Replication** – 複製環境内のサーバのステータスと可用性をモニタリングするための Web ベースのソリューションです。
- **対称型マルチプロセッシング (SMP)** – マルチプロセッサプラットフォームで、アプリケーションのスレッドを並列に実行できる機能です。SAP Replication Server は、サーバのパフォーマンスと効率が高められる SMP をサポートしています。
- **同期コマンド** – クライアントが送信するコマンドです。クライアントは、完了ステータスを受信してから、他のオペレーションを継続できます。
- **システムファンクション** – あらかじめ定義され、SAP Replication Server 製品に組み込まれているファンクションです。 `rs_begin` などの複製アクティビティを調整するシステムファンクション、または `rs_insert`、`rs_delete`、`rs_update` などのデータ操作のオペレーションを実行するシステムファンクションがあります。
- **システム提供クラス** – SAP Replication Server が提供するエラークラス `rs_sqlserver_error_class` とファンクション文字列クラス `rs_sqlserver_function_class`、`rs_default_function_class`、`rs_db2_function_class` のことです。ファンクション文字列は、システムで提供されるファンクション文字列クラスとこれらのクラスから直接的または

間接的に継承する派生クラス用に自動的に生成されます。「エラークラス」と「ファンクション文字列クラス」も参照してください。

- **システムバージョン** – リリース 11.0.2 以前の SAP Replication Server に対して、新しい機能が有効なバージョンを表す複写システムのバージョン番号です。このバージョン番号より低いバージョンには、SAP Replication Server をダウンロードまたはインストールできません。SAP Replication Server バージョン 11.5 では、特定の機能をj使用するために、サイトバージョン 1150 と最低でもシステムバージョン 1102 が必要です。「混合バージョンシステム」、「サイトバージョン」、「ソフトウェアバージョン」も参照してください。
- **テーブル複写定義** – プライマリテーブルを特定し、挿入、更新、または削除時に SAP Replication Server がそのコンテンツを複写できるようにマーク付けします。SAP Replication Server によって使用されるパブリッシュ/サブスクライブ方法でデータが「パブリッシュ」されます。
- **テーブルサブスクリプション** – テーブル複写定義に対応するサブスクリプションです。
- **スレッド** – SAP Replication Server 内で実行されるプロセスです。SAP Open Server で構築された SAP Replication Server は、マルチスレッドアーキテクチャに基づいています。各スレッドは、ユーザセッションを管理したり、Replication Agent または別の SAP Replication Server からメッセージを受信したり、メッセージをデータベースに適用したりする特定のファンクションを実行します。「データサーバインタフェース (DSI)」、「ディストリビュータ」、「Replication Server インタフェース (RSI)」も参照してください。
- **トランザクション** – 文をグループ化するためのメカニズムです。このメカニズムによって、文はグループ内の単なる構成単位として扱われ、グループ内のすべての文が実行されるか、グループ内の文がまったく実行されないこととなります。
- **Transact-SQL** – SAP ASE で使用するリレーショナルデータベース言語です。Sybase 拡張機能付きの標準 Structured Query Language (SQL) に基づいています。
- **トランケーションポイント** – プライマリデータを保存している SAP ASE データベースには、トランザクションログ内で SAP ASE がどこまで処理を完了したかを示すアクティブなトランケーションポイントがあります。これをプライマリトランケーションポイントといいます。
- **ユーザ定義ファンクション** – このファンクションを使用すると、SAP Replication Server を使用して、複写システムのサイト間で複写ファンクションまたは非同期ストアドプロシージャを配信するカスタムアプリケーションを作成できます。複写ファンクションの配信では、ファンクション複写定義を作成すると、SAP Replication Server によって自動的にユーザ定義ファンクションが作成されます。
- **変数** – 「ファンクション文字列変数」を参照してください。

- **バージョン – 混合バージョンシステム**  
「混合バージョンシステム」、「サイトバージョン」、「ソフトウェアバージョン」、「システムバージョン」を参照してください。
- **ウォームスタンバイアプリケーション – SAP Replication Server** を使用して、アクティブデータベースと呼ばれるデータベースに対するスタンバイデータベースを管理するアプリケーションです。アクティブデータベースで障害が発生した場合、SAP Replication Server とクライアントアプリケーションはデータベースをスタンバイデータベースに切り替えられます。
- **広域ネットワーク (WAN)** – データ通信回線で接続されているローカルエリアネットワーク (LAN) のシステムです。
- **ワイドカラム** – char、varchar、binary、varbinary、unichar、univarchar、または Java inrow データで構成されている、255 バイトより大きい複写定義のカラムです。
- **ワイドデータ** – データサーバのデータページのサイズを上限とする、幅の広いデータローです。SAP ASE は、2K、4K、8K、16K のページサイズをサポートしています。
- **ワイドメッセージ** – 複数のブロックにまたがる 16K より大きいメッセージです。





## 索引

## A

- Adaptive Server Enterprise
  - binary データ型 102
  - char データ型 62
  - datetime データ型 62
  - numeric データ型 121
  - Replication Agent 9
  - varbinary データ型 102
  - varchar データ型 62
  - プライマリデータベース 24
  - レプリケートデータベース 23
- admin show connection、'レプリケート' 設定パラメータ 169
- admin who コマンド
  - 専用ルート用 216
- admin コマンド 260

## B

- bcp ユーティリティ 313

## C

- CLASSPATH システム変数 83
- connect source パーミッション 34, 46
- connection to SAP IQ
  - カスタマイズ 157
  - 作成 157

## D

- datatypes
  - rs\_db2\_varchar\_for\_bit、HDS 102
  - varbinary、Sybase 102
- DB2 for UNIX and Windows
  - BLOB データ型 109
  - CLOB データ型 109
  - LVARCHAR データ型 109
  - Replication Agent 用 49, 65
  - クラスレベル変換 292
  - クラスレベル変換スクリプト 111

- プライマリデータベースとして 65
- プライマリデータベースのパーミッション 66
- プライマリデータベースの制限事項 66
- プライマリデータベースの接続 66
- プライマリデータ型の変換 71
- レプリケートデータベースの接続ビティ 109
- レプリケートデータベースのパーミッション 108
- レプリケートデータベースの設定 109

## DB2 for z/OS

- クラスレベル変換 292
- プライマリデータベースのパーミッション 58
- プライマリデータベースの設定 60

## DB2 fUDB or z/OS

- DATE データ型 62

## DB2 UDB

- ヒープ 69

## DB2 UDB for UNIX and Windows

- デフォルトのデータ型変換 287

## DB2 UDB for UNIX および DB2 for Windows

- クラスレベル変換 287

## DB2 UDB for z/OS

- BLOB データ型 102
- CHAR データ型 62
- CLOB データ型 102
- DBCLOB データ型 102
- LTM for z/OS 59
- LTMADMIN ユーザ 58
- LTMLASTCOMMIT テーブル 58
- LTMOBJECTS テーブル 58
- Replication Agent 49
- rs\_info テーブル 100
- rs\_lastcommit テーブル 100
- SAP Log Extract 60
- クラスレベル変換 104, 287
- データ共有環境 58, 60
- デフォルトのデータ型変換 287
- トランザクションログ 57, 59

## 索引

プライマリデータベースとして 57  
プライマリデータベースの設定 60  
プライマリデータ型の変換 62  
レプリケートデータベース 99  
レプリケートデータベースの設定 102  
DB2 UDB for z/OS クラスレベル変換 104  
DSI スレッド  
    スタンバイデータベース 261  
DSI のモニタリング、データベースの再同期  
    273  
dsi\_cdb\_max\_size 設定パラメータ 164  
dsi\_compile\_enable、RTL 159  
dsi\_compile\_max\_cmds 設定パラメータ 164  
dsi\_compile\_retry\_threshold 設定パラメータ  
    162  
dsi\_incremental\_parsing 設定パラメータ 166

## E

ECDA データベースゲートウェイ 37, 52  
    DB2 メタデータ 65  
    DirectConnect for z/OS Option 99  
    ECDA Option for ODBC 119, 121  
    ECDA Option for Oracle 133  
    interfaces ファイル 51  
    Mainframe Connect DirectConnect for z/OS  
        Option 99  
    Microsoft SQL Server メタデータ 74  
    Oracle メタデータ 82  
    トラブルシューティング 330  
error classes  
    HDS 機能 42  
ExpressConnect for SAP HANA データベース  
    SAP HANA レプリケートデータベース用  
        195  
    設定 201

## I

ID サーバ  
    ログイン名 10  
    要件 10  
interfaces ファイル 47, 51, 101, 122, 154, 212  
interfaces ファイル、SAP IQ への複写用に作成  
    175

## J

Java Runtime Environment (JRE) 81

Java ストアドプロシージャ 68  
JDBC 通信プロトコル 83  
    ドライバ 85

## L

LOB データ型  
    制限事項 79  
    部分更新 80  
LTM for z/OS 60  
LTM ロケータ 35  
    次も参照：オリジンキュー ID  
LTMLASTCOMMIT テーブル、DB2 for z/OS デ  
    ータベース内 58  
LTMOBJECTS テーブル、DB2 for z/OS データ  
    ベース内 58

## M

mat\_load\_tran\_size、設定 322  
max\_mat\_load\_threads、設定 322  
Microsoft SQL Server クラスレベル変換 125  
Microsoft SQL Server データサーバ  
    decimal データ型 334  
    identity カラム 335  
    image データ型 119  
    ntext データ型 119  
    numeric データ型 335  
    Replication Agent 49  
    rs\_info テーブル 119  
    rs\_lastcommit テーブル 120  
    text データ型 119  
    クラスレベル変換 292  
    デフォルトのデータ型変換 292  
    数値精度 333  
Multi-Path Replication 209, 212  
    Adaptive Server から Oracle 222  
    Adaptive Server から SAP HANA データベ  
        ース 217  
    ASE から IQ へ 226  
    Oracle から Adaptive Server 230  
    Oracle から IQ 244  
    Oracle から Oracle 239  
    Oracle から SAP HANA データベース 234  
    SAP IQ 167

分散モデルの設定 171  
レプリケートとしての SAPIQ 167  
代替コネクション、概念 211  
並列化 211

## N

none  
トランザクション逐次化メソッド 116,  
130

## O

Oracle アプリケーション用ウォームスタンバイ  
249  
Oracle データサーバ  
JDBC ドライバ 83, 85  
Replication Agent 49  
Replication Agent 設定パラメータ 83  
rs\_info テーブル 133  
rs\_lastcommit テーブル 133, 152  
TNS Listener プロセス 83  
クラスレベル変換 296  
デフォルトのデータ型変換 296  
プライマリデータベースのパーミッション  
83  
プライマリデータ型の変換 86  
レプリケートデータベースの設定 136  
Oracle データベースの再同期 267  
resuming connection コマンドと skip to  
resync パラメータ 268  
skip to resync パラメータ 268  
シナリオ 277  
シナリオ、プライマリデータベース 278  
ダンプデータベースマーカの送信 272  
データベースのダンプの取得 270  
データベースのダンプの適用 273  
トランザクションのスキップ 268, 269  
レプリケートデータベースの再初期化  
273  
再同期マーカ、送信 269  
指示 267  
製品互換性 267  
設定 267  
Oracle、レプリケートデータベースの再同期  
267

## P

pdb\_xlog\_prefix 設定パラメータ 84

## Q

QID (オリジンキュー ID) 37

## R

ra\_set\_autocorrection コマンド 338  
RCL コマンド  
admin logical\_status コマンド 260  
Real-Time Loading  
データベースのサポート 146  
プラットフォームのサポート 146  
Replication Agent  
connect source パーミッション 34, 46  
DB2 for UNIX and Windows 用 65  
LTL バッチモード 36  
LTM ロケータ 35  
Microsoft SQL Server 用 73  
Replication Server 67, 75, 83  
Replication Server コネクション 46  
RSSD の使用 61, 65, 71, 73, 78, 81  
RSSD パラメータ 61, 71, 78, 81  
TCP/IP 通信 67, 75, 83  
オリジンキュー ID 37  
トランザクションログ 75  
トランザクションログのプレフィクス 84  
プライマリ Replication Server コネクション  
67, 75, 83  
メンテナンスユーザトランザクションの  
フィルタリング 60, 69, 85  
要件 12  
Replication Agent for DB2 UDB 8  
Replication Agent for DB2 UDB for z/OS 8, 49,  
57  
Date\_in\_char パラメータ 62  
DB2 の設定に関する問題 60  
interfaces ファイル 47  
LTL 問題 339  
LTM for z/OS 59  
LTM\_process\_maint\_uid\_trans パラメータ  
60

- LTMADMIN ユーザ 58
- RS\_source\_db パラメータ 60
- RS\_source\_ds パラメータ 60
- SAP Log Extract 60
- Use\_repdef パラメータ 61
- Replication Agent for Microsoft SQL Server
  - トランザクションログ 75
- Replication Agent ユーザスレッド 38, 46
- Replication Server
  - connect source パーミッション 34, 46
  - create connection コマンド 34, 52, 60, 69, 85
  - DSI スレッド 52
  - HDS 機能 336
  - interfaces ファイル 47, 51, 101, 122, 154
  - LTM ロケータ 35
  - Replication Agent コネクション 46
  - Replication Agent ユーザスレッド 38, 46
  - resume connection コマンド 53
  - rrs\_mss\_numeric データ型 335
  - rs\_db2\_char\_for\_bit データ型 102
  - rs\_db2\_varchar\_for\_bit データ型 102
  - rs\_dump コマンド 21
  - rs\_dumptran コマンド 22
  - rs\_get\_lastcommit ファンクション 100
  - rs\_marker コマンド 21
  - rs\_msss\_numeric データ型 121
  - rs\_subcmp ユーティリティ 323
  - RSI ユーザ 35
  - RSSD 30, 35
  - SAP HANA データベース機能 203
  - SAP IQ レプリケートデータベース 154
  - SysAdmin ユーザ 34
  - TCP/IP 通信 67, 75, 83
  - アウトバウンドキュー 329
  - インバウンドキュー 327
  - クライアントとしての動作 33
  - サーバとしての動作 33
  - サブスクリプションのマテリアライズ 20
  - データベースコネクション 37
  - メンテナンスユーザ 34, 60, 69, 85
  - ユーザ ID 34
  - 異機種間複写の問題 33
  - 説明 9
  - 通信プロトコル 33
  - Replication Server Data Assurance オプション
    - 概要 323
  - Replication Server システムデータベース (RSSD)
    - システムテーブル 11
    - 説明 11
    - 要件 12
  - Replication Server と SAP IQ InfoPrimer のデータフロー 183, 184
  - resume connection、skip to resync マーカ 268
  - rs\_dump コマンド 21
  - rs\_dumptran コマンド 22
  - rs\_info テーブル
    - DB2 UDB for z/OS データベース 100
    - Microsoft SQL Server データベース 119
    - Oracle データベース内 133
    - SAP HANA データベースインスタンス内 195
  - rs\_lastcommit テーブル
    - DB2 UDB for z/OS データベース 100
    - Microsoft SQL Server データベース 120
    - Oracle データベース内 133, 152
    - SAP HANA データベースインスタンス内 195
    - 問題 336
  - rs\_marker コマンド 21
  - rs\_session\_setting ファンクション文字列 157
  - rs\_status システムテーブル 191
  - rs\_subcmp ユーティリティ
  - RSSD 30
    - Replication Agent、 57
    - Replication Agent、使用 61, 65, 71, 73, 78, 81
    - Replication Server のユーザ ID 35
  - RTL
    - admin config コマンド 173
    - dsi\_bulk\_threshold 160
    - dsi\_cdb\_max\_size 160
    - dsi\_command\_convert 161
    - dsi\_compile\_enable 159
    - dsi\_compile\_max\_cmds 160
    - dsi\_compile\_retry\_threshold 161
    - rs\_helprep ストアドプロシージャ 174
    - SAP IQ への複写の有効化 158
    - インクリメンタル解析 166

- コンパイルできないコマンド、テーブル 150
  - コンパイルとバルク適用 147
  - コンパイルの例 147
  - コンパイルルール 147
  - システムテーブルサポート 174
  - ステージングソリューションからのマイグレート 180
  - ステージングソリューションからのマイグレートの準備 181
  - 設定パラメータ 159
  - データベースとプラットフォームのサポート 146
  - データベースレベルの設定パラメータの表示 173
  - テーブルレベルの設定パラメータの表示 173
  - テーブル参照の表示 174
  - フルインクリメンタルコンパイル 160, 164
  - 下位互換性 174
  - 混合バージョンのサポート 174
  - 最終的な変更のデータベースの表示 149
  - 参照制約 150, 171
  - 情報の表示 173
  - 制限事項 149
  - 複写シナリオ 175
  - 利点 145
  - RTL での `dsi_bulk_threshold` 160
  - RTL での `dsi_cdb_max_size` 160
  - RTL での `dsi_command_convert` 161
  - RTL での `dsi_compile_max_cmds` 160
  - RTL での `dsi_compile_retry_threshold` 161
  - RTL での参照制約 171
  - RTL の最終的な変更のデータベース、表示 149
  - RTL 複写のシナリオ 175
  - RTL 複写の例 175
  - RTL、リトライメカニズムの強化 162
- S**
- SAP HANA データベースデータサーバ
    - `rs_info` テーブル 195
    - `rs_lastcommit` テーブル 195
    - レプリケートデータベース設定 203
  - SAPIQ
    - RTL のコンパイルとバルク適用 147
    - RTL の設定パラメータ 159
    - RTL の有効化 158
    - エラークラスとファンクション文字列クラス 157
    - コネクションの作成 157
    - コネクションパラメータ、設定 157
    - ステージングソリューションからのマイグレート 180
    - レプリケートデータベースのコネクティビティ 154
    - レプリケートデータベースのパーミッション 155
    - レプリケートデータベースの設定 156
    - 干渉、システムテーブル 152
    - 干渉、テンポラリワークテーブル 153
    - 接続プロファイル 156
    - 複写の干渉と影響 152
  - SAP IQ の接続プロファイル 156
  - SAP IQ のファンクション文字列クラス 157
  - SAP IQ 複写ステージングソリューションから RTL へのマイグレート 180
  - SAP IQ への干渉、テンポラリワークテーブルから 153
  - SAP Log Extract 60
  - SAP Replication Server データ保証オプション 22
  - SAP Sybase IQ InfoPrimer Integration
    - Replication Server コンポーネント 191
  - SAP Sybase Replication Agent 8
  - `skip to resync` パラメータ 268
  - `skip to resync` マーカ、Replication Agent から Replication Server への送信 269
  - `sql.ini` ファイル 212
  - Sybase IQ InfoPrimer の統合
    - オートコレクション関数 192
  - Sybase Replication Agent
    - DB2 for UNIX and Windows の制限事項 66
    - DB2 UDB 用 65
    - `filter_maint_userid` パラメータ 69, 85
    - JDBC 通信 83
    - LTL バッチモード 36
    - LTL 問題 340
    - `ltl_character_case` パラメータ 70

## 索引

Microsoft SQL Server データベース用 73  
pdb\_convert\_datetime パラメータ 71  
use\_rssd パラメータ 71, 78, 81  
メタデータコマンド 65, 74, 82  
SysAdmin ユーザ、Replication Server 34

## T

TCP/IP 通信プロトコル 67, 75, 83  
TNS Listener プロセス、Oracle 83

## U

UNIX および Windows 向け DB2  
RS\_INFO テーブル 107  
RS\_LASTCOMMIT テーブル 107

## Z

z/OS オペレーティングシステム  
データ共有環境 58, 60

## あ

アウトバウンドキュー、トラブルシューティング 329  
アクティブデータベース 249  
アクティブデータベースの追加  
Replication Agent の初期化 254  
コネクションの作成 256  
アトミックバルクマテリアライゼーション

## い

インクリメンタル解析  
RTL 166  
インバウンドキュー、トラブルシューティング 327  
引用符付き識別子 70, 86, 208

## う

ウォームスタンバイアプリケーション  
スタンバイデータベースへの切り替えの  
影響 263  
データベース 249

メソッド 252  
制限 250

## え

エラークラス、SAP IQ 157  
エラーメッセージ  
データ型の範囲 332–334  
数値 ID 335

## お

オリジンキュー ID 35, 37  
LTM ロケータ 35

## か

解析、インクリメンタル 166  
カラム値、欠落 338  
カラム値の欠落 338  
干渉と影響、SAP IQ への複写 152

## き

キュー、Replication Server  
アウトバウンド 329  
インバウンド 327

## く

クラスレベル変換  
DB2 for UNIX and Windows データ型 287  
DB2 for UNIX and Windows のデータ型  
111  
DB2 UDB for z/OS データ型 287  
Microsoft SQL Server データ型 292  
Oracle データ型 296  
次も参照：異機種データ型サポート  
(HDS)

## け

ゲートウェイ  
次を参照：データベースゲートウェイ

## こ

コネクション設定パラメータの削除 169

コマンド

create connection 34, 52, 60, 69, 85

resume connection 53

rs\_dump 21

rs\_dumptran 22

rs\_marker 21

rs\_subcmp 323

コマンドおよび設定パラメータ

専用ルート用 213

コンパイルとバルク適用、RTL 147

## さ

作成

SAP IQ への接続 157

サブスクリプション 31

マテリアライズ 20

## し

システムテーブル

rs\_status 191

説明 11

シナリオ、データベース再同期化 277

## す

スタンバイデータベース 249

スタンバイデータベースの追加

Replication Agent の初期化 257

Replication Agents のレジューム 260

コネクションのレジューム 259

コネクションの作成 259

ストアドプロシージャ

複写 31

複写の制限事項 17

## せ

セキュリティ

Secure User Store 206

## ち

直接ロードマテリアライゼーション

テーブルサブスクリプションの作成 317

テーブルの削除サブスクリプション 317

直接ロードマテリアライゼーションの設定

mat\_load\_tran\_size 322

max\_mat\_load\_threads 322

## て

データ型定義 42

データベース

Replication Server コネクション 37

アクティブ 249

スタンバイ 249

データのレプリケートへのロード 308

プライマリからのデータのアンロード  
307

プライマリデータベース 5, 8

マテリアライゼーション 20

レプリケートデータベース 14

所有者指定のオブジェクト名 18

調整 323

論理 249

データベースオブジェクト

トランザクションログオブジェクト名 68,  
84

トランザクションログプレフィクス 84

データベースゲートウェイ 6, 13, 37, 52

DB2 for z/OS レプリケートデータベース  
99

Microsoft SQL Server レプリケートデータ  
ベース 119, 121

Oracle レプリケートデータベース 133

トラブルシューティング 330

データベースのサポート、Real-Time Loading  
146

データベースのダンプ、取得 270

データベースのダンプ、適用 273

データベースの再同期

DSI のモニタリング 273

データベースの再同期の設定 267

DSI スレッド情報のモニタリング 273

Replication Server に対するトランザクシ  
ョンのスキップの指示 268

Replication Server へのダンプデータベース  
マーカの送信 272

Replication Server へのデータベース再同期  
マーカの送信 269

## 索引

- データベースのダンプの取得 270
- レプリケートデータベースの再初期化 273
- 再同期するデータベースへのダンプの適用 273
- データベースの調整 323
- データベースの比較 323
- データベース再同期化シナリオ 277
  - サードパーティダンプユーティリティの使用による再同期 280
  - プライマリデータベースからのレプリケートデータベースの直接的な再同期 278
  - 同じダンプからのプライマリデータベースとレプリケートデータベースの再同期 282
- データ共有環境、DB2 UDB for z/OS 58, 60
- データ型
  - binary、Sybase 102
  - BLOB、DB2 102, 109
  - CHAR、DB2 62
  - char、SAP 62
  - char、Sybase 86
  - CLOB、DB2 102, 109
  - DATE、DB2 71
  - DATE、Oracle 86
  - datetime、SAP 62
  - datetime、Sybase 71
  - DB2 のクラスレベル変換 287
  - DB2 のデフォルト変換 287
  - DBCLOB、DB2 102
  - decimal、Microsoft SQL Server 334
  - image、Microsoft SQL Server 119
  - LVARCHAR、DB2 109
  - Microsoft SQL Server のクラスレベル変換 292
  - Microsoft SQL Server のデフォルト変換 292
  - ntext、Microsoft SQL Server 119
  - numeric、Microsoft SQL Server 335
  - numeric、SAP 121
  - Oracle のクラスレベル変換 296
  - Oracle のデフォルト変換 296
  - Replication Agent による変換 71
  - rs\_db2\_char\_for\_bit、HDS 102
  - rs\_mss\_numeric、HDS 335
  - rs\_msss\_numeric、HDS 121
  - text、Microsoft SQL Server 119
  - TIME、DB2 71
  - TIMESTAMP、DB2 71
  - varchar、SAP 62
  - varchar、Sybase 86
  - デフォルトの HDS 変換 285
  - マテリアライゼーション 307
  - ラージオブジェクト (LOB) 18, 42, 102, 119
  - 範囲 332, 334
  - 変換のトラブルシューティング 336
- データ型変換 121
  - クラスレベル 104
  - マテリアライゼーション 307
- テーブル
  - rs\_info、DB2 UDB for z/OS データベース 100
  - rs\_info、Microsoft SQL Server データベース 119
  - rs\_info、Oracle データベース内 133
  - rs\_info、SAP HANA データベースインスタンス内 195
  - rs\_lastcommit の問題 336
  - rs\_lastcommit、DB2 UDB for z/OS データベース 100
  - rs\_lastcommit、Microsoft SQL Server データベース 120
  - rs\_lastcommit、Oracle データベース内 133, 152
  - rs\_lastcommit、SAP HANA データベースインスタンス内 195
- デバッグ 202
- と
  - ドライバ、JDBC 83
    - Oracle に必要 85
  - トラブルシューティング
    - アウトバウンドキュー 329
    - インバウンドキュー 327
    - レプリケートデータベース 330
  - トランケーション
    - プロシージャ 68



- トランザクション
  - 逐次化メソッド 114, 128, 143
- トランザクションログ
  - DB2 UDB for z/OS 57, 59
  - Replication Agent for Microsoft SQL Server 75
  - オブジェクト名 68, 84
  - プレフィクス 84
  - 隠しテーブル 76
- トリガ
  - 制御 141
- トレース 202
- トレースポイント 202
  
- の
- ノンアトミックバルクマテリアライゼーション
  
- は
- パーミッション
  - DB2 for UNIX and Windows プライマリデータベース 66
  - DB2 fUDB for z/OS プライマリデータベース 58
  - Oracle プライマリデータベース 83
- パーミッション、レプリケート SAP IQ 155
  
- ひ
- ヒープ
  - DB2 UDB 69
  
- ふ
- ファクション文字列クラス
  - HDS 機能 42
  - rs\_translation システムテーブル、更新 42
  - クラスレベル変換、継承 42
- ファンクション文字列
  - LOB 複写に関する変更 19
  - rs\_dump コマンド 21
  - rs\_dumptran コマンド 22
  - rs\_marker コマンド 21
- ファンクション文字列クラス
  - HDS 機能 42
- LOB 複写に関する変更 19
- 複数の複写パス
  - SAP IQ 167
- プライマリデータベース 5, 8
  - DB2 for UNIX and Windows 65
  - DB2 UDB for z/OS 57
  - オリジンキュー ID 37
  - データのアンロード 307
  - 異機種間複写の問題 15
- プラットフォームのサポート、Real-Time Loading 146
- フルインクリメンタルコンパイル
  - dsi\_cdb\_max\_size、効果 164
  - RTL 164
- フルインクリメンタルコンパイル、RTL 内 160
- プレフィクス、トランザクションログ 84
- プロファイル
  - 接続 103, 110, 124, 137, 156, 204, 331
  
- ま
- マーカ隠しテーブル 76
- マテリアライゼーション
  - アトミックバルク 308
  - データ型変換 307
  - ノンアトミックバルク 311
  - プライマリデータベースからのデータのアンロード 307
  - レプリケートデータベースへのデータのロード 308
  
- め
- メモリ消費の制御
  - RTL 163, 165
- メモリ消費パラメータの操作 165
- メンテナンスユーザ
  - 権限の付与 155
- メンテナンスユーザ、Replication Server
  - Replication Agent フィルタリングトランザクション 47
  - トランザクション 60, 69, 85
  - ユーザ ID 39

## 索引

### ゆ

#### ユーザ ID

- LTMADMIN ユーザ、DB2 for z/OS 58
- Replication Server 34
- SysAdmin ユーザ 34

#### ユーティリティ

- bcp 313
- rs\_subcmp 323

### ら

#### ラージオブジェクト (LOB) データ型

- DB2 for UNIX and Windows 109
- DB2 for z/OS データベース 102
- Microsoft SQL Server データベース 19, 119
- 複写の制限事項 18
- 変換の制限事項 42

### り

#### リトライメカニズム、RTL での強化 162

#### リファレンス実装

- 開始する前に 344
- 設定ファイル 345
- プラットフォームサポート 343
- 概要 343

### れ

#### レプリケート SAPIQ のコネクティビティ 154

#### レプリケート SAP IQ の設定 156

#### レプリケート SAP IQ のデータベースパーミッション 155

#### レプリケートコネクション

- 代替、作成 167
- 代替、表示 169
- 代替、変更 169

#### レプリケートデータベース 5, 14

- DB2 UDB for z/OS 99
- SAPIQ 152

#### 設定 119, 133, 195

- データのロード 308
- トラブルシューティング 330
- 異機種間複写の問題 16

#### レプリケートデータベース、再初期化 273

#### レプリケートデータベースの再初期化 273

### ろ

#### ログイン名

- ID サーバ 10

#### ログ転送言語 (LTL)

- 問題 339