Quest

SharePlex® 11.4

リファレンスガイド

© 2024 Quest Software Inc. ALL RIGHTS RESERVED.

本書に記載されている専有情報は、著作権によって保護されています。本書に記載されているソフトウェアは、ソフトウェアライセンスまたは機密保持契約のもとに提供されます。本ソフトウェアは、当該契約の条項に従う場合に限り、使用または複製できるものとします。本書のいかなる部分も、Quest Software Inc.の書面による許可なく、複写および録音を含む電子的または機械的ないかなる形式や手段においても、あるいはいかなる目的においても、複製または転載することはできません。

本書に記載されている情報は、Quest Software製品の概要説明を目的としたものです。本書によって、あるいはQuest Software製品の販売に関連して、明示または黙示にかかわらず、禁反言やその他の方法によって生じる、いかなる知的所有権に対するライセンスも許諾されません。当該製品のライセンス契約で指定されている約款に記載されている場合を除き、Quest Softwareはいかなる責任も負うものではなく、商品性、特定目的への適合性、または非侵害性に関する黙示的保証を含め(ただしこれらに限定されない)、その製品に関連する一切の明示的、黙示的、または法令による保証を行いません。Quest Softwareは、いかなる場合においても、本書の使用または使用不可能に起因する直接損害、間接損害、結果的損害、懲罰的損害、特別損害、または付随的損害(営業利益の損失、ビジネスの中断、情報の紛失を含むがこれらに限定されない)について、仮にそれらの発生の可能性を知らされていたとしても、一切の責任を負いません。Quest Softwareは、本書の内容の正確性または完全性に関する保証または表明を行わず、仕様および製品の説明に対する変更をいつでも予告なく行う権利を有します。Quest Softwareは、本書に記載されている情報を更新する確約を一切行いません。

本書の使用に関して不明な点がありましたら、以下までお問い合わせください。

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

クエスト・ソフトウェア株式会社
160-0023
東京都新宿区西新宿6-10-1日土地西新宿ビル13F

Webサイト https://www.quest.com/jp-ja
お問い合わせ Sales.JP@quest.com

特許

Quest Softwareは、当社の先進的なテクノロジーを誇りにしています。この製品には、特許および出願中の特許が適用される場合があります。この製品に適用される特許の最新情報については、当社のWebサイト(https://www.quest.com/legal)をご覧ください。

商標

Quest、Questのロゴ、SharePlex、およびJoin the Innovationは、Quest Software Inc.の商標および登録商標です。Questの商標の一覧については、https://www.quest.com/legal/trademark-information.aspxをご覧ください。その他すべての商標および登録商標は各所有者に帰属します。

SharePlex リファレンスガイド
更新日 - 2024/09/10
バージョン - 11.4

内容

このガイドについて **28**
その他のSharePlex関連ドキュメント 28

このガイドで使用される表記規則 **29**

Oracle用SharePlexコマンド **30**
SharePlexコマンド一覧 30
設定コマンド 30
接続コマンド 31
暗号化コマンド 32
情報コマンド 32
メンテナンスコマンド 33
パラメーターコマンド 34
パーティション化コマンド 34
プロパティコマンド 34
ルートコマンド 35
実行コマンド 35
同期コマンド 36
ターゲット設定コマンド 36
その他コマンド 37
SharePlexコマンドのアルファベット順リファレンス 38
abort config 38
用途 38
構文 38
abort service 40
用途 40
構文 40
activate config 43
アクティベーションのガイドライン 43
用途 44
構文 45
add partition 48
用途 48
構文 48

analyze config 51
用途 52
構文 53
append status 55
用途 55
構文 55
認証レベル 57
用途 57
構文 57
cancel 59
用途 59
構文 59
clear history 61
用途 61
構文 61
clear status 63
用途 63
構文 63
compare / compare using 65
サポートされている操作 65
サポートなし 65
その他の条件 66
特別な使用例 67
compareの仕組み 68
SQLログファイルの管理 69
複数のプロセスの実行 71
設定のサブセットのcompare 71
compareする行の制御 72
プロセスの特定 72
sp_ctrlでステータスと結果を表示 72
警告とエラーの表示 73
compareジョブのキャンセル 73
compare履歴とログの管理 74
バッチサイズのコントロール 74
非同期行のrepair 74
compare status 85

用途 85
構文 85
connection 88
用途 88
構文 88
copy / append 93
copyコマンド 93
appendコマンド 93
ランチャープロセス 94
コピー/追加プロセスの概要 94
コピープロセスの同時実行 95
同期プロセスの識別 95
同期のチューニングパラメーター 95
コピーステータスの表示 96
同期ルーティングファイルの使用 97
異種ソーステーブルとターゲットテーブルのコピー 97
分割レプリケーションでのコピーの使用 97
処理スレッド数の制御 98
コマンドライン構文でのOracleワイルドカードの使用 98
copy/appendの使用に関するその他の考慮事項 98
用途 99
構文 99
copy cache 103
用途 103
構文 103
copy config 105
用途 105
構文 105
copy status 107
用途 107
構文 107
create config 109
用途 109
構文 109
create encryption key 111
用途 111

構文 111
deactivate config 113
用途 113
構文 113
delete capture queue 115
用途 115
構文 115
delete export process 117
用途 117
構文 117
delete export queue 119
用途 119
構文 119
delete post queue 121
用途 121
構文 121
disable jobs / enable jobs 123
用途 123
構文 123
drop partition 125
用途 125
構文 125
drop partition scheme 127
用途 127
構文 127
editコマンド 128
用途 128
構文 128
edit config 129
用途 129
構文 129
exitコマンド 131
用途 131
構文 131
expandコマンド 132
用途 132

構文 132
flushコマンド 134
Postの再開 134
用途 134
構文 134
helpコマンド 138
用途 138
構文 138
hostコマンド 140
用途 140
構文 140
hostinfoコマンド 142
用途 142
構文 142
job statusコマンド 144
用途 145
構文 145
list config 148
用途 148
構文 148
list param 150
SharePlexパラメーターモジュール 150
用途 151
構文 151
lstatusコマンド 153
SharePlexキューを理解する 153
用途 154
構文 154
modify partition 156
用途 156
構文 156
orainfoコマンド 158
用途 158
構文 158
portコマンド 160
用途 160

構文 160
purge config 162
用途 162
構文 162
qstatusコマンド 164
用途 164
構文 165
quitコマンド 166
用途 166
構文 166
reconcileコマンド 167
用途 167
構文 167
redoコマンド 170
用途 170
構文 170
remove config 171
用途 171
構文 171
remove log 173
用途 173
構文 173
rename config 175
用途 175
構文 175
repair / repair using 177
サポートされている操作 177
サポートなし 177
その他の条件 178
repairコマンドの使用方法 179
repairを実行するタイミング 179
特別な使用例 179
repairの仕組み 182
repairの条件 182
プロセス 183
エラー処理 183

SQLログファイルの管理 183
複数のプロセスの実行 185
設定のサブセットのrepair 185
どの行をrepairするかを制御 186
プロセスの特定 186
sp_ctrlでステータスと結果を表示 186
警告とエラーの表示 187
repairジョブのキャンセル 187
compare履歴とログの管理 187
バッチサイズのコントロール 188
用途 188
構文 188
repair status 198
用途 198
構文 198
report 200
用途 200
構文 200
reset encryption key 202
用途 202
構文 202
reset param 204
コマンドオプションの使用 204
現在のパラメーター設定の表示 204
用途 205
構文 205
set encryption key 208
用途 208
構文 208
set log 210
用途 210
構文 210
set param 212
コマンドオプションの使用 212
用途 213
グローバル設定の構文 213

プロセス固有設定の構文 214
showコマンド 218
Process(プロセス) 218
Source(ソース) 218
Target(ターゲット) 219
State(状態) 219
PID 219
用途 220
構文 220
show captureの使用 222
基本コマンド 222
詳細な統計情報 222
Oracle Capture用の詳細な統計情報 222
用途 224
構文 224
show config 226
用途 228
構文 228
show encryption key 230
用途 230
構文 230
show export 232
用途 232
構文 232
show import 235
用途 235
構文 235
show log 237
用途 237
構文 237
show post 241
基本的なshow postコマンド 241
詳細なshow postコマンド 241
用途 244
構文 244
show read 247

基本コマンド 247
詳細な統計情報 247
用途 249
構文 249
show SQL 251
用途 251
構文 251
show statusdb 254
用途 254
構文 254
show scn 256
用途 256
構文 256
show sync 258
用途 258
構文 258
シャットダウン 260
グレースフルシャットダウン 260
強制シャットダウン 260
用途 260
構文 260
start 262
用途 262
構文 262
startup 265
用途 265
構文 265
Status 267
用途 267
構文 267
Stop 269
Captureプロセスの停止 269
特定の時点でのPostの停止 269
用途 269
構文 270
Target(ターゲット) 274

用途 274
構文 274
CDCカテゴリ 277
characterssetカテゴリ 277
fileカテゴリ 278
formatカテゴリ 279
JSONレコードの形式 285
JMSカテゴリ 287
JSONカテゴリ 289
kafkaカテゴリ 294
metadataカテゴリ 298
resourceカテゴリ 301
rule | filterカテゴリ 302
sourceカテゴリ 304
SQLカテゴリ 305
trace capture 307
用途 307
構文 307
出力 308
trace post 310
用途 310
構文 310
出力 312
trace read 314
用途 314
構文 314
出力 315
truncate log 317
用途 317
構文 317
typemap 319
用途 319
構文 319
usage 321
用途 321
構文 321

verify config 322
用途 323
構文 323
version 325
用途 325
構文 325
view config 327
用途 327
構文 327
view logオプション 329
用途 329
構文 329
view partitions 331
用途 331
構文 331

SharePlexパラメータの設定 332
ユーザ設定可能なパラメーターの説明 332
設定アクティベーションパラメーター 332
SP_OCF_HASH_BY_BLOCK 333
SP_OCF_LOCK_WAIT_TIME 333
SP_OCF_THREAD_COUNT 333
Captureパラメーター 334
SP_OCT_ALLOW_DP_DDL 334
SP_OCT_ARCH_LOC 334
SP_OCT_ASM_CACHE_AGE 335
SP_OCT_ASM_ECACHE_SIZE 335
SP_OCT_ASM_FLIST_SIZE 335
SP_OCT_ASM_MULTI_OCI_BLOCK_SIZE 335
SP_OCT_ASM_MULTI_OCI 336
SP_OCT_ASM_SLIST_SIZE 336
SP_OCT_ASM_SUPPORT 336
SP_OCT_AUTOADD_ENABLE 336
SP_OCT_AUTOADD_MVIEW 337
SP_OCT_AUTOADD_SEQ 338
SP_OCT_CHECKPOINT_LOG 338

SP_OCT_CHECKPOINT_TIME 338
SP_OCT_CK_LOC_FIRST 339
SP_OCT_DATA_OBJ_CACHE_SIZE 339
SP_OCT_DATE_ MSG 339
SP_OCT_DDL_LOGGING 339
SP_OCT_DDL_UPDATE_CONFIG 340
SP_OCT_DEFパラメーター 340
SP_OCT_DENIED_USERID 340
SP_OCT_DLOAD_MAX_COLS_PER_MSG 341
SP_OCT_ENABLE_LOBMAP 341
SP_OCT_INCLUDE_UNCHANGED_COL 341
SP_OCT_INSERT_INCLUDE_NULLS 342
SP_OCT_LOB_BUFFER_SIZE 342
SP_OCT_LOG_FILESIZE 342
SP_OCT_LOG_MEMBER 342
SP_OCT_LOG_NUMFILES 343
SP_OCT_LOG_READ_SIZE 343
SP_OCT_LOGWRAP_RESTART 343
SP_OCT_MIN_SESSIONS 344
SP_OCT_OLOG_DELAY 344
SP_OCT_OLOG_NO_DATA_DELAY 344
SP_OCT_OLOG_QUEUE_SIZE 344
SP_OCT_OLOG_RDS_MINER 345
SP_OCT_OLOG_REOPEN 345
SP_OCT_LOG_USE_OCI 345
SP_OCT_ONELINE_DEBUG 346
SP_OCT_OPS_LOGREADER_DELAY 346
SP_OCT_OPS_LOGREADER_RETRY 346
SP_OCT_OPS_NO_DATA_DELAY 347
SP_OCT_PREFER_ARCHIVE 347
SP_OCT_READ_ARC 347
SP_OCT_REDOLOG_ENSURE 348
SP_OCT_REDUCED_KEY 348
SP_OCT_REPLICATE_ALL_DDL 349
SP_OCT_REPLICATE_COMMENT 350
SP_OCT_REPLICATE_CTAS 350

SP_OCT_REPLICATE_DDL 351
SP_OCT_REPLICATE_DLOAD 352
SP_OCT_REPLICATE_GRANT 352
SP_OCT_REPLICATE_MVIEW 352
SP_OCT_REPLICATE_POSTER 353
SP_OCT_REPLICATE_SEQUENCES 353
SP_OCT_REPLICATE_SYNONYM 353
SP_OCT_REPLICATE_TRIGGER 353
SP_OCT_REQUIRED_DATA_IS_LOGGED 354
SP_OCT_TRUNC_PARTITION_BY_ID 354
SP_OCT_USE_DST 355
SP_OCT_USE_SUPP_KEYS 355
Readパラメーター 356
SP_ORD_BATCH_ENABLE 356
SP_ORD_BATCH_MAX 356
SP_ORD_BATCH_MATCH_MIN 356
SP_ORD_CDA_LIMIT 356
SP_ORD_DATE_ MSG 357
SP_ORD_DELAY_RECORDS 357
SP_ORD_FIRST_FIND 357
SP_ORD_HP_HASH 358
SP_ORD_HP_IN_SYNC 358
SP_ORD_LDA_ARRAY_SIZE 359
SP_ORD_ONELINE_DEBUG 359
SP_ORD_MSGS_CK_FREQおよびSP_ORD_RCM_SKIP_RATIO 360
SP_ORD_RMSG_LIMIT 360
SP_ORD_ROLLBACK_TXNS_MAX 361
SP_ORD_SEND_DDL_TO_FIRST 361
Exportパラメーター 363
SP_XPT_AES_KEY_LENGTH 363
SP_XPT_ENABLE_AES 363
SP_XPT_ENABLE_COMPRESSION 363
SP_XPT_KEEPALIVE 364
SP_XPT_PORT_OVERRIDE 364
SP_XPT_SO_SNDBUF 364
SP_XPT_USE_LOCALHOST 365

Importパラメーター 366
SP_IMP_ENABLE_AES 366
SP_IMP_QUEUE_PAUSE 366
SP_IMP_QUEUE_RESUME 367
SP_IMP_WCMT_MSGCNT 367
SP_IMP_WCMT_TIMEOUT 367
Oracle Postパラメーター 368
SP_OPO_CHANGE_ID_START_VALUE 368
SP_OPO_COMMIT_REDUCE_MSGS 368
SP_OPO_CONNECTION_POOL 368
SP_OPO_CONT_ON_ERR 369
SP_OPO_DEPENDENCY_CHECK 370
SP_OPO_DEPENDENCY_SIZE 370
SP_OPO_DISABLE_OBJECT_NUM 370
SP_OPO_HINTS_LIMIT 371
SP_OPO_LOG_CONFLICT 371
SP_OPO_MAX_CDA 372
SP_OPO_MAX_OEXN_TIME 372
SP_OPO_MAX_ROLLBACK_MESSAGES 372
SP_OPO_NLS_CONVERSION 373
SP_OPO_NLS_DEFAULT_COMPAT 374
SP_OPO_OBJID_DEBUG 374
SP_OPO_ONELINE_DEBUG 374
SP_OPO_OUT_OF_SYNC_SUSPEND 375
SP_OPO_POSTER_DELAY 376
SP_OPO_PRB_MISMATCH_SUSPEND 376
SP_OPO_READRELEASE_INTERVAL 377
SP_OPO_REDUCED_KEY 377
SP_OPO_RETRIES_MAX 378
SP_OPO_SAVE_OOS_TRANSACTION 378
SP_OPO_SQL_CACHE_DISABLE 380
SP_OPO_STOP_ON_DDL_ERR 380
SP_OPO_SUPPRESS_DELETE 380
SP_OPO_SUPRESSED_OOSを 381
SP_OPO_SYNC_LOG_FREQUENCY 381
SP_OPO_THREADS_MAX 381

SP_OPO_TRACK_COMMITS 382
SP_OPO_TRACK_OPERATIONS 382
SP_OPO_TRACK_PREIMAGE 382
SP_OPO_TRUSTED_SOURCE 383
SP_OPO_UPDATE_SCN 383
SP_OPO_USE_VARNUM 383
SP_OPO_WAIT_MSG_DELAY 383
Open Target Postパラメーター 385
SP_OPX_BATCH_ENABLE 385
SP_OPX_COMMIT_REDUCE_MSGS 385
SP_OPX_CONT_ON_ERR 385
SP_OPX_DISABLE_OBJECT_NUM 386
SP_OPX_MAX_BATCH_MBYTES 386
SP_OPX_MSGS_IN_MEMORY 387
SP_OPX_NLS_CONVERSION 387
SP_OPX_ONELINE_DEBUG 388
SP_OPX_ONELINE_DEBUG_COLUMNS 389
SP_OPX_OUT_OF_SYNC_SUSPEND 389
SP_OPX_READRELEASE_INTERVAL 389
SP_OPX_REDUCED_KEY 390
SP_OPX_SQL_CACHE_DISABLE 390
SP_OPX_STOP_ON_DDL_ERR 391
SP_OPX_THREADS 391
SP_OPX_TRUSTED_SOURCE 391
Queueパラメーター 392
SP_QUE_MAX_QUEUES 392
SP_QUE_POST_SHMSIZE 392
SP_QUE_Q_SHMSIZE 392
SP_QUE_SHMDBUF 392
SP_QUE_SYNC 393
SP_QUE_USE_SUBQUE_INDEX 394
sp_copパラメーター 395
SP_COP_IDLETIME 395
SP_COP_SCAN_TIMEOUT 395
SP_COP_SO_RCVBUF 395
SP_COP_TPORT 396

SP_COP_UPORT 396
Logパラメーター 397
SP_SLG_LOG_MAXSIZE 397
SNMPパラメーター 398
SP_SLG_SNMP_ACTIVE 398
SNMPエージェントパラメーター 398
カスタムMIBパラメーター 398
SNMPトラップパラメーター 399
Systemパラメーター 400
SP_SYS_HOST_NAME 400
SP_SYS_IN_SYNC 401
SP_SYS_JOB_HISTORY_RETENTION 401
SP_SYS_JOB_HISTORY_SIZE 401
SP_SYS_SUSPEND_ON_ERROR 402
SP_SYS_TARGET_COMPATIBILITY 402
SP_SYS_VAR_FULL 403
SP_SYS_VAR_OK 403
Compare/Repairパラメーター 404
SP_DEQ_IGNORE_ORACLE_ERROR_NUM 404
SP_DEQ_IP_MAPPING 404
SP_DEQ_LOG_FILESIZE 405
SP_DEQ_MALLOC 405
SP_DEQ_PARALLELISM 405
SP_DEQ_PARTIAL_REPAIR_CLR_OOS 405
SP_DEQ_PART_TABLE_UPDATE 406
SP_DEQ_READ_BUFFER_SIZE 406
SP_DEQ_ROW_LOCK_THRESHOLD 407
SP_DEQ_SKIP_LOB 407
SP_DEQ_THREADS 407
SP_DEQ_TIMEOUT 408
SP_DEQ_USE_SP_CKSUM 408
copy/appendコマンドのパラメーター 409
SP_OSY_COMPRESSION 409
SP_OSY_LOCK_TIMEOUT 409
SP_OSY_POST_TIMEOUT 409
SP_OSY_THREADS 409

analyze configコマンドのパラメーター 410
SP_ANL_CHECKPOINT_FREQ 410
SP_ANL_CHECKPOINT_TIME 410
SP_ANL_RUN_TIME 410
非推奨のパラメーター 411
非推奨のCaptureパラメーター 411
非推奨のReadパラメーター 411
非推奨のOracle Postパラメーター 412
非推奨のOpen Target Postパラメーター 412
非推奨のExportパラメーター 412
非推奨のCompareパラメーター 412
非推奨のqueueパラメーター 413
その他の非推奨パラメーター 413

PostgreSQL用SharePlexコマンド 414
SharePlexコマンド一覧 414
設定コマンド 414
情報コマンド 415
パーティション化コマンド 415
同期コマンド 416
メンテナンスコマンド 416
PostgreSQL向けSharePlexコマンドのアルファベット順リファレンス 417
PostgreSQLの設定の中断 417
用途 417
構文 417
PostgreSQLの設定のアクティベーション 418
アクティベーションのガイドライン 418
アクティベーションのステータスと結果の表示 418
アクティベーションに失敗した場合の対処法 419
用途 419
構文 419
PostgreSQL用パーティションの追加 421
用途 421
構文 421
PostgreSQLのcancelコマンド 424
PostgreSQLのclear historyコマンド 426

PostgreSQLのclear statusコマンド 428
PostgreSQLのcompare/compare using 430
サポートされている操作 430
サポートなし 430
その他の条件 431
特別な使用例 431
compareの仕組み 432
SQLログファイルの管理 433
複数のプロセスの実行 434
設定のサブセットのcompare 434
compareする行の制御 434
プロセスの特定 435
sp_ctrlでステータスと結果を表示 435
警告とエラーの表示 435
compareジョブのキャンセル 436
compare履歴とログの管理 436
バッチサイズのコントロール 436
非同期行のrepair 437
用途 437
構文 437
PostgreSQLのcompare statusコマンド 448
PostgreSQLの設定のコピー 451
用途 451
構文 451
PostgreSQLの設定の作成 452
用途 452
構文 452
PostgreSQLの設定の非アクティベーション 453
用途 453
構文 453
PostgreSQLのパーティションの削除 454
用途 454
構文 454
PostgreSQLのdrop partition schemeコマンド 456
用途 456
構文 456

PostgreSQLの設定の編集 457
用途 457
構文 457
PostgreSQLのフラッシュ 458
Postの再開 458
用途 458
構文 458
PostgreSQLのjob statusコマンド 461
PostgreSQLの設定のリスト 465
用途 465
構文 465
PostgreSQLのパーティションの変更 466
用途 466
構文 466
PostgreSQLの設定のパージ 468
用途 468
構文 468
PostgreSQLの照合 469
用途 470
構文 470
PostgreSQLの設定名の変更 472
用途 472
構文 472
PostgreSQLのrepair/repair using 473
サポートされている操作 473
サポートなし 473
その他の条件 474
repairコマンドの使用方法 474
repairを実行するタイミング 475
特別な使用例 475
repairの仕組み 477
repairの条件 477
プロセス 478
エラー処理 478
SQLログファイルの管理 478
複数のプロセスの実行 479

設定のサブセットのrepair 479
どの行をrepairするかを制御 480
プロセスの特定 480
sp_ctrlでステータスと結果を表示 480
警告とエラーの表示 481
repairジョブのキャンセル 481
compare履歴とログの管理 481
バッチサイズのコントロール 482
用途 482
構文 482
PostgreSQLのrepair statusコマンド 492
PostgreSQLのverify config 494
用途 494
構文 495
PostgreSQLのview partitions 496
用途 496
構文 496
PostgreSQLのshow capture 497
基本コマンド 497
詳細な統計情報 497
用途 499
構文 500
PostgreSQLのshow post 501
基本的なshow postコマンド 501
詳細なshow postコマンド 501
用途 503
構文 503
show_last_posted 505
用途 505
構文 505

PostgreSQL用SharePlexパラメーター 506
ユーザ設定可能なパラメーターの説明 507
Captureパラメーター 507
SP_CAP_INCLUDE_UNCHANGED_COL 507
SP_CAP_REDUCED_KEY 507

SP_CAP_ONELINE_DEBUG 507
SP_CAP_CHECKPOINT_FREQ 508
SP_CAP_CHECKPOINT_TIME 508
SP_CAP_DENIED_SESSION_PG 509
Open Target Posterパラメーター 511
SP_OPX_REDUCED_KEY 511
SP_OPX_XML_MARK_NULL 511
SP_OPX_CREATE_ORIGIN_PG 511
SP_OPX_LOG_CONFLICT 512
PostgreSQL Open Target Posterでサポートされる追加パラメーターのリスト 512
Oracle Posterパラメーター 514
SP_OPO_HINTS_LIMIT 514
Oracle Posterプロセスでサポートされる追加パラメーターのリスト 514
Readパラメーター 515
SP_ORD_FIRST_FIND 515
SP_ORD_HP_IN_SYNC 515
PostgreSQL Readerプロセスでサポートされる追加パラメーターのリスト 516
Compare/Repairパラメーター 517
SP_DEQ_PG_FORCE_PARALLEL 517
SP_DEQ_PG_PARALLEL_MAX 517
SP_DEQ_PG_PARALLEL_SETUP_COST 518
SP_DEQ_PG_PARALLEL_TUPLE_COST 518
SP_DEQ_BATCH_ENABLED 518
SP_DEQ_MAX_BATCH_MBYTES 518
SP_DEQ_COMMIT_FREQ 519
SP_DEQ_USE_SP_CKSUM 519
SP_DEQ_PG_DECLARE_FETCH_SIZE 519
PostgreSQL用のSP_DEQ_READ_BUFFER_SIZE 520
Exportパラメーター 520
Importパラメーター 520
Queueパラメーター 520
sp_copパラメーター 521
Systemパラメーター 521

一般的なSharePlexユーティリティ 522
構成スクリプト 523

サポート対象のソースおよびターゲットデータベース 523
config.sqlの使用 523
build_config.sqlの使用 524
PostgreSQLの設定スクリプト 526
サポート対象のソースおよびターゲットデータベース 526
pg_config.sqlの使用 526
pg_build_config.sqlの使用 527
cleanup.sql 529
サポート対象データベース 529
clean_vardir.sh 530
サポート対象データベース 530
シェル要件 530
create_ignore.sql 532
サポート対象データベース 532
create_ignore.sqlの実行 532
SHAREPLEX_IGNORE_TRANSの実行 533
hana_cleansp 534
使用ガイドライン 534
このユーティリティの機能 534
サポート対象データベース 534
MSS_cleansp 536
サポート対象データベース 536
MySQL_cleansp 538
使用ガイドライン 538
サポート対象データベース 538
ora_cleansp 540
サポート対象データベース 540
ora_cleanspの他の代替手法 540
シェル要件 540
ora_cleanspの仕組み 540
pg_cleanspユーティリティ 543
使用ガイドライン 543
このユーティリティの機能 543
snowflake_cleanspユーティリティ 545
使用ガイドライン 545
このユーティリティの機能 545

SharePlexライセンスユーティリティ 547
UnixおよびLinuxのライセンスユーティリティ 547
ライセンスキーの追加 549
ライセンスキーの表示 550
ライセンスキーの削除 550
OTS 552
サポート対象データベース 552
要件 552
スタンドアロンOTSのインストール方法 552
OTSの実行方法 553
仕組み 553
provision 556
サポート対象データベース 556
provision使用のガイドライン 556
provisionの実行 557
provisionによる変更を取り消す 558
既知の問題 558
qview 559
サポート対象データベース 559
qviewの実行 559
qviewコマンドの概要 559
キューの一覧表示 560
説明 560
構文 560
古いサブキューの削除 560
説明 561
このコマンドの実行方法 561
構文 561
完全なロールバックの実行 562
説明 562
このコマンドの実行方法 562
構文 563
show_scnユーティリティ 564
サポート対象データベース 564
show_scnの実行 564
出力 564

show_last_postedユーティリティ 565
サポート対象データベース 565
show_last_postedの実行 565
出力 565
socket_test 567
サポート対象データベース 567
socket_testの実行 567
sp-bininfo 569
サポート対象データベース 569
サポート対象プラットフォーム 569
sp-bininfoの実行 570
sp_wallet 571
サポート対象データベース 571
sp_walletの実行 571
sp_hsm 573
サポート対象データベース 573
sp_hsmユーティリティの実行 573
sp_security 574
説明 574
SSL/TLSを有効にする 574
SSL/TLSを無効にする 575
現在のSSL/TLS設定を表示する 576
トリガスクリプト 577
サポート対象データベース 577
sp_add_trigger.sql 577
構文 577
sp_change_trigger.sql 578
構文 578
sp_remove_trigger.sql 578
構文 578
PostgreSQL用トリガスクリプト 579
サポート対象データベース 579
sp_pg_add_trigger.sql 579
構文 580
sp_pg_change_trigger.sql 580
構文 580

sp_pg_remove_trigger.sql 580
構文 580
データベース・セットアップ・ユーティリティ 581

SharePlex for Oracleクラウド インフラストラクチャ 582
Oracle Cloud-ExaCSとDBCSのデータベースセットアップ 582
サポート対象データベース 582
Oracleのセットアップを実行するタイミング 582
サポート対象のOracle接続 583
必要な実行の権限 583
SharePlexに付与される権限 584
要件 584
SharePlexスキーマのストレージ要件 585
Oracle cloud-ExaCSおよびDBCSのデータベースセットアップの実行 586

SharePlexの変数 590

会社概要 592
Questのお問い合わせ先 592
テクニカルサポートのリソース 592

このガイドについて

本ガイドは、以下の項目のリファレンスです。

- **SharePlexコマンド** — sp_ctrlを介してレプリケーションを制御するコマンドの概要、コマンドの認証レベルの説明、各コマンドの詳細な説明。
- **SharePlexパラメータ** — SharePlexチューニングパラメータの概要、その設定方法と保存方法、SharePlexユーザが変更できるパラメータの詳細な説明。
- **SharePlexユーティリティ** — SharePlexユーティリティとその使用方法。
- **SharePlexエラーメッセージ** — 一般的なエラーコード、その原因、および推奨される解決方法。

『SharePlexリファレンスガイド』のユーザは、コマンドを発行したり、パラメータを変更したり、問題を解決しようとしたりする前に、SharePlexを十分に理解しておく必要があります。

その他のSharePlex関連ドキュメント

SharePlexドキュメント一式については、https://support.quest.com/shareplex/technical-documentsをご覧ください。

このガイドで使用される表記規則

このマニュアルで使用される表記規則

このガイドでは、以下の表記規則が使用されています。

- **太字**は、コマンドまたはオプションの必須コンポーネントを表しており、表示されている通りに入力する必要があります。
- *斜体*は、ユーザが定義、命名、または入力する変数を表します。
- {中括弧}内の項目は、必須の引数です。
- [角括弧]は、オプションのコマンドコンポーネントを表します。また、コマンド文字列の例で、長い文字列の中の必須のユーザ定義変数を強調するために使用することもできます。

  例:

  reconcile queue {*queuename*} for {*datasource-datadest*} [**on** *host*]

- 角括弧または中括弧内のコンポーネントが縦棒(「パイプ」文字、「|」)で区切られている場合、それらのコンポーネントのいずれか1つのみを使用できることを示しています。

  例:

  abort service {*service* | **all**}

コマンド、プログラム、ディレクトリ、ファイルの名前は**太字**で表示されます。

その他の名前は、デフォルトのフォントを使用して大文字で表示されます。

例えば、次のようなものです。

> **sp_ctrl**プログラムは、**bin**ディレクトリにあります。
>
> **oramsglst**ファイルを開きます。
>
> ORACLE_HOMEの値を見つけます。
>
> 「**適用**」をクリックします。

プロンプトやコマンド出力などのシステム表示は、等幅(固定スペース)フォントで表示されます。

例えば、次のようなものです。

```
sp_ctrl(sysA)>
User is a viewer (level=3)
```

Windowsのメニュー項目、ダイアログボックス、およびダイアログボックス内のオプションは、**太字**で表わされます。

例:

> 「**ファイル**」メニューから、「**印刷**」を選択します。

システム名は、一般的な名前または架空の名前で表わされます。必要に応じて、ソースシステム(またはプライマリシステム)を*SysA*と呼びます。ターゲットシステム(またはセカンダリシステム)は、*SysB*、*SysC*、*SysD*などと呼びます。

1

Oracle用SharePlexコマンド

SharePlexコマンドは、レプリケーションプロセスの設定、開始、停止、制御および監視を行います。SharePlexコマンドは、**sp_ctrl**インターフェイスを通じて発行されます。

内容

SharePlexコマンド一覧

SharePlexコマンドのアルファベット順リファレンス

SharePlexコマンド一覧

このセクションでは、制御対象のプロセスまたは機能に従ってグループ化されたすべてのSharePlexコマンドの概要を示します。

SharePlexコマンドの権限レベルとセキュリティグループの詳細については、『SharePlex管理ガイド』を参照してください。

設定コマンド

設定コマンドは、SharePlex設定ファイルの管理全般を制御します。

設定コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
abort config	1	すべて	データがポストされたかどうかにかかわらず、指定された設定ファイルのレプリケーションを直ちに終了します。設定を無効化し、キューとプロセスを削除します。
activate config	1	すべて	設定で指定されたデータのレプリケーションを開始します。
copy config	2	すべて	設定を複製し、新しいファイルとして編集して保存します。
create config	2	すべて	レプリケートするオブジェクトに関する情報を入力する新しい設定ファイルを作成します。
deactivate config	1	すべて	キューが削除される前にキュー内のデータがターゲットデータベースにポストされるように、設定で指定されたデータのレプリケーションを安全に終了します。
edit config	2	すべて	デフォルトのテキストエディタを使用して、既存の設定ファイルを変更します。
list config	3	すべて	ソースシステム上のすべての設定ファイルを、アクティブと非アクティブの両方で表示します。
purge config	1	すべて	アクティブな設定によって生成されたキューからデータを削除しますが、設定を非アクティブにしたりキュー自体を削除したりはしません。

remove config	2	すべて	システムから設定ファイルを完全に削除します。
rename config	2	すべて	設定ファイルに別の名前を割り当てます。
verify config	3	すべて	設定内のオブジェクトが有効であること、および設定が適切な構造になっていることを検証し、アクティベーションとレプリケーションが正常に行われるようにします。
view config	3	すべて	設定ファイルの内容を表示します。

接続コマンド

接続コマンドは、**sp_ctrl**インターフェイスを介したレプリケーションシステムへの接続を制御します。

接続コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
connection	2	すべて	ソースまたはターゲットデータベースに接続するための接続設定を指定します。
exitコマンド	3	すべて	**sp_ctrl**コマンドラインセッションを終了し、**TCP/IP**リンクを切断します。（**quit**と同じ意味で使用されます）。
hostコマンド	3	すべて	現在の**sp_ctrl**接続のデフォルトマシンを設定します。
portコマンド	3	すべて	現在の**sp_ctrl**セッションのデフォルト**sp_cop**ポート番号を設定します。
quitコマンド	3	すべて	**sp_ctrl**コマンドラインセッションを終了し、**TCP/IP**リンクを切断します。（**exit**と同じ意味で使用されます）。

暗号化コマンド

暗号化コマンドを使用して、ネットワーク上で送信されるデータを暗号化できます。

暗号化コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
create encryption key	2	すべて	ネットワーク上のデータを暗号化するための暗号化キーを作成します。
reset encryption key	2	すべて	暗号化キーを削除します。
set encryption key	2	すべて	Export-Importペアの暗号化キーを設定します。

コマンド	認証レベル	サポート対象のターゲット	説明
create encryption key	**2**	すべて	ネットワーク上のデータを暗号化するための暗号化キーを作成します。
reset encryption key	**2**	すべて	暗号化キーを削除します。
set encryption key	**2**	すべて	Export-Importペアの暗号化キーを設定します。

情報コマンド

情報コマンドはレプリケーション環境に関する情報を提供します。これらのコマンドは、レプリケーションの問題解決を試みるとき、または特定のプロセス条件を表示するときに使用します。

情報コマンド一覧

コマンド	認証レベル	説明
append status	**3**	**append using**および**append**コマンドのステータスと結果を表示します。
copy status	**3**	**copy using**および**copy**コマンドのステータスと結果を表示します。
compare status	**3**	**compare using**および**compare**コマンドのステータスと結果を表示します。
lstatusコマンド	**3**	SharePlexのレプリケーションの状態に関する詳細情報を表示します。
job statusコマンド	**3**	**append**、**compare**、**copy**、および**repair**コマンドの現在のステータスと履歴を表示します。
orainfoコマンド	**3**	Oracleデータベースの情報を表示します。
qstatusコマンド	**3**	capture、export、およびpostキューの状態を表示します。

コマンド	認証レベル	説明
repair status	2	**repair**および**repair using**コマンドのステータスと結果を表示します。
report	3	テーブルの**append**、**compare**、**copy**、および**repair**の履歴を表示します。
showコマンド	3	システム上の各レプリケーションプロセスで処理されているデータのソースとターゲットを表示し、各プロセスのステータスを表示します。
show captureの使用	3	チューニングや問題解決に使用するために、Captureプロセスの簡単な、または詳細な統計情報を表示します。
show config	3	アクティブな設定のプロパティを表示します。
show export	3	ターゲットシステムに送信されたメッセージの数を表示します。
show import	3	ソースシステムから受信したメッセージの数を表示します。
show log	3	イベントログ、コマンドログ、確認ログ、追跡ログ、またはプロセスログを表示します。
show post	3	チューニングや問題解決に使用するために、Postプロセスの簡単な、または詳細な統計情報を表示します。
show read	3	チューニングや問題解決に使用するために、Readプロセスの簡単な、または詳細な統計情報を表示します。
show SQL	3	Postプロセスで処理された現在または最後のSQLステートメントを表示します。
show statusdb	3	重要なレプリケーションイベントの記録を含むステータスデータベースを表示します。
show sync	3	非同期状態に関する情報を表示します。
Status	3	SharePlexのレプリケーションの状態の概要を表示します。

メンテナンスコマンド

メンテナンスコマンドは、コマンドの表示とファイルを制御します。

メンテナンスコマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
clear history	2	Oracle	**compare**、**compare using**、**repair**、**copy**、および**append**コマンドの実行に関するジョブ情報とソースログを削除します。
clear status	2	すべて	ステータスデータベースから警告メッセージを削除します。
remove log	3	Oracle	システムから古いcompareレポートとログを削除します。
set log	3	すべて	**show log**コマンドのデフォルト値を設定します。
truncate log	1	すべて	ログに記録されたコマンドを除き、イベントログからレコードを消去します。

パラメーターコマンド

パラメーターコマンドは、SharePlexの動作の仕方を定義する環境パラメーターを制御します。

パラメーターコマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
list param	3	すべて	SharePlexパラメーターのリストを、現在値、デフォルト値、およびセットアットポイントと共に表示します。
reset param	1	すべて	SharePlexパラメーターの値をデフォルト設定に復元します。
set param	1	すべて	グローバルベースまたは指定されたレプリケーションプロセスに基づいてSharePlexパラメーターの値を変更します。

パーティション化コマンド

パーティション化コマンドは、水平分割レプリケーションに使用されるパーティションスキームと行パーティションを作成、管理、表示します。

パーティション化コマンド一覧

コマンド	認証レベル	サポート対象	説明
add partition	2	すべて	パーティションスキームと行パーティションを作成します。
drop partition	2	すべて	パーティションスキームから行パーティションを削除します。
drop partition scheme	2	すべて	パーティションスキームを削除します。
modify partition	2	すべて	パーティションスキームの行パーティションを変更します。
view partitions	3	すべて	既存のパーティションスキームとその行パーティション仕様を表示します。

プロパティコマンド

プロパティコマンドは、ユーザの権限レベル、ローカルシステム、およびローカルのSharePlexインストールに関する情報を表示します。

プロパティコマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
認証レベル	3	すべて	**sp_ctl**にログインしたユーザのSharePlex認証レベルを表示します。
hostinfoコマンド	3	すべて	マシン名とオペレーティングシステムを表示します。
version	3	すべて	マシンにインストールされているSharePlexのバージョンを表示します。

ルートコマンド

ルートコマンドを使用すると、非推奨のレプリケーションルートに残っているキューやプロセスを削除できます。

コマンド	認証レベル	サポート対象のターゲット	説明
delete capture queue	2	すべて	captureキューを削除します。
delete export process	2	すべて	Exportプロセスを削除します。
delete export queue	2	すべて	exportキューを削除します。
delete post queue	2	すべて	postキューを削除します。

実行コマンド

実行コマンドは、プロセスの開始と停止、レプリケーションの開始とシャットダウン、データの再同期、確認メッセージの有効化と無効化などのオプションで、SharePlexの操作を制御します。

実行コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
abort service	2または1	すべて	処理を終了することなく、1つまたはすべてのレプリケーションプロセスを停止します。(Captureにはレベル1が必須)
clear status	3	すべて	ステータスデータベースから警告メッセージを削除します。
シャットダウン	1	すべて	**sp_cop**およびすべてのレプリケーションプロセスをシャットダウンします。
start	2または1	すべて	レプリケーションプロセスを開始します。(Captureにはレベル1が必須)
startup	1	すべて	**startup**コマンドを使用して、すべてのプロセスを一度に開始します。
Stop	2または1	すべて	レプリケーションプロセスがキューからの読み込みまたはキューへの書き込みを終了した後、レプリケーションプロセスを安全に停止します。(Captureにはレベル1が必須)

同期コマンド

同期コマンドは、データ同期の側面を制御します。

同期コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
cancel	2	Oracle	実行中の**compare**、**repair**、**copy**または**append**コマンドジョブをキャンセルします。
compare / compare using	2	Oracle	ソーステーブルとターゲットテーブルの内容をcompareし、同期を検証します。
copy / append	2	Oracle	Oracle Export/Importユーティリティを使用して、レプリケーションのセットアップまたは既にレプリケーション内にあるテーブルの（再）同期を行えるようにします。
disable jobs / enable jobs	2	Oracle	高可用性設定またはピアツーピア設定のセカンダリシステムで同期コマンドの実行を許可するかどうかを制御します。
flushコマンド	2	すべて	このコマンドはデータストリームにマーカーを置いて、ソースデータとターゲットデータ間の同期ポイントを確立してから、その時点のPostプロセスを停止します。
reconcileコマンド	1	すべて	ターゲットシステムに適用されたバックアップまたはコピーの結果を、ソースシステムのユーザトランザクションと同期します。
repair / repair using	2	Oracle	ターゲットテーブル内の同期していない行をrepairし、ソースとターゲットを同期状態に戻します。

ターゲット設定コマンド

ターゲット設定コマンドは、異種レプリケーションをサポートするようにPostプロセスを設定します。

ターゲットコマンド

コマンド	認証レベル	サポート対象のターゲット	説明
Target(ターゲット)	2	すべて	ターゲット設定に保存されているプロパティは、Postがレプリケートされたデータをどのように処理して、フォーマットし、出力するかを制御します。
typemap	2	すべて	ソースデータ型がターゲットデータベースにどのようにマッピングされるかを示します。

その他コマンド

これらのその他コマンドは、SharePlexを使用するための追加のコントロールを提供します。

その他コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
copy cache	2	Oracle	オブジェクトキャッシュをソースからターゲットにコピーします。
editコマンド	3	すべて	直前に発行したコマンドをテキストエディタに表示して編集し、編集したバージョンを実行できるようにします。
expandコマンド	3	すべて	ワイルドカード指定を展開し、展開された指定を画面に表示します。
helpコマンド	3	すべて	**sp_ctrl**コマンドラインインターフェイスからのSharePlexコマンドの説明と構文を表示します。
redoコマンド	該当なし	すべて	直前のコマンドを再度実行します。
usage	3	すべて	SharePlexコマンドの構文を表示します。
view logオプション	3	すべて	**show log**コマンドのデフォルト値を表示します。

SharePlexコマンドのアルファベット順リファレンス

ここから、SharePlexコマンドの詳細なリファレンスドキュメントをアルファベット順に記載します。

abort config

abort configコマンドを使用すると、設定を即座に非アクティベーションし、関連するすべてのキューとレプリケーションプロセスを同時に削除できます。 このコマンドは、キュー内のデータがポストされたかどうかにかかわらず、ソースシステムとターゲットシステムで設定されたデータソースのすべてのレプリケーションアクティビティを停止します。これは強制的な非アクティベーションです。

abort configコマンドは、システム、設定、またはデータに問題が発生し、ターゲットシステムへのポストを防ぐ必要がある場合に有効です。パイプラインに残っているデータがすべて削除されるため、このコマンドが有効になると、ソースインスタンスとターゲットインスタンスは同期しなくなります。

abort configコマンドの実行前または実行中に、SharePlexレプリケーションプロセスが停止した場合、このコマンドも停止します。プロセスが再開されると、コマンドは動作を再開します。これにより、**abort config**はネットワークが一時的にダウンした場合でも機能し、接続が回復するまでキューに残ります。

同じソースシステム上に複数のアクティブな設定がある場合は、それぞれのレプリケーションストリームを分離する名前付きexportキューがある場合にのみ、**abort config**コマンドを使用してください。名前付きexportキューがない場合、SharePlexは、レプリケートされたすべてのデータを1つのexportキューを通して転送します。そして、1つの設定に対して**abort config**コマンドを実行すると、すべてのアクティブな設定のデータが削除されます。名前付きキューを作成するには、『*SharePlex管理者ガイド*』の第5章を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者
発行対象:	ソースシステム
関連コマンド:	**deactivate config**、**purge config**

構文

基本コマンド	**リモートオプション**
abort config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	中断する設定の名前。設定名では大文字と小文字が区別されます。 例: `sp_ctrl(sysA)>` **abort config sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

abort service

abort serviceコマンドを使用すると、SharePlexレプリケーションプロセス（サービス）を、そのプロセスがキューからの読み取り、またはキューへの書き込みを完了したかどうかにかかわらず、直ちに終了させることができます。このコマンドは、プロセスがグレースフルに停止するのを待てない場合、または、**stop**コマンドを発行してもプロセスがシャットダウンしない異常な状況で有効です。

このコマンドはレプリケートされたデータには影響しません。データは安全にキューに残り、プロセスが再開されると、SharePlexは通常の処理を再開します。

abort serviceコマンドで中断されたプロセスは、SharePlexをシャットダウンして再起動しても、停止したままになります。**start**コマンドを使用した場合のみ、プロセスを再開できます。

abort serviceコマンドには、他のルートのレプリケーションに影響を与えることなく、任意の（あるいはすべての）システムまたは名前付きキューのプロセスを中断できるオプションがあります。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター（2）、Captureを中断する場合は管理者（1）
発行対象:	Capture、Read、およびExportの**abort service**は、ソースシステムに対して発行されます。 ImportおよびPostの**abort service**は、ターゲットシステムに対して発行されます。
関連コマンド:	**start**、**stop**

構文

基本コマンド	コマンドオプション	リモートオプション
abort service *service*	[**to** *host*]	[**on** *host* |
	[**from** *host*]	**on** *host:portnumber* |
	[**for** *datasource*]	**on** *login/password@host* |
	[**for** *datasource-datadest*]	**on** *login/password@host:portnumber*]
	[**queue** *queuename*]	

構文の説明

コンポーネント	説明
service	必須。中断するSharePlexレプリケーションプロセス。有効な値: capture

コンポーネント	説明
	read **export** **import** **post**
to *host*	このオプションは、指定したシステムへのExportを中断しますが、他のシステムへのExportは続行されます。 • hostはターゲットシステムの名前です。 **例**: sp_ctrl(sysA)> **abort service export to sysB** この例では、sysAからsysBへのExportが中断されますが、sysAから他のターゲットシステムへのその他のExportプロセスは続行されます。
from *host*	このオプションは、指定したシステムからのImportを中断しますが、他のシステムからのImportは続行されます。 •*host*はソースシステムの名前です。 **例**: `sp_ctrl(sysD)>` **abort service import from sysC** この例では、sysDのsys CからのImportが中断されますが、sysDの他のソースシステムからのImportプロセスは続行されます。
for *datasource*	このオプションは、指定したデータソースのCaptureまたはReadを中断しますが、同じシステム上の他のレプリケーションデータソースは影響を受けません。 • *datasource*はo.SIDで表されます。ここで、SIDはソースOracleインスタンスのORACLE_SIDです。 **例**: `sp_ctrl(sysA)>` **abort service read for o.oraA** この例では、sysAのインスタンスoraAに対するReadが中断されますが、sysAの他のインスタンスに対する他のReadプロセスは続行されます。
for *datasource-datadest*	このオプションは、指定したソースSIDからターゲットSIDへのデータストリームに対するPostを中断しますが、他のPostアクティビティは続行されます。 • *datasource*は**o.***SID*で表わされます。ここで*SID*はソースOracleインスタンスのORACLE_SIDです。 • *datadest*は**o.***SID*で表わされます。ここで、*SID*はターゲットOracleインスタンスのORACLE_SIDです。 例:

コンポーネント	説明
	`sp_ctrl(sysC)>` **abort service post for o.oraA-o.oraC** この例では、インスタンスoraAからoraCへのレプリケートに対するsysCのPostが中断されますが、sysC上の他のPostプロセスは続行されます。
queue *queuename*	このオプションは、名前付きキューに関連付けられたExport、Import、またはPostプロセスを中断しますが、他の名前付きキューを介したレプリケーションは続行されます。 • *queuename*はキューの名前です。 **例**: `sp_ctrl(sysA)>` **`abort service export queue QA`** この例では、sysAでの名前付きexportキューQAを介したデータのExportが中断されますが、それ以外のすべての名前付きキューに割り当てられたデータのExportは続行されます。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

activate config

設定をアクティベーションするには、**activate config**コマンドを使用します。レプリケーションは、アクティベーションプロセスが完了すると直ちに開始されます。設定をアクティベーションすると何が起こるかの詳細については、『SharePlex管理ガイド』を参照してください。

アクティベーションプロセスは、設定ファイルを読み込み、そこからSharePlexが以下を実行するために必要なすべての情報を取得します。

- レプリケーション対象のオブジェクトを特定する
- レプリケートされた変更を適切なターゲットデータベースにルーティングする
- Postがターゲットに変更を適用するために使用するSQLを生成する
- レプリケーションに追加されたすべてのテーブルをアクティベーションする

sp_copが設定をアクティベーションするために呼び出すプロセスは**sp_tconf**です。

アクティベーションは、複数のテーブルを同時にアクティベーションするために、非同期の並列処理スレッドを作成します。各テーブルは、テーブルをアクティベーションするために最低限必要なだけの、非常に短い時間ロックされます。各テーブルのレプリケーションは、そのアクティベーションが完了するとすぐに開始されます。

1つまたは複数のテーブルがアクティベーションに失敗した場合、SharePlexは他のテーブルのアクティベーションを続行します。レプリケーション設定でアプリケーションがテーブルのNOWAITロックを使用している場合、アクティベーションするオブジェクトのロックを取得しようとすると、失敗する可能性があります。

アクティベーションのガイドライン

- 設定をアクティベーションするには、レプリケートするオブジェクトを含むデータベースをマウントし、開いておく必要があります。アクティベーションにかかる時間は、設定されたオブジェクトのサイズ、数、構造によって異なります。
- 各システム上のデータソース(Oracleインスタンス)ごとに1つの設定をアクティベーションできます。例えば、インスタンスora10にConfigA、ConfigB、ConfigCがある場合、一度にアクティベーションできるのはそのうちの1つだけです。同じデータソースについて別の設定をアクティベーションすると、最初の設定は自動的に非アクティベーションされます。
- アクティベーション中は、DDL(TRUNCATEを含む)を実行しないでください。アクティベーション中に許可される変更はDMLの変更だけです。
- アクティベーションを実行するには、アプリケーションに再試行ロジックが必要です。レプリケーション設定でテーブルがNOWAITロックされている場合、アプリケーションがアクティベーションするオブジェクトのロックを取得しようとすると、失敗する可能性があります。
- アクティベーションプロセスでは、アクティベーションが完了するまで**sp_ctrl**インターフェイスの制御を保持します。同一システム上の異なるデータソースの複数の設定をアクティベーションするには、最初の設定をアクティベーションしてから、**sp_ctrl**の別のセッションを開き、2つ目の設定をアクティベーションします。アクティベーションする設定の数だけ**sp_ctrl**のセッションを開きます。
- 設定をアクティベーションする前に、**verify config**コマンドを使用して、アクティベーションとレプリケーションを正常に実行するための基本要件が満たされていることを確認します。このコマンドは、アクティベーションの失敗を引き起こす可能性のある潜在的な問題を警告します。

アクティベーションのスレッド数の設定

アクティベーションのスレッド数はグローバルに(すべてのアクティベーションに対して)設定することができ、任意のアクティベーションに対してこの設定を上書きすることができます。

スレッド数をグローバルに設定するには:

1. **sp_ctrl**を実行します。
2. 次のコマンドを発行します。最大32スレッドまで使用できます。

   `sp_ctrl>` **set param SP_OCF_THREAD_COUNT** *number_of_threads*

現在のアクティベーションのスレッド数を設定するには:

[**threads**=*n*]オプションを指定して**activate config**コマンドを発行します。

アクティベーションのステータスと結果の表示

SharePlexはオブジェクトを、設定ファイルの順番ではなく、オブジェクトIDに従ってアクティベーションするので、アクティベーションの順番を予測する方法はありません。

SharePlexは個々のテーブルのアクティベーションに失敗してもしなくてもアクティベーションを続行するため、テーブルのアクティベーションに失敗すると、**sp_ctrl**プロンプトに次のエラーメッセージを表示して警告します。「`WARNING, not all objects activated successfully. Check activation log.`」(警告、一部のオブジェクトが正常にアクティベーションされませんでした。アクティベーションログを確認してください。)

アクティベーションの結果を表示するには:

show configコマンドを実行します。

アクティベーションに失敗した場合の対処法

テーブルや設定全体のアクティベーションに失敗する原因は多数あります。例えば、設定ファイルの1つ以上のコンポーネントが誤って入力された場合、影響を受けるオブジェクトのアクティベーションは失敗します。

アクティベーションの前に**verify config**コマンドを実行しなかった場合は、今すぐ実行し、検出された問題を修正してください。その後、もう一度アクティベーションを実行してください。トラブルシューティングの詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle(全オプション)
サポート対象のターゲット:	すべて
認証レベル:	管理者(1)
発行対象:	ソースシステム
関連コマンド:	**abort config**、**copy config**、**create config**、**deactivate config**、**edit config**、**list config**、**purge config**、**remove config**、**rename config**、**show config**、**verify config**、**view config**

構文

基本コマンド	コマンドオプション	リモートオプション
activate config*filename*	[**threads=***n*]	[**on** *host* |
	[**nolock**]	**on** *host:portnumber* |
	scn=*scn_value*	**on** *login/password@host* |
	seqno=*log_sequence_number*	**on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	必須。アクティベーションする設定の名前。設定名では大文字と小文字が区別されます。 例: `sp_ctrl(sysA)>` **activate config sales**
threads=*n*	(Oracleで有効)アクティベーションプロセスが生成する分析スレッドの数を設定するには、このオプションを使用します。このオプションは、SP_OCF_THREAD_COUNTパラメータで設定されたデフォルト値を上書きします。 *n*の有効な値の範囲は1～32ですが、5スレッド以下にすることをお勧めします。一般的に、それより多い数に設定するとスレッドを使用する利点が損なわれるからです。SharePlexは、分析するテーブル数より多くのスレッドを開始しません。 このオプションを使用する場合は、必須コマンド引数の後に記述しなければなりません。 例: `sp_ctrl(sysA)>` **activate config sales threads=3**
nolock	(Oracleで有効)レプリケーションに追加するテーブルをロックせずにアクティベーションするには、このオプションを使用します。
scn=*scn_value*	(Oracleで有効)REDOログの特定のSCNでレプリケーションを開始する設定をアクティベーションするには、このオプションを使用します。設定をアクティベーションする前に、以下を実行してください。 • 以前にアクティブな設定があった場合は、ソースとすべてのターゲットで**ora_cleansp**ユーティリティを実行して、環境をクリーンな状態に戻します。詳細については、「ora_cleansp」の540ページを参照してください。

コンポーネント	説明
	• **show scn**コマンドを使用して、すべてのPostプロセスから最後にポストされたトランザクションのSCNを取得します（名前付きキューを使用している場合）。これらのSCN値のうち最小値を**activate config**に使用します。 このオプションは、**nolock**オプションと併用しないでください。 例： `sp_ctrl>` **activate config myconfig scn=123456**
seqno=*log_sequence_number*	（Oracleで有効）特定のREDOログのシーケンス番号でレプリケーションを開始する設定をアクティベーションするには、このオプションを使用します。 このオプションは、**nolock**オプションと併用しないでください。 以前にアクティブな設定があった場合は、ソースとすべてのターゲットで**ora_cleansp**ユーティリティを実行して、環境をクリーンな状態に戻します。詳細については、「ora_cleansp」の540ページを参照してください。 例： **activate config myconfig seqno=98765**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

add partition

add partitionコマンドを使用すると、水平分割レプリケーションを設定するときに、パーティションスキームに行パーティションを追加できます。作成する行パーティションごとに、**add partition**を発行します。

このコマンドは、以下の必須コンポーネントを含む、パーティション作成に必要なすべての情報を取得します。

- パーティションスキーム名。新しいパーティションスキームを作成するには、そのスキームの最初の行パーティションを作成する**add partition**コマンドでスキームの名前を指定します。SharePlexは自動的にパーティションスキームを作成します。その後、そのパーティションスキームに行パーティションを追加するときに、その名前を指定します。
- 行パーティションを作成するハッシュ値または列条件の指定。
- 行パーティションで指定された行のルーティング。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	modify partition、drop partition、drop partition scheme、view partitions

構文

基本コマンド	コマンドオプション	リモートオプション
add partition to *scheme_name*	[**and name =** *partition_name*]	利用不可
set	[**and tablename =** *target_table*]	
{**condition =** *column_condition* |	[**and description =** *description*]	
hash = *hash_value*}		
および		
route = *routing_map*		

構文の説明

コンポーネント	説明
to *scheme_name*	**to**は、行パーティションが*scheme_name*に追加されることを示す必須キーワードです。 *scheme_name*はパーティションスキームの名前です。パーティションスキームは、最初に

コンポーネント	説明
	発行する**add partition**コマンドによって作成されます。このコマンドは、パーティション化する最初の行のセットも指定します。 水平分割を多用する場合は、パーティションスキームの命名規則を確立しておくことをお勧めします。
set	行パーティションの定義を開始する必須キーワード。
condition = *column_condition*	列条件に基づいて行パーティションを作成します。条件は引用符で囲む必要があります。**((region_id = West) and region_id is not null)**のような標準的なWHERE条件構文を使用します。 **condition**コンポーネントと**hash**コンポーネントは相互に排他的です。
hash = *value*	ハッシュ値に基づいて行パーティションを作成します。指定された値は、パーティションスキームにおける行パーティションの数を決定します。 **condition**コンポーネントと**hash**コンポーネントは相互に排他的です。
route = *routing_map*	このパーティションのルート。これは以下のいずれかになります。 **列条件に基づくパーティション:** 標準のSharePlexルーティングマップを指定します(例: **sysB@o.myora**、**sysB:q1@o.myora**または**sysB@o.myora+sysC@o.myora**(複合ルーティングマップ))。 ターゲットがJMS、Kafka、またはファイルの場合、ターゲットをx.jms、x.kafka、またはx.fileとして指定する必要があります(例: **sysA:hpq1@x.kafka**)。 異なる名前を持つ複数のターゲットテーブルにパーティションをルーティングするには、以下の手順を実行します。 • 異なるターゲット名ごとに**add partition**コマンドを個別に発行します。**tablename**オプションを使って名前を指定します。 • 設定ファイルで、このパーティションスキームを使用するエントリのターゲットテーブルとして、これらのターゲットテーブルのいずれかを指定します。SharePlexは、設定が有効化されると他の名前を検出します。 • SP_ORD_FIRST_FINDパラメーターを0に設定し、パーティションスキームのすべての列条件をSharePlexでチェックするようにします。デフォルトでは、SharePlexは、特定の行変更がパーティションスキームの1つの列条件のみを満たすことを想定しています。 **ハッシュに基づくパーティション:** 以下のフォーマットを使用して、各パーティションに名前付きpostキューを作成するようにSharePlexに指示します。 *host*:*basename*|#{**o.***SID* | **r.***database*} ここで • *host*はターゲットシステムの名前です。

コンポーネント	説明
	• *basename*はすべてのキューに割り当てられる基本名です。 • **|#**は、1から**hash**で設定された値までの整数を基本名に付加することでキューに連番を付けるようにSharePlexに指示します。 • **o.***SID*（Oracleターゲットの場合）または**r.***database*（Open Targetターゲットの場合）。
name = *name*	（推奨）このパーティションの短い名前。このオプションは、列条件に基づくパーティションにのみ有効です。名前を付けておくと、将来パーティションを変更したり削除したりする必要が生じたときに、長い列の条件を入力する必要がなくなります。
tablename = *owner.table*	（オプション）このオプションは、ターゲットテーブルが複数あり、そのうちの1つ以上が異なる名前である場合に使用します。名前ごとに**add partition**コマンドを個別に発行します。 テーブル名は完全修飾名でなければなりません。大文字と小文字が区別される場合は、名前を引用符で囲んで指定する必要があります。 **例:** **add partition to scheme1 set name = p1 and condition = "C1 > 200" and route = sysb:p1@o.orasid and tablename = myschema.mytable**
description = *description*	（オプション）このパーティションの説明。

例

列条件に基づく行パーティション

異なる行のセットを異なるpostキューにルーティング:

```
sp_ctrl> add partition to scheme1 set name = q1 and condition = "C1 >= 200" and route = sysb:q1@o.orasid

sp_ctrl> add partition to scheme1 set name = q2 and condition = "C1 < 200" and route = sysb:q2@o.orasid
```

異なる行のセットを、異なるターゲットシステムおよびソースと異なるテーブル名にルーティング:

```
sp_ctrl> add partition to scheme1 set name = east and condition = "area = east" and route = sys1e@o.orasid and tablename = ora1.targ

sp_ctrl> add partition to scheme1 set name = west and condition = "area = west" and route = sys2w@o.orasid and tablename = ora2.targ
```

ハッシュ値に基づく行パーティション

行を4つのパーティションに分け、それぞれを異なるpostキューで処理します。

```
sp_ctrl> add partition to scheme1 set hash = 4 and route = sysb:hash|#@o.ora112
```

analyze config

analyze configコマンドを使用すると、設定ファイル内のテーブルの分析を実行できます。このコマンドは、テーブルのアクティビティに関する情報を収集します。

> **重要!** 分析を実行する前に、設定をアクティベーションしないでください。また、分析を実行するときに、他のアクティブな設定がないことを確認してください。このコマンドの使用方法は、実際のアクティベーションに似ています。

分析プロセスは、コマンドで指定された時間に収集されたデータに基づいてその結果を書き出します。レプリケーションストリームはその後、自動的にクリーンアップされます。

分析は、変数データディレクトリの**log**サブディレクトリにあるファイルに書き込まれます。ファイル名は次のようになります。

o.*datasource*-analysis.*actid*

分析プロセスは、レプリケーション内の各オブジェクトのアクティビティに関する情報とトランザクション情報を保持します。トランザクション情報は、同じレプリケーションストリーム(同じキューとプロセスのセット)でレプリケートする必要があるような、相互に関連するテーブルのグループを識別するために使用できます。

分析では、関連するテーブルの各グループ、テーブルごとの総操作数、グループの総操作数がリストされます。

例:

```
>cat o.w111a64f-analysis.1575

Activity Analysis


Group 1 of related tables: 1000 total operations in group

      "TEST"."SS2_TEST1" 346

      "TEST"."SS2_TEST2" 348

      "TEST"."SS2_TEST3" 306


Group 2 of related tables: 1124 total operations in group

      "TEST"."SRC_TEST1" 232

      "TEST"."SRC_TEST2" 177

      "TEST"."SRC_TEST3" 178

      "TEST"."SRC_TEST4" 175

      "TEST"."SRC_TEST5" 188

      "TEST"."SRC_TEST6" 174


 Tablename              Inserts         Updates        Deletes     Rollbacks    Total
 "TEST"."SS2_TEST2"     146             169            33          0            348
 "TEST"."SS2_TEST1"     140             176            30          0            346
 "TEST"."SS2_TEST3"     116             158            32          0            306
 "TEST"."SS2_TEST1"     75              114            29          14           232
```

```
Tablename              Inserts         Updates        Deletes     Rollbacks    Total
"TEST"."SS2_TEST5"     61              94             22          11           188
"TEST"."SS2_TEST3"     69              73             28          8            178
"TEST"."SS2_TEST2"     69              77             21          10           177
"TEST"."SS2_TEST4"     54              89             19          13           175
"TEST"."SS2_TEST6"     61              79             25          9            174
```

現在の分析の状況を表示するには:

分析の状況を表示するには、**show analyze**コマンドを使用します。

```
sp_ctrl (alvspxl11:8567)> show analyze detail


Host: alvspxl11.quest.com
                                  Operations
Source          Status                 Processed Since                Total    Backlog
------          ------------         ----------- -------               ------   -------
o.w111a64f      Running                      1497 17-Mar-12 10:41:54       1496         0

     Last operation processed:

          Redo log: 295 Log offset: 32327800

          UPDATE of "TEST"."SRC_TEST3" at 03/17/12 0:59:17


     Activation id                    : 1573
     Operations processed             : 1497
     Transactions processed           : 398
     Analysis complete                : 20-Mar-12 10:41:54
```

完了前に分析を終了するには:

分析が完了する前に終了するには、**abort config**コマンドまたは**deactivate config**コマンドを使用するか、SP_ANL_RUN_TIMEパラメータを変更します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者 (1)
発行対象:	ソースシステム
関連コマンド:	**abort config**、**copy config**、**create config**、**deactivate config**、**edit config**、**list config**、**purge config**、**remove config**、**rename config**、**show config**、**verify config**、**view config**

構文

基本コマンド	コマンドオプション	リモートオプション
analyze config *filename*	*n* {**minutes** | **hours** | **days**}	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	分析する設定ファイルの名前。設定名では大文字と小文字が区別されます。 例: `sp_ctrl(sysA)>`**analyze config sales**
n {**minutes** | **hours** | **days**)	分析するアクティビティの分数、時間数、または日数。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。

オプション	説明
	例: `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

例

```
analyze config testconf 5 days
```

```
sp_ctrl (alvspxl11:8567)> show analyze
Process           Source             Target           State         PID
------            ------------       -----------      -------       ----
Capture           o.w111a64f                          Running       2968
Analyze           o.w111a64f                          Running       2976
```

append status

append statusコマンドは、最後に実行された**copy**または**append**コマンドジョブのステータスを表示するために使用します。**append statusコマンド**を使用すると、copyまたはappendジョブ、またはcopyまたはappendジョブの一部の詳細なステータスを表示したり、SharePlexが履歴を保持するすべてのcopyおよびappendジョブのステータスを表示したりできます。

append statusコマンドの使用方法の詳細については、以下の例を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	ビューアー
発行対象:	ソースまたはターゲット
関連コマンド:	copystatus

構文

基本コマンド	コマンドオプション	リモートオプション
append status	[**job_id**]	[**on** *host* |
	[**Job_id.table_id**]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |
	[**full**]	**on** *login/password@host:portnumber*]
	[**detail**]	
	[**status**]	

構文の説明

コンポーネント	説明
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)` **>append status 28282**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)` **>append status 2828.HR.SRC_TEST3**
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)` **>append status all**
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了

コンポーネント	説明
	していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)`**>append status 2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)`**>append status detail**
[status]	指定されたステータスの過去のジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)`**>append status "Error"**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

認証レベル

authlevelコマンドを使用して、システム上でSharePlexコマンドを発行するための権限レベルを決定します。

以下に示すのは表示例です。

```
User is a viewer (level=3)
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	なし

構文

基本コマンド	リモートオプション
認証レベル	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモー

オプション	説明
	トシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

cancel

cancelコマンドは、実行中の**compare**、**repair**、**copy**、または**append**コマンドジョブをキャンセルするために使用します。

ジョブをキャンセルするには、キャンセルするジョブのジョブIDを指定する必要があります。ジョブIDは、compare、repair、copy、またはappend ジョブを開始するコマンドを発行したときに、そのジョブから返されます。

```
sp_ctrl> repair using 1elliot

repairing 7 of 7 objects

repair started; job id 408
```

または、**job status**コマンドを発行することで、直近にストリームされたジョブのジョブIDを取得することもできます。

```
sp_ctrl> job status

Job ID: 408

PID: 11527

Host: prodsys

Started: 22-FEB-15 18:08:09

Job Type: Repair

Status: Processing - 0 objects completed
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**compare**、**repair**、**copy**、**append**

構文

基本コマンド	リモートオプション
cancel *job_id*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
jobID	キャンセルするジョブのID。 **例**: `sp_ctrl>`**cancel 407**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

clear history

clean historyコマンドは、compare、repair、copy、appendコマンドの古いジョブから情報とログを削除するために使用します。

これらのジョブは、ソースシステムとターゲットシステムにログファイルを生成します。デフォルトでは、ジョブがSP_SYS_JOB_HISTORY_RETENTIONパラメータで設定された値より古くなると、ジョブ情報とログファイルが消去されます。**clear history**コマンドを使用すると、特定のジョブやテーブル、または特定の期間を経過したすべてのジョブについて、ジョブ情報とログをオンデマンドでクリアすることができます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**compare**、**repair**、**copy**、**append**

構文

基本コマンド	コマンドオプション	リモートオプション
clear history	[**for o.***source_sid*]	[**on** *host* |
{ **all** |		**on** *host:portnumber* |
source_owner.source_table |		**on** *login/password@host* |
age days |		**on** *login/password@host:portnumber*]
jobID }		

構文の説明

コンポーネント	説明
all	すべてのジョブを削除します。 **例**: `sp_ctrl(sysA)` > **clear history all**
source_owner.source_table	特定のテーブルの履歴を削除します。 **例**: `sp_ctrl(sysA)` > **clear history clear history user2.employee**
age days	指定した日数より古いジョブ履歴を削除します。

コンポーネント	説明
	例: `sp_ctrl(sysA)` **> clear history age 10**
jobID	指定されたジョブID（job statusコマンドで取得）の履歴を削除します。
for o.*source_sid*	オプション。アクティブな設定がない場合、または複数のアクティブな設定がある状況で**clear history**コマンドを使用するために使用できます。いずれの場合も、forオプションを使用してソースSIDを指定する必要があります。 **例**: `sp_ctrl(sysA)` **>clear history all for o.source_sid1**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>` **status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>` **status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>` **status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>` **status on john/spot5489@SysA:8304**

clear status

ステータスデータベースから古い警告メッセージを削除するには、**clear status**コマンドを使用します。

このコマンドを使用するには:

1. **show statusdb detail**コマンドを実行して、消去できるメッセージを見つけます。消去可能なメッセージは、**Clear**列に`Yes`が指定されています。
2. 一部のメッセージだけ消去したい場合は、消去する各メッセージのステータスIDをメモしておきます。
3. ステータスIDごとに**clear status**コマンドを発行するか、**all**引数を使用して、消去可能なすべてのメッセージを一度に削除します。

SharePlexは、消去されたメッセージを識別するメッセージをイベントログに書き込みます。

このコマンドは、デフォルトシステムのステータスデータベースからメッセージを消去します。別のシステムのステータスデータベースからメッセージを消去するには、[**on** *host*]オプションを使用します。

注: すべてのメッセージが消去できるわけではありません。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**show statusdb**

構文

基本コマンド	リモートオプション
clear status {*statusID*| **all**}	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
statusID	削除したい個々のメッセージのSharePlex割り当てID(show statusdbコマンドを使用して取得)。 **例**:

コンポーネント	説明
	`sp_ctrl(sysA)`**>clear status 20**
all	この引数を使用すると、消去可能なメッセージがすべて削除されます。 **例**: `sp_ctrl(sysA)`**>clear status all**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

compare / compare using

compareおよび**compare using**コマンド（まとめて*compare*コマンドと呼びます）を使用して、ターゲットテーブルがソーステーブルと同期していることを確認します。

- **compare**コマンドは、ワイルドカードを使用して、個々のソーステーブルとターゲットテーブル、または指定されたスキーマ内の任意の数のテーブルをcompareします。個々のテーブルをcompareする場合は、列ベースのフィルタリングを使用して、compare用に選択される行を制御することができます。
- **compare using**コマンドは、アクティブな設定にリストされているすべてのテーブル、またはアクティブな設定内のテーブルのサブセットが含まれる別のファイルにリストされているすべてのテーブルをcompareします。

compareを実行した後、**repair**コマンドまたは**repair using**コマンドを実行し、同期していない行をrepairすることができます。

注意： 実行中のcompareまたはrepairは、ソーステーブルには一切影響を与えません。SharePlexは、読み取りの一貫性についてのクエリを実行するためだけにデータベースにログインし、ソーステーブルのロックは短時間です。SharePlexは、処理中にターゲットテーブルを短時間ロックしますが、ユーザはロックをほとんど意識することなくアクセスを継続できます。

サポートされている操作

SharePlexは、**DML**操作（INSERT、UPDATE、DELETE）によって発生した、ターゲットテーブル内の非同期行を検出し、repairすることができます。

サポートなし

- **SharePlex は、以下のcompareとrepairをサポートせずにスキップします。**
    - 高度なデータ型
        - TEXT
        - CHAR > 2,000
        - VARCHAR > 4000
        - サイズなしのVARCHAR
        - BYTEA
    - シーケンス
    - トランスフォーメーションが使用されたテーブル
    - ソースサーバとターゲットサーバ間のデータ受け渡しにパススルーサーバを使用するネットワーク構成
- カスケードレプリケーション環境では、compareとrepairはサポートされません。

- compareまたはrepair中のテーブルに対してDDLを実行しないでください。compareは、SharePlexがサポートするものも含め、DDL操作によって引き起こされる非同期状態を検出しません。DDLがテーブル定義を変更すると、compareする必要がある行を取得するためにcompareプロセスにより作成されたSELECTステートメントが無効になります。

  DDLによって発生した非同期状態を修正したら、**repair**コマンドを使用して行のデータを再同期することができます。

- 255文字を超えるcompareおよびrepairコマンド文字列はサポートされていません。これはオペレーティングシステムの制限です。この制限を回避するには、ソースシステムで**edit**コマンドを使用します。テキストファイル内にコマンド文字列を入力すると、コマンドが自動的にファイルを実行します。

compareおよび**compare using**でサポートされるデータ型の詳細については、『SharePlexリリースノート』を参照してください。

その他の条件

- compareまたはrepairするテーブルは、アクティブな設定ファイルの一部でなければなりません。
- ソーステーブル上にコミットされていないトランザクションがあると、読み取りの一貫性を得るためにcompareおよびrepair処理に必要な短時間のロックを取得できなくなります。compareやrepairを実行する前に、すべてのトランザクションがコミットされていることを確認してください。
- レプリケーションのレイテンシは、compareやrepairの処理のパフォーマンスを低下させる可能性があります。ターゲット上でcompareおよびrepairプロセスを生成するためにソースから送信されたメッセージは、複製されたデータと共にキューを通じて送信されます。データバックログによる遅延が生じると、生成メッセージの送信が遅れることにより、プロセスが読み取りの一貫性を失うことがあります。可能であれば、compareやrepairはピーク以外の時間帯に行います。
- compareまたはrepair中のテーブルをトランケートしないでください。compareコマンドは、開始時に各ソーステーブルのスナップショットを取ります。テーブルがトランケートされると、スナップショットのテーブルビューもトランケートされ、コマンドが無効な非同期状態を返すことがあります。
- ビューをrepairするには、以下が真でなければなりません。
  - ビューテーブルが、基礎となるテーブルのすべての制約列を含んでいること。
  - ビューの制約列の名前が、基礎となるテーブルの制約列の名前と同一であること。
- Compareプロセスのパフォーマンスを向上させるために、SharePlexはSP_DEQ_PARALLELISMパラメーターによる並列ヒントをサポートしています。このため、並列度、つまり起動するワーカープロセスの数を指定できます。

  デフォルトでは、PostgreSQLデータベースオプティマイザはSQLステートメントに最適な問い合わせ実行計画を選択します。これは、並列ヒントで指定した並列度に直接関連していないこともあります。pg_hint_planにより、クエリを実行するたびに配列ヒントを使用して実行計画を調整することができます。SharePlexは、データベースにpg_hint_plan拡張機能がインストールまたは設定されている場合、内部でこれを利用します。

  並列クエリを使用するための構文の例は以下の通りです。

  pg_hint_plan拡張機能は並列ヒントをサポートしています。`Parallel(table <# of workers> [soft|hard])`

  SharePlexはハードパースをサポートする並列ヒントを使用します。`Parallel(table <# of workers> [hard])`

  > **注意:** ワーカーの数は、データベースシステムのCPU VCoreより少なくなければなりません。

特別な使用例

以下のシナリオでは、compareを実行する際に特別な処理が必要となります。

使用例	サポートcompare
統合レプリケーション	統合レプリケーションは、ターゲットデータベースとPostプロセスがソースホストのIDを各行に追加するように設定されている場合にサポートされます。中央のターゲットテーブルの正しい行をcompareまたはrepairするには、**targetwhere**オプションを使用し、ソースID値を**where**句の基準にします。 例えば、ある企業の東部本社のデータベースのテーブルと、中央の企業データベースの正しい行をcompareするには、東部データベースのソースIDに「East」を使用し、その値を**targetwhere**句の基準にすることができます。**repair**コマンドで同じ**targetwhere**句を使用します。compareプロセスおよびrepairプロセスは、ソースID値を使用して、東部データベースに有効な行のみを選択します。 ソースIDを特定する統合レプリケーション以外の実装でcompareコマンドやrepairコマンドを使用すると、ターゲット行が不要に削除される可能性があります。この設定の詳細については、『SharePlex管理ガイド』を参照してください。 ターゲット行が正確に選択されるように、**targetwhere**オプションと標準の**where**オプションを組み合わせる必要があるかもしれません。
ピアツーピアレプリケーション	ピアツーピア構成では、どのシステムを*信頼できるソースシステム*とし、どのシステムを*セカンダリシステム*、つまりターゲットシステムとするかを決めなければなりません。セカンダリシステムは、repairが行われるシステムです。ピアツーピア環境でcompareやrepairを実行する前に、以下の手順に従ってください。 1. セカンダリシステムのユーザアクセスを停止し、そのシステムから複製された操作が信頼できるソースデータベースにポストされるのを待ちます。ユーザは引き続きソースデータベースにアクセスできます。 2. ソースおよびセカンダリシステムで**qstatus**コマンドを実行します。 3. キュー内のメッセージが10個以下であれば、ソースシステムからcompareを実行します。 4. compare中、**sp_xdesvr**および**sp_xdeclt**の開始後に、ソースおよびセカンダリデータベースへのユーザアクセスを許可できます。 5. **where**オプションを指定して**repair**コマンドを使用すると、ユーザをテーブルからロックアウトすることなく、ターゲットテーブルの選択した行をrepairできます。 この設定の詳細については、『SharePlex管理ガイド』を参照してください。
キーのないテーブル	compareおよびrepairコマンドは、ソースおよびターゲットシステム上でORDER BY句付きのSELECTステートメントを発行します。大規模なテーブルにプライマリキーまたはNULL以外の一意キーとインデックス（一意のインデックスが望ましい）を持つ場合、順序付けはより速くなります。それ以外の場合は、すべての列がキーとして使用されます。 一意の行識別子はないが、一意の行として識別できる列がテーブルに1つ以上ある場合、**orderby**オプションを指定して**compare**コマンドを使用できます。このオプションを使用すると、SharePlexは、ソースシステム上の**sp_desvr**ログに、コマンドがこれらの列をキーとして使用したという通知を出力します。
余分なソース列またはターゲット列があるテーブル	ソーステーブルまたはターゲットテーブルに余分な列があり、それらの列にソートの基準となる一意の値が含まれている場合は、**sourcewhere**句または**targetwhere**句とともに**compare**コマンドを使用します。「compareする行の制御」を参照してください。

使用例	サポートcompare

XMLデータでのcompare操作	XMLデータに対してCompare操作を実行すると、「ORA-04036: PGA memory used by instance exceeds PGA_AGGREGATE_LIMIT(ORA-04036: インスタンスが使用するPGAメモリがPGA_AGGREGATE_LIMITを超えています)」エラーが表示されます。
	回避策:
	1. Oracleをバージョン19.19 DBRUにアップグレードします
	2. 提供されている以下のOracle推奨スクリプトを使用して、適切なPGA_AGGREGATE_LIMITを設定します
	`WITH`
	`MAX_PGA as`
	`(select round(value/1024/1024,1) max_pga from v$pgastat where name='maximum PGA allocated'),`
	`MGA_CURR as`
	`(select round(value/1024/1024,1) mga_curr from v$pgastat where name='MGA allocated (under PGA)'),`
	`MAX_UTIL as`
	`(select max_utilization as max_util from v$resource_limit where resource_name='processes')`
	`SELECT`
	`a.max_pga "Max PGA (MB)",`
	`b.mga_curr "Current MGA (MB)",`
	`c.max_util "Max # of processes",`
	`round(((a.max_pga - b.mga_curr) + (c.max_util * 5)) * 1.1, 1) "New PGA_AGGREGATE_LIMIT (MB)"`
	`FROM MAX_PGA a, MGA_CURR b, MAX_UTIL c`
	`WHERE 1 = 1;`
	3. SharePlexバージョン11.1にアップグレードするか、そのバージョンをインストールします。

compareの仕組み

compareは、DML操作によって引き起こされる次のようなターゲットテーブルの非同期状態を検出します。

- 余分な行または欠けている行
- 値が一致しない行

ソースシステムでcompareコマンドまたはcompare usingコマンドを実行すると、SharePlexは以下のイベントを開始します。

1. **sp_cop**プロセスは、ソースシステム上に**sp_desvr**（サーバー）プロセスを生成します。
2. **sp_desvr**プロセスは、**sp_ctrl**インターフェイスの制御と使用をユーザに返します。compareが進行している間、レプリケーションは続行されます。
3. **sp_desvr**プロセスは、ターゲットシステム上で**sp_declt**（クライアント）プロセスを開始するために、Postプロセスにメッセージを送信します。
4. サーバプロセスとクライアントプロセスによって相互の通信が確立されます。
5. プロセスは、ソーステーブルとターゲットテーブルの行数をcompareすることにより、*サニティチェック*を実行します。ソーステーブルとターゲットテーブルの行数の差が20%より大きい場合、コマンドは終了します。サニティチェックは、ターゲットデータを再読み込みすることでより迅速に修正できるテーブルのcompareやrepairに無駄な時間をかけることを防ぎます。**copy**コマンドまたは**append**コマンドを使用してターゲットデータを再読み込みするか、別の方法を使用することができます。サニティチェックを上書きするには、overrideオプションを指定してcompareコマンドを再発行します。
6. サニティチェックをオーバーライドするには、**Override**オプションを指定して**Compare**コマンドを再発行します。
7. 行数がサニティチェックに合格すると、テーブルは以下のようにcompareされます。
    - **compare**コマンドを使用している場合、**sp_desvr**でソース行を選択し、**sp_declt**でターゲット行を選択して、行をソートしてcompareします。
    - **compare using**コマンドを使用している場合、**sp_desvr**プロセスはターゲットシステム上に多数の処理スレッドを作成します。SP_DEQ_THREADSパラメーターによって設定される値は、作成されるスレッド数を制御します。各スレッドは、**sp_declt**（クライアント）プロセスを生成します。サーバプロセスとクライアントプロセスによって相互の通信が確立されます。処理負荷はクライアントプロセス間で分割されます。ソースとターゲットの各テーブルから行が選択され、ソートされ、compareされます。
8. 終了すると、プロセスによってログファイルが書き出され、**show compare**コマンドで結果を見ることができます。

SQLログファイルの管理

compareおよびrepairコマンドは、すべての非同期行をrepairするために必要なSQLを、ログファイルと同じ場所にあるSQLファイルに書き込みます。compareコマンドのみが発行された場合、SharePlexはこれらのSQLステートメントを実行しません。repairコマンドを実行した場合、このコマンドは、非同期行をrepairするSQLステートメントを実行すること以外はcompareコマンドと同じように動作します。

SQLログファイルの出力を抑制することができます。このファイルを抑制する理由には以下のようなものがあります。

- データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。
- compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。

SQLログファイルを抑制するには、**nosqllog**オプションを**compare**または**repair**コマンドで使用します。

SharePlexの現在のインスタンスの実行中に、compareとrepairのすべての実行についてSQLログファイルの出力を抑制するには、SP_SYS_SECURE_MODE環境変数を1に設定します。この変数はSharePlexを起動する前に設定する必要があるため、**sp_cop**プロセスが実行されている場合は、この変数を設定した後に再起動する必要があります。この環境変数を指定して**sp_cop**を実行すると、compareコマンドとrepairコマンドによってSQLファイルにデータは保存されず、Post処理でもSharePlexのエラーログにデータは保存されません。

複数のプロセスの実行

すべてのcompareおよびrepairコマンドでは、複数のプロセスを同時に実行することができます。複数の**compare**および**repair**コマンドを同時に実行し、それぞれがソーステーブルとターゲットテーブルのペアを処理することも、PostgreSQLのワイルドカードを使用して1つのコマンドで複数のテーブルセットを指定することもできます。SharePlexによるワイルドカードのサポートの詳細については、『SharePlex管理ガイド』を参照してください。

レプリケーションプロセスとcompareおよびrepairプロセスなど、最大で20のSharePlexプロセスでpostキューを同時に使用することができます。一度に最大で5つのcompareおよびrepairプロセスを実行できるようにすることをお勧めします。

制限に達したためにcompareやrepairに失敗した場合、SharePlexはイベントログにメッセージを記録します。

注意: **edit**コマンドを使用し、以前のコマンドを編集して新しいコマンドを作成すれば、複数のコマンドをより簡単に実行できます。

設定のサブセットのcompare

アクティブな設定のサブセットは、以下の方法でcompareできます。

- 1つのスキーマに属するレプリケーション内の全テーブルをcompareするには、以下のように**compare**コマンドにワイルドカードを指定します。

  ```
  sp_ctrl> compare scott.%
  ```

- 設定ファイル内の全テーブルをcompareするには、**compare using**コマンドを使用します。

  ```
  sp_ctrl> compare using myconfig
  ```

- レプリケーション内の全テーブルを1つのターゲットルートとcompareするには、以下のように**at**オプションを指定して**compare using**コマンドを使用します。

  ```
  sp_ctrl> compare using config.active at prodsys@o.ora112
  ```

- 設定内のテーブルのカスタムサブセットをcompareするには、*compareファイル*で指定します。これはプレーン・テキスト・ファイルであり、compareしたいソーステーブルのみをリストします。ターゲットテーブルは、compare時に設定ファイルから取得されます。compareファイルは、**create config**コマンドまたは**copy config**コマンドを使用して作成できます。このファイルには、必ず設定ファイルでないことを明確に示す名前を付けてください。詳細については、コマンドの構文を参照してください。

compareする行の制御

compareおよび**repair**コマンドには、処理の対象として選択する行をフィルタリングできる**where**オプションがあります。デフォルトでは、これらのコマンドはテーブルのすべての行に影響し、ソーステーブルに含まれていないターゲットテーブルの列は無視されます。

- **where**オプションを使用すると、ソーステーブルとターゲットテーブルの同じ名前の列に基づいて行をフィルタリングすることができます。
- 垂直分割レプリケーションを使用するテーブルには**where**オプションを使用します。ソース列とターゲット列には異なる名前を付けることができます。**where**の選択をソース列に基づいて行います。SharePlexは、ターゲットテーブルの正しいWHERE句を構築するために、設定ファイルから列マッピングを読み取ります。
- ソーステーブルまたはターゲットテーブルに1つ以上の余分な列が存在し、それらの行に行の一意性を決定する値が含まれている場合は、**sourcewhere**および**targetwhere**オプションを使用します。
  - ソーステーブルに余分な列が含まれている場合は、**sourcewhere**を使用します。
  - ターゲットテーブルに余分な列が含まれている場合は、**targetwhere**を使用します。

  このオプションを正しく使用するには:

  - 余分な列には**sourcewhere**または**targetwhere**オプション*のみ*を使用します。
  - ソースとターゲットの両方で同じ名前を持つ他の列には、標準の**where**オプションを使用します。
  - SharePlexは、**where**オプションと**sourcewhere**または**targetwhere**オプションを組み合わせて、完全なWHERE句を作成します。

> **重要!** 余分な行を持つターゲットテーブルに対してcompareとrepairの両方を実行する予定がある場合は、UPDATEとDELETEのcompare用に**targetwhere**のみを使用してください。**repair**コマンドはINSERTの正しい値を決定できません。この問題を回避するには、余分な列にデフォルト値を設定するか、挿入された行を手動で更新します。

プロセスの特定

compareまたはrepairコマンドが発行されるたびに、ジョブIDが**sp_ctrl**ディスプレイに表示されます。**sp_ctrl**ディスプレイを利用できない場合は、**compare status**コマンドを実行することでジョブIDを表示できます。

sp_ctrlでステータスと結果を表示

compareのステータスや結果を表示するには、**sp_ctrl**の**compare status**コマンドを使用します。

- 基本コマンドは、最近開始されたcompareジョブの処理ステータスと、まだ実行中のその他のcompareジョブを表示します。
- 追加オプションを使用して、履歴のあるcompareジョブのステータス概要を表示したり、compareジョブの詳細情報を表示したりすることができます。

詳細については、compare status を参照してください。

警告とエラーの表示

sp_xdesvrおよび**sp_xdecltプロセス**は、実行するシステム上にログファイルを書き込みます。SharePlexの変数データディレクトリの**log**サブディレクトリにあるファイルを開きます。

sp_xdesvrプロセスが書き込むログの名前は`xdesvr_<jobid>_r.<dbid>_p<process id>.log`です。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブIDです。
- *SID*は、ソーステーブルがあるOracleインスタンスのORACLE_SIDです。
- *ProcessID*は**sp_xdesvr**プロセスのプロセスIDです。

sp_xdecltプロセスが書き込むファイル名は、`xdeclt_<jobid>-<tableid>_r.<dbid>_p<processid>.log`に**log**または**.sql**を付加したものです。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブのジョブIDです。
- *TableID*は、SharePlexによって割り当てられたジョブ内のテーブルのテーブルIDです。
- *SID*は、ソーステーブルがあるOracleインスタンスのORACLE_SIDです。
- *SourceHost*は、ソースホストの名前またはIPアドレスです。
- *ProcessID*は、**sp_decltプロセス**のプロセスIDです。

ログファイル名の例:

```
xdesvr_7_r.aparopka_p4970.log

xdeclt_7-1_r.aparopka_p25095.log

xdeclt_7-1_r.aparopka_p25095_01.sql
```

ディスク使用量を制御するために、ログは循環方式でエージングされます。現在のログがサイズの限界に達すると、SharePlexは新しいログファイルを生成します。新しいログは最大ログ数まで作成され、その後、SharePlexは最も古いログから上書きを開始します。

注意: **compare using**コマンドの場合は、稀に複数のログファイルが存在します。

compareジョブのキャンセル

実行中のcompareまたはrepairジョブを停止するには、**cancel**コマンドを使用します。

```
sp_ctrl(sysA)>cancel JOBID
```

詳細については、「cancel」を参照してください。

compare履歴とログの管理

SharePlexは、終了した各ジョブの履歴をソースシステムのデータベースに保持します。SP_SYS_JOB_HISTORY_RETENTIONパラメーターは、履歴の保持期間を制御します。

この履歴を必要に応じて消去するには、**clear history**コマンドを使用します。SharePlexがジョブの履歴を削除すると、履歴の元となったログファイルも削除されます。

データベースのジョブ履歴を消去せずにソースシステムからログファイルを削除するには、**remove log**コマンドを使用します。このコマンドは、ターゲットシステムから古いログファイルを削除するためにも使用できます。

ログファイルのサイズを制御するには、SP_DEQ_LOG_FILESIZEパラメーターを設定します。

INSERT、UPDATE、DELETEの各操作に対して別々のSQLファイルを作成するには、すべての操作タイプに対して1つのファイルを使用するのではなく、**log split**オプションを使用します。

バッチサイズのコントロール

プロセスがSELECTクエリを実行するときにフェッチされる行のブロックのサイズを制御することができます。ブロックサイズは、**SP_DEQ_MALLOC**パラメーターで設定された値に基づいて計算されます。この値は、使用されるcompareスレッドの数で等分され、その後、すべての列のサイズを足した値に基づいて再計算されます。

非同期行のrepair

compareによって検出された非同期行をrepairするには、**repair**コマンドまたは**repair using**コマンドを使用します。repair / repair usingを参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	repair / repair using

構文

構文の説明

必須のコマンドコンポーネント

コンポーネント	説明
compare *owner.source_table[.partition]*	基本コマンドは、すべてのソース行をすべてのターゲット行とcompareします。 *owner.source_table*は、ソーステーブルの所有者と名前です。大文字と小文字を区別したり、名前の中にスペースを入れたりするには、二重引用符を使います。例えば、**"HR".emp**。 ワイルドカードのテーブル名（オーナー名は不可）がサポートされています。compareするためには、このコマンドのワイルドカードを満たすテーブルがアクティブなレプリケーション設定に（明示的またはワイルドカードで）リストされていなければなりません。SharePlexがワイルドカードをどのように処理するかについては、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl(sysA)>`**compare scott.emp** `sp_ctrl(sysA)>`**compare scott.emp.west**
compare using *filename*	基本コマンドは、*filename*にリストされたテーブル内のすべてのソース行とすべてのターゲット行をcompareします。 *filename*は、compareするソーステーブルの名前が含まれているファイルの名前です。 **例** `sp_ctrl(sysA)>`**compare using sales**

オプションのコマンドコンポーネント

コンポーネント	説明
at *target_host*@*o.target_sid*	**compareに有効** ソーステーブルをそのターゲットの1つとのみcompareします。ソーステーブルが複数のターゲットシステムにレプリケートされる場合に使用します。 *target_host*はターゲットシステムの名前です。 *target_sid*はターゲットOracleインスタンスのORACLE_SIDです。 **例** `sp_ctrl(SysA)>`**compare scott.emp at prod@o.prodsid**
for o.*SID*	**compareに有効**

コンポーネント	説明
	ソーステーブルを含むOracleインスタンスを指定します。システム上の複数のOracleインスタンスに同じソーステーブルがある場合に使用します。 *SID*は、ソースインスタンスのORACLE_SIDです。大文字と小文字は区別され、**oratab**ファイルまたはV$PARAMETERテーブルに表示される通りに入力する必要があります。 このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションとともに任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**compare scott.emp for o.oraA**
hint "*hint*"	**compareに有効** SELECTステートメントにOracleヒントを含めます。ヒントはソースおよびターゲットシステム上で使用されます。 "*hint*"は2000文字以内のOracle標準のヒントです。ヒント全体を二重引用符で囲みます。ヒント文字列の先頭の/*+と末尾の*/を省略します。SharePlexによって追加されます。 このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションとともに任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**compare scott.emp where "file >001005" hint "emp(salary)"** オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列には、以下の例のようにエスケープされた二重引用符の余分なセットが必要です。 */productdir/***bin/sp_ctrl compare scott.emp hint** "\"**emp (salary)**\""
{**include** | **exclude**} "(*column_list*"	**compareに有効** compareする列をフィルタリングします。 • **include**を使って、compareしたい列を指定します。他の列はcompareされません。**include**句には、すべてのキー列を含める必要があります。 • **exclude**を使用して、**exclude**で指定した列を除くすべての列をcompareします。キー列を除外しないでください。 (*column_list*)は、インクルードまたは除外する列のリストです。

コンポーネント	説明
	• それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションとともに任意の順序で記述できます。 **注意:** compareされなかった列には、同期していない行がまだ残っている可能性があります。 **例** `sp_ctrl (SysA)>`**compare scott.emp exclude "color, weight"**
key	**compareおよびcompare usingに有効** 大きなテーブルのcompareを高速で実行します。このコマンドはすべてのデータ値をcompareするわけではありません。以下のいずれか1つをcompareします。 • PRIMARYキー列またはNULLでないUNIQUEキー列のみ。 または... • **orderby**オプションで指定された列。テーブルにキーがない場合は、このオプションを使用します。 **重要:** キーまたは**orderby**の値が一致しても、他の列の値が一致しなければ、テーブルは同期しないままです。 このオプションを使用する場合は、必須コマンド引数の後に記述しなければなりません。他のオプションとともに任意の順序で表示できます。 このオプションは、*SharePlex*キー定義に基づくcompareには使用しないでください。SharePlexキー定義の詳細については、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl (SysA)>`**compare scott.emp key** `sp_ctrl(sysA)>`**compare using sales key**
log rowdata	**compareおよびcompare usingに有効**

コンポーネント	説明
	ハッシュ値ではなく、実際の行データを記録するSQLファイルを生成するようにクライアントプロセスに指示します。ファイルはターゲットシステムに作成されます。compareするテーブルにLONG、LOB、VARRAYがある場合は、**log rowdata**を使用しないでください。これらのデータ型の値が非常に大きい場合、SharePlexは実際のデータを記録できない可能性があります。 **重要!** このSQLファイルの目的は、compareされたデータのビューを提供することです。ターゲットテーブルをrepairするために使用しないでください。コマンド実行時に取得されたデータ値は、現在データベースに登録されている値とは異なる可能性があります。非同期の行をrepairするには、**repair**コマンドを使用します。 **例** `sp_ctrl(sysA)>`**compare scott.emp log rowdata** `sp_ctrl(sysA)>`**compare using sales log rowdata**
log split	**compareおよびcompare usingに有効** SQLファイルを操作の種類に応じて3つの異なるファイル(INSERT用、UPDATE用、DELETE用)に分割するように、クライアントプロセスに指示します。 **例** `sp_ctrl(sysA)>`**compare scott.emp log split** `sp_ctrl(sysA)>`**compare using sales log split**
{**nolocktarget** | **nolocksource**}	**compareに有効** repairを含む実行のcompareフェーズ中に、compareプロセスがテーブルをロックするのを防ぎます。通常、SharePlexはcompare中に一時的にテーブルをロックして読み取り一貫性ビューを取得し、その後すぐにロックを解除しますが、repair中は、SharePlexは常にターゲットテーブルをロックします。SharePlexは、読み取り一貫性ビューを取得するために、repair中はソーステーブルを一時的にロックします。
nosqllog	SQLログファイルの出力を抑制します。このファイルには、同期していない行をrepairするために必要なSQLが含まれています。このファイルを出力しない理由には、以下のようなものがあります。 • データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。

コンポーネント	説明
	• compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。
not “*exception_list*”	**compareに有効** テーブル指定にワイルドカードが含まれている場合に、compareしないテーブルの例外リストを指定します。 "*exception_list*"は、compareしないテーブル名のリストです。 • *owner.tablename*形式を使用します。 • それぞれの名前はカンマで区切ります。リストにスペースは許容されません。 • リストを二重引用符で囲みます。 • テーブルを任意の順序でリストします。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションとともに任意の順序で記述できます。 **例** `sp_ctrl(SysA)>`**compare scott.% not (%temp%)**
orderby "*column_list*"	**compareに有効** compareプロセスでcompareする行をソートする際に、ORDERBY句で使用する列を指定します。このオプションを使用すると、プライマリキーや一意キーを持たないテーブルでcompareを実行できます。 "*column_list*"は、ORDERBY句で使用する列の名前です。 • それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションとともに任意の順序で記述できます。 **例** `sp_ctrl(SysA)>`**compare scott.emp where “file >001005” orderby “Last Name,Division”** オペレーティングシステムのコマンドラインからcompareを実行する

コンポーネント	説明
	場合、引用符で囲まれた文字列には、以下のようにエスケープされた二重引用符の余分なセットが必要です。 */productdir/***bin/sp_ctrl compare scott.emp orderby** “\“**Last Name**,**Division**\””**
override	**compareおよびcompare usingに有効** サニティチェックを無効にし、サニティチェックのしきい値を超えている、同期していないテーブルのcompareを可能にします。同期していないテーブルをcompareする代わりに、**copy**コマンドを使用して（copy / appendを参照）ターゲットテーブルを再読み込みすることもできます。サニティチェックの詳細については、「compareの仕組み」を参照してください。 **例:** SP_ctrl (sysA) > **compare scott.emp override** SP_ctrl (sysA) > **compare using sales override**
port *port_number*	compareおよびcompare **using**に有効 ソースまたはターゲットシステム上のSharePlexのバージョンが8.0より前の場合、下位互換性のために使用できます。 クライアントプロセスがサーバプロセスとの通信に使用する、ソースシステム上のポートを指定します。SharePlexの以前のバージョンでは、通信は双方向であり、クライアントからサーバへの通信には、デフォルトでランダムなポート番号が選択されます。このオプションは、ファイアウォールが要求するような特定のポート番号で、ランダムなポート選択を上書きします。 **例** `sp_ctrl(sysA)>`**compare scott.emp port 1234**
quickcheck	**compareおよびcompare usingに有効** 同期していない行があるかどうかを確認する初期チェックを実行します。同期していない行が1つでも検出されると、プロセスは直ちに停止します。それ以上の情報は返されません。同期していない行の情報はcompareのSQLファイルに記録されません。 このオプションは、**key**オプションや、**where**オプションのいずれかと併用しないでください。 このオプションはLONG列をサポートしていません。 **例** `sp_ctrl(sysA)>`**compare scott.emp quickcheck** `sp_ctrl(sysA)>`**compare using sales quickcheck**

コンポーネント	説明
parallelism*degree*	**compareおよびcompare usingに有効** SELECTステートメントに並列ヒントを追加します。*degree*では、並列度を設定します。 **例** `sp_ctrl(sysA)>`**compare scott.emp parallelism 4** `sp_ctrl(sysA)>`**compare using sales parallelism 4**
sourcewhere “*clause*”	**compareに有効** ターゲットテーブルに列が存在しない場合に、ソーステーブル内の1つまたは複数の列を基にcompareを行います。 • *clause*を二重引用符で囲み、**scott.emp**のように完全修飾名でテーブルを参照します。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、OracleのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列は、以下の例のようにエスケープされた二重引用符で囲む必要があります。 `sp_ctrl>`**compare scott.emp sourcewhere “\“file >001005\””** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションとともに任意の順序で記述できます。 **例#1**: `sp_ctrl(sysA)>`**compare scott.emp sourcewhere “file >001005”** **例#2**: 次の例は、**sourcewhere**および**where**オプションをどのように組み合わせて、望ましい結果を得るかを示しています。**sourcewhere**句はソースcompareプロセスのみが使用しますが、**where**句はソースcompareプロセスとターゲットcompareプロセスの両方が使用します。 `sp_ctrl(SysA)>`**compare scott.emp sourcewhere**

コンポーネント	説明
	“deptno = 200” where “mgr = ‘SMITH’”
targetwhere "*clause*"	**compareに有効** ソーステーブルに列が存在しない場合に、ターゲットテーブル内の1つまたは複数の列を基にcompareを行います。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、OracleのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列には、以下のようにエスケープされた二重引用符の余分なセットが必要です。 */productdir*/**bin/sp_ctrl compare scott.emp targetwhere “\“file >001005\””** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションとともに任意の順序で記述できます。 **例#1**: `sp_ctrl(SysA)>` **compare scott.emp targetwhere “file >001005”** **例#2**: 次の例は、**targetwhere**および**where**オプションをどのように組み合わせれば、望ましい結果が得られるかを示しています。**targetwhere**句はターゲットcompareプロセスのみが使用しますが、**where**句はソースcompareプロセスとターゲットcompareプロセスの両方が使用します。 `sp_ctrl(SysA)>`**compare scott.emp where “deptno = 200” targetwhere “mgr = ‘SMITH’” repair**
threads *thread_count*	**compareおよびcompare usingに有効** compareプロセスが使用する処理スレッドの数を設定します。 **例** `sp_ctrl(sysA)>`**compare scott.emp threads 4**

コンポーネント	説明
	`sp_ctrl(sysA)>`**compare using sales threads 4**
to *target_owner.target_table*[*.partition*]	**compareに有効** ソーステーブルをそのターゲットの1つとのみcompareします。ソーステーブルが複数のターゲットシステムにレプリケートされ、ターゲットテーブルに異なる名前がある場合に使用します。 このオプションはターゲットパーティションを指定するためにも使用できます。 `compare source_owner.source_table.[source_partition] to target_owner.target_table.[target_partition]` **例** （パーティションのcompare） `sp_ctrl(SysA)>`**compare scott.emp.east to scott.allemp.alleast**
where “*clause*”	**compareに有効** ソースシステムとターゲットシステムの両方で、SELECTステートメントにWHERE句を含めます。WHERE句は、特定の行をcompareするためのフィルタとして機能します。 “*clause*"には、サブクエリを含まない標準のWHERE句を指定します。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、OracleのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションとともに任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**compare scott.emp where “region=4”**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

compare status

compare statusコマンドを使用して、最後に実行されたcompareまたはrepairジョブのステータスを表示します。**compare statusコマンド**を使用すると、compareジョブやrepairジョブ、またはcompareジョブやrepairジョブの一部に関する詳細なステータスを表示したり、SharePlexに履歴があるすべてのcompareジョブおよびrepairジョブのステータスを表示したりできます。

注意: **compare statusコマンド**の詳細および使用例については、**job statusコマンド**を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	ビューアー(3)
発行場所:	ソースシステム
関連コマンド:	compare / compare using、repair / repair using

構文

基本コマンド	コマンドオプション	リモートオプション
job status	[**job_id**]	[**on** *host* |
	[**Job_id.table_id**]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |
	[**full**]	**on** *login/password@host:portnumber*]
	[**detail**]	
	[**status**]	

構文の説明

コンポーネント	説明
job status	SharePlexが履歴を保持するすべてのcompareジョブおよびrepairジョブのステータスを表示します。
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)>`**job status 2828.2**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。

コンポーネント	説明
	例: `sp_ctrl(sysA)`**>job status 2828.HR.SRC_TEST3**
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)`**>job status all**
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)`**>job status 2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)`**>job status detail**
status	指定されたステータスの過去のジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)`**>job status "Error"**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモー

オプション	説明
	トシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

connection

connectionコマンドを使用すると、SharePlexをデータベースに接続するように設定できます。

connectionコマンドの使用方法

connectionコマンドを使用して、データベースに接続する際にSharePlexが使用する接続プロパティを設定します。このコマンドは、データベースで使用できるデータベース・セットアップ・ユーティリティがない場合にのみ使用します。データベースのセットアップユーティリティが存在するかどうかを確認するには、「データベース・セットアップ・ユーティリティ 」の581ページを参照してください。

connectionコマンドは以下のように使用します。

- **set**オプションを使用して接続属性を設定します。使用するオプションごとに、**connection**コマンドを1つ発行します。このコマンドでは、1つのコマンドで複数のキーワード/値のペアを文字列化することができません。
- **reset**オプションを使用して接続設定をクリアします。
- **show**オプションを指定して現在の接続設定を表示します。

注意:

- SharePlexがサポートするOpen Targetターゲットは、SharePlexリリースノートに記載されています。
- ODBCデータベースのDSNが存在する場合、**dsn**、**user**、および**password**キーワードのみが必須の接続設定です。DSNが存在しない場合、**user**、**password**、**server**、**driver**、および**port**キーワードがすべて必須です。
- **connect_string**キーワードは、推奨される場合、または提供されたODBCキーワードが十分でない場合に、ユーザ定義のODBC接続文字列をサポートします。

重要: このコマンドを使用した後は、必ずPostプロセスを停止してから再度開始してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースまたはターゲットシステム
関連コマンド:	**target**

構文

基本コマンド	キーワード=値オプション	リモートオプション
connection {**o.** | **r.**}*database*	[**user**=*username*]	利用不可
{	[**password**=*password*]	
set *keyword*=*value* |	[**tns_alias**=*alias*]	

基本コマンド	キーワード=値オプション	リモートオプション
show [**all**] |	[**asm_sid**=*SID*]	
reset [*keyword*]	[**asm_user**=*username*]	
}	[**asm_password**=*password*]	
	[**asm_tns_alias**=*alias*]	
	[**wallet_location** = path]	
	[**wallet_automode** = Y/N]	
	[**dsn**=*DSN*]	
	[**server**=*servername*]	
	[**driver**=*path*]	
	[**port**=*portnumber*]	
	[**connect_string**=*string*]	
	[**plugin** = pluginname]	
	[**plugin_version** = versionnumber]	
	[**plugin_direction** = source/target/both]	
	[**sp_host_port** = Hostname:port]	
	[**admin_user** = username]	
	[**database** = SID]	
	[**dstype** = Datastorename]	
	[**version** = versionnumber]	
	[**hostaddr** = IPaddress]	
	[**host** = servername]	
	[**dbname** = databasename]	
	[**db_host** = IP]	
	[**db_port** = portnumber]	
	[**oracle_home** = path]	
	[**service_name** = servicename]	

構文の説明

コンポーネント	説明
{**o.** | **r.**}*database*	接続を設定するデータベース。以下のフォーマットを使用します。 Oracle:

コンポーネント	説明
	o.*SID* ここで、*SID*はデータベースのORACLE_SIDです。 Open Target（ODBC）: **r.***database* ここで、*database*はデータベースの名前です（DSNではない）。 **注意:** データベースの指定は、設定ファイルのデータソース行またはルーティングマップのデータベース指定と一致している必要があります。例えば、ルーティングマップが**myhost@r.mydb**の場合、**connection**コマンドには、**connection r.mydb set user=myuser**のように同じデータベース指定が含まれます。
set *keyword=value*	接続プロパティを設定します。キーワードを参照してください。
show [**all**]	**show**は、指定されたデータベースの現在の接続設定を表示します。 **show all**は、すべてのローカルデータベースの接続設定を表示します。以下の例のように、*database*指定を**show all**と一緒に使用しないでください。 **connection show all**
reset [*keyword*]	**reset**は、指定されたデータベースのすべての接続設定をクリアします。 **reset**キーワードは、特定の接続プロパティの接続設定のみをクリアします。キーワードを参照してください

キーワード

キーワード	データベースのタイプ	入力値の説明
user=*username*	すべて	*database*で指定されたデータベースに接続するためにSharePlexが使用するデータベースユーザの名前。
password=*password*	すべて	*username*で指定されたユーザのパスワード。
tns_alias=*alias*	Oracle	SharePlexが接続するTNSエイリアス。データベースへの接続を**tnsnames.ora**ファイルで管理する場合は必須です。
asm_sid=*SID*	Oracle	ASMインスタンスのOracle SID。Oracle Automatic Storage Managementがデータベースに使用されている場合は必要です。
asm_user=*username*	Oracle	SharePlexがASMインスタンスに接続するために使用するASMユーザの名前。
asm_password=*password*	Oracle	ASMユーザのパスワード。
asm_tns_alias=*alias*	Oracle	ASMインスタンスのTNSエイリアス。ASMインスタンスへの接続を**tnsnames.ora**ファイルで管理する場合は必須です。
wallet_location = path	Oracle	設定されている場合、Oracleのウォレットファイルへのパス

キーワード	データベースのタイプ	入力値の説明
wallet_automode = Y/N	Oracle	SharePlexの自動オープンを設定する場合はYに設定
dsn=*DSN*	Open Target（ODBC）	データベースに関連付けられているデータソース名（DSN）。データベースのDSNが存在する場合、このキーワードと、**user**および**password**キーワードのみが必須の接続設定です。
server=*servername*	Open Target（ODBC）	データベースをホストするサーバの名前またはIPアドレス。
driver=*path*	Open Target（ODBC）	データベースへのODBC接続をサポートするODBCドライバのフルパス。
port=*portnumber*	Open Target（ODBC）	データベースのポート番号。
connect_string=*string*	Open Target（ODBC）	ユーザ定義の接続文字列。独自の接続文字列を使用する場合は、ODBC接続を成功させるために必要な要素がすべて含まれていることを確認してください。*ただし*、ユーザ名とパスワードは省略してください。**user**および**password**オプションを指定して**connection**コマンドを使用し、ユーザ情報を提供します。
plugin = pluginname		プラグインの名前
plugin_version = versionnumber		プラグインバージョン
plugin_direction = source/target/both		plugin_directionの値は、source、target、またはbothに設定できます。
sp_host_port = Hostname:port	データベース（Oracle以外）	SharePlexホスト：セットアップを実行したポート
admin_user = username		管理者ユーザのSQL Server
database = SID		データベース名
dstype = Datastorename		データストア名
version = versionnumber		
hostaddr = IPaddress	データベース（Oracle以外）	ODBCが接続するサーバのIPアドレス
host = servername	データベース（Oracle以外）	ODBCが接続するサーバ名
dbname = databasename	データベース（Oracle以外）	ODBCが接続するデータベース名
db_host = IP	データベース（Oracle以外）	DSN経由のデータベースのホスト
db_port = portnumber	データベース（Oracle以外）	DSNを通じてデータベースが使用するポート
oracle_home = path	Oracle	Oracle Homeへのパス
service_name = servicename	Oracle	Oracleサービス名

例

Oracleの例

```
connection o.myora12 set user=myuser

connection o.myora12 set password=mypassword

connection o.myora12 set tns_alias=myora12
```

Open Targetの例

DSNが存在する場合

```
connection r.mydb set user=myuser

connection r.mydb set password=mypassword

connection r.mydb set dsn=mydsn
```

DSNが存在しない場合

```
connection r.mydb set user=myuser

connection r.mydb set password=mypassword

connection r.mydb set port=1234

connection r.mydb set server=server1

connection r.mydb set driver=/database/ODBC/lib/databasedriver.so
```

DSNが存在しないため、接続文字列を使用

```
connection r.mydb set user=myuser

connection r.mydb set password=mypassword

connection r.mydb set connect_
string=”driver=/database/ODBC/lib/databasedriver.so;server=server1;port=1234;uid=myuser;pwd
=mypassword”
```

接続値の削除

```
connection r.mydb reset port

connection r.mydb reset
```

接続設定の表示

```
connection r.mydb show

connection show all
```

copy / append

copy/appendコマンドは、Oracle Export/Importユーティリティを使用して、ユーザがレプリケーションをセットアップしたり、既にレプリケーション内にある個々のテーブルを(再)同期したりできるようにします。

ユーザは、個々のテーブルを単一のターゲットに**コピー/追加**したり、複数のターゲットを指定したり、レプリケーション設定内のすべてのターゲットを指定したりできます。

重要:

- このコマンドはCONSISTENT=YでExportを呼び出します。ORA-01555エラーを避けるために、適切なサイズのRBSを必要とします。
- このコマンドを双方向環境または高可用性環境で使用するには、追加の手順が必要です。ナレッジベースの解決策48020を参照してください。
- このコマンドを使用する前に、このドキュメントをすべて確認してください。**使い方を誤ると、データが失われたり、同期が取れなくなったりする可能性があります。**

copyコマンド

データを同期する前にターゲットテーブルを切り捨てたい場合は、**copy**コマンドを使用して、あるホストから別のホストにテーブル(または複数のテーブル)を同期またはインスタンス化します。

appendコマンド

データを同期する前にターゲットテーブルを切り捨てたくない場合は、**append**コマンドを使用して、あるホストから別のホストにテーブル(または複数のテーブル)を同期またはインスタンス化します。

copy/appendコマンドを使用するとき

copy/appendコマンドは以下のように使用します。

- データベースを移行するため。
- Compare/Repairプロセスでrepairできないテーブルを同期するため。
- テーブルをインスタンス化する際にレプリケーションに入れる。

copy/appendコマンドがサポートするもの

copy/appendコマンドは以下をサポートしています。

- **copy/append**コマンドでサポートされるオブジェクトは、OracleのImport/Exportユーティリティでサポートされているオブジェクトです。
- 下位バージョンのOracleから上位バージョンのOracleへのコピー、または同じバージョンのOracle間でのコピー。
- Oracleワイルドカード指定の結果として含まれる同期ルーティングファイル内のオブジェクトのコピー/追加。
- コマンドライン構文でのOracleワイルドカード指定の使用。

copy/appendコマンドがサポートしていないもの

copy/appendコマンドは、以下をサポートしていません。

- 上位バージョンのOracleから下位バージョンのOracleへのコピー
- テーブルからビューへのコピー
- UDTまたはVARRAY（UDTまたはVARRAYタイプがOracleのImport/Exportユーティリティまたは Oracleのホットバックアップを使用してターゲットデータベースに作成されていない場合）。これはOracleのインポート/エクスポート機能の制限です
- 水平分割または垂直分割されたレプリケーション内のテーブル
- シーケンス
- トランスフォーメーションが使用されているテーブル(トランスフォーメーションがターゲットにも適用されている場合を除く)。
- ソースサーバとターゲットサーバ間のデータ受け渡しにパススルーサーバを使用するネットワーク構成
- 列のマッピング
- 列のサブセット
- テーブル内の行のサブセットの指定
- インデックス、制約、トリガ、およびグラントオプション
- 直接ロード
- 透過的データ暗号化（TDE）
- **appendコマンド**はOracleパーティションをサポートしていません（**copy**コマンドのみがOracleパーティションをサポートしています）。
- Oracle Database 23cでは、暗号化されたテーブルスペースではCopy/Appendコマンドはサポートされません。

重要:

- **copy/append**コマンドは、Oracleワイルドカードをサポートしています。ただし、ワイルドカード指定で示されるオブジェクトのセットに、現在レプリケーションに含まれないオブジェクトが含まれている場合、**copy/append**コマンドは、現在レプリケーションに含まれているオブジェクトに対してのみ作用します。
- **force**オプションを使用し、すべてのターゲット情報を指定した場合、レプリケーションに含まれるテーブルは現在の動作と同様に静的にコピーされます。

ランチャープロセス

copy/appendコマンドを実行する前に、ターゲットシステムでランチャープロセスを実行しておく必要があります。**ランチャー**プロセスを開始するには、各ターゲットシステムで以下のコマンドを実行する必要があります。

```
sp_ctrl(sysB)> start launcher
```

ランチャープロセスの実行中は、**show launcher**コマンドを使用してプロセスの詳細を表示できます。

ユーザが同期を完了したら、ターゲットシステムでランチャーを停止することができます。**ランチャー**プロセスを停止するには、各ターゲットシステムで以下のコマンドを実行する必要があります。

```
sp_ctrl(sysB)>stop launcher
```

コピー/追加プロセスの概要

copy/appendコマンドは、個々のホスト間で個々のテーブルを同期するプロセスを開始します。

以下は、単純な同期プロセスのアクティビティを示すイベントのシーケンスです。

1. ターゲットシステムにログオンし、**sp_ctrl**から**start launcher**コマンドを発行します。
2. **sp_ctrl**セッションから**copy/append**コマンドが発行されると、**sp_cnc**はソースシステム上に**sp_sync_svr**を生成します。
3. **sp_sync_svr**はターゲットシステムに接続し、ターゲットシステムで**sp_sync_clnt**を起動します。
4. **sp_sync_svr**は、**sp_syn_clnt**にテーブルリストを送信し、ターゲットテーブルの情報を確認します。a) オブジェクトがコピーされている場合（追加ではなく）、ターゲットテーブルは切り捨てられます。
5. コピー対象の各オブジェクトに対して、**sp_sync_svr**はOracleのexportプロセスを開始します。
6. コピー対象のオブジェクトがレプリケーションに含まれる場合、**sp_sync_svr**は、**sp_sync_clnt**がPostプロセスの準備ができたことを示すまで待機する必要があります。
7. **sp_sync_clnt**プロセスは、**sp_sync_svr**プロセスがOracleのexportプロセスを開いた各オブジェクトに対してOracleのimportプロセスを開始します。
8. **sp_sync_svr**プロセスは、コピーが完了するまで、データを**sp_sync_clnt**プロセスに転送します。
9. コピー対象のオブジェクトがレプリケーションに含まれる場合、Postプロセスが再度開始され、そのポストアクティビティを再開します。
10. ターゲットシステムにログオンし、**sp_ctrl**からstop launcherコマンドを発行します。

コピープロセスの同時実行

一度に複数の**copy/append**コマンドを発行できます。ユーザは、コピープロセスの進行中にPostプロセスがすべてのレプリケーションを停止することに注意する必要があります。

同期プロセスの識別

同期プロセスは、**sp_cop**プロセスによって開始されないため、**sp_ctrl**コンソールにリストされません。プロセスを識別するには、オペレーティングシステムを使用して実行可能ファイルを探します。

- sp_sync_svr
- sp_sync_clnt
- sp_sync_lstnr
- sp_sync_launcher

同期のチューニングパラメーター

以下は、**copy/ append**コマンドを使用した同期に関連するパラメーターのリストです。これらのパラメーターはすべてライブであり、次にコマンドが発行されたときに有効になります。

- **SP_OSY_COMPRESSION**: このパラメーターは、データ圧縮をオフ（0）からフル（9）まで調整します。デフォルトの整数値は6に設定されています。
- **SP_OSY_EXP_BUFFER_SIZE**: このパラメーターは、ターゲットシステムに送信される前にバッファされるデータ量を調整します。デフォルトの整数値は1024キロバイトです。

- **SP_OSY_IMP_BUFFER_SIZE**: このパラメーターは、ターゲットシステムに適用される前にバッファされるデータ量を調整します。デフォルトの整数値は1024キロバイトです。
- **SP_OSY_LOCK_TIMEOUT**: このパラメーターは、同期プロセスがテーブルロックを取得するまで待機する時間を秒数で設定します。デフォルトの整数値は2秒です。
- **SP_OSY_PORT**: このパラメーターは、同期プロセスのポート番号を設定します。デフォルトの整数値は、ポート番号2501に設定されています。
- **SP_OSY_POST_TIMEOUT**: このパラメーターは、同期プロセスが、Postプロセスの準備が整い、同期が開始されるまで待機する時間を秒数で設定します。デフォルトの整数値は1800秒（30分）に設定されています。
- **SP_OSY_TCP_TIMEOUT**: このパラメーターはIPCタイムアウトの秒数を設定します。デフォルトの整数値は60秒に設定されています。
- **SP_OSY_THREADS**: このパラメーターは、同期処理のスレッド数を設定します。このスレッドパラメーターは、パーティション分割されたテーブルにのみ影響します。デフォルトの整数値は5スレッドに設定されています。最大値は32スレッドです。

コピーステータスの表示

1つまたは複数のcopy/appendコマンドのステータスを表示する最適な方法は、**copy status**コマンドまたはappend statusコマンドを使用することです。基本コマンドは、システム上のすべての**copy/append**プロセスの結果を表示します。

基本コマンドは、システム上で現在実行中のすべての**copy/append**ジョブのステータスと、最後に実行された**copy/append**（現在実行中のリストに含まれていない場合）のステータスを表示します。このリストは、指定したステータスのみを表示するようにフィルタリングすることができます。

1つ以上の**copy/append**コマンドの結果を表示する最適な方法は、**sp_ctrl**で**copy**オプションを指定してreportコマンドを使用することです。

以下の例では、**copy/append**コマンドがレプリケーションに含まれないテーブルに対して発行されています。その後の**copy status**は、テーブルに関する基本情報と手順のステータスを表示します。

```
sp_ctrl (alvlabl17:8708)> copy sp_iot.SYNC_iot_BASIC to sp_iot.SYNC_iot_BASIC for
o.w920a32f at irvqasu15@o.a102u64f force

     copy started; job id 111

sp_ctrl (alvlabl17:8708)> copy status 111 for o.w920a32f

     Job ID : 111

Host : alvlabl17.quest.com

Started : 17-MAR-08 13:59:28

Job Type : Copy

Status : Done

ID   Tablename                  Total rows   %Comp     Status    Status time  Total time
---  -------------------------  -----------  --------  --------  -----------  ----------
1    "SP_IOT"."SYNC_IOT_BASIC1"  3720         100       Done      N/A          0:08
2    "SP_IOT"."SYNC_IOT_BASIC2"  3720         100       Error     N/A          0:08
3    "SP_IOT"."SYNC_IOT_BASIC3"  3720         100       Done      N/A          0:08
```

このリストをフィルタリングして、ステータスがエラーであるテーブルだけを表示するには、以下のコマンドを実行します。

```
sp_ctrl (alvlabl17:8708)> copy status 111 for o.w920a32f "Error"
```

同期ルーティングファイルの使用

同期ルーティングファイルは、アクティベーションに使用される設定ファイルと非常によく似た構造をしており、ファイルの内容には同じフォーマット要件があります。この2つのファイルの違いは、設定ファイルのテーブルやオブジェクトの中には、同期プロセスでサポートされていないため、同期ルーティングファイルに含めるべきではないものがあることです。例えば、水平分割または垂直分割のテーブルは、同期プロセスでサポートされていないため、同期ルーティングファイルに含めるべきではありません。

同期ルーティングファイルでのOracleワイルドカードの使用はサポートされています。同期ルーティングファイルでのOracleワイルドカードの適用方法は、設定ファイルの場合と同じです。SharePlexがワイルドカードをどのようにサポートしているかについては、『SharePlex管理ガイド』を参照してください。

ルーティングファイルの例

```
Datasource:o.s_sid1

s_user0.sync_multi_trgt   d_user0.sync_multi_trgt      d_host1@o.d_sid1

s_user0.sync_multi_trgt   d_user0.sync_multi_trgt      d_host2@o.d_sid1

s_user0.sync_multi_trgt   d_user0.sync_multi_trgt_bu   d_host1@o.d_sid2
```

メモ:

- テーブルのグループに対して**copy/append**コマンドを使用する場合、すべてのテーブルがレプリケーションの対象であるか、すべてのテーブルがレプリケーションの対象外である必要があります。レプリケーション対象のテーブルとレプリケーション対象外のテーブルが混在している場合は、グループごとに個別の同期ルーティングファイルを作成し、それぞれのファイルで個別の同期プロセスを開始する必要があります。
- **copy/append**コマンドはランダムな順序でテーブルをコピーし、参照整合性を考慮しません。したがって、一度に1つのテーブルだけをコピー/追加するか、**コピー/追加**をごく少数のテーブルに制限することをお勧めします。

異種ソーステーブルとターゲットテーブルのコピー

ソーステーブルとターゲットテーブルの構造は同一でなければなりませんが、例外として、ソーステーブルの列数はターゲットテーブルより少なくても構いません。

copy/appendコマンドは、ソーステーブルの列数が対応するターゲットテーブルより多い場合は使用しないでください。

分割レプリケーションでのコピーの使用

設定内のテーブルが分割レプリケーションを使用する場合:

- **copy/append**コマンドは、個々のテーブルのコピーとパーティション化された個々のテーブルのコピーをサポートしますが、個々のパーティションのコピーはサポートしていません。
- 指定されたターゲットオブジェクトがパーティションであっても、データが個々のパーティションにコピーされることはありません。
- パーティションをターゲットとして指定することはサポートされていません。

- データの同期は、**copy**でも**append**でも、常にベーステーブルで行われ、パーティションで行われることはありません。
- **copy/append**コマンドは、現在レプリケーションに含まれている垂直または水平のパーティション分割されたテーブルをサポートしていません。
- [singlethread]オプションを使用すると、パーティション分割されたテーブルが1つのオブジェクトとしてコピーされます。デフォルトの動作は、パーティション分割されたテーブルの各パーティションを個別にコピーするので、コピーを並行して実行できます。

処理スレッド数の制御

複数のプロセッサを搭載したマシンを活用するために、SP_OSY_THREADSパラメーターを使用して、ソースシステム上のコピー/追加処理スレッド数を設定できます。このスレッドパラメーターは、パーティション分割されたテーブルにのみ影響します。このパラメーターのデフォルト値は5スレッドに設定されており、有効な値の範囲は1～32スレッドです。

テーブルがレプリケーションの対象である場合、同期プロセスはシングルスレッドであることに注意してください。マルチスレッド処理が有効になるのは、レプリケーション対象外のテーブルのみです。

コマンドライン構文でのOracleワイルドカードの使用

Oracleワイルドカードは、**copy/append**コマンドのsource_table部分を指定するときに使用できます。さらに、**not**表記を使用して、特定のオブジェクトを除外することもできます。

特定のスキーマ内の全テーブルを指定するには、以下を使用します。

```
sp_ctrl (irvspxu14:8567)> copy scott.%
```

上記の例では、オブジェクトが現在レプリケーションに含まれている場合、scottスキーマ以下のすべてのオブジェクトがコピーされます。

また、以下を使用して除外を設定することもできます。

```
sp_ctrl (irvspxu14:8567)> copy scott.% not (%"foo"%)
```

上記の例では、オブジェクトが現在レプリケーションに含まれている場合、名前に「foo」が含まれるオブジェクトを除く、scottスキーマ以下のすべてのオブジェクトがコピーされます。

SharePlexがワイルドカードをどのようにサポートしているかの詳細については、『SharePlex管理ガイド』を参照してください。

copy/appendの使用に関するその他の考慮事項

- **copy/append**コマンドを使用して同期されていないテーブルを同期する場合、または**copy/append**コマンドを使用してテーブルをインスタンス化する際にレプリケーションに入れる場合、テーブルのコピー中にPostプロセスが一時停止します。**copy/append**コマンドが完了すると、Postプロセスは再開されます。これは、新しいルートを通じてレプリケーションに追加されるテーブルには適用されません。
- **copy/append**コマンドは、テーブルが現在レプリケーションに含まれている場合、一度に1つのテーブルに対してのみ操作できます。
- 既知のレプリケーションルートにないテーブルに**copy/append**コマンドを使用する場合は、**force**オプションを使用する必要があります。レプリケーション対象外のオブジェクトに**copy/append**を使用するには、**force**オプションを使用する必要があります。

- レプリケーション対象のテーブルに**copy/append**コマンドを使用する場合、**copy/append**コマンドの発行時に、すべてのSharePlexプロセス(Capture、Read、Export、Import、Post)が実行中でなければならず、プロセス中も実行中のままである必要があります。
- カスケードレプリケーション環境では、**copy/append**コマンドを使用しないでください。
- 外部キーを持つテーブルは、それらのキーを一旦無効にしてから再度有効にする必要があります。
- テーブルインデックス、制約、トリガ、およびグラントオプションはサポートされていないため、同期完了後に(再度)有効にする必要があります。
- オブジェクトがレプリケーション対象の場合、**copy/append**は、ソーステーブルとターゲットテーブル間のロックや同期は必要ありません。**copy/append**コマンドが実行されている間、ソーステーブルとターゲットテーブルが変更されないことを前提としているからです。
- 複数のオブジェクトを含む設定ファイルが指定された場合、ソーススキーマ名とターゲットスキーマ名が異なることが許容されるのは、すべてのターゲットオブジェクトが同じスキーマ名を持つ場合のみです。
- **copy/append**コマンドは、オブジェクトが*異なる*スキーマに存在する場合のみ、同じデータベース上の2つのオブジェクト間でデータをコピーまたは追加することができます。
- **copy/append**プロセスの実行中に新しい設定をアクティベーションすると、同期プロセスが失敗する可能性があります。

注意: **copy**コマンドと**append**コマンドの構文はまったく同じです。以下に示す例では、**copy**コマンドを使用していますが、**copy**を**append**に置き換えるだけで**append**コマンドを呼び出すことができます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	なし

構文

基本コマンド	コマンドオプション	リモートオプション
{**copy** | **append**} *source_owner.source_table*	**[not** (*list of exceptions*)]	[**on** *host* |
	[to*target_owner.target_table*[*.partition*]**]**	**on** *host:portnumber* |
	[at *target_host*[*@target_SID*]	**on** *login/password@host* |
	[**for o.***source_SID*]	**on** *login/password@host:portnumber*]
	[singlethread]	
	[force]	

基本コマンド	コマンドオプション	リモートオプション
	[status]	
{**copy** | **append**} **using** *filename*	[**for o.***source_SID*]	[**on** *host* |
	[force]	**on** *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
source_owner.source_table	個々のソース所有者とソーステーブル名、またはワイルドカードを使用してオブジェクトのグループを指定します。**using***filename*構文とともに{**copy** | **append**}を使用している場合は無効です。 **to***target_owner.target_table*オプションなしで使用する場合、この構文は同期ターゲットがレプリケーションターゲットと同じであると想定します。 **例**: `sp_ctrl(SysA)>`**copy** s_user1.sync_single_tbl `sp_ctrl(sysA)>`**append** s_user1.sync_single_tbl
filename	同期するテーブルを含む同期ルーティングファイルの名前。ファイル名では大文字と小文字が区別されます。 このオプションは、{**copy** |**append**}**using**コマンドに使用します。 **例**: `sp_ctrl(SysA)>`**copy using** sync_file_2.txt `sp_ctrl(sysA)>`**append using** sync_file_2.txt
[**not** (list of exceptions)]	テーブル指定にワイルドカードが含まれている場合に、コピー/追加しないテーブルの例外リストを指定します。 `exception_list`は、コピー/追加してはならないテーブル名のリストです。 copytest1、copytest2、copytest3という3つのテーブルがあるとします。以下の例では、copytest3テーブルのデータのみがターゲットにコピーされます。 `sp_ctrl > copy s_user1.% not (copytest1,copytest2)`
to*target_owner.target_table* [*.partition*]	同期のターゲットテーブルを指定します。このオプションは、*source_owner.source_table*オプションと併用する場合のみ有効です。 テーブルがレプリケーションに含まれていない場合、このオプションを**at***target_host*@ [*target_sid*]オプションと組み合わせて使用します。

コンポーネント	説明
	例: `sp_ctrl(SysA)>`**copy** s_user1.sync_single_tbl **to** d_user1.sync_single_tbl `sp_ctrl(SysA)>`**append** s_user1.sync_single_tbl **to** d_user1.sync_single_tbl
at *target_host@* [*target_SID*]	同期のターゲットテーブルの場所を指定します。**to** *target_owner.target_table* [*.partition*]オプションと組み合わせて使用します。 **例**: `sp_ctrl(SysA)>`**copy** s_user1.sync_single_tbl **to** d_user1.sync_single_tbl at d_host1@o.d_sid1 `sp_ctrl(SysA)>`**append** s_user1.sync_single_tbl **to** d_user1.sync_single_tbl at d_host1@o.d_sid1
for o.*source_SID*	アクティブな設定がない場合、または複数のアクティブな設定がある場合に、同期するオブジェクトが含まれるデータソースを指定するために使用します。 **例**: `sp_ctrl(SysA)>`**copy** s_user1.sync_single_tbl **for** o.source_sid at target_host@o.target_sid force `sp_ctrl(SysA)>`**append** s_user1.sync_single_tbl **for** o.source_sid at target_host@o.target_sid force
singlethread	パーティション分割されたテーブルを1つのオブジェクトとしてコピーするには、このオプションを使用します。デフォルトの動作は、パーティション分割されたテーブルの各パーティションを個別にコピーするので、コピーを並行して実行できます。
force	このオプションは、同期を意図したテーブルがレプリケーションに*含まれない*、またはテーブルルートが既存または既知のルートに一致しない場合に使用します。 **例**: `sp_ctrl(SysA)>`**copy** s_user1.sync_single_tbl at target_host@o.target_sid **force** `sp_ctrl(SysA)>`**append** s_user1.sync_single_tbl at target_host@o.target_sid **force**
status	以前の**copy**/**append**コマンドのステータス履歴を表示します。 `sp_ctrl(SysA)>`**copy status** `sp_ctrl(SysA)>`**append status**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

copy cache

copy cacheコマンドを使用して、オブジェクトキャッシュをソースからターゲットにコピーします。

SharePlexイベントログに以下のメッセージがある場合は、**copy cache**コマンドにPostプロセスのターゲットホストとターゲットSID、およびエラーメッセージにある**actid**（アクティベーションID）を含めてください。これは、Postプロセスが現在ポストしているアクティベーションのアクティベーションIDです。

```
Poster [SP-OPO01009] cannot read object cache for actid <nnnn>
```

copy cacheコマンドをオプションなしで発行すると、SharePlexはソース上の現在のアクティベーションIDを識別し、そのアクティベーションIDのオブジェクトキャッシュをソースからすべてのターゲットにコピーします。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター（2）
発行場所:	ソースシステム
関連コマンド:	**show log**

構文

基本コマンド	コマンドオプション	リモートオプション
copy cache	[**actid** *actid*]	[**on** *host* |
	[**at** *target_host*[**@***target_sid*]]	**on** *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
copy cache	**copy cache**コマンドをオプションなしで発行すると、SharePlexはソース上の現在のアクティベーションID（actid）を特定し、そのactidのオブジェクトキャッシュをすべてのターゲットにコピーします。
actid *actid*	指定されたターゲットまたはすべてのターゲット（ターゲットが指定されていない場合）にコピーするオブジェクトキャッシュのアクティベーションIDを指定します。
at*target_host*[**@***target_sid*]	指定されたactidまたは現在のactid（actidが指定されていない場合）のオブジェクトキャッシュをコピーするターゲットを指定します。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

copy config

設定ファイルをコピーし、そのコピーを別のファイル名で保存するには、**copy config**コマンドを使用します。

設定をコピーすることで、以下のことが可能になります。

- バックアップ用に複製を保存する
- **edit config**コマンドを使用して、新しい設定を作成する際のベースとしてコピーを使用する
- アクティブな設定を変更する。現在の設定を使用したレプリケーションを継続しながら、**edit config**コマンドを使用してコピーに変更を加え、準備ができたら新しい設定をアクティベーションすることができます。

このコマンドは、元の設定と同じソースシステムに設定をコピーします。設定ファイルを*異なる*システムにコピーするには、テープ、CD、またはFTP経由でファイルを転送します。

システムの設定のリストを表示するには、**list config**コマンドを使用します。設定の内容を表示するには、**view config**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**activate config**、**create config**、**edit config**、**list config**、**remove config**、**rename config**、**show config**、**verify config**、**view config**

構文

基本コマンド	リモートオプション
copy config *filename* **to***newname*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename **to** *newname*	• *filename*はコピーする設定の名前です。設定名では大文字と小文字が区別されます。 • **to**は構文の必須部分です。 • *newname*はコピーに付ける名前です。システム上の設定間で一意である必要があります。

コンポーネント	説明
	例: `sp_ctrl(sysA)>` **copy config sales to sales2**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

copy status

copy statusコマンドは、最後に実行された**copy**または**append**ジョブのステータスを表示するために使用します。**copy statusコマンド**を使用すると、copyジョブまたはappendジョブ、またはcopyジョブまたはappendジョブの一部の詳細なステータスを表示したり、SharePlexが履歴を保持するすべてのcopyジョブおよびappendジョブのステータスを表示したりできます。

copy statusコマンドの詳細および使用例については、**job statusコマンド**を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	ビューアー
発行対象:	ソースまたはターゲット
関連コマンド:	**append status**

構文

基本コマンド	コマンドオプション	リモートオプション
job status	**[job_id]**	[**on** *host* |
	[job_id.table_id]	**on** *host:portnumber* |
	[all]	**on** *login/password@host* |
	[full]	**on** *login/password@host:portnumber*]
	[detail]	
	[status]	

構文の説明

コンポーネント	説明
job status	オプションを指定しないと、このコマンドはSharePlexが履歴を持つすべてのジョブのステータスを表示します。
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)>` **job status 2828.2**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)>` **job status 2828.HR.SRC_TEST3**

コンポーネント	説明
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)>` **job status all**
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)>` **job status2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)>` **job status detail**
[status]	指定されたステータスの過去のジョブのステータス履歴を表示します。 `sp_ctrl(SysA)>` **job status "Error"**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

create config

新しい設定ファイルを作成するには、**create config**コマンドを使用します。このコマンドは、SharePlexのデフォルトのテキストエディタを実行して、新しいファイルを開きます。設定を作成するには、『*SharePlex管理者ガイド*』の第5章の手順に従ってください。

新しい設定に既存の設定と同じ名前を付けようとすると、SharePlexは次のエラーメッセージを返します。「`The parameter for create config must be a new filename.`」(create configのパラメーターは新しいファイル名にする必要があります。)

設定を作成する代わりに、**copy config**コマンドを使用して既存の設定を複製し、**edit config**コマンドを使用してそのコピーを編集することもできます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**activate config**、**copy config**、**edit config**、**list config**、**remove config**、**show config**、**verify config**、**view config**

構文

基本コマンド	リモートオプション
create config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	作成する設定の名前。設定名では大文字と小文字が区別されます。この名前は、システム上の設定間で一意である必要があります。 **例**: `sp_ctrl(sysA>` **create config sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

create encryption key

create encryption keyコマンドを使用して、SharePlexがネットワーク上のデータを暗号化するために使用する暗号化キーを作成します。SharePlexはAdvanced Encryption Standard（AES）暗号化を使用します。

create encryption keyコマンドは、ランダムに生成された256ビットのAESキーを返します。デフォルトでは、SharePlexはこの長さの128ビットを使用してデータを暗号化します。

SharePlexが使用するキー長を長くするには、**SP_XPT_AES_KEY_LENGTH**パラメーターを192ビットまたは256ビットに設定します。長くするとキーのハッキングは難しくなりますが、CPUに多くの処理能力が必要になります。

```
sp_ctrl> set param sp_xpt_aes_key_length {192 | 256}
```

例: **set param sp_xpt_aes_key_length 256**

以下はキーの例です。

```
E5F5D4CBA329D2C86B5D7ABA096C18600595490129F55A1422AAB0248B28D0E4
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター（2）
発行場所:	ソースシステム
関連コマンド:	**set encryption key**、**reset encryption key**、**show encryption key**

構文

基本コマンド	リモートオプション
create encryption key	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

deactivate config

アクティブな設定のレプリケーションをグレースフルに終了するには、**deactivate config**コマンドを使用します。このコマンドは、設定のすべてのCaptureアクティビティを停止し、現在キューにあるすべてのデータをポストし、関連するSharePlexプロセスとキューを削除します。

deactivate configコマンドは、システムから設定を削除するのではなく、再度アクティベーションするまで、そのオブジェクトのレプリケーションを停止するだけです。設定を非アクティベーションした後、ユーザが設定されたソースオブジェクトに変更を加え続けると、ターゲットデータが同期しなくなる可能性があります。システム上の1つの設定を非アクティベーションしても、同じシステム上の他のアクティブな設定は非アクティベーションされません。

複数の設定がアクティブな場合の非アクティベーション

同じソースシステム上の複数のデータベースからレプリケートする場合、各設定にインスタンスのデータストリームを分離する名前付きの(別個の)exportキューがある場合のみ、**deactivate config**コマンドを使用します。通常、すべてのレプリケートされたインスタンスのデータは1つのexportキューに蓄積されます。**deactivate config**コマンドを発行すると、exportキュー内のすべてのデータが(キュー自体と一緒に)削除されます。これには、非アクティベーションしたくない設定のデータも含まれます。名前付きexportキューを使用した設定の作成方法については、『*SharePlex管理者ガイド*』の第5章を参照してください。

アクティブな設定とわずかに異なる設定をアクティベーションする場合、時間を節約するためには、アクティブな設定を最初に非アクティブする必要はありません。設定がまだアクティブである間に、**copy config**コマンドを使用してアクティブな設定をコピーできます。**edit config**コマンドを使用してコピーを変更してから、コピーをアクティベーションします。この方法を使用すると、SharePlexは両方の設定に共通するオブジェクトを再分析しないため、時間を節約できます。SharePlexは新しいオブジェクトのみを分析します(コピーをアクティベーションする前に最初の設定を非アクティベーションすると、SharePlexはすべてのオブジェクトを分析します。このプロセスには時間がかかります)。コピーをアクティベーションする前に、最初の設定のアクティベーションプロセスが完了していることを確認してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者(1)
発行場所:	ソースシステム
関連コマンド:	**abort config**、**list config**、**purge config**、**remove config**、**view config**

構文

基本コマンド	リモートオプション
deactivate config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	無効にする設定の名前。設定名では大文字と小文字が区別されます。 **例**: `sp_ctrl(sysA)>` **deactivate config sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

delete capture queue

非推奨のレプリケーションルートから残っているcaptureキューを削除するには、**delete capture queue**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**delete export queue**、**delete post queue**、**delete export process**

構文

基本コマンド	コマンドオプション	リモートオプション
delete capture queue for *datasource*	なし	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
datasource	**lstatus**コマンドの出力の**Queue Name**フィールドに表示されるcaptureキューのデータソース:

```
sp_ctrl (mysys)> lstatus on sys102

 Type    # Msgs   Size (Mb)  Age (mn) Oldest Msg Time    Newest Msg Time
------- --------- ---------- -------- ------------------ ------------------
-
Capture         0          2        0 11-Jan-16 11:16:02 11-Jan-16
11:16:02
  Queue Name:       o.ora112
Export          0          9        0 10-Jan-16 23:44:54 10-Jan-16
23:44:54
  Queue Name:       sysl02
  DataSrc-DataDst:  o.ora112-o.ora112
```

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

delete export process

非推奨のレプリケーションルートから残っているExportプロセスを削除するには、**delete export process**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**delete capture queue**、**delete export queue**、**delete post queue**

構文

基本コマンド	コマンドオプション	リモートオプション
delete export process to *host*	**queue** *queuename*	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

	説明
host	ターゲットシステムの名前。

queue *queuename*	システム上に複数のExportプロセスがある場合に、名前付きexportキューを通じて使用します。*queuename*には、削除するExportにリンクされているexportキューの名前を指定します。
	Exportプロセスとそのキューを表示するには、以下の例に示すように**show export**コマンドを使用します。
	`sp_ctrl (sys1:8567)> show export`
	`Host : sysl`
	`Queue : expquel`
	`Target Status Kbytes Exported Since Total Backlog`
	`sys2 Idle 245690 05-Feb-16 18:17:39 245690 0`
	Exportプロセスを削除すると、ターゲットシステムのExportプロセスに関連付けられたpostキューとPostプロセスも削除されます。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

delete export queue

非推奨のレプリケーションルートから残っているexportキューを削除するには、**delete export queue**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**delete capture queue**、**delete post queue**、**delete export process**

構文

基本コマンド	コマンドオプション	リモートオプション
delete export queue *queuename*	なし	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
queuename	**lstatus**コマンドの**Queue Name**フィールドに表示されるexportキューの名前:

```
sp_ctrl (mysys)> lstatus on sys102

 Type     # Msgs    Size (Mb)  Age (mn) Oldest Msg Time    Newest Msg Time
------- --------- ---------- -------- ------------------- -----------------
--
Capture          0          2        0 11-Jan-16 11:16:02 11-Jan-16
11:16:02
  Queue Name:       o.ora112
Export           0          9        0 10-Jan-16 23:44:54 10-Jan-16
23:44:54
  Queue Name:       sys102
  DataSrc-DataDst:  o.ora112-o.ora112
```

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

delete post queue

非推奨のレプリケーションルートから残っているpostキューを削除するには、**delete post queue**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**delete capture queue**、**delete export queue**、**delete export process**

構文

基本コマンド	コマンドオプション	リモートオプション
delete post queue *queuename* **for** *datasource-datadest*	**cleartrans**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明

queuename	**lstatus**コマンドの出力の**Queue Name**フィールドに表示されるpostキューの名前:
	`sp_ctrl (mysys)> lstatus on sys103`
	`Queues:`
	`Type # Msgs Size (Mb) Age (mn) Oldest Msg Time Newest Msg Time`
	`------- --------- ---------- -------- ------------------ ----------------`
	`--`
	`Post 0 5 0 10-Jan-16 23:44:54 10-Jan-16 23:44:54`
	`Queue Name: sys103`
	`DataSrc-DataDst: o.ora112-o.ora112`

for *datasource-datadest*	**lstatus**コマンドの出力の**DataSrc-DataDst**フィールドに表示されるルート指定:
	`sp_ctrl (mysys)> lstatus on sys103`
	`Queues:`
	`Type # Msgs Size (Mb) Age (mn) Oldest Msg Time Newest Msg Time`

コンポーネント	説明
	------- --------- ---------- -------- ------------------ ---------------- -- Post 0 5 0 10-Jan-16 23:44:54 10-Jan-16 23:44:54 Queue Name: sys103 **DataSrc-DataDst: o.ora112-o.ora112**
cleartrans	指定されたpostキューを削除し、SHAREPLEX_TRANSテーブルから関連する行を削除します。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

disable jobs / enable jobs

disable jobsコマンドおよび**enable jobs**コマンドを使用して、以下の同期コマンドによって実行されるジョブの実行を許可するかどうかを制御します。

- **copy**および**copy using**
- **append**および**append using**
- **compare**および**compare using**
- **repair**および**repair using**

disable jobsコマンドは、ピアツーピア設定または高可用性設定において、これらのジョブによって実行されるアクションが発行されないようにするセーフガードです。このコマンドは、**copy**、**append**、**repair**、または**repair using**コマンドがセカンダリシステム上で発行された場合に、プライマリ(信頼できるソース)のデータが信頼できないデータで上書きされるのを防ぎます。

enable jobsコマンドは、プライマリ(信頼できる)システムが再びアクティブなシステムになったときに、そのシステム上のジョブを有効にするために使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**copy**、**append**、**compare**、**compare using**、**repair**、**repair using**

構文

基本コマンド	コマンドオプション	リモートオプション
disable jobs	**for o.***source_SID*	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
for o.*source_SID*	アクティブな設定がない場合、または異なるOracleソースインスタンスに対して複数のアクティブな設定がある場合は、このオプションを使用します。 **例**: `sp_ctrl(SysA)>`**disable jobs for o.ora1** `sp_ctrl(SysA)>`**enable jobs for o.ora1**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

drop partition

drop partitionコマンドを使用して、水平分割レプリケーション設定内の1つのパーティションスキームから行パーティションを削除します。削除する行パーティションごとに、**drop partition**コマンドを発行します。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	add partition、modify partition、drop partition scheme、view partitions

構文

基本コマンド	リモートオプション
drop partition from *scheme_name*	利用不可
where	
{	
{**condition =** *column_condition* | **hash =** *hash_value* }|	
route = *routing_map* |	
name = *partition_name* |	
tablename = *target_table* |	
description = *description*	
}	

構文の説明

構文コンポーネント	説明
scheme_name	パーティションスキームの名前。
condition	削除するパーティションの行を定義する列条件。このオプションと**hash**オプションは相互に排他的です。
hash	削除するパーティションのハッシュ値。このオプションと**condition**オプションは相互に排他的です。
route	削除するパーティションのルーティングマップ。
tablename	削除するパーティション内のターゲットテーブルの完全修飾名。
name	削除するパーティションの短縮名。
説明	削除するパーティションの説明。

例:

```
sp_ctrl> drop partition from scheme1 where name = q1
sp_ctrl> drop partition from scheme1 where condition = "C1 < 200"
sp_ctrl> drop partition from scheme1 where route = sysb:q2@o.orasid
sp_ctrl> drop partition from scheme1 where hash = 4
```

drop partition scheme

SharePlexレプリケーション設定からパーティションスキームを削除するには、**drop partition scheme**コマンドを使用します。このコマンドを使用すると、パーティションスキームに指定されているすべての行パーティションも削除されます。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	add partition、modify partition、drop partition、view partitions

構文

基本コマンド	リモートオプション
drop partition scheme *scheme_name*	利用不可

構文の説明

構文コンポーネント	説明
scheme_name	パーティションスキームの名前。

例

```
sp_ctrl> drop partition scheme sales
```

editコマンド

editコマンドは以下のように使用します。

- 以前に発行したSharePlexコマンドに変更を加える。これにより、構文を少し変えて再度コマンドを発行する必要がある場合や、構文の間違いを修正する必要がある場合に、時間を節約することができます。
- オペレーティングシステムの制限である255文字を超えるコマンドを実行する。

editコマンドは、**sp_ctrl**インターフェイス内から、デフォルトのテキストエディタで一時テキストファイルを開きます。ファイルにコマンドを入力すると、ファイルを保存したときに自動的に実行されます。

editコマンドを使用するには:

1. **sp_ctrl**の現在のセッションで以前にコマンドが発行されていない場合、**sp_ctrl**プロンプトにコマンドの最初の単語を入力し、**Enter**キーを押します。それ以外の場合は、このステップをスキップします(表示されるオンラインヘルプやエラーは無視してください)。
2. **sp_ctrl**プロンプトで、editコマンドを発行して一時テキストファイルを開きます。

   ```
   sp_ctrl(sysA)> edit
   ```
3. テキストファイルで、エディタの標準コマンドを使用して不要な文字を削除し、コマンド文字列を入力します。
4. エディタの標準の保存コマンドを使用してファイルを保存し、終了します。コマンドはすぐに実行されます。

注意: SharePlexには、editコマンドのキーボードショートカットとして、上矢印(↑)キーとスラッシュ(/)キーが用意されています。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ユーザは、以前のコマンドの認証レベル以上が必要です。
発行対象:	以前のコマンドの影響を受けるシステム
関連コマンド:	なし

構文

基本コマンド	リモートオプション
edit | ed	利用不可

edit config

非アクティブな設定ファイルを修正するには、**edit config**コマンドを使用します。このコマンドは、SharePlexのデフォルトのテキストエディタで設定を開きます。エディタが提供するツールを使用して設定を修正し、保存します。

アクティブな設定を修正するには、**copy config**コマンドでアクティブな設定をコピーし、**edit config**コマンドでコピーを編集することをお勧めします。コピーをアクティベーションすると、元の設定は非アクティブになります。

システムの設定のリストを表示するには、**list config**コマンドを使用します。設定の内容を表示するには、**view config**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**activate config**、**copy config**、**create config**、**list config**、**remove config**、**rename config**、**verify config**、**view config**

構文

基本コマンド	リモートオプション
edit config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	編集する設定の名前。設定名では大文字と小文字が区別されます。
	例: `sp_ctrl(sysA)>` **edit config sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

exitコマンド

sp_ctrlの現在のセッションを閉じるには、**exit**コマンドを使用します。**sp_ctrl**を閉じても、レプリケーションはシャットダウンされません。ユーザによって停止されない限り、すべてのレプリケーションプロセスは中断されることなく続行されます。このコマンドは、そのシステム上の**sp_ctrl**でのセッションを単に終了します。sp_ctrlを再度実行するには、SharePlexバイナリを含むディレクトリに移動し、**./sp_ctrl**コマンドを入力します(UnixおよびLinuxシステムの場合)。

exitコマンドには [**on** *host*]オプションはありません。実行されている**sp_ctrl**を停止するシステム上で発行する必要があります。

このコマンドは**quit**コマンドと同じです。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**quit**

構文

基本コマンド	リモートオプション
exit | e	利用不可

expandコマンド

ワイルドカード指定を展開して画面に表示するには、**expand**コマンドを使用します。これは、レプリケートしたいテーブルのリストがワイルドカード指定によって生成されるかどうかが不確かな場合に特に便利です。設定ファイルでワイルドカードを使用する方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**create config**、**edit config**、**show config**、**verify config**

構文

基本コマンド	コマンドオプション	リモートオプション
expand *filespec*	[**not** (*list of exclusions*)]	[**on** *host* |
	[**for o.***source_sid*]	**on** *host:portnumber* |
	[**on** *host*]	**on** *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filespec	コマンドが検証するワイルドカード指定(所有者を含む)。**for o.***source_sid* オプションがない場合、ワイルドカードオブジェクトが含まれるアクティブな設定ファイルが存在するものと見なされます。 **例**: `sp_ctrl(sysA)>` **expand scott.%** この例では、**scott**スキーマ内のすべてのオブジェクトが結果に表示されます。
not (*list of exclusions*)	ワイルドカード指定の除外。リストされたオブジェクトは展開から除外されます。 **例**: `sp_ctrl(sysA)>` **expand scott.% for o.oraA not (%"sal"%)** この例では、名前に「sal」が含まれるオブジェクトを除いて、oraAインスタンスの**scott**スキーマ内のすべてのオブジェクトが結果に表示されます。この例では、大文字と小文字を区別するため、salが引用符("")で囲まれていることに

コンポーネント	説明
	注意してください。
for o.*source_sid*	アクティブな設定ファイルがない場合に使用し、コマンドを実行するOracleインスタンスを指定します。 例: `sp_ctrl(sysA)>` **expand scott.% for o.oraA** この例では、oraAインスタンスのscottスキーマ内のすべてのオブジェクトが結果に表示されます。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

flushコマンド

特定の時点でPostプロセスを停止するには、**flush**コマンドを使用します。このコマンドは、データストリームにマーカーを配置し、マーカーポイントで自動的にPostプロセスを停止します。例えば月末のレポートを作成する必要があるときなど、このコマンドは特定の日時で発行することができ、ターゲットシステム上のデータは、コマンドが発行された時点でソースシステム上にあったものが反映されます。

バックアップを実行するとき、または特定の文書化されたSharePlexの手順に従うときは、ソースシステムのユーザアクセスを停止した後に**flush**コマンドを発行して、Postが停止される前に最後のトランザクションがターゲットデータベースにポストされるようにします。

Postの再開

flushコマンドの後に生成された変更はキューに蓄積され、Postを再度起動した後にターゲットインスタンスに適用されます。**start post**コマンドが発行されるまで、Postは停止したままになります。Postは停止した場所の記録を保持し、同期を維持するためにその時点からポストを再開します。

重要: Postが停止している間にユーザがソースシステムで変更を続けると、データがpostキューに蓄積され、使用可能なディスク容量がすべて消費される可能性があります。Postは可能な限り早く開始してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**stop**

構文

基本コマンド	コマンドオプション	リモートオプション
flush *datasource*	[**to** *host*]	[**on** *host* |
	[**queue** *queuename*]	**on** *host:portnumber* |
	[**to** *host* **queue** *queuename*]	**on** *login/password@host* |
	[**to** *host@target_SID*]	**on** *login/password@host:portnumber*]
	[**to** *host@target_SID* **queue** *queuename*]	

構文の説明

コンポーネント	説明
datasource	この引数は、コマンドが作用するソースデータを指定します。追加オプションがない場合、すべてのターゲットシステムのすべてのキューを通してデータをフラッシュし、Postを停止します。 *datasource*は**o.***SID*で表わされます。ここで*SID*はソースOracleインスタンスのORACLE_SIDです。 **例**: `sp_ctrl(sysA)>` **flush o.oraA** この例では、インスタンスoraAのデータがすべてのターゲットシステムにフラッシュされます
to *host*	このオプションは、指定されたターゲットシステムにレプリケートしているすべてのデータをフラッシュします。他のターゲットシステムへのレプリケーションは影響を受けません。 **例**: `sp_ctrl(sysA)>` **flush o.oraA to sysB** この例では、データはsysAからsysBにフラッシュされます。
queue *queuename*	このオプションは、指定されたpostキューのデータをフラッシュします。これは、すべてのターゲットシステム上のその名前のキューを流れるデータをフラッシュします。このオプションは、名前付きpostキューがある場合に最も便利です。 **例**: `sp_ctrl(sysA)>` **flush o.oraA queue sales** この例では、salesという名前のpostキューを持つすべてのターゲットシステム上のpostキュー*sales*のデータがフラッシュされます。
to *host* **queue** *queuename*	このオプションは、指定されたターゲットシステム上の指定されたpostキューのデータをフラッシュします。そのシステムおよび他のターゲットシステム上の他のpostキューは影響を受けません。 **例**: `sp_ctrl(sysA)>` **flush o.oraA to sysB queue sales** この例では、sysB上のpostキュー*sales*のデータがフラッシュされます。
to *host@datadest*	このオプションは、指定されたターゲットシステム上の指定されたターゲットインスタンスにデータをフラッシュします。次のものには影響しません。 • そのシステム上の他のターゲットインスタンス。 • 他のターゲットシステム上の同じORACLE_SIDを持つ他のターゲットインスタンス この構文では:

コンポーネント	説明
	• *host*は、ターゲットシステムの名前です。 • *datadest*は**o.***SID*で表されます。ここで、*SID*はターゲットインスタンスのORACLE_SIDです。 **例**: `sp_ctrl(sysA)>` **flush o.oraA to sysB@o.oraB** この例では、データはsysB上のインスタンスoraBにフラッシュされます。
to *host@datadest* **queue** *queuename*	このオプションは、指定されたターゲットシステム上の指定されたpostキューとターゲットインスタンスのデータをフラッシュします。次のものには影響しません。 • そのインスタンスまたはそのシステム上の他のターゲットインスタンスの他のpostキュー。 • 他のターゲットシステム上の他のpostキュー。 この構文では: • *host*は、ターゲットシステムの名前です。 • *datadest*はo.SIDで表されます。ここで、SIDはターゲットインスタンスのORACLE_SIDです。 • *queuename*はpostキューの名前です。 **例**: `sp_ctrl(sysA)>` **flush o.oraA to sysB@o.oraB queue sales** この例では、sysB上のターゲットインスタンスoraBのpostキュー*sales*のデータがフラッシュされます。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**

オプション	説明
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

helpコマンド

sp_ctrl内で作業中にSharePlexコマンドのヘルプを取得するには、**help**コマンドを使用します。

- このコマンドをオプションなしで使用すると、SharePlexコマンドグループのリストが生成されます。
- 追加オプションを使用すると、各グループをドリルダウンして特定のコマンドのヘルプを取得できます。

このコマンドには**[on** *host***]**オプションはありません。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**用途**

構文

基本コマンド	コマンドオプション	リモートオプション
help	{*group_name*}	利用不可
	{*command_name*}	

構文の説明

コンポーネント	説明
group_name	各グループ内で、特定のヘルプを取得できるコマンドを表示します。有効な値: **config** **connect** **info** **maint** **misc** **param** **properties** **queues** **run** **sync**

コンポーネント	説明

	各グループのコマンドの一覧については、「SharePlexコマンド一覧」を参照してください。
	
	例:
	
	`sp_ctrl(sysA)>` **help run**
	
	`Control processes`
	
	`abort service - Stop one or all replication processes immediately.`
	
	`shutdown - Shut down SharePlex.`
	
	`start - Restart one or all stopped replication processes.`
	
	`stop - Stop one or all replication processes gracefully.`

command_name	このオプションは、指定されたコマンドのヘルプを提供します。 **例**: `sp_ctrl(sysA)>` **help start** `Restart a SharePlex process that was stopped with a "stop" or "abort service" command.`

hostコマンド

sp_ctrlのデフォルトシステムを確立するには、**host**コマンドを使用します。デフォルトのマシンでは、それぞれに**on** *host*オプションを使うことなく、一連のコマンドを入力できます。デフォルトシステムへのTCP/IP接続は、**sp_ctrl**を終了するか、別のhostコマンドを発行するまでアクティブなままになります。

デフォルトシステムの名前は、sp_ctrlプロンプトの一部として、以下のように表示されます。

```
sp_ctrl(hostname)>
```

デフォルトシステムは、**sp_ctrl**を実行しているマシン、またはそのマシンがTCP/IPで接続できる他のシステムになります。

デフォルト以外のシステムに対してコマンドを発行するには、コマンドの[on host]オプションを使用して、一時的に代替システムに接続します。コマンドが終了すると、デフォルトの**sp_ctrl**接続に戻ります。

hostコマンドは、**sp_ctrl**内の**sp_ctrl**プロンプトからのみ発行できます。

ヒント: sp_ctrlセッションのデフォルトポート番号を設定するには、portコマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**port**

構文

基本コマンド	**リモートオプション**
host *hostname*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	**説明**
hostname	デフォルトとして設定するマシンの名前。この名前は、システムの/etc/hostsファイルに存在する必要があります(UnixおよびLinux)。 **例**: `sp_ctrl(sysA)>` **host sysB**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

hostinfoコマンド

hostinfoコマンドを使用して、オペレーティングシステム、マシン名、OSバージョン、リビジョンなど、システムのハードウェアに関する情報を表示します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	なし

構文

基本コマンド	リモートオプション
hostinfo	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。

オプション	説明
	例:`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

job statusコマンド

job statusコマンドを使用して、最後の**compare**、**compare using**、**repair**、**repair using**、**copy**、または**append**ジョブのステータス、およびまだ実行中の他のジョブのステータスを表示します。**job status**コマンドを使用すると、ジョブまたはジョブの一部の詳細なステータスを表示したり、SharePlexが履歴を保持するすべてのジョブのステータスを表示したりできます。SharePlexは、SP_SYS_JOB_HISTORY_RETENTIONパラメーターで指定されたすべてのジョブの履歴を保持します。履歴やジョブ・プロセス・ログを能動的に削除する方法については、**clear history**コマンドおよび**remove log**コマンドを参照してください。

```
sp_ctrl (sysA) > job status

Job ID     : 861
PID        : 20571
Host       : sysa.domm.com
Started    : 06-NOV-12 11:07:05
Job Type   : Compare
Status     : Processing - 4 objects completed


ID  Tablename                Status       Time       Total rows %Comp  Total time
--- ------------------------ -------- --------- ---------- ------ ----------
1   "SCOTT"."SRC_TEST1"      Out Sync     N/A        19         100    0:05
4   "SCOTT"."SRC_TEST4"      WaitMarker   0:02       27392             0:04
5   "SCOTT"."SRC_TEST5"      Init         0:01       27498             0:01
```

SharePlexが履歴を保持するすべてのジョブの概要を表示するには:

```
sp_ctrl (alvspxl11:8567)> job status all


Job ID Type      filename/tablename      Status     Started          Completed
------ ------    ------------------      --------   ---------------  ---------------
3441   Compare   prod.conf               Done - errors 16-DEC-11 15:39 16-DEC-11 15:40
3442   Repair    scott.src_test4         Done       16-DEC-11 15:50 16-DEC-11 15:50
3443   Repair    prod.conf               Done       19-DEC-11 10:42 19-DEC-11 10:42
3444   Copy      "SCOTT"."SRC_TEST5"     Done       19-DEC-11 10:43 19-DEC-11 10:43
3445   Compare   scott.src_test33        Done - errors 20-DEC-11 12:02 20-DEC-11 12:03
```

ジョブのステータスを詳細に表示するには:

```
sp_ctrl (alvspxl11:8567)> compare status detail


Job ID      : 3448
PID         : 763
Host        : sysa.domm.com
Started     : 20-DEC-11 12:40:46
Job Type    : Compare
Status      : Done - 6 objects completed

      Table ID            : 3448.3
      Table               : "SCOTT"."SRC_TEST3"
      From                : sysa.domm.com@o.w111a64f
      To                  : "SYSPROD"."SRC_TEST3" sysb.domm.com@o.w111a64f
      Started             : 20-DEC-11 12:40:55
      Percent complete    : 100%
      Total Rows          : 234452
      Rows processed      : 234452
      Rows out-of-sync    : 2
      Status              : Out Sync
      Status Elapsed      : N/A
      Total Elapsed       : 0:07
                  Inserts : 2
                  Updates : 0
                  Deletes : 0
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**copy status**、**append status**、**compare status**、**repair status**

構文

基本コマンド	コマンドオプション	リモートオプション
job status	[*job_id*]	[**on** *host* |
	[*job_id.table_id*]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |

基本コマンド	コマンドオプション	リモートオプション
	[**full**]	**on** *login/password@host:portnumber*]
	[**detail**]	
	[**status**]	

構文の説明

コンポーネント	説明
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)>` **job status 2828.2**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。 **例: `sp_ctrl(sysA)>` job status 2828.HR.SRC_TEST3**
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)>` **job status all**
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)>` **job status2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)>` **job status detail**
status	指定されたステータスの過去のジョブのステータス履歴を表示します。 `sp_ctrl(SysA)>` **job status “Error”**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

list config

ソースシステム上のすべてのアクティブおよび非アクティブな設定のリストを表示するには、**list configコマンド**を使用します。

このコマンドは、次の情報を表示します。

- **ファイル名**: 設定の名前、変更された日時、およびファイルのサイズ。
- **状態**: 設定がアクティブか非アクティブか。*アクティブ*は、設定ファイルが現在レプリケーションに関与していることを意味します。*非アクティブ*は、設定が現在レプリケーションに関与していないことを意味します。
- **データソース**: 設定によってレプリケートされるオブジェクトを含むOracleインスタンス。
- **内部名**: アクティブな設定の内部コピーの名前。これは、SharePlexが実際にレプリケートするファイルです。その名前は、アクティブな設定の**State**見出しの下に表示されます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**view config**、**show config**

構文

基本コマンド	リモートオプション
list config	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

list param

SharePlexチューニングパラメーターを表示するには、**list param**コマンドを使用します。オプションがない場合、コマンドは基本(ユーザが設定可能な)パラメーターとその現在の設定を表示します。

以下を表示するためのオプションが用意されています。

- SharePlexユーザが変更してはならないパラメーター(内部パラメーター)を含む、すべてのSharePlexパラメーター。
- 設定がデフォルト値から変更された基本パラメーターと内部パラメーターのみ(これらの設定は、SharePlex変数データディレクトリに記録されます)。
- SharePlexモジュールに従ってフィルタリングされた前述のオプションのいずれか。

パラメーターは*モジュール*ごとにグループ化されており、各モジュールはソフトウェアの機能コンポーネントを表します。次の表は、ユーザが設定可能なパラメーター(Questからの指導なしに変更可能)を含むモジュールの一覧です。

SharePlexパラメーターモジュール

モジュール	命名規則	制御対象の機能
analyze	SP_ANL	**analyze config**コマンド
cap	SP_CAP	Oracle以外のキャプチャ
capture	SP_OCTまたはSP_CAP	Captureプロセス
compare	SP_DEQまたはSP_CMP	**compare**コマンド
config	SP_OCF	設定のアクティベーションプロセス
cop	SP_COP	**sp_cop**
copy	SP_OSYまたはSP_CPY	copy/appendコマンド
export	SP_XPT	Exportプロセス
import	SP_IMP	Importプロセス
logging	SP_SLG	SNMP機能
post	SP_OPOまたはSP_OPX	Postプロセス
queue	SP_QUE	SharePlexキュー
read	SP_ORD	Readプロセス
SNMP	SP_SNMP	SNMPのサポート
system	SP_SYS	システム関連項目

各**list param**表示には、パラメーターの以下の情報が表示されます。

- **パラメーター名**
- **実際の値**: パラメーターの現在の設定値、およびそのデフォルト値(現在の設定値と異なる場合)。

注意: このフィールドには36文字までしか表示されません。

- **単位**: 秒、キロバイト、操作など、パラメーターの測定単位。
- **セットアットポイント**: set paramコマンドを発行したときに、パラメーターが有効になるポイント。パラメーターが設定された直後、影響を受けるプロセスが再起動された後、または**sp_cop**が再起動された後のいずれか。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**set param**、**reset param**

構文

基本コマンド	コマンドオプション	リモートオプション
list param	[**basic** | **all** | **modified**] [*module*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
basic	このオプションは、Questサポートエンジニアや技術専門家の指導なしに、ユーザが設定できるパラメータを表示します。 **例**: `sp_ctrl(sysA)>` **list param basic**
all	このオプションは、すべてのSharePlexパラメーターを表示します。これには、基本パラメーター(ユーザが設定可能なパラメーター)と、Questからの指導なしに設定を変更してはならない**内部**パラメーターが含まれます。 **例**: `sp_ctrl(sysA)>` **list param all**
modified	このオプションは、デフォルト値から変更されたユーザが設定可能なパラメーターと内部パラメーターをリストします。 **例**: `sp_ctrl(sysA)>` **list param modified**

コンポーネント	説明
module	このオプションは、出力を特定のモジュールのパラメーターに制限します。「SharePlexパラメーターモジュール」を参照してください。 このオプションを使用する場合は、**list param [all | modified]**構文の後に指定する必要があります。 **例**: `sp_ctrl>` **list param all post** この例では、SharePlexはPostプロセスのすべてのパラメーターを表示します。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

lstatusコマンド

lstatusコマンドを使用して、ソースまたはターゲットシステム上のレプリケーションのステータスに関する詳細情報を表示します。このコマンドは**sp_ctrl**での最も包括的な情報コマンドです。レプリケーションの簡単なステータスを表示するには、**status**コマンドを使用します。

lstatusコマンドは以下の情報を表示します。

- 関連付けられたそれぞれのキューを含むレプリケーションプロセスのステータス。
- 各プロセスのオペレーティングシステムのプロセスID。
- プロセスがいつ開始されたか。
- プロセスが実行されているマシン。
- キュー内のメッセージ数（SQLまたはSharePlex操作）とその経過時間。
- マシンがソースシステムかターゲットシステムか。
- **sp_cop**の開始後に行われたイベント。
- アクティブな設定のリスト。

lstatusコマンドは、以下のようなキューに関する情報も提供します。

- **タイプ**: キューの種類（captureキュー、exportキュー、postキュー）。
- **キュー名**: キューの名前。デフォルトの名前か、ユーザが割り当てた名前付きキューの名前。
- **メッセージ数**: キュー内のメッセージの数。メッセージはほぼSQL操作に対応しますが、LONG列やLOB列に対する1つの操作に対して複数のメッセージが存在することもあります。逆に、配列の挿入では、多数の操作に対して1つのメッセージが存在する可能性があります。
- **サイズ**: キューの現在のサイズ。これはデータ量によって異なります。
- **経過時間**: キューの中で最も古いメッセージと最も新しいメッセージが、そのキューに書き込まれた時間の差。
- **最も古いメッセージの日時**: 最も古いメッセージがキューに入った日時。
- **最も新しいメッセージの日時**: 最も新しいメッセージがキューに入った日時。

lstatusコマンドは、システムの使用状況、設定アクティビティ、発生したエラーなど、その他の情報も提供します。

SharePlexキューを理解する

以下は、SharePlexキューの統計情報を理解するのに役立ちます。

キューサイズ

SharePlexはキューにメモリを使用しますが、データはチェックポイント・リカバリ・システムの一部として定期的にディスクに書き込まれます。SharePlexキューファイルのデフォルトサイズは8MBです。ただし、キューファイルはスパースファイルです。つまり、システムの観点から見るとファイルサイズは8MBですが、ファイルシステムはキューに書き込まれたデータのために、ファイルの一部のみを割り当てる可能性があります。

このようなことから、**qstatus**コマンドは8MB未満のキューサイズを表示できますが、**qstatus**で表示されるのは、*実際のファイルサイズを示す真の表示*ではありません。キューファイルのディスク使用量を予測するには、ファイルシステムではなく、**qstatus**からの情報を使用します。

キューの数

通常、captureキューとexportキューは、postキューよりもキューファイルの数が少なくなります。ソースシステム上の1つのキューファイルに存在するデータは、ターゲットシステム上の複数のサブキューファイルに分離され、それぞれがソースシステム上のユーザセッションにほぼ対応します。

キューの名前

キュー名はすべてのプラットフォームで大文字と小文字が区別されます。SharePlexキューの命名規則を以下で説明します。

- デフォルトのcaptureキューは、**o**.*SourceSID*で表されるデータソースによって識別されます。例えば、*o.ora64*。
- デフォルトのexportキューは、ソースシステムの名前、例えば*SysA*によって識別されます。
- デフォルトのpostキューは、ソースシステムの名前で識別され、*datasource-datadestination*が追加されます。そこでの*datasource*は**o**.*sourceSid*、*datadestination*は**o**.*targetSID*で　、例えば*lajolla　(o.ora10a-o.ora10b)*のようになります。
- 名前付きexportキューと名前付きpostキューは、デフォルトの名前ではなく、ユーザが割り当てた名前によって識別されます。名前付きキューの詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**qstatus**、**show**、**show statusdb**、**status**

構文

基本コマンド	リモートオプション
lstatus	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)で

オプション	説明
	コマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

modify partition

modify partitionコマンドを使用して、水平分割レプリケーション設定内の1つのパーティションスキームの行パーティションを変更します。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	add partition、drop partition、drop partition scheme、view partitions

構文

基本コマンド	リモートオプション
modify partition in *scheme_name*	利用不可
set	
keyword **=** *value*	
[**and** *keyword* **=** *value*]	
[...]	
where	
keyword **=** *value*	
[**and** *keyword* **=** *value*]	
[...]	

構文の説明

注: これらのオプションの詳細については、add partitionを参照してください。

構文コンポーネント	説明
scheme_name	パーティションスキームの名前。このコンポーネントは変更しません。変更すると行パーティションが新しいパーティションスキームに移行します。
keyword	*scheme_name*を除く、以下の構文コンポーネントのいずれか。
condition	行パーティションを定義する列条件。**condition**コンポーネントと**hash**コンポーネントは相互に排他的です。
hash	行IDハッシュベースのパーティションスキームに基づいて、SharePlexが作成する行パーティションの数を指定するハッシュカウント。**hash**コンポーネントと**condition**コンポーネントは相互に排他的です。
route	このパーティションのルーティングマップ。
tablename	完全修飾ターゲットテーブル名。
name	このパーティションの略称。
説明	このパーティションの説明。

例

```
sp_ctrl> modify partition in scheme1 set condition = "C1 > 400" and route = sysc:q1@o.orasid where name = q1

sp_ctrl> modify partition in scheme1 set condition = "C1 > 400" where condition = "C1 > 300"

sp_ctrl> modify partition in scheme1 set hash = 5  where hash = 4
```

orainfoコマンド

SIDを含むOracleデータベース情報を表示するには、**orainfo**コマンドを使用します。以下に示すのは表示例です。

```
sp_ctrl (mysysl11:2101)> orainfo
Oracle instance #1:
     Oracle SID ora12
     Oracle HOME /oracle/products/12
     Oracle Version 12
Oracle instance #2:
     Oracle SID ora12
     Oracle HOME /oracle/products/12
     Oracle Version 12
```

データベースアカウントがSharePlex用に作成されていないか、**statusdb**が削除されている場合、またはコマンドをサポートしていないバージョンのSharePlexから実行した場合は、次のように表示されます。

```
sp_ctrl (alvspxl11:2101)> orainfo


Oracle information not available
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	なし

構文

基本コマンド	リモートオプション
orainfo	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

portコマンド

sp_ctrlから、デフォルト以外のTCP/IPポート番号上で実行されている**sp_cop**のインスタンスに接続するには、**port**コマンドを使用します。**sp_cop**のデフォルトポート番号は、2100（SharePlexインストール時のデフォルトポート）か、SP_COP_TPORTおよびSP_COP_UPORTパラメーターで設定されたユーザ定義ポートのいずれかです。

sp_ctrlの現在のセッション中に発行されたすべてのコマンドは、**sp_ctrl**セッションが終了するまで、指定されたポートで実行中の**sp_cop**に影響を与えます。**sp_ctrl**の新しいセッションは、デフォルトのポート番号、またはSharePlexのインスタンスにユーザ定義ポートが定義されていれば、そのポートに接続します。

portコマンドの使用時期

sp_copのインスタンスを1つ実行している場合は、**port**コマンドを使用する必要はありません。起動時に、**sp_ctrl**は、まずユーザ定義ポートをチェックし、存在しない場合はポート2100をデフォルトとして、ポート番号を決定します。

sp_copの複数のインスタンスを専用ポートで実行しており（統合レプリケーションなど）、それらのデフォルトポートが**SP_COP_TPORT**および**SP_COP_UPORT**パラメーターで設定されていない場合は、**port**コマンドを使用します。複数の**sp_cop**インスタンスに対してコマンドを発行するには、**sp_ctrl**のセッションをそれぞれ開き、portコマンドを使って各セッションにポートを設定します。

sp_ctrlを起動する際、**port**コマンドが必要な場合、SharePlexは以下のように通知します。

> `Your tcp port is not set properly or "sp_cop" is not running.`（tcpポートが正しく設定されていないか、`"sp_cop "`が実行されていません。）

この警告は、**sp_cop**が実行されていないか、デフォルトとは異なるポートで実行されていることを示しています。**port**コマンドはリモートマシンからは発行できないので、[**on** *host*]オプションはありません。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー（3）
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**host**

構文

基本コマンド	リモートオプション
port *number*	利用不可

構文の説明

コンポーネント	説明
number	使用したいポート番号。**port**とポート番号の間には必ずスペースを入れてください。 **例**: `sp_ctrl>` **port 2200**

purge config

purge configコマンドを使用して、キューそのものを削除したり設定を非アクティブにしたりすることなく、設定に関連付けられたすべてのキューからデータを削除します。非アクティベーションを回避することで、SharePlexでは、設定データを再計算する必要がなくなります。これにより、テーブルが大きく数が多い場合に、時間を節約し、レプリケーションをより早く開始できるようになります。

ソースシステムで**purge config**コマンドを発行して、設定されたルートにあるソースシステムとすべてのターゲットシステムに影響を与えます。**purge config**アクティビティの実行前または実行中にSharePlexプロセスが停止すると、このコマンドも動作を停止します。プロセスが再開されると、コマンドは動作を再開します。したがって、**purge config**は、ネットワークが一時的に利用できなくなった場合でも機能し、接続が復元されるまでコマンドはキューに残ります。

purge configコマンド使用上の注意点:

- 設定をアクティブにしてから、**activate config**コマンドの後に、**purge config**コマンドを実行しないでください。レプリケーションを制御する設定情報など、キューに入れられたデータ以上のものをパージし、アクティベーションが無効になる可能性があります。
- 同じソースシステム上に複数のアクティブな設定がある場合、**purge config**コマンドは、それぞれのレプリケーションストリームを分離する名前付きexportキューがある場合にのみ使用します。名前付きexportキューがない場合、SharePlexは、レプリケートされたすべてのデータを1つのexportキューを通して転送します。そして、1つの設定に対して**purge config**コマンドを実行すると、それらのすべてに対してデータが削除されます。名前付きキューを作成するには、『*SharePlex管理者ガイド*』の第5章を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者(1)
発行対象:	ソースシステム
関連コマンド:	**abort config**、**deactivate config**

構文

基本コマンド	リモートオプション
purge config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	パージする設定の名前。設定名では大文字と小文字が区別されます。

コンポーネント	説明
	例: `sp_ctrl(sysA)>` **purge config sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

qstatusコマンド

qstatusコマンドを使用して、任意のシステムのcapture、post、およびexportキューの統計情報を表示します。各キューのメッセージ数、その経過時間、キューの現在のサイズが表示されます。通常の場合、メッセージはほぼSQL操作に対応しますが、LONG列やLOB列に対する1つの操作に対して複数のメッセージが存在することもあります。また、配列挿入の多数の操作に対して1つのレコードしか存在しないこともあります。また、メッセージはSharePlexの内部操作に対応する場合もあります。

qstatusコマンドを使用するタイミング

次の目的にqstatusコマンドを使用します。

- レプリケーションプロセスによって読み取られるかまたはターゲットデータベースにポストされるのを待機しているデータがまだあるかどうかを判断する。
- メッセージの蓄積速度を分析することで、SharePlexの処理速度を推定する。
- キューのサイズを表示し、利用可能なディスク容量を超えないようにする。
- 特定の運用手順で必要になる場合、またはシステムのメンテナンス、アップグレードなどの管理タスクのためにレプリケーションをシャットダウンする必要がある場合に、すべてのキューが空であることを確認する。
- データが同期されない原因となるようなユーザアクティビティがターゲットシステム上にあるかどうかを判断する。

出力について

- キュー内のメッセージ数は、次のSharePlexプロセスによって読み取られたメッセージと、読み取られていないメッセージを反映します。チェックポイント・リカバリ・システムの一部として、SharePlexは次のプロセスに送信されたメッセージのコピーを保持します。これらのメッセージは、そのプロセスによる受信が確認された時点で削除されます。
- Backlog（バックログ）フィールドは、次のSharePlexプロセスでまだ読み取られていないメッセージの数を示します。
- Age（経過時間）フィールドは、キュー内の最も古いメッセージと最も新しいメッセージがキューに書き込まれた時刻の差を示します。
- captureキューとexportキューもあるシステム上のpostキューの存在は、このシステムがソースシステムとしてもターゲットシステムとしても使用されていることを示します。
- Size（サイズ）フィールドは、キューの真のサイズを示します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー（3）
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**lstatus、show**

構文

基本コマンド	リモートオプション
qstatus	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

quitコマンド

sp_ctrlの現在のセッションを閉じるには、**quit**コマンドを使用します。**sp_ctrl**を閉じても、レプリケーションはシャットダウンされません。ユーザによって停止されない限り、すべてのレプリケーションプロセスは中断されることなく続行されます。このコマンドは、そのシステム上の**sp_ctrl**でのセッションを単に終了します。**sp_ctrl**を再度実行するには、SharePlexバイナリを含むディレクトリに移動し、./**sp_ctrl**コマンドを入力します（UnixおよびLinuxシステムの場合）。

quitコマンドには [**on** *host*]オプションはありません。実行されている**sp_ctrl**を停止するシステム上で発行する必要があります。このコマンドは**exit**コマンドと同じです。

用途

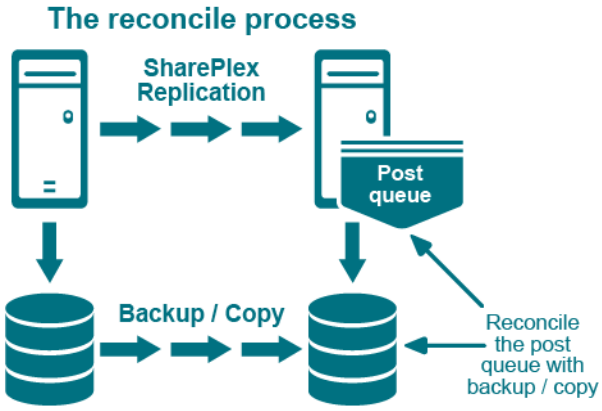
サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー（3）
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**exit**

構文

基本コマンド	リモートオプション
quit | q	利用不可

reconcileコマンド

reconcileコマンドをプロシージャの一部として使用して、データベースユーザの中断を最小限に抑えると共に、ソースデータとターゲットデータを同期（インスタンス化）します。**reconcile**コマンドは、進行中のレプリケーションの結果を、ホットバックアップやネイティブ・コピー・ユーティリティなどによってターゲットシステムに適用されたソースデータのコピーと照合します。照合機能は、postキュー内のレプリケートされた変更を、リカバリプロセス後のターゲットデータベースの状態とcompareします。リカバリ中に適用されたトランザクションと、まだ適用されていない（postキューで待機している）トランザクションを区別し、両システムが同期するように、重複していない変更のみをポストします。



reconcileコマンドは、大容量環境で使用するように設計されていますが、状況によっては照合プロセスが停滞する可能性があることを理解すれば、小容量環境でも使用することができます。この状況は、ソースシステムから継続して届くデータに**reconcile**コマンドが依存しているために生じます。ホットバックアップまたはコピーの実施以降、ソースシステムにレプリケーションアクティビティがない場合、照合プロセスはソースのアクティビティが再開するまで待機します。

reconcileコマンド使用時の注意点

reconcileコマンドは、ソースデータとターゲットデータの初期同期を特定の手順で行う場合に使用します。したがって、これは独立したコマンドではありません。初期同期手順については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者（1）
発行対象:	ターゲットシステム
関連コマンド:	**flush**

構文

基本コマンド	コマンドオプション	リモートオプション
reconcile queue *queuename* **for** *datasource-datadest*	[**seq** *sequence_number*]	[**on** *host* |

基本コマンド	コマンドオプション	リモートオプション
	[**scn** *scn_number*]	**on** *host:portnumber* |
	[**to flush**]	**on** *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明*
queue	**queue**はコマンドの必須部分です。
queuename	照合するターゲットシステム上のpostキュー。有効な値: • デフォルトキューを使用する場合のソースシステムの名前 • 名前付きキューを使用している場合のキューの名前。 名前付きpostキューを使用している場合は、それぞれに対して**reconcile**コマンドを実行します。キュー名を決定するには、**sp_ctrl**で**qstatus**コマンドを発行します。キュー名はすべてのプラットフォームで大文字と小文字が区別されます。
for *datasource-datadest*	• *datasource*は**o**.*SID*で表されます。ここで、*SID*はソースインスタンスのORACLE_SIDです。 • *datadest*は**o**.*SID*で表されます。ここで、*SID*はターゲットインスタンスのORACLE_SIDです。 **例**: `sp_ctrl (sysB)` > **reconcile queue SysA for o.oraA-o.oraB**
seq *sequence_number*	(Oracle)Oracleホットバックアップを使用して同期手順でOracleターゲットデータを確立する場合に、このオプションを使用します。これは、Oracleがリカバリに使用するのと同じログの最後まで照合するようSharePlexに指示します。 *sequence_number*は、Oracleがリカバリするログのシーケンス番号です。 この構文は、基本コマンドの構文の後に表示する必要があります。このオプションは、**to flush**オプションと併用しないでください。 **例**: `sp_ctrl (sysB)` > **reconcile queue SysA for o.oraA-o.oraB seq 1234**
scn *scn_number*	(Oracle)Oracleホットバックアップを使用して同期手順でOracleターゲットデータを確立する場合に、このオプションを使用します。これは、特定のOracle System Change Number(SCN)に対して照合するようSharePlexに指示します。 *scn_number*はOracleがリカバリするSCNです。 この構文は、基本コマンドの構文の後に表示する必要があります。このオプションは、**to flush**オプションと併用しないでください。 **例**: `sp_ctrl (sysB)` > **reconcile queue SysA for o.oraA-o.oraB scn**

コンポーネント	説明*
	0123456789
to flush	このオプションは、**flush**コマンドで確立されたフラッシュマーカーに対して照合するために使用します。ピアツーピアレプリケーション環境で複数のOracleデータベースを同期する場合に使用します。 この構文は、基本コマンドの構文の後に表示する必要があります。このオプションは**seq** *sequence_number*オプションと併用しないでください。 **例**: `sp_ctrl (sysA)`**> reconcile queue SysA for o.oraA-o.oraB to flush**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

redoコマンド

redoコマンドを使用すると、直前のコマンドを再入力することなく再度実行できます。このコマンドは、例えばキュー量の変化を監視するために**qstatus**コマンドを使用するなど、情報コマンドを使用して頻繁にステータスチェックを行う場合に有用です。

このコマンドは、**sp_ctrl**内からのみ発行できます。オペレーティングシステムのコマンドラインから直接発行することはできません。このコマンドには[**on** *host*]オプションはありません。

SharePlexには、**redo**コマンドと同じタスクを実行するキーボードショートカットも用意されています。**sp_ctrl**内でフォワードスラッシュまたは上矢印を押すと、直前のコマンドが実行されます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	直前のコマンドを「やり直す」には、そのコマンドに対応する権限レベルまたはそれ以上のレベルが必要です。
発行対象:	直前のコマンドの影響を受けるシステム。
関連コマンド:	なし

構文

基本コマンド	リモートオプション
redo |/ |**r**	利用不可

remove config

システムから設定ファイルを完全に削除するには、**remove config**コマンドを使用します。このコマンドは、確認のプロンプトを*表示せず*、設定ファイルの削除を元に戻すことはできません。アクティブな設定を削除することはできません。アクティブな設定を削除するには、まずその設定を非アクティブにします。

ヒント: 誤って削除された設定をリカバリできる可能性があるのは、その設定が以前にアクティブになっており、アクティベーションされてから**ora_cleansp**を実行していない場合です。設定をリカバリするには、イベントログを表示して、その設定ファイルのアクティベーションIDを特定します。SharePlexの変数データディレクトリの**save**サブディレクトリで.**conf**.*actid*ファイルを確認します。ここで、actidは、イベントログから取得したアクティベーションIDです。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**deactivate config**、**list config**、**show config**、**view config**

構文

基本コマンド	リモートオプション
remove config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	削除する設定の名前。設定名では大文字と小文字が区別されます。 **例**: `sp_ctrl(sysA)>` **remove config sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

remove log

compare、**compare using**、**repair**、**copy**、**append**コマンドは、ソースとターゲットの両方でログファイルを生成します。ジョブ情報とソース・ログ・ファイルは、ジョブがSP_SYS_JOB_HISTORY_RETENTIONより古い場合、またはclear historyコマンドが使用された場合にクリーンアップされます。データベースからのジョブ履歴の消去、またはターゲットからのログファイルの削除を行わずに、ソース上のログファイルを削除するには、remove logコマンドを使用します。

例:

```
sp_ctrl>remove log age 5
Logs removed
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**compare**、**repair**、**copy**、**append**

構文

基本コマンド	リモートオプション
remove log {all | age *days* | **copy** | **compare**}	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
all	この引数を指定すると、すべてのログが削除されます。 **例**: `sp_ctrl(sysA)>remove log all`
age *days*	この引数を指定すると、指定した日数より古いログが削除されます。 **例**: `sp_ctrl(sysA)>remove log age 10`
copy	この引数を指定すると、copyまたはappendコマンドのログが削除されます。

コンポーネント	説明
	例: `sp_ctrl(sysA)` **> remove log copy**
compare	この引数を指定すると、compareおよび/またはrepairコマンドのログが削除されます。 **例**: `sp_ctrl(sysA)` **> remove log**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

rename config

設定ファイルに別の名前を付けるには、**rename config**コマンドを使用します。システム上の設定ファイル間で固有の名前を使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**copy config**、**edit config**、**list config**、**view config**

構文

基本コマンド	リモートオプション
rename config {*filename***to** *newname*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename **to***newname*	• *filename*は、名前を変更する設定の名前です。設定名では大文字と小文字が区別されます。 • **to**は構文の必須部分です。 • *newname*は設定に付ける新しい名前です。 **例**: `sp_ctrl(sysA)` **> rename config sales to sales2**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

repair / repair using

repairおよび**repair using**コマンド（*repair*コマンドと総称される）を使用して、ターゲットテーブル内の同期していない行をrepairします。

- **repair**コマンドは、ワイルドカードを使用して指定されたスキーマ内の個々のターゲットテーブルまたは任意の数のターゲットテーブルをrepairします。個々のテーブルをrepairする場合は、列ベースのフィルタリングを使用して、repair用に選択される行を制御することができます。
- **repair using**コマンドは、アクティブ設定またはアクティブ設定内のテーブルのサブセットを含む別のファイルにリストされたターゲットテーブルをすべてrepairします。

repairコマンドは、まずcompareを実行してrepairが必要な行を特定してからrepairを実行します。テーブルがどのようにcompareされるかについては、compare / compare usingを参照してください。

注意: 実行中のcompareまたはrepairは、ソーステーブルには一切影響を与えません。SharePlexは、読み取りの一貫性についてのクエリを実行するためだけにデータベースにログインし、ソーステーブルのロックは短時間です。SharePlexは、処理中にターゲットテーブルを短時間ロックしますが、ユーザはロックをほとんど意識することなくアクセスを継続できます。

サポートされている操作

SharePlexは、**DML**操作（INSERT、UPDATE、DELETE）によって発生した、ターゲットテーブル内の非同期行を検出し、repairすることができます。

サポートなし

- **SharePlex は、以下のcompareとrepairをサポートせずにスキップします。**
    - 高度なデータ型
        - TEXT
        - CHAR > 2,000
        - VARCHAR > 4000
        - サイズなしのVARCHAR
        - BYTEA
    - シーケンス
    - トランスフォーメーションが使用されたテーブル
    - ソースサーバとターゲットサーバ間のデータ受け渡しにパススルーサーバを使用するネットワーク構成
- カスケードレプリケーション環境では、compareとrepairはサポートされません。

- compareまたはrepair中のテーブルに対してDDLを実行しないでください。compareは、SharePlexがサポートするものも含め、DDL操作によって引き起こされる非同期状態を検出しません。DDLがテーブル定義を変更すると、compareする必要がある行を取得するためにcompareプロセスにより作成されたSELECTステートメントが無効になります。

  DDLによって発生した非同期状態を修正したら、**repair**コマンドを使用して行のデータを再同期することができます。

- 255文字を超えるcompareおよびrepairコマンド文字列はサポートされていません。これはオペレーティングシステムの制限です。この制限を回避するには、ソースシステムで**edit**コマンドを使用します。テキストファイル内にコマンド文字列を入力すると、コマンドが自動的にファイルを実行します。

compareおよび**compare using**でサポートされるデータ型の詳細については、『SharePlexリリースノート』を参照してください。

その他の条件

- compareまたはrepairするテーブルは、アクティブな設定ファイルの一部でなければなりません。
- ソーステーブル上にコミットされていないトランザクションがあると、読み取りの一貫性を得るためにcompareおよびrepair処理に必要な短時間のロックを取得できなくなります。compareやrepairを実行する前に、すべてのトランザクションがコミットされていることを確認してください。
- レプリケーションのレイテンシは、compareやrepairの処理のパフォーマンスを低下させる可能性があります。ターゲット上でcompareおよびrepairプロセスを生成するためにソースから送信されたメッセージは、複製されたデータと共にキューを通じて送信されます。データバックログによる遅延が生じると、生成メッセージの送信が遅れることにより、プロセスが読み取りの一貫性を失うことがあります。可能であれば、compareやrepairはピーク以外の時間帯に行います。
- compareまたはrepair中のテーブルをトランケートしないでください。compareコマンドは、開始時に各ソーステーブルのスナップショットを取ります。テーブルがトランケートされると、スナップショットのテーブルビューもトランケートされ、コマンドが無効な非同期状態を返すことがあります。
- ビューをrepairするには、以下が真でなければなりません。
  - ビューテーブルが、基礎となるテーブルのすべての制約列を含んでいること。
  - ビューの制約列の名前が、基礎となるテーブルの制約列の名前と同一であること。
- Compareプロセスのパフォーマンスを向上させるために、SharePlexはSP_DEQ_PARALLELISMパラメーターによる並列ヒントをサポートしています。このため、並列度、つまり起動するワーカープロセスの数を指定できます。

  デフォルトでは、PostgreSQLデータベースオプティマイザはSQLステートメントに最適な問い合わせ実行計画を選択します。これは、並列ヒントで指定した並列度に直接関連していないこともあります。pg_hint_planにより、クエリを実行するたびに配列ヒントを使用して実行計画を調整することができます。SharePlexは、データベースにpg_hint_plan拡張機能がインストールまたは設定されている場合、内部でこれを利用します。

  並列クエリを使用するための構文の例は以下の通りです。

  pg_hint_plan拡張機能は並列ヒントをサポートしています。`Parallel(table <# of workers> [soft|hard])`

  SharePlexはハードパースをサポートする並列ヒントを使用します。`Parallel(table <# of workers> [hard])`

  > **注意:** ワーカーの数は、データベースシステムのCPU VCoreより少なくなければなりません。

repairコマンドの使用方法

compareおよびrepairコマンドによって同期されたデータを維持するための推奨手順は、まず**compare**または**compare using**コマンドを実行し、次に**repair status**コマンドで結果を表示することです。このコマンドは、非同期行とその考えられる原因を表示します。非同期状態の原因が修正されない限り、今回行をrepairしても、レプリケーションは再び非同期になります。問題が解決した後、**repair**または**repair using**コマンドを実行します。

repairまたは**repair using**コマンドは、予めcompareを行うことなく実行できます。このコマンドはまずcompareを行い、同期していない行を特定し、その行をrepairします。しかし、将来的な非同期状態を防ぐには、非同期状態の根本的な原因を解決しなければなりません。

非同期状態の原因と解決策については、『SharePlex管理ガイド』を参照してください。

repairを実行するタイミング

ターゲットテーブルをrepairする最適なタイミングは、テーブルのサイズ、問題の原因、非同期行の範囲、ユーザをロックアウトできる時間の長さによって異なります。repairを開始する前に、以下を考慮してください。

- 通常、テーブルのユーザは、テーブルのcompare時に適用される短時間のロックの影響は受けませんが、repair処理中はターゲットテーブルからロックアウトされます。小さなテーブルであれば支障はないかもしれませんが、大規模なrepairが必要な大きなテーブルの場合、待ち時間が非常に長くなることがあります。
- ターゲットテーブルのロックは、Postがrepairの終了を待ってから、そのテーブルに変更を適用し、他のテーブルに移動しなければならない場合、ポストのパフォーマンスを低下させる可能性があります。これがターゲットデータのレイテンシを増加させ、postキューに操作が蓄積される原因となります。Postが変更する必要があるオブジェクトと、repair中のオブジェクトが異なる場合、2つのプロセスは同時に実行されます。
- テーブルをすぐにrepairしなければならないが、ロックやレプリケーションのレイテンシを許容できない場合、**where**オプションを使用して、特定の行に限定してrepairすることができます。別の方法として**key**オプションを使用することもできますが、このオプションを使用すると、非同期行を見逃してしまう可能性があります。
- repairを待つことができる場合は、問題の原因をすぐに修正し、ピーク時以外の時間帯にrepairを行います。

特別な使用例

以下のシナリオでは、compareを実行する際に特別な処理が必要となります。

使用例	サポートcompare
統合レプリケーション	中央データベース内のターゲットテーブルは、どのソースデータベースよりも行数が多く、ソースデータベースよりも列数が多いことがよくあります。このような環境でrepairコマンドを使用する場合は、特別な配慮が必要です。 **repair usingコマンド** 統合レプリケーションは、**repair using**コマンドではサポートされていません。**repair using**コマンドでは、ソーステーブルに存在しないターゲット行が不要に削除されます。 回避策として、統合レプリケーションに関係するテーブルを除外した設定のサブセットを作成し、代わりにサブセット設定をrepairします。**repair**コマンドを使用すると、統合レプリケーションに関与するテーブルをrepairできます。 **repairコマンド**

使用例	サポートcompare
	統合レプリケーションは、ターゲットデータベースとPostプロセスがソースホストのIDを各行に追加するように設定されている場合にサポートされます。中央のターゲットテーブルの正しい行をcompareまたはrepairするには、**targetwhere**オプションを使用し、ソースID値を**where**句の基準にします。 例えば、ある企業の東部本社のデータベースのテーブルと、中央の企業データベースの正しい行をcompareするには、東部データベースのソースIDに「East」を使用し、その値を**targetwhere**句の基準にすることができます。**repair**コマンドで同じ**targetwhere**句を使用します。compareプロセスおよびrepairプロセスは、ソースID値を使用して、東部データベースに有効な行のみを選択します。 ソースIDを特定する統合レプリケーション以外の実装でcompareコマンドやrepairコマンドを使用すると、ターゲット行が不要に削除される可能性があります。この設定の詳細については、『SharePlex管理ガイド』を参照してください。 ターゲット行が正確に選択されるように、**targetwhere**オプションと標準の**where**オプションを組み合わせる必要があるかもしれません。
ピアツーピアレプリケーション	ピアツーピア構成では、*どのシステムを信頼できるソースシステムとし、どのシステムをセカンダリシステム*、つまりターゲットシステムとするかを決めなければなりません。セカンダリシステムは、repairが行われるシステムです。ピアツーピア環境でcompareやrepairを実行する前に、以下の手順に従ってください。 1. セカンダリシステムのユーザアクセスを停止し、そのシステムから複製された操作が信頼できるソースデータベースにポストされるのを待ちます。ユーザは引き続きソースデータベースにアクセスできます。 2. ソースおよびセカンダリシステムで**qstatus**コマンドを実行します。 3. キュー内のメッセージが10個以下であれば、ソースシステムからcompareを実行します。 4. compare中、**sp_xdesvr**および**sp_xdeclt**の開始後に、ソースおよびセカンダリデータベースへのユーザアクセスを許可できます。 5. **where**オプションを指定して**repair**コマンドを使用すると、ユーザをテーブルからロックアウトすることなく、ターゲットテーブルの選択した行をrepairできます。
キーのないテーブル	compareおよびrepairコマンドは、ソースおよびターゲットシステム上でORDER BY句付きのSELECTステートメントを発行します。大規模なテーブルにプライマリキーまたはNULL以外の一意キーとインデックス(一意のインデックスが望ましい)を持つ場合、順序付けはより速くなります。それ以外の場合は、すべての列がキーとして使用されます。 一意の行識別子はないが、一意の行として識別できる列がテーブルに1つ以上ある場合、**orderby**オプションを指定して**compare**コマンドを使用できます。このオプションを使用すると、SharePlexは、ソースシステム上の**sp_desvr**ログに、コマンドがこれらの列をキーとして使用したという通知を出力します。
ソーステーブルよりも列の数が多いターゲットテーブル	**repair**および**repair using**コマンドは、ソーステーブルに含まれていないターゲット列を無視します。repairでは以下のことを行います。 • INSERTは、ソーステーブルに対応する列を持つターゲット列に値を挿入しますが、それ以外の列には挿入しません。NOT NULL制約があるがデフォルト値がない列は、Oracleエラーの原因となります。ターゲットテーブル内の余分な列にはデフォルト値が推奨されます。 • UPDATEは、ソーステーブルに対応する列を持つターゲット列の値を再同期しますが、余分な列の値は再同期しません。 • DELETEはターゲットテーブル内の余分な列の影響を受けません。これは、repairコマンドは照合する列の列データに基づいて行を選択するためです。

使用例	サポートcompare
UNIQUE制約を持つテーブル	UNIQUE制約で定義された列では、**repair**、または**repair using**コマンドによって一意制約違反エラーが返されることがあります。次の例では、ソーステーブルとターゲットテーブルがそれぞれ2列ずつになっています。最初の列は主キーで、2番目の列にはUNIQUE制約があります。 **Source Table** 111 ABC 222 XYZ **Target Table** 111 XYZ 222 ABC SharePlexがターゲットテーブルの1行目をrepairしてソースと一致させようとすると、2列目のUNIQUE制約がエラーを返します。これは、値「ABC」が既に2行目に存在するためです。ターゲットテーブルの2行目でも、「XYZ」が既に1行目に存在するため、同じことが起こります。回避策は以下の通りです。 • **repair**コマンドを実行する前に、ターゲットテーブル上のUNIQUE制約を無効にし、repair終了後に再度有効にします。 • 固有の制約違反が発生したターゲット行を削除し、SharePlexによって正しいデータと共にこれらの行が挿入されるようrepairを再度実行します。
LOB列を持つテーブル	いずれかのターゲットテーブルにLOB列があるとrepairにかかる時間が長くなります。より高速なrepairを行うには、SP_DEQ_SKIP_LOBパラメーターを0に設定して、compareとrepairでLOB列がスキップされるようにします。詳細については、「SP_DEQ_SKIP_LOB 」の407ページを参照してください。
XMLデータでのrepair操作	XMLデータに対してrepair操作を実行すると、「`ORA-04036: PGA memory used by instance exceeds PGA_AGGREGATE_LIMIT(ORA-04036: インスタンスが使用するPGAメモリがPGA_AGGREGATE_LIMITを超えています)`」エラーが表示されます。 **回避策:** 1. Oracleをバージョン19.19 DBRUにアップグレードします

2. 提供されている以下のOracle推奨スクリプトを使用して、適切なPGA_AGGREGATE_LIMITを設定します

```
WITH

MAX_PGA as

(select round(value/1024/1024,1) max_pga from v$pgastat where
name='maximum PGA allocated'),

MGA_CURR as

(select round(value/1024/1024,1) mga_curr from v$pgastat where
name='MGA allocated (under PGA)'),

MAX_UTIL as

(select max_utilization as max_util from v$resource_limit where
resource_name='processes')

SELECT

a.max_pga "Max PGA (MB)",

b.mga_curr "Current MGA (MB)",

c.max_util "Max # of processes",

round(((a.max_pga - b.mga_curr) + (c.max_util * 5)) * 1.1, 1) "New
PGA_AGGREGATE_LIMIT (MB)"

FROM MAX_PGA a, MGA_CURR b, MAX_UTIL c

WHERE 1 = 1;
```

3. SharePlexバージョン11.1にアップグレードするか、そのバージョンをインストールします。

repairの仕組み

repair操作では、以下のように**DML**操作によって発生した、ターゲットテーブルの非同期状態をrepairします。

- 余分な行または欠けている行
- 値が一致しない行

repairの条件

repairおよび**repair using**コマンドは、以下の修正SQLステートメントを発行します。

- ソース側に行が存在し、ターゲット側に行が存在しない場合、SharePlexはINSERTステートメントを発行します。
- ターゲット側に行が存在し、ソース側に存在しない場合、SharePlexはDELETEステートメントを発行します。

- ターゲット行がソース行と異なり、キー列が一致する場合、SharePlexはソース値に基づいてUPDATEステートメントを発行します。
- Oracleパーティション分割テーブルで**repair**または**repair using**コマンドを使用している場合、repairプロセスのデフォルト動作では、INSERTとDELETEのみを使用して行をrepairします。UPDATEを必要とするrepairは、UPDATEによって行がパーティションを変更し、行の移動がテーブルで有効になっていない場合のエラーを防ぐため、DELETEに変換され、INSERTが続きます。SharePlexが、INSERT、UPDATE、DELETEを適切に使用してパーティション分割されたテーブルをrepairするように、repair動作を変更できます。詳細については、「SP_DEQ_PART_TABLE_UPDATE」の406ページを参照してください。

プロセス

repairには常に、ターゲットテーブル内の非同期状態を特定するためのcompareが含まれます。**repair**または**repair using**コマンドを実行すると、SharePlexは以下の一連のイベントを開始します。

1. **sp_cop**プロセスは、ソースシステム上に**sp_desvr**（サーバー）プロセスを生成します。
2. **sp_desvr**プロセスは、**sp_ctrl**インターフェイスの制御と使用をユーザに返します。compareが進行している間、レプリケーションは続行されます。
3. **sp_desvr**プロセスは、ターゲットシステム上で**sp_declt**（クライアント）プロセスを開始するために、Postプロセスにメッセージを送信します。
4. サーバプロセスとクライアントプロセスによって相互の通信が確立されます。
5. 行の選択とrepairは次のように進行します。
    - **repair**コマンドを使用している場合、**sp_desvr**はソーステーブルから行を選択し、**sp_declt**はターゲットテーブルから行を選択します。行はソートされ、compareされてrepairされます。
    - **repair using**コマンドを使用している場合、**sp_desvr**プロセスはターゲットシステム上に多数の処理スレッドを作成します。SP_DEQ_THREADSパラメーターによって設定される値は、作成されるスレッド数を制御します。各スレッドは、**sp_declt**（クライアント）プロセスを生成します。サーバプロセスとクライアントプロセスによって相互の通信が確立されます。処理負荷はクライアントプロセス間で分割されます。ソースとターゲットの各テーブルから行が選択され、ソートされ、compareされてrepairされます。ターゲットテーブルは、repairの順番になるとロックされ、その後ロックが解除されます。
    - ターゲットテーブルは、repairの順番になるとロックされ、その後ロックが解除されます。
6. 終了すると、プロセスによってログファイルが書き出され、**show repair**コマンドで結果を見ることができます。

エラー処理

SharePlexは、repair SQLステートメントの適用時にデータベースエラーが発生すると、そのステートメントから先のrepairを停止し、直前に適用された有効なステートメントのみをコミットします。このように、テーブルは部分的にrepairされますが、まだ同期していない可能性があります。**repair status**コマンドは、この状況を警告します。

SQLログファイルの管理

compareおよびrepairコマンドは、すべての非同期行をrepairするために必要なSQLを、ログファイルと同じ場所にあるSQLファイルに書き込みます。compareコマンドのみが発行された場合、SharePlexはこれらのSQLステートメントを実行しません。repairコマンドを実行した場合、このコマンドは、非同期行をrepairするSQLステートメントを実行すること以外はcompareコマンドと同じように動作します。

SQLログファイルの出力を抑制することができます。このファイルを抑制する理由には以下のようなものがあります。

- データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。
- compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。

SQLログファイルを抑制するには、**nosqllog**オプションを**compare**または**repair**コマンドで使用します。

SharePlexの現在のインスタンスの実行中に、compareとrepairのすべての実行についてSQLログファイルの出力を抑制するには、SP_SYS_SECURE_MODE環境変数を1に設定します。この変数はSharePlexを起動する前に設定する必要があるため、**sp_cop**プロセスが実行されている場合は、この変数を設定した後に再起動する必要があります。この環境変数を指定して**sp_cop**を実行すると、compareコマンドとrepairコマンドによってSQLファイルにデータは保存されず、Post処理でもSharePlexのエラーログにデータは保存されません。

複数のプロセスの実行

すべてのcompareおよびrepairコマンドでは、複数のプロセスを同時に実行することができます。複数の**compare**および**repair**コマンドを同時に実行し、それぞれがソーステーブルとターゲットテーブルのペアを処理することも、PostgreSQLのワイルドカードを使用して1つのコマンドで複数のテーブルセットを指定することもできます。SharePlexによるワイルドカードのサポートの詳細については、『SharePlex管理ガイド』を参照してください。

レプリケーションプロセスとcompareおよびrepairプロセスなど、最大で20のSharePlexプロセスでpostキューを同時に使用することができます。一度に最大で5つのcompareおよびrepairプロセスを実行できるようにすることをお勧めします。

制限に達したためにcompareやrepairに失敗した場合、SharePlexはイベントログにメッセージを記録します。

注意: **edit**コマンドを使用し、以前のコマンドを編集して新しいコマンドを作成すれば、複数のコマンドをより簡単に実行できます。

設定のサブセットのrepair

アクティブな設定のサブセットは、以下の方法でrepairできます。

- 1つのスキーマに属するレプリケーション内の全ターゲットテーブルをrepairするには、以下のように**repair**コマンドにワイルドカードを指定します。

  ```
  sp_ctrl> repair scott.%
  ```

- 設定ファイル内のすべてのターゲットテーブルをrepairするには、以下のように**repair using**コマンドを使用します。

  ```
  sp_ctrl>repair using myconfig
  ```

- 1つのターゲットルート内のすべてのターゲットテーブルをrepairするには、以下のように**at**オプションを指定して、**repair using**コマンドを使用します。

  ```
  sp_ctrl>repair using config.active at prodsys@o.ora112
  ```

- 設定内のテーブルのカスタムサブセットをrepairするには、*repairファイル*から指定します。これはプレーン・テキスト・ファイルであり、repairしたいターゲットのソーステーブルのみをリストします。ターゲットテーブルは、コマンドが発行された時点の設定ファイルから取得されます。**create config**または**copy config**コマンドを使用してrepairファイルを作成できます。このファイルには、必ず設定ファイルでないことを明確に示す名前を付けてください。詳細については、コマンドの構文を参照してください。

どの行をrepairするかを制御

compareおよび**repair**コマンドには、処理の対象として選択する行をフィルタリングできる**where**オプションがあります。デフォルトでは、これらのコマンドはテーブルのすべての行に影響し、ソーステーブルに含まれていないターゲットテーブルの列は無視されます。

- **where**オプションを使用すると、ソーステーブルとターゲットテーブルの同じ名前の列に基づいて行をフィルタリングすることができます。
- 垂直分割レプリケーションを使用するテーブルには**where**オプションを使用します。ソース列とターゲット列には異なる名前を付けることができます。**where**の選択をソース列に基づいて行います。SharePlexは、ターゲットテーブルの正しいWHERE句を構築するために、設定ファイルから列マッピングを読み取ります。
- ソーステーブルまたはターゲットテーブルに1つ以上の余分な列が存在し、それらの行に行の一意性を決定する値が含まれている場合は、**sourcewhere**および**targetwhere**オプションを使用します。
  - ソーステーブルに余分な列が含まれている場合は、**sourcewhere**を使用します。
  - ターゲットテーブルに余分な列が含まれている場合は、**targetwhere**を使用します。

  このオプションを正しく使用するには:
  - 余分な列には**sourcewhere**または**targetwhere**オプション*のみ*を使用します。
  - ソースとターゲットの両方で同じ名前を持つ他の列には、標準の**where**オプションを使用します。
  - SharePlexは、**where**オプションと**sourcewhere**または**targetwhere**オプションを組み合わせて、完全なWHERE句を作成します。

> **重要!** 余分な行を持つターゲットテーブルに対してcompareとrepairの両方を実行する予定がある場合は、UPDATEとDELETEのcompare用に**targetwhere**のみを使用してください。**repair**コマンドはINSERTの正しい値を決定できません。この問題を回避するには、余分な列にデフォルト値を設定するか、挿入された行を手動で更新します。

プロセスの特定

compareまたはrepairコマンドが発行されるたびに、ジョブIDが**sp_ctrl**ディスプレイに表示されます。**sp_ctrl**ディスプレイを利用できない場合は、**compare status**コマンドを実行することでジョブIDを表示できます。

sp_ctrlでステータスと結果を表示

repairのステータスや結果を表示するには、**sp_ctrl**の**repair status**コマンドを使用します。

- 基本コマンドは、最近開始されたrepairジョブの処理ステータスと、まだ終了していないその他のジョブを表示します。
- 追加オプションを使用すると、履歴のあるすべての修理ジョブのサマリーステータスを表示したり、1つのジョブに関する詳細情報を表示したりすることができます。

詳細については、repair statusを参照してください。

警告とエラーの表示

sp_xdesvrおよび**sp_xdecltプロセス**は、実行するシステム上にログファイルを書き込みます。SharePlexの変数データディレクトリの**log**サブディレクトリにあるファイルを開きます。

sp_xdesvrプロセスが書き込むログの名前は`xdesvr_<jobid>_r.<dbid>_p<process id>.log`です。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブIDです。
- *SID*は、ソーステーブルがあるOracleインスタンスのORACLE_SIDです。
- *ProcessID*は**sp_xdesvr**プロセスのプロセスIDです。

sp_xdecltプロセスが書き込むファイル名は、`xdeclt_<jobid>-<tableid>_r.<dbid>_p<processid>.log`に**.log**または**.sql**を付加したものです。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブのジョブIDです。
- *TableID*は、SharePlexによって割り当てられたジョブ内のテーブルのテーブルIDです。
- *SID*は、ソーステーブルがあるOracleインスタンスのORACLE_SIDです。
- *SourceHost*は、ソースホストの名前またはIPアドレスです。
- *ProcessID*は、**sp_decltプロセス**のプロセスIDです。

ログファイル名の例:

```
xdesvr_7_r.aparopka_p4970.log

xdeclt_7-1_r.aparopka_p25095.log

xdeclt_7-1_r.aparopka_p25095_01.sql
```

ディスク使用量を制御するために、ログは循環方式でエージングされます。現在のログがサイズの限界に達すると、SharePlexは新しいログファイルを生成します。新しいログは最大ログ数まで作成され、その後、SharePlexは最も古いログから上書きを開始します。

repairジョブのキャンセル

実行中のcompareまたはrepairジョブを停止するには、**cancel**コマンドを使用します。

```
sp_ctrl(sysA)>cancel JOBID
```

詳細については、「cancel」を参照してください。

compare履歴とログの管理

SharePlexは、終了した各ジョブの履歴をソースシステムのデータベースに保持します。SP_SYS_JOB_HISTORY_RETENTIONパラメーターは、履歴の保持期間を制御します。

この履歴を必要に応じて消去するには、**clear history**コマンドを使用します。SharePlexがジョブの履歴を削除すると、履歴の元となったログファイルも削除されます。

データベースのジョブ履歴を消去せずにソースシステムからログファイルを削除するには、**remove log**コマンドを使用します。このコマンドは、ターゲットシステムから古いログファイルを削除するためにも使用できます。

ログファイルのサイズを制御するには、SP_DEQ_LOG_FILESIZEパラメーターを設定します。

バッチサイズのコントロール

プロセスがSELECTクエリを実行するときにフェッチされる行のブロックのサイズを制御することができます。ブロックサイズは、**SP_DEQ_MALLOC**パラメーターで設定された値に基づいて計算されます。この値は、使用されるcompareスレッドの数で等分され、その後、すべての列のサイズを足した値に基づいて再計算されます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	compare / compare using

構文

基本コマンド	コマンドオプション	リモートオプション
repair *owner.source_table [.partition]*	[**at** *target_host*@**o.***target_sid*] |	[**on** *host* |
	[**for o.***source_sid*] |	**on** *host:portnumber* |
	[**hint** "*hint*"] |	**on** *login/password@host* |
	[{**include** | **exclude**} "*column_list*"] |	**on** *login/password@host:portnumber*]
	[**insertonly**] |	
	[**key**] |	
	[**nosqllog**] |	
	[**not** "*exception_list*"] |	
	[**onepass**] |	
	[**orderby** "*column_list*"] |	
	[**parallelism** *degree*] |	
	[**port***port_number*] |	
	[**sourcewhere** "*clause*"] |	
	[**threads** *thread_count*] |	
	[**targetwhere** "*clause*"] |	
	[**to** target_owner.target_table [.partition]] |	
	[**where** "*clause*"]	
repair using *filename*	[**key**] |	[**on** *host* |

基本コマンド	コマンドオプション	リモートオプション
	[**onepass**] |	**on** *host:portnumber* |
	[**port** *port_number*] |	**on** *login/password@host* |
	[**threads** *threads_count*] |	**on** *login/password@host:portnumber*]
	[**parallelism** *degree*]	

構文の説明

必須コマンドコンポーネント

コンポーネント	説明
repair *owner.source_table[.partition]*	基本コマンドは、すべてのソース行をすべてのターゲット行と共にrepairします。 *owner.source_table*は、ソーステーブルの所有者と名前です。大文字と小文字を区別したり、名前の中にスペースを入れたりするには、二重引用符を使います。例えば、**"HR".emp**。 ワイルドカードのテーブル名（オーナー名は不可）がサポートされています。repairするためには、このコマンドのワイルドカードを満たすテーブルがアクティブレプリケーション設定に（明示的またはワイルドカードで）リストされていなければなりません。SharePlexがワイルドカードをどのように処理するかについては、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl(sysA)>`**repair scott.emp**
repair using *filename*	基本コマンドは、*filename*にリストされているターゲットテーブル内の同期していない行をすべてrepairします。 *filename*は、ターゲットをrepairするソーステーブルの名前が含まれているファイルの名前です。 **例** `sp_ctrl(sysA)>`**repair using sales**

オプションのコマンドコンポーネント

コンポーネント	説明
at *target_host@o.target_sid*	**repairに有効** ソーステーブルが複数のターゲットシステムにレプリケートされる設定では、ターゲットテーブルの1つだけをrepairします。 *target_host*はターゲットシステムの名前です。

コンポーネント	説明
	*target_sid*はターゲットOracleインスタンスのORACLE_SIDです。 **例** `sp_ctrl(SysA)>`**repair scott.emp at prod@o.prodsid**
for o.*SID*	**repairに有効** ソーステーブルを含むOracleインスタンスを指定します。システム上の複数のOracleインスタンスに同じソーステーブルがある場合に使用します。 *SID*は、ソースインスタンスのORACLE_SIDです。大文字と小文字は区別され、**oratab**ファイルまたはV$PARAMETERテーブルに表示される通りに入力する必要があります。 このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**repair scott.emp for o.oraA**
hint "*hint*"	**repairに有効** SELECTステートメントにOracleヒントを含めます。ヒントはソースおよびターゲットシステム上で使用されます。 "*hint*"は2000文字以内のOracle標準のヒントです。ヒント全体を二重引用符で囲みます。ヒント文字列の先頭の/*+と末尾の*/を省略します。SharePlexによって追加されます。 このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**repair scott.emp where “file >001005” hint “emp(salary)”** オペレーティングシステムのコマンドラインからrepairを実行する場合、引用符で囲まれた文字列は、以下のように*エスケープされた*二重引用符で囲む必要があります。 */productdir/***bin/sp_ctrl repair scott.emp hint** “\“**emp(salary)**\””
{**include** | **exclude**} "(*column_list*)"	**repairに有効** repairする列をフィルタリングします。 • **include**を使って、repairする列を指定します。他の列はrepairされません。**include**句には、すべてのキー列を含める必要があります。

コンポーネント	説明
	• **exclude**で指定した列を除くすべての列をrepairするには、**exclude**を使用します。キー列を除外しないでください。 (*column_list*)は、インクルードまたは除外する列のリストです。 • それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションと共に任意の順序で記述できます。 **注:** repairされなかった列には、まだ同期していない行が残っている可能性があります。 **例** `sp_ctrl (SysA)>`**repair scott.emp exclude "color, weight"**
insertonly	**repairに有効** INSERTステートメントの場合にのみターゲットテーブルをrepairします。 **例** `sp_ctrl(SysA)>`**repair scott.emp insertonly**
key	**repairとrepair usingに有効** 大きなテーブルのcompareとrepairを高速で実行します。このコマンドは、すべてのデータ値をcompareするのではなく、以下のうちの1つだけをcompareします。 • PRIMARYキー列またはNULLでないUNIQUEキー列のみ。 または... • **orderby**オプションで指定された列。テーブルにキーがない場合は、このオプションを使用します。 キーまたは**orderby**列が一致しない場合、SharePlexは、行全体を削除してrepairし、ソース値に基づいて再度挿入します。 **重要!** このオプションの使用には注意が必要です。キーの値が一致していても、キー以外の列の値が同期していない可能性があります。

コンポーネント	説明
	このオプションを使用する場合は、必須コマンド引数の後に記述しなければなりません。他のオプションと共に任意の順序で表示できます。 このオプションは、SharePlexキー定義に基づくrepairには使用しないでください。SharePlexキー定義の詳細については、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl (SysA)>`**repair scott.emp key** `sp_ctrl(sysA)>`**repair using sales key**
nosqllog	SQLログファイルの出力を抑制します。このファイルには、同期していない行をrepairするために必要なSQLが含まれています。このファイルを出力しない理由には、以下のようなものがあります。 • データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。 • compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。
not “*exception_list*”	**repairに有効** テーブル指定にワイルドカードが含まれている場合に、repairしないテーブルの例外リストを指定します。 "*exception_list*"は、repairしないテーブル名のリストです。 • *owner.tablename*形式を使用します。 • それぞれの名前はカンマで区切ります。リストにスペースは許容されません。 • リストを二重引用符で囲みます。 • テーブルを任意の順序でリストします。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl(SysA)>`**repair scott.% not (%temp%)**
onepass	**repairとrepair usingに有効** compareとrepairを同時に実行するには、このオプションを使用します。同期していない大きなテーブルに使用します。

コンポーネント	説明
	通常、repairは2つのパスで実行されます。まずcompareを行い、次にrepairでターゲットテーブルをロックします。どちらのパスにも一貫したビューが必要です。**onepass**では、compareクライアントが一貫性のあるビューマーカーを受け取るとすぐに、ターゲットテーブルがロックされ、repairされます。 **例** `sp_ctrl(SysA)>`**repair scott.emp onepass**
orderby "*column_list*"	**repairに有効** repairプロセスでcompareする行をソートする際に、ORDERBY句で使用する列を指定します。このオプションは、プライマリキーや一意キーを持たないテーブル上でrepairを実行できるようにします。 "*column_list*"は、ORDERBY句で使用する列の名前です。 • それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションと共に任意の順序で記述できます。 • オペレーティングシステムのコマンドラインからrepairを実行する場合、引用符で囲まれた文字列は、以下のようにエスケープされた二重引用符で囲む必要があります。 */productdir/***bin/sp_ctrl repair scott.emp orderby "\"Last Name,Division\""** **例** `sp_ctrl(SysA)>`**repair scott.emp where "file >001005" orderby "Last Name,Division"**
parallelism*degree*	**repairとrepair usingに有効** SELECTステートメントに並列ヒントを追加します。*degree*では、並列度を設定します。 **例** `sp_ctrl(sysA)>`**repair scott.emp parallelism 4** `sp_ctrl(sysA)>`**repair using sales parallelism 4**
port *port_number*	**repairとrepair using**に有効

コンポーネント	説明
	ソースまたはターゲットシステム上のSharePlexのバージョンが8.0より前の場合、下位互換性のために使用できます。 クライアントプロセスがサーバプロセスとの通信に使用する、ソースシステム上のポートを指定します。SharePlexの以前のバージョンでは、通信は双方向であり、クライアントからサーバへの通信には、デフォルトでランダムなポート番号が選択されます。このオプションは、ファイアウォールが要求するような特定のポート番号で、ランダムなポート選択を上書きします。 **例** `sp_ctrl(sysA)>`**compare scott.emp port 1234**
sourcewhere “*clause*”	**repairに有効** ターゲットテーブルに列が存在しない場合に、ソーステーブル内の1つまたは複数の列を基にrepairを行います。 • *clause*を二重引用符で囲み、**scott.emp**のように完全修飾名でテーブルを参照します。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、OracleのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • オペレーティングシステムのコマンドラインからrepairを実行する場合、引用符で囲まれた文字列は、以下の例のようにエスケープされた二重引用符で囲む必要があります。 `sp_ctrl>`**repair scott.emp sourcewhere “\“file >001005\””** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例#1**: `sp_ctrl(sysA)>`**repair scott.emp sourcewhere "file >001005"** **例#2**: 次の例は、**sourcewhere**および**where**オプションをどのように組み合わせて、望ましい結果を得るかを示しています。ソースrepairプロセスのみが**sourcewhere**句を使用しますが、ソースrepairプロセスとターゲットrepairプロセスの両方が**where**句を使用します。

コンポーネント	説明
	`sp_ctrl(SysA)>`**repair scott.emp sourcewhere “deptno = 200” where “mgr = ‘SMITH’”**
targetwhere "*clause*"	**repairに有効** ソーステーブルに列が存在しない場合に、ターゲットテーブル内の1つまたは複数の列を基にrepairを行います。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、OracleのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列には、以下のようにエスケープされた二重引用符の余分なセットが必要です。 */productdir/***bin/sp_ctrl repair scott.emp targetwhere “\“file >001005\””** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例#1**: `sp_ctrl(SysA)>` **repair scott.emp targetwhere “file >001005”** **例#2**: 次の例は、**targetwhere**および**where**オプションをどのように組み合わせれば、望ましい結果が得られるかを示しています。ターゲットrepairプロセスのみが**targetwhere**句を使用しますが、ソースrepairプロセスとターゲットrepairプロセスの両方が**where**句を使用します。 `sp_ctrl(SysA)>`**repair scott.emp where “deptno = 200” targetwhere “mgr = ‘SMITH’” repair**
threads *thread_count*	**repairとrepair usingに有効** repairプロセスが使用する処理スレッドの数を設定します。 **例** `sp_ctrl(sysA)>`**repair scott.emp threads 4**

コンポーネント	説明
	`sp_ctrl(sysA)>`**repair using sales threads 4**
to *target_owner.target_table [.partition]*	**repairに有効** ソーステーブルのターゲットの1つだけをrepairします。ソーステーブルが複数のターゲットシステムにレプリケートされ、ターゲットテーブルに異なる名前がある場合に使用します。 このオプションはターゲットパーティションを指定するためにも使用できます。 `compare source_owner.source_table.[source_partition] to target_owner.target_table.[target_partition]` **例** （パーティションのrepair） `sp_ctrl(SysA)>`**repair scott.emp.east to scott.allemp.alleast**
where “*clause*”	**repairに有効** ソースシステムとターゲットシステムの両方で、SELECTステートメントにWHERE句を含めます。WHERE句は、特定の行をrepairするためのフィルタとして機能します。 “*clause*"には、サブクエリを含まない標準のWHERE句を指定します。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、OracleのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**repair scott.emp where “region=4”**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

repair status

repair statusコマンドを使用して、最後に実行されたcompareまたはrepairジョブのステータスを表示します。**repair statusコマンド**を使用すると、compareジョブやrepairジョブ、またはcompareジョブやrepairジョブの一部に関する詳細なステータスを表示したり、SharePlexに履歴があるすべてのcompareジョブおよびrepairジョブのステータスを表示したりできます。

repair statusコマンドの詳細および使用例については、**job statusコマンド**を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**copy status**、**append status**、**compare status**、**job status**

構文

基本コマンド	コマンドオプション	リモートオプション
repair status	[*job_id*]	[**on** *host* |
	[*job_id.table_id*]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |
	[**full**]	**on** *login/password@host:portnumber*]
	[**detail**]	
	[*status*]	

構文の説明

コンポーネント	説明
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)` > **repair status 2828.2**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)` > **repair status 2828.HR.SRC_TEST3**
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)` > **repair status all**

コンポーネント	説明
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)` > **repair status2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)` > **repair status detail**
status	指定されたステータスの過去のジョブのステータス履歴を表示します。 **例**: `sp_ctrl(SysA)` > **repair status "Error"**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

report

テーブルの**copy/append**の履歴を表示するには、このコマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**copy/append**

構文

基本コマンド	コマンドオプション	リモートオプション
report	*source_owner.source_table* [**copy** | **append**] [**toeditor**] [**for o.***source_sid*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
source_owner.source_table	このオプションは、同期レポートのソーステーブルを指定します。ソースオーナーとテーブル名のコンポーネント間にスペースを入れることはできません。 **例**: `sp_ctrl(SysA)>` **report s_user1.sync_single_tbl**
copy | append	このオプションは、**copy**または**append**のいずれかに固有の結果を返します。 **例**: `sp_ctrl(SysA)>` **report copy** s_user1.sync_single_tbl `sp_ctrl(SysA)>` **report append** s_user1.sync_single_tbl
toeditor	このオプションはターゲットを設定します。 **例**: `sp_ctrl(SysA)>` **report** s_user1.sync_single_tbl **toeditor** `sp_ctrl(SysA)>` **report copy** s_user1.sync_single_tbl **toeditor**

コンポーネント	説明
for o.*source_sid*	このオプションは、指定されたソースSIDに固有の結果を指定するために使用されます。 **例**: `sp_ctrl(SysA)` > **report** s_user1.sync_single_tbl **for** o.source_sid1 `sp_ctrl(SysA)` > **report copy** s_user1.sync_single_tbl **for** o.source_sid1

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

reset encryption key

SharePlexがデータの暗号化に使用しているAES暗号化キーを削除するには、**reset encryption key**コマンドを使用します。

SharePlexの暗号化の詳細な設定方法については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**create encryption key**、**set encryption key**、**show encryption key**

構文

基本コマンド	リモートオプション
reset encryption key	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の

オプション	説明
	構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

reset param

パラメーターをそのデフォルト値に復元するには、**reset param**コマンドを使用します。

変更は、以下のようにパラメーターのセットアットポイントに基づいて有効になります。

- *Live*のセットアットポイントの場合、変更はすぐに有効になります。
- *Restart Process*のセットアットポイントの場合、変更は、影響を受けるSharePlexプロセスが次に開始されたときに有効になります。
- セットアットポイントが*Restart Cop*の場合、変更は**sp_cop**の次回起動時に有効になります(システムの再起動は*行いません*)。

コマンドオプションの使用

以下は、SharePlexパラメーターをリセットするためのガイドラインです。

- **set param**コマンドで設定されたパラメーターをグローバルベースで(プロセス固有のオプションなしで)リセットするには、オプションを指定せずに**reset param**コマンドを使用します。
- *プロセス固有*の値(**set param**のプロセス固有オプションのいずれかで設定)を持つパラメーターをリセットするには、**reset param**のプロセス固有オプションのいずれかを使用します。プロセス固有のオプションは、以下のモジュールでパラメーターに使用できます。

モジュール	命名規則	制御対象の機能
export	SP_XPT	Exportプロセス
import	SP_IMP	Importプロセス
capture	SP_OCT	Captureプロセス
post	SP_OPO	Postプロセス
read	SP_ORD	Readプロセス

現在のパラメーター設定の表示

SharePlexパラメーターの名前、その値、セットアットポイント、およびプロセス固有の値が有効であるかどうかを表示するには、**list param**コマンドを使用します。以下の例では、Exportプロセスのグローバル値と、Export to sysBのプロセス固有の値の両方が示されています。

```
Parameter name                Actual value          Unit       Set at
----------------------        --------------------  ---------- ----------
SP_XPT_SO_SNDBUF              0                     bytes      Restart Process
SP_XPT_SO_SNDBUF              1024                  bytes      Restart Process
      To sysB.ABC.com
            Default Value: 0
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者（1）
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**list param**、**set param**

構文

基本コマンド	プロセス固有のオプション	リモートオプション
reset param {*paramname*	[**to** *host*]	[**on** *host* |
	[**from** *host*]	**on** *host:portnumber* |
	[**for** *datasource*]	**on** *login/password@host* |
	[**for** *datasource-datadest*]	**on** *login/password@host:portnumber*]
	[**queue** *queuename*]	

重要: このコマンドを使い始める前に、このドキュメントをすべて確認してください。使い方を誤ると、ターゲットデータが失われたり、同期が取れなくなったりする可能性があります。

構文の説明

コンポーネント	説明
paramname	デフォルト値に復元するパラメーターの名前。この引数では大文字と小文字を区別しません。 **例**: `sp_ctrl(sysA)>` **reset param SP_XPT_SO_SNDBUF** これにより、システム上のすべてのExportプロセスのパラメーターがリセットされます。

プロセス固有のコマンドオプション

オプション	説明
to *host*	このオプションは、指定されたターゲットシステムに対するExportプロセスのExportパラメーター（SP_XPT_）をリセットします。コマンドの効果を名前付きexportキューに関連付けられたExportプロセスに限定するには、このオプションに続けて[**queue** *queuename*]オプションを指定します。 **例**:

オプション	説明
	`sp_ctrl(sysA)` > **reset param sp_xpt_so_sndbuf to sysB** この例では、Export to sysBに対してパラメーターがリセットされますが、他のExportプロセスには影響しません。
from *host*	このオプションは、指定されたソースシステムからのImportプロセスのImportパラメーター(SP_IMP_)をリセットします。コマンドの効果を名前付きexportキューに関連付けられたImportプロセスに限定するには、このオプションに続けて[**queue** *queuename*]オプションを指定します。 **例**: `sp_ctrl(sysD)` > **reset param sp_imp_wcmt_msgcnt from sysC** この例では、Import from sys Cに対してパラメーターがリセットされますが、他のImportプロセスには影響しません。
for*datasource*	このオプションは、指定されたソースOracleインスタンスのCaptureまたはReadのパラメーター(SP_OCT_またはSP_ORD_)をリセットします。 • datasourceは**o**.*SID*で表されます。ここで、*SID*はソースインスタンスのORACLE_SIDです。 **例**: `sp_ctrl(sysA)` > **reset param sp_oct_replicate_dload for o.oraA** この例では、インスタンスoraAのCaptureプロセスに対してパラメーターがリセットされますが、他のCaptureプロセスには影響しません。
for *datasource-datadest*	このオプションは、あるデータベースから別のデータベースにデータをポストするすべてのPostプロセスに対してPostパラメーター(SP_OPO_)をリセットします。コマンドの効果を名前付きpostキューに関連付けられたPostプロセスに限定するには、このオプションに続けて[**queue** *queuename*]オプションを指定します。 • *datasource*は**o**.*SID*で表されます。ここで、*SID*はソースインスタンスのORACLE_SIDです。 • *datadest*は**o**.*SID*で表されます。ここで、*SID*はターゲットインスタンスのORACLE_SIDです。 **例**: `sp_ctrl(sysC)` > **reset param sp_opo_cont_on_err for o.oraA-o.oraC** この例では、インスタンスoraAのoraCへのレプリケートを行うPostプロセスに対してパラメーターがリセットされますが、他のPostプロセスには影響しません。
queue *queuename*	このオプションは、指定された名前付きキューに関連付けられたExport、ImportまたはPostプロセスのパラメーターをリセットします。

オプション	説明
	例: `sp_ctrl(sysA)` **> reset param sp_opo_poster_delay queue QA** この例では、名前付きpostキューQAに関連付けられているPostプロセスに対してパラメーターがリセットされますが、他のPostプロセスは影響を受けません。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

set encryption key

set encryption keyコマンドを使用して、**create encryption key**コマンドで生成されるAES暗号化キーを使用するようにExportおよびImportプロセスを設定します。

このコマンドを使用して、暗号化キーをソースシステムとターゲットシステムで設定する必要があります。

AES暗号化の設定の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**create encryption key**、**show encryption key**、**reset encryption key**

構文

基本コマンド	リモートオプション
set encryption key *key_value*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
key	**create encryption key**コマンドで生成された暗号化キー。
	例:
	例: `set encryption key E5F5D4CBA329D2C86B5D7ABA096C18600595490129F55A1422AAB0248B28D0E4`

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

set log

show logコマンドのデフォルトパラメーターを設定するには、**set log**コマンドを使用します。このコマンドは以下の場合のデフォルトを設定します。

- ログから抽出する最大行数。
- 画面に一度に表示する行の最大数。
- エントリを表示する方向（新しいものから古いものへ、または古いものから新しいものへ）。

このコマンドを使うと、これらのパラメーターのいずれか、あるいはすべてを一度に設定できます。このコマンドには[on host]オプションはありません。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー（3）
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**show log**、**view log options**

構文

基本コマンド	リモートオプション
set log {**maxlines=***number* | **lpp=***number* | **reverse** | **forward**}	利用不可

構文の説明

コンポーネント	説明
filespec	この引数は、コマンドの実行対象になるワイルドカード指定Oracleオブジェクト（スキーマの所有者を含む）を指定します。forデータソースオプションがない場合、スキーマとオブジェクトはレプリケーションされ、アクティブな設定内に含まれていると想定されます。 **例**: `sp_ctrl(sysA)>` **expand scott.%** この例では、scottスキーマ内のすべてのオブジェクトが結果に表示されます。
maxlines=*number*	このオプションを使用して、**show log**コマンドが表示するログの最大行数のデフォルトを設定します。このオプションは単独でも、他の**set log**コマンドオプションと組み合わせても使用できます。 **例**: `sp_ctrl(sysA)>` **set log maxlines=50**

コンポーネント	説明
	`sp_ctrl(sysA)` > **set log maxlines=50 lpp=34 reverse**
lpp=*number*	このオプションを使用して、**show log**コマンドの画面あたりの行数のデフォルトを設定します。このオプションは単独でも、他の**set log**コマンドオプションと組み合わせても使用できます。 **例**: `sp_ctrl(sysA)` > **set log lpp=34** `sp_ctrl(sysA)` > **set log maxlines=50 lpp=34 reverse**
reverse	このオプションを使うと、**show log**出力が最新のエントリと共に開始されます。このオプションは単独でも、他の**set log**コマンドオプションと組み合わせても使用できます。 **例**: `sp_ctrl(sysA)` > **set log reverse** `sp_ctrl(sysA)` > **set log maxlines=50 lpp=34 reverse**
forward	このオプションを使うと、**show log**出力が最も古いエントリと共に開始されます。このオプションは単独でも、他の**set log**コマンドオプションと組み合わせても使用できます。 **例**: `sp_ctrl(sysA)` > **set log forward** `sp_ctrl(sysA)` > **set log maxlines=50 lpp=34 forward**

set param

set paramコマンドを使用して、SharePlexパラメーターの値を変更します。

新しい設定は、以下のようにパラメーターのセットアットポイントに基づいて有効になります。

- *Live*のセットアットポイントの場合、変更はすぐに有効になります。
- *Restart Process*のセットアットポイントの場合、変更は、影響を受けるSharePlexプロセスが次に開始されたときに有効になります。
- *Restart Cop*のセットアットポイントの場合、変更は、**sp_cop**の次回起動時に有効になります。システムを再起動*しないでください*。

パラメーターのデフォルト値からの変更

パラメーターを*デフォルト*値から変更すると、SharePlexは、すべてのユーザ定義パラメーター値が保存されている変数データディレクトリに変更を記録します。デフォルト値は、製品ディレクトリの**param-defaults**ファイルに保存されます。

ほとんどのSharePlexパラメーターのデフォルト設定は、ほとんどのレプリケーション操作に適していますが、必要な場合に、SharePlexの正規ユーザが変更できるものもあります。これらのパラメーターは、**list param basic**コマンドで表示します。

残りのパラメーターは、ユーザが変更できないものとされています。これらは**list param all**コマンドで表示します。Quest開発チームまたはテクニカル・サポート・チームの監督なしに、またはSharePlexのガイドに文書化された手順に従わない限り、これらのパラメーターを変更しないでください。これらのパラメーターに精通していない限り、変更するとレプリケーションに悪影響を及ぼす可能性があります。

コマンドオプションの使用

以下では、set paramコマンドオプションの使い方を説明します。

- 基本的な**set param**コマンド（オプションなし）は、**sp_cop**のインスタンスにグローバルベースでパラメーターを設定します。例えば、以下のコマンドは、**sp_cop**の1つのインスタンスによって生成されるすべてのExportプロセスに対してパラメーターを設定します。

  ```
  sp_ctrl(sysA)> set param SP_XPT_SO_SNDBUF 1024
  ```

- 以下のレプリケーションプロセスのパラメーターは、*プロセス固有*にすることができます。これにより、**sp_cop**の1つのインスタンスから実行されるプロセスの複数のインスタンスに対応するために、同じパラメーターに異なる値を設定することができます。

モジュール	命名規則	制御対象の機能
export	SP_XPT	Exportプロセス
import	SP_IMP	Importプロセス
capture	SP_OCT	Captureプロセス
post	SP_OPO	Postプロセス
read	SP_ORD	Readプロセス

プロセス固有の値を、同じパラメーターのグローバル値と組み合わせて使用することができます。例えば、SP_XPT_SO_SNDBUFを、SysCに送信するプロセスを除くすべてのExportプロセスに対してグローバルに設定することができます。そのプロセスに対しては、[**to** *host value*]プロセス固有のオプションを使用して、別の値を設定することができます。

現在のパラメーター設定の表示

SharePlexパラメーターの名前、その値、セットアットポイント、およびプロセス固有の値が有効であるかどうかを表示するには、**list param**コマンドを使用します。以下の例では、Exportプロセスのグローバル値と、Export to sysBのプロセス固有の値の両方が示されています。

```
Parameter name                 Actual value           Unit        Set at
-----------------------        --------------------   -----------  -----------
SP_XPT_SO_SNDBUF               0                      bytes       Restart Process
SP_XPT_SO_SNDBUF               1024                   bytes       Restart Process
      To sysB.ABC.com
            Default Value: 0
```

パラメーターをデフォルト値に戻す

パラメーターの値をデフォルト値に変更するには、**reset param**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者(1)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**list param**、**reset param**

グローバル設定の構文

基本コマンド	リモートオプション
set param *paramname value*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

プロセス固有設定の構文

基本コマンド	コマンドオプション	リモートオプション

set param {
paramname **to** *host value* |
paramname **from** *host value* |
paramname **for** *datasource value* |
paramname **for** *datasource-datadest value*
}
	[**queue** *queuename*]	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

グローバル設定の構文の説明

コンポーネント	説明
paramname value	• *paramname*は、設定するパラメーターの名前です。この引数では大文字と小文字を区別しません。 • *value*は、パラメーターの新しい値です。 **例**: `sp_ctrl(sysA)` > **set param SP_XPT_SO_SNDBUF 1024** この例では、パラメーターはシステム上のすべてのExportプロセスに対してグローバルに 設定されています。

プロセス固有の設定に関する構文の説明

コンポーネント	説明
to *host value*	このオプションは、指定されたターゲットシステムに対するすべてのExportプロセスのExportパラメーター(SP_XPT_)を設定します。変更を名前付きexportキューに関連付けられたExportプロセスに制約するには、このオプションを[**queue***queuename*]オプションと共に使用します。 • *host*は、ターゲットシステムの名前です。 • *value*は、パラメーターの新しい値です。 **例**: `sp_ctrl(sysA)` > **set param sp_xpt_so_sndbuf to sysB 1024** この例では、Export to sysBに対してパラメーターが設定されますが、他のターゲットシステムには影響しません。

コンポーネント	説明
from *host value*	このオプションは、指定されたソースシステムからのすべてのImportプロセスのImportパラメーター（SP_IMP_）を設定します。変更を名前付きexportキューに関連付けられたImportプロセスに制約するには、このオプションを[**queue***queuename*]オプションと共に使用します。 • *host*は、ソースシステムの名前です。 • *value*は、パラメーターの新しい値です。 **例**: `sp_ctrl(sysD)` > **set param sp_imp_wcmt_msgcnt from sysC 10000** この例では、Import from sys Cに対してパラメーターが設定されますが、他のソースシステムからのImportプロセスには影響しません。
for *datasource value*	このオプションは、指定されたデータソースのCaptureまたはReadパラメーター（SP_OCT_またはSP_ORD_）を設定します。 • *datasource*は**o.***SID*で表わされます。ここで*SID*はソースOracleインスタンスのORACLE_SIDです。 • *value*は、新しいパラメーター値です。 **例**: `sp_ctrl(sysA)` > **set param sp_oct_replicate_dload for o.oraA 1** この例では、インスタンスoraAのCaptureに対してパラメーターが設定されますが、**sp_cop**の同じインスタンスに対する他のCaptureプロセスは影響を受けません。
for *datasource-datadest value*	このオプションは、あるデータベースから別のデータベースにデータをポストするすべてのPostプロセスに対してPostパラメーター（SP_OPO_）を設定します。変更を名前付きpostキューに関連付けられたPostプロセスに制約するには、このオプションを[**queue***queuename*]オプションと共に使用します。 • *datasource*は**o.***SID*で表わされます。ここで*SID*はソースOracleインスタンスのORACLE_SIDです。 • *datadest*は**o.***SID*で表わされます。ここで、*SID*はターゲットOracleインスタンスのORACLE_SIDです。 • *value*は、新しいパラメーター値です。 **例**: `sp_ctrl(sysC)` > **set param sp_opo_cont_on_err for o.oraA-o.oraC 1** この例では、oraCにレプリケートされるインスタンスoraAのPostプロセスに対してパラメーターが設定されますが、他のPostプロセスには影響しません。
queue *queuename*	このオプションは、コマンドの効果を、名前付きキューに関連付けられた

コンポーネント	説明
	Export、ImportまたはPostプロセスに制限します。 • *queuename*は名前付きキューの名前です。 使用される場合、**queue** *queuename*オプションは、以下のように、必須構文とオプション構文の後、および新しい値の前に表示されなければなりません。 **例1: Exportプロセス** `sp_ctrl(sysA)` **> set param sp_xpt_so_sndbuf queue QA to sysB 1024** この例では、名前付きexportキューQAに関連付けられているExportプロセスに対してパラメーターが設定されますが、他のExportプロセスは影響を受けません。 **例2: Importプロセス** `sp_ctrl(sysD)` **> set param sp_imp_wcmt_msgcnt from sysC queue QA 10000** この例では、名前付きexportキューQAに関連付けられているImportプロセスに対してパラメーターが設定されますが、他のImportプロセスは影響を受けません。 **例3: Postプロセス** `sp_ctrl(sysC)` **> set param sp_opo_cont_on_err queue QA for o.oraA-o.oraC 1** この例では、oraAとoraC間のレプリケーションのために、名前付きpostキューQAに関連付けられているPostプロセスに対してパラメーターが設定されますが、他のPostプロセスは影響を受けません。 **例4: Postプロセス** `sp_ctrl(sysC)` **> set param sp_opo_cont_on_err queue QA 1** この例では、名前付きpostキューQAに関連付けられているPostプロセスに対してパラメーターが設定されますが、他のPostプロセスは影響を受けません。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

showコマンド

sp_copのインスタンスのレプリケーションプロセスとデータのパスを表示するには、**show**コマンドを使用します。

以下で、showコマンド表示での各列について説明します。

Process（プロセス）

Process（プロセス）列には、現在の**sp_ctrl**セッションが接続されている**sp_cop**のインスタンスによって生成されたSharePlexレプリケーションプロセス（サービス）が表示されます。コマンドは、コマンドと制御プロセス（**sp_cnc**）または実行中のcompareプロセスを表示しません。

- 名前付きexportキューがある場合は、ソースまたはターゲットの**sp_cop**のどちらかに応じて複数のExportまたはImportプロセスが存在します。
- 名前付きpostキューが存在する場合、Postプロセスは複数存在します（各キューに1つずつ）。
- Postプロセスのラベルは`MTPost`です。
- ピアツーピアレプリケーションを実行している**sp_cop**のインスタンス、またはソースとターゲットの両方の**sp_cop**インスタンスとして機能するインスタンスの場合は、すべてのSharePlexレプリケーションプロセスが存在します。

Source（ソース）

Source（ソース） 列は、SharePlexプロセスで処理されるデータのソースを示します。ソースには以下のいずれかが該当します。

- *CaptureまたはReadプロセスの場合*: データソース（ソースOracleインスタンス）。
- *Exportプロセスの場合*: 関連するexportキューの名前。デフォルトのexportキューには、ソースシステムの名前が付きます。名前付きexportキューには、例えばexptq1のように、ユーザによって割り当てられた名前が付きます。
- *Importプロセスの場合*: 関連するデフォルトのexportキューの名前（例えばsysA）、あるいはexptq1のような、関連する名前付きexportキューの名前。
- *Postプロセスの場合*: Postプロセスのソースでは、以下のようにレプリケーション設定に応じて、いくつかの表示のいずれか。
    - *デフォルト*のexportキューとデフォルトのpostキューがある場合、Postソースは関連するpostキュー の短縮名となります。短縮名は、データソースとソースシステム名で構成され、o.ora10-elliotの形式が使用されます。
    - 名前付きexportキューがある場合（ターゲットシステム上に名前付きpostキューが自動的に作成される）、Postソースはデータソースと名前付きexportキューの名前で構成され、o.ora10-exptq1の形式が使用されます。
    - デフォルトのexportキューと名前付きpostキューが存在する場合、Postソースはデータソースと名前付きpostキューの名前で構成され、例えば、o.ora10- postq1のようになります。

Target（ターゲット）

Target（ターゲット）列は、SharePlexプロセスによって処理されるデータの宛先、つまりターゲットを示します。

ターゲットには以下のいずれかが該当します。

- *CaptureまたはReadプロセスの場合*: これらのプロセスには、表示するターゲットがありません。Captureプロセスでキャプチャされ、Readプロセスでルーティングされたデータは、1つまたは複数の宛先にルーティングすることができます。
- *Exportプロセスの場合*: ターゲットシステムの名前。
- *Importプロセスの場合*: ターゲットシステムの名前。
- *Postプロセスの場合*: ターゲット。

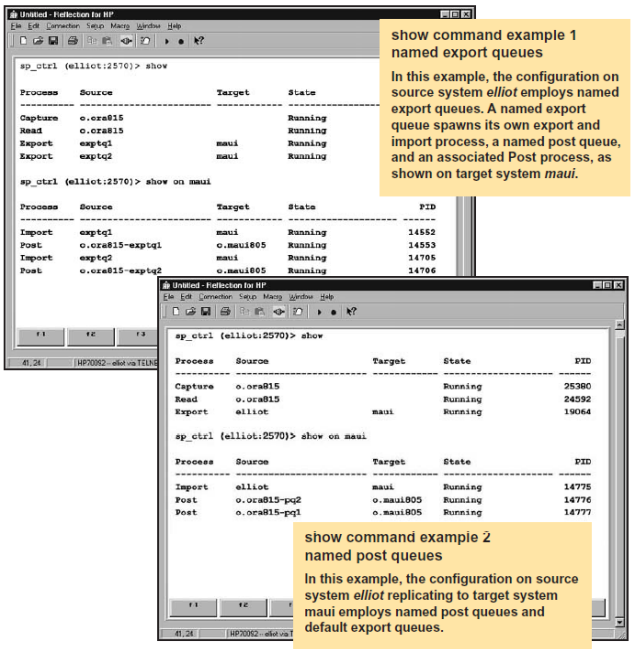
State（状態）

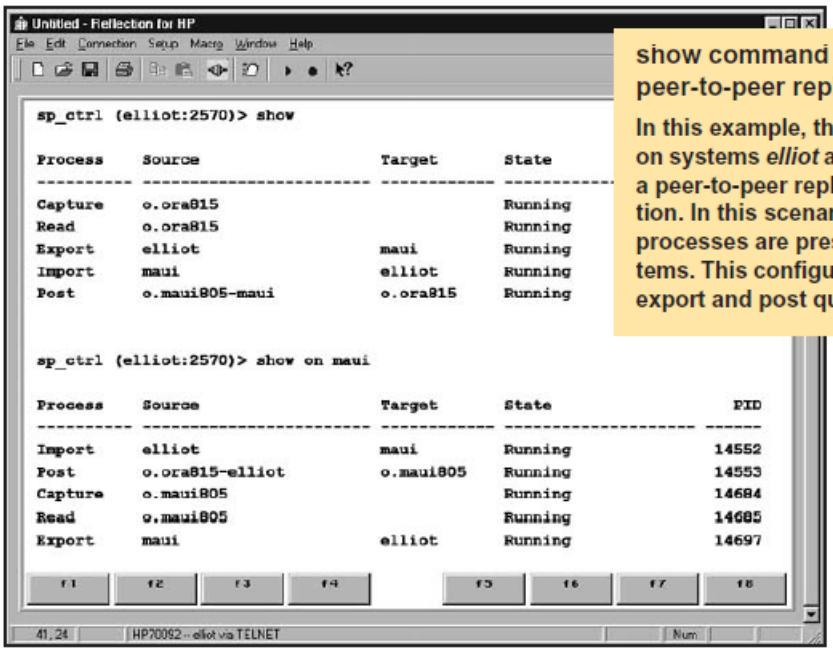
State（状態）列は、以下のように各プロセスのステータスを示します。

- 実行中
- アイドル状態（データ処理待ち）
- ユーザが停止
- エラーにより停止。

PID

この列には、各プロセスのプロセスID番号がリストされます。





show command example 3
peer-to-peer replication

In this example, the configurations on systems *elliot* and *maui* are part of a peer-to-peer replication configuration. In this scenario, all SharePlex processes are present on both systems. This configuration uses default export and post queues.

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**lstatus**、**qstatus**、**status**

構文

基本コマンド	リモートオプション
show	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show captureの使用

Captureプロセスの統計情報を表示するには、**show capture**コマンドを使用します。

基本コマンド

基本的な**show capture**コマンドは、データソース、プロセスが実行中か停止中、その他の基本情報など、プロセスの概要を示します。

詳細な統計情報

Captureプロセスの詳細な統計情報を表示するには、**[detail]**オプションを指定した**show capture**コマンドを使用します。このオプションは、プロセスのパフォーマンスを評価し、チューニングパラメーターを調整する必要があるかどうかを決定し、問題やボトルネックを検出するのに役立つ詳細な統計情報を表示します。

Oracle Capture用の詳細な統計情報

統計情報	説明
Host（ホスト）	ローカルマシンの名前（ソースシステム）
System time（システム時刻）	システムクロックによる現在の時刻。
Source（ソース）	ソースOracleインスタンスの名前。
Status（ステータス）	Captureプロセスのステータス（実行中または停止中）。
Since（開始時刻）	Captureが開始された時刻。
Oracle current redo log（Oracleの現在のREDOログ）	Oracleが書き込むREDOログのシーケンス番号。
Capture current redo log（Captureの現在のREDOログ）	Captureが読み込んでいるREDOログのシーケンス番号。
Capture log offset（Captureのログオフセット）	Captureが処理するレコードのREDOログ内の場所。
Last redo record processed（最後に処理されたREDOレコード）	Captureが処理中のレコード、またはCaptureが現在データをレプリケートしていない場合は最後に処理されたレコード。
Capture state（Captureの状態）	実行するレプリケーション作業に関連するプロセスの状態 **以下のいずれかが該当します。** • 初期化中: Captureが開始されます。 • 待機中: CaptureはREDOログを読み取り、レプリケートする必要があるレコードを待機しています。 • ログファイル待ち: Captureは、必要なログにアクセスできない場合、待機してから再試行するように設定されています。 • 処理中: CaptureはレプリケーションのためにREDOログレコードを処理しています。

統計情報	説明
	• アーカイブログが見つからないために停止した: Captureは、必要なログにアクセスできない場合に停止するように設定されています。
Activation ID（アクティベーションID）	設定のアクティベーションの内部識別番号です。関連するプロセスとキューを識別します。
Error count（エラー数）	Captureの開始後、Oracleのエラーによりスキップされたレコードの数。スキップされたレコードのデータはターゲットデータベースに反映されません。
Operations captured（キャプチャされたオペレーション数）	開始後に、Captureがレプリケーション用に正常に処理したREDOレコードの数。
Transactions captured（キャプチャされたトランザクション数）	開始後に、Captureがオペレーションを正常にレプリケートしたコミット済みOracleトランザクションの数。
Concurrent sessions（同時セッション数）	同時に処理されているOracleセッションの数。
HWM concurrent sessions（HWM同時セッション数）	Capture開始後の同時Oracleセッションの最大数。
Checkpoints performed（実行されたチェックポイント数）	Capture開始後のCaptureの状態を保存するチェックポイントの数。頻繁にチェックポイントを実行すると、システムにさらなるオーバーヘッドが発生しますが、チェックポイントの頻度が低いと、SharePlexでのシステムやインスタンスの障害からのリカバリが遅くなります。デフォルトでは、Captureは2,000件のメッセージごとにチェックポイントを実行しますが、SP_OCT_CHECKPOINT_FREQパラメーターで調整できます。
Total operations processed（処理されたオペレーションの総数）	レプリケーション用にキャプチャされたレコードや、設定に含まれていないオブジェクトのレコードなど、Captureが開始後に処理したすべてのOracleオペレーションとSharePlex内部オペレーションの数。
Total transactions completed（完了したトランザクションの総数）	レプリケーション用にキャプチャされたトランザクションや、レプリケーション設定に含まれていないオブジェクトのトランザクションなど、Captureが開始後に処理したコミット済みOracleトランザクションの数。
Total Kbytes read（読み取られたキロバイト総数）	Captureが開始後に処理したデータのキロバイト単位のサイズ。
Redo records in progress（処理中のREDOレコード数）	Captureが処理中のレコードの数。
Redo records processed（処理されたREDOレコード数）	処理されたREDOレコードの総数。
Redo records ignored（無視されたREDOレコード数）	設定内のオブジェクトに関連付けられていないためにCaptureが無視したレコードの数。
Redo records - last HRID（REDOレコード - 最後のHRID）	Captureが最後に処理した連鎖行の先頭行ID値。

注意: RACシステム用の**show capture detail**出力は、ここで説明されているものとは若干異なります。情報は基本的に同じですが、必然的に表現が異なります。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**show export**、**show import**、**show post**、**show read**

構文

基本コマンド	コマンドオプション	リモートオプション
show capture	[**detail**] [**for** *datasource*]	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
show capture	Captureプロセスの状態とキャプチャされた操作の概要を表示します。
detail	Captureのパフォーマンスを調整したり、問題を診断したりするのに役立つ詳細な統計情報が表示されます。 **例**: `sp_ctrl(sysA)` > **show capture detail**
for *datasource*	このオプションは、特定のデータソースのCapture統計情報のみを表示します。 *datasource*は**o.***SID*と表します。ここで、*SID*はORACLE_SIDです。 **例**: `sp_ctrl(sysA)` > **show capture for o.oraA**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show config

アクティブな設定の統計情報を表示するには、**show config**コマンドを使用します。

例:

```
sp_ctrl (irvspxu14:8567)> show config

Materialized Views Replicating:
     “SCOTT”.”V5_ROWID”
Tables Replicating with Key:
     “SCOTT”.”S1TEST2” KEY: KC
     “SCOTT”.”S2TEST2” KEY: KC
     “SCOTT”.”T1TEST2” KEY: KC
     “SCOTT”.”T2TEST2” KEY: KC
Tables Replicating with no Key:
     “TED”.”DEMO_SRC”
     “TED”.”DEMO_DEST”
     “SCOTT”.”FOOS”

     filename : allscott
     Datasource : o.ora920
     Activated : 29-Jan-09 10:11:41
     Actid : 1810

     Total Objects : 160
     Total Objects Replicating : 160
     Total Objects Not Replicating : 0
     View config summary in /splex/rr/svn/var92/log/ora920_config_log
```

以下はCaptureが起動する前にこのコマンドを実行すると表示されます。

```
sp_ctrl (irvspxu14:8567)> show config

Capture is not yet replicating the most recent activation
Re-issue this command when capture’s replication is current
```

Show Configは以下の統計情報を提供します。

- **Config（設定）**: 設定名。
- **Datasource（データソース）**: レプリケートされるオブジェクトを含むOracleインスタンス。
- **Activated（アクティベーション日時）**: 設定がアクティベーションされた日時。
- **Actid（アクティベーションID）**: 設定のアクティベーションID番号（主にQuestソフトウェアテクニカルサポートが使用）。
- **Total Objects（オブジェクト総数）**: 設定ファイル内のオブジェクトの数。
- **Total Objects Replicating（レプリケートしているオブジェクト総数）**: レプリケートしている設定内のオブジェクトの数。
- **Total Objects Not Replicating（レプリケートしていないオブジェクト総数）**: アクティベーションに失敗したため、レプリケートされていない設定内のオブジェクトの数。
- レプリケーション対象でプライマリキーまたは一意キーを持つテーブル、レプリケーション対象でキーを持たないテーブル、およびレプリケーション対象外のテーブルのリスト。

設定のオブジェクトの詳細については、*SID*_**config_log**を参照してください。ここで、*SID*はインスタンスのORACLE_SIDです。SharePlexはアクティベーションの結果とエラーメッセージを、SharePlex変数データディレクトリのlogサブディレクトリにあるこのファイルに出力します。

ログファイルの例を以下に示します。

```
File Name :wild

Datasource :ora920

Activated :29-Jan-09 10:11:41

Actid :1810


Total Objects :9

Total Objects Replicating :9

Total Objects Not Replicating :0


Objects Replicating:

"SCOTT"."SRC_TEST1"

"SCOTT"."SRC_TEST2"

"SCOTT"."SS2_TEST1"

"SCOTT"."SRC_TEST3"

"SCOTT"."SRC_TEST4"

"SCOTT"."SS2_TEST2"

"SCOTT"."SRC_TEST5"

"SCOTT"."SRC_TEST6"

"SCOTT"."SS2_TEST3"
```

```
Objects Not Replicating:
```

注意: 統計情報の表示には数秒またはそれ以上かかる場合があります。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**activate config**、**view config**、**verify config**

構文

基本コマンド	リモートオプション
show config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	統計情報を表示する設定の名前。設定名では大文字と小文字が区別されます。 **例**: `sp_ctrl(sysA)` > **show config sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。

オプション	説明
	例: `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show encryption key

SharePlexで使用されているAES暗号化キーを表示するには、**show encryption key**コマンドを使用します。SharePlex暗号化の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**create encryption key**、**set encryption key**、**reset encryption key**

構文

基本コマンド	リモートオプション
show encryption key	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。

オプション	説明
	例:`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show export

Exportプロセスに関する統計情報を表示するには、**show export**コマンドを使用します。このコマンドは、ローカルマシン上のすべてのExportプロセスによってターゲットシステムに送信されたメッセージ数のレコードと、Exportのパフォーマンスに関するオプションの統計情報を保持します。Exportが停止し、再び開始されるたびに、メッセージのカウントがやり直されます。

注意: 通常の場合、メッセージはほぼSQL操作に対応しますが、LONG列やLOB列に対する1つの操作に対して複数のメッセージが存在することもあります。また、配列挿入の多数の操作に対して1つのレコードしか存在しないこともあります。また、メッセージはSharePlexの内部操作に対応する場合もあります。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**show capture**、**show import**、**show post**、**show read**

構文

基本コマンド	コマンドオプション	リモートオプション
show export	[**detail**]	[**on** *host* |
	[**queue** *queuename*]	**on** *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
queue *queuename*	出力を特定の名前付きExportキューに制限します。
detail	以下の統計情報を表示します。 • Exportプロセスが有効化された後の最高レートとそれが行われた日時 • Exportが開始された後の最高レートとそれが行われた日時 • 現在のレートはKB/秒として報告され、60秒のタイムスライスにおける総KBとして測定されます。 • 直近60秒のタイムスライス中の平均パケットサイズ

コンポーネント	説明
	• 直近60秒のタイムスライス中に送信されたパケットの数 • 直近60秒のタイムスライス中の平均TCP送信時間 • 直近60秒のタイムスライス中の平均からの最大偏差 • 直近の60秒間のタイムスライス中に50%以上逸脱したパケットの数

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

例

```
Host   : abcl23
Queue  : abcl23
                          Kbytes
Target     Status         Exported     Since              Total       Backlog
---------- -------------- ------------ ------------------ ---------- ----------
abcl23.abc Running                1063 18-Aug-14 17:05:29       1082          0

  Since activated
    Transmission rate                     : 5 Kb/sec
    Highest transmission rate             : 10 Kb/sec Mon Aug 18 14:45:49 2014
    Average packet size                   : 680 bytes
    Average send time                     : 68 microseconds

  Since export started
    Transmission rate                     : 4 Kb/sec
    Highest transmission rate             : 2 Kb/sec Mon Aug 18 17:07:18 2014
    Average packet size                   : 615 bytes
    Average send time                     : 16 microseconds

  During the last 27 seconds:
    Transmission rate                     : 25 Kb/sec
    Average packet size                   : 678 bytes
    Number of packets                     : 1081
    Average send time                     : 19 microseconds
    Largest deviation from average        : 6002 microseconds
    Send times deviating by more than %23 : %0.9
```

show import

Importプロセスに関する統計情報を表示するには、**show import**コマンドを使用します。このコマンドは、ローカルマシン上のすべてのImportプロセスがソースシステムから受信したメッセージ数のレコードを保持します。Importが停止し、再び開始されるたびに、メッセージのカウントがやり直されます。

注意: 通常の場合、メッセージはほぼSQL操作に対応しますが、LONG列やLOB列に対する1つの操作に対して複数のメッセージが存在することもあります。また、配列挿入の多数の操作に対して1つのレコードしか存在しないこともあります。また、メッセージはSharePlexの内部操作に対応する場合もあります。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**show capture**、**show export**、**show post**、**show read**

構文

基本コマンド	コマンドオプション	リモートオプション
show import	[**queue** *queuename*]	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
queue *queuename*	出力を特定の名前のimportキューに制限します。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要

オプション	説明
	素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show log

show logコマンドを使用すると、オペレーティングシステムから直接ログファイルを開くのではなく、**sp_ctrl**インターフェイスを通して、SharePlexイベントログまたはユーザ発行コマンドを表示することができます。このコマンドは、サイズが2 MB未満のログに使用することができます。それ以外の場合は、ファイルシステムから直接ログを表示します。オプションなしで使用されるデフォルトのコマンドは、イベントログから60行を1ページあたり15行で画面に表示します。最も古いエントリから開始されます。

- 現在のshow logのデフォルトパラメーターを表示するには、**view log**のオプションコマンドを使用します。
- デフォルトを変更するには、**set log**コマンドを使用します。

set logのデフォルトをオーバーライドするには、以下のようにオプションの構文を使用できます。

- エントリの順序の設定
- ログから抽出された行の数
- 画面上の行数
- キーワードによるエントリの絞り込み
- 完全なイベントログではなく、ユーザが発行したコマンドの表示

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**set log**、**view log options**

構文

基本コマンド	コマンドオプション	リモートオプション
show log	[**event** | **command** | **trace** | **post for** *datadest* | **capture for** *datasource*| **read for** *datasource*] [**maxlines=***number*] [**lpp=***number*] [**reverse**] [**forward**] [**filter=***keyword*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

注意: このコマンドで*datasource*、*datadest*、*queuename*の値を取得するには、**lstatus**コマンドを使用して、出力の**Queues**セクションを表示します。

コンポーネント	説明
event	イベントログを表示するには、このオプションを使用します。このオプションを使用する場合、構文では他のどのオプションよりも前に記述しなければなりません。**[command]**オプションとは併用できません。 **例**: `sp_ctrl(sysA)` > **show log event**
command	ユーザが発行したコマンドを表示するには、このオプションを使用します。このオプションを使用する場合、構文では他のどのオプションよりも前に記述しなければなりません。**[event]**オプションとは併用できません。 **例**: `sp_ctrl(sysA)` > **show log command**
post for *datadest* **queue** *queuename*	Postログを表示するには、このオプションを使用します。このオプションでは、データ送信先とキュー名を指定する必要があります。 **例**: `sp_ctrl(sysA)` > **show log post for mydbqueue q1**
capture for *datasource*	Captureログを表示するには、このオプションを使用します。このオプションではデータソースを指定する必要があります。 **例**: `sp_ctrl(sysA)` > **show log capture for o.mySID**
read for *datasource*	Readログを表示するには、このオプションを使用します。このオプションではデータソースを指定する必要があります。 **例**: `sp_ctrl(sysA)` > **show log read for o.mySID**
maxlines=*number*	このオプションを使用して、ログから抽出する最大行数を指定します。このオプションがない場合、show logのデフォルトは60行またはset logコマンドで設定された値になります。 **例**: `sp_ctrl(sysA)` > **show log maxlines=50**
lpp=*number*	このオプションを使用して、モニター画面に合わせて出力をカスタマイズします。画面に表示する行数を指定します。このオプションがない場合、**show log**のデフォルトは15行または**set log**コマンドで設定された値になります。 **例**:

コンポーネント	説明
	`sp_ctrl(sysA)` > **show log lpp=34**
reverse	このオプションを使用すると、最新のエントリから順番に表示されます。このオプションがない場合、**show log**はデフォルトで順方向になるか、**set log**コマンドで設定された値になります。 **例**: `sp_ctrl(sysA)` > **show log reverse**
forward	このオプションを使用すると、最も古いエントリから順番に表示されます。このオプションがない場合、**show log**はデフォルトで順方向になるか、**set log**コマンドで設定された値になります。 **例**: `sp_ctrl(sysA)` > **show log forward**

filter=*keyword*	キーワードに基づいてログエントリをフィルタするには、このオプションを使用します。このオプションを使うには次の2つの方法があります。
	• キーワードを*含む*行のみを表示するには、**filter=***keyword*と入力します
	• キーワードを含む行を*除外*するには、**filter=!***keyword*のように、キーワードの前に感嘆符(!)を入力します。
	キーワードに空白を含めることはできません。
	例:
	`sp_ctrl(sysA)` > **show log filter=compare**
	前述の例では、Eventログからcompare関連のメッセージのみを抽出しています。
	`Notice 08-07-08 22:47:21.906001 96492 1 User command: qarun remove log all (from irvqasu21.quest.com)`
	`sp_ctrl(sysA)` > **show log filter=!Notice**
	前述の例では、Eventログからすべての**Notice**エントリを*除外*しますが、他のすべてのタイプのエントリが表示されます。
	`Info 08-07-08 22:47:19.642379 96490 1 Command server launched, pid = 96490 (connecting from irvqasu21.quest.com)`
	`Info 08-07-08 22:47:20.825598 96492 1 Command server launched, pid = 96492 (connecting from irvqasu21.quest.com)`
	`Info 08-07-08 22:47:22.334040 23710 1 Command server launched, pid = 23710 (connecting from`

コンポーネント	説明
	irvqasl03.quest.com) Info 08-07-08 22:47:23.969925 125996 1 Compare server launched, pid = 125996 Info 08-07-08 22:47:24.632481 125996 1 Compare server completed

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show post

Postプロセスの統計情報を表示するには、**show post**コマンドを使用します。

基本的なshow postコマンド

基本的な**show post**コマンドは、Postプロセスの全セッションのグローバル統計情報を表示します。Postプロセスのステータスと、そのプロセスが開始されてからポストされたメッセージの数が表示されます。特定のpostキューまたはデータソースの出力をフィルタリングするには(複数のレプリケート対象データストリームがある場合に便利)、**queue***queuename*または**for***datasourcedatadest*オプションを使用します。

詳細なshow postコマンド

Postプロセスの詳細な統計情報を表示するには、**show post**コマンドに**detail**オプションを指定します。このオプションは、処理された最新のSQLステートメントと、Postのパフォーマンスを評価し、チューニングパラメーターを調整する必要があるかどうかを決定し、問題やボトルネックを検出するのに役立つその他の統計情報を表示します。

以下では、**show post**で表示される詳細な統計情報の例について説明します。これらの統計情報は、ソースとターゲットのタイプによって若干異なります。

統計情報	説明
Host(ホスト)	ローカルマシンの名前(ターゲットシステム)
Source(ソース)	Postが処理するデータのソース。
Queue(キュー)	このPostプロセスのpostキュー。デフォルトのpostキューの場合は、ソースシステムの名前です。名前付きキューの場合は、ユーザ定義名です。
Target(ターゲット)	このPostプロセスのターゲットの名前。例えば、Oracleインスタンスまたは Open Targetデータベースの名前です。
Status(ステータス)	Postプロセスのステータス(実行中または停止中)。可能なステータスは以下の通りです。 • 実行中 • 停止中 • ユーザが停止 • フラッシュにより停止 • エラーにより停止
Operations posted(ポストされた操作数) **Operations processed(処理された操作数)**	このPostプロセスが開始後に処理したトランザクション操作とSharePlex内部操作の数。
Since(開始時刻)	Postが開始された時刻
Total(合計)	キュー内のまだ読み取り-リリースされていないメッセージの数。この数字は、**qstatus**を実行して返された「メッセージ数」に相当します。
Backlog(バックログ)	Postが処理するのをキューで待機しているメッセージの数。

統計情報	説明
Last operation posted（ポストされた最後の操作）	Postがアクティブな場合はターゲットにポストされている最新の操作、非アクティブな場合は最後にポストされた操作の識別情報。この情報は、データを生成したデータストアのタイプに固有です。操作の内容は以下の通りです。 • INSERT • UPDATE • DELETE • TRUNCATE • COMMIT • DDLステートメント • INSERT_MULTIPLEまたはDELETE_MULTIPLE（配列操作）。 • SharePlex内部操作。
Last transaction posted（ポストされた最後のトランザクション） **Last transaction processed（処理された最後のトランザクション）**	最後にポストされたトランザクションの識別情報。この情報は、データを生成したデータストアのタイプに固有です。
Last file switch（最新ファイルスイッチ）	ターゲットがファイルに出力される場合に表示されます。アクティブファイルが新しいファイルに切り替わった後のレコードのレコード識別子を表示します。
Post state（Postの状態）	実行対象のレプリケーション作業に関連するPostプロセスの状態。以下のいずれかが該当します。 • **Waiting（待機中）**: Postはメッセージが処理されるまで待機しています。 • **Active（アクティブ）**: Postは変更をデータベースにポストしています。 • **Committed（コミット済み）**: Postはトランザクションをコミットしています。 • **Idle（アイドル）**: Postには処理する未決済トランザクションがありません。 • **Rollback（ロールバック）**: Postはロールバックを処理中です。 • **Recovery（リカバリ）**: Postはクラッシュリカバリモードになっています。
Activation ID（アクティベーションID）	現在の設定のアクティベーションID。
Number of messages read released（読み取り-リリースされたメッセージの数）	Oracleターゲットの場合に表示されます。ポストされた操作の数と、実際にキューから削除された操作の数の差を示します。このフィールドは、Postが停滞しているときに主にテクニカル・サポート・スタッフが使用します。

統計情報	説明
Number of threads（スレッドの数）	Oracleターゲットの場合に表示されます。メインスレッドとタイムキーパースレッドを含む、マルチスレッドPost内の処理スレッド数を示します。
Number of Oracle connections（Oracle接続の数）	Oracleターゲットの場合に表示されます。マルチスレッドPostが開いている接続の数を表示します。
Concurrency (Active sessions)（同時実行（アクティブセッション））	Oracleターゲットの場合に表示されます。マルチスレッドPostが処理している同時トランザクションの数を示します。
Peak number of sessions（セッションのピーク数）	Oracleターゲットの場合に表示されます。マルチスレッドPostが開始されてから処理した同時トランザクション数の最高値を示します。
Operations posted（ポストされた操作数） **Operations processed（処理された操作数）**	COMMITを受け取ったかどうかにかかわらず、Postがターゲットに適用したSQL操作の数。
Transactions posted（ポストされたトランザクション数） **Transactions processed（処理されたトランザクション数）**	開始以後にPostがターゲットに適用したコミット済みトランザクションの数。
Full rollbacks（フルロールバック数）	Oracleターゲットの場合に表示されます。Postで処理されたロールバックトランザクションの数を示します。
Full rollback operations posted（ポストされたフルロールバック操作数）	Oracleターゲットの場合に表示されます。トランザクションがロールバックされる前にPostが適用した操作の数を表示します。（Postは、キュー内の操作の適用を開始するためにコミットを待機しません）
Full rollback operations skipped（スキップされたフルロールバック操作数）	Oracleターゲットの場合に表示されます。トランザクションがロールバックされたことが早期に通知されたため、Postがスキップした操作の数を示します。
Transactions <= 2（2以下のトランザクション数）	Oracleターゲットの場合に表示されます。2つ以下の操作を含むトランザクションの数を示します。
Transactions > 10000（10000を超えるトランザクション数）	Oracleターゲットの場合に表示されます。10,000を超える操作を含むトランザクションの数を示します。
Largest transaction（最大トランザクション）	Oracleターゲットの場合に表示されます。処理された最大トランザクションのサイズを示します。
Insert operations（INSERT操作数）	Postが開始後に処理したINSERT操作の数。
Update operations（UPDATE操作数）	Postが開始後に処理したUPDATE操作の数。
Delete operations（DELETE操作数）	Postが開始後に処理したDELETE操作の数。
Insert batch operations / average（INSERT_BATCH操作数/平均）	Oracleターゲットの場合に表示されます。処理されたINSERT_BATCH操作の数/1つのINSERT_BATCHステートメント内のINSERT操作の平均数を示します。
Delete batch operations / average（ELETE_BATCH操作数/平均）	Oracleターゲットの場合に表示されます。処理されたDELETE_BATCHステートメントの数/1つのDELETE_BATCHステートメント内のDELETE操作の平均数を示します。
Other operations（その他の操作数）	Oracleターゲットの場合に表示されます。Postが開始後に処理した、

統計情報	説明
	INSERT、UPDATE、DELETE以外の操作の数を示します。
LOB changes（LOB変更数）	Oracleターゲットの場合に表示されます。LOB変更操作の回数を示します。
Key cache hit count（キー・キャッシュ・ヒット数）	Oracleターゲットの場合に表示されます。UPDATEステートメントを生成するためにPostプロセスで使用されたSharePlex内部キーキャッシュの回数を示します。SharePlexは、キーキャッシュを使用してデータをターゲットインスタンスに迅速にポストします。
SQL cache hit count（SQLキャッシュヒット数）	Oracleターゲットの場合に表示されます。これは、*解析とバインディングを行わずに実行されたメッセージの総数*を、*INSERT*、*UPDATE*、および*DELETE操作の総数*で割った比率を示しています。SharePlexのSQL Cache機能の詳細については、『SharePlex管理ガイド』を参照してください。
File switches（ファイルスイッチ数）	ターゲットがファイルに出力される場合に表示されます。Postによって実行されたファイルスイッチの総数を示します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー（3）
発行対象:	ターゲットシステム
関連コマンド:	**show capture**、**show read**、**show export**、**show import**

構文

基本コマンド	コマンドオプション	リモートオプション
show post	[**detail**]	[**on** *host* |
	[**queue** *queuename*]	**on** *host:portnumber* |
	[**for** *datasource-datadest*]	**on** *login/password@host* |
	[**sessions**]	**on** *login/password@host:portnumber*]
	[**session=***n*]	

構文の説明

コンポーネント	説明
show post	Postプロセスの状態と処理された操作の概要を表示します。
detail	このオプションはPostプロセスの詳細な統計情報を表示します。 **例**:

コンポーネント	説明
	`sp_ctrl(sysB)` > **show post detail**

queue *queuename*	このオプションは**show post**の表示を特定のpostキューに対してフィルタリングします。
	• **queue**は構文の必須部分です。
	• *queuename*はPost統計情報を表示したいpostキューです。有効な値:
	  • デフォルトキューを使用する場合のソースシステムの名前。
	  • 名前付きキューを使用している場合は、ユーザ定義のキュー名。
	キュー名が不明な場合は、**qstatus**コマンドを実行します。キュー名はすべてのプラットフォームで大文字と小文字が区別されます。
	このオプションは、他のオプションと共に任意の順序で表示できます。
	例:
	`sp_ctrl(sysB)` > **show post queue sysA**

for *datasource-datadest*	このオプションはshow postの表示を特定のデータストリームに対してフィルタリングします。 • **for**は構文の必須部分です。 • *datasource*は**o.***SID*と表します。ここで、*SID*はORACLE_SIDです。 • *datadest*は、ターゲットによって以下のいずれかで表されます。 **o.***ORACLE_SID* **r.***database_name* **x.kafka** **x.jms** **x.file** このオプションは、他のオプションと共に任意の順序で表示できます。 **例**: `sp_ctrl(sysB)` > **show post for o.oraA-r.ssB**
sessions	Oracleターゲットの場合、このオプションはPostプロセスによって生成されたすべてのスレッドの統計情報を表示します。 シングルスレッドであるOpenターゲットの場合、このオプションを使用すると、そのスレッドの詳細を表示できます。 このオプションは、他のオプションと共に任意の順序で表示できます。 **例**: `sp_ctrl(sysB)` > **show post sessions queue *queuename***

コンポーネント	説明
session=*n*	（Oracleのみ）このオプションは、**show post sessions**出力に表示される統計情報に加え、スレッドの状態、スレッドで処理待ちのメッセージ数、およびトランザクション全体の状態に関する追加の詳細を表示します。 •**session=**は構文の必須部分です。 •*n*は、**show post sessions**コマンドで表示されるセッション番号です。コンポーネント間にスペースを空けないでください。 このオプションは、他の任意のオプションと共に任意の順序で表示できます。 **例**: `sp_ctrl(sysB)>` **show post session=1234 queue *queuename***

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show read

Readプロセスに関する統計情報を表示するには、**show read**コマンドを使用します。

基本コマンド

基本的な**show read**コマンドは、データソース、プロセスが実行中か停止中、その他の基本情報など、プロセスの概要を示します。

詳細な統計情報

Readプロセスの詳細な統計情報を表示するには、**show read**コマンドに**[detail]**オプションを指定します。このオプションは、プロセスのパフォーマンスを評価し、チューニングパラメーターを調整する必要があるかどうかを決定し、問題やボトルネックを検出するのに役立つ詳細な統計情報を表示します。

Oracle Read用の詳細な統計情報

統計情報	説明
Host（ホスト）	ローカルマシンの名前（ソースシステム）
Source（ソース）	ソースOracleインスタンスの名前。
Status（ステータス）	Readプロセスのステータス（実行中または停止中）。
Operations forwarded（転送された操作数）	ReadがExportキューに送信した操作の数。
Since（開始時刻）	Readが開始された時刻。
Total（合計）	captureキュー内のまだ読み取り-リリースされていないメッセージの数。この数字は、**qstatus**を実行して返された「メッセージ数」に相当します。
Backlog（バックログ）	Readによって処理されるのをcaptureキュー内で待機しているメッセージの数。
最後に転送された操作	Readがexportキューに送信した直近の操作、またはReadが非アクティブの場合は最後に送信した操作に関する情報を示します。操作は、DMLまたはDDL操作、あるいは内部SharePlex操作に関連するデータです。
Read state（Readの状態）	プロセスが実行するレプリケーション作業に関連したプロセスの状態：以下のいずれかになります。 • **IDLE（アイドル）**：Readは、captureキューから次のレコードバッチを収集して処理するのを待機しています。Readのアイドル時間が長すぎる場合は、Captureの実行が遅れているか、captureキューに新しいデータがないためにアイドル状態になっている可能性があります。そうではなく、Readプロセスのレイテンシを短縮したい場合は、SP_ORD_DELAY_RECORDSパラメーターを調整することで、Readがより速くレコードを収集して処理できるようになります。

統計情報	説明
	• **PASS1（パス1）**: Readはデータを処理し、ルーティングのためにパッケージ化する最初のフェーズにあります。 • **PASS2（パス2）**: Readは処理の2番目のフェーズにあります。 **Read State（Readの状態）**フィールドは、Readが操作の処理に時間がかかりすぎているように見える場合に役立ちます。
Activation ID（アクティベーションID）	設定のアクティベーションの内部識別番号です。関連するプロセスとキューを識別します。
Operations forwarded（転送された操作数）	Readが開始後にexportキューに送信したトランザクション操作と内部SharePlex操作の数。
Transactions forwarded（転送されたトランザクション数）	Readが開始後にexportキューに送信したコミット済みトランザクションの数。この値と、*転送された操作数*の値をcompareすることで、トランザクションが小さい傾向にあるのか、大きい傾向にあるのかがわかります。その情報を使って、なぜReadが遅れているように見えるのか、なぜキューが空にならないのか（COMMITのない大きなトランザクション）を判断することができます。
Full rollbacks（フルロールバック数）	Readで処理されたロールバックトランザクションの数を示します。
Full rollback operations skipped（スキップされたフルロールバック操作数）	トランザクションがロールバックされたため、Readがexportキューに転送しない操作の数。
Cursor cache hit count（カーソル・キャッシュ・ヒット数）	Readがキャッシュされたカーソルを使用した回数。
Cursor cache miss count（カーソル・キャッシュ・ミス数）	Readがキャッシュされたカーソルを使用できなかった回数。
Number of open cursors（オープンカーソルの数）	必要に応じてOracleにアクセスするためにReadが予約しているオープンカーソルの数。
Number of active batches（アクティブバッチの数）	バッチトランザクションとして処理されている、現在アクティブなトランザクションの数。有効にすると、Readはバッチ操作を結合し、より迅速にポストできるようになります。
Batch message total（バッチメッセージ合計）	Readプロセスがバッチ操作に結合する操作の数。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**show capture**、**show export**、**show import**、**show post**

構文

基本コマンド	コマンドオプション	リモートオプション
show read	[**detail**] [**for** *datasource*]	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
show read	Readプロセスの状態と処理された操作の概要を表示します。
detail	このオプションは、Readプロセスの詳細な統計情報を表示します。 **例**: `sp_ctrl(sysA)` > **show read detail**
for *datasource*	このオプションは、特定のデータソースのRead統計情報のみを表示します。 *datasource*は**o.***SID*と表します。ここで、*SID*はORACLE_SIDです。 **例**: `sp_ctrl(sysA)` > **show read for o.oraA**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show SQL

show SQLコマンドを使用して、Postプロセスがターゲットデータベースにデータをポストするために書き込んでいるSQLステートメントを表示します。postキューが空か、Postがステートメントを処理していない場合、**show SQL**コマンドは、処理された最新のSQLステートメントを表示します。

このコマンドは、レプリケーションに時間がかかりすぎていると思われる場合やPostがエラーで停止した場合に使用します。どのSQLステートメントに問題があるのかを知ることで、何が間違っているか、そしてさらなる対処が必要かどうかを判断することができます。

以下にshow sqlの表示例を示します。

```
sp_ctrl (tustin:8852)> show sql

Last SQL statement of queue tustin and instance o.ora920-0.ora920 on tustin

insert into "KWONG"."KCWTAB2" ("C1","C2") values (:V001,:V002)
```

ターゲットシステムに対してこのコマンドを実行します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ターゲットシステム
関連コマンド:	なし

構文

基本コマンド	コマンドオプション	リモートオプション
show SQL	[**queue** *queuename*]	[**on** *host* |
	[**for** *datasource-datadest*]	**on** *host:portnumber* |
	[**thread=***n*]	**on** *login/password@host* |
	[**session=***n*]	**on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
show SQL	オプションを指定しない場合は、すべてのSQLステートメントが表示されます。
queue *queuename*	特定の名前付きキューのSQLステートメントを表示するには、このオプションを使用します。 • **queue**は構文の必須部分です。

コンポーネント	説明
	• *queuename*はキューの名前です。キュー名が不明な場合は、qstatusコマンドを実行します。 キュー名はすべてのプラットフォームで大文字と小文字が区別されます。 **例**: `sp_ctrl(sysA)` > **show sql queue q1**
for *datasource-datadest*	複数のOracleインスタンスにレプリケートする場合、または複数のOracleインスタンスからレプリケートする場合に特定のレプリケーションストリームのSQLを表示するには、このオプションを使用します。 • **for**は構文の必須部分です。 • *datasource*は**o**.*SID*で表わされます。ここで*SID*はソースOracleインスタンスのORACLE_SIDです。 • *datadest*は**o**.*SID*で表わされます。ここで、*SID*はターゲットOracleインスタンスのORACLE_SIDです。 **例**: `sp_ctrl(sysA)` > **show sql for o.oraA-o.oraB**
thread=*n*	ポスト処理スレッドのSQLを表示するには、このオプションを使用します。 • **thread=**は構文の必須部分です。 • *n*はshow post threadsコマンドで表示されるスレッド番号です。 コンポーネント間にスペースを空けないでください。このオプションは、他のオプションと共に任意の順序で表示できます。 **例**: `sp_ctrl(sysB)` > **show sql thread=1234**
session=*n*	ユーザセッションのSQLを表示するには、このオプションを使用します。 •**session=**は構文の必須部分です。 •*n*は、**show post threads**コマンドで表示されるセッション番号です。コンポーネント間にスペースを空けないでください。このオプションは、他のオプションと共に任意の順序で表示できます。 **例**: `sp_ctrl(sysB)` > **show sql session=1234**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show statusdb

ステータスデータベースを表示するには、**show statusdb**コマンドを使用します。レプリケーションに関与する各マシンには、ユーザインターフェイスでエラーメッセージや警告を生成しなかったものも含め、主要なレプリケーションイベントのレコードを含む独自のステータスデータベースがあります。この情報は潜在的な問題を警告し、既存の問題を解決するのに役立つ場合があります。

show statusdbの表示には以下の情報が含まれます。

- **Level(レベル)**: そのエントリが情報目的のみで存在しているのか、またはエラーや警告状態の結果を示しているのかどうか。
- **Details(詳細)**: イベントの理由。

必要に応じて、ステータスデータベースは、エントリに関する詳細情報がある場合、イベントログを参照します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**status**、**lstatus**

構文

基本コマンド	コマンドオプション	リモートオプション
show statusdb	**detail**	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
detail	このオプションは、ステータスデータベースの高度なレベルの詳細を表示します。 **例**: `sp_ctrl(sysB)` > **show statusdb detail**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show scn

Postプロセスがターゲットに適用した最後のトランザクションのOracle SCNを表示するには、**show scn**コマンドを使用します。このコマンドは、複数のpostキューを使用している場合、最後のSCNのすべてを表示します。

注意: このコマンドは、SCN情報を取得するためにすべてのPostプロセスを停止します。

```
sp_ctrl (sysB)> show scn

For resume replication from ora112

On source activate to scn=4550108289

reconcile queue spdl13 for o.ora112-o.ora112 scn 4550108290

reconcile queue spdl14 for o.ora112-o.ora112 scn 4574108174

reconcile queue spdl15 for o.ora112-o.ora112 scn 5374667318
```

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	Oracle
認証レベル:	ビューアー(3)
発行対象:	ターゲットシステム
関連コマンド:	**activate config**

構文

基本コマンド	リモートオプション
show scn	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

show sync

show syncコマンドは、Postで生じた可能性がある非同期の問題を表示します。情報はステータスデータベースから抽出されます。

同期していない各オブジェクトについて、コマンドは以下のように表示します。

- **Count(数)**: 失敗したステートメントの数
- **Detail(詳細)**: テーブル名、キュー名、日時が含まれます。

どのSQLステートメントでエラーが発生したかを確認するには、**SID_errlog.sql**ログファイルを表示します。イベントログにも問題のレコードが含まれます。

show syncコマンドを実行したときに同期がとれていないものがなければ、**sp_crtl**プロンプトに戻ります。

ターゲットシステムに対してこのコマンドを実行します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ターゲットシステム
関連コマンド:	**show statusdb**

構文

基本コマンド	リモートオプション
show sync	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。

オプション	説明
	例: `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

シャットダウン

レプリケーションをシャットダウンするには、**shutdown**コマンドを発行します。shutdownコマンドが完了すると、SharePlexは**sp_ctrl**を自動的にシャットダウンします。

グレースフルシャットダウン

基本的な**shutdown**コマンドは、**sp_cop**とその他すべてのSharePlexプロセスを正常にシャットダウンすることで各プロセスの状態を保存し、ディスクへのチェックポイントを実行し、バッファされたデータを読み取り/リリースし、子プロセスをクリーンアップします。キュー内のデータは、権限を持つユーザが**sp_cop**を再び開始したときに処理できるように、安全な場所に残されています。

強制シャットダウン

shutdownコマンドに**[force]**オプションを指定すると、通常のシャットダウン手順が完了したか否かにかかわらず、SharePlexプロセスが直ちに終了します。**sp_cop**を起動するとSharePlexは強制シャットダウンから回復しますが、レプリケーション処理は以前に保存したポイントから再開されるため、起動に若干時間がかかることがあります。

[force]オプションは、SharePlexを正常にシャットダウンしようとしたときに子プロセスが実行中だった場合にのみ使用します。**ps -ef | grep sp_**コマンドをコマンドシェルで実行すると、SharePlexプロセスがまだ実行中かどうかを検証できます。

重要: レプリケーションをシャットダウンしたときに、ユーザがアクティブな設定のオブジェクトの変更を続けた場合、SharePlexを再起動する前にOracleのREDOログがラップされる可能性があります。遅延が長く、アーカイブログを利用できなくなった場合は、データを再同期する必要があります。REDOログは、SharePlexがシャットダウンされる時間に合わせて十分な大きさと数にする必要があります。レプリケーション用のREDOログの適切なサイズ設定の詳細については、『SharePlexインストールおよびセットアップガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者(1)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**startup**

構文

基本コマンド	コマンドオプション	リモートオプション
shutdown	[**force**]	[**on** *host* |
		on *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
force	このオプションは、**sp_cop**を含むすべてのレプリケーションプロセスを、キューからの読み取りやキューへの書き込みの終了の有無にかかわらず直ちにシャットダウンします。このオプションは、デフォルトの**shutdown**コマンドが失敗するか、グレースフルシャットダウンを待つことができない場合に使用します。 **例**: `sp_ctrl(sysA)` **> shutdown force**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

start

stopまたは**abort** [*service*]コマンドを使用して停止した後、または**flush**コマンドを使用してPostを停止した後にレプリケーションプロセスを開始するには、**start**コマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)、Captureに必要な管理者(1)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**stop**

構文

基本コマンド	コマンドオプション	リモートオプション
start *service*	[**to** *host*]	[**on** *host* |
	[**from** *host*]	**on** *host:portnumber* |
	[**for** *datasource*]	**on** *login/password@host* |
	[**for** *datasource-datadest*]	**on** *login/password@host:portnumber*]
	[**queue** *queuename*]	

構文の説明

コンポーネント	説明
service	開始するSharePlexプロセス。有効な値: • Capture • Read • Export • Import • Post **例**: `sp_ctrl(sysA)` > **start export**
to *host*	このオプションは、指定されたターゲットシステムのみに対してExportを開始します。 •**to**は構文の必須部分です。

コンポーネント	説明
	•*host*はターゲットシステムの名前です。 **例**: `sp_ctrl(sysA)`**> start export to sysB** この例では、sysAからsysBに対してExportが開始されます。
from *host*	このオプションは、指定されたソースシステムからのみImportを開始します。 • **from**は構文の必須部分です。 • *host*はソースシステムの名前です。 **例**: `sp_ctrl(sysD)`**> start import from sysA** この例では、sysAからsysDに対してImportが開始されます。
for *datasource*	このオプションは、指定されたデータソースのCaptureまたはReadを開始します。 • **for**は構文の必須部分です。 • *datasource*は**o.***SID*と表します。ここで、*SID*はORACLE_SIDです。 **例**: `sp_ctrl(sysA)`**> start read for o.oraA** この例では、sysA上でインスタンスoraAに対してReadが開始されます。
for *datasource-datadest*	このオプションは、指定されたソースSID-ターゲットSIDデータストリームに対してのみPostを開始します。 • **for**は構文の必須部分です。 • *datasource*は**o.***SID*と表します。ここで、*SID*はORACLE_SIDです。 • *datadest*は**o***SID*または**r.***database*と表します。ここで、*SID*はORACLE_SID、*database*はOpen Targetデータベースの名前です。 **例**: `sp_ctrl(sysC)`**> start post for o.oraA-r.mssA** この例では、sysC上でインスタンスoraAに対してPostが開始され、SQL Serverデータベース mssAにレプリケートされます。
queue *queuename*	このオプションは、指定された名前付きキューに関連付けられたExport、ImportまたはPostプロセスを開始します。 •**queue**は構文の必須部分です。 •*queuename*は、設定ファイルに記載されているキューの名前です。

コンポーネント	説明
	例: `sp_ctrl(sysA)` **> start export queue QA** この例では、名前付きexportキューQAを通じてsysA上でデータのExportが開始されます。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

startup

startupコマンドを使用して、すべてのプロセスを一度に開始します。**sp_cop**をシャットダウンする前に**stop**コマンドで停止していなければ、すべてのプロセスが開始されます。

アクティブな設定がある場合、コマンドラインから**sp_cop**を起動すると、通常はレプリケーションが自動的に開始されます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者(1)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**shutdown**

構文

基本コマンド	リモートオプション
startup	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の

オプション	説明
	構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Status

statusコマンドは、レプリケーションのステータスの概要を表示する、プロセスが実行されていることを確認する、またエラー、警告、通知をチェックするために使用します。詳細なステータスレポートが必要な場合は、**lstatus**コマンドを使用します。

ステータスディスプレイには以下が表示されます。

- **Process（プロセス）**:プロセスの名前
- **State（状態）**: 各プロセスのステータス（実行中、アイドル、エラーによる停止、ユーザによる停止のいずれか）
- **PID**: オペレーティングシステムのプロセスID番号
- **Running Since（実行開始日）**:プロセスが開始された日時
- **その他の情報**: システムの使用状況、システム上にアクティブな設定があるかどうか、レプリケーションエラーが発生したかどうかなど

SharePlexは、SP_COP_IDLETIMEパラメーターで指定された間隔でステータス表示を更新します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー（3）
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**lstatus**、**qstatus**、**show**、**show statusdb**

構文

基本コマンド	リモートオプション
status	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求める

オプション	説明
	プロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Stop

stopコマンドは、SharePlexレプリケーションプロセスを正常に停止し、関連するキューからの読み取りまたはキューへの書き込みを終了させるために使用します。レプリケーションプロセスが停止すると、データがキューに安全に蓄積されます。

stopコマンドで中断されたプロセスは、SharePlexをシャットダウンして再起動しても、停止したままになります。このプロセスは、**start**コマンドでのみ開始できます。プロセスが再び開始されると、停止した時点から処理が再開されるため、データは同期された状態を維持します。

注意: **stop**コマンドは**abort service**コマンドとは異なり、プロセスが関連するキューからの読み取りまたは書き込みを終了した後にレプリケーションプロセスを停止します。**abort service**コマンドは、プロセスがデータ処理中であるか否かにかかわらず、即座にプロセスを終了します。

Captureプロセスの停止

Captureプロセスを停止するときは注意してください。

以下のような場合に、ソースとターゲットのデータが同期しなくなる可能性があります。

- Captureが停止している間にユーザがソースデータの変更を続けた場合

  および…

- OracleのREDOログがその間にラップされた場合

  および…

- アーカイブログが利用できなくなった場合

特定の時点でのPostの停止

Postプロセスの停止のタイミングを制御するには、いずれかの[**at sourcetime**]オプションを使用します。これらのいずれかのオプションは、指定された時刻または日時のスタンプが付いた最初のメッセージを受信すると、自動的にPostを停止します。それ以降のメッセージは、Postが再開されるまでpostキューに蓄積されます。

[**at sourcetime**]オプションは、ジョブがソースシステムでスケジューリングされているときに、その操作が、ターゲットデータベースへの適用前にソースシステムで成功したことを確認する場合に使用できます。Postを停止するオプションは、ジョブの実行直前に設定します。

このオプションは、ターゲットシステム上でレポートを実行しているときに、Post操作の競合によりオーバーヘッドが発生しないようにする場合にも使用できます。

SharePlexは同期していないため、ターゲットシステム上でPostが実際に停止する時刻は、おそらく操作のタイムスタンプよりも後になります。このタイミングは、先行するメッセージをキューにポストするために必要な時間に左右されます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)、Captureの停止に必要な管理者(1)

発行対象:

- Capture、Read、およびExportの**stop**は、ソースシステムに対して発行されます。
- ImportおよびPostの**stop**は、ターゲットシステムに対して発行されます。

関連コマンド: **abort service**、**start**

構文

基本コマンド	コマンドオプション	リモートオプション
stop *service*	[**to** *host*]	[**on** *host* |
	[**from** *host*]	**on** *host:portnumber* |
	[**for** *datasource*]	**on** *login/password@host* |
	[**for** *datasource-datadest*]	**on** *login/password@host:portnumber*]
	[**queue** *queuename*]	
	[**at sourcetime** *hh:mm:ss*]	
	[**at sourcetime** *mm/dd/yyyy hh:mm:ss*]	

構文の説明

コンポーネント	説明
service	停止するSharePlexプロセス。有効な値: • Capture • Read • Export • Import • Post **例**: `sp_ctrl(sysA)` > **stop export**
to *host*	このオプションは、指定したターゲットシステムへのExportを停止しますが、他のシステムへのExportは続行されます。 • **to**は構文の必須部分です。 • *host*はターゲットシステムの名前です。 **例**: `sp_ctrl(sysA)` > **stop export to sysB**

コンポーネント	説明
	この例では、sysAからsysBへのExportは停止されますが、sysAから他のターゲットシステムへのExportは続行されます。
from *host*	このオプションは、指定したシステムからのImportを停止しますが、他のシステムからのImportは続行されます。 • **from**は構文の必須部分です。 • *host*はソースシステムの名前です。 **例**: `sp_ctrl(sysD)` > **stop import from sysC** この例では、sysDのsys CからのImportは停止されますが、sysDの他のシステムからのImportは続行されます。
for *datasource*	このオプションは、指定されたデータソースのCaptureまたはReadを停止します。 • **for**は構文の必須部分です。 • *datasource*は**o**.*SID*と表します。ここで、*SID*はORACLE_SIDです。 **例**: `sp_ctrl(sysA)` > **stop read for o.oraA** この例では、sysAのインスタンスoraAに対するReadは停止されますが、sysAの他のインスタンスに対する他のReadプロセスは続行されます。
for *datasource-datadest*	このオプションは、指定されたソース-ターゲット・データ・ストリームのPostを停止します。 • **for**は構文の必須部分です。 • *datasource*は**o**.*SID*と表します。ここで、*SID*はORACLE_SIDです。 • *datadest*は**o***SID*または**r**.*database*と表します。ここで、*SID*はORACLE_SID、*database*はOpen Targetデータベースの名前です。 **例**: `sp_ctrl(sysC)` > **stop post for o.oraA-o.oraC** この例では、インスタンスoraAからoraCへのレプリケートに対するsysCのPostが停止されますが、sysC上の他のPostプロセスは続行されます。
queue *queuename*	このオプションは、名前付きキューに関連付けられたExport、ImportまたはPostプロセスを停止します。他の名前付きキューによるレプリケーションは、影響を受けずに続行されます。 • **queue**は構文の必須部分です。

コンポーネント	説明
	• *queuename*は、設定ファイルに記載されているキューのユーザ定義名です。 **例**: `sp_ctrl(sysA)` > **stop export queue QA** この例では、sysAでの名前付きexportキューQAを介したデータのExportは停止されますが、それ以外のすべての名前付きキューに割り当てられたデータのExportは続行されます。
at sourcetime *hh:mm:ss*	このオプションは、指定された時刻のスタンプが付いた最初のメッセージを受信するとPost処理を停止します。それ以降のメッセージは、Postが再開されるまでpostキューに蓄積されます。 • **at sourcetime**は構文の必須部分です。 • *hh:mm:ss*はPostが停止するタイムスタンプで、24時間制で指定する必要があります。1桁のコンポーネントはゼロ(0)で埋めます。空白は許可されません。 **例**: `sp_ctrl(sysC)` > **stop post at sourcetime 24:00:00** この例では、午前0時のタイムスタンプが付いた最初のメッセージを受信したときに、PostがsysC上で停止します。
at sourcetime *mm/dd/yyyy hh:mm:ss*	このオプションは、指定された日時のスタンプが付いた最初のメッセージを受信するとPost処理を停止します。このオプションの目的と機能は**stop post at sourcetime** *hh:mm:ss*と同じですが、日付オプションが追加されています。 • **at sourcetime**は構文の必須部分です。 • *mm/dd/yyyy*は、Postを停止させたい指定の日付です。構文上、日付の要素は時刻の要素より前になければなりません。年号には4桁すべての数字を含めなければなりません。1桁のコンポーネントはゼロ(0)で埋めます。空白は許可されません。このコンポーネントと時刻のコンポーネントはスペースで区切ります。 • *hh:mm:ss*はPostが停止するタイムスタンプで、24時間制で指定する必要があります。1桁のコンポーネントはゼロ(0)で埋めます。 **例**: `sp_ctrl(sysC)` > **stop post at sourcetime 01/31/2002 24:00:00** この例では、2002年1月31日午前0時のスタンプが付いた最初のメッセージを受信したときに、PostがsysC上で停止します。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Target（ターゲット）

targetコマンドは、Postが特定のタイプのターゲットをサポートするよう設定するために使用します。これはターゲットの設定と呼ばれます。ターゲット設定に保存されているプロパティは、Postがレプリケートされたデータをどのように処理して、フォーマットし、出力するかを制御します。

targetコマンドは、ターゲットの設定をリセットしたり表示したりするオプションも提供します。

targetコマンドは、グローバルレベルでも、また**queue**オプションが使用されている場合にはプロセスごとのレベルでも、ターゲット設定を制御するために使用することができます。

重要: このコマンドを使用した後は、必ずPostプロセスを停止してから再度開始してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター（2）
発行場所:	ターゲットシステム
関連コマンド:	**connection**

構文

基本コマンド	コマンドオプション	リモートオプション
target {**o.** | **r.** | **x.** | **c.**}*target*	{	利用不可
[**queue** *queuename*]	{ **set** *category property* |	
{*command_option*}	**reset** [*category* [*property*]] |	
	show [*category*] } |	
	switch	
	}	

構文の説明

コンポーネント	説明
{**o.** | **r.** | **x.** | **c.**}*target*	Postを設定するターゲット。ターゲット名の前の文字は、Oracleデータベース、Open Targetデータベース、JMS、Kafkaブローカー、ファイル、変更履歴ターゲットなど、ターゲットの種類を示します。 **考えられるターゲットは以下の通りです。** **o.***SID* ここで、*SID*はOracleデータベースのORACLE_SIDです。

コンポーネント	説明
	r.*database* ここで、*database*はOpen Targetデータベースの名前です（DSNではない）。* **x.jms** ここで、**jms**はJava Message Service（JMS）のキューまたはトピックです。 **x.kafka** ここで、**kafka**はApache Kafkaブローカーです。 **x.file** ここで、**file**は、サポートされている形式のいずれかのデータを含む構造化ファイルです（「formatカテゴリ」を参照）。 **c.***SID* ここで、*SID*は、変更履歴ターゲットとして構成されているターゲットOracleデータベースのORACLE_SIDです。 * **注意:** SharePlexがサポートするデータストアのタイプ、データ型、および操作は『SharePlexリリースノート』に記載されています。
queue *queuename*	コマンドの動作を、指定されたキューに関連付けられているSharePlex Postプロセスに制約します。この動作は、設定しているターゲットプロパティのグローバル設定をオーバーライドします。**queue**オプションがない場合、targetコマンドは、ローカルのSharePlexインスタンス内のすべてのPostプロセスに影響を与えます。 例: 以下は、**myjms1**キューに関連付けられているPostプロセスのメタデータプロパティを設定します。 **target x.jms queue myjms1 set metadata time, userid, trans, size**

{
 	
set *category property* |
 	
reset [*category* [*property*]] |
 	
show [*category*] |
 	
switch
 	
}
	• **set**は、設定プロパティをターゲット設定に適用します。
	• **reset**は、設定をデフォルトに戻します（存在する場合）。特定のカテゴリと、（該当する場合は）特定のプロパティに限定することができます。
	
	注意: resetは、あるプロパティ値を別のプロパティ値にリセットすることはありません。その場合は、別の**set**コマンドを使用します。
	
	• **show**は設定を表示します。
	• *category*は、設定プロパティの任意のカテゴリです。プロパティは、コマンドによる効率的な管理のためにカテゴリごとにグループ化されています。「カテゴリの説明」を参照してください。
	• *property*は、設定、リセット、または表示する設定プロパティです。カテゴリごとに異なる特性があります。

コンポーネント	説明
	• **switch**は、ファイルターゲット（**x.file** ）にのみ適用されます。このオプションは、新しいアクティブな出力ファイルを作成し、古い順にシリアル番号を付けたファイルとして現在のファイルを保存するために使用します。このコマンドの発行後、Postが新しいレコードを処理した後に切り替わります。このオプションは入力を必要とせず、いかなる設定カテゴリとも関連はありません。

カテゴリの説明

カテゴリ	説明
cdc	特定のテーブルのSP_OPO_TRACK_PREIMAGEパラメーターで設定されている、Oracle変更履歴ターゲットのグローバル設定をオーバーライドします。「CDCカテゴリ」を参照してください。
characterset	Open Targetターゲットデータベースにポストする前に、レプリケートされたデータを指定された文字セットに変換するようPostプロセスに指示します。「charactersetカテゴリ」を参照してください。
file	ファイルターゲットのプロパティを設定します。詳細については、「fileカテゴリ」を参照してください。
format	ターゲットに書き込むデータの形式を設定します。詳細については、「formatカテゴリ」を参照してください。
json	**format record=json**のときのJSON出力のプロパティを設定します。詳細については、「JSONカテゴリ」を参照してください。
jms	JMSターゲットのプロパティを設定します。詳細については、「JMSカテゴリ」を参照してください。
kafka	Kafkaターゲットのプロパティを設定します。詳細については、「kafkaカテゴリ」を参照してください。
metadata	SharePlexによってレプリケートされるデータにメタデータプロパティを追加します。詳細については、「metadataカテゴリ」を参照してください。
resources	ターゲットシステムのリソースに影響を与えるパラメーターを設定します。「resourceカテゴリ」を参照してください。
rule | filter	Oracleの変更履歴ターゲットの特定のテーブルに追跡ルールまたはフィルターを適用する条件文を指定します。「rule | filterカテゴリ」を参照してください。
source	Oracle変更履歴ターゲットのソース識別子プロパティを設定します。詳細については、「sourceカテゴリ」を参照してください。
SQL	**format record=sql**のときのSQL出力のプロパティを設定します。詳細については、「SQLカテゴリ」を参照してください。

CDCカテゴリ

サポート対象のターゲット

Oracle変更履歴ターゲット

コマンドオプション

target c.*SID* [**queue** *queuename*]

table *tablename*

{

set cdc preimage={**yes** | **no**}

reset cdc

show cdc

}

用途

特定のテーブルのSP_OPO_TRACK_PREIMAGEパラメーターで設定されている更新のグローバル設定をオーバーライドします。これは、更新操作の更新前イメージを履歴テーブルで追跡するかどうかを制御します。有効な値は、更新前イメージを追跡する場合は**yes**、更新前イメージを除外する場合は**no**です。

charactersetカテゴリ

サポート対象のターゲット

Open Targetデータベース

コマンドオプション

target r.*database* [**queue** *queuename*]

{

set characterset *character_set* |

reset characterset *character_set* |

show characterset *character_set*

}

用途

ターゲットデータベースにポストする前に、レプリケートされたデータをローカルのOracleクライアントに送信し、指定された文字セットに変換するようPostプロセスに指示します。指定された文字セットは、Open Targetターゲットに文字データを投稿するときにPostで使用されるUnicodeのデフォルトの文字セットをオーバーライドします。

UTF8からUTF8の絵文字のレプリケーションは、OracleからOpen Posterでは問題なく機能します。ただし、UTF8形式ではない（CESU8など）Oracleソースデータベースに絵文字を挿入する場合、このようなシナリオでは、非Oracleターゲットに絵文字をレプリケートするためにターゲットSharePlexインスタンスに以下の設定を適用する必要があります。

1. データ変換のターゲットにOracleクライアントをインストールします。
2. **SP_OPX_NLS_CONVERSION**パラメーターを1（デフォルト）に設定します。
3. ターゲットDBに必要な有効なUTF8バイトシーケンスにデータを変換するために、targetコマンドを使用してターゲット文字セットをAL32UTF8に設定します。

注意: このオプションを使用するには、ターゲットシステム上にOracleクライアントが存在しなければなりません。詳細については、SharePlexインストールおよびセットアップガイドの「Open Targetのチェックリスト」のセクションを参照してください。

プロパティ	入力値	デフォルト
character_set	ターゲットの文字セットに相当するOracleの文字セット（そのOracle名で指定）	Unicode

例

GB2312文字セットのデータをターゲットにポストするには、OracleのZHS16GBK文字セットを指定します。

target r.mymss set characterset ZHS16GBK

fileカテゴリ

サポート対象のターゲット

ファイル

コマンドオプション

target x.file [**queue** *queuename*]

{

set file *property=value* |

reset file [*property*] |

show file

}

用途

ファイルターゲットのプロパティを設定します。

プロパティ	入力値	デフォルト
location=*pathname*	ファイルを作成するSharePlex変数データディレクトリのパス名	opx
max_records=*number*	ファイルを切り替える前の、レコード数で測定されるアクティブファイルの最大サイズ	50,000

プロパティ	入力値	デフォルト
max_size=*megabytes*	ファイルを切り替える前のファイルの最大サイズ（メガバイト単位）	50
max_time=*seconds*	ファイルを切り替えるまでの最大待機時間（秒）	300
record_length=*number*	レコードの最大サイズ（文字数）	132

例

target x.file set file max_size=320

formatカテゴリ

サポート対象のターゲット

ファイル

kafka

コマンドオプション

target x.{**file** | **kafka**} [**queue** *queuename*]

{

set format *property=value* |

reset format [*property*] |

show format

}

用途

ターゲットに書き込むデータの形式を設定します。

プロパティ	入力値	デフォルト
date=*format*	日付形式	yyyy-MM-dd HH:mm:ss
decimal=*character*	10進数文字	.（ピリオド）
enotation=*notation*	指数表記	14
record=*record_format*	出力レコードの形式。有効な値は、**sql**、**xml**、または**json**です。	**xml**
timestamp=*format*	タイムスタンプの形式	yyyy-MM-ddTHH:mm:ss.fffffffff

例

target x.file set format record=sql

target x.kafka set format record=json

出力形式の説明

以下は、**format record**=*record_format*が提供する出力形式の説明です。

SQLレコードの形式

SQL形式のファイルの各トランザクションは、SQLファイル内のトランザクションシーケンスと一意のトランザクションIDを含むコメントで始まります。SQLファイルの末尾にあるコメント行にはファイル内の行数が表示されます。例えば、以下は1つのトランザクションを持つSQLファイルです。この例では、トランザクションIDは2-113319です。ファイルには9つの行があります。

```
/installed/vardir> cat opx/0000000010_20140305140820_legacy.sql

-- 0000000001 2-113319 03052014140813 03052014140813

DELETE FROM "ROBIN"."TEST_TYPES" WHERE ORA_NUMBER = '22345' AND ROWNUM = 1;

INSERT INTO "ROBIN"."TEST_TYPES" (ORA_NUMBER, ORA_DATE, ORA_RAW, ORA_ROWID,

ORA_FLOAT, ORA_CHAR, ORA_VARCHAR2, ORA_TIMESTAMP, ORA_TIMESTAMP_TZ,

ORA_TIMESTAMP_LTZ) VALUES('22345', '08132066000000', '0123456789ABCDEF'

, 'AAAAAAAAAAAAAAAAAA', '12350', 'Character ', 'Variable data'

, '10201998021300.22000', '06172002080000.00000', '06172002160000.00000');

COMMIT;

-- EOF 0000000009
```

XMLレコードの形式

XMLフォーマットは操作とスキーマの「型」に分けられ、より簡単に利用できるようになっています。XSDの観点から見ると、これらは実際には同じであり、別個の型ではありません。テンプレートXMLは、すべての可能な属性と要素を表します。個々のXMLは、サポートされている各操作の最小限の出力を表します。

起動後、Postが初めて任意のテーブルの変更レコードを書き込む場合は、まずそのテーブルのスキーマレコードを書き込みます。各スキーマレコードには、テーブル名と各列の詳細が含まれています。スキーマレコードは、スキーマに変更がない限り、Postの実行中に各テーブルに対して1回だけ書き込まれ、その後は新しいスキーマレコードが書き込まれます。Postが停止し、開始すると、Postが各テーブルの変更レコードを受信するたびに、スキーマレコードが再度書き込まれます。

スキーマレコードのテンプレート

```
<?xml version="1.0" encoding="UTF-8" ?>
<?opentarget version="1.0" ?>
<opentarget>
    <txn
        id="xs:integer"
        oracleTxnId="xs:string"
        commitTime="xs:dateTimeStamp" />
    <tbl
        name="xs:string"
        utcOffset="xs:integer"
```

```
        <cmd ops="schema">
            <schema>
                <col
                    name="xs:string"
                    xmlType="xs:string"
                    key="xs:boolean"
                    nullable="xs:boolean"
                    length="xs:integer"
                  />
            </schema>
        </cmd>
    </tbl>
</opentarget>
```

表1: スキーマテンプレートの説明 (* =オプション)

要素	属性	説明
txn		トランザクションメタデータ
	id	現在のトランザクションのID
	oracleTxnId *	OracleのトランザクションID
	commitTime*	トランザクションのコミットタイムスタンプ
tbl		テーブルメタデータ
	name	テーブルの完全修飾名
	utcOffset	ログのUTCオフセット
cmd		操作のメタデータ(スキーマの場合は操作なし)
	ops	このテーブルについて生成されるレコードのタイプ。スキーマの場合、値は**schema**です。
schema		列のメタデータ
col		列のメタデータ(テーブルのレコードごとにこれらの要素の1つが表示される)
	name	列名
	xmlType	XMLデータ型
	key	キーフラグ(真、偽)
	nullable	NULL許容フラグ
	length	列の長さ

操作レコードテンプレート

```
<?xml version="1.0" encoding="UTF-8" ?>
<?opentarget version="1.1" ?>
```

```
<opentarget>
    <txn
        id="xs:integer"
        msgIdx="xs:integer"
        msgTot="xs:integer"
        oracleTxnId="xs:string"
        commitTime="xs:dateTimeStamp"
        userId="xs:string" />

    <tbl
        name="xs:string"
        <cmd ops="xs:string">
            <row id="xs:string">
                <col name="xs:string"></col>
                <lkup>
                    <col name="xs:string"></col>
                </lkup>
            </row>
        </cmd>
    </tbl>
</opentarget>
```

表2: 操作テンプレートの説明 (* =オプション)

要素	属性	説明
txn		操作のトランザクションメタデータ
	id	現在のトランザクションのID
	msgIdx	トランザクション内の現在のレコードのインデックス
	msgTot*	トランザクション内のメッセージ総数
	oracleTxnId *	システム変更番号(SCN)から取得したOracleのトランザクションID
	commitTime*	トランザクションのコミットタイムスタンプ
	userId *	操作を実行したユーザID
tble		テーブルメタデータ
	name	完全修飾テーブル名
cmd		操作のメタデータ
	ops	操作のタイプ(挿入、アップデート、削除、トランケート)
row		操作で変更された行のメタデータ
	id	Oracle ROWID
col		列のデータを変更(操作で変更された列ごとに、これらの要素の1つが表示される)
	name	列名とその列の変更後の値
lkup		アップデートおよび削除操作で使用する変更前イメージ
col		列の変更前イメージ(操作で変更された列ごとに、これらの要素の1つが表示される)
	name	列名と、その列の変更前の値またはキー値(操作によって異なる)

注意: **id**および**msgIdx**属性を組み合わせて操作を一意に識別します。

サポートされているデータ型

Oracleのデータ型をXMLに変換する方法を示すチャートについては、『SharePlexリリースノート』を参照してください。

サンプルXMLレコード

ソーステーブル

これはサンプル操作が生成されるテーブルです。

```
SQL> desc products

 Name                  Null?                Type

 PRODUCT_ID            NOT NULL             NUMBER

 DESCRIPTION                                VARCHAR2(600)

 PRICE                                      NUMBER
```

ソースDML操作

```
insert into products values (230117, 'Hamsberry vintage tee, cherry', 4099);
commit;
update products set price=3599 where product_id=230117 and price=4099;
commit;
delete products where product_id=230117;
commit;
truncate table products;
```

スキーマレコード

```
<?xml version="1.0" encoding="UTF-8"?>
<?opentarget version="1.1"?>
<opentarget>
<txn id="2218316945" commitTime="2014-10-10T13:18:43" userId="85"
oracleTxnId="3.10.1339425" />
<tbl name="MFG.PRODUCTS" utcOffset="-5:00">
<cmd ops="schema">
<schema>
<col name="PRODUCT_ID" xmlType="decimal" key="true" nullable="false" length="22" />
<col name="DESCRIPTION" xmlType="string" key="false" nullable="true" length="600" />
<col name="PRICE" xmlType="decimal" key="false" nullable="true" length="22" />
</schema>
</cmd>
</tbl>
</opentarget>
```

挿入レコード

```
<?xml version="1.0" encoding="UTF-8"?>
<?opentarget version="1.1"?>
<opentarget>
<txn id="2218316945" msgIdx="1" msgTot="1" commitTime="2014-10-10T13:18:43"
userId="85" oracleTxnId="3.10.1339425" />
<tbl name="MFG.PRODUCTS">
<cmd ops="ins">
<row id="AAAmDbAAEAAApRrAAA">
<col name="PRODUCT_ID">230117</col>
<col name="DESCRIPTION">Hamsberry vintage tee, cherry</col>
<col name="PRICE">4099</col>
</row>
</cmd>
</tbl>
</opentarget>
```

更新レコード

```
<?xml version="1.0" encoding="UTF-8"?>
<?opentarget version="1.1"?>
<opentarget>
<txn id="2218318728" msgIdx="1" msgTot="1" commitTime="2014-10-10T13:19:12"
userId="85" oracleTxnId="1.17.970754" />
<tbl name="MFG.PRODUCTS">
<cmd ops="upd">
<row id="AAAmDbAAEAAApRrAAA">
<col name="PRICE">3599</col>
<lkup>
<col name="PRODUCT_ID">230117</col>
<col name="PRICE">4099</col>
</lkup>
</row>
</cmd>
</tbl>
</opentarget>
```

削除レコード

```
<?xml version="1.0" encoding="UTF-8"?>
<?opentarget version="1.1"?>
<opentarget>
<txn id="2218319446" msgIdx="1" msgTot="1" commitTime="2014-10-10T13:19:25"
userId="85" oracleTxnId="5.23.1391276" />
<tbl name="MFG.PRODUCTS">
<cmd ops="del">
<row id="AAAmDbAAEAAApRrAAA">
<lkup>
<col name="PRODUCT_ID">230117</col>
</lkup>
</row>
</cmd>
```

```
</tbl>
</opentarget>
```

トランケートレコード

```
<?xml version="1.0" encoding="UTF-8"?>
<?opentarget version="1.1"?>
<opentarget>
<txn id="2218319938" commitTime="1988-01-01T00:00:00" userId="85"
oracleTxnId="11.4.939801" />
<tbl name="MFG.PRODUCTS">
<cmd ops="trunc" />
</tbl>
</opentarget>
```

JSONレコードの形式

注意: これは、出力を読みやすくするために使用した**target x.kafka set json indent=2**を除き、デフォルトの出力です。

これはサンプル操作が生成されるテーブルです。

```
SQL> desc products

 Name                  Null?                Type

 PRODUCT_ID            NOT NULL             NUMBER

 DESCRIPTION                                VARCHAR2(600)

 PRICE                                      NUMBER
```

ソースDMLの操作

```
insert into products values (230117, 'Hamsberry vintage tee, cherry', 4099);
commit;
update products set price=3599 where product_id=230117 and price=4099;
commit;
delete products where product_id=230117;
commit;
truncate table products;
```

スキーマレコード

```
{
  "meta":{
    "op":"schema",
    "table":""
  },
  "schema":{
    "name":"BILL.PRODUCTS",
    "utcOffset":"-7:00",
    "PRODUCT_ID":{
```

```
      "jsonType":"decimal",
      "num":1,
      "key":1,
      "nullable":1,
      "length":22,
      "precision":0,
      "scale":0,
      "src_name":"PRODUCT_ID"
    },
    "DESCRIPTION":{
      "jsonType":"string",
      "num":2,
      "key":1,
      "nullable":1,
      "length":600,
      "precision":0,
      "scale":0,
      "src_name":"DESCRIPTION"
    },
    "PRICE":{
      "jsonType":"decimal",
      "num":3,
      "key":1,
      "nullable":1,
      "length":22,
      "precision":0,
      "scale":0,
      "src_name":"PRICE"
    }
  }
}
```

レコードの挿入

```
{
    "meta":{
       "op":"ins",
       "table":"BILL.PRODUCTS"
    },
    "data":{
       "PRODUCT_ID":230117,
       "DESCRIPTION":"Hamsberry vintage tee, cherry",
       "PRICE":4099
    }
}
```

レコードの更新

```
{
  "meta":{
    "op":"upd",
    "table":"BILL.PRODUCTS"
  },
```

```
  "data":{
    "PRICE":3599
  },
  "key":{
    "PRODUCT_ID":230117,
    "DESCRIPTION":"Hamsberry vintage tee, cherry",
    "PRICE":4099
  }
}
```

レコードの削除

```
{
    "meta":{
       "op":"del",
       "table":"BILL.PRODUCTS"
    },
    "data":{
       "PRODUCT_ID":230117,
       "DESCRIPTION":"Hamsberry vintage tee, cherry",
       "PRICE":3599
    }
 }
```

レコードの切り捨て

```
{
    "meta":{
       "op":"truncate",
       "table":"BILL.PRODUCTS"
    }
 }
```

JMSカテゴリ

サポート対象のターゲット

JMS

コマンドオプション

target x.jms [**queue** *queuename*]

{

set jms *property=value* |

reset jms [*property*] |

show jms

}

用途

JMSターゲットのプロパティを設定します。

プロパティ	入力値	デフォルト
factory_class=*factory_class*	*必須*。ファクトリクラスの完全修飾クラス名。JNDI環境プロパティ**java.naming.factory.initial**を設定して、プロバイダの初期コンテキストファクトリのクラス名を指定します。	None
provider_url=*url*	*必須*。オブジェクト名コンポーネントを持たないRMI URL。これは、JNDI環境プロパティ**java.naming.provider.url**を設定し、初期コンテキストとして使用されるレジストリの場所を指定します。 JMSプロバイダとURLのタイプに応じて正しいフォーマットを使用してください。例えばLDAPを使用する場合、URLは以下のようになります。 **ldap:**//*hostname.company.com/contextName* JMSプロバイダのURLについては、JMSプロバイダ管理者にお尋ねください。	None
lib_location=*path*	*必須*。クライアント・ライブラリ・ファイルをインストールしたディレクトリへのパス。	None
destination={**queue** | **topic**}	メッセージングドメイン。有効な値は、**queue**（ポート間）または**topic**（パブリッシャ・サブスクライバ・モデル）です。	**キュー**
factory_name=*factory_name*	JNDI接続ファクトリの検索名。カンマ区切りリストで複数の名前を指定することができます（例：jndi.name1, jndi.name2）。	None
user=*user*	JMSに接続しているユーザ名。認証が不要な場合は、このオプションと**password**オプションを省略します。	None
password=*password*	JMSユーザのパスワード。	None
queuename=*JMS_topic_queuename*	JMSキューまたはトピックの名前。	**OpenTarget**
persistent={**yes** | **no**}	**yes**は、送信操作の一部として、メッセージをディスクストレージにログ記録します。 **no**は、ログが記録されなくなります。	**yes**
session_transacted ={**yes** | **no**}	**no**は、複製された各メッセージに対してJMSコミットを発行するようPostlに指示し、各メッセージを即座に可視化して使用可能にします。これがデフォルトです。 **yes**は、トランザクション方式で動作するようにPostlに指示します。このモードでPostlは、以下に基づく間隔で、メッセージを可視化して使用可能にするJMSコミットを発行します。	**no**

プロパティ	入力値	デフォルト
	• SP_OPX_READRELEASE_INTERVALパラメーターに設定された値によって制御されるように、データがPostキューから読み取られてリリースされる前にコミットを発行します。 • SP_OPX_CHECKPOINT_FREQパラメーターに設定された値によって制御されるように、SP_OPX_READRELEASE_INTERVALに達するまで、JMSの書き込みごとにコミットを発行します。	
properties	使用しているJMSプロバイダが、SharePlexによって提供されるプロパティのデフォルトセットを含むメッセージを消費できない場合、このオプションを使用します。このオプションではプロパティを削除または追加することができます。プロパティをカンマ区切りリストで指定します。 • プロパティを追加するには、*name*=*value*と指定します。 • SharePlexプロパティを削除するには、名前の前にダッシュを付けます。例えば、この文字列は2つのSharePlexプロパティを削除します。-JMSXDeliveryCount, -JMSXGroupSeq	None
client_id	使用しているJMSプロバイダが、デフォルトのSharePlexクライアントIDを含むメッセージを使用できない場合は、このオプションを使用します。この値は、プロバイダが受け付けるクライアントIDを設定します。	None
commit_frequency	このオプションは、Postがトランザクション方式でJMSサーバにポストするように設定されている場合に使用します（**session_transacted**プロパティで指示されるようにすべてのメッセージの後ではなく、一定間隔でJMSコミットを発行）。このパラメーターはJMSコミットの間隔を指定します。SP_OPX_READRELEASE_INTERVALパラメーターと連動して動作します。有効な値は1から任意の正の整数です。	1,500

例

target x.jms set jms queuename=SharePlexJMS

JSONカテゴリ

サポート対象のターゲット

ファイル（**format record=json**）

Kafka（**format record=json**）

コマンドオプション

target x.{file | **kafka** } [**queue** *queuename*]

{

set json *property=value* |

reset json [*property*] |

show json

}

用途

format record =**json**のときのJSON出力のプロパティを設定します。

プロパティ	入力値	デフォルト

before	• **yes**に設定すると、行の変更前イメージが含まれます。
	変更前イメージが行末にある、以下のようなレコードが作成されます。
	`{"meta":{"op":"upd","table":"BILL.JSON"},"data":{"C2":"Bye"},`
	`"key":{"C1":"1","C2":"Hello"}}`
	`{"meta":{"op":"upd","table":"BILL.JSON"},`
	`"data":{"C1":"2"},"key":{"C1":"1"}}`
	• **no**に設定すると、行の変更前イメージが除外されます。以下のようなレコードが作成されます。
	`{"meta":{"op":"upd","table":"BILL.JSON"},"data":{"C2":"Bye"}}`
	`{"meta":{"op":"upd","table":"BILL.JSON"},"data":{"C1":"2"}}`
	ただし、Update操作の場合、更新されたレコードのキーを表示するには、「before」を「yes」に設定する必要があります。
		yes

commit	• **yes**に設定すると、コミットレコードが含まれます。以下のようなレコードが作成されます。 `{"meta":{"op":"commit","table":""}}` • **no**に設定すると、コミットレコードは省略されます。 **重要!commit**が**yes**に設定されている場合、対象となるトピックは1つだけです。	**yes**

ddl	• **yes**に設定すると、ALTER TABLEコマンドが含まれます。以下のようなレコードが作成されます。
	`{"meta":{"op":"ddl","table":"BILL.JSON"},`
	`"sql":{"ddl":"alter table \"BILL\".\"JSON\""}`
	• **No**に設定すると、ALTER TABLEコマンドは省略されます。
		yes

eol	**yes**に設定すると、JSONドキュメントの最後に行末文字が含まれます。 **no**に設定すると、行末文字が除外されます。	**yes**
indent	レコードのインデントのレベルを制御します。	**0**

プロパティ	入力値	デフォルト

	• **0**に設定すると、以下のようにレコード全体が1行に配置されます。
	
	`{"meta":{"op":"ins","table":"BILL.JSON"},`
	`"data":{"C1":"1","C2":"Hello ",`
	`"C3":"There","C4":"2017-03-20T09:46:34",`
	`"C5":"2017-03-20T09:46:34.735370000"}}`
	
	• 1以上に設定すると、メタデータ(含まれている場合)とデータ行が、指定した文字数だけ字下げされます。
	以下の例のように、レコードが論理的に複数の行に分割されます(indent= 4)。
	
	`{`
	`"meta":{`
	`"op":"ins",`
	`"table":"BILL.JSON"`
	`},`
	`"data":{`
	`"C1":"1",`
	`"C2":"Hello",`
	`"C3":"There",`
	`"C4":"2017-03-20T10:02:37",`
	`"C5":"2017-03-20T10:02:37.456949000"`
	`}`
	`}`	

meta	• **yes**に設定すると、JSONレコードのメタデータセクションが含まれます。以下のようなレコードが作成されます。
	
	`{"meta":{"op":"ins","table":"BILL.JSON"},`
	`"data":{"C1":"1","C2":"Hello","C3":"There",`
	`"C4":"2017-03-20T10:04:44",`
	`"C5":"2017-03-20T10:04:44.957758000"}}`
	`{"meta":{"op":"upd","table":"BILL.JSON"},"data":{"C2":"Bye"}}`
	`{"meta":{"op":"upd","table":"BILL.JSON"},"data":{"C1":"2"}}`
	`{"meta":{"op":"del","table":"BILL.JSON"},"data":{"C1":"2"}}`
	`{"meta":{"op":"commit","table":""}}`
	
	• **no**に設定すると、メタデータセクションは省略され、以下のようなレコードが作成されます。
	
	`{"data":{"C1":"1","C2":"Hello","C3":"There",`
	`"C4":"2017-03-20T10:05:09",`
	`"C5":"2017-03-20T10:05:09.268094000"}}`
	`{"data":{"C2":"Bye"}}`
	`{"data":{"C1":"2"}}`
	`{"data":{"C1":"2"}}`
	`{}`
	
	注意: レコードに表示するメタデータフィールドを指定するには、**set metadata** *property*コマンド
		yes

プロパティ	入力値	デフォルト
	を使用します。「metadataカテゴリ」を参照してください。 例: **target x.kafka set metadata time,userid,op,scn**	
schema	• **yes**に設定すると、スキーマレコードが含まれます。	**yes**

- **no**に設定すると、スキーマレコードが省略されます。

  レプリケーション中のオブジェクトのスキーマレコードは、新しいSharePlex設定がアクティベーションされたとき、またはテーブルでDDLの変更があったときに作成されます。
  yesの値（indent=4）に設定すると、以下のようなレコードが作成されます。

```
{
    "meta":{
        "op":"schema",
        "table":""
    },
    "schema":{
        "name":"BILL.JSON",
        "utcOffset":"-7:00",
        "C1":{
            "jsonType":"decimal",
            "num":1,
            "key":1,
            "nullable":0,
            "length":22,
            "precision":0,
            "scale":0,
            "src_name":"C1"
        },
        "C2":{
            "jsonType":"string",
            "num":2,
            "key":0,
            "nullable":1,
            "length":10,
            "precision":0,
            "scale":0,
            "src_name":"C2"
        },
        "C3":{
            "jsonType":"string",
            "num":3,
            "key":0,
            "nullable":1,
            "length":10,
            "precision":0,
            "scale":0,
            "src_name":"C3"
        }

  }
}
```

例

target x.kafka set json meta=no

kafkaカテゴリ

サポート対象のターゲット

kafka

コマンドオプション

target x.kafka [**queue** *queuename*]

{

set kafka *property=value* |

reset kafka [*property*] |

show kafka

}

用途

Kafkaターゲットのプロパティを設定します。

プロパティ	入力値	デフォルト
broker=*broker*	必須。Kafkaブローカーのホストおよびポート番号、または複数のブローカーのコンマ区切りリスト。このリストがKafkaクラスタへのブートストラップとなります。これらのブローカーのいずれかに接続できれば、Postはクラスタ内の他のブローカーを検出します。	**localhost:9092**
client_id=*ID*	オプション。呼び出しの追跡に役立つようにPostが各リクエストで送信するユーザ定義の文字列。	None
compression.code={**none**, **gzip**, **snappy**}	オプション。Kafkaでデータを圧縮するかどうかを制御します。オプションは**none**、**gzip**、または**snappy**です。	None
partition={*number* | **rotate** | **rotate trans**| **messagekey**}	**必須。次のいずれか:** • 固定のパーティション番号: 指定されたパーティション番号にのみメッセージをポストするようにPostに指示します。例えば0に設定すると、Postはパーティション0にのみポストするようになります。このオプションは、テストでの使用や、同じKafkaトピックにポストするデータの複数のチャンネルがターゲットにある場合の用途に適しています。	**0**

プロパティ	入力値	デフォルト
	• キーワード **rotate**: トピックのすべてのパーティションにラウンドロビン方式でメッセージを適用するようにPostに指示します。パーティションは、メッセージが新しくなるたびに変わります。例えば、あるトピックに3つのパーティションがある場合、メッセージはパーティション0、1、2、0、1、2…の順番にポストされます。 • キーワード **rotate trans**: これは**rotate**オプションと似ていますが、パーティションの番号が、メッセージごとではなくトランザクションごとに大きくなる点が異なります。例えば、あるトピックに3つのパーティションがある場合、コミットまではパーティション0に、コミットまではパーティション1にといった順番でメッセージがポストされます。このオプションは、複数のテーブルを1つのトピックに複製する場合に適しています。これにより、複数のパーティションにデータを分散させながら、トランザクションのすべての操作を単一のパーティションにまとめて保持することができます。その結果、単一のパーティションから読み取るコンシューマーは、完全なトランザクションのストリームを受け取ることができます。 • キーワード **messagekey**: パーティションへのメッセージのポストをPostに指示します。Kafkaのトピックはいくつかのパーティションに分かれています。これらのパーティションは、デフォルトのパーティションハッシュ関数に基づいて選択されます。ハッシュ値はmessagekeyに基づいて計算されます。messagekeyパーティションを使用して、同じキー値を持つすべてのメッセージを同じパーティションに入れます。 **注意:** • LOB列とCLOB列はKafkaパーティションキーとは見なされません。 • プライマリキー、一意キー、複合キー、または一意のインデックスを持たないテーブルでは、すべての列（LOB列とCLOB列を除く）がキー列と見なされます。このようなテーブルに対してalterクエリを実行すると、DDLステートメントはすべてのパーティションに複製され、後続のDMLステートメントは既存の列に基づいて特定のパーティションに送信されます。	

プロパティ	入力値	デフォルト

	• レプリケーションテーブルにキーが定義されていない場合、SharePlexはすべてのテーブル列をKafkaメッセージキーと見なします。キー以外のテーブルでは、SharePlexのユーザ定義キーを使用することをお勧めします。詳細については、『SharePlex管理者ガイド』の「**Define a Unique Key: PostgreSQL to PostgreSQL（一意キーの定義: PostgreSQLからPostgreSQLへ）**」のセクションを参照してください。
	• 複数のテーブルがレプリケーションに関与しているときに、特定のテーブルに異なるパーティションタイプを指定し、残りのテーブルをメッセージキーに基づいてパーティション化する場合、これらの特定のテーブルに名前付きpostキューを定義することができます。
	例:
	`target x.kafka queue <queue_name> set kafka partition={number/rotate/rotate trans}`
	残りのテーブルには以下のコマンドを使用します。
	`target x.kafka set kafka partition=messagekey`
	重要:
	パーティション化がメッセージキーに基づいて行われる場合、キー情報を含まないメッセージはKafka内部のハッシュ関数に従ってマッピングされます。これらのメッセージには、コミット、スキーマ、ロールバック、セーブポイント、およびDDLステートメントが含まれることがあります。
	レプリケーション中にパーティション数が増えると、キーからパーティションへの既存のマッピングは無効になります。
	いくつかの列がインデックスとして機能し、他の制約が定義されていないテーブルの場合、SharePlex設定ファイルで一意キーとしてこれらのインデックスを使用します。
	例えば、以下のテーブルでは2つの列（IDとNAME）に一意のインデックスが定義されています。
	`create table mytable(ID NUMBER(25,2),NAME CHAR(200),COL_VARCHAR2 VARCHAR2(400),COL_RAW RAW(1000));`
	`CREATE INDEX indx_mytable ON mytable (ID,NAME);`
	SharePlex設定ファイルで、インデックス列を一意キーとして	

プロパティ	入力値	デフォルト
	定義します。 **datasource:o.**SID src.mytable !key(ID,NAME) host 詳細については、『SharePlex管理者ガイド』の「**Define a Unique Key: Oracle to Oracle（一意キーの定義: Oracle からOracleへ）**」のセクションを参照してください。 制約やインデックスが定義されていないテーブルについては、SharePlexでの設定中に一意キーを定義することができます。 複合キーを持つテーブルの場合、キー値のいずれかが変更されると、その変更メッセージは現在のパーティションに置かれます。それ以降のメッセージは同じパーティションに割り当てられることもあれば、割り当てられないこともあります。	
request.required.acks=*value*	オプション。これはKafkaクライアントのパラメーターです。デフォルトでは-1に設定されており、**all**を意味します。実際には**all**は *all in-sync replicas*を意味するため、これについてはKafkaのドキュメントを参照してください。このパラメーターは、**min.insync.replicas** brokerパラメーターと併用することで、可用性とデータの一貫性の関係を調整することができます。 **重要:** これらの設定によっては、Kafkaプロデューサー（この場合はSharePlex）とKafkaクラスタの間でデータが失われる可能性があります。	**-1**
topic=*topic_name*	必須。ターゲットのKafkaトピックの名前。 この文字列には特殊なシーケンス**%o**または**%t**を含めることができます。**%o**のシーケンスは、複製されるテーブルの所有者名に置き換えられます。**%t**のシーケンスは、複製されるテーブルのテーブル名に置き換えられます。この機能は、Kafkaサーバの設定**auto.create.topics.enabled**が「**true**」に設定されている場合に使用できます。また、トピックが自動的に生成されるときのデフォルトとして使用されるため、**default.replication.factor**と**num.partitions**のサーバ設定も確認してください。 **重要!** 複数のトピックを使用する場合は、**target**コマンドで以下のプロパティも設定する必要があります。 • 出力はJSONでなければなりません。**format**カテゴリの**record**プロパティを**json**に設定します。 **target x.kafka set format record=json** • コミットは無効にしなければなりません。**json**カテゴリの**commit**プロパティを**no**に設定します。 **target x.kafka set json commit=no**	**shareplex**

*レイテンシを避けるために、Postはそれ以上着信メッセージがないことを検知すると、しきい値が満たされるのを待たずにKafkaに即座にパケットを送信します。

例

```
sp_ctrl> target x.kafka set kafka broker=host1:9092,host2:9092,host3:9092
sp_ctrl> target x.kafka set kafka topic=shareplex
```

metadataカテゴリ

サポート対象のターゲット

Oracleレプリケーションターゲット

Oracle変更履歴ターゲット

HANAレプリケーションターゲット

ファイル

JMS

kafka

コマンドオプション

target {**o.***SID* | **c.***SID* | **r**.*database* | **x.file | x.jms** | **x.kafka**} [**queue** *queuename*]

{

set metadata [*colname*:]*property*[, ...] |

reset metadata |

show metadata

}

用途

set metadataを指定して**target**コマンドを使用すると、データベースターゲット、XMLまたはJSONファイルターゲット、JMSターゲット、Kafkaターゲットに設定されているデフォルトのメタデータプロパティをオーバーライドできます。これらのターゲットタイプはすべて、Postの出力にメタデータを提供します。

- **OracleおよびHANAレプリケーションターゲット:** メタデータをOracleおよびHANAレプリケーションターゲットに設定できます。メタデータの列はターゲット上に存在していなければなりません。SharePlexはメタデータ列を追加しません。
- **Oracle変更履歴ターゲット: target**の指定がない場合、Postはデフォルトのメタデータ列を更新します。デフォルトのメタデータ列は、レプリケーションを開始する前にデフォルトの名前で存在していなければなりません。以下のテーブルでは、プロパティがデフォルトで変更履歴ターゲットに含まれている場合、デフォルトの列名がリストされます。**set metadata**を使用するのは、メタデータを追加したり、列にカスタム名を割り当てる場合のみです（*colname*オプションを使用）。

- XML入力をサポートするターゲット（JMS、Kafka、ファイル）の場合、メタデータはXMLスキーマに従ってフォーマットされます。以下の表に示すように、特定のメタデータはデフォルトで出力に含まれます。**set metadata**を使う必要はありません。**set metadata**は、メタデータを追加する場合にのみ使用します。
- JSON入力をサポートするターゲットの場合（Kafka）、**json meta=no**を設定しない限り、**op**および**table**プロパティがデフォルトで含まれます。

変更履歴ターゲットデータベースをサポートするためのSharePlexの設定については、『SharePlex管理者ガイド』を参照してください。

複数のメタデータプロパティを設定するには、カンマ区切りリストとして指定します。

プロパティ	Postにより挿入される値	以下について有効
time	ソースに操作が適用された時刻（Oracleにより提供） **重要!** Oracleバージョン11g以降では、時刻はトランザクションの開始など、トランザクションの境界レコードにのみ表示されます。OracleのREDOログの線形設計と、SharePlexがトランザクションの順序で操作をレプリケートする方法の両方に基づいて、SharePlexは、異なるタイムスタンプを持つレコードが見つかるまで、元のレコードに含まれるタイムスタンプを後続のレコードに適用します。 **デフォルトの列名**: SHAREPLEX_SOURCE_TIME **データ型**: TIMESTAMP このプロパティは、XML出力と変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSONおよびXML出力
userid	操作を実行したユーザID **デフォルトの列名**: SHAREPLEX_SOURCE_USERID **データ型**: NUMBER このプロパティは、XML出力と変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSONおよびXML出力
op	操作の種類（INSERT、UPDATE、DELETE、TRUNCATE、DROP COLUMN、UPDATE BEFORE、UPDATE AFTER）。JMSの実装の場合、このフィールドは、常に**SPOps**という名前のフィールドとしてXML出力に追加されます。これをJMSプロパティに追加するには、**op**オプションを指定して明示的に**target**コマンドを発行します。 **デフォルトの列名**: SHAREPLEX_SOURCE_OPERATION **データ型**: VARCHAR2 このプロパティは、XMLおよびJSON出力と変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSONおよびXML出力
scn	操作が適用されたときのソースSCN デフォルトの列名: SHAREPLEX_SOURCE_SCN データ型: NUMBER	Oracleレプリケーションターゲット 変更履歴ターゲット

プロパティ	Postにより挿入される値	以下について有効
	このプロパティは、変更履歴テーブルにデフォルトで含まれます。	HANAレプリケーションターゲット JSON出力
rowid	変更された行のROWID デフォルトの列名: SHAREPLEX_SOURCE_ROWID データ型: ROWID このプロパティは、変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSON出力
trans	操作のトランザクションID。JMSの実装の場合、このフィールドは、**SPTxnId**という名前のフィールドとしてXML出力とJMSプロパティの両方に追加されます。 デフォルトの列名: SHAREPLEX_SOURCE_TRANS データ型: VARCHAR2 このプロパティは、変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSON出力
seq	トランザクション内での操作の順序 デフォルトの列名: SHAREPLEX_OPERATION_SEQ データ型: NUMBER このプロパティは、変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSON出力
host	ソースホストの名前またはIPアドレス デフォルトの列名: SHAREPLEX_SOURCE_HOST データ型: VARCHAR2 このプロパティは、変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSON出力
posttime	操作がターゲットにポストされた時刻	JSON出力
queue	SharePlexキューの名前 デフォルトの列名: SHAREPLEX_QUEUENAME データ型: VARCHAR2 このプロパティは、変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSON出力
source	**target**の**set source**オプションで設定されたユーザ定義のソース識別子。「sourceカテゴリ」を参照してください。 デフォルトの列名: SHAREPLEX_SOURCE_ID データ型: VARCHAR2 このプロパティは、変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット JSON出力

プロパティ	Postにより挿入される値	以下について有効
changeid	操作のユニークな連続ID デフォルトの列名: SHAREPLEX_CHANGE_ID データ型: NUMBER このプロパティは、変更履歴テーブルにデフォルトで含まれます。	Oracleレプリケーションターゲット 変更履歴ターゲット
size	トランザクション内の操作数	JSON出力 kafka
table	ターゲットテーブルの名前。ターゲットテーブルの名前は、**SPTblName**という名前のフィールドとしてJMSプロパティに追加されます。 このプロパティは、JSON出力にデフォルトで含まれます。	JMS JSONおよびXML出力
idx	トランザクション内のこの操作のインデックス。これは、**SPMsgIdx**という名前のフィールドとしてJMSプロパティに追加されます。JMSプロパティでは*m*/*n*の形式になります。ここで、*n*はトランザクション内の操作の総数、*m*はトランザクション内の操作番号です。	JMS

JMSとKafkaのset metadataの例

以下では、ソースのIPアドレスまたはホスト名、SharePlexキューの名前、およびトランザクションの操作数をJMSターゲットの出力に追加するように、すべてのPostプロセスを設定しています。

target x.jms set metadata host, queue, size

または...

target x.kafka set metadata host, queue, size

Oracleのset metadataの例

以下の例では、操作の時刻とそれを実行したユーザIDを**Timestamp**および**User**列に追加するように、**myqueue1**キューに関連付けられたPostプロセスを設定しています。これらの列は、いずれも大文字と小文字が区別されるユーザ定義の名前です。

target c.targSID queue myqueue1 set metadata time:"Timestamp", userid:"User"

resourceカテゴリ

サポート対象のターゲット

すべてのターゲット

コマンドオプション

target {**o.** | **r.** | **x.** | **c.**} *target* [**queue** *queuename*]

{

set resources *property=value* |

reset resources [*property*] |

show resources [**for** *datastore*]

}

用途

ターゲットシステムのリソースに影響を与えるパラメーターを設定します。

プロパティ	入力値	デフォルト
commit_frequency=*number_of_operations*	Postがコミットを発行するまでの操作の最大回数を指定します。2以上の任意の整数に設定できます。大きなトランザクションを、指定された*number_of_operations*以下の操作をそれぞれが含む2つ以上の小さなトランザクションに分割するために使用できます。このオプションでは、トランザクションごとに許可される行ロックの数など、大規模なトランザクションに影響するリソース制限を回避することができます。 **注意:**「commit_frequency」プロパティは、JMSターゲットでのみサポートされています。	50000
max_active_statements=*number_of_cursors*	1つのセッションでPostがターゲットデータベースに対して同時に処理できるアクティブなSQLステートメントの数を指定します。データベースまたはクライアントドライバで許可されるアクティブなステートメントの最大数を上限とする任意の整数を指定できます。これは、SP_OPX_SQL_CACHE_DISABLEパラメーターが0（有効）に設定されている場合にのみPostで使用されます。	16

例

target r.mydb queue q1 set resources commit_frequency=10000

target r.mydb queue q1 set resources max_active_statements=10

rule | filterカテゴリ

サポート対象のターゲット

Oracle変更履歴ターゲット

コマンドオプション

target c.*SID* [**queue** *queuename*]

table *tablename*

{

set {**rule** | **filter**} { '*column_condition*' | !*filename* }

reset {**rule** | **filter**}

show {**rule** | **filter**}

}

用途

特定のテーブルに追跡ルールやフィルターを適用する条件文を指定します。UPDATE操作に適用されます。追跡するデータをターゲットテーブルに限定し、関心のないデータは除外する条件付きロジックを適用します。1つのターゲットテーブルごとに1つの**rule**または**filter**だけを指定できますが、括弧を使用した入れ子にした式や、AND、OR、およびNOTの論理結合子と組み合わせてさまざまな条件を作成することができます。

構文要素	説明
rule | **filter**	• **rule**は、UPDATE操作が*column_condition*に一致する場合にのみ、行を挿入するようPostlに指示します。UPDATEが列条件に一致しない場合、その行は破棄されます。 • **filter**は、UPDATEが*column_condition*に一致しない場合にのみ、行を挿入するようPostlに指示します。UPDATEが列条件に一致した場合、その行は破棄されます。
table *tablename*	ルールまたはフィルターを設定するターゲットテーブルの名前。
column_condition	UPDATE操作により満たす必要のある条件付きロジック。*column_condition*は条件文全体を表し、一重引用符で囲む必要があります。列の条件は、以下の1つの式、または任意の組み合わせにすることができます(それぞれを括弧で囲む)。 (*column* **is changed**) (*column* **is not changed**) ([*column1*, *column2*, ...] **matches change_list**) ([*column1*, *column2*, ...] **contains change_list**) ここで: • *column*はターゲット列の名前です。リスト内の列名の区切りにはカンマを使い、リストは括弧で囲みます。 • **is changed**は、指定された列がUPDATE操作で変更されることを意味します。 • **is not changed**は、指定された列がUPDATE操作で変更されないことを意味します。 • **change_list**は、変更されるソース列の論理表現です。 • **matches change_list**が満たされるのは、条件が変更リストと完全に一致する場合です(変更リストより少ない、または多い列は対象とならない、同じ名前のみ)。 • **contains change_list**は、変更リストのすべての列が条件に含まれる場合に満たされます。変更リストは条件の列のサブセットにすることができますが、変更リストより多くの列を含めることはできません。名前は一致していなければなりません。
!*filename*	大きな列の条件をテキストファイルに保存し、**target**コマンドを実行する際に列条件の構文全体ではなく、そのファイルを指定できるようにします。

注意: **rule**/**filter**機能では、バッチおよび直接ロード操作はサポートされていません。

例

これらのコマンドは、式を組み合わせてさまざまなルールを設定する方法の例を示しています。

target c.mySIDtable mytable set rule '(col3 is not changed) and (col5 is not changed)'

target c.mySIDtable mytable set rule '([col1, col3] matches change_list)'

target c.mySIDtable mytable set rule 'not ([col2, col5] contains change_list)'

次の表は、行を含めたり破棄したりするために**rule**と**filter**を機能させるさまざまな方法を示しています。

ルール	更新された列	列が挿入されたかどうか
rule = '(c2 is not changed)'	c2	no
rule = '(c2 is not changed)'	c3	yes
rule = '(c2 is changed)'	c2	yes
rule = '(c2 is changed or c4 is changed)'	c1	no
rule = '(c2 is changed or c4 is changed)'	c1,c3,c5	no
rule = '(c2 is changed or c4 is changed)'	c2	yes
rule = '(c2 is changed or c4 is changed)'	c3,c4,c5	yes
rule = '([c3,c6,c7,c8] matches change_list)'	c3,c6,c7,c8	yes
rule = '([c3,c6] matches change_list)'	c3,c6,c7	no
rule = '([c3,c6,c7] contains change_list)'	c3,c6	yes
rule = '([c3,c6] contains change_list)'	c3,c8	no

フィルタ	更新された列	列が挿入されたかどうか
filter = '(c2 is not changed)'	c2	yes
filter = '(c2 is not changed)'	c3	no
filter = '(c3 is changed and c4 is changed)'	c1	yes
filter = '(c3 is changed and c4 is changed)'	c3,c4	no
filter = '(c3 is changed and c4 is changed)'	c1,c2,c3	yes
filter = '([c3,c6,c7] matches change_list)'	c3,c6	yes
filter = '([c3,c6,c7] matches change_list)'	c3,c6,c7	no
filter = '([c3,c6] contains change_list)'	c3,c6	no
filter = '([c3,c6,c7] contains change_list)'	c3,c8	yes

sourceカテゴリ

サポート対象のターゲット

Oracle変更履歴ターゲット

コマンドオプション

target c.*SID* [**queue** *queuename*]

{

set source *sourceID* |

reset source |

show source

}

用途

SHAREPLEX_SOURCE_IDメタデータ列のユーザ定義のソース識別子を設定します。**set metadata source**オプションを指定して**target**を使用し、この列を入力するようにPostを設定します。任意の英数字文字列を1つ指定します。

例

以下に、**set source**を使用して、複数のソースデータベースのデータを同じターゲットデータベースにレプリケートする設定をサポートするための方法を示します。このターゲットの設定により、各Postプロセスは、ターゲットデータベース内のデータを区別するために、ソースデータベースに応じて異なるソースIDでSHAREPLEX_SOURCE_ID列を更新します。この例では4つのPostプロセスを使用しています。

target c.target1 queue source718 set source east

target c.target1 queue source92 set source south

target c.target1 queue source101 set source west

target c.target1 queue source75 set source north

キュー「source718」のPostプロセスは、ストリーム内のテーブルへの挿入または更新のたびに、列SHAREPLEX_SOURCE_IDを値「east」で更新します。キュー「source92」のPostプロセスは、列SHAREPLEX_SOURCE_IDを値「south」で更新します。

SQLカテゴリ

サポート対象のターゲット

ファイル（**format record=sql**）

コマンドオプション

target x.file [**queue** *queuename*]

{

set sql *property=value* |

reset sql [*property*] |

show sql

}

用途

formatrecord=sqlのときのSQL出力のプロパティを設定します。

プロパティ	入力値	デフォルト
add_rownum={**yes** | **no**}	行番号を含めるには**yes**に、除外するには**no**に設定します。	**yes**
begin_transaction={**yes** | **no**}	開始トランザクションレコードを含めるには**yes**に、除外するには**no**に設定します。	**no**
comment=*character*	コメントを示す文字	**--**
concatenate=*character*	文字列を連結する文字	**||**
end_transaction={**yes** | **no**}	終了トランザクションレコードを含めるには**yes**に、除外するには**no**に設定します。	**no**
legacy={**yes** | **no**}	*SharePlex 8.6*で必要です。レガシーSQLの日付とタイムスタンプの形式を使用します。 `MMDDYYYYHH24MISS`および`MMDDYYYYHH24MISS.FFFFFF` (`yes`/`no`)	**no**
name_delimiter=*character*	SID、テーブル、オーナー、列名を区切る文字。この値は、PostgresがODBCドライバにSQL_DELIMITERの設定を問い合わせたときに返される値をオーバーライドします。	**なし**
record_terminator=*character*	SQLを終了する文字	**;**

例

target x.file set sql comment=#

trace capture

trace captureコマンドは、Captureによる処理とパフォーマンスの統計情報をトレースファイルに出力するために使用します。

Captureのトレースには以下が含まれます。

- トレース自体の詳細
- Captureが、さまざまなタスク全体とスレッドごとに費やした時間

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**trace read**、**trace post**

構文

基本コマンド	コマンドオプション	リモートオプション
trace capture	[*minutes*]	[**on** *host* |
	[**for** *datasource*]	**on** *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
minutes	トレースを実行する時間(分)。デフォルトは15分です。
for *datasource*	Captureのトレースを実行するデータソース。同じSharePlexインスタンスで複数のCaptureを実行する場合にのみ必要です。使用する場合は、*minutes*に従う必要があります。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

出力

トレースが完了すると、プロセスは、変数データディレクトリの**log**サブディレクトリにあるファイルにトレースデータを以下の形式で書き込みます。

process_id`_trace_`*time*

例:

`orcl_ocap_trace_Sep30_15_24_2014`

ここで、`orcl`はデータソース、`ocap`はCaptureプロセスの名前です。

例

以下のコマンドは、データソース**myora**のトレースを1分間実行します。

trace capture 1 for myora

このコマンドは、以下のようなトレース出力を生成します。

```
CAPTURE TRACE: Started:   Tue Sep 30 15:23:20 2014
               Completed: Tue Sep 30 15:24:21 2014

   Time Elapsed     #Operations       Kbytes read     Kbytes processed    %Processed      Latency
           1:01           64025            162486                36971           22%            2

   Time breakdown:
         63% - Idle waiting for data
          4% - Process the redo log record
          3% - Write to the queue

   Log reader threads:  %Data read  Time spent in thread
         Instance  1           66%       21% - Idle waiting for data, 47% - Getting data
         Instance  2            5%       19% - Idle waiting for data, 27% - Getting data, 26% - Querying Oracle
         Instance  3           27%       36% - Idle waiting for data, 34% - Getting data, 16% - Querying Oracle

   RAC threads:         %Data read  Time spent in thread
         Instance  1           66%       78% - Idle waiting for data, 16% - Waiting to send data
         Instance  2            5%       72% - Idle waiting for data, 26% - Waiting for next log
         Instance  3           27%       72% - Idle waiting for data, 26% - Waiting for next log

   Sequencer thread:                     85% - Idle waiting for data
```

trace post

trace postコマンドは、Postによる処理とパフォーマンスの統計情報をトレースファイルに出力するために使用します。

Postのトレースには以下が含まれます。

- トレース自体の詳細
- トランザクション活動に関する統計情報
- 操作タイプごとの詳細な統計情報

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ターゲットシステム
関連コマンド:	**trace capture**、**trace read**

構文

基本コマンド	コマンドオプション	リモートオプション
trace post	[*minutes*]	[**on** *host* |
	[**for** *datasource-datadest*]	**on** *host:portnumber* |
	[**queue** *queuename*]	**on** *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
minutes	トレースを実行する時間(分)。デフォルトは15分です。
for *datasource-datadest*	ソースデータストアとターゲットデータストアで指定される特定のPostプロセスにトレースを限定します。 • 同じSharePlexインスタンスで複数のPostプロセスを実行する場合(異なるソースからのレプリケーション)、または異なるターゲットデータベースに複数のPostプロセスを実行する場合に必要です。使用する場合は、*minutes*に従う必要があります。 • *datasource*は**o**.*SID*と表します。ここで、*SID*はORACLE_SIDです。

コンポーネント	説明
	• *datadest*は、ターゲットによって以下のいずれかで表されます。 **o**.*ORACLE_SID* **r.***database_name* **x.kafka** **x.jms** **x.file**
queue *queuename*	指定された名前付きキューに関連付けられているPostプロセスにトレースを制約します。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

出力

トレースが完了すると、プロセスは、変数データディレクトリの**log**サブディレクトリにあるファイルにトレースデータを以下の形式で書き込みます。

*process_id*_trace_*time*

例:

orcl_orcl2_opo_trace_Feb_5_17_24_2014

ここで、orclはソースデータソース、orcl2はターゲットデータソース、opoはPostプロセスの名前です。

例

以下のコマンドは、データソース**myora**と**myora2**のトレースを1分間実行します。

trace post 1 for myora-myora2

このコマンドは、以下のようなトレース出力を生成します。

```
POST TRACE: Started:   Wed Jul 23 13:04:04 2014
            Completed: Wed Jul 23 13:05:05 2014

  Summary
  -------
     Rate (operations/sec)                       126.6
     Transaction rate (transactions/sec)          15.2
     Average transaction size                      8.3

     Elapsed time                                 1:01

     Time spent in SharePlex
     -----------------------
          Read from the queue                     0:00
          Time spent with SHAREPLEX_TRANS table   0:04

     Time spent in Oracle
     --------------------
          Oracle execute                          0:05
          Oracle commit                           0:00
          Update LOB columns                      0:00


  Detail
  ------
                        Number Executions              Cost
     Operation                 Count  Pct       Seconds  Pct
     ---------                 -----  ---       -------  ---
     Insert                     2255  29%             0  10%
     Update                      514   6%             0   7%
     Delete                       75   0%             0   1%
     Commit                      914  12%             1  30%
     Direct load                1511  19%             0   0%
     Update Lob                  207   2%             2  49%
     Batch Insert               1998  26%             0   0%
     Batch Delete                123   1%             0   0%

     Operation            Table                    Avg(microsec)        Count    Total(sec)
     ---------            -----                    -------------        -----    ----------
     Insert               "TEST"."SRC_TEST4"                 262          362             0
     Insert               "TEST"."SRC_TEST1"                 256          370             0
     Insert               "TEST"."SRC_TEST6"                 252          381             0
     Insert               "TEST"."SRC_TEST3"                 249          389             0
     Insert               "TEST"."SRC_TEST5"                 240          378             0

     Update               "TEST"."SRC_TEST6"                1585           16             0
     Update               "TEST"."SRC_TEST4"                1301           26             0
     Update               "TEST"."SRC_TEST5"                1091           27             0
     Update               "TEST"."SRC_TEST2"                 752           22             0
     Update               "TEST"."SRC_TEST3"                 705          198             0

     Delete               "TEST"."SRC_TEST3"                 747           30             0
     Delete               "TEST"."SRC_TEST1"                 635           26             0

     Commit                                                 1858          914             1

     Direct load          "TEST"."SRC_TEST1"                   4         1511             0

     Update Lob           "TEST"."BARRS1000"               15698           22             0
     Update Lob           "TEST"."BARRS21"                 15202           19             0
     Update Lob           "TEST"."BARRS210"                12315           18             0
     Update Lob           "TEST"."BARRS2000"               11170           22             0
     Update Lob           "TEST"."BARRS20"                 10937           29             0

     Batch Insert         "TEST"."SRC_TEST3"                  26         1998             0

     Batch Delete         "TEST"."SRC_TEST1"                 108           66             0
     Batch Delete         "TEST"."SRC_TEST3"                 101           57             0
```

trace read

trace readコマンドは、Readによる処理とパフォーマンスの統計情報をトレースファイルに出力するために使用します。

Readのトレースには以下が含まれます。

- キーの値を取得するために行われたクエリの詳細
- Readがさまざまな処理タスクに費やした時間を示す統計情報

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**trace capture**、**trace post**

構文

基本コマンド	コマンドオプション	リモートオプション
trace read	[*minutes*]	[**on** *host* |
	[**for** *datasource*]	**on** *host:portnumber* |
		on *login/password@host* |
		on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
minutes	トレースを実行する時間(分)。デフォルトは15分です。
for *datasource*	トレースを実行するデータソース。同じSharePlexインスタンスで複数のCaptureおよびReadプロセスを実行する場合にのみ必要です。使用する場合は、*minutes*に従う必要があります。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

出力

トレースが完了すると、プロセスは、変数データディレクトリの**log**サブディレクトリにあるファイルにトレースデータを以下の形式で書き込みます。

process_id`_trace_`*time*

例:

```
orcl_ord_trace_Sep30_13_07_2014
```

ここで、`orcl`はデータソース、`ord`はReadプロセスの名前です。

例

以下のコマンドは、データソース**myora**のトレースを1分間実行します。

trace read 1 for myora

このコマンドは、以下のようなトレース出力を生成します。

```
READ TRACE: Started:   Tue Sep 30 13:06:41 2014
            Completed: Tue Sep 30 13:07:41 2014

============================ Queries for Keys from ORACLE ============================
  Object Name                                                (Object ID)  #Queries #Cols
  ---------------------------------------------------------------------- --------- -----
  "TEST"."HP_TEST2"                                          (   155549)       501     1
  "TEST"."HP_TEST3"                                          (   155550)       485     1
  "TEST"."HP_TEST1"                                          (   155548)       421     1
  "TEST"."HP_TEST4"                                          (   155551)       331     1

============================ Queries for Keys from CACHE  ============================
  Object Name                                                (Object ID)  #Queries #Cols
  ---------------------------------------------------------------------- --------- -----
  "TEST"."HP_TEST1"                                          (   155548)       399     3
  "TEST"."HP_TEST4"                                          (   155551)       349     3
  "TEST"."HP_TEST2"                                          (   155549)       339     3
  "TEST"."HP_TEST3"                                          (   155550)       324     3

================================ Statistics  =================================
  Total Time        Total op      Oracle time       SharePlex time       %Idle
       1:00           19391             0:03                 0:05         76%

   Time breakdown:
         76% - Idle waiting for data
          3% - Checkpointing transaction cache
          3% - Evaluating HP column conditions
          3% - Creating Read Consistency Markers
```

truncate log

truncate logコマンドは、SharePlex変数ディレクトリ内の**log**サブディレクトリのイベントログとトレース・ログ・ファイルを切り捨てる(すべてのデータを削除する)ために使用します。これらのログはデータを蓄積し、最終的には大量のディスクスペースを消費することがあります。**truncate log**コマンドは、データが有用性を失った後にログを切り捨てることができます。このコマンドはレプリケーションには影響を与えません。レプリケーションがアクティブで、データがキューにあるときに発行できます。

sp_eventmon実行中のログの切り捨て

sp_eventmon監視スクリプトが実行されている場合に、**truncate log**コマンドを実行して*sp_cop_name*.**mrk**ファイルを切り捨てます。ここで、*sp_cop_name*は、スクリプトの実行時に-s引数で使用された値です。このファイルは、SharePlexの製品ディレクトリの**util**サブディレクトリにあります。

重要: **truncate log**コマンドは、すべてのエントリをログから削除します。お客様からサポートのお問合せがあったときに、Questテクニカルサポートはログの情報を参照いたしますので、できるだけ長期間データを保持するようにしてください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	管理者(1)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	なし

構文

基本コマンド	リモートオプション
truncate log	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

typemap

typemapコマンドは、データ型マッピングを表示するために使用します。データ型マップは、ソース上の列のデータ型をターゲット上の対応するデータ型にマッピングするために使用します。SharePlexでは、DDLをレプリケートする際やターゲット上でテーブルを作成する際にデータ型マップを使用します。例えば、PostはALTER TABLE...ADD COLUMN DDL操作をレプリケートする際に、ソースとターゲットの組み合わせに対応するデータ型マップを使用してターゲット列のデータ型を決定します。

データ型マップには、ソースとターゲットの組み合わせに関するルールのリストが含まれます。各ルールには、Postがソースのデータ型とのマッチングに使用するパターンがあります。パターンが一致すれば、対応するパターンを使用してターゲットのデータ型が構築されます。

ルールの例

Source（ソース）	Target（ターゲット）	
CHAR(n)	VARCHAR(n)	ソースからのDDLでCHARとして定義された列は、ターゲット上では同じ長さ（n）でVARCHARにマッピングされます。
VARCHAR (4000:)	CLOB	長さが4,000以上のVARCHARとして定義された列は、ターゲット上でCLOBデータ型にマッピングされます。

データ型マップを表示するには: 現在のtypemapを表示するには**show**オプションを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター（2）
発行場所:	ターゲットシステム
関連コマンド:	**connection**、**target**

構文

基本コマンド	コマンドオプション	リモートオプション
typemap show	[*source-target*]	利用不可

構文の説明

コンポーネント	説明
show	現在の型マップ情報の一部またはすべてを表示するには、このオプションを使用します。 例: **typemap show**はすべての型マップを表示します。 **typemap show oracle-sqlserver**はOracleからSQL Serverへの型マップを表示します。
source-target	*source-target*は表示するソース-ターゲットマップを指定します。

usage

SharePlexコマンドの構文を表示するには、**usage**コマンドを実行します。コマンド全体を入力することも、最初の数キーワードだけを入力することもできます。例えば「**usage compare**」と入力すると、**compare using**コマンドと**compare**コマンドの両方の構文が表示されます。

コマンドやコマンドグループの構文と説明を表示するには、helpコマンドを使用します。このコマンドには[**on** *host*]オプションはありません。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**help**

構文

基本コマンド	リモートオプション
usage *commandname*	利用不可

構文の説明

コンポーネント	説明
commandname	構文を表示するコマンド **例**: `sp_ctrl(sysA)>` **usage stop**

verify config

verify configコマンドの目的は、アクティベーションとレプリケーションの特定の問題を回避するための予防措置として使用することです。これは、アクティベーションが正常に完了することを確認するためのテストに使用することを意図しています。

verify configコマンドはテーブルとシーケンスのみを検証します。

このコマンド以下に使用できます。

- 設定ファイル内のエントリの構文を確認する。
- SharePlexによるレプリケーションでソースオブジェクトがサポートされていない場合にエラーを報告する。
- ルートで指定されたホスト名に到達できない場合に報告する。
- 1つのオブジェクトに重複した指定がある場合に報告する。
- オブジェクトの指定がスキップされるかどうかと、その理由を報告する。

verify configコマンドがサポートしていないもの

verify configは以下を実行しません。

- アクティベーション時刻の検証
- ターゲットオブジェクトまたはターゲットSIDの検証

アクティブな設定で追加または変更されたオブジェクトの検証

アクティブな設定に追加するオブジェクトや変更するオブジェクト（ルーティングの変更など）を検証するには、アクティブな設定をコピーして変更し、そのコピーに対して**verifyコマンド**を実行することをお勧めします。

分割レプリケーションでの使用

分割レプリケーションを使用するテーブルを含む設定では、verify configコマンドを使用できます。SharePlexは分割レプリケーションでの**verify config**の使用をサポートしていますが、パーティションやその定義を検証するための追加機能は含まれません。

検証結果の表示

verify configコマンドは、検証が完了するまでsp_ctrlインターフェイスの制御を保持します。

このコマンドは設定ファイル全体を読み取り、エラーや重複を記録し、ワイルドカードに該当するオブジェクトを展開します。

検証の結果は**sp_ctrl**内の画面に表示されます。

詳細な結果を表示するには、以下を実行します。

- **sp_ctrl**で、**detail**オプションを使用してverify**config**コマンドを発行します。
- **verify config**コマンドを発行した後、画面に表示されるパスを使用して結果ファイルに直接移動します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**activate config**、**copy config**、**create config**、**edit config**

構文

基本コマンド	コマンドオプション	リモートオプション
verify config *filename*	**detail**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	*filename*は、検証する設定の名前です。
detail	このオプションは、より詳細なレベルを画面に表示します。 **例**: `sp_ctrl(sysA)` **> verify config myconfig detail** この例では**myconfig**ファイルが検証され、その結果がより詳細なレベルで表示されます。

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>` **status on SysA**

オプション	説明
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

version

システムのSharePlexソフトウェアのバージョン番号を表示するには、versionコマンドを使用します。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	なし

構文

基本コマンド	リモートオプション
version	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**

オプション	説明
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>` **status on john/spot5489@SysA:8304**

view config

sp_ctrlインターフェイス内からアクティブまたは非アクティブな設定ファイルの内容を表示するには、**view config**コマンドを使用します。

使用のタイミング:

- 正しく記述されていない設定が、アクティベーションやレプリケーションの問題を引き起こしている可能性が疑われる場合
- どのオブジェクトがレプリケートされ、ルートがどのように設定されているかを確認したい場合

注意: 設定を編集するには、**edit config**コマンドを使用します。アクティブな設定を変更するには、**copy config**コマンドでアクティブな設定をコピーし、edit configコマンドでコピーを編集することをお勧めします。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**activate config**、**copy config**、**deactivate config**、**edit config**、**list config**、**remove config**、**rename config**、**verify config**

構文

基本コマンド	リモートオプション
view config *filename*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
filename	表示する設定ファイルの名前。設定名では大文字と小文字が区別されます。 **例**: `sp_ctrl(sysB)` **> view config Sales**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

view logオプション

show logコマンドのデフォルト値を表示するには、**view log** optionsコマンドを使用します。

表示例:

```
sp_ctrl(sysA)> view log options
Log current display options:
     direction = reverse
     maxlines = 50
     lpp = 30
```

これらのパラメーターは、**set log**コマンド、または**show log**コマンド自体のオプションを使用して変更できます。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	ビューアー(3)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**set log**、**show log**

構文

基本コマンド	リモートオプション
view logオプション	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**

オプション	説明
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

view partitions

view partitionsコマンドを使用して、水平分割レプリケーション設定内の1つのパーティションスキームまたはすべてのパーティションスキームの行パーティションを表示します。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	Oracle
サポート対象のターゲット:	すべて
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	add partition、drop partition、drop partition scheme、modify partition

構文

基本コマンド	リモートオプション
view partitions for {*scheme_name* | **all**}	利用不可

構文の説明

コンポーネント	説明
scheme_name	指定されたパーティションスキームの行パーティションを表示します。
all	すべての行パーティションを、パーティションスキームの名前に従ってグループ化して表示します。

例

```
sp_ctrl> view partitions for scheme1
sp_ctrl> view partitions all
```

2

SharePlexパラメータの設定

SharePlexパラメーターは、レプリケーションのさまざまな側面を制御し、調整します。

パラメーターの設定手順については、『SharePlex管理ガイド』を参照してください。

内容

ユーザ設定可能なパラメーターの説明

非推奨のパラメーター

ユーザ設定可能なパラメーターの説明

このセクションでは、**ユーザ設定可能な**パラメーターSharePlexについて説明します。この章に記載されていないパラメーターは内部パラメーターであり、Questの開発者またはテクニカルサポート担当者の指導の下でのみ変更してください。

パラメーターは以下のように分類されます。

- 設定アクティベーションパラメーター
- Captureパラメーター
- Readパラメーター
- Exportパラメーター
- Importパラメーター
- Oracle Postパラメーター
- Open Target Postパラメーター
- Queueパラメーター
- sp_copパラメーター
- Logパラメーター
- SNMPパラメーター
- Systemパラメーター
- Compare/Repairパラメーター
- copy/appendコマンドのパラメーター
- analyze configコマンドのパラメーター
- PostgreSQLパラメーター

設定アクティベーションパラメーター

これらのパラメーターは、SharePlexのアクティベーションプロセスで使用されます。これらは、設定ファイルをアクティベーションした時点で有効になります。

SP_OCF_HASH_BY_BLOCK

このパラメーターは、水平分割レプリケーションで使用されるハッシュアルゴリズムが行IDに基づくか、行が存在するブロックに基づくかを制御します。デフォルトは0で、行IDに基づくハッシュを使用します。水平分割レプリケーションを使用しているテーブルを処理する場合、ブロックに基づくハッシュを使用することで、Postプロセスのパフォーマンスが向上する可能性があります。ブロックに基づくハッシュを有効にするには、このパラメーターを1に設定し、設定ファイルを再度アクティベーションします。

デフォルト: 0（無効）

有効な値の範囲: 0 または1（ブロックに基づくハッシュを有効化）

有効になるタイミング: 次のアクティベーション時

SP_OCF_LOCK_WAIT_TIME

このパラメーターは、特定のテーブルのアクティベーションに失敗するまでの待ち時間をSharePlexに指定します。アクティベーションするにはテーブルをロックする必要があるため、指定された期間ロックを再試行するロジックになっており、この指定された期間はSP_OCF_LOCK_WAIT_TIMEパラメーターによって制御されます。

デフォルト: 5分

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_OCF_THREAD_COUNT

このパラメーターは、設定のアクティベーション中にSharePlexが生成するスレッド数のデフォルトを制御します。最適なパフォーマンスレベルを決定するために、このスレッド数はいくつか試す必要があるかもしれません。例えば、大容量のディスクアレイを持つ32 CPUマシンの場合、10スレッド以上でパフォーマンスが向上する可能性があります。スレッド数の値は分析対象のテーブル数には依存しません。

デフォルト: 3スレッド

有効な値の範囲: 1～32

有効になるタイミング: 即時

Captureパラメーター

これらのパラメーターは、SharePlexCaptureプロセスで使用されます。

SP_OCT_ALLOW_DP_DDL

このパラメーターは、Oracle Data Pumpのexport/importを実行する際に発生するDDL操作のレプリケーションにSharePlexが失敗した場合に有効にすることができます。場合によっては、SharePlexは、Data Pumpのロードでの DDLを無視すべき再帰的なDDLとして識別することがあります。このパラメーターは、そのDDLをキャプチャするようにSharePlexに指示します。

1を設定すると、このパラメーターが有効になります。ロード終了後、このパラメーターを0に戻し、Captureを再開します。

デフォルト: 0

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Captureプロセスの再開時

SP_OCT_ARCH_LOC

このパラメーターはアーカイブログへのパスを定義します。REDOログがラップした場合、SharePlexはOracleのアーカイブ・ログ・リストでアーカイブログを検索します。ここでアーカイブログを見つけられない場合、SharePlexはこのパラメーターによって指定されたディレクトリを検索します。サブディレクトリも検索します。デフォルトは**/home**ディレクトリです。アーカイブログのパスが異なる場合は、このパラメーターで正しいフルパス名を指定します。以下の例のようにセミコロン(;)で区切れば、複数のディレクトリパスを指定できます。この例では、Captureは**/disk1/log**と**/disk2/log**の両方のパスでログを検索します。

```
sp_ctrl> set param SP_OCT_ARCH_LOC /disk1/log;/disk2/log
```

SharePlex for RACのディレクトリパスを指定するには、以下を使用します。

```
sp_ctrl> set param SP_OCT_ARCH_LOC !;<node1_oracle_sid>;<node1_arch_absolutepath>;<node2_oracle_sid>;<node_arch_absolute_path>
```

パス名の非RAC形式では、セミコロンで区切られたディレクトリパス名のリストを使用できることに注意してください。リストの長さは、1,023バイトの制限まで任意の長さにできます。ただし、!;の後にsid;pathnameのペアのリストが続くRAC形式では、非RAC形式のようにパス名をパス名のリストにすることはできません。各SIDの後には、厳密に1つのディレクトリパス名を付けなければなりません。1つのインスタンスSIDに複数のパス名を指定する場合は、各エントリが同じSIDを持つ2つのエントリの対を入れなければなりません。したがって、以下のように指定することはできません。

```
!;sid1;path2;path2;path3;sid2;path4
```

この例の適切な形式は以下の通りです。

```
!;sid1;path2;sid1;path2;sid1;path3;sid2;path4
```

デフォルト: 空の文字列

有効な値の範囲: 復元されたアーカイブログへの完全なディレクトリパス

有効になるタイミング: 即時

SP_OCT_ASM_CACHE_AGE

このパラメーターは、ASMブロックの再クエリを実行してキャッシュを更新する前に、現在のASMキャッシュを保持する期間を制御します。

デフォルト: 40

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_OCT_ASM_ECACHE_SIZE

エクステントをキャッシュするバッファの要素数。エクステントは、Oracle ASMにおけるファイルストレージの単位です。この数値を大きくするとメモリ使用量が増えます。

デフォルト: 20

有効な値の範囲: 9を超える任意の値

有効になるタイミング: Captureプロセスの再開時

SP_OCT_ASM_FLIST_SIZE

同時に読み取まれるASMファイルの最大数。この数値を大きくするとメモリ使用量が増えます。

デフォルト: 10

有効な値の範囲: 4を超える任意の値

有効になるタイミング: Captureプロセスの再開時

SP_OCT_ASM_MULTI_OCI_BLOCK_SIZE

このパラメーターは、ExadataシステムでマルチスレッドキャプチャをSP_OCT_ASM_MULTI_OCIが設定されている場合に、各Captureスレッドで使用されるバッファのサイズを制御します。このパラメーターはデフォルトのままにしておく必要があります。Captureは、ログが置かれたディスクグループに設定されているAU_SIZEパラメーターの値に合わせてバッファサイズを自動的に調整します。これは、最高のパフォーマンスを発揮するために推奨されるバッファサイズです。

デフォルト: 0

有効な値の範囲: 任意の正の整数

有効になるタイミング: Captureプロセスの再開時

SP_OCT_ASM_MULTI_OCI

このパラメーターは、CaptureがREDOログの読み取りに使用するスレッド数を制御します。このパラメーターを使用するには、SP_OCT_OLOG_USE_OCIを1に設定する必要があります。

このパラメーターの値は、REDOログ・ディスク・グループ内のディスク数を超えない範囲で2以上に設定する必要があります。

多くのスレッドは必要ありません。実際に、スレッド数が多すぎるとパフォーマンスが低下します。スレッド数が多ければ多いほど、Captureに多くのメモリが必要になります。少ないスレッド数から始めてパフォーマンスを監視し、パフォーマンス向上とメモリ使用量の理想的なバランスが得られるまで、必要に応じてスレッドを追加します。

デフォルト: 0（シングル・スレッド・キャプチャ）

有効な値の範囲: 最小値は2、最大値はREDOログを格納するASMディスクグループのディスク数です。

有効になるタイミング: Captureプロセスの再開時

SP_OCT_ASM_SLIST_SIZE

同時にファイルにアクセスするセッションの最大数。

デフォルト: 10

有効な値の範囲: 4を超える任意の値

有効になるタイミング: Captureプロセスの再開時

SP_OCT_ASM_SUPPORT

このパラメーターは、ASMがサポートするプラットフォーム上でREDOログとアーカイブログのサポートを有効または無効にします。これは、SharePlexデータベース接続がASMに接続するように設定されている場合に自動的に有効になります。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Captureプロセスの再開時

SP_OCT_AUTOADD_ENABLE

このパラメーターは、新たに作成されたオブジェクトをアクティブなレプリケーション設定に自動的に追加するためのグローバルパラメーターです。デフォルトでは、テーブルとインデックスに適用されます。他のオブジェクトタイプを自動で追加するには、追加のパラメーター設定が必要です。

この機能は以下のように動作します。

SP_OCT_AUTOADD_ENABLEを有効にすると、以下のオブジェクトの自動追加がデフォルトで有効になります。

- CREATE TABLE操作で指定されたテーブル名は、アクティブな設定ファイルのワイルドカード指定と一致する場合に、自動的にレプリケーションに追加されます。
- CREATE INDEX操作で作成されたインデックスは、そのインデックスが作成されたテーブルが設定ファイルで指定されている場合に、自動的にレプリケーションに追加されます。
- CREATE TABLE AS SELECT操作で指定されたテーブル名は、アクティブな設定ファイルのワイルドカード指定と一致する場合に、自動的にレプリケーションに追加されます。SharePlexによりターゲット上に新しいテーブルが作成され、その後、ソーステーブルのデータが複製されます。
- 追加されたオブジェクトのALTERおよびDROP。

自動追加をサポートするには、SP_OCT_AUTOADD_ENABLEを有効にすることに加えて、以下のオブジェクトを明示的に有効にする必要があります。

CREATE MATERIALIZED VIEW操作で指定されたマテリアライズドビューのサポートテーブルは、マテリアライズドビューの名前がアクティブな設定ファイルのワイルドカードと一致し、SP_OCT_AUTOADD_MVIEWパラメーターが1に設定されている場合に、自動的にレプリケーションに追加されます。新しいマテリアライズドビューの自動追加をサポートするには、SP_SYS_TARGET_COMPATIBILITYパラメーターを8.6.2以上に設定する必要があります。

CREATE SEQUENCE操作で指定されたシーケンスは、アクティブな設定ファイルのワイルドカードと名前が一致し、SP_OCT_AUTOADD_SEQパラメーターが1に設定されている場合に、自動的にレプリケーションに追加されます。新しいシーケンスの自動追加をサポートするには、SP_SYS_TARGET_COMPATIBILITYパラメーターを8.6.3以上に設定する必要があります。

注意: シーケンスへの変更を複製するには、SP_OCT_REPLICATE_SEQUENCESパラメーターを1（デフォルト）に設定する必要があります。

パラメーターSP_OCT_REPLICATE_DDLが1または3に設定されていることを確認して、DDLレプリケーションが完全に有効になっていることを確認してください。

デフォルト: 1（オン）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

SP_OCT_AUTOADD_MVIEW

このパラメーターは、アクティベーションの後にマテリアライズドビューが作成された場合に、SharePlexがこれを自動的にレプリケーションに追加するかどうかを決定します。この機能を有効にすると、マテリアライズドビューの名前がアクティブな設定ファイルのワイルドカードを満たす場合、SharePlexは新しいマテリアライズドビューの基礎となるテーブルをレプリケーションに追加します。SharePlexはCREATE MATERIALIZED VIEWをCREATE TABLEステートメントに変換し、CREATE TABLEをターゲットにポストし、ビューに入力するDMLをレプリケートします。このテーブルは、将来サポートされるDDLとDMLの変更を通じて、レプリケーションによって維持されます。

マテリアライズドビューを自動的に追加するには、このパラメーターとSP_OCT_AUTOADD_ENABLEパラメーターの両方を1に設定し、ターゲット上のSharePlexのバージョンを8.6.2以降にする必要があります。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: プロセスの再開時

SP_OCT_AUTOADD_SEQ

このパラメーターは、アクティベーションの後にシーケンスが作成された場合に、SharePlexがこれを自動的にレプリケーションに追加するかどうかを決定します。有効にすると、シーケンスの名前がアクティブな設定ファイル内のワイルドカードを満たす場合、SharePlexはCREATEステートメントをレプリケートするようになります。次にSharePlexは、将来のDDLとDMLの変更を通じてターゲット上のオブジェクトを維持します。シーケンスを自動的に追加するには、このパラメーターとSP_OCT_AUTOADD_ENABLEパラメーターの両方を1に設定し、ターゲット上のSharePlexのバージョンを8.6.3以降にする必要があります。

重要! シーケンスをレプリケートするには、データベースレベルでプライマリキーとユニークキーのサプリメンタルロギングを有効にするか、**sys.seq$**テーブルでプライマリキーのサプリメンタルロギングを有効にする必要があります。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: プロセスの再開時

SP_OCT_CHECKPOINT_LOG

場合によっては、Captureプロセスが定期的にチェックポイントを実行しないことがあります。チェックポイントは、リカバリに必要な場合に備えてプロセスの状態を保存します。Captureプロセスが何らかの理由で終了し、REDOログがラップした場合、SharePlexは、存在していないチェックポイントへのリカバリを試みます。SP_OCT_CHECKPOINT_LOGは、ログが切り替わる前にチェックポイントを確実に実行するようにします。

チェックポイントは、Captureが指定されたログ数だけOracleから遅れたときにトリガされます。例えば、デフォルトの2の場合、CaptureはOracleからの遅れがログ2個分以上になったときにチェックポイントを実行します。

このパラメーターの許容値の範囲は、2（デフォルト）から、使用しているログの数に等しい値までです。0を指定すると、この機能は無効になります。このパラメーターは、頻繁なログの切り替えが原因で、SharePlexの内部チェックポイントメカニズムをトリガできるようになる前に切り替えが発生する環境で有効です。

デフォルト: ログ2個

有効な値の範囲: 2～REDOログの数

有効になるタイミング: 即時

SP_OCT_CHECKPOINT_TIME

このパラメーターはSP_OCT_CHECKPOINT_FREQパラメーターと連動します。Captureプロセスがチェックポイントを実行するまでの遅延時間（秒）を定義します。SP_OCT_CHECKPOINT_FREQに設定された値より前にこのパラメーターに設定された値に到達すると、チェックポイントがトリガされます（チェックポイントは、障害復旧に必要な場合に備えてプロセスの状態を保存する）。

デフォルト: 120秒

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_OCT_CK_LOC_FIRST

このパラメーターを使用すると、Captureプロセスは、SP_OCT_ARCH_LOCで指定された場所にあるアーカイブのREDOログを最初に検索します。SP_OCT_ARCH_LOCが空の文字列の場合、Captureは最初にOracleで設定されているアーカイブログの保存先を検索します。

デフォルト: 0(オフ)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: プロセスの再開時

SP_OCT_DATA_OBJ_CACHE_SIZE

このパラメーターは、データオブジェクトIDからテーブルオブジェクトIDへのマッピングキャッシュで使用されるエントリ/要素の数を指定します。DLOADまたはLOBデータなどの特定の操作では、そのような操作のREDOレコードにはデータオブジェクトIDのみが含まれます。Captureプロセスは、データオブジェクトIDをテーブルオブジェクトIDにマッピングし、その操作に注目する必要があるかどうかを確認する必要があります。データオブジェクトIDからテーブルオブジェクトIDへのキャッシュはルックアップソリューションを提供しますが、提供しない場合は、CaptureプロセスでOracleに問い合わせて、データオブジェクトIDをテーブルオブジェクトIDにマッピングする必要があります。

デフォルト: 50

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_OCT_DATE_ MSG

このパラメーターは、無効な日付列を検出したときに、Captureプロセスがイベントログとログに警告メッセージを表示するように設定できます。0に設定すると無効になり、1に設定すると有効になります。Captureにより生成されるエラーメッセージは以下の通りです。

`Invalid DATE format detected in record with rowid=rowid, on obj object_id. See capture log for detail.`(`obj object_id`の`rowid=rowid`のレコードで無効な`DATE`形式が検出されました。詳細については、`capture`ログを参照してください。)

デフォルト: 0(メッセージを印刷しない)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: 即時

SP_OCT_DDL_LOGGING

このパラメーターは、DDLアクティビティのCaptureログのレベルを制御します。DDLログへのDDLの記録はSP_OCT_REPLICATE_ALL_DDLの設定に依存しません。SP_OCT_REPLICATE_ALL_DDLが0であっても、DDLは記録されます。以下の設定が可能です。

0 = ログへの記録なし

1 = レプリケートされたDDLのみ

2 = すべてのDDL

デフォルト: 2

有効な値の範囲: 0、1、2

有効になるタイミング: 即時

SP_OCT_DDL_UPDATE_CONFIG

このパラメーターは、ALTER TABLE RENAMEの処理時に、SharePlexが設定ファイルのテーブル名を更新するかどうかを制御します。デフォルトでは、このDDL操作を処理するときにSharePlexはテーブル名を更新します。設定ファイルを更新しないようにするには、このパラメーターを無効にします。

デフォルト: 1（有効）

有効な値の範囲: 0または1

有効になるタイミング: Captureの再開時

SP_OCT_DEFパラメーター

以下のパラメーターを設定することで、ユーザやアプリケーションによって日付や時刻が誤って入力され、データベースの有効性チェックをバイパスした場合に、SharePlexがその書式を修正することができます。これらのパラメーターは、アクティベーションされるとすぐに有効になります。

SP_OCT_DEF_MONTH	値の範囲: 1～12
SP_OCT_DEF_DAY	値の範囲: 1～31
SP_OCT_DEF_YEAR	値の範囲: 1987～9999
SP_OCT_DEF_HOUR	値の範囲: 0～23
SP_OCT_DEF_MIN	値の範囲: 0～59
SP_OCT_DEF_SEC	値の範囲: 0～59

SP_OCT_DENIED_USERID

このパラメーターを使用すると、Captureプロセスによりすべての DMLおよびDDLトランザクションを無視またはフィルタリングするOracleユーザIDを指定できます。

重要: ソースマシン上のトランザクションを無視すると、非同期の状態になることがあります。

SharePlexは、指定されたユーザIDが存在するかどうかを検証しません。

デフォルト: 0x00000000

有効な値の範囲: 整数（任意の有効なOracleユーザID）

有効になるタイミング: 即時

SP_OCT_DLOAD_MAX_COLS_PER_MSG

このパラメーターはDLOADメッセージの最大サイズを制御します。メッセージの行数は、このパラメーターを各行の列数で割ることで決まります。例: テーブルの列数が99で90行のDLOADがあり、このパラメーターが5,000に設定されている場合は、メッセージは2つのDLOADメッセージに分割され、最初のメッセージには99列×50行（合計4950列）が、2つ目のメッセージには残りが含まれるようになります。

デフォルト: 10,000

有効な値の範囲: 99を超える任意の値

有効になるタイミング: 即時

SP_OCT_ENABLE_LOBMAP

このパラメーターは、行の外のLOB列を含むテーブルをレプリケートする際にSharePlexがLOBマップを使用するかどうかを制御します。LOBマップは、PK/UKロギングが有効でない場合に、Captureプロセスでの LOBIDと行のマッピングに使用されます。LOBマッピングはデフォルトで有効になっています。これらのマッピングはSHAREPLEX_LOBMAPテーブルに格納されます。Captureがマッピングを維持して参照する必要があるため、多数のLOB操作を伴うトランザクションではCaptureの処理速度が低下する可能性があります。データベースでPK/UKロギングが有効になっている場合、このパラメーターを0に設定することでLOBマッピングを無効にすることができます。

デフォルト: 1（オン）

有効な値の範囲: 0または1

有効になるタイミング: プロセスの再開時

SP_OCT_INCLUDE_UNCHANGED_COL

このパラメーターは、UPDATE操作の変更後イメージに未変更の列を含めるかどうかを制御します。デフォルトでは、SharePlexは変更された値のみを更新後イメージに含めます。

デフォルト: 0（オフ）

有効な値の範囲: 0または1

有効になるタイミング: 即時

SP_OCT_INSERT_INCLUDE_NULLS

このパラメーターは、NULL値を持つ列をINSERTステートメントでレプリケートするかどうかを制御します。デフォルトでは、SharePlexはNULL値をレプリケートしません。例えば、INSERT INTO mytable (col1, col2, col3) VALUEs ('red', 'green', 2)のようなステートメントで、mytableに6つの列がある場合、SharePlexは、明示的な'red'、'green'、および2の値のみをレプリケートし、他の3つの列の暗黙的なNULL値はレプリケートしません。SP_OCT_INSERT_INCLUDE_NULLSが1に設定されている場合、SharePlexは'red'、'green'、2、'、'、'をレプリケートします。

デフォルト: 0（オフ）

有効な値の範囲: 0または1

有効になるタイミング: 即時

SP_OCT_LOB_BUFFER_SIZE

このパラメーターはCaptureのLOBバッファのサイズを制御します。LOBバッファは、Captureが関連するトランザクション情報をアセンブルするまで維持しなければなりません。トランザクションを特定できない場合、SharePlexは、トランザクションを含むキュー以外の特別なサブキューを使用します。これらのサブキューのヘッダはより多くの共有メモリを必要とし、SP_OCT_LOB_BUFFER_SIZEパラメーターの値を大きくしなければならないことがあります。

LOBバッファが大きければ大きいほど、システムメモリが不足する可能性が高くなり、メモリが再び利用可能になるまで、後続のLOB操作のバッファリングが妨げられます。

したがって、SP_OCT_LOB_BUFFER_SIZEパラメーターの値を大きくするのと同時に、SP_QUE_Q_SHMSIZEパラメーターの値も大きくしなければならない可能性があります。一般にVARRAYのLOBデータのみがバッファリングされるため、通常、このパラメーターはVARRAYを含むトランザクションにのみ影響を与えます。

デフォルト: LOBあたり5 MBのメモリ

有効な値の範囲: 任意の正の整数

有効になるタイミング: Captureの再開時

SP_OCT_LOG_FILESIZE

このパラメーターは、Captureデバッグ・ログ・ファイルのサイズを設定します。

デフォルト: 50,000,000バイト

有効な値の範囲: 9,999バイトを超える任意の値

有効になるタイミング: Captureの再開時

SP_OCT_LOG_MEMBER

SP_OCT_LOG_MEMBERパラメーターは、データ辞書のクエリを強化して、REDOログの場所を指定するために使用します。このパラメーターを使用して、操作するログファイルを正確に指定することができます。

デフォルト: %（パーセント記号）

有効な値の範囲: ワイルドカード文字を含む、正しく形成された任意のファイル名指定

有効になるタイミング: Captureの再開時

SP_OCT_LOG_NUMFILES

このパラメーターは、最も古いCaptureデバッグ・ログ・ファイルが削除され、新しいCaptureデバッグ・ログ・ファイルが作成されるまでに許可されるCaptureデバッグ・ログ・ファイルの数を制御します。

デフォルト: 3

有効な値の範囲: 1を超える任意の値

有効になるタイミング: Captureの再開時

SP_OCT_LOG_READ_SIZE

Captureプロセスは、複数のREDOログ・ファイル・ブロックを一度に読み取ります。このパラメーターは、一度に読み取むブロック数を制御するもので、システムの設定や、CaptureがOracleの処理ペースに遅れをとっているかどうかに応じて値を調整することができます。必要以上に高い値を設定すると、システムのオーバーヘッドが増えることに留意してください。

デフォルト: 64ブロック

有効な値の範囲: 任意の正の整数

有効になるタイミング: Captureの再開時

SP_OCT_LOGWRAP_RESTART

このパラメーターは、以下の状況に対するCaptureの動作を制御します。

- アーカイブを利用できない。
- Captureがログを開く許可を拒否された。

Captureがアーカイブログを見つけることができない場合、またはログを開くことができない場合、プロセスはデフォルト（設定0）で停止します。この設定では、Captureがログにアクセスできない場合、**show capture**コマンドは「`Stopped due to missing archive log`（アーカイブログがないため停止しました）」というステータスを返します。

一定時間（秒）待機してから自動的に再開するようにCaptureを設定できます。これにより、正しい読み取り権限を割り当てたり、SharePlexが期待する場所にアーカイブを移動する外部ログ管理処理を実行したりする時間ができます。Captureは待機してログをチェックし、まだ利用できなければ、または開けなければ停止し、ログがリストアされるか開かれるまでチェックと停止を続けます。この設定では、**show capture**コマンドは「`waiting for logfile`（ログファイルを定期しています）」というステータスを返します。

デフォルト: 0秒（ログがない場合は自動的に再開しない）

有効な値の範囲: 1～任意の正の秒数

有効になるタイミング: 即時

SP_OCT_MIN_SESSIONS

このパラメーターは、Captureが同時実行トランザクションからのデータを格納するために作成および維持するサブキューの最小数を制御します。このパラメーターは、Postがそれ自体の同時性を高められるようにすることでPost Enhanced Performance（PEP）機能をサポートします。PEP機能はSP_OPO_DEPENDENCY_CHECKパラメーターで制御されます。

最良の結果を得るには、SP_OCT_MIN_SESSIONSをターゲットシステム上のCPUコア数以上に設定します。

デフォルト: 8

有効な値の範囲: 任意の正の整数

有効になるタイミング: Captureの再開時

SP_OCT_OLOG_DELAY

このパラメーターは、読み取む必要のあるデータがそれ以上ないことをCaptureのREDOログリーダーが検出したときにスリープする時間（マイクロ秒）を制御します。このパラメーターは、基本的にCaptureがアイドル状態のときにのみ有効になります。Captureの処理量が多い場合は、読み取む必要があるレコードがあるため、基本的にスリープ状態にはなりません。

デフォルト: 10,000（マイクロ秒）

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_OCT_OLOG_NO_DATA_DELAY

このパラメーターは、処理するデータがないときに、ログリーダーが再試行前にスリープする時間の長さを制御します。

デフォルト: 10,000（マイクロ秒）

有効な値の範囲: 任意の正の整数

有効になるタイミング: Readの再開時

SP_OCT_OLOG_QUEUE_SIZE

このパラメーターは、ログリーダーのキューのサイズを設定します。

デフォルト: 16,384

有効な値の範囲: 100～任意の正の整数

有効になるタイミング: プロセスの再開時

SP_OCT_OLOG_RDS_MINER

注意: SP_OCT_OLOG_RDS_MINERパラメーターの有効化はOracle 19c以降は推奨されなくなり、サポートされなくなりました。

このパラメーターは、Oracle RDSデータベースからキャプチャするときに、CaptureがOracleに遅れないようにするために2番目のスレッドを使用するかどうかを制御します。

このスレッドを使用すると処理の負荷がかかるため、デフォルトでは無効になっています。有効にするには、このパラメーターを1に設定します。

デフォルト: 0(オフ)

有効な値の範囲: 0または1(有効)

有効になるタイミング: プロセスの再開時

SP_OCT_OLOG_REOPEN

このパラメーターは、Captureプロセスがファイルの最後に達したときに、オンラインREDOログを閉じて再度開くかどうかを制御します。このパラメーターが有効な場合、SP_OCT_OLOG_NO_DATA_DELAYパラメーターを使用することで、Captureプロセスが再試行するまでの待機時間を制御することができます。

デフォルト: 1(オフ)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: プロセスの再開時

SP_OCT_LOG_USE_OCI

このパラメーターは、Captureがソース変更レコードをキャプチャする際に、REDOログをファイルから直接読み取むか、OCIコールを使用するかを制御します。デフォルトの0は、REDOログから直接読み取むようにCaptureに指示します。

- ソースデータベースがASM上で実行されている場合、このパラメーターを有効にすることはオプションです。Captureは、REDOログを直接読み取むことも、OCIコールを使用することもできます。
- ソースデータベースが、Captureが実行されているマシンから離れている場合(クラウドサービスなど)、CaptureがOCIの呼び出しを使用してソース変更レコードを取得するように、このパラメーターを1に設定する必要があります。

デフォルト: 0(オフ、CaptureはREDOログから直接読み取む)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: プロセスの再開時

SP_OCT_ONELINE_DEBUG

このパラメーターは、Captureプロセスのオンラインデバッグ機能を制御します。

オンラインデバッグ機能により、システムリソースを大量に消費することなく、非同期エラーのデバッグを実行できます。オンラインデバッグでは、データストリームで非同期状態が発生した場所を検出するのに十分な情報を1行で記録します。問題を引き起こしているプロセスがオンラインデバッグで特定されたら、そのプロセスについて通常のデバッグを有効にすることができます。

このパラメーターを有効にするには、デバッグするオブジェクトをオブジェクトIDで指定し、それぞれをカンマで区切ります。例:

```
sp_ctrl> set sp_oct_oneline_debug 230230, 351626
```

以下の項目が記録されます。

- SharePlexが使用するトランザクション識別子
- 影響を受けたDMLまたはDDL操作のタイプ
- 影響を受けるソーステーブルのオブジェクトID
- 影響を受ける行の行識別子(Oracle行ID)
- 操作が行われた時刻(Oracle)
- 影響を受ける操作のログシーケンス番号とログ内のオフセット
- トランザクションのOracle SCN
- 冗長な操作を防ぐ内部SharePlexコード

デフォルト: 無効

有効な値の範囲: オブジェクトIDによりリストされるオブジェクトのリストを形成する文字列(カンマで区切る)

有効になるタイミング: 即時

SP_OCT_OPS_LOGREADER_DELAY

このパラメーターは、no_dataが検出されたときに再試行するまでのログリーダーの遅延時間を制御します。

デフォルト: 50,000(マイクロ秒)

有効な値の範囲: 任意の正の整数

有効になるタイミング: Captureの再開時

SP_OCT_OPS_LOGREADER_RETRY

このパラメーターは、no_dataが検出されたときにログリーダーが行う再試行の回数を制御します。

デフォルト: 3

有効な値の範囲: 任意の正の整数

有効になるタイミング: Captureの再開時

SP_OCT_OPS_NO_DATA_DELAY

このパラメーターは、シーケンサーにデータがないときにノードごとに強制されるスリープの長さを制御します。

デフォルト: 5（デシ秒）

有効な値の範囲: 任意の正の整数

有効になるタイミング: Readの再開時

SP_OCT_PREFER_ARCHIVE

このパラメーターは、CaptureがREDOログのオンラインバージョンを読み取むか、アーカイブバージョン（ある場合）を読み取むかを制御します。デフォルトでは、Captureはオンラインログのみから読み取り、オンラインログが利用できなくなった場合に限りアーカイブバージョンを読み取ります。

このパラメーターが設定され、Captureが必要なログをオンラインで検出した場合、Captureはログもアーカイブされているかどうかのチェックを追加で行います。ログがアーカイブされている場合、Captureはアーカイブパスを照会し、オンラインのログの代わりにアーカイブされたログを読み取ります。ログがアーカイブされていない場合、Captureはオンラインログを読み取ります。

通常SharePlexは、オンラインREDOログから読み取むときのレイテンシを最小限にしますが、このパラメーターは、ExadataでのCaptureのパフォーマンス向上に役立つことがあります。SharePlexは、Exadata ASMファイルシステムの外部にある多重化されたアーカイブの場所から読み取むと、より大量のデータを処理することができます。Exadata上の大量のデータでバックログが発生した場合は、SP_OCT_PREFER_ARCHIVEパラメーターを1に設定し、SharePlex管理ガイドの「ExadataでのCaptureの調整」のセクションの指示に従ってSharePlexを設定することができます。

このパラメーターを使用するには、REDOログの多重化を有効にし、アーカイブログへのパスをSharePlexに設定しなければなりません。詳細については、「SP_OCT_ARCH_LOC」の334ページを参照してください。

デフォルト: 0（無効）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Captureの再開時

SP_OCT_READ_ARC

このパラメーターは、SharePlexがOracleアーカイブログからのみ読み取むかどうかを制御します。デフォルト設定の0は、OracleのREDOログを読み取むために、SharePlexがオンラインログとアーカイブログの両方を利用することを意味します。ゼロ以外の値は、SharePlexCaptureプロセスがアーカイブされたOracle REDOログからのみ読み取むことを意味します。SharePlexは、Oracle REDOログのそのシーケンスを読み取もうとする前に、オンラインログがアーカイブされるまで待ちます。

0 = SharePlexのデフォルトの動作、オンラインとアーカイブの両方のREDOログを読み取む

1 = アーカイブされたREDOログの読み取りのみアーカイブされたログの検索は、Oracleで定義されたアーカイブの場所から始まり、その後、SP_OCT_ARCH_LOCディレクトリの指定に従って行われます。

2 = アーカイブされたREDOログの読み取りのみ検索は、SP_OCT_ARCH_LOCディレクトリの指定のみに従って行われます*（Oracle定義のアーカイブの場所では検索されない）。*SP_OCT_CK_LOC_FIRST値は、この設定では無視されます。1と仮定します。

3 = アーカイブされたREDOログの読み取りのみOracleで定義されたアーカイブの場所からの読み取りのみ。SP_OCT_ARCH_LOCディレクトリは無視されます。SP_OCT_CK_LOC_FIRST値は、この設定では無視されます。0と仮定します。

ベストプラクティス: SP_OCT_READ_ARCを有効にする場合、少なくともオンラインログからアーカイブログを生成し、コピーするまでの自然な遅延がデータレイテンシに現れます。このレイテンシは許容できる程度のものですが、リーダーのキーの取得性能は、一貫性のあるビューの読み取りに依存しています。リーダーによるキーの取得は、一貫性のあるビューが最新（またはそれに近い）のものかどうかに依存しますが、このパラメーターを使用すると、少なくともレイテンシの時間だけビューの経過時間を遅らせることができます。リーダーがキーを取得する必要がないように、前述のcaptureパラメーターを使用するお客様には、レプリケーション中のテーブルのプライマリキーと一意キー（PU/UK）に対するソースデータベースのサプリメンタルロギングを有効にすることを強く推奨します。

デフォルト: 0

有効な値の範囲: 0～3

有効になるタイミング: Captureの再開時

SP_OCT_REDOLOG_ENSURE

このパラメーターは、オンラインREDOログからの変更レコードを処理する前にCaptureが待機する時間（秒）を制御します。この待ち時間は、レコードのOracleタイムスタンプと現在のシステム時間との差です。これは、Oracleログライターが書き込みを終了していないレコードをCaptureが読み取もうとするのを防ぎます。この結果、Captureは完全なブロックを確実に処理できるようになります。

リモートデータベースからキャプチャする場合、Captureはこのパラメーターを自動的に無効にします。

デフォルト: 2秒

有効な値の範囲: 0～任意の正の整数

有効になるタイミング: Captureの再開時

SP_OCT_REDUCED_KEY

このパラメーターは、REDOログ内の操作のどの部分をCaptureによってターゲットに送信し、Postで使用するかを制御します。SharePlexの機能によって、Postプロセスで使用可能にする必要のあるデータの量が異なることがあります。

- 0に設定すると、このパラメーターは、OracleがREDOレコードに書き込むLONGとLOB以外のすべてのデータをCaptureに送信するように指示します。この設定によりネットワーク経由で送信されるデータが最大になりますが、SharePlexの特定の機能をサポートするためには必要です。
- 1（デフォルト）に設定すると、このパラメーターは、キー値と、UPDATEの場合は操作で変更された列の変更前後の値を送信するようにCaptureに指示します。この設定はデフォルトです。この設定により、PostはWHERE句を作成する際に、変更された列の変更前後のcompareができるため、レプリケーションのパフォーマンスとターゲットデータ整合性のバランスが適切になります。SharePlexによる前後の値の使用方法の詳細については、『SharePlex管理ガイド』を参照してください。
- 2に設定すると、このパラメーターは、キー値と、UPDATEの場合は、ターゲットに対して変更された列の変更後の値を送信するようにCaptureに指示します。この設定により、ネットワーク経由で送信されるデータ量が最小限になり、さらにPostがデータをポストする際にキー値のみを使用できるようになるため、パフォーマンスが向上します。

「SP_OPO_REDUCED_KEY」も参照してください。

デフォルト: 1

有効な値の範囲: 0、1、2

有効になるタイミング: 即時

SP_OCT_REPLICATE_ALL_DDL

このパラメーターは拡張DDLレプリケーションを制御します。これは、SharePlexのレプリケーション設定ファイルにないオブジェクトのDDLのレプリケーションです。設定操作の前に、オブジェクトはソースデータベースと、(CREATEを除き)ターゲットにもなければなりません。SharePlexはDDLステートメントをレプリケートしますが、ターゲット上のオブジェクトは今後のDMLの変更を通じて維持されません。

SP_OCT_REPLICATE_ALL_DDLを設定する場合、レプリケートされたDDLをターゲットシステム上で正常に実行するために、ソースデータベースとターゲットデータベースが同一でなければなりません。このDDLはすべてのターゲットマシンにブロードキャストされるため、すべてのターゲットシステムは同一でなければなりません。

SharePlexは、アクティブなデータソースのすべてのスキーマのすべてのオブジェクトに対してサポートされているDDLをレプリケートするため、操作を成功させるためには、それらのコンポーネントがターゲットデータベースに存在していなければなりません。

レプリケートされたDDL操作をターゲットデータベースに適用するときのエラー発生時にPostプロセスを停止するように設定するには、SP_OPO_STOP_ON_DDL_ERRパラメーターを設定します。

パッケージやストアドプロシージャの作成をレプリケートする場合、パッケージやストアドプロシージャの本体にあるオブジェクトの名前を完全修飾する必要があります。これは、他のユーザやスキーマがこのパッケージやストアドプロシージャを実行する可能性があるからです。

SharePlexでサポートされる操作のリストについては、使用中のSharePlexのバージョンに対応するリリースノートを参照してください。

> **注:** 拡張DDLのレプリケーションは、テーブルやシーケンスだけでなく、プロシージャ、関数、ユーザ、ビューなど、レプリケーションには含まれないその他のオブジェクトも幅広くサポートしています。これらのオブジェクトの中には、*レプリケーション対象*の基礎となるオブジェクトが含まれることもあります。このような場合、拡張DDLレプリケーションは、レプリケーション設定に含まれないオブジェクトだけでなく、レプリケーション対象の基礎となるオブジェクトにも適用されます。

SharePlexはOracle Flashback Table機能をサポートしていません。SP_REPLICATE_ALL_DDLパラメーターが有効(値は1)の場合、SharePlexはフラッシュバックDDLを複製しようとすることがありますが、エラーが返されます。レプリケーション中のテーブルでFlashback Tableを実行するには、『SharePlex管理者ガイド』の以下の手順を使用して、この問題を回避してください。

1. レプリケーションからのソースオブジェクトの削除
2. フラッシュバックの実行
3. アクティブな設定のオブジェクトの追加または変更

デフォルト: 0(オフ)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: 即時

SP_OCT_REPLICATE_COMMENT

このパラメーターは、設定ファイルにリストされているテーブルに対してSharePlexが以下のDDL操作をレプリケートするかどうかを制御します。

- ALTER TABLE to ADD COMMENT
- COMMENT ON

デフォルトでは、先行するDDL操作がレプリケートされます。これらの操作のレプリケーションを無効にするには、このパラメーターを0に設定します。これは、これらの操作のレプリケーションがパフォーマンスに影響を与えている場合に有効です。

デフォルト: 1（有効）

有効な値の範囲: 0または1

有効になるタイミング: Captureの再開時

SP_OCT_REPLICATE_CTAS

このパラメーターは、新しいテーブルの名前がSharePlexの設定ファイルのワイルドカード指定と一致する場合に、SharePlexがCREATE TABLE AS SELECT (CTAS)ステートメントをレプリケートする方法を決定します。ターゲットにテーブルが存在するかどうかに応じて、CTASステートメントがSharePlexによってレプリケートされる方法を指定するオプションがあります。

パラメーター値	使用例	説明
1（デフォルト）	SELECT FROMテーブルがターゲットに存在する	この設定では、元のCTASステートメントのみが複製され、データが含まれるSELECT FROMテーブルがターゲット上になければなりません。 例えば、以下のCTASステートメントはそのままターゲットにレプリケートされます。 **create table my_table as select * from table_a;** この例では、SELECTが成功し、新しいテーブル**my_table**にデータが入力されるように**table_a**がターゲット上に存在する必要があります。
2	SELECT FROMテーブルがターゲット上に存在しない	この設定によりSharePlexは以下を実行します。 1. CREATE TABLEステートメントを複製して、ターゲットに新しいテーブルを作成します。 2. ソースのSELECTステートメントによって返されたDMLをすべてINSERTSとして複製し、新しいターゲットテーブルに入力します。 例えば、ソースステートメントが以下の場合、 **create table my_table as select * from table_a;** **table_a**が2つの列と2つの行を持つと仮定した場合、SharePlexがターゲットにポストするステートメントは以下のようになります。 **Create table my_table (c1 number, c2 varchar2(20));**

パラメーター値	使用例	説明
		Insert into my_table values (1,'a'); **Insert into my_table values (2,'b');** **Commit;** この結果はCREATE TABLE AS SELECTと同じになりますが、SELECT FROMソーステーブルがターゲット上に存在している必要はありません。 **重要!** 複製されたDMLデータは、Captureによって割り当てられたメモリに収まらなければなりません。データ量が非常に多い場合、Captureは失敗する可能性があります。

デフォルト: 1

有効な値の範囲: 1または2

有効になるタイミング: プロセスの再開時

SP_OCT_REPLICATE_DDL

このパラメーターは、SharePlexによってレプリケートされているアクティブな設定内のオブジェクトに対して、SharePlexが以下の操作をレプリケートするかどうかを制御します。

- ALTER TABLE to ADD COLUMN、MODIFY COLUMN、DROP COLUMN
- ALTER TABLE to ADD、MODIFY、DROP、SPLIT、COALESCE、MOVE、TRUNCATE、EXCHANGE PARTITION/SUBPARTITION
- TRUNCATE TABLE

両方の操作タイプをレプリケートするか、どちらか一方だけレプリケートするか、あるいはレプリケートしないかを制御できます。

注意: Oracle 21CをソースとするSharePlex 11.1は、現時点ではDML操作のみをサポートしています。DDL操作のサポートは将来のバージョンで追加される予定です。

その他のデフォルトのDDLレプリケーション機能については、SP_OCT_AUTOADD_ENABLEも参照してください。

設定ファイル外のオブジェクトのDDLをレプリケートするには、SP_OCT_REPLICATE_ALL_DDLパラメーターを参照してください。

デフォルト: 3（ALTER TABLEとTRUNCATEのレプリケート）

有効な値の範囲:

0（ALTER TABLEとTRUNCATEの両方のレプリケーションを無効化）

1（ALTERレプリケーションのみを有効化）

2（TRUNCATEレプリケーションのみを有効化）

3（ALTERとTRUNCATEのレプリケーションを有効化）

有効になるタイミング: 即時

SP_OCT_REPLICATE_DLOAD

このパラメーターで、SQL*Loaderのダイレクトパスロードをレプリケートするかどうかを制御します。デフォルト設定の1では、ダイレクトパスロードのレプリケーションが有効になります。SharePlexは、非並列ロードのレプリケーションのみをサポートしています（PARALLEL=FALSE）。データベースをアーカイブモードにし、テーブルロギングを有効にしなければなりません。ダイレクトパスロードのレプリケーションを無効にするには、このパラメーターを0に変更します。

デフォルト: 1（ダイレクトパスロードをレプリケート）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

SP_OCT_REPLICATE_GRANT

このパラメーターは、設定ファイルにリストされているテーブルに対して発行されたGRANTコマンドをSharePlexがレプリケートするかどうかを制御します。

デフォルト: 0（無効）

有効な値の範囲: 0または1

有効になるタイミング: Captureの再開時

SP_OCT_REPLICATE_MVIEW

このパラメーターは、アクティベーション中に見つかった既存のマテリアライズドビューをレプリケーションに入れるかどうかを制御します。デフォルトでは有効になっています。マテリアライズドビューは、アクティベーションする前にソースとターゲットに存在していなければなりません。

このパラメーターは、アクティベーション時にのみ使用され、マテリアライズドビューをレプリケーションに入れるかどうかを決定します。アクティベーションの後にマテリアライズドビューが作成されたときに、レプリケーションにマテリアライズド ビューを追加するようにSharePlexを設定するには、「SP_OCT_AUTOADD_MVIEW」を参照してください。

デフォルト: 1（マテリアライズドビューをレプリケート）

有効な値の範囲: 0または1

有効になるタイミング: プロセスの再開時

SP_OCT_REPLICATE_POSTER

このパラメーターは、あるシステムのCaptureプロセスが、そのシステムのPostプロセスによってポストされたデータをレプリケートするかどうかを制御します。このパラメーターをデフォルト設定の0のままにしておくと、Captureは同じシステム上のPostのアクティビティを無視します。特定のレプリケーション設定（主にカスケードレプリケーション）を確立する場合、このパラメーターを1に設定するように指示されることがあります。これにより、Captureはポストされた変更をレプリケートします。

デフォルト: 0（Postトランザクションをレプリケートしない）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Captureの再開時

SP_OCT_REPLICATE_SEQUENCES

このパラメーターは、設定ファイルにリストされているシーケンスに加えられた変更を、SharePlexが明示的に、またはワイルドカードによってレプリケートするかどうかを制御します。デフォルトでは有効になっています。シーケンスは、アクティベーションする前にソースとターゲットに存在していなければなりません。

アクティベーション後に作成されたシーケンスをレプリケーションに追加するようにSharePlexを設定できます。詳細については、「SP_OCT_AUTOADD_SEQ」の338ページを参照してください。

0に設定すると、たとえシーケンスが設定ファイルにリストされていても、またはシーケンス名がワイルドカード指定を満たしていても、レプリケーションから除外されます。

重要! シーケンスをレプリケートするには、データベースレベルでプライマリキーとユニークキーのサプリメンタルロギングを有効にするか、**sys.seq$**テーブルでプライマリキーのサプリメンタルロギングを有効にする必要があります。

デフォルト: 1（シーケンスをレプリケート）

有効な値の範囲: 0または1

有効になるタイミング: プロセスの再開時

SP_OCT_REPLICATE_SYNONYM

このパラメーターは、設定ファイルにリストされているテーブルに対して発行されたCREATE SYNONYMおよびDROP SYNONYMコマンドをSharePlexがレプリケートするかどうかを制御します。

デフォルト: 0（無効）

有効な値の範囲: 0または1

有効になるタイミング: Captureの再開時

SP_OCT_REPLICATE_TRIGGER

このパラメーターは、設定ファイルにリストされているテーブルに対して発行されたCREATETRIGGERおよびDROP TRIGGERコマンドをSharePlexがレプリケートするかどうかを制御します。

注意: アクティブ-アクティブのレプリケーションシナリオではトリガをレプリケートしないでください。詳細については、『SharePlex管理ガイド』の「*複数のピアデータベースを維持するためのレプリケーションの設定*」のセクションを参照してください。

デフォルト: 0（無効）

有効な値の範囲: 0または1

有効になるタイミング: Captureの再開時

SP_OCT_REQUIRED_DATA_IS_LOGGED

このパラメーターは、ロールバック処理を改善するためにCaptureプロセスが追加情報を収集することを可能にします。ロールバックの処理により、Readプロセスはより多くのリソースを必要とし、それによりパフォーマンスが向上します。システムリソースが不足し、システムでロールバックが多発する場合は、Readプロセスのリソース消費を抑えるために、このパラメーターを無効にすることができます。このパラメーターは、デフォルトでは有効になっています。

デフォルト: 1（オン）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Captureの再開時

SP_OCT_TRUNC_PARTITION_BY_ID

システムにより生成されたパーティションをトランケートまたは削除するALTER TABLEのレプリケーションが、ターゲット上の正しいパーティションに影響を与えるようにするには、このパラメーターを使用します。

システムにより生成されたインターバルパーティションの名前はデータベースが生成するため、ソース上のこれらのパーティションの名前はターゲット上の対応するパーティションの名前と一致しません。しかし、ターゲットがソースの完全なコピーであれば、同じ時間枠内のパーティション位置は一致します。

レプリケーションがターゲット上の正しいパーティションに影響を与えるようにするには:

1. ソースとターゲットのテーブル構造とパーティションの定義が同一であることを確認してください。同じ大きな値を持つパーティションについては、レプリケーション開始前に、**dba_tab_partitions**からのパーティション位置をソースとターゲットで同じにする必要があります。
2. SP_OCT_TRUNC_PARTITION_BY_IDパラメーターを1に設定します。この設定により、SharePlexは元のALTER TABLEコマンドで指定されたパーティション名ではなく、パーティション位置を使用してパーティションを識別するようになります。Postは、ターゲット側のテーブルの正しいパーティション名にパーティション位置をマッピングします。

このパラメーターを有効にすると、SharePlexはSP_SYS_TARGET_COMPATIBILITYパラメーターをチェックし、ターゲットのPostプロセスのバージョンが8.6.4以降であることを確認します。これは、システムにより生成されるパーティションをIDによってトランケートまたは削除するALTER TABLEをサポートするための最小バージョンです。ターゲットのPostプロセスが以前のバージョンの場合、ALTER TABLEはパーティション名を使用して複製され、ターゲットのSharePlex **event_log**に警告メッセージが記録されます。

デフォルト: 0（無効）

有効な値の範囲: 0または1

有効になるタイミング: 即時

SP_OCT_USE_DST

このパラメーターを使用すると、REDOログでサマータイム補正を無効にし、ロジックを確保するようにSharePlexに指示します。

デフォルト: 0（無効）

有効な値の範囲: 0または1

有効になるタイミング: プロセスの再開時

SP_OCT_USE_SUPP_KEYS

このパラメーターを使用すると、行が更新または削除されたときに、Oracleのサプリメンタルロギングによって設定された列をキー列として使用するようにSharePlexに指示します。

典型的なレプリケーションスキームでは、Postが行を更新または削除する際に使用するキー列のセットをSharePlexが選択します。OracleのPK/UKサプリメンタルロギングが有効な場合、Oracleは更新ごとにキー列をログに記録します。これらのキー列は、SharePlexがキーとして選んだものと必ずしも一致するとは限りません。一致しないと、Readプロセスで不要な作業が行われることになります。このパラメーターを設定すると、SharePlexのデフォルトの動作がオーバーライドされ、SharePlexの速度と効率が向上します。

設定ファイルでテーブルが水平パーティショニングで設定されている場合、その列条件が既にそのテーブルのPK/UKの一部になっていない限り、REDOロググループに水平パーティショニング列条件を含める必要があります。

注意： SP_OCT_USE_SUPP_KEYSパラメーターは、PK/UIでサプリメンタルロギングが有効になっている場合にのみ有効になります。

このパラメーターはSP_ORD_HP_IN_SYNCパラメーターとは互換性がありません。両方のパラメーターが有効な場合、SP_OCT_USE_SUPP_KEYSがSP_ORD_HP_IN_SYNCよりも優先されます。

デフォルト: 0（無効）

有効な値の範囲: 0または1（有効）

有効になるタイミング: 即時

Readパラメーター

これらのパラメーターは、SharePlexReadプロセスで使用されます。

SP_ORD_BATCH_ENABLE

このパラメーターは、バッチ処理機能の有効化を制御します。デフォルトでは有効になっています。これにより、Readプロセスは複数の同一レコードを1つのレコード、つまりバッチに結合して、プロセス内で処理できるようになります。

デフォルト: 1（オン）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

SP_ORD_BATCH_MAX

このパラメーターは、同時バッチトランザクションの最大数を制御します。

デフォルト: 50（トランザクション）

有効な値の範囲: 5から250までの任意の整数

有効になるタイミング: Readの再開時

SP_ORD_BATCH_MATCH_MIN

このパラメーターは、バッチを処理する前に、一致した操作の最小数を制御します。

デフォルト: 2

有効な値の範囲: 任意の正の整数

有効になるタイミング: Readの再開時

SP_ORD_CDA_LIMIT

このパラメーターは、Readプロセスの各ログインによってキャッシュされるカーソル数を制御します。レプリケーションがソースシステムのOracleアクティビティから遅れ始めたら、この値を増やす必要があるかもしれません。レプリケーションに多数のテーブルがある場合は、初期設定を15カーソルにすることをお勧めします。

デフォルト: 5カーソル

有効な値の範囲: 任意の正の整数

有効になるタイミング: Readの再開時

SP_ORD_DATE_ MSG

このパラメーターは、無効な日付列を検出したときに、Readプロセスがイベントログとログに警告メッセージを表示するように設定できます。0を設定するとパラメーターは無効になり、1を設定すると有効になります。Readにより生成されるエラーメッセージは以下の通りです。

```
An oerr#1801 has occurred on record with rowid rowid, on object object_id. Rec
skipped. It is usually caused by invalid column data of type DATE. Creating a unique
index that doesn't include column of type DATE and reactivating same configuration may
solve the problem.
```

デフォルト: 0(警告メッセージを表示しない)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: 即時

SP_ORD_DELAY_RECORDS

Readプロセスはレコードをバッチで処理します。バッチのサイズは、captureキュー内のレコードの数に依存します。レコード数が多い場合、Readはこのパラメーターの値を無視し、バッチは可能な限り小さく保たれます。ReadがCaptureと同じペースで進行している場合、バッチのサイズはほぼこのパラメーターで設定された値になります。

SharePlexがイベントログに`rollback segment too old`(ロールバックセグメントが古すぎる)というメッセージを生成している場合、このパラメーターの値を小さくします。これにより、SharePlexにデータを早く渡すように指示します。このパラメーターは、最小限のレイテンシしか許容できない場合に使用します。

> **重要:** このパラメーターは慎重に使用してください。レコード数を減らすとSharePlexが実行する必要があるI/Oが増加し、システムのオーバーヘッドが増加するからです。また、SharePlexプロセス間のスムーズな相互作用にも悪影響を及ぼす可能性があります。

デフォルト: 200レコード

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_ORD_FIRST_FIND

このパラメーターは、レプリケートされた行の変更が列条件を満たすかどうかを判断するために、Readプロセスが列条件をチェックする方法を制御します。

- デフォルトの1では、行の変更がある列条件を満たす場合、SharePlexはその行の変更が他の列条件も満たすかどうかをチェックしません。
- 値が0の場合、SharePlexは列条件を満たすすべてのターゲットシステムにデータを送信します。

水平パーティショニングの詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 1(オン)

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Readの再開時

SP_ORD_HP_HASH

このパラメーターは、水平パーティショニングに使用されるハッシュテーブルのスロット数を制御します。デフォルトでは、メモリ使用量を最小限に抑えるため、このパラメーターは16スロットに設定されています（使用量は、挿入を伴うトランザクションごとにスロット数×32バイトとなります）。ユーザのシステムが、水平にパーティショニングされたテーブルに対して（同じトランザクション内で）多くの挿入操作とそれに続く更新操作を行う場合、SharePlexはこのハッシュテーブルを多用することになります。その場合、パフォーマンス低下を防ぐためにこの値を大きくする必要があるかもしれません。また、ユーザのシステムで水平にパーティショニングされたテーブルへの挿入を伴う長いトランザクションが多い場合も、このパラメーターの値を大きくする必要があります。

デフォルト: 16スロット

有効な値の範囲: 任意の正の整数

有効になるタイミング: Readの再開時

SP_ORD_HP_IN_SYNC

このパラメーターは、水平分割レプリケーションで、列条件の列の値が変更され、行が条件を満たさなくなった場合に、データが適切にレプリケートされるようにするために使用します。

これにより、SharePlexは以下を自動的に修正できます。

- UPDATEにより、ある行が、その行のために作成された列条件とは異なる列条件を満たすようになり、変更が別の場所に送信される。例えば、列条件**region=East**である行に対するUPDATEにより、**region**列の値がWESTに変更されるとします。この操作は失敗します。なぜなら、その行に対する元のINSERTステートメントは元の場所（東部地域）にレプリケートされたため、Postが更新を試みたときには、その行が新しい場所（西部地域）に存在しないからです。
- UPDATEにより、レプリケートされるはずのない行が列条件を満たし、レプリケートされる。例えば、region列がHEADQUARTERSの値（行がレプリケートされない）からWESTの値に更新されたとします。この操作は失敗します。なぜなら、その行の元のINSERTステートメント（本社システムへの）が西部地域のシステムにレプリケートされないため、Postはそこで更新を実行できないからです。
- UPDATEにより、行がどの列条件も満たさなくなる。例えば、**region**列がWESTの値からHEADQUARTERSの値に更新されたとします。元のINSERTステートメントは西部地域のシステムにレプリケートされますが、新しい値が列条件を満たしていない（本社データは共有されていない）ため、新しい値への更新はレプリケートされません。行は非同期になりますが、エラーは発生しません。

このパラメーターが有効な場合、SharePlexはUPDATEによって上記の状態が発生する行を自動的に修正します。SharePlexはUPDATEをDELETEに変換し、必要であればINSERTに変換します。

UPDATEステートメント（通常は変更された列とキーのみを使用）をINSERTステートメントに変換するには、SharePlexはすべての列の値が必要です。SP_ORD_HP_IN_SYNCを有効にすると、SharePlexは水平分割レプリケーションを使用しているテーブルに対するUPDATEがあった場合、行のすべての列をPostプロセスに送信して、INSERTを構築できるようになります。

設定をアクティベーションする前に、**ソース**システムでこのパラメーターを設定してください。レプリケーションがアクティブな場合は、パラメーターを設定してから設定を再アクティベーションして、SharePlexがオブジェクトキャッシュを再構築できるようにします。

水平分割レプリケーションを使用しているテーブルの列条件の列が変更されないことが分かっている場合は、このパラメーターを使用すると処理オーバーヘッドが発生するため、0に設定したままにします。

このパラメーターは、SP_OCT_REDUCED_KEYおよびSP_OPO_REDUCED_KEY [任意の値: 1または2]と互換性がありません。このパラメーターは両方のパラメーターの動作をオーバーライドするからです。

デフォルト: 0(オフ)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: Readの再開時

SP_ORD_LDA_ARRAY_SIZE

このパラメーターは、読み取りの一貫性を保つためにデータベースに対するログインの回数を制御します。Readプロセスが遅くなる場合は、このパラメーターの値を増やしてみてください。最大設定は、init_oraファイルのMAX_PROCESSESパラメーターによって決定されます。

デフォルト: 5ログイン

有効な値の範囲: 任意の正の整数

有効になるタイミング: Readの再開時

SP_ORD_ONELINE_DEBUG

このパラメーターは、Readプロセスのオンラインデバッグ機能を制御します。

オンラインデバッグ機能により、システムリソースを大量に消費することなく、非同期エラーのデバッグを実行できます。オンラインデバッグでは、データストリームで非同期状態が発生した場所を検出するのに十分な情報を1行で記録します。問題を引き起こしているプロセスがオンラインデバッグで特定されたら、そのプロセスについて通常のデバッグを有効にすることができます。

このパラメーターを有効にするには、デバッグするオブジェクトをオブジェクトIDで指定し、それぞれをカンマで区切ります。例:

```
sp_ctrl> set sp_ord_oneline_debug 230230, 351626
```

以下の項目が記録されます。

- 操作の状態。**Dispatch**(キューからの読み込み)、**processMessage**(SQLコードの処理)、または**ExecSQL**(ターゲットへの適用)のいずれか
- SharePlexが使用するトランザクション識別子
- 影響を受けたDMLまたはDDL操作のタイプ
- ターゲットテーブルの所有者および名前
- 影響を受けるソーステーブルのオブジェクトID
- 影響を受ける行の行識別子(Oracle行ID)

- 操作が行われた時刻（Oracle）
- 影響を受ける操作のログシーケンス番号とログ内のオフセット
- トランザクションのOracle SCN
- 冗長な操作を防ぐ内部SharePlexコード
- ルーティング情報

注意: 行が複数のターゲットにルーティングされる場合、この情報はターゲットごとに繰り返されます。

デフォルト: 無効

有効な値の範囲: オブジェクトIDによりリストされるオブジェクトのリストを形成する文字列（カンマで区切る）

有効になるタイミング: 即時

SP_ORD_MSGS_CK_FREQおよびSP_ORD_RCM_SKIP_RATIO

これらのパラメーターは連動して動作するため、一方のパラメーターをデフォルト値の0に設定した場合、もう一方のパラメーターを0以外の数値に設定しても動作しません。どちらのパラメーターも、実行されたクエリあたりのディスク取得率をチェックすることで、クエリ処理に時間がかかりすぎていることをReadプロセスが検知する自動プロセスをサポートします。Readがこの判断を行うと、既存の読み取り一貫性のあるビューを取り除き、新しいビューに置き換えます。

ORD_MSGS_CK_FREQパラメーターは、比率が基準を超えたかどうかをReadプロセスがチェックする頻度です。例えば、このパラメーターを100に設定すると、Readプロセスは100クエリーごとに、比率に問題がないかどうかをチェックします。推奨設定は1,000クエリーです。

ORD_RCM_SKIP_RATIOパラメーターは、読み取り一貫性ビューが置き換えられる基準値（実行されたクエリに対するディスクの取得率）を指定します。最も細かな設定は1で、これは1回のクエリ実行につき1回のディスク取得を意味します。推奨設定は2です。

これらのパラメーターをオフにするには、どちらかを0に設定します。これによりもう一方のパラメーターも無効になります。

デフォルト:

SP_ORD_MSGS_CK_FREQ: 10,000

SP_ORD_RCM_SKIP_RATIO: 2

有効な値の範囲:

SP_ORD_MSGS_CK_FREQ: 0～100,000

SP_ORD_RCM_SKIP_RATIO: 0～1,000（5より大きくすることは非推奨）

有効になるタイミング: プロセスの再開時

SP_ORD_RMSG_LIMIT

このパラメーターは、Readプロセスがcaptureキューからメッセージを読み込み、キーの値を決定する際に実行するチェックポイントの頻度を制御します。チェックポイントは、captureキューをキャッシュファイルに保存し、（exportキューに渡される）

送信キューのメッセージをコミットし、既にexportキューによって受信されたcaptureキューの残留レコードに対して読み取りリリース(削除)を行います。

このパラメーターの値が高いほど、チェックポイントがトリガされる前に、より多くのレコードがメモリに保持されます。非常に大きい値に設定すると、システム上のI/Oが少なくなり、処理が速くなりますが、その代償としてメモリ使用量が増え、予期せぬことが原因でReadプロセスが停止した場合の回復時間が長くなります。値を小さくするとI/Oが増加し、リカバリ速度が向上しますが、スループット速度は低下します。

デフォルト値の100,000レコードは、スピードの必要性とメモリの節約およびプロセスリカバリの間の合理的なバランスを確立するはずですが、処理要件に合わせてこのパラメーターを調整できます。調整オプションの範囲は、すべてのレコードごとにチェックポイントを実行するものから、システムとメモリが収容できる限り多くのレコードを保持するものまであります。

デフォルト: 100,000レコード

有効な値の範囲: システムの制限内の任意の正の整数

有効になるタイミング: 即時

SP_ORD_ROLLBACK_TXNS_MAX

このパラメーターは、リーダーがトランザクションキャッシュに保持する、トランザクションが完全にロールバックされたことを示すエントリの数を制限します。

デフォルト: 5,000レコード

有効な値の範囲: システムの制限内の任意の正の整数

有効になるタイミング: プロセスの再開時

SP_ORD_SEND_DDL_TO_FIRST

このパラメーターは、複数のpostキューが使用中で、レプリケーション設定にないオブジェクトのDDLをレプリケートしている場合(SP_OCT_REPLICATE_ALL_DDL=1)に、Postのパフォーマンスを向上させるのに役立ちます。

このパラメーターは、レプリケーション設定内にないオブジェクトのDDLを特定のpostキューに送信するようにImportに指示します。通常、レプリケーション設定内のオブジェクトに関連しないDDLは、利用可能な任意のキューに送信できます。このDDLが大量にあると、レプリケーション中のオブジェクトのDMLがブロックされ、データのレイテンシが長くなる可能性があります。

このDDLは、レプリケーション中のオブジェクトと関連性がないため、これらのオブジェクトに関連する順序で適用する必要はありません。これを専用のキューで個別にルーティングすることで、他のキューを解放して、レプリケートされたデータとそれに関連するDDLを処理することができます。

このパラメーターを有効にする場合は、レプリケートされていないオブジェクトのDDLを処理するpostキューを指定する必要があります。この指定は、設定ファイルの「Datasource:o.SID」行の後の最初の行に、以下のように記述する必要があります(!は、実際のオブジェクト指定に置き換えるプレースホルダです)。

```
Datasource: o.dbprod
#レプリケーション対象外のDDLのルート
! sysmm:queddl1@o.dbprep
#残りの設定エントリ
```

```
SCOTT.EMP SCOTT.EMP sysmm:que1@o.dbrep

SCOTT.FOO SCOTT.FOO sysmm:que2@o.dbrep
```

このように明示的に指定しない場合、DDLは最初にリストされているルートにルーティングされます。例えば、以下の設定ファイルでは、レプリケーション中のオブジェクトに関連付けられないDDLはすべてsysmm:que1@o.dbrepに送信されます。

```
Datasource: o.dbprod

SCOTT.EMP SCOTT.EMP sysmm:que1@o.dbrep

SCOTT.FOO SCOTT.FOO sysmm:que2@o.dbrep
```

デフォルト: 1（オン）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

Exportパラメーター

これらのパラメーターは、SharePlexのExportプロセスで使用されます。

SP_XPT_AES_KEY_LENGTH

このパラメーターは、ソースからターゲットへネットワークを介して送信されるデータを暗号化するためにExportが使用するAES暗号化キーのサイズを制御します。キー自体は、256ビットの完全なキーを生成する**create encryption key**コマンドを発行することで生成されます。SP_XPT_AES_KEY_LENGTHパラメーターは、Exportがデータを暗号化する際に、その長さのどの分量をキーとして使用するかを制御します。デフォルトは128ビットで、これは最小の長さです。キーは長いほどハッキングしにくくなりますが、より高いCPUの処理能力が必要になります。AES暗号化のためにSharePlexを設定するには、『SharePlex管理ガイド』を参照してください。

デフォルト: 128ビット

有効な値の範囲: 128、192、または256ビット

有効になるタイミング: Exportの再開時

SP_XPT_ENABLE_AES

このパラメーターは、Advanced Encryption Standard（AES）を有効にして、Exportがネットワーク経由で送信するレプリケートデータを暗号化するかどうかを制御します。AES暗号化のためにSharePlexを設定するには、『SharePlex管理ガイド』を参照してください。

デフォルト: 0（無効）

有効な値の範囲: 0または1（有効）

有効になるタイミング: Exportの再開時

SP_XPT_ENABLE_COMPRESSION

このパラメーターは、TCP/IPを介したデータの圧縮を制御します。

圧縮を有効にすると、SharePlexがネットワーク経由で送信するデータ量を減らすことができます。SharePlexはLZIPロスレス圧縮を使用します。ソースのSharePlexインスタンスで圧縮を有効にすると、ソースのSharePlexインスタンスのすべてのターゲットへの圧縮が自動的に有効になります。

デフォルトでは圧縮は無効になっています。圧縮は単独で、または暗号化と組み合わせて有効にすることができます。暗号化の詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 0（無効）

有効な値の範囲: 0または1（有効）

有効になるタイミング: Exportの再開時

SP_XPT_KEEPALIVE

このパラメーターは、レプリケーションアクティビティが低いときにTCPタイムアウトを防ぐために、Exportプロセスが一定間隔で「hello」メッセージをImportに送信するかどうかを制御します。レプリケーションパケットが送信されずにネットワークがタイムアウトした場合、SharePlexは、「Export cannot connect to import on server2: timeout waiting for ack（Exportをserver2のImportに接続できません: ACKの待機中にタイムアウトが発生しました）」などの警告メッセージを表示します。SP_XPT_KEEPALIVEを1に設定すると、このメッセージを削除し、SharePlexのネットワーク接続を維持することができます。

デフォルト: 0（無効）

有効な値の範囲: 0または1（有効）

有効になるタイミング: Exportの再開時

SP_XPT_PORT_OVERRIDE

このパラメーターはSP_XPT_USE_LOCALHOSTパラメーターと連動し、SharePlexがSSH® Secure Shell™ ソフトウェアで作成された安全なトンネルポートを介してデータを送信できるようにします。このパラメーターはローカル接続を設定し、デフォルトのSharePlexポートをオーバーライドします。そのポート番号から、SSHデーモンは別のポート番号とSSHデータ暗号化を使用して、別のシステム（SharePlexターゲットマシン）に接続を指示します。

デフォルト: 0（無効）

有効な値の範囲: 0または任意の正の整数

有効になるタイミング: Exportの再開時

SP_XPT_SO_SNDBUF

このパラメーターは、ソースマシンのTCP/IPウィンドウサイズを調整します。このパラメーターは、ターゲットマシンのSP_COP_SO_RCVBUFパラメーターと連動し、ネットワークを介して送信されるパケットのサイズを確立します。SharePlexは、sp_copとExportプロセスの開始時、TCP/IPソケットが作成されるときに両方のパラメーターを参照します。

SharePlexがWANを通じてレプリケートされ、exportキューが常に滞留している場合は、両方のパラメーターを調整してみてください。SP_XPT_SO_SNDBUFは、1,024バイトの倍数でSP_COP_SO_RCVBUFの値以上に設定しなければなりません。パラメーターのサイズを設定するには、ソースマシンとターゲットマシンの間のping時間を決定し、両方のパラメーターに以下の式を使用します。

param_value / ping_time **= 1秒あたりのバイト数**

例えば、ping時間が200ミリ秒、2つのパラメーターの値が64Kの場合、SharePlexは、毎秒5個の64Kパケットを送信し、合計で毎秒320Kを送信することになります。

帯域幅の問題が確認されない限り、Questでは、両方のパラメーターをデフォルト（システムの設定を使用する）のままにしておくことを推奨します。SP_XPT_SO_SNDBUFを変更するには、これをソースシステム上で設定し、そのシステム上でExportを停止してから開始します。

それでも転送が遅い場合は、ターゲットシステムのSP_IMP_WCMT_MSGCNTパラメーターとSP_IMP_WCMT_TIMEOUTパラメーターを大きくしてください。SP_IMP_WCMT_MSGCNTを少なくとも10,000に設定します。

デフォルト: 0（デフォルトはオペレーティングシステムによって設定される）

有効な値の範囲: 1,024の倍数を使用したバイト単位の正の整数。最大値はオペレーティングシステムによって設定されます。

有効になるタイミング: Exportの再開時

SP_XPT_USE_LOCALHOST

このパラメーターによってSharePlexは、SSH® Secure Shell TMソフトウェアで作成された安全なトンネルポートを介してデータを送信できるようになります。Exportプロセスは、TCP接続を行う前にこのパラメーターを読み取ります。このパラメーターが有効な場合、Exportプロセスはローカルポート番号を通じてローカルホストに接続し、SSHデーモンは別のポート番号とSSHデータ暗号化を使用して、別のシステム（SharePlexターゲットマシン）に接続を指示します。

デフォルト: 0（無効）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Exportの再開時

Importパラメーター

これらのパラメーターは、SharePlexのImportプロセスで使用されます。

SP_IMP_ENABLE_AES

このパラメーターは、ソースシステムと、それが設定されているターゲットとの間でデータ暗号化の使用を強制します。これは、Exportプロセスでデータ暗号化が有効になっていない限り、Importで受信データを受け入れないようにするものです。SP_XPT_ENABLE_AESが有効になっている場合は、有効化しなければなりません。

AES暗号化のためにSharePlexを設定するには、『SharePlex管理ガイド』を参照してください。

デフォルト: 0（無効）

有効な値の範囲: 0または1（有効）

有効になるタイミング: Importの再開時

SP_IMP_QUEUE_PAUSE

このパラメーターは、postキューに指定された数のメッセージが含まれる場合、そのキューへのデータの書き込みを一時停止します。Postは、チェックポイントを発行するまでキューメッセージを共有メモリに保存し、その後メモリからデータを解放します。

postキューが共有メモリを使い果たした場合、読み書き関数はメモリバッファを解放するためにファイルIOを開始します。キューの書き込みを一時停止することでディスクストレージが不要になり、その結果生じるIOの減速を回避できるため、このパラメーターはPostのパフォーマンス維持に役立ちます。

Importがpostキューへの書き込みを要求され、そのキューのバックログがSP_IMP_QUEUE_PAUSE以上である場合、Importは関連するExportとチェックポイントを実行し、postキューへの書き込みを停止します。ただし、Importは実行されたままになります。警告がイベントログに書き込まれ、**show statusdb**コマンド出力のプロセスのステータスが「paused（一時停止中）」と表示されます。**status**コマンドでは、「Running（実行中）」ではなく「Paused（一時停止中）」と表示されます。

Importは、一旦停止して再開するか、バックログがSP_IMP_QUEUE_RESUMEパラメーター以下であることを検出した場合に、postキューへの書き込みを再開します。

SP_IMP_QUEUE_RESUMEパラメーターを使用して、Importがpostキューへの書き込みを再開するメッセージの数を設定します。

この機能を使用するには、SP_IMP_QUEUE_PAUSEとSP_IMP_QUEUE_RESUMEの両方がゼロより大きく、SP_IMP_QUEUE_PAUSEがSP_IMP_QUEUE_RESUMEよりも大きくなければなりません。

デフォルト: 0個のメッセージ（無効）

有効な値の範囲: *n*千個のメッセージ（*n*は任意の正の整数）

有効になるタイミング: 即時

SP_IMP_QUEUE_RESUME

このパラメーターはSP_IMP_QUEUE_PAUSEと連動して機能します。postキュー内のメッセージ数がこのパラメーターで設定された値以下であれば、Importはpostキューへの書き込みを再開します。

この機能を使用するには、SP_IMP_QUEUE_PAUSEとSP_IMP_QUEUE_RESUMEの両方がゼロより大きく、SP_IMP_QUEUE_PAUSEがSP_IMP_QUEUE_RESUMEよりも大きくなければなりません。

デフォルト: 0個のメッセージ（無効）

有効な値の範囲: *n*千個のメッセージ（*n*は任意の正の整数）

有効になるタイミング: 即時

SP_IMP_WCMT_MSGCNT

このパラメーターはSP_IMP_WCMT_TIMEOUTパラメーターと連動します。Importプロセスがチェックポイントを実行するまでに処理されるメッセージの数を定義します。チェックポイントは、障害復旧に必要な場合に備えてプロセスの状態を保存します。

Importがチェックポイントを実行すると、それがトリガとなってExportプロセスがチェックポイントを実行します。SP_IMP_WCMT_TIMEOUTに設定された値より前にこのパラメーターの値に達すると、チェックポイントがトリガされます。

WAN環境では、10,000個のメッセージまでこのパラメーターを大きくすることができます。ただし、チェックポイント間のメッセージ間隔を長くすると、SharePlexのフォールトトレランスに悪影響を及ぼす可能性があることを理解しておいてください。ネットワークに継続的に障害が発生する環境でSharePlexを使用している場合は、このパラメーターを小さくすることができます。

デフォルト: 10,000メッセージ

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_IMP_WCMT_TIMEOUT

このパラメーターはSP_IMP_WCMT_MSGCNTパラメーターと連動します。Importプロセスがチェックポイントを通過するまでの時間（秒）を定義します。チェックポイントは、障害復旧に必要な場合に備えてプロセスの状態を保存します。

Importがチェックポイントを実行すると、それがトリガとなってExportプロセスがチェックポイントを実行します。SP_IMP_WCMT_MSGCNTに設定された値より前にこのパラメーターの値に達した場合、チェックポイントがトリガされます。

デフォルト: 30秒

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

Oracle Postパラメーター

これらのパラメーターは、Oracleターゲットにデータを適用する際にSharePlexのPostプロセスで使用されます。

SP_OPO_CHANGE_ID_START_VALUE

このパラメーターは**changeid**メタデータ列の開始値を設定します。この列は変更履歴ターゲットに含めることができ、**target**コマンドで設定します。changeidは、レコードを一意に識別し、重複を防ぐためにSharePlexによって生成されます。

デフォルト: 0（オフ）

有効な値の範囲: 0～9,223,372,036,854,775,807

有効になるタイミング: Postの再開時

SP_OPO_COMMIT_REDUCE_MSGS

Post Enhanced Performance機能のコミット削減コンポーネントにしきい値を設定します。コミット削減機能により、小規模なトランザクションのバッチを大規模なトランザクションにまとめるようにPostを設定することができます。これにより、処理しなければならないコミットや確認の数を減らすことができます。トランザクションが小さければ小さいほど、パフォーマンスは向上します。

指定されたメッセージ数に達すると、Postはコミットを発行します。トランザクションの境界がこのメッセージの範囲内にあるトランザクションのコミットはスキップされ、それらのトランザクションはすべて1つのトランザクションとしてコミットされます。コミット削減はデフォルトでオンになっています。無効にするには、このパラメーターの値を1に設定します。

このパラメーターの設定は絶対的なしきい値ではありません。SharePlexは、結合された異なるトランザクションにトランザクションを分割することはありません。したがって、すべての操作とグループ内の最後のトランザクションのコミットを含めるために、Postがそのしきい値を超えなければならない可能性があります。

デフォルト: 100個のメッセージ

有効な値の範囲: 1または任意の正の整数

有効になるタイミング: Postの再開時

SP_OPO_CONNECTION_POOL

このパラメーターは、Postが接続プールをどのように使用するかを制御します。

接続プールが有効（デフォルト）で、サブキューがOracleへの接続を必要とする場合、サブキューはコミットされているサブキューを探そうとします。もし見つかれば、Oracleへの新しい接続を開くのではなく、その接続を使用します。これにより、PostはOracleへの接続を減らして操作することができます。

接続プールを使用しない場合、各サブキューはOracleへの個別の接続を持ち、サブキューがコミットされるまで、その接続を（SP_OPO_IDLE_LOGOUTパラメーターによって決定される）時間（秒）保持します。

このパラメーターを使用するには:

- 接続プールを有効にすると、SharePlexが多数のサブキューを使用して実行されている場合に便利です。
- SharePlexが、すぐに再利用される少数のサブキュー（20個未満）で実行されている場合、プールを無効にして接続を移動しない方が効率的な場合があります。

デフォルト: 1（有効）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Postの再開時

SP_OPO_CONT_ON_ERR

このパラメーターは、修正可能なエラーが発生したときにPostを停止するかどうかを制御します。このフラグをデフォルトの0に設定すると、Postは、すべてのOracleおよびSharePlexのエラーで停止します。SharePlexまたはOracleの特定のエラーが発生してもPostにポストを続けさせるには、変数ディレクトリの**data**サブディレクトリにある**oramsglist**ファイルにそれらのエラーをリストし、このパラメーターを1に設定します。このパラメーターを2に設定すると、Postはテーブルエラーを他のエラーと同じように扱い、そのエラーが**oramsglist**ファイルにリストされていない限りポストを停止します。

パラメーターの設定にかかわらず、以下のエラーが発生してもPostはポストを継続します。

Postが停止しないデフォルトのエラー

- ユニークキー違反
- 操作中断
- データなし
- そのようなテーブルがない
- 無効な数字
- 数値以外の日付
- 無効な行ID
- 無効な16進数
- not-nullをnullに更新できない
- 年は-4713か～+9999でなければならない
- チェック制約違反
- パケットライターの失敗*
- シーケンスが見つからない
- 内部エラー

*パケットライターの失敗とnowaitでのリソースビジーは、SP_OPO_RETRIES_MAXパラメーター（デフォルトは10）に基づいて再試行され、その後にPostが終了します。

エラー発生時に続行するようにPostを設定する方法の詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 0（すべてのエラーで停止）

有効な値の範囲: 0、1、2（フラグ）

有効になるタイミング: 即時

SP_OPO_DEPENDENCY_CHECK

このパラメーターは以下の機能を制御します。

- Post Enhanced Performance機能のTransaction Concurrencyコンポーネント。Postのパフォーマンスを調整する方法の詳細については、『SharePlex管理ガイド』を参照してください。
- ターゲットテーブルの有効なON DELETE CASCADE制約のPost処理（レプリケートされた親削除を適用し、レプリケートされたカスケード削除を無視）。

これらの機能をサポートするには、以下のようにします。

1. **SP_OPO_DEPENDENCY_CHECKを適切な値に設定します。**
    - コミット削減と同時トランザクションを有効にするには、このパラメーターを1に設定します。この機能を使用するには、プライマリキーとユニークキーのサプリメンタルロギングをソースで有効にする必要があります。
    - ON DELETE CASCADEを有効にするには、このパラメーターを2に設定します。

注意: ON DELETE CASCADE機能は、設定2で有効になるPost Enhanced Performance機能に依存しています。SP_OPO_DEPENDENCY_CHECKの設定に加えて、ソース上でプライマリキー、一意インデックス列、外部キー列のロギングを有効にする必要があります。

2. SP_OCT_REDUCED_KEYパラメーターを0に設定し、Postが使用するために必要なデータをすべてターゲットに送信します。「SP_OCT_REDUCED_KEY 」を参照してください。
3. SP_OPO_REDUCED_KEYパラメーターを0、1（デフォルト）または2に設定します。「SP_OPO_REDUCED_KEY」を参照してください。

デフォルト: 0（オフ）

有効な値の範囲: 0、1、2（フラグ）

有効になるタイミング: Postの再開時

SP_OPO_DEPENDENCY_SIZE

このパラメーターは、Post Enhanced Performance機能の依存性チェックで使用されるメモリのサイズを設定します。デフォルト値で十分ですが、必要であれば大きくすることもできます。

デフォルト: 7019

有効な値の範囲: 任意の有効なメモリ値

有効になるタイミング: Postの再開時

SP_OPO_DISABLE_OBJECT_NUM

このパラメーターは、**ソース**テーブルのオブジェクトIDに基づいて、レプリケートされたDMLおよびDDL操作をPostがターゲットにポストしないようにします。ソーステーブルのデータが無効であるか破損している場合、ソーステーブルにターゲットでサポートされていないデータ型が含まれている場合、またはテーブルの操作をターゲットデータベースに反映させたくないその他の理由がある場合に、このパラメーターを設定できます。

このパラメーターは1つまたは複数のテーブルに設定できます。これにより、データを再同期して設定ファイルを再度アクティベーションする時間ができるまで、ソーステーブルのターゲットテーブルでそれ以上レプリケーションが実行されないようにします。Postプロセスは、これらのテーブルの複製されたすべてのメッセージをpostキューから破棄するため、メッセージはキューに蓄積されません。

このパラメーターは注意して使用してください。これを有効にしてソーステーブルに対してDDLまたはDMLを実行すると、変更がポストされないため、ターゲットデータが古くなります。レプリケーション設定外の他のテーブルの外部キーなど、テーブルに依存関係がある場合は、ポストを無効にすると依存関係が満たされなくなります。

このパラメーターは、デフォルトでは無効になっています。この機能を有効にするには、ターゲットシステムで以下のコマンドを実行します。ここに表示される数字は、ポストから除外するソーステーブルのオブジェクトIDです。

```
sp_ctrl(sysB)> set param SP_OPO_DISABLE_OBJECT_NUM 12345,67890
```

各オブジェクトIDをカンマで区切り、その間にはスペースを入れません。エントリ間にスペースを使用するには、以下の例のようにリスト全体を引用符で囲みます。

```
sp_ctrl(sysB)> set param SP_OPO_DISABLE_OBJECT_NUM "498438, 1000, 497109"
```

無効なオブジェクトIDは無視されます。

ターゲットテーブルへのポストを再開する準備ができたら、SP_OPO_DISABLE_OBJECT_NUMを0に設定します。

デフォルト: 0(オフ)

有効な値の範囲: カンマで区切られた有効なOracleオブジェクトIDのリスト

有効になるタイミング: 即時

SP_OPO_HINTS_LIMIT

このパラメーターは、ヒントファイルに記載できるヒント(テーブルとインデックスの組み合わせ)の最大数を制御します。ヒントは、定義されたインデックスがあるテーブルに対してPostがフルテーブルスキャンを行っている場合にのみ、使用してください。大量のヒントを使用すると、Postプロセスのパフォーマンスが低下する可能性があります。ヒント機能の詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 100ヒント

有効な値の範囲: 任意の正の整数

有効になるタイミング: Postの再開時

SP_OPO_LOG_CONFLICT

このパラメーターを使用すると、成功したコンフリクト解決プロシージャに関する情報を**SHAREPLEX_CONF_LOG**テーブルに記録できるようになります。これは、SharePlexが提供する準備されたルーチンにのみ適用されます。

- 1に設定すると、**SHAREPLEX_CONF_LOG**テーブルへのコンフリクト解決のロギングが有効になります。

> **注意:** 1に設定すると、SHAREPLEX_CONF_LOGテーブルのEXISTING_TIMESTAMPおよびTARGET_ROWID列(既存のデータが置換されない場合)は更新されません。

- 2に設定すると、**SHAREPLEX_CONF_LOG**テーブルへのコンフリクト解決のロギングが、Postによる追加のメタデータのクエリと共に有効になります。

  準備されたルーチンLeastRecentRecordまたはMostRecentRecordを使用すると、Postは既存のレコードのタイムスタンプ列のターゲットデータベースにクエリを実行します。クエリの結果は、SHAREPLEX_CONF_LOGテーブルのEXISTING_TIMESTAMP列にログ記録されます。

  準備されたルーチンでは、入力されたレコードで置換されなかった行に対し、Postは置換される可能性のある既存の行のTARGET_ROWIDを問い合わせます。そうでない場合、既存の行のROWIDは記録されません。

  注: 2と設定すると、クエリを実行した結果、Postのパフォーマンスに影響を与える可能性があります。

デフォルト: 0(無効)

有効な値の範囲: 0、1、または2

有効になるタイミング: Postの再開時

SP_OPO_MAX_CDA

このパラメーターはSQLキャッシュモジュール用であり、セッションが同時に開くことができるカーソルキャッシュの最大数を制御します。この数値は、v$parameterビューのOPEN_CURSORSに対するOracle設定より小さくする必要があります。SP_OPO_SQL_CACHE_DISABLEが0(有効)に設定されている場合にのみ使用されます。

デフォルト: 50

有効な値の範囲: 任意の正の整数

有効になるタイミング: Postの再開時

SP_OPO_MAX_OEXN_TIME

このパラメーターは、SQLスレッドがOCIStmtExecute ( )の呼び出しに費やすことができる最大時間を秒単位で制御します。この時間を経過すると、タイムキーパー スレッドがデッドロック状態になり、マルチスレッドPostプロセスは強制終了されます。

デフォルト: 900 [秒]

有効な値の範囲: 任意の正の整数

有効になるタイミング: Postの再開時

SP_OPO_MAX_ROLLBACK_MESSAGES

このパラメーターは、部分的ロールバックで後方カウントを増分する際の最大メッセージ数を指定します。メインスレッドは、前方にスキャンし、後方メッセージをカウントします。後方カウントがこのパラメーターに設定された最大数に達すると、メインスレッドはロールバックを解決します。ロールバックが解決されると、メインスレッドはメッセージのディスパッチを再開します。まだ後方にメッセージがある場合、メインスレッドは再びサブキューを前方にスキャンします。次の10,000の後方メッセージのグループ(パラメーターが10,000に設定されている場合)も、同じプロシージャを使用して解決されます。

デフォルト: 10,000メッセージ

有効な値の範囲: 11～1,000,000

有効になるタイミング: Postの再開時

SP_OPO_NLS_CONVERSION

このパラメーターは、OracleソースとOracle ターゲット間の文字セット変換を制御します。

SharePlexが、使用しているOracleの文字セット内のすべての文字を複製するには、以下のいずれかが真でなければなりません。

- ソースとターゲットの文字セットが同一である
- ソースデータベースの文字セットが、ターゲットデータベースの文字セットのサブセットである(ソースに含まれるすべての文字がターゲットの文字セットに存在する)

SharePlexでは、以下の文字セットがテストされ、サポートされています。

US7ASCII

UTF8

WE8ISO8859P1

AL16UTF16

AL32UTF8

KO16KSC5601

デフォルトでは、SharePlexにより、Oracleターゲット データベースは文字変換を行うことができます。Postはソースデータの文字エンコーディングをOracleに通知し、Oracleは必要な変換を実行します。

関係する文字セットによっては、Oracleの変換はデータロスにつながる可能性があります。次などを考慮します。

例1: JA16SJIS文字セットの「米」という日本語文字は、US7ASCII文字セットには対応する記号がありません。この記号をUS7ASCIIデータベースに複製しようとすると、Oracleはこの記号を「?」文字に変換します。

例2: Oracleによると、WE8ISO8859P1文字セットはUS7ASCII文字セットのスーパーセットであるため、US7ASCIIの文字はすべて、変換されずにWE8ISO8859P1ターゲット データベースにポストされると考えるのが論理的です。これは、0x00から0x7Fの範囲の文字に当てはまります。しかし、Oracleは、0x80から0xFFの範囲の文字の最上位ビットを除去します。この「変換」は、ソースのスーパーセットである文字セットにレプリケートする際に、データロスを引き起こす可能性があります。

> **注意:** Oracleは、文字セットが同一の場合は文字を変換しません。したがって、WE8ISO8859P1のデータをWE8ISO8859P1の文字セットを持つデータベースにポストすると、Oracleの変換プロセスがバイパスされます。

データを変換せずに適用するには:

SP_OPO_NLS_CONVERSIONパラメーターを1に設定し、データを変換して適用します。

> **注意:** ソースデータベースのNLS_NCHAR_CHARACTERSETとターゲット データベースのNLS_NCHAR_CHARACTERSETが異なる場合、SharePlexは、常にNVARCHARとNCLOBのデータを変換します。

デフォルト: 1

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Postの再開時

SP_OPO_NLS_DEFAULT_COMPAT

このパラメーターは、Postが文字データ（CHAR、VARCHAR）を変換するかどうかを制御します。デフォルトでは変換は行われません。7に設定すると、Postは文字データは変換しませんが、CLOBデータは変換します。compare/repairコマンドは、無変換の文字データと変換後のCLOBデータをcompare・repairします。NCHARデータのレプリケーションには影響しません。

SP_OPO_NLS_DEFAULT_COMPATはSP_OPO_NLS_CONVERSIONよりも優先されます。

デフォルト: 0

有効な値の範囲: 0または7

有効になるタイミング: Postの再開時

SP_OPO_OBJID_DEBUG

このパラメーターは、Postの特定のオブジェクト（テーブルなど）のデバッグを可能にします。SP_OPO_OBJID_DEBUGパラメーターは通常、非同期のデバッグに使用されます。SP_OPO_OBJID_DEBUGパラメーターを使用するには、ターゲットシステムでその値を、ソースデータベースのオブジェクトのオブジェクトIDの値に設定します。

デフォルト: 0

有効な値の範囲: 0以上の任意の数値

有効になるタイミング: 即時

SP_OPO_ONELINE_DEBUG

このパラメーターは、Postプロセスのオンラインデバッグ機能を制御します。

オンラインデバッグ機能により、システムリソースを大量に消費することなく、非同期エラーのデバッグを実行できます。オンラインデバッグでは、データストリームで非同期状態が発生した場所を検出するのに十分な情報を1行で記録します。問題を引き起こしているプロセスがオンラインデバッグで特定されたら、そのプロセスについて通常のデバッグを有効にすることができます。

このパラメーターを有効にするには、デバッグするオブジェクトをオブジェクトIDで指定し、それぞれをカンマで区切ります。

例:

```
sp_ctrl> set sp_opo_oneline_debug 230230, 351626
```

以下の項目が記録されます。

- 操作の状態。**Dispatch**（キューからの読み込み）、**ProcessMessage**（SQLコードの処理）、**Skipped**（コミット削減機能が有効な場合はコミット）、または**ExecSQL**（ターゲットへの適用）のいずれか
- SharePlexが使用するトランザクション識別子
- 影響を受けたDMLまたはDDL操作のタイプ
- 影響を受けるソーステーブルのオブジェクトID
- 影響を受ける行の行識別子（Oracle行ID）
- 操作が行われた時刻（Oracle）
- 影響を受ける操作のログシーケンス番号とログ内のオフセット
- トランザクションのOracle SCN
- 冗長な操作を防ぐ内部SharePlexコード

デフォルト: 無効

有効な値の範囲: オブジェクトIDによりリストされるオブジェクトのリストを形成する文字列（カンマで区切る）

有効になるタイミング: 即時

SP_OPO_OUT_OF_SYNC_SUSPEND

このパラメーターは、非同期状態を検出したときのSharePlexの動作を以下に基づいて制御します。

- INSERT: 行が既に存在するため、一意制約違反エラーが発生します。
- UPDATEとDELETE: 行がデータベースにありません。

値0（デフォルト）

トランザクションに非同期の操作が含まれる場合のPostのデフォルトの動作は、トランザクション内の他の有効な操作の処理を継続し、レイテンシを最小限に抑え、ターゲットを可能な限り最新の状態に保つことです。レイテンシとは、ソースでトランザクションが発生してから、それがターゲットに適用されるまでの時間のことです。きわめて量の多いトランザクションやネットワークトラフィックの中断など、さまざまな要因がレプリケーションのレイテンシの大きさに影響を与えます。

Postは、SQLステートメントと非同期の操作のデータを`ID_errlog.sql`ログファイルに記録します。ここで、*ID*はデータベース識別子です。このファイルは、ターゲットシステムの変数データディレクトリの**log**サブディレクトリにあります。

値1

値1を指定すると、Postは非同期状態を検出したときにポストを停止します。Postは、SQLステートメントと非同期の操作のデータを*ID*_**errlog.sql**ログファイルに記録し、その後停止します。

このパラメーターを1の値に設定した場合は、レプリケーションプロセスのステータスを頻繁にチェックしてください。Postを停止すると、ソースデータベースとターゲットデータベース間にレイテンシが発生し、レプリケーションキューにデータが蓄積されるため、利用可能なディスク容量を超える可能性があります。SharePlexには、レプリケーションのステータスをチェックするためのいくつかの**sp_ctrl**コマンドと、無人でSharePlexを監視するためのツールがあります。SharePlexを監視する方法の詳細については、『SharePlex管理ガイド』を参照してください。

トランザクションに非同期の操作が含まれている場合にトランザクション全体をロールバックするようにPostを設定するには、SP_OPO_SAVE_OOS_TRANSACTIONも参照してください。

デフォルト: 0（非同期状態では停止しない）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

SP_OPO_POSTER_DELAY

このパラメーターは、Postプロセスがターゲットインスタンスにレコードをポストするまでの遅延時間を制御します。通常、Postはソースシステムとターゲットシステム間のレイテンシを最小にするために変更を即座に適用しますが、最大5日間（7200分）ポストを遅らせることができます。

ポストを遅らせることで、ソースシステムでの不要な削除やオブジェクトのドロップによる偶発的な損失からデータを保護することが可能になります。この遅延により、間違いを検知し、間違いがレプリケートされる前にターゲットインスタンスからデータを回収するのに十分な時間が得られます。「what-if」分析を実行することも、このパラメーターをデフォルトから変更する理由のひとつです。ターゲットデータベースを時間的に遅らせることで、実際のものとcompareして予測モデリングを検証することができます。

SP_OPO_POSTER_DELAYによる遅延は、メッセージが最初にREDOログに表示された時点から測定されます。

このパラメーターを使用する際は、以下の点を考慮してください。

- 指定された時間が経過してデータがデータベースに適用されるまでの間に、postキューにデータを収集するために十分なディスク容量があることを確認してください。
- パラメーターの設定に基づいていつポストを開始するかを決定するために、SharePlexはターゲットシステムの現在のタイムスタンプと、ソースシステムのREDOログにレコードが記録された時刻をcompareします。これらのマシンは、別々の、互いに何千マイルも離れている場所にある可能性があります。タイムゾーンの差を考慮し、それを遅延時間に追加します。

  例えば、アメリカのカリフォルニア州ロサンゼルスとオーストラリアのシドニーでは18時間の時差があります。ポストを5時間遅らせるには、希望する5時間の遅延と18時間の時差を考慮して、SP_OPO_POSTER_DELAYを23時間（1,380分）に設定する必要があります。また、その場所がサマータイムを採用しているかどうかも考慮してください。それにより、2地点間の時差が変わる可能性があります。サマータイムの実施方法は、サマータイムを導入している国によって異なり、同じ国内でも異なる場合があります。
- このパラメーターは、例えば、ソースシステムがシドニーにあり、ターゲットシステムがロサンゼルスにある場合など、時間の流れを逆行するような形で使用しないでください。この場合、SharePlexが各システムのタイムスタンプをcompareすると、マイナスの時間差になります。例えば、シドニーの4月2日午前2時は、ロサンゼルスの4月1日午前8時です。このパラメーターは0未満の設定には対応していません。

デフォルト: 0分

有効な値の範囲: 0～7,200分

有効になるタイミング: 即時

SP_OPO_PRB_MISMATCH_SUSPEND

このパラメーターを1に設定すると、部分的ロールバックで不一致が発生したときにPostが停止します。これにより、問題を調査し、解決してから、Postの処理を再開できます。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

SP_OPO_READRELEASE_INTERVAL

このパラメーターは、Postの読み取り/リリースの頻度を制御します。通常、Postは各COMMITを受信した後に読み取り/リリースを実行します。これは、チェックポイントのリカバリプロセスの一環として、キューからそのトランザクションのデータを削除することを意味します。小規模なトランザクションの場合は、これが要因となってターゲットシステムで過剰なI/Oが行われ、Postの速度を低下させます。ほとんどのトランザクションが小規模の場合は、PostがCOMMITの後に毎回行うのではなく、一定のメッセージ数に達した後に読み取り/リリースを行うようにこのパラメーターを設定できます。

読み取り/リリースの間隔を設定する場合、コミットされたトランザクションの後に新しいデータが続かない場合（例えば、その日の最後のトランザクションの場合や、それ以上のユーザアクティビティがない場合）、PostはCOMMITを処理し、内部パラメーターSP_OPO_IDLE_LOGOUTによって制御される一定時間待機することに注意してください。データが到着しない場合、PostはSP_OPO_READRELEASE_INTERVAL間隔が満たされていなくても読み取り/リリースを実行します。postキュー内のメッセージ数が0になり、Postがキューからのメッセージをすべて処理し終わったことを示します。

注意: Postプロセスには複数のスレッドがあるため、メッセージ数はキューではなく各スレッドに関連付けられます。そのため、読み取り/リリースの間隔が予想よりも長くなり、デフォルト値を小さくする必要がある場合があります。

デフォルト: 100

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_OPO_REDUCED_KEY

このパラメーターはPostのWHERE句の内容を制御します。PostはWHERE句を使用して、ソースから複製されたUPDATEによって変更する必要がある行をターゲットで見付けます。SharePlexの機能によって、PostのWHERE句で使用する必要のあるデータの量が異なることがあります。

- 0に設定すると、このパラメーターは、Captureから送信されるすべてのデータのWHERE句を作成するようにPostに指示します。Captureが送信するデータは、SP_CAP_REDUCED_KEYパラメーターの設定によって異なります。キーの値と、LONGまたはLOB以外のすべての列をWHERE句に含めるには、両方のパラメーターを0に設定します。

  重要! ピアツーピア構成では0に設定する必要があります。

- 1に設定した場合、このパラメーターは、変更された列のキー値と変更前の値でWHERE句を構築するようにPostに指示します。この設定により、PostはWHERE句を作成する際に、変更された列の変更前後のcompareができるため、レプリケーションのパフォーマンスとターゲットデータ整合性のバランスが適切になります。この設定を使用するには、SP_CAP_REDUCED_KEYパラメーターを0または1に設定する必要があります。SharePlexが変更前後の値をどのように使用するかの詳細については、『SharePlex管理ガイド』を参照してください。
- 2に設定すると、このパラメーターはキー列のみのWHERE句を作成するようにPostに指示します。この設定は、ポストのパフォーマンスを最大化するために使用できます。この設定では、変更された列の変更前後のcompareが省略されるため、SharePlex以外のプロセスやユーザがターゲットデータを変更できないようにする必要があります。さらに、**compare**コマンドを使用して、定期的に完全性の検証を行うことをお勧めします。

> **注意:** 水平分割レプリケーションでSP_OPX_REDUCED_KEYパラメーターが2に設定されている場合、PostはUPDATE句のために追加される列のキーと変更前のデータでWHERE句を準備します。

SP_CAP_REDUCED_KEYも参照してください。

デフォルト: 0

有効な値の範囲: 0、1、2

有効になるタイミング: Postの再開時

SP_OPO_RETRIES_MAX

このパラメーターは、初回に失敗したSQLステートメントのポストをPostプロセスが試行する回数を制御します。

Postは、失敗した操作のうち、再試行すれば成功する可能性があるものを再試行します。Postが再試行する主な操作は、TNSの書き込みの失敗、接続の失敗、またはPostがTRUNCATEを適用する必要があるテーブルのロックです。

失敗した操作が成功する可能性を高めるには、SP_OPO_RETRIES_MAXパラメーターの値を大きくして、Postが試行できる操作の回数を増やします。同時に、SP_OPO_RETRY_DELAY_TIMEパラメーターの値を大きくして、試行間隔を長くします。その結果、試行と次の試行の間にロックやその他のブロック操作を解決するのに十分な時間を確保できます。

Postプロセスがエラー発生時に続行するように設定されている場合（SP_SYS_SUSPEND_ON_ERROR=0）、またはエラーメッセージが**oramsglist**ファイルに記載されている場合、Postはキュー内の次のトランザクションに移ります。それ以外のすべての場合では、最大試行回数に達するとPostは停止します。

> **注意:** **oramsglist**ファイルの詳細については、『SharePlex管理ガイド』を参照してください。

> **重要:** このパラメーターの値を小さくすると、キューにデータが蓄積され、利用可能なディスク容量を超える可能性があります。

デフォルト: 10回

有効な値の範囲: 0または任意の正の整数

有効になるタイミング: 即時

SP_OPO_SAVE_OOS_TRANSACTION

このパラメーターは、トランザクションに非同期操作が含まれている場合に、Postがトランザクションをロールバックして破棄するかどうかを制御します。この機能は、`SP_OPO_SAVE_OOS_TRANSACTION`パラメーターによって制御されます。

このパラメーターの使用方法

このパラメーターが1に設定され、トランザクションに非同期エラーを発生させる操作が含まれている場合、Postはトランザクション全体を破棄し、そのトランザクションのすべての操作をSQLファイルに保存します。このファイルを編集して失敗したSQLの問題をrepairしてから、ファイルを実行してターゲットデータベースにトランザクションを適用することができます。

このパラメーターを1に設定すると、Postはロールバックされたトランザクションに続く有効なトランザクションの処理をデフォルトで継続します。これはターゲットのレイテンシを防止するためです。トランザクションをロールバックした後にPostを停止するように設定するには、SP_OPO_OUT_OF_SYNC_SUSPENDパラメーターを1に設定します。

重要! このパラメーターは、アプリケーションが*1つのトランザクション内*で相互に依存するすべての変更を行うことが分かっている場合にのみ使用してください。

ロールバックされるトランザクションに関係するすべてのテーブルには、以下のデータ型のみが含まれている必要があります。

- CHAR – US7ASCII
- VARCHAR – US7ASCII
- Number
- DATE
- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE
- INTERVAL
- ROWID
- RAW
- BINARY FLOAT
- BINARY DOUBLE

ロールバックされたトランザクションには、それぞれ独自のSQLファイルがあります。ファイル名は*SCN_queue*.**sql**です。ここで、

- *SCN*はトランザクションのコミットシステム変更番号(SCN)です。
- *queue*は、トランザクションが含まれるPostキューの名前です。

ファイル名の例:

4346118046_postq1.sql

メモ:

- 各SQLファイルは、SharePlex変数データディレクトリ内の**oos**ディレクトリに保存されます。**ora_cleansp**ユーティリティを使用してSharePlex環境を再起動する場合は、ファイルの場所を変更する必要があるかもしれません。このユーティリティを実行すると、トランザクションファイルが削除されます。**target**コマンドの**file**カテゴリのオプションのいずれかを使用して、SQLファイルの場所、サイズ、数を変更することができます。
- この機能が有効な場合、Postはトランザクションの非同期の部分のみを**errlog.sql**ファイルに書き込むのではなく、トランザクション全体を*SCN_queue*.**sql**ファイルに書き込みます。
- この機能はOracleターゲットのみをサポートしています。
- Postは引き続き非同期メッセージを**statusdb**に生成し、**sp_ctrl**で**show statusdb**コマンドが発行されると、そのメッセージが表示されます。SQLファイルを使用してトランザクションを適用した後、**clear status**コマンドを使用して、これらのステータスメッセージを消去できます。

- ロールバックしたトランザクションを「ロールバック解除」することはできません。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Postの再開時

SP_OPO_SQL_CACHE_DISABLE

このパラメーターはSQL Cache機能を有効または無効にします。デフォルトではSQL Cacheはオンで、このパラメーターは0に設定されます。1の設定では、SQL Cacheは無効になります。3の設定ではバッチ操作のSQL Cacheが無効になり、Postが使用するメモリの量が減少します（2の設定は不可）。

SQL Cacheを無効にすると、SharePlexはイベントログに以下のメッセージを表示します。`SQL Cache disabled.`（SQL Cacheが無効になりました。）

SQL Cacheの詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 0（オン）

有効な値の範囲: 0、1、3

有効になるタイミング: Postの再開時

SP_OPO_STOP_ON_DDL_ERR

このパラメーターは、DDLの適用中にエラーが発生した場合にPost処理を停止するかどうかを制御します。デフォルトの1は、エラーが発生した場合にPostを停止させます。エラーは通常、DDLが実行されたソースコンポーネントがターゲットデータベースに存在しないことを示し、後続のDMLの変更も失敗する可能性があることを示します。Postを停止することでDMLの失敗を防止し、問題を修正してデータベースの同期を保つことができます。

デフォルト: 1（オン）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

SP_OPO_SUPPRESS_DELETE

このパラメーターは、PostがターゲットにDELETE操作を適用するかどうかを制御します。DELETE操作を抑制することは、データウェアハウスのように、ソーステーブルから削除されても行が一元的に存在しなければならない状況では適切である可能性があります。このパラメーターは、設定ファイルを変更したりアクティベーションしたりすることなく、設定および設定解除できます。DELETEを抑制するには、1に設定します。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: プロセスの再開時

SP_OPO_SUPRESSED_OOSを

このパラメーターは、ソースからレプリケート中の変更がターゲット行に既に含まれていることをPostが検出した場合に、非同期メッセージを返すかどうかを制御します。この条件が満たされた場合、Postはレプリケートされ たSQLを破棄し、イベントログまたは**errlog.sqlログ**ファイルに非同期メッセージを書き込みません。

このパラメーターは、以下の場合にINSERT、UPDATE、およびDELETE操作をサポートします。

- レプリケートされたINSERTのすべての値(キー値を含む)が、ターゲット内の行の既存の値と一致する。
- UPDATEのキー値がターゲットの行と一致し、その行の既存の値が、ソースからレプリケートされた(変更)後の値と一致する。

注意: SharePlex は、ターゲット値がレプリケートされた後の値と一致しない場合、非同期メッセージを返します。

- DELETE操作のターゲット行が存在しない。

デフォルトでは、このパラメーターは有効になっています。ただし、Postで検出されない非同期値が存在する可能性があるため、**compare**コマンドのような、すべてのターゲットデータを検証するツールを使用する必要があります。

デフォルト: 1

有効な値の範囲: 0(無効)または1

有効になるタイミング: 即時

SP_OPO_SYNC_LOG_FREQUENCY

このパラメーターは、Postが非同期メッセージをイベントログ(変数データディレクトリの**event_log**ファイル)に記録する頻度を制御します。Postは非同期状態を検出すると、非同期イベントを生成します。ターゲットテーブルが非同期であるため、イベントログが非同期メッセージでいっぱいになっている場合、このパラメーターを使用して、これらのメッセージがログに記録される頻度を制御することができます。

このパラメーターが0より大きい場合、Postは最初の非同期エラーをログに記録し、その後は指定された間隔でのみ非同期メッセージをログに記録します。Postは、非同期テーブルが再同期できるようになるまで、有効なデータをポストし続けます。

デフォルト: 1(非同期ッセージを毎回記録)

有効な値の範囲: 0より大きい整数

有効になるタイミング: 即時

SP_OPO_THREADS_MAX

このパラメーターは主にテストで使用され、SQLスレッドが複数のサブキューを処理しているときの動作を確認します。最小のスレッド数は4で、Postプロセスは、メインスレッド、タイムキーパースレッド、シグナル・ウェイター・スレッド、および1つのSQLスレッドを作成できます。

デフォルト: 128スレッド

有効な値の範囲: 4スレッド（最小）～1024スレッド（最大）

有効になるタイミング: Postプロセスの再開時

SP_OPO_TRACK_COMMITS

このパラメーターは、Postプロセスがコミットを追跡するかどうかを制御します。1に設定すると、Postはコミットごとに行を挿入します。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Postプロセスの再開時

SP_OPO_TRACK_OPERATIONS

このパラメーターは、変更履歴ターゲットデータベースを管理する際に、PostがどのDMLを追跡するかを制御します。デフォルトでは、Postはすべての DMLタイプを追跡します。例えば、挿入と更新のみを追跡するようにPostを設定するには、パラメーターをI/Uに設定します。

デフォルト: I/U/D（挿入、更新、および削除）

有効な値の範囲: スラッシュ(/)で区切られたI、U、Dの任意の組み合わせ

有効になるタイミング: Postプロセスの再開時

SP_OPO_TRACK_PREIMAGE

このパラメーターは、Postプロセスが挿入および更新の変更前イメージ、または削除の変更後イメージを追跡するかどうかを制御します。挿入の変更前イメージと削除の変更後イメージには、キー値が含まれ、その他の列はすべてNULLに設定されます。

このパラメーターは、I、U、D、またはスラッシュで区切られたこれらの値の任意の組み合わせ(I/Uなど)に設定できます。このパラメーターを使用すると、Postは指定されたタイプの操作ごとに2つの行を適用します。1つには変更前イメージの列値、もう1つには変更後イメージの値が入ります。

デフォルト: オフ

有効な値の範囲: **I**、**U**、**D**

有効になるタイミング: Postプロセスの再開時

SP_OPO_TRUSTED_SOURCE

このパラメーターは、*信頼できるソースシステム*を指定します。このシステムには、ピアツーピアレプリケーション環境でプライマリデータセットとみなされるデータが格納されています。信頼できるホストは、カスタムのコンフリクト解決ルーチンで使用することができ、SharePlexで提供される準備されたルーチンの1つの基礎でもあります。コンフリクト解決とピアツーピアレプリケーションの詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: なし

有効な値の範囲: SharePlexレプリケーション環境で有効な任意のホスト名

有効になるタイミング: Postプロセスの再開時

SP_OPO_UPDATE_SCN

このパラメーターはOracle SCNの追跡を制御し、高可用性クラスタでの操作時にSharePlexをすばやく回復させることができます。このパラメーターを有効にすると、SharePlexは内部トランザクションテーブルでSCNを追跡します。これによりSharePlexは、最も適用されていないトランザクションまでアクティベーションするため、フェールオーバーの発生時に正しいトランザクションに合わせて調整することができます。

注意: このパラメーターを有効にすると、PEP（Post Enhanced Performance）機能のコミット削減機能が無効になります。SCNの追跡時にはこの機能はサポートされません。

デフォルト: 0

有効な値の範囲: 0（無効）または1（有効）

有効になるタイミング: Postの再開時

SP_OPO_USE_VARNUM

このパラメーターを指定すると、Postはターゲットデータベースに数値を入力する際にVARNUMを使用するようになります。これにより、データベースの数値列に非標準の高精度値を入力することができます。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Postの再開時

SP_OPO_WAIT_MSG_DELAY

このパラメーターは、Postが古いまたは停止中のOracleセッションを強制終了することを伝えるメッセージを生成するまでに待機する時間を制御します。これらのメッセージはPostの起動時に発生します。

デフォルト: 300秒

有効な値の範囲: 86,400までの任意の正の整数

有効になるタイミング: Postの再開時

Open Target Postパラメーター

これらのパラメーターは、Open Target（Oracle以外）にデータを適用する際にSharePlexのPostプロセスで使用されます。

SP_OPX_BATCH_ENABLE

同じ操作の複数のトランザクションを適用するときにパフォーマンスを最大化するために、Postプロセスはトランザクションをグループ化し、バッチとして適用することができます。SP_OPX_BATCH_ENABLEパラメーターはこの機能を制御します。デフォルトでは有効になっています。SP_OPX_MAX_BATCH_MBYTESはバッチの最大サイズを制御します。

デフォルト: 1（有効）

有効な値の範囲: 0または1

有効になるタイミング: 即時

SP_OPX_COMMIT_REDUCE_MSGS

Post Enhanced Performance機能のコミット削減コンポーネントにしきい値を設定します。コミット削減機能により、小規模なトランザクションのバッチを大規模なトランザクションにまとめるようにPostを設定することができます。これにより、処理しなければならないコミットや確認の数を減らすことができます。トランザクションが小さければ小さいほど、パフォーマンスは向上します。

指定されたメッセージ数に達すると、Postはコミットを発行します。トランザクションの境界がこのメッセージの範囲内にあるトランザクションのコミットはスキップされ、それらのトランザクションはすべて1つのトランザクションとしてコミットされます。コミット削減はデフォルトでオンになっています。無効にするには、このパラメーターの値を1に設定します。

このパラメーターの設定は絶対的なしきい値ではありません。SharePlexは、結合された異なるトランザクションにトランザクションを分割することはありません。したがって、すべての操作とグループ内の最後のトランザクションのコミットを含めるために、Postがそのしきい値を超えなければならない可能性があります。

デフォルト: 100個のメッセージ

有効な値の範囲: 任意の正の整数

有効になるタイミング: Postの再開時

SP_OPX_CONT_ON_ERR

このパラメーターは、特定のODBCエラーが発生したときにPostを停止するかどうかを制御します。このフラグをデフォルトの0に設定すると、PostはすべてのODBCエラーで停止します。特定のエラーを無視して処理を続行するようにPostを設定するには、以下のファイル（データベースによって異なる）のいずれかにそのエラーをリストし、SP_OPX_CONT_ON_ERRを1に設定します。

hanamsglist

mysqlmsglist

postgresmsglist

sqlservermsglist

sybasemsglist

tdmsglist

これらのファイルの保存場所は、変数データディレクトリの**data**サブディレクトリです。これらのファイルの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

無視した各エラーについて、PostはSharePlex **errlog.sql**ファイルにエラーを記録し、その後にポストを再開します。ファイルに指定されていないエラーをPostが受け取った場合、プロセスは停止します。

デフォルト: 0(すべてのエラーで停止)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: 即時

SP_OPX_DISABLE_OBJECT_NUM

このパラメーターは、**ソース**テーブルのオブジェクトIDに基づいて、レプリケートされたDMLおよびDDL操作をPostがターゲットにポストしないようにします。ソーステーブルのデータが無効であるか破損している場合、ソーステーブルにターゲットでサポートされていないデータ型が含まれている場合、またはテーブルの操作をターゲットデータベースに反映させたくないその他の理由がある場合に、このパラメーターを設定できます。

このパラメーターは1つまたは複数のテーブルに設定できます。これにより、データを再同期して設定ファイルを再度アクティベーションする時間ができるまで、ソーステーブルのターゲットテーブルでそれ以上レプリケーションが実行されないようにします。Postプロセスは、これらのテーブルの複製されたすべてのメッセージをpostキューから破棄するため、メッセージはキューに蓄積されません。

このパラメーターは注意して使用してください。これを有効にしてソーステーブルに対してDDLまたはDMLを実行すると、変更がポストされないため、ターゲットデータが古くなります。レプリケーション設定外の他のテーブルの外部キーなど、テーブルに依存関係がある場合は、ポストを無効にすると依存関係が満たされなくなります。

このパラメーターは、デフォルトでは無効になっています。この機能を有効にするには、ターゲットシステムで以下のコマンドを実行します。ここに表示される数字は、ポストから除外するソーステーブルのオブジェクトIDです。

```
sp_ctrl(sysB)> set param SP_OPX_DISABLE_OBJECT_NUM 12345,67890
```

各オブジェクトIDをカンマで区切り、その間にはスペースを入れません。エントリ間にスペースを使用するには、以下の例のようにリスト全体を引用符で囲みます。

```
sp_ctrl(sysB)> set param SP_OPX_DISABLE_OBJECT_NUM "498438, 1000, 497109"
```

無効なオブジェクトIDは無視されます。

ターゲットテーブルへのポストを再開する準備ができたら、SP_OPX_DISABLE_OBJECT_NUMを0に設定します。

デフォルト: 0(オフ)

有効な値の範囲: カンマで区切られた有効なオブジェクトIDのリスト

有効になるタイミング: 即時

SP_OPX_MAX_BATCH_MBYTES

このパラメーターは、バッチDML操作の最大サイズを制御します。ターゲットにSQL操作を適用する際のパフォーマンスを最大化するために、Postプロセスは小さなレプリケート操作を1つの大きなバッチ操作にグループ化します。操作が同一で

SQLが類似している場合、Postはレプリケートされた操作をグループ化します。Postは、SP_OPX_MAX_BATCH_MBYTESで指定されたバッチサイズに達するまで、(条件が一致する)操作をバッチに追加し続けます。

SP_OPX_BATCH_ENABLEパラメーターはこの機能を有効にします(デフォルトでは有効)。

デフォルト: 1メガバイト

有効な値の範囲: 1～任意の正の整数

有効になるタイミング: 即時

SP_OPX_MSGS_IN_MEMORY

このパラメーターは、Postプロセスが使用するメモリの量を制御します。Open TargetのPostプロセスは、トランザクションデータを一時的に保存するために内部キューを使用します。このパラメーターは、この内部キューが使用するメモリの最大量を制御します。

デフォルト: 5,000個のメッセージ

有効な値の範囲: 整数、最大値なし

有効になるタイミング: Postの再開時

SP_OPX_NLS_CONVERSION

このパラメーターは、Oracleソースと非Oracleターゲット間の文字セットの変換を制御します。

Open Targetターゲット(Oracle以外のターゲット)にレプリケートする場合、SharePlexは、任意のOracle Unicode文字セットおよびUS7ASCII文字セットからのレプリケーションをサポートします。SharePlexはUnicode文字セットでOpen Targetターゲットにデータをポストするため、ソースデータがUnicodeまたはUS7ASCIIの場合、ターゲット上で変換する必要はありません。

ただし、以下の項目に当てはまる場合は、ターゲット上で変換が必要となります。

- ソースデータの文字セットがOracle UnicodeまたはUS7ASCII以外の場合、ターゲットに転記するためにUnicodeに変換するには、ターゲットにOracleクライアントをインストールする必要があります。
- データをUnicode以外の文字セットでターゲットデータベースにポストする必要がある場合は、ターゲットにOracleクライアントをインストールして変換を実行し、**target**コマンドを使用してPostが使用するターゲット文字セットを特定する必要があります。
- LOBデータをレプリケートする場合は、ソースの文字セットにかかわらず変換が必要です。

Linux上のOracleクライアントで変換を実行するには:

1. Oracle *Administrator*クライアントをターゲットシステムにインストールします。クライアントは管理者インストールタイプでなければなりません。Instant ClientおよびRuntimeインストールタイプはサポートされていません。
2. ORACLE_HOMEをクライアントのインストールに設定します。ORACLE_SIDをエイリアスまたは存在しないSIDに設定します。SharePlexはこれらのSIDを使用せず、データベースが実行されている必要はありません。
3. お使いのオペレーティングシステム用のLinux/Unixインストーラーを使用してSharePlexをします。
4. SP_OPX_NLS_CONVERSIONパラメーターがデフォルトの1に設定されていることを確認してください。

UnicodeとUS7ASCIIのデータを変換せずに適用するには:

ソースデータがUnicodeまたはUS7ASCIIで、LOBデータをレプリケートしない場合は、変換やOracleクライアントは必要ありません。SP_OPX_NLS_CONVERSIONパラメーターを0に設定して変換を無効にし、Postが実行中であれば再起動します。

デフォルト: 1

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: Postの再開時

SP_OPX_ONELINE_DEBUG

このパラメーターは、Open TargetのPostプロセスのオンラインデバッグ機能を制御します。

オンラインデバッグ機能により、システムリソースを大量に消費することなく、非同期エラーのデバッグを実行できます。オンラインデバッグでは、データストリームで非同期状態が発生した場所を検出するのに十分な情報を1行で記録します。問題を引き起こしているプロセスがオンラインデバッグで特定されたら、そのプロセスについて通常のデバッグを有効にすることができます。

このパラメーターを有効にするには、デバッグするオブジェクトをオブジェクトIDで指定し、それぞれをカンマで区切ります。

例:

```
sp_ctrl> set sp_opo_oneline_debug 230230, 351626
```

以下の項目が記録されます。

- 操作の状態(**Process**(キューからの読み取り)、**Skipped**(コミット削減機能が有効な場合はコミット)、**Applied**(ターゲットへの適用))
- SharePlexが使用するトランザクション識別子
- 影響を受けたDMLまたはDDL操作のタイプ
- 影響を受けるソーステーブルのオブジェクトID
- 影響を受ける行の行識別子(Oracle行ID)
- 操作が行われた時刻(Oracle)
- 影響を受ける操作のログシーケンス番号とログ内のオフセット
- トランザクションのOracle SCN
- 冗長な操作を防ぐ内部SharePlexコード

デフォルト: 無効

有効な値の範囲: オブジェクトIDによりリストされるオブジェクトのリストを形成する文字列(カンマで区切る)

有効になるタイミング: 即時

SP_OPX_ONELINE_DEBUG_COLUMNS

このパラメーターは、SP_OPX_ONELINE_DEBUGが有効な場合に列の値をログに記録するかどうかを制御します。これは、非同期の状態を診断するのに役立ちます。オンラインデバッグが有効になっているときに指定されたオブジェクトの列の値をログに記録するには、SP_OPX_ONELINE_DEBUG_COLUMNSを1に設定します。

デフォルト: 0(無効)

有効な値の範囲: 0または1

有効になるタイミング: 即時

SP_OPX_OUT_OF_SYNC_SUSPEND

このパラメーターは、非同期状態を検出したときのSharePlexの動作を以下に基づいて制御します。

- INSERT: 行が既に存在するため、一意制約違反エラーが発生します。
- UPDATEとDELETE: 行がデータベースにありません。

値0(デフォルト)

トランザクションに非同期の操作が含まれる場合のPostのデフォルトの動作は、トランザクション内の他の有効な操作の処理を継続し、レイテンシを最小限に抑え、ターゲットを可能な限り最新の状態に保つことです。レイテンシとは、ソースでトランザクションが発生してから、それがターゲットに適用されるまでの時間のことです。きわめて量の多いトランザクションやネットワークトラフィックの中断など、さまざまな要因がレプリケーションのレイテンシの大きさに影響を与えます。

Postは、SQLステートメントと非同期の操作のデータを`ID_errlog.sql`ログファイルに記録します。ここで、*ID*はデータベース識別子です。このファイルは、ターゲットシステムの変数データディレクトリの**log**サブディレクトリにあります。

値1

値1を指定すると、Postは非同期状態を検出したときにポストを停止します。Postは、SQLステートメントと非同期の操作のデータを*ID*_**errlog.sql**ログファイルに記録し、その後停止します。

このパラメーターを1の値に設定した場合は、レプリケーションプロセスのステータスを頻繁にチェックしてください。Postを停止すると、ソースデータベースとターゲットデータベース間にレイテンシが発生し、レプリケーションキューにデータが蓄積されるため、利用可能なディスク容量を超える可能性があります。SharePlexには、レプリケーションのステータスをチェックするためのいくつかの**sp_ctrl**コマンドと、無人でSharePlexを監視するためのツールがあります。SharePlexを監視する方法の詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 0(非同期状態では停止しない)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: 即時

SP_OPX_READRELEASE_INTERVAL

このパラメーターは、Postがトランザクション方式でJMSサーバにポストするように設定されている場合に使用します(**target**コマンドの**session_transacted**プロパティで指示されるようにすべてのメッセージの後ではなく、一定間隔でJMSコミットを発行)。

このパラメーターは、読み取り/リリースの間隔(Postがキューからトランザクションのデータをパージする頻度)を制御します。

デフォルト: 1,000

有効な値の範囲: 任意の正の整数

有効になるタイミング: Postの再開時

SP_OPX_REDUCED_KEY

このパラメーターはPostのWHERE句の内容を制御します。PostはWHERE句を使用して、ソースから複製されたUPDATEによって変更する必要がある行をターゲットで見付けます。SharePlexの機能によって、PostのWHERE句で使用する必要のあるデータの量が異なることがあります。

- 0に設定すると、このパラメーターは、Captureから送信されるすべてのデータのWHERE句を作成するようにPostに指示します。Captureが送信するデータは、SP_CAP_REDUCED_KEYパラメーターの設定によって異なります。キーの値と、LONGまたはLOB以外のすべての列をWHERE句に含めるには、両方のパラメーターを0に設定します。

  重要! ピアツーピア構成では0に設定する必要があります。

- 1に設定した場合、このパラメーターは、変更された列のキー値と変更前の値でWHERE句を構築するようにPostに指示します。この設定により、PostはWHERE句を作成する際に、変更された列の変更前後のcompareができるため、レプリケーションのパフォーマンスとターゲットデータ整合性のバランスが適切になります。この設定を使用するには、SP_CAP_REDUCED_KEYパラメーターを0または1に設定する必要があります。SharePlexが変更前後の値をどのように使用するかの詳細については、『SharePlex管理ガイド』を参照してください。
- 2に設定すると、このパラメーターはキー列のみのWHERE句を作成するようにPostに指示します。この設定は、ポストのパフォーマンスを最大化するために使用できます。この設定では、変更された列の変更前後のcompareが省略されるため、SharePlex以外のプロセスやユーザがターゲットデータを変更できないようにする必要があります。さらに、**compare**コマンドを使用して、定期的に完全性の検証を行うことをお勧めします。

  注意: 水平分割レプリケーションでSP_OPX_REDUCED_KEYパラメーターが2に設定されている場合、PostはUPDATE句のために追加される列のキーと変更前のデータでWHERE句を準備します。

SP_CAP_REDUCED_KEYも参照してください。

デフォルト: 0

有効な値の範囲: 0、1、2

有効になるタイミング: Postの再開時

SP_OPX_SQL_CACHE_DISABLE

このパラメーターはSQL Cache機能を有効または無効にします。デフォルトではSQL Cacheはオンで、このパラメーターは0に設定されます。1の設定では、SQL Cacheは無効になります。3の設定ではバッチ操作のSQL Cacheが無効になり、Postが使用するメモリの量が減少します(2の設定は不可)。

SQL Cacheを無効にすると、SharePlexはイベントログに以下のメッセージを表示します。`SQL Cache disabled.`(SQL Cacheが無効になりました。)

SQL Cacheの詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 0(オン)

有効な値の範囲: 0、1、3

有効になるタイミング: Postの再開時

SP_OPX_STOP_ON_DDL_ERR

このパラメーターは、DDLの適用中にエラーが発生した場合にPost処理を停止するかどうかを制御します。デフォルトの1は、エラーが発生した場合にPostを停止させます。エラーは通常、DDLが実行されたソースコンポーネントがターゲットデータベースに存在しないことを示し、後続のDMLの変更も失敗する可能性があることを示します。Postを停止することでDMLの失敗を防止し、問題を修正してデータベースの同期を保つことができます。

デフォルト: 1（オン）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 即時

SP_OPX_THREADS

このパラメーターは、Post Enhanced Performance機能のTransaction Concurrencyコンポーネントを制御します。2以上の値に設定すると、指定されたスレッド数（同時処理数）でPostが実行されます。

このパラメーターはSQL ServerおよびPostgreSQLデータベースをサポートしています。通常は、システム上のプロセッサーのコア数に設定しなければなりません。

Postのパフォーマンスを調整する方法の詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 1（無効）

有効な値の範囲: Max_connection値未満

有効になるタイミング: Postの再開時

SP_OPX_TRUSTED_SOURCE

このパラメーターは、*信頼できるソースシステム*を指定します。このシステムには、ピアツーピアレプリケーション環境でプライマリデータセットとみなされるデータが格納されています。このパラメーターは、SharePlexによって提供される準備されたコンフリクト解決ルーチンで使用されます。SP_OPX_TRUSTED_SOURCEパラメーターは、両方のサーバで同じ値に設定する必要があります。コンフリクト解決とピアツーピアレプリケーションの詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: なし

有効な値の範囲: SharePlexレプリケーション環境で有効な任意のホスト名

有効になるタイミング: Postプロセスの再開時

Queueパラメーター

これらのパラメーターは、SharePlexキューのプロパティを制御します。

SP_QUE_MAX_QUEUES

このパラメーターは、**sp_cop**のインスタンスごとに許可されるキューの最大数を決定します。デフォルトでは、各captureキューとexportキューは8 MBの共有メモリを使用し、各postキューは32 MBを使用します。システムに十分なメモリがある場合、このパラメーターの値を増やすことができます。SharePlexは起動時に、アクティブな設定ファイル内のキュー数に必要な共有メモリ量を表示します。

デフォルト: 25

有効な値の範囲: 9～254

有効になるタイミング: **sp_cop**の再開時

SP_QUE_POST_SHMSIZE

このパラメーターは、postキューの固有の共有メモリサイズを決定します。

重要: このパラメーターの値を大きくするには、システム設定で共有メモリの最大セグメントサイズを大きくする必要がある場合があります。

デフォルト: 32メガバイト

有効な値の範囲: 0より大きい整数

有効になるタイミング: **sp_cop**の再開時

SP_QUE_Q_SHMSIZE

このパラメーターは、captureキューおよびexportキューの固有の共有メモリサイズを決定します。

デフォルト: 8メガバイト

有効な値の範囲: 0より大きい整数、最大値なし

有効になるタイミング: **sp_cop**の再開時

SP_QUE_SHMDBUF

このパラメーターは、SharePlexプロセスが一度にキューから読み込んだり、キューに書き込んだりする情報量を決定する共有メモリのバッファサイズを制御します。バッファは、データの高速で安全な非同期転送を促進するSharePlexチェックポイント・リカバリ・プロセスの一部です。デフォルト設定は最適な値であり、テクニカルサポート担当者の指導または文書化

されたSharePlex手順がない限り変更しないでください。ただし、トランザクションの大半が大規模である場合は、このパラメーターを変更してもかまいません。初期設定を65536にしてみてください。このパラメーターに設定する値はすべて2の累乗にする必要があります。

重要: このパラメーターの値を64kより大きくするには、システム設定で共有メモリの最大セグメントサイズを大きくする必要がある場合があります。

デフォルト: 32,768バイト

有効な値の範囲: 2の累乗の任意の正の整数

有効になるタイミング: SharePlexの再開時

SP_QUE_SYNC

SP_QUE_SYNCは、キューデータのディスクへの書き込みがディスクメディアに書き込まれたことを検証してから戻るようにキューモジュールに指示します。これは、UnixやLinuxの標準のディスク書き込みモデルではありません。

デフォルトのOSディスクI/O手順では、ディスク書き込みはOSの内部バッファキャッシュに書き込まれ、その後ディスクに書き込まれます。この動作によって、ディスクへの書き込みのオーバーヘッドが分散され、プロセスはデータがディスクに書き込まれるのを待つ必要がなくなります。データブロックがバッファキャッシュに書き込まれると、データにアクセスするアプリケーションは、バッファキャッシュに書き込まれたデータとディスクに書き込まれたデータを区別できません。OracleのCOMMITとは異なり、ファイルにアクセスできるすべてのプロセスは、バッファキャッシュ内のデータにもアクセスできます。データブロックがバッファキャッシュに書き込まれてから物理メディアに書き込まれるまでの間にシステムクラッシュが発生した場合、メディアに書き込まれていないデータはシステムリカバリ時に失われる危険性があります。

SP_QUE_SYNCパラメーターは以下のように実装されます。

SP_QUE_SYNC=0

2つ目の段落で説明されているOSのデフォルトケースは、SharePlexキューデータに使用されるディスク書き込みアルゴリズムです。

SP_QUE_SYNC=1

SP_QUE_SYNC を**1**に設定すると、各キューデータとヘッダーファイルを開く際にO_SYNCフラグが設定されます。このフラグは、データがディスクに正常に書き込まれるまで、書き込み呼び出しを戻さないようにOSに指示します。同期フラグがない場合（「通常の」I/O）、ファイルデータ用に領域が割り当てられますが、システムクラッシュ、クラスタフェールオーバー、またはOSの実行を停止させるようなその他の重大な問題のために、ファイルデータが書き込まれない可能性があります。

SP_QUE_SYNC=2（SharePlexのデフォルト設定）

SP_QUE_SYNCパラメーターを**2**に設定しても、O_SYNCフラグはオンになりません。その代わり、キューへの書き込みCOMMITが呼び出されるまで、通常の書き込みが行われます。書き込みCOMMITの一部として、キューモジュールは各キュー・データ・ファイルに対してシステム呼び出しfsyncを実行し、さらにキュー・ヘッダー・ファイルに対してシステム呼び出しfsyncを実行します。これによって、後で書き込みCOMMITによって書き換えられる可能性のあるデータに対する冗長な同期操作が排除されます。

SP_QUE_SYNCの設定の影響

SP_QUE_SYNCを、ディスク書き込みが完了してから戻るような値に設定すると、SharePlexのパフォーマンスに影響を与える可能性があり、SharePlex以外のプロセスのI/O処理がわずかに増加する可能性があります。オーバヘッドの量は、キュー内のデータ量、ファイルシステムのタイプ、および使用中のディスクドライブとディスクコントローラのタイプに依存します。

SP_QUE_SYNCが実行しない機能

SP_QUE_SYNCパラメーターが実行しない機能は多数あります。最も一般的な問題は次の通りです。

- SP_QUE_SYNCは、共有メモリとディスクのどちらかを選択するわけではありません。また、ディスクへの書き込みの決定がいつ行われるかにも影響を与えません。このパラメーターは、物理ディスクへのディスク書き込みが完了してから戻るように、OSに指示するだけです。
- 「サブキュー不足」エラーを防ぐことはできません。
- 「ディスク容量不足」エラーには対処しません。物理メディアへの書き込みや同期が実行される前に、ファイルデータ用の領域が割り当てられるからです。容量不足エラーが発生する場合は、同期が実行される前に発生します。データが失われる可能性がありますが、空き領域が利用可能になるとキューを使用できるようになることが期待されます。
- 「共有メモリ不足」エラーや「仮想メモリ不足」エラーを引き起こしたり、防止したりしません。共有メモリの喪失を示すメッセージが表示される場合は、キューモジュールのデマンド・ページング・システムの障害である可能性が最も高いです。「仮想メモリ不足」エラーが発生した場合は、データセグメントの制限に関するOSのパラメーターを確認してください。通常または予想される制限値では不十分な場合は、テクニカルサポートにお問い合わせください。

クラスタでのSP_QUE_SYNCの使用

クラスタでは、クラスタのプライマリノード（共有変数データディレクトリがマウントされているノード）、および同じレプリケーション環境の一部である、クラスタ外の各ソースマシンおよびターゲットマシンでSP_QUE_SYNCを設定します。

デフォルト: 2

有効な値の範囲: 0～2

有効になるタイミング: SharePlexの再開時

SP_QUE_USE_SUBQUE_INDEX

このパラメーターはキューのパフォーマンスを向上させます。ソースで大量の同時処理が行われる場合、postキューには多数のサブキューが含まれる可能性があります（各サブキューはトランザクションセッションを表す）。これがPostプロセスのパフォーマンスを低下させる可能性があります。このパラメーターは、サブキュー構造へのアクセスに改善されたインデックスを使用するようにSharePlexに指示します。このパラメーターが有効になっているすべてのpostキューに対して、メッセージ「Subqueue index enabled *queuename*（サブキューインデックスが有効になっているキュー名）」がイベントログに書き込まれます。

このパラメーターはVARRAYをサポートしません。VARRAYを複製しているときにこのパラメーターが有効になっている場合、このパラメーターは無視されます。

デフォルト: 0

有効な値の範囲: 0（無効）または1（有効）

有効になるタイミング: SharePlexの再開時

sp_copパラメーター

これらのパラメーターは、SharePlex **sp_cop**プログラムによって使用されます。

SP_COP_IDLETIME

このパラメーターは、**sp_cop**が非アクティブである時間間隔を決定します。アイドル時間の目的は、マシンのリソースを節約することです。一般的に、**sp_cop**はレプリケーションの中心的役割を担っているため、長い時間アイドル状態にすべきではありません。**sp_cop**がオーバーヘッドアクティビティにCPU時間を使いすぎている場合、このパラメーターを使用してアイドル時間を増やすことができます。オーバーヘッドを減らすための推奨値は600秒です。

デフォルト: 60秒

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_COP_SCAN_TIMEOUT

このパラメーターは、プロセスを停止または再起動する必要があるかどうかを判断するために、sp_copがサービステーブルをスキャンする最初の時間間隔を設定します。

SharePlexの実行が継続すると、各プロセスの履歴に応じてこの間隔が変わる可能性があります。

デフォルト: 300秒

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_COP_SO_RCVBUF

このパラメーターは、ターゲットマシンのTCP/IPウィンドウサイズを調整します。ソースマシンのSP_XPT_SO_SNDBUFパラメーターと連動して、ネットワークを介して送信されるパケットのサイズを確立します。SharePlexは、sp_copとExportプロセスの開始時、TCP/IPソケットが作成されるときに両方のパラメーターを参照します。

SharePlexがWANを通じてレプリケートされ、exportキューが常に滞留している場合は、両方のパラメーターを調整してみてください。SP_COP_SO_RCVBUFは、1024バイトの倍数で、SP_XPT_SO_SNDBUFの値以上に設定しなければなりません。パラメーターのサイズを設定するには、ソースマシンとターゲットマシンの間のping時間を決定し、両方のパラメーターに以下の式を使用します。

```
param_value / ping_time= 1秒あたりのバイト数
```

例えば、ping時間が200ミリ秒、2つのパラメーターの値が64Kの場合、SharePlexは、毎秒5個の64Kパケットを送信し、合計で毎秒320Kを送信することになります。

帯域幅の問題が確認されない限り、Questでは、両方のパラメーターをデフォルト（システムの設定を使用する）のままにしておくことを推奨します。SP_COP_SO_RCVBUFを変更するには、これをターゲットシステム上で設定し、そのシステム上でsp_copを停止してから開始します。

それでも転送が遅い場合は、ターゲットシステムのSP_IMP_WCMT_MSGCNTパラメーターとSP_IMP_WCMT_TIMEOUTパラメーターを大きくしてください。SP_IMP_WCMT_MSGCNTを少なくとも10,000に設定します。

デフォルト: 0（デフォルトはオペレーティングシステムによって設定される）

有効な値の範囲: 1,024の倍数を使用したバイト単位の正の整数。最大値はオペレーティングシステムによって設定されます。

有効になるタイミング: SharePlexの再開時

SP_COP_TPORT

このパラメーターは、SharePlex sp_copプロセス間のTCP/IP通信に使用するTCP/IPポート番号を設定します。相互にレプリケートするすべての**sp_cop**プロセスは、同じTCP/IPポート番号を使用する必要があります。SP_COP_TPORTのデフォルト設定は2100ですが、インストール時に別のポート番号を設定することもできます。インストール後は、**set port**コマンドを使用して、別のポート番号を設定することができます。それ以降、**sp_cop**は新しいポート番号を使用します。

デフォルト: 2100

有効な値の範囲: 任意の正の整数

有効になるタイミング: SharePlexの再開時

SP_COP_UPORT

このパラメーターは、SharePlex sp_copプロセス間の通信に使用するUDPポート番号を設定します。相互にレプリケートするすべての**sp_cop**プロセスは、同じUDPポート番号を使用する必要があります。SP_COP_UPORTのデフォルト設定は2100ですが、インストール時に別のポート番号を設定することもできます。インストール後は、**set port**コマンドを使用して、別のポート番号を設定することができます。それ以降、**sp_cop**は新しいポート番号を使用します。

デフォルト: 2100

有効な値の範囲: 任意の正の整数

有効になるタイミング: SharePlexの再開時

Logパラメーター

これらのパラメーターは、SharePlexログのプロパティを制御するために使用されます。

SP_SLG_LOG_MAXSIZE

このパラメーターは、SharePlexのイベントおよびトレースログのサイズを制御します。

ファイルが最大サイズに達すると、ログに応じて以下のようになります。

- イベントログの名前が**event_log_***n*（*n*はファイルシーケンスでの次の番号）に変更され、アクティブなイベントログとして新しいファイルが作成されます。例えば、最初のイベントログが最大サイズに達すると、event_log_0000000001に名前が変更され、その後、新しいファイルがアクティブログとして開かれます。そのファイルが最大サイズに達すると、event_log_0000000002に名前が変更され、その後も同様になります。
- トレースログは切り捨てられます。このログに書き込む前に、SharePlexはそのサイズをチェックします。サイズがこのパラメーターの値より大きい場合、SharePlexはファイルを切り捨ててから書き込みます。

このパラメーターを大きくする前に、新しいファイルサイズを収容できる十分なディスク容量があることを確認してください。

デフォルト: 500 MB

有効な値の範囲: 1～2048 MB（1 MBきざみ）

有効になるタイミング: SharePlexの再開時

SNMPパラメーター

これらのパラメーターは、SharePlex SNMPサポートのプロパティを制御するために使用されます。

SP_SLG_SNMP_ACTIVE

このパラメーターは、SNMPサポートを有効または無効にするフラグです。

SharePlexは、SharePlexのレプリケーションでサポートされるすべてのUnixおよびLinuxプラットフォームでSNMP（Simple Network Management Protocol）エージェントをサポートします。

注意: SharePlexは、SNMPの**エージェント**のみをサポートします。SNMPトラップのみを送信し、トラップを傍受するためのSNMPシグナルデーモン（SNMPマネージャ）はSharePlex *ありません*。SNMP信号を管理するネットワーク管理ステーション（NMS）がある場合のみ、SharePlexのSNMP機能を使用してください。SharePlex SNMPエージェントは**snmptrap**という名前で、SharePlexと共にSharePlexの製品ディレクトリの**bin**サブディレクトリにインストールされます。このプログラムは実行しないでください。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（オン）

有効になるタイミング: SharePlexの再開時

SNMPエージェントパラメーター

以下のパラメーターは、NMSと通信するようにSNMPエージェントを設定します。SP_SLG_SNMP_ACTIVEパラメーターが有効になっている場合、各パラメーターには値が必要です。

パラメーター	値
SP_SLG_SNMP_HOST	トラップの送信先のシステム（ホスト）の名前
SP_SLG_SNMP_COMMUNITY	コミュニティのセキュリティ文字列
SP_SLG_SNMP_MJR_ERRNUM	トラップで使用される重大なエラーの番号
SP_SLG_SNMP_MNR_ERRNUM	トラップで使用される重大ではないエラー番号

カスタムMIBパラメーター

以下のパラメーターは、カスタムMIBに必要な情報を指定します。

パラメーター	値
SP_SLG_SNMP_ENTERPRISE_OID	トラップと共に送信するエンタープライズオブジェクト識別子。デフォルトは1.3.6.1.4.1.3.1.1です。
SP_SLG_SNMP_TRAP_OID	トラップにバインドされるカスタムオブジェクト識別子。デフォルトは1.3.6.1.2.1.1.0です。
SP_SLG_SNMP_TRAP_PROGRAM	トラッププログラムの名前。デフォルトはiwsnmptrapです。

SNMPトラップパラメーター

以下のパラメーターは、特定のレプリケーションイベントに関するトラップを送信するようにSNMPエージェントを設定します。イベントのメッセージまたはエラーテキストはトラップに含まれます。これは、イベントログに表示されるのと同じエラーです。

イベントのSNMPトラップを有効にするには、対応するパラメーターを値1に設定します。デフォルトでは、すべてのトラップは無効になっています（パラメーター値は0）。

パラメーター	SharePlex のイベント
SP_SLG_SNMP_INT_ERROR	SharePlex ロジックエラーとプロセスを終了させるエラー
SP_SLG_SNMP_SYS_ERROR	SharePlexで発生したシステム関連のエラー
SP_SLG_SNMP_ERROR	SharePlexのその他のエラー
SP_SLG_SNMP_OUT_OF_SYNC	レプリケーションが同期していない
SP_SLG_SNMP_STARTUP	SharePlex の起動
SP_SLG_SNMP_SHUTDOWN	SharePlex のシャットダウン
SP_SLG_SNMP_LAUNCH	SharePlexプロセスの開始
SP_SLG_SNMP_EXIT	SharePlexプロセスの停止

Systemパラメーター

これらのパラメーターは、システム関連のSharePlexプロパティを制御します。

SP_SYS_HOST_NAME

このパラメーターは、クラスタや代替ネットワークインターフェイスを持つシステムなど、複数のノードで1つのSharePlex変数データディレクトリを共有する環境で使用します。これは、いずれかのプロセスが名前検索を発行したときに、SharePlexに正しい論理ホスト名（グローバル・クラスタ・パッケージ名）を提供して、ローカルシステム名に置き換えます。このパラメーターで設定された名前により、SharePlexはフェールオーバー中に適切に移行できます。

- 設定をアクティベーションする前に、プライマリノードとすべてのセカンダリ（アダプティブ）ノードにSP_SYS_HOST_NAMEを設定し、フェールオーバーするパッケージに設定します。
- UnixおよびLinuxシステムでは、このパラメーターを**.profiles**ファイルに設定します。
- UnixおよびLinuxシステムでは、共有変数データディレクトリの**sp_ctrl**を通してこのパラメーターを設定します。**sp_ctrl**を通してSP_SYS_HOST_NAMEを設定すると、フェールオーバーの一部である変数データディレクトリに永続的に設定され、レプリケーションを開始する準備ができたときに、**sp_cop**の開始前にパッケージ名がエクスポートされるようになります。
- SP_SYS_HOST_NAMEは、クラスタ内のマシンにのみ設定します。

クラスタ化されたマシンに影響を与えるシステムから**sp_ctrl**コマンドを発行する場合は、[*on*host]オプションのホストとして**SP_SYS_HOST_NAME**で設定した名前を使用するか、hostコマンドを使用してsp_ctrlのデフォルトとして設定します。

クラスタ化されたシステムでSharePlexを設定する方法の詳細については、『SharePlexインストールおよびセットアップガイド』を参照してください。

デフォルト: なし

有効な値の範囲: パッケージ名の文字列

有効になるタイミング: SharePlexの再開時

SP_SYS_IN_SYNC

このパラメーターは、NULL値が許容されるキー列を持つテーブルをSharePlexがレプリケートする方法を制御します。

- このパラメーターがデフォルトの0（オフ）に設定されている場合、SharePlexは、列のいずれかにNULLが含まれているかどうかに関係なく、キーを使用してターゲットシステムで行を見つけます。このため、行のキーにNULLが含まれている場合、UPDATEおよびDELETEで非同期状態になる可能性があります。行の一意性は保証されず、SharePlexがターゲットシステムで間違った行を変更する可能性があります。
- このパラメーターを1（オン）に設定すると、SharePlexは行のキーにNULLがない場合のみ、そのキーを使用します。キーにNULL値が含まれている場合、SharePlexは行のすべての列をシミュレートされたキーとして使用し、正しいターゲット行を確実に見つられるようにします。

（NULLが宣言されていても）アプリケーションのキーにNULLが含まれないことが分かっている場合は、このパラメーターをオフにしてください。オンにすると、使用する追加ロジックによってオーバーヘッドが増加するからです。このパラメーターは、設定をアクティベーションする前に、ソースシステムで設定してください。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: CaptureとReadの再開時

SP_SYS_JOB_HISTORY_RETENTION

このパラメーターは、**job status**コマンドや**report**コマンドで表示するために、過去の**compare**、**compare using**、**repair**、**copy**、または**append**ジョブに関する情報を何日間保持するかを定義します。

デフォルト: 90日

有効な値の範囲: 0または任意の正の整数

有効になるタイミング: 即時

SP_SYS_JOB_HISTORY_SIZE

このパラメーターは、SharePlex変数データディレクトリのdataサブディレクトリに存在するジョブ・ステータス・データベースのサイズを制御します。指定された値によって、データベースに許容される最大サイズ（キロバイト（KB）単位）が決まります。データベースのサイズが指定された値を超えると、エントリは古いものから順に削除されます。デフォルト値は0で、無制限を指定します。

デフォルト: 0（無制限）

有効な値の範囲: 0または任意の正の整数

有効になるタイミング: 即時

SP_SYS_SUSPEND_ON_ERROR

このパラメーターは、システムエラーまたはSharePlex内部エラーが発生したときに、CaptureプロセスまたはPostプロセスを停止するかどうかを制御します。デフォルトの1は、エラー発生時にプロセスを停止させます。0を設定すると、プロセスは続行されますが、イベントログにメッセージが生成されます。

> **重要:** Captureを長時間停止すると、Oracleとのペースが崩れ、REDOログがラップする可能性があります。その場合、Captureはアーカイブログを読み込みますが、アーカイブログが利用できなくなった場合は、データを再同期する必要があります。

デフォルト: 1(エラー時に停止)

有効な値の範囲: 0または1(フラグ)

有効になるタイミング: 即時

SP_SYS_TARGET_COMPATIBILITY

このパラメーターは、SharePlexの異なるバージョン間の互換性を維持し、スムーズな移行とアップグレードを可能にします。SharePlexは新しいユーザ要件を満たすために進化し続けているため、新しいバージョンで追加された特徴や機能が、必ずしも以前のバージョンと互換性があるとは限りません。このパラメーターは、ソースシステムのSharePlexの上位バージョンからターゲットシステムの下位バージョンへのレプリケーションを可能にします。

デフォルトでは、このパラメーターはインストールされているSharePlexソフトウェアのバージョンに設定されています。

- ソースシステムの上位バージョンからターゲットの下位バージョンのSharePlexにレプリケートする場合、このパラメーターをターゲットのSharePlexのバージョンに最も近い値に設定します。例えば、SharePlexのバージョンがソースで8.6.3、ターゲットで8.6.2の場合、このパラメーターを8.6.2に設定します。
- 同一のバージョン間でレプリケートする場合、両方のシステムでこのパラメーターをデフォルトのままにしておきます。

デフォルト: インストールされているSharePlexソフトウェアのバージョン

有効な値の範囲: 6.0.0から最新バージョンまでのSharePlexリリースバージョンで、コンマ区切りの数値を3つまで指定できます(例: 8.6.3)。

有効になるタイミング: Captureの再開時

SP_SYS_VAR_FULL

このパラメーターは、SharePlex変数データディレクトリがインストールされているディスクの使用可能な領域のしきい値を設定します。これにより、キューが使用可能なディスク容量を超えないようにできます。Capture、Read、Importで処理中の操作によって、使用可能なディスク容量がこのパラメーターに設定された値を下回ると、プロセスが停止します。使用可能なディスク容量がSP_SYS_VAR_OKパラメーターで設定されたしきい値に達すると、処理が再開されます。

デフォルト: 30 MB

有効な値の範囲: SP_SYS_VAR_OKの値より大きい任意の正の整数

有効になるタイミング: 即時

SP_SYS_VAR_OK

このパラメーターは、SP_SYS_VAR_FULLの値に達したために停止したCapture、Read 、またはImportの処理を再開できる使用可能なディスク容量を設定します。

デフォルト: 50 MB

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

Compare/Repairパラメーター

これらのパラメーターは、SharePlex**compare**コマンドのプロパティを制御します。

SP_DEQ_IGNORE_ORACLE_ERROR_NUM

このパラメーターでは、無視するOracleエラー番号を指定することができます。指定されたエラー番号は500回まで無視されます。このパラメーターは主にデバッグ用で、**repair**、**repair1**オプションと併用してはなりません。

デフォルト: 0（なし）

有効な値の範囲: 任意の有効なOracleエラー番号（1406など）

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_IP_MAPPING

このパラメーターは、ソースシステムとターゲットシステム間でネットワークアドレス変換（NAT）が実装されている場合に、IPアドレスをマッピングします。NATが実装されている場合、ソースマシンのターゲットIPアドレス（192.168.32.10など）とターゲットマシンのIPアドレス（213.18.123.103など）が異なることがあり、その結果、**compare**および**repair**コマンドの発行時に混乱を招くことがあります。ターゲットマシンでSP_DEQ_IP_MAPPINGパラメーターを設定すると、IPアドレス192.168.32.10はIPアドレス213.18.123.103と等価とみなされるため、Postがcompareメッセージを処理し、compare **sp_declt** クライアントプロセスがCompareプロセスを終了することができます。

ターゲットシステム上でSP_DEQ_IP_MAPPINGパラメーターを以下の例のように設定します。

```
sp_ctrl> set param SP_DEQ_IP_MAPPING "192.168.32.10:213.18.123.103"
```

注意: ソースマシンとターゲットマシンには適切なIPアドレスを使用します。上記のIPアドレスは一例です。

さらに、SP_DEQ_IP_MAPPINGパラメーターでは、以下の例のように、カンマで区切った複数のマッピングを使用することができます。

```
sp_ctrl> set param SP_DEQ_IP_MAPPING "192.168.32.10:213.18.123.103,10.2.3.4:50.64.70.88"
```

デフォルト: なし

有効な値の範囲: カンマで区切られた1つ以上の有効なIPアドレス

有効になるタイミング: 次のcompareコマンド発行時

SP_DEQ_LOG_FILESIZE

このパラメーターは、compareサーバとcompareクライアントが生成するSQLログファイルのサイズを制御します。

デフォルト: 50 MB

有効な値の範囲: 0を超える任意の値

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_MALLOC

このパラメーターはフェッチのバッチサイズを制御します。バッチサイズは、SharePlexがcompareのために一度に選択する行数を制御します。バッチサイズが大きいほど処理速度は向上しますが、より多くのメモリを必要とします。この値は、使用するcompareスレッドの数で等分され、次に、すべての列のサイズを足した値に基づいてバッチサイズが再計算されます。

デフォルト: 500（MB）

有効な値の範囲: 50～32,767（MB）

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_PARALLELISM

このパラメーターは、select文のDegree of Parallelismのヒントを管理します。

SP_DEQ_PARRALLISMがゼロに設定されている場合、並列ヒントは使用されません。ユーザがコマンドラインから提供したヒントの方が優先されます。

デフォルト: 2

有効な値の範囲: 0～500

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_PARTIAL_REPAIR_CLR_OOS

このパラメーターを使用すると、WHERE句やOracleの[サブ]パーティション名で指定された行のサブセット（水平分割レプリケーションを含む）など、テーブルの行のサブセットのみのcompare・repair後に、repairプロセスでそのテーブルの**statusdb**から非同期メッセージを消去できるようになります。

重要! このパラメーターはターゲットシステムで設定します。ソースシステムには何の影響もありません。

このパラメーターは以下のように機能します。

- 値0（無効、デフォルト）: 行のサブセットがcompareされ、repairされた後、非同期メッセージが**statusdb**に保持されます。このメッセージは、選択基準に適合しなかった非同期行が存在する可能性があることをユーザに認識さ

せるために保持されます。

- 値1（有効）: 行のサブセットがcompareされ、repairされた後、非同期メッセージはクリアされます。このパラメーターは、サブセット内の行だけがテーブル内で同期していないことが分かっている場合にのみ有効にします。

注意: テーブル内の非同期行がすべてrepairされていることを確認するには、テーブル全体に対してcompareとrepairを実行してください。

デフォルト: 0（無効）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_PART_TABLE_UPDATE

このパラメーターは、repair usingコマンドとrepairコマンドをOracleパーティション化ターゲットテーブルに対して発行した場合の動作に影響します。

- このパラメーターをデフォルトの0に設定すると、パーティション化Oracleターゲットテーブルは INSERTとDELETEのみを使用してrepairされます。UPDATEによって行がパーティションを変更したときに、行の移動がテーブルで有効になっていない場合に発生するエラーを防ぐために、UPDATEを必要とするrepairはDELETEに変換され、その後にINSERTが続きます。
- このパラメーターを1に設定すると、パーティション化Oracleテーブルは、INSERT、UPDATE、およびDELETEを適切に使用してrepairされます。このモードは、UPDATEによってターゲットテーブルのパーティションが変更されないことがわかっている場合、またはターゲットテーブルで行の移動が有効になっている場合にのみ使用してください。

このパラメーターは**ターゲット**システムで設定します。

デフォルト: 0（UPDATEでrepairしない）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_READ_BUFFER_SIZE

このパラメーターは、compare/repair操作中のLOB列とLONG列にのみ適用されます。これらの列がcompare/repairされる際に、取得されたLOBまたはLONGデータを保持するバッファのサイズを調整します。パラメーターの値は、使用可能なシステムメモリとデータサイズに基づいて調整しなければなりません。

デフォルト: 1 MB

有効な値の範囲: 1～100の任意の整数（単位はMB）

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_ROW_LOCK_THRESHOLD

このパラメーターは、repair実行時にrepairプロセスがターゲットテーブル全体をロックするか、非同期行のみをロックするかを制御します。非同期行の数が指定された値より少ない場合、非同期行のみがrepairのためにロックされます。これにより、Postプロセスはテーブルの他の行へのポストを続けることができます。ソーステーブル上の非同期行の数がこの値より少ない場合、それらの行はロックされ、そうでない場合は、読み取り値の一貫性を確保するためにテーブルに対する短時間のロックが行われます。

デフォルト: 1,000の非同期行

有効な値の範囲: 任意の正の整数

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_SKIP_LOB

このパラメーターは、compare/repair処理にLOBが含まれるかどうかを決定します。

- このパラメーターをデフォルトの0に設定すると、処理時のcompareプロセスにLOBが含まれます。
- このパラメーターを1に設定すると、LOB以外の列のみがcompareされ、repairされます。一度挿入されたLOBが変更されない場合、このパラメーターを1に設定することで、処理を高速化することができます。

このパラメーターは**ソース**システムで設定します。

デフォルト: 0

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_THREADS

このパラメーターは、ソースシステム上でcompareコマンドが使用する処理スレッド数を制御します。各スレッドは、ターゲットマシン上に**sp_deqlt**プロセスを作成します。

デフォルトの2スレッドはUPマシンに最適なスレッド数であることが証明されていますが、最大15スレッドでパフォーマンスが向上する可能性があります。スレッドの数が多すぎると、特に大きなテーブルを同時にcompareすることになった場合に利点が減少します。

スレッド数の値は、compareするテーブルの数とは無関係で、SharePlexは、compareするテーブルの数を超えるスレッドを生成することはありません。マシンにプロセッサーが1つしかない場合は、このパラメーターを1に設定します。システムのプロセッサ数を超える値に設定しないでください。

デフォルト: 2スレッド

有効な値の範囲: 1～15

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_TIMEOUT

このパラメーターは、compareコマンドとrepairコマンドの接続タイムアウトを制御します。**sp_desvr**プロセスは、ターゲットシステム上の**sp_declt**プロセス（複数可）をインスタンス化するためにレプリケーションキューに依存しているため、キュー内のそれまでのメッセージがすべて処理されるまで**sp_declt**プロセスを開始できません。このパラメーターで指定されたタイムアウトよりも遅延が長い場合、**sp_desvr**は終了し、エラーを返します。キューが30分以上滞留していることが分かっている場合は、必要に応じてこのパラメーターを大きくすることができます。

デフォルト: 1,800秒（30分）

有効な値の範囲: 121秒以上

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_USE_SP_CKSUM

このパラメーターは、compareの際にOracle ora_hash()を使用するかどうかを決定します。

- このパラメーターをデフォルト値の0に設定すると、Oracle ora_hash()が使用されます。
- このパラメーターを1に設定すると、SharePlexの内部チェックサム方式が使用されます。

デフォルト: 0

有効な値の範囲: 0または1

有効になるタイミング: 次のcompareで即時利用可能

copy/appendコマンドのパラメーター

これらのパラメーターは、SharePlex **copy/append**コマンドに関連する**sp_sync**プロセスで使用されます。

SP_OSY_COMPRESSION

このパラメーターはデータ圧縮レベルを調整します。

デフォルト: 6

有効な値の範囲: 0～9（9は最高レベルの圧縮、0は圧縮なし）

有効になるタイミング: 次の同期で即時利用可能

SP_OSY_LOCK_TIMEOUT

このパラメーターは、同期プロセスがテーブルロックを取得するまで待機する時間を秒数で設定します。

デフォルト: 2

有効な値の範囲: 0～900

有効になるタイミング: 次の同期で即時利用可能

SP_OSY_POST_TIMEOUT

このパラメーターは、同期プロセスが、Postプロセスの準備が整い、同期が開始されるまで待機する時間を秒数で設定します。

デフォルト: 1,800

有効な値の範囲: 0～36,000（秒）

有効になるタイミング: 次の同期で即時利用可能

SP_OSY_THREADS

このパラメーターは、ソースシステムとターゲットシステム間の同期処理スレッド（export/import）のペア数を設定します。このパラメーターは、SharePlexの同期プロセス（copy/appendコマンドなど）で使用されます。

デフォルト: 5

有効な値の範囲: 1～32

有効になるタイミング: 次の同期で即時利用可能

analyze configコマンドのパラメーター

これらのパラメーターは、SharePlex**analyze config**コマンドのプロパティを制御します。

SP_ANL_CHECKPOINT_FREQ

このパラメーターは、**analyze config**コマンドプロセスによって発行されるチェックポイントの頻度を定義するために使用できるパラメーターの1つです。チェックポイントごとにプロセスはその状態をディスクに保存し、分析が終了する前にプロセスが停止したり、失敗したりした場合でも、迅速に復旧できるようにします。

デフォルト: 20,000メッセージ

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_ANL_CHECKPOINT_TIME

このパラメーターは、**analyze config**コマンドプロセスによって発行されるチェックポイントの頻度を定義するために使用できるパラメーターの1つです。チェックポイントごとにプロセスはその状態をディスクに保存し、分析が終了する前にプロセスが停止したり、失敗したりした場合でも、迅速に復旧できるようにします。

デフォルト: 120秒

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_ANL_RUN_TIME

このパラメーターは、**analyze config**コマンドの実行時間を制御します。コマンドの実行時間によって、分析されるトランザクション活動の量が決まります。このパラメーターはデフォルトを設定し、実行時に*n* {**minutes** |**hours** |**days**} オプションでオーバーライドができます。

デフォルト: 4,320分

有効な値の範囲: 1～432,000分

有効になるタイミング: 即時

非推奨のパラメーター

このセクションでは、非推奨となったSharePlexパラメーターについて説明します。

SharePlexが非推奨のパラメーターを処理する方法

アップグレード後も、使用中の環境で非推奨パラメーターが設定されていれば、SharePlexはその機能を引き続きサポートします。アップグレードを完了するためにパラメーターの交換やその他のアクションが必要な場合は、イベントログで通知されます。非推奨パラメーターの使用を停止するには、**sp_ctrl**で**reset param**コマンドを発行します。

非推奨のCaptureパラメーター

パラメーター	非推奨バージョン	注意
SP_OCT_ASM_CONNECT_DESC	8.6	データベースのセットアップが強化されたため
SP_OCT_ASM_SID	8.6	データベースのセットアップが強化されたため
SP_OCT_ASM_USE_OCI	9.1	SP_OCT_OLOG_USE_OCIに置き換わったため
SP_OCT_OLOG_THREAD	8.0.7	
SP_OCT_REPLICATE_DR_DDL	8.6.6	SP_OCT_REPLICATE_TRIGGERに名称が変更されました。後方互換性のため、古いパラメーターもまだ利用できます。
SP_OCT_TARGET_COMPATIBILITY	9.0	SP_SYS_TARGET_COMPATIBILITYに名称が変更されました。

非推奨のReadパラメーター

パラメーター	非推奨バージョン	注意
SP_ORD_NLS_LANG	8.0	
SP_ORD_LOGIN_O	8.6	データベースのセットアップが強化されたため
SP_ORD_OWNER_O	8.6	データベースのセットアップが強化されたため
SP_ORD_SKIP_OBJECT	8.0.7	この機能は、ルーティングマップの「**c.**」の指定を使用することにより、ソース設定ファイルで設定するようになりました。この機能の詳細については、『SharePlex管理ガイド』を参照してください。

非推奨のOracle Postパラメーター

パラメーター	非推奨バージョン	注意
SP_OPO_CHARACTERSET	8.0	
SP_OPO_CONF_LOG	8.6.6	SP_OPO_LOG_CONFLICTに名称が変更されました。後方互換性のため、古いパラメーターもまだ利用できます。
SP_OPO_GENERIC_CR	8.0.7	
SP_OPO_NCHAR_CHARACTERSET	8.0	
SP_OPO_NLS_LANG	8.0	
SP_OPO_RESTRICT_ALL_REF_CONS	8.6.2	
SP_OPO_TRACK_CHANGES	8.0.1	
SP_OPO_TYPE	8.6.3	

非推奨のOpen Target Postパラメーター

パラメーター	非推奨バージョン	注意
SP_OPX_CHECKPOINT_FREQ	8.6.4	
SP_OPX_MAX_CDA	9.1	現在は**target**コマンドで処理されます。

非推奨のExportパラメーター

パラメーター	非推奨バージョン	注意
SP_XPT_ALTERNATE_HOST	9.0	マシン名またはIPアドレスが変更された場合は、SharePlexのアーティファクトを変更するためのプロビジョニングを実行します。

非推奨のCompareパラメーター

パラメーター	非推奨バージョン	注意
SP_DEQ_CHARACTERSET	8.0	
SP_DEQ_BATCHSIZE	8.5	
SP_DEQ_CONFIG_LINE_SIZE	8.5	

パラメーター	非推奨バージョン	注意
SP_DEQ_LOG_ROW_DATA	8.6.3	
SP_DEQ_LOG_ROWID	8.0.7	
SP_DEQ_NCHAR_CHARACTERSET	8.0	
SP_DEQ_NLS_LANG	8.0	
SP_DEQ_NLS_LANG_SELECTOR	8.0	
SP_DEQ_ORACLE_LOGOFF	8.0.7	
SP_DEQ_RESULTS_RETENTION	8.0.7	
SP_DEQ_REUSE_CONFIG	8.0.7	

非推奨のqueueパラメーター

パラメーター	非推奨バージョン	注意
SP_QUE_SHMSIZE	8.0.7	

その他の非推奨パラメーター

パラメーター	非推奨バージョン	注意
SP_CFP_DEBUG	8.0.7	
SP_COP_WALLET_PATH	8.6.3	

3

PostgreSQL用SharePlexコマンド

SharePlexコマンドは、レプリケーションプロセスの設定、開始、停止、制御および監視を行います。SharePlexコマンドは、`sp_ctrl`インターフェイスを通じて発行されます。

SharePlexコマンド一覧

このセクションでは、制御対象のプロセスまたは機能に従ってグループ化されたすべてのSharePlexコマンドの概要を示します。

SharePlexコマンドの権限レベルとセキュリティグループの詳細については、『SharePlex管理ガイド』を参照してください。

設定コマンド

設定コマンドは、SharePlex設定ファイルの管理全般を制御します。

設定コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
abort config	1	すべて	データがポストされたかどうかにかかわらず、指定された設定ファイルのレプリケーションを直ちに終了します。設定を無効化し、キューとプロセスを削除します。
activate config	1	すべて	設定で指定されたデータのレプリケーションを開始します。
copy config	2	すべて	設定を複製し、新しいファイルとして編集して保存します。
create config	2	すべて	レプリケートするオブジェクトに関する情報を入力する新しい設定ファイルを作成します。
deactivate config	1	すべて	キューが削除される前にキュー内のデータがターゲットデータベースにポストされるように、設定で指定されたデータのレプリケーションを安全に終了します。
edit config	2	すべて	デフォルトのテキストエディタを使用して、既存の設定ファイルを変更します。
list config	3	すべて	ソースシステム上のすべての設定ファイルを、アクティブと非アクティブの両方で表示します。
purge config	1	すべて	アクティブな設定によって生成されたキューからデー

コマンド	認証レベル	サポート対象のターゲット	説明
			タを削除しますが、設定を非アクティブにしたりキュー自体を削除したりはしません。
rename config	2	すべて	設定ファイルに別の名前を割り当てます。
verify config	3	すべて	設定内のオブジェクトが有効であること、および設定が適切な構造になっていることを検証し、アクティベーションとレプリケーションが正常に行われるようにします。

情報コマンド

情報コマンドはレプリケーション環境に関する情報を提供します。これらのコマンドは、レプリケーションの問題解決を試みるとき、または特定のプロセス条件を表示するときに使用します。

情報コマンド一覧

コマンド	認証レベル	説明
show capture	3	チューニングや問題解決に使用するために、Captureプロセスの簡単な、または詳細な統計情報を表示します。
show post	3	チューニングや問題解決に使用するために、Postプロセスの簡単な、または詳細な統計情報を表示します。
job status	3	**compare**および**repair**コマンドの現在のステータスと履歴を表示します。
compare status	3	**compare using**および**compare**コマンドのステータスと結果を表示します。
repair status	2	**repair**および**repair using**コマンドのステータスと結果を表示します。

パーティション化コマンド

パーティション化コマンドは、水平分割レプリケーションに使用されるパーティションスキームと行パーティションを作成、管理、表示します。

パーティション化コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
add partition	2	すべて	パーティションスキームと行パーティションを作成します。
drop partition	2	すべて	パーティションスキームから行パーティションを削除します。
drop partition scheme	2	すべて	パーティションスキームを削除します。
modify partition	2	すべて	パーティションスキームの行パーティションを変更し

コマンド	認証レベル	サポート対象のターゲット	説明
			ます。
view partitions	3	すべて	既存のパーティションスキームとその行パーティション仕様を表示します。

同期コマンド

同期コマンドは、データ同期の側面を制御します。

同期コマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
compare / compare using	2	PostgreSQL	ソーステーブルとターゲットテーブルの内容をcompareし、同期を検証します。
cancel	2	PostgreSQL	実行中の**compare**または**repair**コマンドジョブをキャンセルします。
repair / repair using	2	PostgreSQL	ターゲットテーブル内の同期していない行をrepairし、ソースとターゲットを同期状態に戻します。
flush	2	すべて	このコマンドはデータストリームにマーカーを置いて、ソースデータとターゲットデータ間の同期ポイントを確立してから、その時点のPostプロセスを停止します。
reconcile	1	すべて	ターゲットシステムに適用されたバックアップまたはコピーの結果を、ソースシステムのユーザトランザクションと同期します。

メンテナンスコマンド

メンテナンスコマンドは、コマンドの表示とファイルを制御します。

メンテナンスコマンド一覧

コマンド	認証レベル	サポート対象のターゲット	説明
Clear History	2	PostgreSQL	**compare**、**compare using**および**repair**コマンドの実行に関するジョブ情報とソースログを削除します。
Clear Status	2	PostgreSQL	ステータスデータベースから警告メッセージを削除します。

PostgreSQL向けSharePlexコマンドのアルファベット順リファレンス

ここから、SharePlexコマンドの詳細なリファレンスドキュメントをアルファベット順に記載します。

PostgreSQLの設定の中断

abort configコマンドを使用すると、設定を即座に非アクティベーションし、関連するすべてのキューとレプリケーションプロセスを同時に削除できます。 このコマンドは、キュー内のデータがポストされたかどうかにかかわらず、ソースシステムとターゲットシステムで設定されたデータソースのすべてのレプリケーションアクティビティを停止します。これは強制的な非アクティベーションです。

abort configコマンドは、システム、設定、またはデータに問題が発生し、ターゲットシステムへのポストを防ぐ必要がある場合に有効です。パイプラインに残っているデータがすべて削除されるため、このコマンドが有効になると、ソースインスタンスとターゲットインスタンスは同期しなくなります。

abort configコマンドの実行前または実行中に、SharePlexレプリケーションプロセスが停止した場合、このコマンドも停止します。プロセスが再開されると、コマンドは動作を再開します。これにより、**abort config**はネットワークが一時的にダウンした場合でも機能し、接続が回復するまでキューに残ります。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ソースシステムおよびターゲットシステム
関連コマンド:	**deactivate config**、**purge config**

構文

基本コマンド
abort config *filename*

構文の説明

コンポーネント	説明
filename	中断する設定の名前。設定名では大文字と小文字が区別されます。

コンポーネント	説明
	例: `sp_ctrl(sysA)>`**abort config sales**

PostgreSQLの設定のアクティベーション

設定をアクティベーションするには、**activate config**コマンドを使用します。レプリケーションは、アクティベーションプロセスが完了すると直ちに開始されます。

アクティベーションプロセスは、設定ファイルを読み込み、そこからSharePlexが以下を実行するために必要なすべての情報を取得します。

- レプリケーション対象のオブジェクトを特定する
- レプリケートされた変更を適切なソースおよびターゲットデータベースにルーティングする
- Postがターゲットに変更を適用するために使用するSQLを生成する
- レプリケーションに追加されたすべてのテーブルをアクティベーションする

sp_copが設定をアクティベーションするために呼び出すプロセスは**sp_tconf**です。

注意:

- 論理レプリケーションを使用した設定のアクティベーション中に、コミットされていないトランザクションがPostgreSQLインスタンス上で実行されている場合、設定のアクティベーションに遅延が発生する可能性があります。
- パーティション化されたテーブルの場合、すべてのサブパーティションについてレプリカIDをfullに設定する必要があります。fullに設定されていない場合、「Object may not be replicated because replica identity is not full for its partitions(レプリカIDがそのパーティションについてfullでないため、オブジェクトがレプリケートされない可能性があります)」というメッセージが表示されます。アクティベーションしても、パーティション化されたテーブルはレプリケーションに追加されません。レプリカIDがfullに設定されていないすべてのサブパーティションテーブル名がアクティベーションログに記録されます。

アクティベーションのガイドライン

- 設定をアクティベーションするには、レプリケートするオブジェクトを含むデータベースをマウントし、開いておく必要があります。アクティベーションにかかる時間は、設定されたオブジェクトのサイズ、数、構造によって異なります。
- 各システム上のPostgreSQLサーバごとに1つの設定をアクティベーションできます。例えば、インスタンス`dbname1`にConfigA、ConfigB、ConfigCがある場合、一度にアクティベーションできるのはそのうちの1つだけです。同じデータソースについて別の設定をアクティベーションすると、最初の設定は自動的に非アクティベーションされます。
- 設定をアクティベーションする前に、**verify config**コマンドを使用して、アクティベーションとレプリケーションを正常に実行するための基本要件が満たされていることを確認します。このコマンドは、アクティベーションの失敗を引き起こす可能性のある潜在的な問題を警告します。

アクティベーションのステータスと結果の表示

SharePlexはオブジェクトを、設定ファイルの順番ではなく、オブジェクトIDに従ってアクティベーションするので、アクティベーションの順番を予測する方法はありません。

アクティベーションの結果を表示するには、**show config**コマンドを実行します。

アクティベーションに失敗した場合の対処法

テーブルや設定全体のアクティベーションに失敗する原因は多数あります。例えば、設定ファイルの1つ以上のコンポーネントが誤って入力された場合、影響を受けるオブジェクトのアクティベーションは失敗します。

アクティベーションの前に**verify config**コマンドを実行しなかった場合は、今すぐ実行し、検出された問題を修正してください。その後、もう一度アクティベーションを実行してください。

用途

サポート対象のソース:	PostgreSQL（pglsnオプションは物理レプリケーションにのみ適用可）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
認証レベル:	管理者（1）
発行対象:	ソースシステム
関連コマンド:	**abort config**、**copy config**、**create config**、**deactivate config**、**edit config**、**list config**、**purge config**、**remove config**、**rename config**、**show config**、**verify config**、**view config**

構文

基本コマンド	コマンドオプション
activate config *filename*	`pglsn=lsn_value`

構文の説明

コンポーネント	説明
filename	必須。アクティベーションする設定の名前。設定名では大文字と小文字が区別されます。 **例:** `sp_ctrl(sysA)>`**activate config sales**
pglsn=lsn_number	WALファイルの特定のLSNでレプリケーションを開始する設定をアクティベーションするには、このオプションを使用します。設定をアクティベーションする前に、以下を実行してください。 以前にアクティブな設定があった場合は、ソースとすべてのターゲットでpg_cleanspユーティリティを実行して、環境をクリーンな状態に戻します。詳細については、「pg_cleansp」を参照してください。 `show last_posted`コマンドを使用して、すべてのPostプロセスから最後にポストされたトランザクションのLSNを取得します（名前付きキューを使用している場合）。これらのLSN値のうち最小値をactivate configに使用します。

コンポーネント	説明
	例: `sp_ctrl> activate config myconfig pglsn=6/555FAE0` **注意:** LSNによるアクティベーションは、論理レプリケーションではサポートされていません。データベースが非同期になった場合は、外部ユーティリティを使用して手動でデータを同期する必要があります。

注意: アクティベーションに成功すると、物理スロットと論理スロットがデータベースに作成されます。

重要!

進行中のレプリケーションの一部であるalter table add/drop column操作を実行する際の手順。

警告:

- テーブルの変更中にCaptureプロセスを停止しないでください。
- 列が追加または削除されるテーブルではDML操作を実行してはなりません。

以下の手順に従います。

1. テーブルにDMLを実行することなく、alter table add/drop列にクエリを実行します。
2. 設定ファイルを再度アクティベーションして、最新の詳細がすべてのプロセスのオブジェクトキャッシュに取得され、保存されるようにします。
3. 再アクティベーションが完了すると、テーブルに対してDML操作を実行できるようになります。

進行中のレプリケーションの一部であるalter table add/drop partition操作を実行する際の手順。

警告:

- テーブルの変更中にCaptureプロセスを停止しないでください(再アクティベーションを行う場合のみ適用可)。
- パーティションが追加または削除されるテーブルでは、DML操作を実行してはなりません。

以下の手順に従います。

1. テーブルにDMLを実行することなく、alter table add/drop partitionにクエリを実行します。
2. 設定ファイルを再度アクティベーションするか、Captureプロセスを再起動して、最新のパーティション詳細が取得され、Captureプロセスで使用されるようにします。
3. Captureプロセスの再アクティベーションまたは再起動が完了すると、テーブルに対してDML操作を実行できるようになります。

PostgreSQL用パーティションの追加

add partitionコマンドを使用すると、水平分割レプリケーションを設定するときに、パーティションスキームに行パーティションを追加できます。作成する行パーティションごとに、**add partition**を発行します。

このコマンドは、以下の必須コンポーネントを含む、パーティション作成に必要なすべての情報を取得します。

- パーティションスキーム名。新しいパーティションスキームを作成するには、そのスキームの最初の行パーティションを作成する**add partition**コマンドでスキームの名前を指定します。SharePlexは自動的にパーティションスキームを作成します。その後、そのパーティションスキームに行パーティションを追加するときに、その名前を指定します。
- 行パーティションで指定された行のルーティング。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行場所:	ソースシステム
関連コマンド:	modify partition、drop partition、drop partition scheme、view partitions

構文

基本コマンド	コマンドオプション
add partition to*scheme_name* **set** {**condition =** *column_condition* | **route =** *routing_map*	[**and name =***partition_name*] [**and tablename =***target_table*] [**and description =***description*]

構文の説明

コンポーネント	説明
to *scheme_name*	**to**は、行パーティションが*scheme_name*に追加されることを示す必須キーワードです。 *scheme_name*はパーティションスキームの名前です。パーティションスキームは、最初に発行する**add partition**コマンドによって作成されます。このコマンドは、パーティション化する最初の行のセットも指定します。 水平分割を多用する場合は、パーティションスキームの命名規則を確立しておくことをお勧めします。
set	行パーティションの定義を開始する必須キーワード。
condition = *column_condition*	列条件に基づいて行パーティションを作成します。条件は引用符で囲む必要があります。**((region_id = West) and region_id is not null)**のような標準的なWHERE条件構文を使用します。
route = *routing_map*	このパーティションのルート。これは以下のいずれかになります。 **列条件に基づくパーティション:** 標準のSharePlexルーティングマップを指定します。例: **sysB@r.dbname**、**sysB:q1@r.dbname**または**sysB@r.dbname+sysC@r.dbname**（複合ルーティングマップ） **異なる名前を持つ複数のターゲットテーブルにパーティションをルーティングするには、以下の手順を実行します。** • 異なるターゲット名ごとに**add partition**コマンドを個別に発行します。**tablename**オプションを使って名前を指定します。 • 設定ファイルで、このパーティションスキームを使用するエントリのターゲットテーブルとして、これらのターゲットテーブルのいずれかを指定します。SharePlexは、設定が有効化されると他の名前を検出します。 • SP_ORD_FIRST_FINDパラメーターを0に設定し、パーティションスキームのすべての列条件をSharePlexでチェックするようにします。デフォルトでは、SharePlexは、特定の行変更がパーティションスキームの1つの列条件のみを満たすことを想定しています。
name = *name*	（推奨）このパーティションの短い名前。このオプションは、列条件に基づくパーティションにのみ有効です。名前を付けておくと、将来パーティションを変更したり削除したりする必要が生じたときに、長い列の条件を入力する必要がなくなります。
tablename = *owner.table*	（オプション）このオプションは、ターゲットテーブルが複数あり、そのうちの1つ以上が異なる名前である場合に使用します。名前ごとに**add partition**コマンドを個別に発行します。 テーブル名は完全修飾名でなければなりません。大文字と小文字が区別される場合は、名前を引用符で囲んで指定する必要があります。 **例:**

コンポーネント	説明
	add partition to scheme1 set name = p1 and condition = "C1 > 200" and route = sysb:p1@r.dbname and tablename = myschema.mytable
description = *description*	（オプション）このパーティションの説明。

例

列の条件に基づく行のパーティション

異なる行のセットを異なるpostキューにルーティング:

```
sp_ctrl> add partition to scheme1 set name = q1 and condition = "C1 >= 200" and route = sysb:q1@r.dbname
```

```
sp_ctrl> add partition to scheme1 set name = q2 and condition = "C1 < 200" and route = sysb:q2@r.dbname
```

異なる行のセットを、異なるターゲットシステムおよびソースと異なるテーブル名にルーティング:

```
sp_ctrl> add partition to scheme1 set name = east and condition = "area = east" and route = sys1e@r.dbname and tablename = schema1.targ
```

```
sp_ctrl> add partition to scheme1 set name = west and condition = "area = west" and route = sys2w@r.dbname and tablename = schema2.targ
```

PostgreSQLのcancelコマンド

cancelコマンドは、実行中の**compare**および**repair**コマンドジョブをキャンセルするために使用します。

ジョブをキャンセルするには、キャンセルするジョブのジョブIDを指定する必要があります。ジョブIDは、compareおよびrepairジョブを開始するコマンドを発行したときに、そのジョブから返されます。

```
sp_ctrl>repair using 1elliot
repairing 7 of 7 objects
repair started; job id 408
```

または、**job status**コマンドを発行することで、直近にストリームされたジョブのジョブIDを取得することもできます。

```
sp_ctrl>job status
Job ID: 408
PID: 11527
Host: prodsys
Started: 22-FEB-15 18:08:09
Job Type: Repair
Status: Processing - 0 objects completed
```

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**compare**、**repair**

構文

基本コマンド	リモートオプション
cancel *job_id*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
jobID	キャンセルするジョブのID。 **例**: `sp_ctrl>`**cancel 407**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

PostgreSQLのclear historyコマンド

clear historyコマンドは、compareおよびrepairコマンドの古いジョブから情報とログを削除するために使用します。

これらのジョブは、ソースシステムとターゲットシステムにログファイルを生成します。デフォルトでは、ジョブが**SP_SYS_JOB_HISTORY_RETENTION**パラメーターで設定された値より古くなると、ジョブ情報とログファイルが消去されます。**clear history**コマンドを使用すると、特定のジョブやテーブル、または特定の期間を経過したすべてのジョブについて、ジョブ情報とログをオンデマンドでクリアすることができます。

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	オペレーター（2）
発行場所:	ソースシステム
関連コマンド:	**compare**、**repair**

構文

基本コマンド	コマンドオプション	リモートオプション
clear history { **all** | *source_owner.source_table* | **age days** | *jobID* }	[**for r.***dbid*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	説明
all	すべてのジョブとログ履歴を削除します。 **例**: `sp_ctrl(sysA)>` **clear history all**
source_owner.source_table	特定のテーブルの履歴を削除します。履歴は**SP_SYS_JOB_HISTORY_RETENTION**パラメーターに従って削除されます。 **例**: `sp_ctrl(sysA)>` **clear history clear history user2.employee**
age days	指定した日数より古いジョブ履歴を削除します。

コンポーネント	説明
	例: `sp_ctrl(sysA)` > **clear history age 10**
jobID	指定されたジョブID（job statusコマンドで取得）のジョブを削除します。履歴は**SP_SYS_JOB_HISTORY_RETENTION**パラメーターに従って削除されます。 例: sp_ctrl(sysA)> **clear history jobID**
for r.*dbid*	オプション。アクティブな設定がない場合、または複数のアクティブな設定がある状況で**clear history**コマンドを使用するために使用できます。いずれの場合も、forオプションを使用してソースSIDを指定する必要があります。 **例**: `sp_ctrl(sysA)` >**clear history all for r.dbid**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)` >**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)` >**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)` >**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)` >**status on john/spot5489@SysA:8304**

PostgreSQLのclear statusコマンド

ステータスデータベースから古い警告メッセージを削除するには、**clear status**コマンドを使用します。

このコマンドを使用するには:

1. **show statusdb detail**コマンドを実行して、消去できるメッセージを見つけます。消去可能なメッセージは、**Clear**列に`Yes`が指定されています。
2. 一部のメッセージだけ消去したい場合は、消去する各メッセージのステータスIDをメモしておきます。
3. ステータスIDごとに**clear status**コマンドを発行するか、**all**引数を使用して、消去可能なすべてのメッセージを一度に削除します。

SharePlexは、消去されたメッセージを識別するメッセージをイベントログに書き込みます。

このコマンドは、デフォルトシステムのステータスデータベースからメッセージを消去します。別のシステムのステータスデータベースからメッセージを消去するには、[**on***host*]オプションを使用します。

注: すべてのメッセージを消去できるわけではありません。

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	オペレーター(2)
発行対象:	ソースまたはターゲットシステム
関連コマンド:	**show statusdb**

構文

基本コマンド	**リモートオプション**
clear status {*statusID*| **all**}	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

コンポーネント	**説明**
statusID	削除したい個々のメッセージのSharePlex割り当てID(show statusdbコマンドを使用して取得)。 **例**:

コンポーネント	説明
	`sp_ctrl(sysA)`**>clear status 20**
all	この引数を使用すると、消去可能なメッセージがすべて削除されます。 **例**: `sp_ctrl(sysA)`**>clear status all**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

PostgreSQLのcompare/compare using

compareおよび**compare using**コマンド（まとめて*compare*コマンドと呼びます）を使用して、ターゲットテーブルがソーステーブルと同期していることを確認します。

- **compare**コマンドは、ワイルドカードを使用して、個々のソーステーブルとターゲットテーブル、または指定されたスキーマ内の任意の数のテーブルをcompareします。個々のテーブルをcompareする場合は、列ベースのフィルタリングを使用して、compare用に選択される行を制御することができます。
- **compare using**コマンドは、アクティブな設定にリストされているすべてのテーブル、またはアクティブな設定内のテーブルのサブセットが含まれる別のファイルにリストされているすべてのテーブルをcompareします。

compareを実行した後、**repair**コマンドまたは**repair using**コマンドを実行し、同期していない行をrepairすることができます。

注意： 実行中のcompareまたはrepairは、ソーステーブルには一切影響を与えません。SharePlexは、読み取りの一貫性についてのクエリを実行するためだけにデータベースにログインし、ソーステーブルのロックは短時間です。SharePlexは、処理中にターゲットテーブルを短時間ロックしますが、ユーザはロックをほとんど意識することなくアクセスを継続できます。

サポートされている操作

SharePlexは、**DML**操作（INSERT、UPDATE、DELETE）によって発生した、ターゲットテーブル内の非同期行を検出し、repairすることができます。

サポートなし

SharePlexは、以下のcompareとrepairをサポートせずにスキップします。

- シーケンス
- トランスフォーメーションが使用されたテーブル
- ソースサーバとターゲットサーバ間のデータ受け渡しにパススルーサーバを使用するネットワーク構成
- カスケードレプリケーション環境では、compareとrepairはサポートされません。
- compareまたはrepair中のテーブルに対してDDLを実行しないでください。compareは、SharePlexがサポートするものも含め、DDL操作によって引き起こされる非同期状態を検出しません。DDLがテーブル定義を変更すると、compareする必要がある行を取得するためにcompareプロセスにより作成されたSELECTステートメントが無効になります。

  DDLによって発生した非同期状態を修正したら、**repair**コマンドを使用して行のデータを再同期することができます。
- 255文字を超えるcompareおよびrepairコマンド文字列はサポートされていません。これはオペレーティングシステムの制限です。この制限を回避するには、ソースシステムで**edit**コマンドを使用します。テキストファイル内にコマンド文字列を入力すると、コマンドが自動的にファイルを実行します。

compareおよび**compare using**でサポートされるデータ型の詳細については、『SharePlexリリースノート』を参照してください。

その他の条件

- compareまたはrepairするテーブルは、アクティブな設定ファイルの一部でなければなりません。
- ソーステーブル上にコミットされていないトランザクションがあると、読み取りの一貫性を得るためにcompareおよびrepair処理に必要な短時間のロックを取得できなくなります。compareやrepairを実行する前に、すべてのトランザクションがコミットされていることを確認してください。
- レプリケーションのレイテンシは、compareやrepairの処理のパフォーマンスを低下させる可能性があります。ターゲット上でcompareおよびrepairプロセスを生成するためにソースから送信されたメッセージは、複製されたデータと共にキューを通じて送信されます。データバックログによる遅延が生じると、生成メッセージの送信が遅れることにより、プロセスが読み取りの一貫性を失うことがあります。可能であれば、compareやrepairはピーク以外の時間帯に行います。
- compareまたはrepair中のテーブルをトランケートしないでください。compareコマンドは、開始時に各ソーステーブルのスナップショットを取ります。テーブルがトランケートされると、スナップショットのテーブルビューもトランケートされ、コマンドが無効な非同期状態を返すことがあります。
- ビューをrepairするには、以下が真でなければなりません。
  - ビューテーブルが、基礎となるテーブルのすべての制約列を含んでいること。
  - ビューの制約列の名前が、基礎となるテーブルの制約列の名前と同一であること。
- Compareプロセスのパフォーマンスを向上させるために、SharePlexはSP_DEQ_PARALLELISMパラメーターによる並列ヒントをサポートしています。このため、並列度、つまり起動するワーカープロセスの数を指定できます。

  デフォルトでは、PostgreSQLデータベースオプティマイザはSQLステートメントに最適な問い合わせ実行計画を選択します。これは、並列ヒントで指定した並列度に直接関連していないこともあります。pg_hint_planにより、クエリを実行するたびに配列ヒントを使用して実行計画を調整することができます。SharePlexは、データベースにpg_hint_plan拡張機能がインストールまたは設定されている場合、内部でこれを利用します。

  並列クエリを使用するための構文の例は以下の通りです。

  pg_hint_plan拡張機能は並列ヒントをサポートしています。`Parallel(table <# of workers> [soft|hard])`

  SharePlexはハードパースをサポートする並列ヒントを使用します。`Parallel(table <# of workers> [hard])`

  > **注意:** ワーカーの数は、データベースシステムのCPU VCoreより少なくなければなりません。

特別な使用例

以下のシナリオでは、compareを実行する際に特別な処理が必要となります。

使用例	サポートcompare
統合レプリケーション	統合レプリケーションは、ターゲットデータベースとPostプロセスがソースホストのIDを各行に追加するように設定されている場合にサポートされます。中央のターゲットテーブルの正しい行をcompareまたはrepairするには、**targetwhere**オプションを使用し、ソースID値を**where**句の基準にします。 例えば、ある企業の東部本社のデータベースのテーブルと、中央の企業データベースの正しい行を

使用例	サポートcompare
	compareするには、東部データベースのソースIDに「East」を使用し、その値を**targetwhere**句の基準にすることができます。**repair**コマンドで同じ**targetwhere**句を使用します。compareプロセスおよびrepairプロセスは、ソースID値を使用して、東部データベースに有効な行のみを選択します。 ソースIDを特定する統合レプリケーション以外の実装でcompareコマンドやrepairコマンドを使用すると、ターゲット行が不要に削除される可能性があります。この設定の詳細については、『SharePlex管理ガイド』を参照してください。 ターゲット行が正確に選択されるように、**targetwhere**オプションと標準の**where**オプションを組み合わせる必要があるかもしれません。
ピアツーピアレプリケーション	ピアツーピア構成では、どのシステムを*信頼できるソースシステム*とし、どのシステムを*セカンダリシステム*、つまりターゲットシステムとするかを決めなければなりません。セカンダリシステムは、repairが行われるシステムです。ピアツーピア環境でcompareやrepairを実行する前に、以下の手順に従ってください。 1. セカンダリシステムのユーザアクセスを停止し、そのシステムから複製された操作が信頼できるソースデータベースにポストされるのを待ちます。ユーザは引き続きソースデータベースにアクセスできます。 2. ソースおよびセカンダリシステムで**qstatus**コマンドを実行します。 3. キュー内のメッセージが10個以下であれば、ソースシステムからcompareを実行します。 4. compare中、**sp_xdesvr**および**sp_xdeclt**の開始後に、ソースおよびセカンダリデータベースへのユーザアクセスを許可できます。 5. **where**オプションを指定して**repair**コマンドを使用すると、ユーザをテーブルからロックアウトすることなく、ターゲットテーブルの選択した行をrepairできます。 この設定の詳細については、『SharePlex管理ガイド』を参照してください。
キーのないテーブル	compareおよびrepairコマンドは、ソースおよびターゲットシステム上でORDER BY句付きのSELECTステートメントを発行します。大規模なテーブルにプライマリキーまたはNULL以外の一意キーとインデックス（一意のインデックスが望ましい）を持つ場合、順序付けはより速くなります。それ以外の場合は、すべての列がキーとして使用されます。 一意の行識別子はないが、一意の行として識別できる列がテーブルに1つ以上ある場合、**orderby**オプションを指定して**compare**コマンドを使用できます。このオプションを使用すると、SharePlexは、ソースシステム上の**sp_xdesvr**ログに、コマンドがこれらの列をキーとして使用したという通知を出力します。 **制限事項**: SharePlexレプリケーションで、重複データを含む非キー列を含むテーブルを使用する場合、レプリケーションまたはcompare・repair処理でデータの不整合が発生する可能性があります。
余分なソース列またはターゲット列があるテーブル	ソーステーブルまたはターゲットテーブルに余分な列があり、それらの列にソートの基準となる一意の値が含まれている場合は、**sourcewhere**句または**targetwhere**句とともに**compare**コマンドを使用します。「compareする行の制御」を参照してください。

compareの仕組み

compareは、DML操作によって引き起こされる次のようなターゲットテーブルの非同期状態を検出します。

- 余分な行または欠けている行
- 値が一致しない行

ソースシステムで**compare**コマンドまたは**compare using**コマンドを実行すると、SharePlexは以下のイベントを開始します。

1. **sp_cop**プロセスは、ソースシステム上に**sp_xdesvr**（サーバ）プロセスを生成します。
2. **sp_desvr**プロセスは、**sp_ctrl**インターフェイスの制御と使用をユーザに返します。compareが進行している間、レプリケーションは続行されます。
3. **sp_xdesvr**プロセスは、ターゲットシステム上で**sp_xdeclt**（クライアント）プロセスを開始するために、Postプロセスにメッセージを送信します。
4. サーバプロセスとクライアントプロセスによって相互の通信が確立されます。
5. プロセスは、ソーステーブルとターゲットテーブルの行数をcompareすることにより、*サニティチェック*を実行します。ソーステーブルとターゲットテーブルの行数の差が20%より大きい場合、コマンドは終了します。サニティチェックは、ターゲットデータを再読み込みすることでより迅速に修正できるテーブルのcompareやrepairに無駄な時間をかけることを防ぎます。**copy**コマンドまたは**append**コマンドを使用してターゲットデータを再読み込みするか、別の方法を使用することができます。サニティチェックを上書きするには、overrideオプションを指定してcompareコマンドを再発行します。
6. サニティチェックをオーバーライドするには、**Override**オプションを指定して**Compare**コマンドを再発行します。
7. 行数がサニティチェックに合格すると、テーブルは以下のようにcompareされます。
    - **compare**コマンドを使用している場合、**sp_xdesvr**でソース行を選択し、**sp_xdeclt**でターゲット行を選択し、行をソートしてcompareします。
    - **compare using**コマンドを使用している場合、**sp_xdesvr**プロセスはターゲットシステム上に多数の処理スレッドを作成します。SP_DEQ_THREADSパラメーターによって設定される値は、作成されるスレッド数を制御します。各スレッドは**sp_xdeclt**（クライアント）プロセスを生成します。サーバプロセスとクライアントプロセスによって相互の通信が確立されます。処理負荷はクライアントプロセス間で分割されます。ソースとターゲットの各テーブルから行が選択され、ソートされ、compareされます。
8. 終了すると、プロセスによってログファイルが書き出され、**show compare**コマンドで結果を見ることができます。

SQLログファイルの管理

compareおよびrepairコマンドは、すべての非同期行をrepairするために必要なSQLを、ログファイルと同じ場所にあるSQLファイルに書き込みます。compareコマンドのみが発行された場合、SharePlexはこれらのSQLステートメントを実行しません。repairコマンドを実行した場合、このコマンドは、非同期行をrepairするSQLステートメントを実行すること以外はcompareコマンドと同じように動作します。

SQLログファイルの出力を抑制することができます。このファイルを抑制する理由には以下のようなものがあります。

- データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。
- compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。

SQLログファイルを抑制するには、**nosqllog**オプションを**compare**または**repair**コマンドで使用します。

SharePlexの現在のインスタンスの実行中に、compareとrepairのすべての実行についてSQLログファイルの出力を抑制するには、SP_SYS_SECURE_MODE環境変数を1に設定します。この変数はSharePlexを起動する前に設定する必

要があるため、**sp_cop**プロセスが実行されている場合は、この変数を設定した後に再起動する必要があります。この環境変数を指定して**sp_cop**を実行すると、compareコマンドとrepairコマンドによってSQLファイルにデータは保存されず、Post処理でもSharePlexのエラーログにデータは保存されません。

複数のプロセスの実行

すべてのcompareおよびrepairコマンドでは、複数のプロセスを同時に実行することができます。複数の**compare**および**repair**コマンドを同時に実行し、それぞれがソーステーブルとターゲットテーブルのペアを処理することも、PostgreSQLのワイルドカードを使用して1つのコマンドで複数のテーブルセットを指定することもできます。SharePlexによるワイルドカードのサポートの詳細については、『SharePlex管理ガイド』を参照してください。

レプリケーションプロセスとcompareおよびrepairプロセスなど、最大で20のSharePlexプロセスでpostキューを同時に使用することができます。一度に最大で5つのcompareおよびrepairプロセスを実行できるようにすることをお勧めします。

制限に達したためにcompareやrepairに失敗した場合、SharePlexはイベントログにメッセージを記録します。

注意: **edit**コマンドを使用し、以前のコマンドを編集して新しいコマンドを作成すれば、複数のコマンドをより簡単に実行できます。

設定のサブセットのcompare

アクティブな設定のサブセットは、以下の方法でcompareできます。

- 1つのスキーマに属するレプリケーション内の全テーブルをcompareするには、以下のように**compare**コマンドにワイルドカードを指定します。

  ```
  sp_ctrl> compare scott.%
  ```

- 設定ファイル内の全テーブルをcompareするには、**compare using**コマンドを使用します。

  ```
  sp_ctrl> compare using myconfig
  ```

- レプリケーション内の全テーブルを1つのターゲットルートとcompareするには、以下のように**at**オプションを指定して**compare using**コマンドを使用します。

  ```
  sp_ctrl> compare using config.active at prodsys@r.database_name
  ```

- 設定内のテーブルのカスタムサブセットをcompareするには、*compareファイル*で指定します。これはプレーン・テキスト・ファイルであり、compareしたいソーステーブルのみをリストします。ターゲットテーブルは、compare時に設定ファイルから取得されます。compareファイルは、**create config**コマンドまたは**copy config**コマンドを使用して作成できます。このファイルには、必ず設定ファイルでないことを明確に示す名前を付けてください。詳細については、コマンドの構文を参照してください。

compareする行の制御

compareおよび**repair**コマンドには、処理の対象として選択する行をフィルタリングできる**where**オプションがあります。デフォルトでは、これらのコマンドはテーブルのすべての行に影響し、ソーステーブルに含まれていないターゲットテーブルの列は無視されます。

- **where**オプションを使用すると、ソーステーブルとターゲットテーブルの同じ名前の列に基づいて行をフィルタリングすることができます。

- 垂直分割レプリケーションを使用するテーブルには**where**オプションを使用します。ソース列とターゲット列には異なる名前を付けることができます。**where**の選択をソース列に基づいて行います。SharePlexは、ターゲットテーブルの正しいWHERE句を構築するために、設定ファイルから列マッピングを読み取ります。
- ソーステーブルまたはターゲットテーブルに1つ以上の余分な列が存在し、それらの行に行の一意性を決定する値が含まれている場合は、**sourcewhere**および**targetwhere**オプションを使用します。
  - ソーステーブルに余分な列が含まれている場合は、**sourcewhere**を使用します。
  - ターゲットテーブルに余分な列が含まれている場合は、**targetwhere**を使用します。

  このオプションを正しく使用するには:
  - 余分な列には**sourcewhere**または**targetwhere**オプション*のみ*を使用します。
  - ソースとターゲットの両方で同じ名前を持つ他の列には、標準の**where**オプションを使用します。
  - SharePlexは、**where**オプションと**sourcewhere**または**targetwhere**オプションを組み合わせて、完全なWHERE句を作成します。

> **重要!** 余分な行を持つターゲットテーブルに対してcompareとrepairの両方を実行する予定がある場合は、UPDATEとDELETEのcompare用に**targetwhere**のみを使用してください。**repair**コマンドはINSERTの正しい値を決定できません。この問題を回避するには、余分な列にデフォルト値を設定するか、挿入された行を手動で更新します。

プロセスの特定

compareまたはrepairコマンドが発行されるたびに、ジョブIDが**sp_ctrl**ディスプレイに表示されます。**sp_ctrl**ディスプレイを利用できない場合は、**compare status**コマンドを実行することでジョブIDを表示できます。

sp_ctrlでステータスと結果を表示

compareのステータスや結果を表示するには、**sp_ctrl**の**compare status**コマンドを使用します。

- 基本コマンドは、最近開始されたcompareジョブの処理ステータスと、まだ実行中のその他のcompareジョブを表示します。
- 追加オプションを使用して、履歴のあるcompareジョブのステータス概要を表示したり、compareジョブの詳細情報を表示したりすることができます。

詳細については、compare status を参照してください。

警告とエラーの表示

sp_xdesvrおよび**sp_xdecIt**プロセスは、実行するシステム上にログファイルを書き込みます。SharePlexの変数データディレクトリの**log**サブディレクトリにあるファイルを開きます。

sp_xdesvrプロセスが書き込むログの名前は`xdesvr_<jobid>_r.<dbid>_p<process id>.log`です。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブIDです。
- *DBID*はPostgreSQLインスタンスのデータベースIDです。
- *ProcessID*は**sp_xdesvr**プロセスのプロセスIDです。

sp_xdecltプロセスが書き込むファイル名は、`xdeclt_<jobid>-<tableid>_r.<dbid>_p<processid>.log`
に.logまたは.sqlを付加したものです。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブのジョブIDです。
- *TableID*は、SharePlexによって割り当てられたジョブ内のテーブルのテーブルIDです。
- *DBID*はPostgreSQLインスタンスのデータベースIDです。
- *SourceHost*は、ソースホストの名前またはIPアドレスです。
- *ProcessID*は、**sp_decltプロセスのプロセスIDです。**

ログファイル名の例:

```
xdesvr_7_r.aparopka_p4970.log

xdeclt_7-1_r.aparopka_p25095.log

xdeclt_7-1_r.aparopka_p25095_01.sql
```

ディスク使用量を制御するために、ログは循環方式でエージングされます。現在のログがサイズの限界に達すると、SharePlexは新しいログファイルを生成します。新しいログは最大ログ数まで作成され、その後、SharePlexは最も古いログから上書きを開始します。

> **注意:** **compare usingコマンドの場合は、稀に複数のログファイルが存在します。**

compareジョブのキャンセル

実行中のcompareまたはrepairジョブを停止するには、**cancelコマンドを使用します。**

```
sp_ctrl(sysA)>cancel JOBID
```

詳細については、「cancel」を参照してください。

compare履歴とログの管理

SharePlex は、終了した各ジョブの履歴をソースシステムのデータベースに保持します。SP_SYS_JOB_HISTORY_RETENTIONパラメーターは、履歴の保持期間を制御します。

この履歴を必要に応じて消去するには、**clear historyコマンドを使用します。SharePlexがジョブの履歴を削除すると、**履歴の元となったログファイルも削除されます。

データベースのジョブ履歴を消去せずにソースシステムからログファイルを削除するには、**remove logコマンドを使用します。このコマンドは、ターゲットシステムから古いログファイルを削除するためにも使用できます。**

ログファイルのサイズを制御するには、SP_DEQ_LOG_FILESIZEパラメーターを設定します。

INSERT、UPDATE、DELETEの各操作に対して別々のSQLファイルを作成するには、すべての操作タイプに対して1つのファイルを使用するのではなく、**log splitオプションを使用します。**

バッチサイズのコントロール

プロセスがSELECTクエリを実行するときにフェッチされる行のブロックのサイズを制御することができます。ブロックサイズは、**SP_DEQ_MALLOCパラメーターで設定された値に基づいて計算されます。この値は、使用されるcompareスレッド**の数で等分され、その後、すべての列のサイズを足した値に基づいて再計算されます。

非同期行のrepair

compareによって検出された非同期行をrepairするには、**repair**コマンドまたは**repair using**コマンドを使用します。repair / repair usingを参照してください。

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	オペレーター（2）
発行場所:	ソースシステム
関連コマンド:	repair / repair using

構文

コマンド	コマンドオプション	リモートオプション
compare *owner.source_table[.partition]*	[**at** *target_host*@**r.***target_db*] | [**forr.***source_db*] | [{**include | exclude}** "*column_list*"] | [**key**] | [**log rowdata**] | [**log split**] | [{**nolocksource** | **nolocktarget**}] | [**nosqllog**] | [**not** "*exception_list*"] | [**orderby** "*column_list*"] | [**Override** parameter to the compare and repair command] | [**parallelism** *degree*] | [**quickcheck**] | [**sourcewhere** “*clause*”] | [**threads***thread_count*] | [**targetwhere** “*clause*”] | [**to***target_owner.target_table*[*.partition*]] |	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

コマンド	コマンドオプション	リモートオプション
	[**where** “*clause*”]	
compare using *filename*	[**key**] | [**log rowdata**] | [**log split**] | [**parallelism***degree*] | [**port** *port_number*] | [**quickcheck**] | [**threads***threads_count*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

必須のコマンドコンポーネント

コンポーネント	説明
compare *owner.source_table[.partition]*	基本コマンドは、すべてのソース行をすべてのターゲット行とcompareします。 *owner.source_table*は、ソーステーブルの所有者と名前です。大文字と小文字を区別したり、名前の中にスペースを入れたりするには、二重引用符を使います。例えば、**"HR".emp**。 ワイルドカードのテーブル名（オーナー名は不可）がサポートされています。compareするためには、このコマンドのワイルドカードを満たすテーブルがアクティブなレプリケーション設定に（明示的またはワイルドカードで）リストされていなければなりません。SharePlexがワイルドカードをどのように処理するかについては、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl(sysA)>`**compare scott.emp** `sp_ctrl(sysA)>`**compare scott.emp.west**
compare using *filename*	基本コマンドは、*filename*にリストされたテーブル内のすべてのソース行とすべてのターゲット行をcompareします。 *filename*は、compareするソーステーブルの名前が含まれているファイルの名前です。 **例** `sp_ctrl(sysA)>`**compare using sales**

オプションのコマンドコンポーネント

コンポーネント	説明
at *target_host@r.target_database_name*	**compareに有効** ソーステーブルをそのターゲットの1つとのみcompareします。ソーステーブルが複数のターゲットシステムにレプリケートされる場合に使用します。 *target_host*はターゲットシステムの名前です。 *target_db*はターゲットデータベース名の名前です。 **例** `sp_ctrl(SysA)>`**compare scott.emp at prod@r.database_name**
for r.*DBID*	**compareに有効** ソーステーブルを含むPostgreSQLインスタンスを指定します。システム上の複数のPostgreSQLインスタンスに同じソーステーブルがある場合に使用します。 *DBID*はPostgreSQLソースインスタンスのデータベース名です。大文字と小文字は区別され、**config**ファイルに表示される通りに入力する必要があります。 このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**compare scott.emp for r.database_name**
{**include** | **exclude**} "(*column_list*")	**compareに有効** compareする列をフィルタリングします。 • **include**を使って、compareしたい列を指定します。他の列はcompareされません。**include**句には、すべてのキー列を含める必要があります。 • **exclude**を使用して、**exclude**で指定した列を除くすべての列をcompareします。キー列を除外しないでください。 (*column_list*)は、インクルードまたは除外する列のリストです。 • それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。

コンポーネント	説明
	• このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションとともに任意の順序で記述できます。 **注意:** compareされなかった列には、同期していない行がまだ残っている可能性があります。 **例** `sp_ctrl (SysA)>`**compare scott.emp exclude "color, weight"**
key	**compareおよびcompare usingに有効** 大きなテーブルのcompareを高速で実行します。このコマンドはすべてのデータ値をcompareするわけではありません。以下のいずれか1つをcompareします。 • PRIMARYキー列またはNULLでないUNIQUEキー列のみ。 または... • **orderby**オプションで指定された列。テーブルにキーがない場合は、このオプションを使用します。 **重要:** キーまたは**orderby**の値が一致しても、他の列の値が一致しなければ、テーブルは同期しないままです。 このオプションを使用する場合は、必須コマンド引数の後に記述しなければなりません。他のオプションとともに任意の順序で表示できます。 このオプションは、*SharePlex*キー定義に基づくcompareには使用しないでください。SharePlexキー定義の詳細については、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl (SysA)>`**compare scott.emp key** `sp_ctrl(sysA)>`**compare using sales key**
log rowdata	**compareおよびcompare usingに有効** ハッシュ値ではなく、実際の行データを記録するSQLファイルを生成するようにクライアントプロセスに指示します。ファイルはターゲットシステムに作成されます。これらのデータ型の値が非常に大きい場合、SharePlexは実際のデータを記録できない可能性があります。 **重要!** このSQLファイルの目的は、compareされたデータのビュー

コンポーネント	説明
	を提供することです。ターゲットテーブルをrepairするために使用しないでください。コマンド実行時に取得されたデータ値は、現在データベースに登録されている値とは異なる可能性があります。非同期の行をrepairするには、**repair**コマンドを使用します。 **例** `sp_ctrl(sysA)>`**compare scott.emp log rowdata** `sp_ctrl(sysA)>`**compare using sales log rowdata**
log split	**compareおよびcompare usingに有効** SQLファイルを操作の種類に応じて3つの異なるファイル(INSERT用、UPDATE用、DELETE用)に分割するように、クライアントプロセスに指示します。 **例** `sp_ctrl(sysA)>`**compare scott.emp log split** `sp_ctrl(sysA)>`**compare using sales log split**
{**nolocktarget** | **nolocksource**}	**compareに有効** repairを含む実行のcompareフェーズ中に、compareプロセスがテーブルをロックするのを防ぎます。通常、SharePlexはcompare中に一時的にテーブルをロックして読み取り一貫性ビューを取得し、その後すぐにロックを解除しますが、repair中は、SharePlexは常にターゲットテーブルをロックします。SharePlexは、読み取り一貫性ビューを取得するために、repair中はソーステーブルを一時的にロックします。 **例** `sp_ctrl(sysA)>`**compare scott.emp nolocksource** `sp_ctrl(sysA)>`**compare scott.emp nolocktarget**
nosqllog	SQLログファイルの出力を抑制します。このファイルには、同期していない行をrepairするために必要なSQLが含まれています。このファイルを出力しない理由には、以下のようなものがあります。 • データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。 • compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。
not "*exception_list*"	**compareに有効**

コンポーネント	説明
	テーブル指定にワイルドカードが含まれている場合に、compareしないテーブルの例外リストを指定します。 "*exception_list*"は、compareしないテーブル名のリストです。 • *owner.tablename*形式を使用します。 • それぞれの名前はカンマで区切ります。リストにスペースは許容されません。 • リストを二重引用符で囲みます。 • テーブルを任意の順序でリストします。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl(SysA)>`**compare scott.% not (%temp%)**
orderby "*column_list*"	**compareに有効** compareプロセスでcompareする行をソートする際に、ORDERBY句で使用する列を指定します。このオプションを使用すると、プライマリキーや一意キーを持たないテーブルでcompareを実行できます。 "*column_list*"は、ORDERBY句で使用する列の名前です。 • それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションとともに任意の順序で記述できます。 **例** `sp_ctrl(SysA)>`**compare scott.emp where “file >001005” orderby “Last Name,Division”** オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列には、以下のようにエスケープされた二重引用符の余分なセットが必要です。 */productdir/***bin/sp_ctrl compare scott.emp orderby** “\“**Last Name**,**Division**\””
override	**compareおよびcompare usingに有効**

コンポーネント	説明
	サニティチェックをオーバーライドし、サニティチェックのしきい値を超えて、同期していないテーブルのcompareを行えるようにします。ターゲットテーブルを再同期するには、ターゲットデータベース/テーブルのインスタンス化に使用される初期プロセスに類似した代替的な方法を使用します。サニティチェックの詳細については、「compareの仕組み」を参照してください。 **例:** SP_ctrl (sysA) > **compare scott.emp override** SP_ctrl (sysA) > **compare using sales override**
quickcheck	**compareおよびcompare usingに有効** 同期していない行があるかどうかを確認する初期チェックを実行します。同期していない行が1つでも検出されると、プロセスは直ちに停止します。それ以上の情報は返されません。同期していない行の情報はcompareのSQLファイルに記録されません。 このオプションは、**key**オプションや、**where**オプションのいずれかと併用しないでください。 このオプションはLONG列をサポートしていません。 **例** `sp_ctrl(sysA)>`**compare scott.emp quickcheck** `sp_ctrl(sysA)>`**compare using sales quickcheck**
parallelism *degree*	**compareおよびcompare usingに有効** SELECTステートメントに並列ヒントを追加します。*degree*では、並列度を設定します。 **例** `sp_ctrl(sysA)>`**compare scott.emp parallelism 4** `sp_ctrl(sysA)>`**compare using sales parallelism 4**
sourcewhere “*clause*”	**compareに有効** ターゲットテーブルに列が存在しない場合に、ソーステーブル内の1つまたは複数の列を基にcompareを行います。ソーステーブル上でこの条件によってフィルタリングされた行はロックされます。 • *clause*を二重引用符で囲み、**scott.emp**のように完全修飾名でテーブルを参照します。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。

コンポーネント	説明
	• 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、PostgreSQLのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句"`c1 > '10-SEP-2001'`"は機能しませんが、"`c1 > to_date('10-SEP-2001', 'DD-MON-YYYY')`"は機能します。 • オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列は、以下の例のようにエスケープされた二重引用符で囲む必要があります。 `sp_ctrl>`**compare scott.emp sourcewhere "\"file >001005\""** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例#1**: `sp_ctrl(sysA)>`**compare scott.emp sourcewhere "file >001005"** **例#2**: 次の例は、**sourcewhere**および**where**オプションをどのように組み合わせて、望ましい結果を得るかを示しています。**sourcewhere**句はソースcompareプロセスのみが使用しますが、**where**句はソースcompareプロセスとターゲットcompareプロセスの両方が使用します。 `sp_ctrl(SysA)>`**compare scott.emp sourcewhere "deptno = 200" where "mgr = 'SMITH'"**
targetwhere "*clause*"	**compareに有効** ソーステーブルに列が存在しない場合に、ターゲットテーブル内の1つまたは複数の列を基にcompareを行います。ターゲットテーブル上でこの条件によってフィルタリングされた行はロックされます。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、PostgreSQLのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句"`c1 > '10-SEP-2001'`"は機能しませんが、"`c1 > to_date('10-SEP-2001', 'DD-MON-YYYY')`"は機能します。

コンポーネント	説明
	• オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列には、以下のようにエスケープされた二重引用符の余分なセットが必要です。 */productdir/***bin/sp_ctrl compare scott.emp targetwhere "\"file >001005\""** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例#1**: `sp_ctrl(SysA)>` **compare scott.emp targetwhere "file >001005"** **例#2**: 次の例は、**targetwhere**および**where**オプションをどのように組み合わせれば、望ましい結果が得られるかを示しています。**targetwhere**句はターゲットcompareプロセスのみが使用しますが、**where**句はソースcompareプロセスとターゲットcompareプロセスの両方が使用します。 `sp_ctrl(SysA)>`**compare scott.emp where "deptno = 200" targetwhere "mgr = 'SMITH'" repair**
threads *thread_count*	**compareおよびcompare usingに有効** compareプロセスが使用する処理スレッドの数を設定します。 **例** `sp_ctrl(sysA)>`**compare scott.emp threads 4** `sp_ctrl(sysA)>`**compare using sales threads 4**
to *target_owner.target_table*[*.partition*]	**compareに有効** ソーステーブルをそのターゲットの1つとのみcompareします。ソーステーブルが複数のターゲットシステムにレプリケートされ、ターゲットテーブルに異なる名前がある場合に使用します。 このオプションはターゲットパーティションを指定するためにも使用できます。 `compare source_owner.source_table.[source_partition] to target_owner.target_table.[target_partition]` **例** （パーティションのcompare） `sp_ctrl(SysA)>`**compare scott.emp.east to scott.allemp.alleast**

コンポーネント	説明
where “*clause*”	**compareに有効** ソースシステムとターゲットシステムの両方で、SELECTステートメントにWHERE句を含めます。WHERE句は、特定の行をcompareするためのフィルタとして機能します。この条件によってフィルタリングされた行はロックされます。 “*clause*"には、サブクエリを含まない標準のWHERE句を指定します。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、PostgreSQLのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10-SEP-2001', 'DD-MON-YYYY')"`は機能します。 • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**compare scott.emp where “region=4”**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

PostgreSQLのcompare statusコマンド

compare statusコマンドを使用して、最後に実行されたcompareまたはrepairジョブのステータスを表示します。**compare statusコマンド**を使用すると、compareジョブやrepairジョブ、またはcompareジョブやrepairジョブの一部に関する詳細なステータスを表示したり、SharePlexに履歴があるすべてのcompareジョブおよびrepairジョブのステータスを表示したりできます。

注意: **compare statusコマンド**の詳細および使用例については、**job statusコマンド**を参照してください。

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	ビューアー(3)
発行場所:	ソースシステム
関連コマンド:	compare / compare using、repair / repair using

構文

基本コマンド	コマンドオプション	リモートオプション
job status	[**job_id**]	[**on** *host* |
	[**Job_id.table_id**]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |
	[**full**]	**on** *login/password@host:portnumber*]
	[**detail**]	
	[**status**]	

構文の説明

コンポーネント	説明
job status	SharePlexが履歴を保持するすべてのcompareジョブおよびrepairジョブのステータスを表示します。
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)>`**job status 2828.2**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。

コンポーネント	説明
	例: `sp_ctrl(sysA)`**>job status 2828.HR.SRC_TEST3**
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)>`**job status all**
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)`**>job status 2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)`**>job status detail**
status	指定されたステータスの過去のジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)`**>job status "Error"**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモー

オプション	説明
	トシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

PostgreSQLの設定のコピー

設定ファイルをコピーし、そのコピーを別のファイル名で保存するには、**copy config**コマンドを使用します。

設定をコピーすることで、以下のことが可能になります。

- バックアップ用に複製を保存する
- **edit config**コマンドを使用して、新しい設定を作成する際のベースとしてコピーを使用する
- アクティブな設定ファイルを変更します。現在の設定を使用したレプリケーションを継続しながら、**edit config**コマンドを使用してコピーに変更を加え、準備ができたら新しい設定をアクティベーションすることができます。

このコマンドは、元の設定と同じソースシステムに設定をコピーします。設定ファイルを*異なる*システムにコピーするには、テープ、CD、またはFTP経由でファイルを転送します。システムの設定のリストを表示するには、**list config**コマンドを使用します。設定の内容を表示するには、**view config**コマンドを使用します。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行場所:	ソースシステム
関連コマンド:	**activate config**、**create config**、**list config**、**show config**、**verify config**、**view config**

構文

基本コマンド
copy config *filename* **to** *newname*

構文の説明

コンポーネント	説明
filename **to***newname*	• *filename*はコピーする設定の名前です。設定名では大文字と小文字が区別されます。 • **to**は構文の必須部分です。 • *newname*はコピーに付ける名前です。システム上の設定間で一意である必要があります。 **例**: `sp_ctrl(sysA)>`**copy config sales to sales2**

PostgreSQLの設定の作成

新しい設定ファイルを作成するには、**create config**コマンドを使用します。このコマンドは、SharePlexのデフォルトのテキストエディタを実行して、新しいファイルを開きます。設定を作成するには、『*SharePlex管理者ガイド*』の第5章の手順に従ってください。

新しい設定に既存の設定と同じ名前を付けようとすると、SharePlexは次のエラーメッセージを返します。「`The parameter for create config must be a new filename.`」(create configのパラメーターは新しいファイル名にする必要があります。)

設定を作成する代わりに、**copy config**コマンドを使用して既存の設定を複製し、**edit config**コマンドを使用してそのコピーを編集することもできます。

用途

サポート対象のソース:	PostgreSQL(オンプレミス)、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
認証レベル:	オペレーター(2)
発行場所:	ソースシステム
関連コマンド:	**activate config**、**copy config**

構文

基本コマンド
create config *filename*

構文の説明

コンポーネント	**説明**
filename	作成する設定の名前。設定名では大文字と小文字が区別されます。この名前は、システム上の設定間で一意である必要があります。 **例**: `sp_ctrl(sysA>`**create config sales**

PostgreSQLの設定の非アクティベーション

アクティブな設定のレプリケーションをグレースフルに終了するには、**deactivate config**コマンドを使用します。このコマンドは、設定のすべてのCaptureアクティビティを停止し、現在キューにあるすべてのデータをポストし、関連するSharePlexプロセスとキューを削除し、レプリケーションスロット（物理/倫理）を削除します。

deactivate configコマンドは、システムから設定を削除するのではなく、再度アクティベーションするまで、そのオブジェクトのレプリケーションを停止するだけです。設定を非アクティベーションした後、ユーザが設定されたソースオブジェクトに変更を加え続けると、ターゲットデータが同期しなくなる可能性があります。システム上の1つの設定を非アクティベーションしても、同じシステム上の他のアクティブな設定は非アクティベーションされません。

注意: SharePlex 11.1では、アクティベーションの際に作成された物理スロットまたは論理スロットは、非アクティベーションに成功するとデータベースから削除されます。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
認証レベル:	管理者（1）
発行場所:	ソースシステム
関連コマンド:	**list config**

構文

基本コマンド
deactivate config *filename*

構文の説明

コンポーネント	**説明**
filename	無効にする設定の名前。設定名では大文字と小文字が区別されます。 **例**: `sp_ctrl(sysA)>`**deactivate config sales**

PostgreSQLのパーティションの削除

drop partitionコマンドを使用して、水平分割レプリケーション設定内の1つのパーティションスキームから行パーティションを削除します。削除する行パーティションごとに、**drop partition**コマンドを発行します。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行場所:	ソースシステム
関連コマンド:	add partition、modify partition、drop partition scheme、view partitions

構文

基本コマンド

drop partition from *scheme_name*

where

{

{**condition =***column_condition*

route = *routing_map* |

name = *partition_name* |

tablename = *target_table* |

description = *description*

}

構文の説明

構文コンポーネント	説明
scheme_name	パーティションスキームの名前。
condition	削除するパーティションの行を定義する列条件。
route	削除するパーティションのルーティングマップ。
tablename	削除するパーティション内のターゲットテーブルの完全修飾名。
name	削除するパーティションの短縮名。
説明	削除するパーティションの説明。

例

```
sp_ctrl> drop partition from scheme1 where name = q1
sp_ctrl> drop partition from scheme1 where condition = "C1 < 200"
sp_ctrl> drop partition from scheme1 where route = sysb:q2@r.dbname
```

PostgreSQLのdrop partition schemeコマンド

SharePlexレプリケーション設定からパーティションスキームを削除するには、**drop partition scheme**コマンドを使用します。このコマンドを使用すると、パーティションスキームに指定されているすべての行パーティションも削除されます。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行場所:	ソースシステム
関連コマンド:	add partition、modify partition、drop partition、view partitions

構文

基本コマンド
drop partition scheme *scheme_name*

構文の説明

構文コンポーネント	説明
scheme_name	パーティションスキームの名前。

例

```
sp_ctrl> drop partition scheme sales
```

PostgreSQLの設定の編集

非アクティブな設定ファイルを修正するには、**edit config**コマンドを使用します。このコマンドは、SharePlexのデフォルトのテキストエディタで設定を開きます。エディタが提供するツールを使用して設定を修正し、保存します。

アクティブな設定を修正するには、**copy config**コマンドでアクティブな設定をコピーし、**edit config**コマンドでコピーを編集することをお勧めします。コピーをアクティベーションすると、元の設定は非アクティブになります。

システムの設定のリストを表示するには、**list config**コマンドを使用します。設定の内容を表示するには、**view config**コマンドを使用します。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ソースシステム
関連コマンド:	**activate config**、**copy config**、**create config**、**list config**、**rename config**、**verify config**、**view config**

構文

基本コマンド
edit config *filename*

構文の説明

コンポーネント	**説明**
filename	編集する設定の名前。設定名では大文字と小文字が区別されます。 **例**: `sp_ctrl(sysA)>`**edit config sales**

PostgreSQLのフラッシュ

特定の時点でPostプロセスを停止するには、**flush**コマンドを使用します。このコマンドは、データストリームにマーカーを配置し、マーカーポイントで自動的にPostプロセスを停止します。例えば月末のレポートを作成する必要があるときなど、このコマンドは特定の日時で発行することができ、ターゲットシステム上のデータは、コマンドが発行された時点でソースシステム上にあったものが反映されます。

バックアップを実行するとき、または特定の文書化されたSharePlexの手順に従うときは、ソースシステムのユーザアクセスを停止した後に**flush**コマンドを発行して、Postが停止される前に最後のトランザクションがターゲットデータベースにポストされるようにします。

Postの再開

flushコマンドの後に生成された変更はキューに蓄積され、Postを再度起動した後にターゲットインスタンスに適用されます。**start post**コマンドが発行されるまで、Postは停止したままになります。Postは停止した場所の記録を保持し、同期を維持するためにその時点からポストを再開します。

重要: Postが停止している間にユーザがソースシステムで変更を続けると、データがPostキューに蓄積され、使用可能なディスク容量がすべて消費される可能性があります。Postは可能な限り早く開始してください。

注意: ポストの**Status**コマンドによって「stopped due to flush（フラッシュのために停止しました）」と表示されます。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ソースシステム

構文

基本コマンド	コマンドオプション
flush*datasource*	[**to***host*] [**queue***queuename*] [**to***host***queue***queuename*] [**to***host@target_SID*] [**to***host@target_SID***queue***queuename*]

構文の説明

コンポーネント	説明
datasource	この引数は、コマンドが作用するソースデータを指定します。追加オプションがない場合、すべてのターゲットシステムのすべてのキューを通してデータをフラッシュし、Postを停止します。 datasourceは**r.**dbnameで表されます。ここで、dbnameはソースPostgreSQLインスタンスのデータベース名です。 **例**: `sp_ctrl(sysA)>`**flush r.dbname1** 上の例では、インスタンス**r.dbname1**のデータがすべてのターゲットシステムにフラッシュされます。
to*host*	このオプションは、指定されたターゲットシステムにレプリケートしているすべてのデータをフラッシュします。他のターゲットシステムへのレプリケーションは影響を受けません。 **例**: sp_ctrl(sysA)> **flush r.dbname1 to sysB** 上の例では、データが**dbname1**から**sysB**にフラッシュされます。
queue*queuename*	このオプションは、指定されたpostキューのデータをフラッシュします。これは、すべてのターゲットシステム上のその名前のキューを流れるデータをフラッシュします。このオプションは、名前付きpostキューがある場合に最も便利です。 **例**: sp_ctrl(sysA)> **flush r.dbname1 queue sales** 上の例では、salesという名前のpostキューを持つすべてのターゲットシステム上のpostキュー*sales*のデータがフラッシュされます。
to*host***queue** *queuename*	このオプションは、指定されたターゲットシステム上の指定されたpostキューのデータをフラッシュします。そのシステムおよび他のターゲットシステム上の他のpostキューは影響を受けません。 **例**: sp_ctrl(sysA)> **flush r.dbname1 to sysB queue sales** 上の例では、sysB上のpostキューsalesのデータがフラッシュされます。
to*host@datadest*	このオプションは、指定されたターゲットシステム上の指定されたターゲットインスタンスにデータをフラッシュします。次のものには影響しません。 • そのシステム上の他のターゲットインスタンス。 • 他のターゲットシステム上の同じORACLE_SID/データベース名を持つ他のターゲットインスタンス **この構文では:** • *host*は、ターゲットシステムの名前です。 • *datadest*は**o.***SID*で表されます。ここで、*SID*はターゲットインスタンスのORACLE_SIDです。 • **datasource**は**r.**dbnameで表されます。ここで、dbnameはPostgreSQLターゲットインスタンスのデータベース名です。

コンポーネント	説明
	例: sp_ctrl(sysA)> **flush r.dbname1 to sysB@r.dbname2**（PostgreSQLからPostgreSQLへの実装） 上の例では、データがsysB上のデータベースdbname2にフラッシュされます。 sp_ctrl(sysA)> **flush r.dbname1 to sysB@o.oraB**（PostgreSQLからOracleへの実装） 上の例では、データがsysB上のOracleインスタンスoraBにフラッシュされます。
to*host*@*datadest***queue***queuename*	このオプションは、指定されたターゲットシステム上の指定されたpostキューとターゲットインスタンスのデータをフラッシュします。次のものには影響しません。 • そのインスタンスまたはそのシステム上の他のターゲットインスタンスの他のpostキュー。 • 他のターゲットシステム上の他のpostキュー。 **この構文では:** • *host*は、ターゲットシステムの名前です。 • *datadest*はo.SIDで表されます。ここで、SIDはターゲットインスタンスのORACLE_SIDです。 • *queuename*はpostキューの名前です。 **例**: sp_ctrl(sysA)> **flush r.dbname1 to sysB@r.dbname2 queue sales**（PostgreSQLからPostgreSQLへの実装） 上の例では、sysB上のターゲットPostgreSQLデータベースdbname2のpostキューsalesのデータがフラッシュされます。 sp_ctrl(sysA)> **flush r.dbname1 to sysB@o.oraB queue sales**（PostgreSQLからOracleへの実装） 上の例では、sysB上のターゲットインスタンスoraBのpostキューsalesのデータがフラッシュされます。

PostgreSQLのjob statusコマンド

job statusコマンドを使用して、最後の**compare**、**compare using**、**repair**、または**repair using**ジョブのステータスと、実行中の他のジョブのステータスを表示します。**job statusコマンド**を使用すると、ジョブまたはジョブの一部の詳細なステータスを表示したり、SharePlexが履歴を保持するすべてのジョブのステータスを表示したりできます。SharePlexは、SP_SYS_JOB_HISTORY_RETENTIONパラメーターで指定されたすべてのジョブの履歴を保持します。履歴やジョブ・プロセス・ログを能動的に削除する方法については、**clear history**コマンドおよび**remove log**コマンドを参照してください。

```
sp_ctrl (sysA) > job status

Job ID     : 861
PID        : 20571
Host       : sysa.domm.com
Started    : 06-NOV-12 11:07:05
Job Type   : Compare
Status     : Processing - 4 objects completed


ID  Tablename                Status       Time      Total rows %Comp  Total
                                                                     time
--- ---------------------- --------     ---------  ---------- ------ ---------
1   "SCOTT"."SRC_TEST1"     Out Sync     N/A       19         100    0:05
4   "SCOTT"."SRC_TEST4"     WaitMarker   0:02      27392             0:04
5   "SCOTT"."SRC_TEST5"     Init         0:01      27498             0:01
```

SharePlexが履歴を保持するすべてのジョブの概要を表示するには:

```
sp_ctrl (alvspxl11:8567)> job status all


Job ID Type      filename/tablename      Status     Started          Completed
------ ------    ------------------      --------   --------------   -------------
3441   Compare   prod.conf               Done -     16-DEC-11 15:39  16-DEC-11
                                         errors                      15:40
3442   Repair    scott.src_test4         Done       16-DEC-11 15:50  16-DEC-11
                                                                      15:50
3443   Repair    prod.conf               Done       19-DEC-11 10:42  19-DEC-11
                                                                      10:42
3445   Compare   scott.src_test33        Done -     20-DEC-11 12:02  20-DEC-11
                                         errors                      12:03
```

ジョブのステータスを詳細に表示するには:

```
sp_ctrl (alvspxl11:8567)> compare status detail


Job ID      : 3448
PID         : 763
Host        : sysa.domm.com
Started     : 20-DEC-11 12:40:46
Job Type    : Compare
Status      : Done - 6 objects completed


      Table ID            : 3448.3
      Table               : "SCOTT"."SRC_TEST3"
      From                : sysa.domm.com@r.w111a64f
      To                  : "SYSPROD"."SRC_TEST3" sysb.domm.com@r.w111a64f
      Started             : 20-DEC-11 12:40:55
      Percent complete    : 100%
      Total Rows          : 234452
      Rows processed      : 234452
      Rows out-of-sync    : 2
      Status              : Out Sync
      Status Elapsed      : N/A
      Total Elapsed       : 0:07
                  Inserts : 2
                  Updates : 0
                  Deletes : 0
```

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**compare status**、**repair status**

構文

基本コマンド	コマンドオプション	リモートオプション
job status	[*job_id*]	[**on** *host* |
	[*job_id.table_id*]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |
	[**full**]	**on** *login/password@host:portnumber*]

基本コマンド	コマンドオプション	リモートオプション
	[**detail**]	
	[**status**]	

構文の説明

コンポーネント	説明
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)>`**job status 2828.2**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。 **例: `sp_ctrl(sysA)>`job status 2828.HR.SRC_TEST3**
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)>`**job status all**
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)>`**job status2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)>`**job status detail**
status	指定されたステータスの過去のジョブのステータス履歴を表示します。 `sp_ctrl(SysA)>`**job status "Error"**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム（現在の**sp_ctrl**セッションが実行されているシステム以外）でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

PostgreSQLの設定のリスト

ソースシステム上のすべてのアクティブおよび非アクティブな設定のリストを表示するには、**list config**コマンドを使用します。

このコマンドは、次の情報を表示します。

- **ファイル名**: 設定の名前、変更された日時、およびファイルのサイズ。
- **状態**: 設定がアクティブか非アクティブか。*アクティブ*は、設定ファイルが現在レプリケーションに関与していることを意味します。*非アクティブ*は、設定が現在レプリケーションに関与していないことを意味します。
- **データソース**: 設定によってレプリケートされるオブジェクトを含むPostgreSQLインスタンス。
- **内部名**: アクティブな設定の内部コピーの名前。これは、SharePlexが実際にレプリケートするファイルです。その名前は、アクティブな設定の**State**見出しの下に表示されます。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
認証レベル:	ビューアー（3）
発行対象:	ソースシステム
関連コマンド:	**view config**、**show config**

構文

基本コマンド
list config

PostgreSQLのパーティションの変更

modify partitionコマンドを使用して、水平分割レプリケーション設定内の1つのパーティションスキームの行パーティションを変更します。

既にレプリケートされているテーブルにコマンドが影響する場合は、設定ファイルを再アクティベーションします。SharePlexは設定が変更されたテーブルのみをロックします。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行場所:	ソースシステム
関連コマンド:	add partition、drop partition、drop partition scheme、view partitions

構文

基本コマンド
modify partition in *scheme_name* **set** *keyword*=*value* [**and** *keyword*=*value*] [...] **where** *keyword*=*value* [**and** *keyword*=*value*] [...]

構文の説明

注: これらのオプションの詳細については、add partitionを参照してください。

構文コンポーネント	説明
scheme_name	パーティションスキームの名前。このコンポーネントは変更しません。変更すると行パーティションが新しいパーティションスキームに移行します。
keyword	*scheme_name*を除く、以下の構文コンポーネントのいずれか。
condition	行パーティションを定義する列条件。
route	このパーティションのルーティングマップ。
tablename	完全修飾ターゲットテーブル名。
name	このパーティションの略称。
説明	このパーティションの説明。

例

```
sp_ctrl> modify partition in scheme1 set condition = "C1 > 400" and route = sysc:q1@r.dbname where name = q1
```

```
sp_ctrl> modify partition in scheme1 set condition = "C1 > 400" where condition = "C1 > 300"
```

PostgreSQLの設定のページ

purge configコマンドを使用して、キューそのものを削除したり設定を非アクティベーションにしたりすることなく、設定に関連付けられたすべてのキューからデータを削除します。非アクティベーションを回避することで、SharePlexでは、設定データを再計算する必要がなくなります。これにより、テーブルが大きく数が多い場合に、時間を節約し、レプリケーションをより早く開始できるようになります。

ソースシステムで**purge config**コマンドを発行して、設定されたルートにあるソースシステムとすべてのターゲットシステムに影響を与えます。**purge config**アクティビティの実行前または実行中にSharePlexプロセスが停止すると、このコマンドも動作を停止します。プロセスが再開されると、コマンドは動作を再開します。したがって、**purge config**は、ネットワークが一時的に利用できなくなった場合でも機能し、接続が復元されるまでコマンドはキューに残ります。

purge configコマンド使用上の注意点:

設定をアクティブにしてから、**activate config**コマンドの後に、**purge config**コマンドを実行しないでください。レプリケーションを制御する設定情報など、キューに入れられたデータ以上のものをパージし、アクティベーションが無効になる可能性があります。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ソースシステムおよびターゲットシステム
関連コマンド:	**abort config**、**deactivate config**

構文

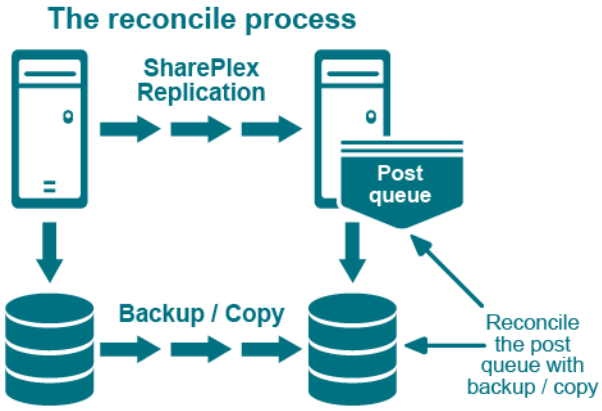
基本コマンド
purge config *filename*

構文の説明

コンポーネント	説明
filename	パージする設定の名前。設定名では大文字と小文字が区別されます。 **例**: `sp_ctrl(sysA)>`**purge config sales**

PostgreSQLの照合

reconcileコマンドをプロシージャの一部として使用して、データベースユーザの中断を最小限に抑えると共に、ソースデータとターゲットデータを同期（インスタンス化）します。**reconcile**コマンドは、進行中のレプリケーションの結果を、ホットバックアップやネイティブ・コピー・ユーティリティなどによってターゲットシステムに適用されたソースデータのコピーと照合します。照合機能は、postキュー内のレプリケートされた変更を、リカバリプロセス後のターゲットデータベースの状態とcompareします。リカバリ中に適用されたトランザクションと、まだ適用されていない（postキューで待機している）トランザクションを区別し、両システムが同期するように、重複していない変更のみをポストします。



reconcileコマンドは、大容量環境で使用するように設計されていますが、状況によっては照合プロセスが停滞する可能性があることを理解すれば、小容量環境でも使用することができます。この状況は、ソースシステムから継続して届くデータに**reconcile**コマンドが依存しているために生じます。ホットバックアップまたはコピーの実施以降、ソースシステムにレプリケーションアクティビティがない場合、照合プロセスはソースのアクティビティが再開するまで待機します。

reconcileコマンド使用時の注意点

reconcileコマンドは、ソースデータとターゲットデータの初期同期を特定の手順で行う場合に使用します。したがって、これは独立したコマンドではありません。初期同期手順については、『SharePlex管理ガイド』を参照してください。

注意:

- LSNの前にコミットされたトランザクションがターゲットに確実にレプリケートされるようにします。
- コミットされていないトランザクションは削除されず、postキューに置かれます。
- ROLLBACK/COMMITがLSNの前にある場合、トランザクションはpostキューから削除されます。
- ROLLBACK/COMMITがLSNの後ろにある場合、トランザクションはpostキューに残ります。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ターゲットシステム
関連コマンド:	**flush**

構文

基本コマンド	コマンドオプション
reconcile queue*queuename***for***datasource-datadest*	[**to flush**] [**pglsn** Log Sequence Number]

構文の説明

コンポーネント	説明*
queue	**queue**はコマンドの必須部分です。
queuename	照合するターゲットシステム上のpostキュー。有効な値: • デフォルトキューを使用する場合のソースシステムの名前 • 名前付きキューを使用している場合のキューの名前。 名前付きpostキューを使用している場合は、それぞれに対して**reconcile**コマンドを実行します。キュー名を決定するには、**sp_ctrl**で**qstatus**コマンドを発行します。キュー名はすべてのプラットフォームで大文字と小文字が区別されます。
for*datasource-datadest*	• datadestは**o.**SIDで表されます。ここで、SIDはターゲットインスタンスのORACLE_SIDです（Oracleターゲットの場合）。 • datasourceは**r.**dbnameで表されます。ここで、dbnameはソースインスタンスのデータベース名です（PostgreSQLソースの場合）。 • datasourceは**r.**dbnameで表されます。ここで、dbnameはターゲットインスタンスのデータベース名です（PostgreSQLターゲットの場合）。 **例**: sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-r.dbname2**

コンポーネント	説明*
	（PostgreSQLからPostgreSQLへの実装） sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-o.oraB** （PostgreSQLからOracleへの実装）
pglsn LSN number	PostgreSQLの実装の場合はLSN番号を使用します。LSN番号は10進数または16進数形式で指定できます。 **現在のLSNを検索するクエリ:** `select pg_current_wal_lsn();` **例**: sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-r.dbname2 pglsn 0/B817B360**（PostgreSQLからPostgreSQL - 16進形式のLSN） sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-r.dbname2 pglsn 3088560992**（PostgreSQLからPostgreSQL - 10進形式のLSN） sp_ctrl (sysB)>**reconcile queue SysA for r.dbname1-o.oraB pglsn 3088560991**（PostgreSQLからOracle - 10進形式のLSN） sp_ctrl (sysB)>**reconcile queue SysA for r.dbname1-o.oraB pglsn 0/B817B361**（PostgreSQLからOracle - 16進形式のLSN）
to flush	このオプションは、**flush**コマンドで確立されたフラッシュマーカーに対して照合するために使用します。ピアツーピアレプリケーション環境で複数のOracleデータベースを同期する場合に使用します。 この構文は、基本コマンドの構文の後に表示する必要があります。 **例**: sp_ctrl (sysA)> **reconcile queue SysA for r.dbname1-r.dbname2 to flush** **注意:** ターゲットでReconcileコマンドを実行する前に、ソースでFlushコマンドを発行し、フラッシュマーカーを作成します。最初にReconcileコマンドを発行しても、ソースマシン上でフラッシュマーカーが作成されるのを待つことになります。

PostgreSQLの設定名の変更

設定ファイルに別の名前を付けるには、**rename config**コマンドを使用します。システム上の設定ファイル間で固有の名前を使用します。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ソースシステム
関連コマンド:	**copy config**、**edit config**、**list config**、**view config**

構文

基本コマンド
rename config {*filename* **to** *newname*

構文の説明

コンポーネント	説明
filename **to** *newname*	• *filename*は、名前を変更する設定の名前です。設定名では大文字と小文字が区別されます。 • **to**は構文の必須部分です。 • *newname*は設定に付ける新しい名前です。 **例**: `sp_ctrl(sysA)` **> rename config sales to sales2**

PostgreSQLのrepair/repair using

repairおよび**repair using**コマンド（*repair*コマンドと総称される）を使用して、ターゲットテーブル内の同期していない行をrepairします。

- **repair**コマンドは、ワイルドカードを使用して指定されたスキーマ内の個々のターゲットテーブルまたは任意の数のターゲットテーブルをrepairします。個々のテーブルをrepairする場合は、列ベースのフィルタリングを使用して、repair用に選択される行を制御することができます。
- **repair using**コマンドは、アクティブ設定またはアクティブ設定内のテーブルのサブセットを含む別のファイルにリストされたターゲットテーブルをすべてrepairします。

repairコマンドは、まずcompareを実行してrepairが必要な行を特定してからrepairを実行します。テーブルがどのようにcompareされるかについては、compare / compare usingを参照してください。

注意： 実行中のcompareまたはrepairは、ソーステーブルには一切影響を与えません。SharePlexは、読み取りの一貫性についてのクエリを実行するためだけにデータベースにログインし、ソーステーブルのロックは短時間です。SharePlexは、処理中にターゲットテーブルを短時間ロックしますが、ユーザはロックをほとんど意識することなくアクセスを継続できます。ターゲットテーブル（またはしきい値より小さい場合は非同期行）は、非同期行を修正するためにrepairプロセスにロックされます。

サポートされている操作

SharePlexは、**DML**操作（INSERT、UPDATE、DELETE）によって発生した、ターゲットテーブル内の非同期行を検出し、repairすることができます。

サポートなし

SharePlexは、以下のcompareとrepairをサポートせずにスキップします。

- シーケンス。
- トランスフォーメーションが使用されたテーブル
- ソースサーバとターゲットサーバ間のデータ受け渡しにパススルーサーバを使用するネットワーク構成
- カスケードレプリケーション環境では、compareとrepairはサポートされません。
- compareまたはrepair中のテーブルに対してDDLを実行しないでください。compareは、SharePlexがサポートするものも含め、DDL操作によって引き起こされる非同期状態を検出しません。DDLがテーブル定義を変更すると、compareする必要がある行を取得するためにcompareプロセスにより作成されたSELECTステートメントが無効になります。

  DDLによって発生した非同期状態を修正したら、**repair**コマンドを使用して行のデータを再同期することができます。
- 255文字を超えるcompareおよびrepairコマンド文字列はサポートされていません。これはオペレーティングシステムの制限です。この制限を回避するには、ソースシステムで**edit**コマンドを使用します。テキストファイル内にコマンド文字列を入力すると、コマンドが自動的にファイルを実行します。

compareおよび**compare using**でサポートされるデータ型の詳細については、『SharePlexリリースノート』を参照してください。

その他の条件

- compareまたはrepairするテーブルは、アクティブな設定ファイルの一部でなければなりません。
- ソーステーブル上にコミットされていないトランザクションがあると、読み取りの一貫性を得るためにcompareおよびrepair処理に必要な短時間のロックを取得できなくなります。compareやrepairを実行する前に、すべてのトランザクションがコミットされていることを確認してください。
- レプリケーションのレイテンシは、compareやrepairの処理のパフォーマンスを低下させる可能性があります。ターゲット上でcompareおよびrepairプロセスを生成するためにソースから送信されたメッセージは、複製されたデータと共にキューを通じて送信されます。データバックログによる遅延が生じると、生成メッセージの送信が遅れることにより、プロセスが読み取りの一貫性を失うことがあります。可能であれば、compareやrepairはピーク以外の時間帯に行います。
- compareまたはrepair中のテーブルをトランケートしないでください。compareコマンドは、開始時に各ソーステーブルのスナップショットを取ります。テーブルがトランケートされると、スナップショットのテーブルビューもトランケートされ、コマンドが無効な非同期状態を返すことがあります。
- ビューをrepairするには、以下が真でなければなりません。
    - ビューテーブルが、基礎となるテーブルのすべての制約列を含んでいること。
    - ビューの制約列の名前が、基礎となるテーブルの制約列の名前と同一であること。
- Compareプロセスのパフォーマンスを向上させるために、SharePlexはSP_DEQ_PARALLELISMパラメーターによる並列ヒントをサポートしています。このため、並列度、つまり起動するワーカープロセスの数を指定できます。

  デフォルトでは、PostgreSQLデータベースオプティマイザはSQLステートメントに最適な問い合わせ実行計画を選択します。これは、並列ヒントで指定した並列度に直接関連していないこともあります。pg_hint_planにより、クエリを実行するたびに配列ヒントを使用して実行計画を調整することができます。SharePlexは、データベースにpg_hint_plan拡張機能がインストールまたは設定されている場合、内部でこれを利用します。

  並列クエリを使用するための構文の例は以下の通りです。

  pg_hint_plan拡張機能は並列ヒントをサポートしています。`Parallel(table <# of workers> [soft|hard])`

  SharePlexはハードパースをサポートする並列ヒントを使用します。`Parallel(table <# of workers> [hard])`

  > **注意:** ワーカーの数は、データベースシステムのCPU VCoreより少なくなければなりません。

repairコマンドの使用方法

compareおよびrepairコマンドによって同期されたデータを維持するための推奨手順は、まず**compare**または**compare using**コマンドを実行し、次に**repair status**コマンドで結果を表示することです。このコマンドは、非同期行とその考えられる原因を表示します。非同期状態の原因が修正されない限り、今回行をrepairしても、レプリケーションは再び非同期になります。問題が解決した後、**repair**または**repair using**コマンドを実行します。

repairまたは**repair using**コマンドは、予めcompareを行うことなく実行できます。このコマンドはまずcompareを行い、同期していない行を特定し、その行をrepairします。しかし、将来的な非同期状態を防ぐには、非同期状態の根本的な原因を解決しなければなりません。

非同期状態の原因と解決策については、『SharePlex管理ガイド』を参照してください。

repairを実行するタイミング

ターゲットテーブルをrepairする最適なタイミングは、テーブルのサイズ、問題の原因、非同期行の範囲、ユーザをロックアウトできる時間の長さによって異なります。repairを開始する前に、以下を考慮してください。

- 通常、テーブルのユーザは、テーブルのcompare時に適用される短時間のロックの影響は受けませんが、repair処理中はターゲットテーブルからロックアウトされます。小さなテーブルであれば支障はないかもしれませんが、大規模なrepairが必要な大きなテーブルの場合、待ち時間が非常に長くなることがあります。
- ターゲットテーブルのロックは、Postがrepairの終了を待ってから、そのテーブルに変更を適用し、他のテーブルに移動しなければならない場合、ポストのパフォーマンスを低下させる可能性があります。これがターゲットデータのレイテンシを増加させ、postキューに操作が蓄積される原因となります。Postが変更する必要があるオブジェクトと、repair中のオブジェクトが異なる場合、2つのプロセスは同時に実行されます。
- テーブルをすぐにrepairしなければならないが、ロックやレプリケーションのレイテンシを許容できない場合、**where**オプションを使用して、特定の行に限定してrepairすることができます。別の方法として**key**オプションを使用することもできますが、このオプションを使用すると、非同期行を見逃してしまう可能性があります。
- repairを待つことができる場合は、問題の原因をすぐに修正し、ピーク時以外の時間帯にrepairを行います。

特別な使用例

以下のシナリオでは、compareを実行する際に特別な処理が必要となります。

使用例	サポートcompare
統合レプリケーション	中央データベース内のターゲットテーブルは、どのソースデータベースよりも行数が多く、ソースデータベースよりも列数が多いことがよくあります。このような環境でrepairコマンドを使用する場合は、特別な配慮が必要です。 **repair usingコマンド** 統合レプリケーションは、**repair using**コマンドではサポートされていません。**repair using**コマンドでは、ソーステーブルに存在しないターゲット行が不要に削除されます。 回避策として、統合レプリケーションに関係するテーブルを除外した設定のサブセットを作成し、代わりにサブセット設定をrepairします。**repair**コマンドを使用すると、統合レプリケーションに関与するテーブルをrepairできます。 **repairコマンド** 統合レプリケーションは、ターゲットデータベースとPostプロセスがソースホストのIDを各行に追加するように設定されている場合にサポートされます。中央のターゲットテーブルの正しい行をcompareまたはrepairするには、**targetwhere**オプションを使用し、ソースID値を**where**句の基準にします。 例えば、ある企業の東部本社のデータベースのテーブルと、中央の企業データベースの正しい行をcompareするには、東部データベースのソースIDに「East」を使用し、その値を**targetwhere**句の基準にすることができます。**repair**コマンドで同じ**targetwhere**句を使用します。compareプロセスおよびrepairプロセスは、ソースID値を使用して、東部データベースに有効な行のみを選択します。 ソースIDを特定する統合レプリケーション以外の実装でcompareコマンドやrepairコマンドを使用すると、ターゲット行が不要に削除される可能性があります。この設定の詳細については、『SharePlex管理ガイド』を参照してください。

使用例	サポートcompare
	ターゲット行が正確に選択されるように、**targetwhere**オプションと標準の**where**オプションを組み合わせる必要があるかもしれません。
ピアツーピアレプリケーション	ピアツーピア構成では、*どのシステムを信頼できるソースシステムとし、どのシステムをセカンダリシステム*、つまりターゲットシステムとするかを決めなければなりません。セカンダリシステムは、repairが行われるシステムです。ピアツーピア環境でcompareやrepairを実行する前に、以下の手順に従ってください。 1. セカンダリシステムのユーザアクセスを停止し、そのシステムから複製された操作が信頼できるソースデータベースにポストされるのを待ちます。ユーザは引き続きソースデータベースにアクセスできます。 2. ソースおよびセカンダリシステムで**qstatus**コマンドを実行します。 3. キュー内のメッセージが10個以下であれば、ソースシステムからcompareを実行します。 4. compare中、**sp_xdesvr**および**sp_xdeclt**の開始後に、ソースおよびセカンダリデータベースへのユーザアクセスを許可できます。 5. **where**オプションを指定して**repair**コマンドを使用すると、ユーザをテーブルからロックアウトすることなく、ターゲットテーブルの選択した行をrepairできます。 この設定の詳細については、『SharePlex管理ガイド』を参照してください。
キーのないテーブル	compareおよびrepairコマンドは、ソースおよびターゲットシステム上でORDER BY句付きのSELECTステートメントを発行します。大規模なテーブルにプライマリキーまたはNULL以外の一意キーとインデックス(一意のインデックスが望ましい)を持つ場合、順序付けはより速くなります。それ以外の場合は、すべての列がキーとして使用されます。 一意の行識別子はないが、一意の行として識別できる列がテーブルに1つ以上ある場合、**orderby**オプションを指定して**compare**コマンドを使用できます。このオプションを使用すると、SharePlexは、ソースシステム上の**sp_xdesvr**ログに、コマンドがこれらの列をキーとして使用したという通知を出力します。 1つの行に対してrepairを行う必要がある場合、キーと見なされる列にNULL値が含まれていてはなりません。含まれているとエラーが発生することがあります。この場合、その行を一意に識別し、NULL値を含まない列で**orderby**オプションを使用します。 **制限事項**: SharePlexレプリケーションで、重複データを含む非キー列を含むテーブルを使用する場合、レプリケーションまたはcompare・repair処理でデータの不整合が発生する可能性があります。
ソーステーブルよりも列の数が多いターゲットテーブル	**repair**および**repair using**コマンドは、ソーステーブルに含まれていないターゲット列を無視します。repairでは以下のことを行います。 • INSERTは、ソーステーブルに対応する列を持つターゲット列に値を挿入しますが、それ以外の列には挿入しません。NOT NULL制約があるがデフォルト値がない列は、PostgreSQLエラーの原因となります。ターゲットテーブル内の余分な列にはデフォルト値が推奨されます。 • UPDATEは、ソーステーブルに対応する列を持つターゲット列の値を再同期しますが、余分な列の値は再同期しません。 • DELETEはターゲットテーブル内の余分な列の影響を受けません。これは、repairコマンドは照合する列の列データに基づいて行を選択するためです。
UNIQUE制約を持つテーブル	UNIQUE制約で定義された列では、**repair**、または**repair using**コマンドによって一意制約違反エラーが返されることがあります。次の例では、ソーステーブルとターゲットテーブルがそれぞれ2列ずつになって

<table>
<tr><th>使用例</th><th>サポートcompare</th></tr>
<tr><td></td><td>います。最初の列は主キーで、2番目の列にはUNIQUE制約があります。

Source Table
<table><tr><td>111</td><td>ABC</td></tr><tr><td>222</td><td>XYZ</td></tr></table>
Target Table
<table><tr><td>111</td><td>XYZ</td></tr><tr><td>222</td><td>ABC</td></tr></table>
SharePlexがターゲットテーブルの1行目をrepairしてソースと一致させようとすると、2列目のUNIQUE制約がエラーを返します。これは、値「ABC」が既に2行目に存在するためです。ターゲットテーブルの2行目でも、「XYZ」が既に1行目に存在するため、同じことが起こります。回避策は以下の通りです。

• repairコマンドを実行する前に、ターゲットテーブル上のUNIQUE制約を無効にし、repair終了後に再度有効にします。
• 固有の制約違反が発生したターゲット行を削除し、SharePlexによって正しいデータと共にこれらの行が挿入されるようrepairを再度実行します。</td></tr>
<tr><td>LOB列を持つテーブル</td><td>いずれかのターゲットテーブルにLOB列があるとrepairにかかる時間が長くなります。より高速なrepairを行うには、SP_DEQ_SKIP_LOBパラメーターを0に設定して、compareとrepairでLOB列がスキップされるようにします。詳細については、「SP_DEQ_SKIP_LOB 」の407ページを参照してください。</td></tr>
</table>

repairの仕組み

repair操作では、以下のように**DML**操作によって発生した、ターゲットテーブルの非同期状態をrepairします。

- 余分な行または欠けている行
- 値が一致しない行

repairの条件

repairおよび**repair using**コマンドは、以下の修正SQLステートメントを発行します。

- ソース側に行が存在し、ターゲット側に行が存在しない場合、SharePlexはINSERTステートメントを発行します。
- ターゲット側に行が存在し、ソース側に存在しない場合、SharePlexはDELETEステートメントを発行します。
- ターゲット行がソース行と異なり、キー列が一致する場合、SharePlexはソース値に基づいてUPDATEステートメントを発行します。

プロセス

repairには常に、ターゲットテーブル内の非同期状態を特定するためのcompareが含まれます。**repair**または**repair using**コマンドを実行すると、SharePlexは以下の一連のイベントを開始します。

1. **sp_cop**プロセスは、ソースシステム上に**sp_xdesvr**（サーバ）プロセスを生成します。
2. **sp_xdesvr**プロセスは、**sp_ctrl**インターフェイスの制御と使用をユーザに返します。compareが進行している間、レプリケーションは続行されます。
3. **sp_xdesvr**プロセスは、ターゲットシステム上で**sp_xdeclt**（クライアント）プロセスを開始するために、Postプロセスにメッセージを送信します。
4. サーバプロセスとクライアントプロセスによって相互の通信が確立されます。
5. 行の選択とrepairは次のように進行します。
    - **repair**コマンドを使用している場合、**sp_xdesvr**はソーステーブルから行を選択し、**sp_declt**はターゲットテーブルから行を選択します。行はソートされ、compareされてrepairされます。
    - **repair using**コマンドを使用している場合、**sp_xdesvr**プロセスはターゲットシステム上に多数の処理スレッドを作成します。SP_DEQ_THREADSパラメーターによって設定される値は、作成されるスレッド数を制御します。各スレッドは**sp_xdeclt**（クライアント）プロセスを生成します。サーバプロセスとクライアントプロセスによって相互の通信が確立されます。処理負荷はクライアントプロセス間で分割されます。ソースとターゲットの各テーブルから行が選択され、ソートされ、compareされてrepairされます。ターゲットテーブルは、repairの順番になるとロックされ、その後ロックが解除されます。
    - ターゲットテーブルは、repairの順番になるとロックされ、その後ロックが解除されます。
6. 終了すると、プロセスによってログファイルが書き出され、**show repair**コマンドで結果を見ることができます。

エラー処理

SharePlexは、repair SQLステートメントの適用時にデータベースエラーが発生すると、そのステートメントから先のrepairを停止し、直前に適用された有効なステートメントのみをコミットします。このように、テーブルは部分的にrepairされますが、まだ同期していない可能性があります。**repair status**コマンドは、この状況を警告します。

SQLログファイルの管理

compareおよびrepairコマンドは、すべての非同期行をrepairするために必要なSQLを、ログファイルと同じ場所にあるSQLファイルに書き込みます。compareコマンドのみが発行された場合、SharePlexはこれらのSQLステートメントを実行しません。repairコマンドを実行した場合、このコマンドは、非同期行をrepairするSQLステートメントを実行すること以外はcompareコマンドと同じように動作します。

SQLログファイルの出力を抑制することができます。このファイルを抑制する理由には以下のようなものがあります。

- データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。
- compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。

SQLログファイルを抑制するには、**nosqllog**オプションを**compare**または**repair**コマンドで使用します。

SharePlexの現在のインスタンスの実行中に、compareとrepairのすべての実行についてSQLログファイルの出力を抑制するには、SP_SYS_SECURE_MODE環境変数を1に設定します。この変数はSharePlexを起動する前に設定する必要があるため、**sp_cop**プロセスが実行されている場合は、この変数を設定した後に再起動する必要があります。この環境変数を指定して**sp_cop**を実行すると、compareコマンドとrepairコマンドによってSQLファイルにデータは保存されず、Post処理でもSharePlexのエラーログにデータは保存されません。

複数のプロセスの実行

すべてのcompareおよびrepairコマンドでは、複数のプロセスを同時に実行することができます。

- 複数の**compare**および**repair**コマンドを同時に実行し、それぞれがソーステーブルとターゲットテーブルのペアを処理することも、PostgreSQLのワイルドカードを使用して1つのコマンドで複数のテーブルセットを指定することもできます。SharePlexによるワイルドカードのサポートの詳細については、『SharePlex管理ガイド』を参照してください。
- **compare using**と**repair using**コマンドはファイル全体で機能します。例えば、設定ファイル全体のテーブルをcompareまたはrepairすることも、ターゲットテーブルのサブセットに影響を与える1つ以上の*compare files*または*repair files*を作成し、そのうちの1つまたは複数を同時に実行することもできます。手順については、コマンドの構文を参照してください。

レプリケーションプロセスとcompareおよびrepairプロセスなど、最大で20のSharePlexプロセスでpostキューを同時に使用することができます。一度に最大で5つのcompareおよびrepairプロセスを実行できるようにすることをお勧めします。**compare using**および**repair using**コマンドを使用すると、compareするプロセスあたりのテーブル数を増やすことで20回のプロセス制限を回避できます。

制限に達したためにcompareやrepairに失敗した場合、SharePlexはイベントログにメッセージを記録します。

注意: **edit**コマンドを使用し、以前のコマンドを編集して新しいコマンドを作成すれば、複数のコマンドをより簡単に実行できます。

設定のサブセットのrepair

アクティブな設定のサブセットは、以下の方法でrepairできます。

- 1つのスキーマに属するレプリケーション内の全ターゲットテーブルをrepairするには、以下のように**repair**コマンドにワイルドカードを指定します。

  ```
  sp_ctrl> repair scott.%
  ```

- 設定ファイル内のすべてのターゲットテーブルをrepairするには、以下のように**repair using**コマンドを使用します。

  ```
  sp_ctrl>repair using myconfig
  ```

- 1つのターゲットルート内のすべてのターゲットテーブルをrepairするには、以下のように**at**オプションを指定して、**repair using**コマンドを使用します。

  ```
  sp_ctrl> repair using config.active at prodsys@r.dbid
  ```

- 設定内のテーブルのカスタムサブセットをrepairするには、*repairファイル*から指定します。これはプレーン・テキスト・ファイルであり、repairしたいターゲットのソーステーブルのみをリストします。ターゲットテーブルは、コマンドが発行された時点の設定ファイルから取得されます。**create config**または**copy config**コマンドを使用してrepairファイルを作成できます。このファイルには、必ず設定ファイルでないことを明確に示す名前を付けてください。詳細については、コマンドの構文を参照してください。

どの行をrepairするかを制御

compareおよび**repair**コマンドには、処理の対象として選択する行をフィルタリングできる**where**オプションがあります。デフォルトでは、これらのコマンドはテーブルのすべての行に影響し、ソーステーブルに含まれていないターゲットテーブルの列は無視されます。

- **where**オプションを使用すると、ソーステーブルとターゲットテーブルの同じ名前の列に基づいて行をフィルタリングすることができます。
- 垂直分割レプリケーションを使用するテーブルには**where**オプションを使用します。ソース列とターゲット列には異なる名前を付けることができます。**where**の選択をソース列に基づいて行います。SharePlexは、ターゲットテーブルの正しいWHERE句を構築するために、設定ファイルから列マッピングを読み取ります。
- ソーステーブルまたはターゲットテーブルに1つ以上の余分な列が存在し、それらの行に行の一意性を決定する値が含まれている場合は、**sourcewhere**および**targetwhere**オプションを使用します。
  - ソーステーブルに余分な列が含まれている場合は、**sourcewhere**を使用します。
  - ターゲットテーブルに余分な列が含まれている場合は、**targetwhere**を使用します。

  このオプションを正しく使用するには:
  - 余分な列には**sourcewhere**または**targetwhere**オプション*のみ*を使用します。
  - ソースとターゲットの両方で同じ名前を持つ他の列には、標準の**where**オプションを使用します。
  - SharePlexは、**where**オプションと**sourcewhere**または**targetwhere**オプションを組み合わせて、完全なWHERE句を作成します。

> **重要!** 余分な行を持つターゲットテーブルに対してcompareとrepairの両方を実行する予定がある場合は、UPDATEとDELETEのcompare用に**targetwhere**のみを使用してください。**repair**コマンドはINSERTの正しい値を決定できません。この問題を回避するには、余分な列にデフォルト値を設定するか、挿入された行を手動で更新します。

プロセスの特定

compareまたはrepairコマンドが発行されるたびに、ジョブIDが**sp_ctrl**ディスプレイに表示されます。**sp_ctrl**ディスプレイを利用できない場合は、**compare status**コマンドを実行することでジョブIDを表示できます。

sp_ctrlでステータスと結果を表示

repairのステータスや結果を表示するには、**sp_ctrl**の**repair status**コマンドを使用します。

- 基本コマンドは、最近開始されたrepairジョブの処理ステータスと、まだ終了していないその他のジョブを表示します。
- 追加オプションを使用すると、履歴のあるすべての修理ジョブのサマリーステータスを表示したり、1つのジョブに関する詳細情報を表示したりすることができます。

詳細については、repair statusを参照してください。

警告とエラーの表示

sp_xdesvrおよび**sp_xdecltプロセス**は、実行するシステム上にログファイルを書き込みます。SharePlexの変数データディレクトリの**log**サブディレクトリにあるファイルを開きます。

sp_xdesvrプロセスが書き込むログの名前は`xdesvr_<jobid>_r.<dbid>_p<process id>.log`です。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブIDです。
- *DBID*はPostgreSQLインスタンスのデータベースIDです。
- *ProcessID*は**sp_xdesvr**プロセスのプロセスIDです。

sp_xdecltプロセスが書き込むファイル名は、`xdeclt_<jobid>-<tableid>_r.<dbid>_p<processid>.log`に**.log**または**.sql**を付加したものです。ここで、

- *JobID*は、SharePlexによって割り当てられたジョブのジョブIDです。
- *TableID*は、SharePlexによって割り当てられたジョブ内のテーブルのテーブルIDです。
- *DBID*はPostgreSQLインスタンスのデータベースIDです。
- *SourceHost*は、ソースホストの名前またはIPアドレスです。
- *ProcessID*は**sp_xdecltプロセス**のプロセスIDです。

ログファイル名の例:

```
xdesvr_7_r.aparopka_p4970.log

xdeclt_7-1_r.aparopka_p25095.log

xdeclt_7-1_r.aparopka_p25095_01.sql
```

ディスク使用量を制御するために、ログは循環方式でエージングされます。現在のログがサイズの限界に達すると、SharePlexは新しいログファイルを生成します。新しいログは最大ログ数まで作成され、その後、SharePlexは最も古いログから上書きを開始します。

repairジョブのキャンセル

実行中のcompareまたはrepairジョブを停止するには、**cancel**コマンドを使用します。

```
sp_ctrl(sysA)>cancel JOBID
```

詳細については、「cancel」を参照してください。

compare履歴とログの管理

SharePlexは、終了した各ジョブの履歴をソースシステムのデータベースに保持します。SP_SYS_JOB_HISTORY_RETENTIONパラメーターは、履歴の保持期間を制御します。

この履歴を必要に応じて消去するには、**clear history**コマンドを使用します。SharePlexがジョブの履歴を削除すると、履歴の元となったログファイルも削除されます。

データベースのジョブ履歴を消去せずにソースシステムからログファイルを削除するには、**remove log**コマンドを使用します。このコマンドは、ターゲットシステムから古いログファイルを削除するためにも使用できます。

ログファイルのサイズを制御するには、SP_DEQ_LOG_FILESIZEパラメーターを設定します。

バッチサイズのコントロール

プロセスがSELECTクエリを実行するときにフェッチされる行のブロックのサイズを制御することができます。ブロックサイズは、**SP_DEQ_MALLOC**パラメーターで設定された値に基づいて計算されます。この値は、使用されるcompareスレッドの数で等分され、その後、すべての列のサイズを足した値に基づいて再計算されます。

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	オペレーター（2）
発行場所:	ソースシステム
関連コマンド:	compare / compare using

構文

基本コマンド	コマンドオプション	リモートオプション
repair *owner.source_table [.partition]*	[**at** *target_host*@**r.***target_dbid*] | [**for r.***source_dbid*] | [{**include | exclude**} "*column_list*"] | [**insertonly**] | [**key**] | [**nosqllog**] | [**not** "*exception_list*"] | [**onepass**] | [**orderby** "*column_list*"] | [**parallelism** *degree*] | [**sourcewhere** "*clause*"] | [**threads** *thread_count*] | [**targetwhere** "*clause*"] | [**to** target_owner.target_table [.partition]] | [**where** "*clause*"]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]
repair using *filename*	[**key**] |	[**on** *host* |

基本コマンド	コマンドオプション	リモートオプション
	[**onepass**] | [**port** *port_number*] | [**threads** *threads_count*] | [**parallelism** *degree*]	**on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

構文の説明

必須コマンドコンポーネント

コンポーネント	説明
repair *owner.source_table[.partition]*	基本コマンドは、すべてのソース行をすべてのターゲット行と共にrepairします。 *owner.source_table*は、ソーステーブルの所有者と名前です。大文字と小文字を区別したり、名前の中にスペースを入れたりするには、二重引用符を使います。例えば、**"HR".emp**。 ワイルドカードのテーブル名（オーナー名は不可）がサポートされています。repairするためには、このコマンドのワイルドカードを満たすテーブルがアクティブレプリケーション設定に（明示的またはワイルドカードで）リストされていなければなりません。SharePlexがワイルドカードをどのように処理するかについては、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl(sysA)>`**repair scott.emp**
repair using *filename*	基本コマンドは、*filename*にリストされているターゲットテーブル内の同期していない行をすべてrepairします。 *filename*は、ターゲットをrepairするソーステーブルの名前が含まれているファイルの名前です。 **例** `sp_ctrl(sysA)>`**repair using sales**

オプションのコマンドコンポーネント

コンポーネント	説明
at ***target_host@r.target_database_name***	**repairに有効** ソーステーブルが複数のターゲットシステムにレプリケートされる設定では、ターゲットテーブルの1つだけをrepairします。 *target_host*はターゲットシステムの名前です。

コンポーネント	説明
	*target_db*はターゲットデータベース名の名前です。 **例** `sp_ctrl(SysA)>`**compare scott.emp at prod@r.database_name**
for r.*DBID*	**compareに有効** ソーステーブルを含むPostgreSQLインスタンスを指定します。システム上の複数のPostgreSQLインスタンスに同じソーステーブルがある場合に使用します。 *DBID*はPostgreSQLソースインスタンスのデータベース名です。大文字と小文字は区別され、**config**ファイルに表示される通りに入力する必要があります。 このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**compare scott.emp for r.database_name**
{**include** | **exclude**} "(*column_list*)"	**repairに有効** repairする列をフィルタリングします。 • **include**を使って、repairする列を指定します。他の列はrepairされません。**include**句には、すべてのキー列を含める必要があります。 • **exclude**で指定した列を除くすべての列をrepairするには、**exclude**を使用します。キー列を除外しないでください。 (*column_list*)は、インクルードまたは除外する列のリストです。 • それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションと共に任意の順序で記述できます。 **注:** repairされなかった列には、まだ同期していない行が残っている可能性があります。 **例**

コンポーネント	説明
	`sp_ctrl (SysA)>`**repair scott.emp exclude "color, weight"**
insertonly	**repairに有効** INSERTステートメントの場合にのみターゲットテーブルをrepairします。 **例** `sp_ctrl(SysA)>`**repair scott.emp insertonly**
key	**repairとrepair usingに有効** 大きなテーブルのcompareとrepairを高速で実行します。このコマンドは、すべてのデータ値をcompareするのではなく、以下のうちの1つだけをcompareします。 • PRIMARYキー列またはNULLでないUNIQUEキー列のみ。 または... • **orderby**オプションで指定された列。テーブルにキーがない場合は、このオプションを使用します。 キーまたは**orderby**列が一致しない場合、SharePlexは、行全体を削除してrepairし、ソース値に基づいて再度挿入します。 **重要!** このオプションの使用には注意が必要です。キーの値が一致していても、キー以外の列の値が同期していない可能性があります。 このオプションを使用する場合は、必須コマンド引数の後に記述しなければなりません。他のオプションと共に任意の順序で表示できます。 このオプションは、SharePlexキー定義に基づくrepairには使用しないでください。SharePlexキー定義の詳細については、『SharePlex管理ガイド』を参照してください。 **例** `sp_ctrl (SysA)>`**repair scott.emp key** `sp_ctrl(sysA)>`**repair using sales key**
nosqllog	SQLログファイルの出力を抑制します。このファイルには、同期していない行をrepairするために必要なSQLが含まれています。このファイルを出力しない理由には、以下のようなものがあります。 • データには機密情報が含まれています。SQLログファイルはクリアテキストで書き込まれます。SQLログファイルを生成しないことで、機密データはディスクに保存されず、PCIコンプライアンス基準を満たすために要求されるような保存データのセキュリティ要件を満たすことができます。

コンポーネント	説明
	• compareまたはrepairされたテーブルには、非常に多くの非同期行があります。このサイズのログファイルは、大量のディスクスペースを消費する可能性があります。
not “*exception_list*”	**repairに有効** テーブル指定にワイルドカードが含まれている場合に、repairしないテーブルの例外リストを指定します。 "*exception_list*"は、repairしないテーブル名のリストです。 • *owner.tablename*形式を使用します。 • それぞれの名前はカンマで区切ります。リストにスペースは許容されません。 • リストを二重引用符で囲みます。 • テーブルを任意の順序でリストします。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl(SysA)>`**repair scott.% not (%temp%)**
onepass	**repairとrepair usingに有効** compareとrepairを同時に実行するには、このオプションを使用します。同期していない大きなテーブルに使用します。 通常、repairは2つのパスで実行されます。まずcompareを行い、次にrepairでターゲットテーブルをロックします。どちらのパスにも一貫したビューが必要です。**onepass**では、compareクライアントが一貫性のあるビューマーカーを受け取るとすぐに、ターゲットテーブルがロックされ、repairされます。 **例** `sp_ctrl(SysA)>`**repair scott.emp onepass** **注意:** PostgreSQLでは、テーブルに複数のキーが定義されている場合、「onepass」コマンドオプションを使用すると、repair処理中に重複制約違反エラーが発生することがあります。repairはこのオプションなしで正しく機能するため、このようなテーブルには「onepass」コマンドオプションを使用しないことを推奨します。 単一のキーが定義されたテーブルでは、「onepass」オプションによるrepairも適切に機能します。 キーが定義されていないテーブルの場合、「onepass」オプションによるrepairは、重複レコードがなければ適切に機能します。これは、同じ値を持つ複数のレコードがターゲットテーブルにある可能性があり、DELETEステートメントがWHERE句のキーとしてすべての列の

コンポーネント	説明
	値または部分的な列の値を操作するからです。その結果、期待された補正が行われない可能性があります。
orderby "*column_list*"	**repairに有効** ORDERBY句コマンドオプションとKEYコマンドオプションを併用し、ORDERBYオプションで指定された列をcompareします。キーまたはORDERBY列が一致しない場合、SharePlexは行全体を削除し、次にソース値に基づいて再度挿入することでrepairします。 repairプロセスでcompareする行をソートする際に、ORDERBY句で使用する列を指定します。このオプションは、プライマリキーや一意キーを持たないテーブル上でrepairを実行できるようにします。 "*column_list*"は、ORDERBY句で使用する列の名前です。 • それぞれの名前はカンマで区切ります。列名にスペースが含まれていない限り、リストにスペースは許可されません。 • 列リストは二重引用符で囲みます。 • 列を任意の順序でリストします。ソートは昇順で実行されます。 • 列名は大文字と小文字を区別しません。 • このオプションを使用する場合、コマンドの必須引数の後に記述しなければなりませんが、他のオプションと共に任意の順序で記述できます。 • オペレーティングシステムのコマンドラインからrepairを実行する場合、引用符で囲まれた文字列は、以下のようにエスケープされた二重引用符で囲む必要があります。 */productdir/***bin/sp_ctrl repair scott.emp orderby** “\“**Last Name**,**Division**\”” **例** `sp_ctrl(SysA)>`**repair scott.emp where “file >001005” orderby “Last Name,Division”**
parallelism *degree*	**repairとrepair usingに有効** SELECTステートメントに並列ヒントを追加します。*degree*では、並列度を設定します。 **例** `sp_ctrl(sysA)>`**repair scott.emp parallelism 4** `sp_ctrl(sysA)>`**repair using sales parallelism 4**
sourcewhere “*clause*”	**repairに有効** ターゲットテーブルに列が存在しない場合に、ソーステーブル内の1つ

コンポーネント	説明
	または複数の列を基にrepairを行います。ソーステーブル上でこの条件によってフィルタリングされた行はロックされます。 • *clause*を二重引用符で囲み、**scott.emp**のように完全修飾名でテーブルを参照します。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、PostgreSQLのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • オペレーティングシステムのコマンドラインからrepairを実行する場合、引用符で囲まれた文字列は、以下の例のように*エスケープされた*二重引用符で囲む必要があります。 `sp_ctrl>`**repair scott.emp sourcewhere "\"file >001005\""** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例#1**: `sp_ctrl(sysA)>`**repair scott.emp sourcewhere "file >001005"** **例#2**: 次の例は、**sourcewhere**および**where**オプションをどのように組み合わせて、望ましい結果を得るかを示しています。ソースrepairプロセスのみが**sourcewhere**句を使用しますが、ソースrepairプロセスとターゲットrepairプロセスの両方が**where**句を使用します。 `sp_ctrl(SysA)>`**repair scott.emp sourcewhere "deptno = 200" where "mgr = 'SMITH'"**
targetwhere "*clause*"	**repairに有効** ソーステーブルに列が存在しない場合に、ターゲットテーブル内の1つまたは複数の列を基にrepairを行います。ターゲットテーブル上でこの条件によってフィルタリングされた行はロックされます。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。

コンポーネント	説明
	• 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、PostgreSQLのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句"c1 > '10-SEP-2001'"は機能しませんが、"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"は機能します。 • オペレーティングシステムのコマンドラインからcompareを実行する場合、引用符で囲まれた文字列には、以下のようにエスケープされた二重引用符の余分なセットが必要です。 */productdir/***bin/sp_ctrl repair scott.emp targetwhere "\"file >001005\""** • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例#1**: sp_ctrl(SysA)> **repair scott.emp targetwhere "file >001005"** **例#2**: 次の例は、**targetwhere**および**where**オプションをどのように組み合わせれば、望ましい結果が得られるかを示しています。ターゲットrepairプロセスのみが**targetwhere**句を使用しますが、ソースrepairプロセスとターゲットrepairプロセスの両方が**where**句を使用します。 sp_ctrl(SysA)>**repair scott.emp where "deptno = 200" targetwhere "mgr = 'SMITH'" repair**
threads *thread_count*	**repairとrepair usingに有効** repairプロセスが使用する処理スレッドの数を設定します。 **例** sp_ctrl(sysA)>**repair scott.emp threads 4** sp_ctrl(sysA)>**repair using sales threads 4**
to *target_owner.target_table [.partition]*	**repairに有効** ソーステーブルのターゲットの1つだけをrepairします。ソーステーブルが複数のターゲットシステムにレプリケートされ、ターゲットテーブルに異なる名前がある場合に使用します。 このオプションはターゲットパーティションを指定するためにも使用できます。 compare source_owner.source_table.[source_partition] to target_owner.target_table.[target_partition]

コンポーネント	説明
	例 （パーティションのrepair） `sp_ctrl(SysA)>`**repair scott.emp.east to scott.allemp.alleast**
where “*clause*”	**repairに有効** ソースシステムとターゲットシステムの両方で、SELECTステートメントにWHERE句を含めます。WHERE句は、特定の行をrepairするためのフィルタとして機能します。この条件によってフィルタリングされた行はロックされます。 “*clause*"には、サブクエリを含まない標準のWHERE句を指定します。 • clauseを二重引用符で囲み、完全修飾名でテーブルを参照します。例えば、**scott.emp**。 • 大文字と小文字を区別したり、テーブル名内にスペースを入れたりする場合は、二重引用符を使用します。 • 日付は「YYYYSMMDDHH24MISS」の形式でなければなりません。日付をこの形式に変換するには、PostgreSQLのTO_DATE関数を使用します。例えば、c1がDATE列の場合、WHERE句`"c1 > '10-SEP-2001'"`は機能しませんが、`"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"`は機能します。 • このオプションを使用する場合は、必要なコマンド引数の後に記述する必要がありますが、他のオプションと共に任意の順序で記述できます。 **例** `sp_ctrl (SysA)>`**repair scott.emp where “region=4”**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

PostgreSQLのrepair statusコマンド

repair statusコマンドを使用して、最後に実行されたcompareまたはrepairジョブのステータスを表示します。**repair statusコマンド**を使用すると、compareジョブやrepairジョブ、またはcompareジョブやrepairジョブの一部に関する詳細なステータスを表示したり、SharePlexに履歴があるすべてのcompareジョブおよびrepairジョブのステータスを表示したりできます。

repair statusコマンドの詳細および使用例については、**job statusコマンド**を参照してください。

用途

サポート対象のソース:	PostgreSQL
サポート対象のターゲット:	PostgreSQL
認証レベル:	オペレーター(2)
発行対象:	ソースシステム
関連コマンド:	**compare status**、**job status**

構文

基本コマンド	コマンドオプション	リモートオプション
repair status	[*job_id*]	[**on** *host* |
	[*job_id.table_id*]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |
	[**full**]	**on** *login/password@host:portnumber*]
	[**detail**]	
	[*status*]	

構文の説明

コンポーネント	説明
job_id	指定されたSharePlex割り当てジョブIDを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)` > **repair status 2828.2**
job_id.table_id	指定されたSharePlex割り当てジョブIDとテーブルを持つジョブのステータス履歴を表示します。 **例**: `sp_ctrl(sysA)` > **repair status 2828.HR.SRC_TEST3**
all	データベースに履歴があるすべてのジョブの概要行を表示します。 **例**: `sp_ctrl(sysA)` > **repair status all**

コンポーネント	説明
full	ジョブ内の各オブジェクトのステータスを表示します。デフォルトでは、job statusコマンドは、完了していないオブジェクト、または例外的なステータスで完了したオブジェクトのステータスを表示します。 **例**: `sp_ctrl(sysA)` > **repair status2828 full**
detail	報告されたすべてのオブジェクトの詳細情報を表示します。デフォルトでは、job statusコマンドは報告されたすべてのオブジェクトの概要行を表示します。詳細情報は、job_id.table_idオプションで表示されるものと同じであることに注意してください。 **例**: `sp_ctrl(sysA)` > **repair status detail**
status	指定されたステータスの過去のジョブのステータス履歴を表示します。 **例**: `sp_ctrl(SysA)` > **repair status "Error"**

リモートオプション

これらのオプションにより、リモートマシンにコマンドを発行したり、ログイン名、パスワード、ポート番号、またはそれらの組み合わせを含むコマンドをスクリプト化したりすることができます。

オプション	説明
on *host*	リモートシステム(現在の**sp_ctrl**セッションが実行されているシステム以外)でコマンドを実行します。リモートシステムのログイン認証情報の入力を求めるプロンプトが表示されます。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	リモートログインとポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	リモートログイン、パスワード、ホスト名を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	リモートログイン、パスワード、ホスト名、ポート番号を指定する必要がある場合は、リモートシステムでコマンドを実行します。使用する場合は、コマンド構文の最後の構成要素でなければなりません。 **例:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

PostgreSQLのverify config

verify configコマンドの目的は、アクティベーションとレプリケーションの特定の問題を回避するための予防措置として使用することです。これは、アクティベーションが正常に完了することを確認するためのテストに使用することを意図しています。

verify configコマンドは、テーブルのみを検証します。

このコマンド以下に使用できます。

- 設定ファイル内のエントリの構文を確認する。
- SharePlexによるレプリケーションでソースオブジェクトがサポートされていない場合にエラーを報告する。
- ルートで指定されたホスト名に到達できない場合に報告する。
- 1つのオブジェクトに重複した指定がある場合に報告する。
- オブジェクトの指定がスキップされるかどうかと、その理由を報告する。

verify configコマンドがサポートしていないもの

verify configは以下を実行しません。

- アクティベーション時刻の検証
- ターゲットオブジェクトまたはターゲットデータベース名の検証

アクティブな設定で追加または変更されたオブジェクトの検証

アクティブな設定に追加するオブジェクトや変更するオブジェクト（ルーティングの変更など）を検証するには、アクティブな設定をコピーして変更し、そのコピーに対して**verifyコマンド**を実行することをお勧めします。

検証結果の表示

verify configコマンドは、検証が完了するまでsp_ctrlインターフェイスの制御を保持します。

このコマンドは、設定ファイル全体を読み込み、エラーを記録します。

検証の結果は**sp_ctrl**内の画面に表示されます。詳細な結果を表示するには、以下を実行します。

- **sp_ctrl**で、**detail**オプションを使用してverify**config**コマンドを発行します。
- **verify config**コマンドを発行した後、画面に表示されるパスを使用して結果ファイルに直接移動します。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL

認証レベル:	ビューアー(3)
発行対象:	ソースシステム
関連コマンド:	**activate config**

構文

基本コマンド	コマンドオプション
verify config *filename*	**detail**

サポートされるワイルドカード構文

SharePlexは、テーブル名に対して以下のSQLワイルドカードをサポートしています。

- パーセント(%)ワイルドカードで文字列を指定します
- アンダースコア(_)ワイルドカードで1文字を指定します。

サポートされているワイルドカード構文の詳細については、ワイルドカードを使用した複数のテーブルの指定を参照してください。

構文の説明

コンポーネント	説明
filename	*filename*は、検証する設定の名前です。
detail	このオプションは、より詳細なレベルを画面に表示します。 **例**: `sp_ctrl(sysA)` > **verify config myconfig detail** この例では**myconfig**ファイルが検証され、その結果がより詳細なレベルで表示されます。

PostgreSQLのview partitions

view partitionsコマンドを使用して、水平分割レプリケーション設定内の1つのパーティションスキームまたはすべてのパーティションスキームの行パーティションを表示します。

水平分割レプリケーションの設定方法の詳細については、『SharePlex管理ガイド』を参照してください。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行場所:	ソースシステム
関連コマンド:	add partition、drop partition、drop partition scheme、modify partition

構文

基本コマンド
view partitions for {*scheme_name* | **all**}

構文の説明

コンポーネント	説明
scheme_name	指定されたパーティションスキームの行パーティションを表示します。
all	すべての行パーティションを、パーティションスキームの名前に従ってグループ化して表示します。

例

```
sp_ctrl> view partitions for scheme1
sp_ctrl> view partitions all
```

PostgreSQLのshow capture

Captureプロセスの統計情報を表示するには、**show capture**コマンドを使用します。

基本コマンド

基本的な**show capture**コマンドは、データソース、プロセスが実行中か停止中、その他の基本情報など、プロセスの概要を示します。

詳細な統計情報

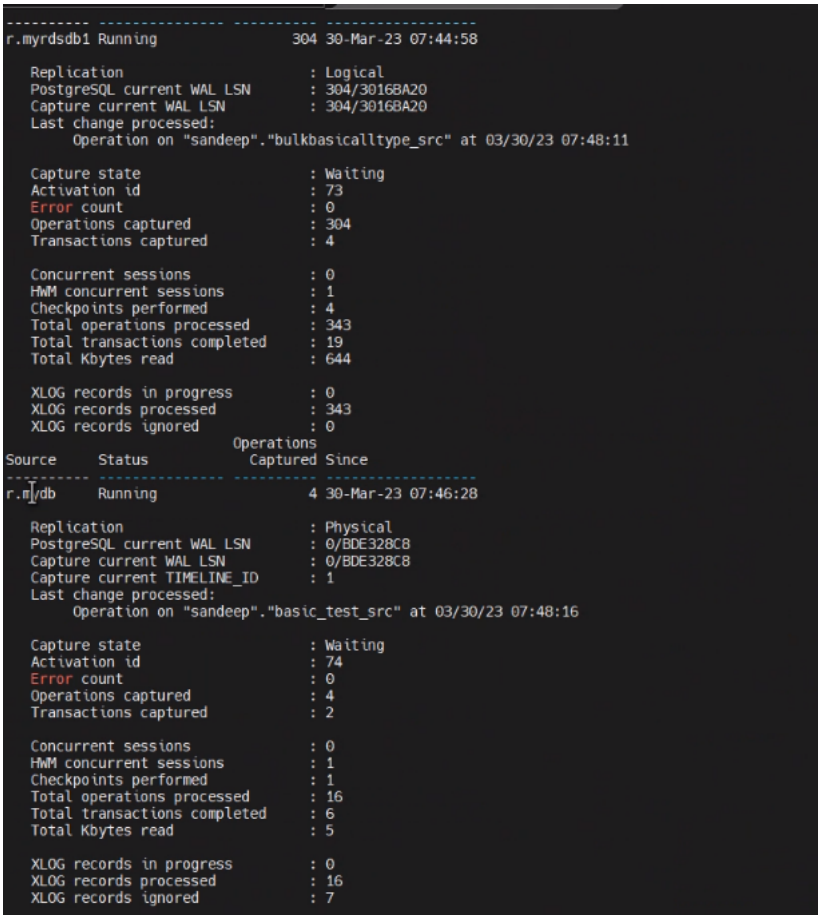
Captureプロセスの詳細な統計情報を表示するには、**[detail]**オプションを指定した**show capture**コマンドを使用します。このオプションは、プロセスのパフォーマンスを評価し、チューニングパラメーターを調整する必要があるかどうかを決定し、問題やボトルネックを検出するのに役立つ詳細な統計情報を表示します。

PostgreSQL Captureの詳細な統計情報

統計情報	説明
Host（ホスト）	ローカルマシンの名前（ソースシステム）
System time（システム時刻）	システムクロックによる現在の時刻。
Source（ソース）	ソースPostgreSQLデータベースの名前。
Status（ステータス）	Captureプロセスのステータス（実行中または停止中）。
Since（開始時刻）	Captureが開始された時刻。
PostgreSQL current WAL LSN（PostgreSQLの現在のWAL LSN）	PostgreSQLが書き込んでいるWALファイルログのLSN（ログシーケンス番号）。
Capture current WAL LSN（Captureの現在のWAL LSN）	Captureが読み込んでいるWALファイルログのLSN（ログシーケンス番号）。 この値は、データがレプリケートされたテーブルからのものかどうかに関係なく、Captureが読み込んだ最新のLSN値を示す必要があります。アイドル状態の場合、PostgreSQLの現在のWAL LSNと一致する必要があります。
Last WAL file record processed（最後に処理されたWALファイルレコード）	Captureが処理中のレコード、またはCaptureが現在データをレプリケートしていない場合は最後に処理されたレコード。
Capture state（Captureの状態）	プロセスが実行するレプリケーション作業に関連したプロセスの状態：以下のいずれかになります。 • WAITING（待機中）：CaptureはWAL送信者からのレコードを待っています。 • PROCESSING（処理中）：Captureはレプリケーションのために WALファイルのログレコードを処理しています。 • STOPPED BY ERROR（エラーにより停止）：Captureがエラーにより停止し、イベントログにエラーが記録されています。

統計情報	説明
Activation ID（アクティベーションID）	設定のアクティベーションの内部識別番号です。関連するプロセスとキューを識別します。この値は、アクティベーション直後の、DMLレプリケーションが開始される前に表示される必要があります。
Error count（エラー数）	Captureの開始後、PostgreSQLのエラーによりスキップされたレコードの数。スキップされたレコードのデータはターゲットデータベースに反映されません。
Operations captured（キャプチャされたオペレーション数）	開始後に、Captureがレプリケーション用に正常に処理したDML操作の数。
Transactions captured（キャプチャされたトランザクション数）	開始後に、Captureがオペレーションを正常にレプリケートしたコミット済みPostgreSQLトランザクションの数。
Concurrent sessions（同時セッション数）	同時に処理されているPostgreSQLセッションの数。
HWM concurrent sessions（HWM同時セッション数）	Capture開始後の同時PostgreSQLセッションの最大数。
Checkpoints performed（実行されたチェックポイント数）	Capture開始後のCaptureの状態を保存するチェックポイントの数。頻繁にチェックポイントを実行すると、システムにさらなるオーバーヘッドが発生しますが、チェックポイントの頻度が低いと、SharePlexでのシステムやインスタンスの障害からのリカバリが遅くなります。デフォルトでは、Captureは40,000件のメッセージごと、または120秒ごとにチェックポイントを実行しますが、`SP_OCT_CHECKPOINT_FREQ`パラメーターおよび`SP_OCT_CHECKPOINT_TIME`パラメーターを使用して調整できます。
Total operations processed（処理されたオペレーションの総数）	レプリケーション用にキャプチャされたレコードや、設定に含まれていないオブジェクトのレコード（レプリケートされたレコードとレプリケート対象外のレコードの両方）など、Captureが開始後に処理したすべてのPostgreSQL操作とSharePlex内部操作の数。
Total transactions completed（完了したトランザクションの総数）	レプリケーション用にキャプチャされたトランザクションや、レプリケーション設定に含まれていないオブジェクトのトランザクション（レプリケートされたトランザクションとレプリケート対象外のトランザクションの両方）など、Captureが開始後に処理したコミット済みPostgreSQLトランザクションの数。
Total Kbytes read（読み取られたキロバイト総数）	Captureが開始後に処理したデータのキロバイト単位のサイズ。
XLOG records in progress（処理中のXLOGレコード数）	Captureが処理中のレコードの数。
XLOG records processed（処理したXLOGレコード数）	処理したXLOGレコードの総数。
XLOG records ignored（無視したXLOGレコード数）	設定内のオブジェクトに関連付けられていないためにCaptureが無視したレコードの数。
Replication（レプリケーション）	レプリケーションの種類（物理または論理）
Capture current TIMELINE_ID（Captureの現在のTIMELINE_ID）	現在のタイムラインIDを表示（物理レプリケーションにのみ適用）

PostgreSQL Captureのサンプル統計情報の例を以下に示します。



```
---------- --------------- ---------- ------------------
r.myrdsdb1 Running                304 30-Mar-23 07:44:58

   Replication                     : Logical
   PostgreSQL current WAL LSN      : 304/3016BA20
   Capture current WAL LSN         : 304/3016BA20
   Last change processed:
        Operation on "sandeep"."bulkbasicalltype_src" at 03/30/23 07:48:11

   Capture state                   : Waiting
   Activation id                   : 73
   Error count                     : 0
   Operations captured             : 304
   Transactions captured           : 4

   Concurrent sessions             : 0
   HWM concurrent sessions         : 1
   Checkpoints performed           : 4
   Total operations processed      : 343
   Total transactions completed    : 19
   Total Kbytes read               : 644

   XLOG records in progress        : 0
   XLOG records processed          : 343
   XLOG records ignored            : 0
                          Operations
Source     Status           Captured Since
---------- --------------- ---------- ------------------
r.m/db     Running                 4 30-Mar-23 07:46:28

   Replication                     : Physical
   PostgreSQL current WAL LSN      : 0/BDE328C8
   Capture current WAL LSN         : 0/BDE328C8
   Capture current TIMELINE_ID     : 1
   Last change processed:
        Operation on "sandeep"."basic_test_src" at 03/30/23 07:48:16

   Capture state                   : Waiting
   Activation id                   : 74
   Error count                     : 0
   Operations captured             : 4
   Transactions captured           : 2

   Concurrent sessions             : 0
   HWM concurrent sessions         : 1
   Checkpoints performed           : 1
   Total operations processed      : 16
   Total transactions completed    : 6
   Total Kbytes read               : 5

   XLOG records in progress        : 0
   XLOG records processed          : 16
   XLOG records ignored            : 7
```

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ソースシステムおよびターゲットシステム
関連コマンド:	**show post**

構文

基本コマンド	コマンドオプション
show capture	[**detail**] [**for***datasource*]

構文の説明

コンポーネント	説明
show capture	Captureプロセスの状態とキャプチャされた操作の概要を表示します。
detail	Captureのパフォーマンスを調整したり、問題を診断したりするのに役立つ詳細な統計情報が表示されます。 **例**: `sp_ctrl(sysA)` > **show capture detail**
for *datasource*	このオプションは、特定のデータソースのCapture統計情報のみを表示します。 *datasource*は、**r**.*database*で表されます。ここで、*database*はdbnameです。 **例**: `sp_ctrl(sysA)` > **show capture for r.dbname**

PostgreSQLのshow post

Postプロセスの統計情報を表示するには、**show post**コマンドを使用します。

基本的なshow postコマンド

基本的な**show post**コマンドは、Postプロセスの全セッションのグローバル統計情報を表示します。Postプロセスのステータスと、そのプロセスが開始されてからポストされたメッセージの数が表示されます。特定のpostキューまたはデータソースの出力をフィルタリングするには(複数のレプリケート対象データストリームがある場合に便利)、**queue** *queuename*または**for** *datasource-datadest*オプションを使用します。

詳細なshow postコマンド

Postプロセスの詳細な統計情報を表示するには、**show post**コマンドに**detail**オプションを指定します。このオプションは、処理された最新のSQLステートメントと、Postのパフォーマンスを評価し、チューニングパラメーターを調整する必要があるかどうかを決定し、問題やボトルネックを検出するのに役立つその他の統計情報を表示します。

以下では、**show post**で表示される詳細な統計情報の例について説明します。これらの統計情報は、ソースとターゲットのタイプによって若干異なります。

統計情報	説明
Host(ホスト)	ローカルマシンの名前(ターゲットシステム)
Source(ソース)	Postが処理するデータのソース。
Queue(キュー)	このPostプロセスのPostキュー。デフォルトのPostキューの場合は、ソースシステムの名前です。名前付きキューの場合は、ユーザ定義名です。
Target(ターゲット)	このPostプロセスのターゲットの名前。例えば、PostgreSQLインスタンスまたはOpen Targetデータベースの名前です。
Status(ステータス)	Postプロセスのステータス(実行中または停止中)。可能なステータスは以下の通りです。 • 実行中 • 停止中 • ユーザが停止 • エラーにより停止
Operations posted(ポストされた操作数) **Operations processed(処理された操作数)**	このPostプロセスが開始後に処理したトランザクション操作とSharePlex内部操作の数。
Since(開始時刻)	Postが開始された時刻
Total(合計)	キュー内のまだ読み取り-リリースされていないメッセージの数。この数字は、**qstatus**を実行して返された「メッセージ数」に相当します。(TOTALの値は時間とともに減少し、QSTATUSのメッセージ数と同じ値を示します)
Backlog(バックログ)	Postが処理するのをキューで待機しているメッセージの数。

統計情報	説明
Last operation posted（ポストされた最後の操作）	Postがアクティブな場合はターゲットにポストされている最新の操作、非アクティブな場合は最後にポストされた操作の識別情報。この情報は、データを生成したデータストアのタイプに固有です。操作の内容は以下の通りです。 • INSERT • UPDATE • DELETE • COMMIT • INSERT_MULTIPLEまたはDELETE_MULTIPLE（配列/一括操作）。 • SharePlex内部操作。
Last transaction posted（ポストされた最後のトランザクション）	最後にポストされたトランザクションの識別情報。この情報は、データを生成したデータストアのタイプに固有です。
Post state（Postの状態）	実行対象のレプリケーション作業に関連するPostプロセスの状態。以下のいずれかが該当します。 • **Waiting（待機中）**: Postはメッセージが処理されるまで待機しています。 • **Active（アクティブ）**: Postは変更をデータベースにポストしています。 • **Committed（コミット済み）**: Postはトランザクションをコミットしています。 • **Idle（アイドル）**: Postには処理する未決済トランザクションがありません。 • **Rollback（ロールバック）**: Postはロールバックを処理中です。 • **Recovery（リカバリ）**: Postはクラッシュリカバリモードになっています。
Activation ID（アクティベーションID）	現在の設定のアクティベーションID。この値は、アクティベーション直後の、DMLレプリケーションが開始される前に表示される必要があります。
Operations processed（処理された操作数）	COMMITを受け取ったかどうかにかかわらず、Postがターゲットに適用したSQL操作の数。
Transactions processed（処理されたトランザクション数）	開始以後にPostがターゲットに適用したコミット済みトランザクションの数。
Insert operations（INSERT操作数）	Postが開始後に処理したINSERT操作の数。
Update operations（UPDATE操作数）	Postが開始後に処理したUPDATE操作の数。
Delete operations（DELETE操作数）	Postが開始後に処理したDELETE操作の数。
レイテンシ	レプリケーションの処理に要した時間（ソースのデータベースが要した時間を除く）

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
発行対象:	ターゲットシステム
関連コマンド:	**show capture**

構文

基本コマンド	コマンドオプション
show post	[**detail**] [**queue** *queuename*] [**for***datasource-datadest*] [**sessions**]

構文の説明

コンポーネント	説明
show post	Postプロセスの状態と処理された操作の概要を表示します。
detail	このオプションはPostプロセスの詳細な統計情報を表示します。 **例**: `sp_ctrl(sysB)` > **show post detail**

queue*queuename*	このオプションは**show post**の表示を特定のpostキューに対してフィルタリングします。
	• **queue**は構文の必須部分です。
	• *queuename*はPost統計情報を表示したいpostキューです。有効な値:
	  • デフォルトキューを使用する場合のソースシステムの名前。
	  • 名前付きキューを使用している場合は、ユーザ定義のキュー名。
	キュー名が不明な場合は、**qstatus**コマンドを実行します。キュー名はすべてのプラットフォームで大文字と小文字が区別されます。
	このオプションは、他のオプションと共に任意の順序で表示できます。

コンポーネント	説明
	例: `sp_ctrl(sysB)` > **show post queue sysA**
for*datasource-datadest*	このオプションはshow postの表示を特定のデータストリームに対してフィルタリングします。 • **for**は構文の必須部分です。 • *datasource*は、**r**.*database*で表されます。ここで、*database*はdbnameです。 • *datadest*は、ターゲットによって以下のいずれかで表されます。 **r.***database_name* **x.kafka** **x.jms** **x.file** このオプションは、他のオプションと共に任意の順序で表示できます。 **例**: `sp_ctrl(sysB)` > **show post for r.dbnameA-r.ssB**
sessions	PostgreSQLターゲットの場合、このオプションはPostプロセスによって生成されたすべてのスレッドの統計情報を表示します。 シングルスレッドであるOpenターゲットの場合、このオプションを使用すると、そのスレッドの詳細を表示できます。 このオプションは、他のオプションと共に任意の順序で表示できます。 **例**: `sp_ctrl(sysB)` > **show post sessions queue *queuename***

show_last_posted

Postプロセスがターゲットに適用した最後のトランザクションのPostgreSQL LSNを表示するには、**show_last_posted**コマンドを使用します。このコマンドは、複数のpostキューを使用している場合、最後のLSNをすべて表示します。

```
sp_ctrl (sysB)>show last_posted

$> show_last_posted r.dbname1

For resume replication from r.dbname1

On source activate to pglsn=<LSN in hexadecimal format>

reconcile queue sp01 for r.dbname1-r.dbname1 pglsn <LSN1 in hexadecimal format>

reconcile queue sp02 for r.dbname1-r.dbname1 pglsn <LSN2 in hexadecimal format>

reconcile queue sp03 for r.dbname1-r.dbname1 pglsn <LSN3 in hexadecimal format>
```

注意: このコマンドは、LSN情報を取得するためにすべてのPostプロセスを停止します。

用途

サポート対象のソース:	PostgreSQL（オンプレミス）、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
サポート対象のターゲット:	PostgreSQL、Oracle、SQL Server、Kafka、Amazon RDS for PostgreSQL、Amazon Aurora for PostgreSQL、Azure Database for PostgreSQL Flexible Server、Google Cloud SQL for PostgreSQL
認証レベル:	ビューアー（3）
発行対象:	ターゲットシステム
関連コマンド:	**activate config**

構文

基本コマンド
show last_posted

4

PostgreSQL用SharePlexパラメーター

PostgreSQLパラメーターは、PostgreSQLデータベースのレプリケーションのさまざまな側面を制御し、調整します。

内容

- ユーザ設定可能なパラメーターの説明
  - Captureパラメーター
  - Open Target Posterパラメーター
  - Oracle Posterパラメーター
  - Readパラメーター
  - Compare/Repairパラメーター
  - Exportパラメーター
  - Importパラメーター
  - Queueパラメーター
  - sp_copパラメーター
  - Systemパラメーター

ユーザ設定可能なパラメーターの説明

このセクションでは、PostgreSQLデータベースのユーザ設定可能なSharePlexパラメーターについて説明します。この章に記載されていないパラメーターは内部パラメーターであり、Questの開発者またはテクニカルサポート担当者の指導の下でのみ変更してください。

パラメーターは以下のように分類されます。

Captureパラメーター

これらのパラメーターは、SharePlex Captureプロセスで使用されます。

SP_CAP_INCLUDE_UNCHANGED_COL

このパラメーターは、UPDATE操作の変更後イメージに未変更の列を含めるかどうかを制御します。デフォルトでは、SharePlexは変更された値のみを変更後イメージに含めます。

デフォルト: 0(オフ)

有効な値の範囲: 0または1

有効になるタイミング: 即時

SP_CAP_REDUCED_KEY

このパラメーターは、WALファイル内の操作のどの部分をCaptureによってターゲットに送信し、Postで使用するかを制御します。SharePlexの機能によって、Postプロセスで使用可能にする必要のあるデータの量が異なることがあります。

- 0に設定すると、このパラメーターは、PostgreSQLがWALファイルに書き込むTEXT以外のすべてのデータを送信するようにCaptureに指示します。この設定によりネットワーク経由で送信されるデータが最大になりますが、SharePlexの特定の機能をサポートするためには必要です。
- 1(デフォルト)に設定すると、このパラメーターは、キー値と、UPDATEの場合は操作で変更された列の変更前後の値を送信するようにCaptureに指示します。この設定はデフォルトです。この設定により、PostはWHERE句を作成する際に、変更された列の変更前後のcompareができるため、レプリケーションのパフォーマンスとターゲットデータ整合性のバランスが適切になります。SharePlexが変更前後の値をどのように使用するかについての詳細は、『SharePlex管理ガイド』を参照してください。
- 2に設定すると、このパラメーターは、キー値と、UPDATEの場合は、ターゲットに対して変更された列の変更後の値を送信するようにCaptureに指示します。この設定により、ネットワーク経由で送信されるデータ量が最小限になり、さらにPostがデータをポストする際にキー値のみを使用できるようになるため、パフォーマンスが向上します。

デフォルト: 1

有効な値の範囲: 0、1、2

有効になるタイミング: 即時

SP_CAP_ONELINE_DEBUG

このパラメーターは、Captureプロセスのオンラインデバッグ機能を制御します。

オンラインデバッグ機能により、システムリソースを大量に消費することなく、非同期エラーのデバッグを実行できます。オンラインデバッグでは、データストリームで非同期状態が発生した場所を検出するのに十分な情報を1行で記録します。問題を引き起こしているプロセスがオンラインデバッグで特定されたら、そのプロセスについて通常のデバッグを有効にすることができます。

このパラメーターを有効にするには、デバッグするオブジェクトをオブジェクトIDで指定し、それぞれをカンマで区切ります。例:

```
sp_ctrl>set sp_cap_oneline_debug 230230, 351626
```

以下の項目が記録されます。

- SharePlexが使用するトランザクション識別子
- 影響を受けたDMLまたはDDL操作のタイプ
- 影響を受けるソーステーブルのオブジェクトID
- 操作が行われた時刻(PostgreSQL)
- トランザクションのLSN値
- 冗長な操作を防ぐ内部SharePlexコード

デフォルト: 無効

有効な値の範囲: オブジェクトIDによりリストされるオブジェクトのリストを形成する文字列(カンマで区切る)

有効になるタイミング: 即時

SP_CAP_CHECKPOINT_FREQ

このパラメーターは、show captureコマンドによって発行されるチェックポイントの頻度を定義するために使用できるパラメーターの1つです。このパラメーターは、SP_CAP_CHECKPOINT_TIMEパラメーターと連動します。

デフォルト: 40,000メッセージ

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_CAP_CHECKPOINT_TIME

このパラメーターは、SP_CAP_CHECKPOINT_FREQパラメーターと連動します。Captureプロセスがチェックポイントを実行するまでの遅延時間(秒)を定義します。SP_CAP_CHECKPOINT_FREQに設定された値より前にこのパラメーターに設定された値に達すると、チェックポイントがトリガされます。(チェックポイントは、障害リカバリに必要な場合に備えてプロセスの状態を保存します。)

デフォルト: 120秒

有効な値の範囲: 任意の正の整数

有効になるタイミング: 即時

SP_CAP_DENIED_SESSION_PG

このパラメーターは、SharePlex PostgreSQL Captureにのみ適用されます。これは、Captureプロセスがユーザセッションのすべてのトランザクションを無視するか、フィルタリングするかを制御します。

0に設定すると、Captureは、ユーザセッションがオリジン名にバインドされているかどうかにかかわらず、トランザクションを無視しません。

1に設定すると、Captureは、'sp_deny_anyString'の形式でオリジン名にバインドされているユーザセッションのすべてのトランザクションを無視します。sp_deny_プレフィックスを使用する必要があり、「anyString」の代わりに文字列を挿入できます。**例:** sp_deny_1。

ユーザは、PostgreSQL関数の'pg_replication_origin_create ()'と 'pg_replication_origin_session_setup ()'を呼び出して、PostgreSQLユーザセッションを作成し(存在しない場合)、オリジン名にバインドする必要があります。

例:

- select pg_replication_origin_create('sp_deny_1');
- select pg_replication_origin_session_setup('sp_deny_1');

重要: ソースマシン上のトランザクションを無視すると、非同期の状態になることがあります。

デフォルト: 0

有効な値の範囲: 0または1

有効になるタイミング: 即時

このパラメーターの使用方法:

指定されたユーザがソースデータベースでトランザクションを実行する前に、ソースsp ctrlで以下のコマンドを発行します(ソースおよびターゲットとしてPostgreSQLを使用する場合)。

```
sp_ctrl>set param SP_CAP_DENIED_SESSION_PG 1
```

注意: パラメーターはライブです。このパラメーターを設定した後、ユーザはpg_replication_origin_session_setup()関数を呼び出して、PostgreSQLセッションをオリジンにバインドする必要があります。これ以降にソースデータベースに対して実行されるすべての操作は、パラメーターが設定解除されるまですべて無視されます。SP_CAP_DENIED_SESSION_PGはライブパラメーターであり、いつでも変更できます。

パラメーターを設定解除するには、以下のコマンドを実行します。

```
sp_ctrl>reset param SP_CAP_DENIED_SESSION_PG
```

異なるオリジンに10を超えるPostgreSQLセッションを同時にバインドするには、以下の手順を実行します。

1. PostgreSQLデータディレクトリ(cd /var/lib/pgsql/13/data/)に移動します。
2. postgresql.confファイルを編集し、max_replication_slotsパラメーターのコメントを解除します(デフォルトではコメント化されているため、コメントを解除し、異なるオリジンにバインドする同時セッションの最大数より大

きい値に設定する必要があります）。

3. データベースを再起動します。

注意:

- 有効なオリジンを追加するための構文は、sp_deny_nameです。ユーザ定義のオリジン名の前にsp_deny_プレフィックスを指定する必要があります。
- max_replication_slotsの値は、セッションセットアップを行おうとする同時ユーザセッション数より大きくする必要があります。[例えば、ユーザが100セッションのオリジンセットアップを行おうとする場合、max_replication_slotsの値を100より大きく設定します]

Open Target Posterパラメーター

これらのパラメーターは、Open Target（Oracle以外のターゲット）にデータを適用する際にSharePlex Postプロセスで使用されます。

SP_OPX_REDUCED_KEY

このパラメーターはPostのWHERE句の内容を制御します。PostはWHERE句を使用して、ソースから複製されたUPDATEによって変更する必要がある行をターゲットで見付けます。SharePlexの機能によって、PostのWHERE句で使用する必要のあるデータの量が異なることがあります。

- 0に設定すると、このパラメーターは、Captureから送信されるすべてのデータのWHERE句を作成するようにPostに指示します。Captureが送信するデータは、SP_OCT_REDUCED_KEYパラメーターの設定によって異なります。キーの値と、TEXT以外のすべての列をWHERE句に含めるには、両方のパラメーターを0に設定します。
- 1に設定した場合、このパラメーターは、変更された列のキー値と変更前の値でWHERE句を構築するようにPostに指示します。この設定により、PostはWHERE句を作成する際に、変更された列の変更前後のcompareができるため、レプリケーションのパフォーマンスとターゲットデータ整合性のバランスが適切になります。この設定を使用するには、SP_OCT_REDUCED_KEYパラメーターを0または1に設定する必要があります。SharePlexが変更前後の値をどのように使用するかの詳細については、『SharePlex管理ガイド』を参照してください。
- 2に設定すると、このパラメーターはキー列のみのWHERE句を作成するようにPostに指示します。この設定は、ポストのパフォーマンスを最大化するために使用できます。この設定では、変更された列の変更前後のcompareが省略されるため、SharePlex以外のプロセスやユーザがターゲットデータを変更できないようにする必要があります。

デフォルト: 0

有効な値の範囲: 0、1、2

有効になるタイミング: Postの再開時

SP_OPX_XML_MARK_NULL

このパラメーターは、ユーザがKafkaのターゲット出力形式としてXMLを選択した場合に、Null値をどのように表示するかを制御します。JSON形式は常にNULLである必要があります。NULL値を`xsi:nil="true"`として表示するには、Kafkaターゲットで SP_OPX_XML_MARK_NULLパラメーターを1に設定します。

デフォルト: 1

有効な値の範囲: 0または1

有効になるタイミング: プロセスの再開時

SP_OPX_CREATE_ORIGIN_PG

このパラメーターは、ターゲットがPostgreSQLの場合にのみ、Open Postに適用されます。これは、PostgreSQLデータベースにレプリケーションのオリジンを作成し、作成したオリジンにPostユーザセッションをバインドするために使用されます。これは、**双方向レプリケーション**（ピアツーピアレプリケーション）で、Postプロセスによってポストされた操作がループバックするのを防ぐために有用です。有効にすると、Captureプロセスは、Postプロセスによってポストされたそのような操作を無視します。デフォルトでは無効になっています。

注: このパラメーターを有効にすると、各Postプロセスでオリジンが作成されます。作成されるオリジンの数は、PostgreSQLデータベースの**max_replication_slots**パラメーターによって決定されます。この結果、Postプロセスの数がこのパラメーターの値を超える場合は、パラメーターの値を増やしてデータベースを再起動してください。

デフォルト: 0(無効)

有効な値の範囲: 0または1

有効になるタイミング: Postの再開時

SP_OPX_LOG_CONFLICT

このパラメーターを使用すると、成功したコンフリクト解決プロシージャに関する情報を**shareplex_conf_log**テーブルに記録できるようになります。これは、SharePlexが提供する準備されたルーチンにのみ適用されます。

- 1に設定すると、**shareplex_conf_log**テーブルへのコンフリクト解決のロギングが有効になります。

  注: 1に設定すると、`shareplex_conf_log`テーブルの列`existing_timestamp`(既存のデータが置換されない場合)は更新されません。

- 2に設定すると、**shareplex_conf_log**テーブルへのコンフリクト解決のロギングが、Postによる追加のメタデータのクエリと共に有効になります。

  準備されたルーチン**LeastRecentRecord**または**MostRecentRecord**を使用すると、Postは既存のレコードのタイムスタンプ列のターゲットデータベースにクエリを実行します。クエリの結果は、`shareplex_conf_log`テーブルの`existing_timestamp`列にログ記録されます。

  注: 2と設定すると、クエリを実行した結果、Postのパフォーマンスに影響を与える可能性があります。

デフォルト: 0(無効)

有効な値の範囲: 0、1、または2

有効になるタイミング: Postの再開時

PostgreSQL Open Target Posterでサポートされる追加パラメーターのリスト

- SP_OPX_BATCH_ENABLE
- SP_OPX_COMMIT_REDUCE_MSGS
- SP_OPX_CONT_ON_ERR
- SP_OPX_DISABLE_OBJECT_NUM
- SP_OPX_MAX_BATCH_MBYTES
- SP_OPX_MSGS_IN_MEMORY

- SP_OPX_NLS_CONVERSION
- SP_OPX_ONELINE_DEBUG
- SP_OPX_OUT_OF_SYNC_SUSPEND
- SP_OPX_READRELEASE_INTERVAL
- SP_OPX_REDUCED_KEY
- SP_OPX_SQL_CACHE_DISABLE
- SP_OPX_THREADS
- SP_OPX_TRUSTED_SOURCE

Oracle Posterパラメーター

これらのパラメーターは、Oracleターゲットにデータを適用する際にSharePlex Posterプロセスで使用されます。

SP_OPO_HINTS_LIMIT

このパラメーターは、ヒントファイルに記載できるヒント（テーブルとインデックスの組み合わせ）の最大数を制御します。ヒントは、定義されたインデックスがあるテーブルに対してPostがフルテーブルスキャンを行っている場合にのみ、使用してください。大量のヒントを使用すると、Postプロセスのパフォーマンスが低下する可能性があります。

デフォルト: 100ヒント

有効な値の範囲: 任意の正の整数

有効になるタイミング: Postの再開時

Oracle Posterプロセスでサポートされる追加パラメーターのリスト

- SP_OPO_DISABLE_OBJECT_NUM
- SP_OPO_COMMIT_REDUCE_MSGS
- SP_OPO_DEPENDENCY_CHECK
- SP_OPO_SUPRESSED_OOSを
- SP_OPO_READRELEASE_INTERVAL

Readパラメーター

これらのパラメーターは、SharePlex Readerプロセスで使用されます。

SP_ORD_FIRST_FIND

このパラメーターは、レプリケートされた行の変更が列条件を満たすかどうかを判断するために、Readプロセスが列条件をチェックする方法を制御します。

- デフォルトの1では、行の変更がある列条件を満たす場合、SharePlexはその行の変更が他の列条件も満たすかどうかをチェックしません。
- 値が0の場合、SharePlexは列条件を満たすすべてのターゲットシステムにデータを送信します。

水平パーティショニングの詳細については、『SharePlex管理ガイド』を参照してください。

デフォルト: 1（オン）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Readの再開時

SP_ORD_HP_IN_SYNC

このパラメーターは、水平分割レプリケーションで、列条件の列の値が変更され、行が条件を満たさなくなった場合に、データが適切にレプリケートされるようにするために使用します。これにより、SharePlexは以下を自動的に修正できます。

- UPDATEにより、ある行が、その行のために作成された列条件とは異なる列条件を満たすようになり、変更が別の場所に送信される。例えば、列条件**region=East**である行に対するUPDATEにより、**region**列の値がWESTに変更されるとします。この操作は失敗します。なぜなら、その行に対する元のINSERTステートメントは元の場所（東部地域）にレプリケートされたため、Postが更新を試みたときには、その行が新しい場所（西部地域）に存在しないからです。
- UPDATEにより、レプリケートされるはずのない行が列条件を満たし、レプリケートされる。例えば、region列がHEADQUARTERSの値（行がレプリケートされない）からWESTの値に更新されたとします。この操作は失敗します。なぜなら、その行の元のINSERTステートメント（本社システムへの）が西部地域のシステムにレプリケートされないため、Postはそこで更新を実行できないからです。
- UPDATEにより、行がどの列条件も満たさなくなる。例えば、**region**列がWESTの値からHEADQUARTERSの値に更新されたとします。元のINSERTステートメントは西部地域のシステムにレプリケートされますが、新しい値が列条件を満たしていない（本社データは共有されていない）ため、新しい値への更新はレプリケートされません。行は非同期になりますが、エラーは発生しません。

このパラメーターが有効な場合、SharePlexはUPDATEによって上記の状態が発生する行を自動的に修正します。SharePlexはUPDATEをDELETEに変換し、必要であればINSERTに変換します。

UPDATEステートメント（通常は変更された列とキーのみを使用）をINSERTステートメントに変換するには、SharePlexはすべての列の値が必要です。`SP_ORD_HP_IN_SYNC`を有効にすると、SharePlexは水平分割レプリケーションを使用しているテーブルに対するUPDATEがあった場合、行のすべての列をPostプロセスに送信して、INSERTを構築できるようになります。

設定をアクティベーションする前に、**ソース**システムでこのパラメーターを設定してください。レプリケーションがアクティブな場合は、パラメーターを設定してから設定を再アクティベーションして、SharePlexがオブジェクトキャッシュを再構築できるようにします。

水平分割レプリケーションを使用しているテーブルの列条件の列が変更されないことが分かっている場合は、このパラメーターを使用すると処理オーバーヘッドが発生するため、0に設定したままにします。

このパラメーターは、SP_OCT_REDUCED_KEYおよびSP_OPO_REDUCED_KEY [任意の値: 1または2]と互換性がありません。このパラメーターは両方のパラメーターの動作をオーバーライドするからです。

デフォルト: 0（オフ）

有効な値の範囲: 0または1（フラグ）

有効になるタイミング: Readの再開時

PostgreSQL Readerプロセスでサポートされる追加パラメーターのリスト

- SP_ORD_BATCH_ENABLE
- SP_ORD_BATCH_MAX
- SP_ORD_BATCH_MATCH_MIN
- SP_ORD_FIRST_FIND
- SP_ORD_HP_IN_SYNC
- SP_ORD_ONELINE_DEBUG
- SP_ORD_RMSG_LIMIT
- SP_ORD_ROLLBACK_TXNS_MAX

Compare/Repairパラメーター

これらのパラメーターは、SharePlexのcompareおよびrepairコマンドのプロパティを制御します。

SP_DEQ_PG_FORCE_PARALLEL

このパラメーターは、PostgreSQLデータベースの実行中のセッションで並列クエリを使用できるようにします。

このパラメーターの値に基づいて、以下の値がPostgreSQLデータベースパラメーター、 `force_parallel_mode (DB version < 16.0)/ debug_parallel_query (DB version >= 16.x)`に割り当てられます。

0 - オフ

1 - オン

2 - 回帰

デフォルト値: 0

有効な値の範囲: 0～2

有効になるタイミング: 次のcompareで即時利用可能

注意:

- SharePlexは、PostgreSQLデータベースオプティマイザがSELECTクエリで並列処理に割り当てるワーカースレッドの数を制御できません。
- このパラメーターは、`pg_hint_plan`拡張機能ではサポートされていません。

SP_DEQ_PG_PARALLEL_MAX

このパラメーターは、PostgreSQLデータベースの実行中のセッションで、クエリに対して許可される並列ワーカースレッドの最大数を管理します。

このパラメーターの値は、PostgreSQLデータベースパラメーターの`max_parallel_workers_per_gather`に割り当てられます。

デフォルト値: 6

有効な値の範囲: 0～1,024

有効になるタイミング: 次のcompareで即時利用可能

注意:

- SharePlexは、PostgreSQLデータベースオプティマイザがSELECTクエリで並列処理に割り当てるワーカースレッドの数を制御できません。
- このパラメーターは、`pg_hint_plan`拡張機能ではサポートされていません。

SP_DEQ_PG_PARALLEL_SETUP_COST

このパラメーターは、実行中のセッションでPostgreSQLデータベースの`parallel_setup_cost`パラメーターを設定するために使用します。

デフォルト値: 10

有効な値の範囲: 0～2147483647

有効になるタイミング: 次のcompareで即時利用可能

注意:

- SharePlexは、PostgreSQLデータベースオプティマイザがSELECTクエリで並列処理に割り当てるワーカースレッドの数を制御できません。
- このパラメーターは、`pg_hint_plan`拡張機能ではサポートされていません。

SP_DEQ_PG_PARALLEL_TUPLE_COST

このパラメーターは、実行中のセッションでPostgreSQLデータベースの`parallel_tuple_cost`パラメーターを設定するために使用します。

デフォルト値: 0.001

有効な値の範囲: 0.0001～3.4E+38

有効になるタイミング: 次のcompareで即時利用可能

注意:

- SharePlexは、PostgreSQLデータベースオプティマイザがSELECTクエリで並列処理に割り当てるワーカースレッドの数を制御できません。
- このパラメーターは、`pg_hint_plan`拡張機能ではサポートされていません。

SP_DEQ_BATCH_ENABLED

1に設定すると、このパラメーターはrepairプロセス中のバッチDML操作を有効にします。0に設定すると、repair中に一度に1つのトランザクションを実行することによって、同期外（OOS）レコードが修正されます。

デフォルト値: 1

有効な値の範囲: 0または1

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_MAX_BATCH_MBYTES

このパラメーターは、repairプロセス中に一度に実行されるバッチDML操作の最大サイズを制御します。repairプロセスは、このパラメーター値で指定されたバッチサイズに達するまで、操作（I/U/D）を追加し続けます。このパラメーターは、クライアントプロセスが実行されるターゲット側で適用されます。

デフォルト: 1メガバイト

有効な値の範囲: 1～任意の正の整数

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_COMMIT_FREQ

このパラメーターは、バッチモードがオフのときのrepairに適用されます。これは、コミットされる必要がある単一トランザクションの数を決定します。

デフォルト値: 1,000

有効な値の範囲: 0～10,000

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_USE_SP_CKSUM

このパラメーターは、compareでSHA256ハッシュを使用するかどうかを決定します。デフォルト値の0に設定すると、SHA256ハッシュを使用します。1に設定すると、SharePlexは内部チェックサム方式を使用して、テーブルデータをそのまま選択します。

注意:

- 大規模なテーブルでこのパラメーターを1に設定すると、パフォーマンスに影響する可能性があります。`hash_values`の方が、そのままのテーブルデータを処理するよりも大幅に時間がかからないからです。
- テーブルにキー列がなく、いずれかの列にNULL値がある場合、パラメーター `SP_DEQ_USE_SP_CKSUM`を1に設定して、compareおよびrepair操作を実行する必要があります。

デフォルト値: 0

有効な値の範囲: 0または1

有効になるタイミング: 次のcompareで即時利用可能

SP_DEQ_PG_DECLARE_FETCH_SIZE

このパラメーターは、ドライバが1回のフェッチで結果セットを返そうとするか、複数のフェッチにまたがって返そうとするかを決定します。

このパラメーターを0に設定すると、ドライバは1回のフェッチを実行します。これにより、全体的な応答時間が向上しますが、その代償として追加メモリが必要になります。

このパラメーターに0以外の数値を設定すると、ドライバは、このパラメーターに設定されたサイズに応じて、1回のフェッチまたは複数のフェッチを実行します。このパラメーターにユーザが設定したサイズが、Selectクエリが返す結果セットの実際のサイズより大きい場合、ドライバは1回のフェッチを実行します。そうでない場合は、複数のフェッチを実行します。複数のフェッチが使用される場合、フェッチのサイズはこのパラメーターの値によって決定されます。大きな結果セットの場合、複数のフェッチで行を取得することで、応答時間を改善し、タイムアウトの可能性を回避できます。

さらに、このパラメーターは、スループットと応答時間のバランスを調整することができます。

デフォルト値: 100

有効な値の範囲: 0～2147483647。サーバプロセスとクライアントプロセスの両方に適用されます。

有効になるタイミング: CompareまたはRepairコマンドの実行直後。

PostgreSQL用のSP_DEQ_READ_BUFFER_SIZE

このパラメーターは、compare/repair操作中に、高度なデータ型の列にのみ適用されます。これらの列がcompare/repairされる際に、フェッチされた高度なデータ型に関連するデータを保持するバッファのサイズを調整します。パラメーターの値は、使用可能なシステムメモリとデータサイズに基づいて調整しなければなりません。

デフォルト: 1 MB

有効な値の範囲: 1～100の任意の整数（単位はMB）

有効になるタイミング: 次のcompareで即時利用可能

Exportパラメーター

これらのパラメーターは、SharePlex Exportプロセスで使用されます。

- SP_XPT_AES_KEY_LENGTH
- SP_XPT_ENABLE_AES
- SP_XPT_ENABLE_COMPRESSION
- SP_XPT_KEEPALIVE
- SP_XPT_PORT_OVERRIDE
- SP_XPT_SO_SNDBUF
- SP_XPT_USE_LOCALHOST

Importパラメーター

これらのパラメーターは、SharePlex Importプロセスで使用されます。

- SP_IMP_ENABLE_AES
- SP_IMP_QUEUE_PAUSE
- SP_IMP_QUEUE_RESUME
- SP_IMP_WCMT_MSGCNT
- SP_IMP_WCMT_TIMEOUT

Queueパラメーター

これらのパラメーターは、SharePlexキューのプロパティを制御します。

- SP_QUE_MAX_QUEUES
- SP_QUE_POST_SHMSIZE
- SP_QUE_Q_SHMSIZE
- SP_QUE_SHMDBUF

- SP_QUE_SYNC
- SP_QUE_USE_SUBQUE_INDEX

sp_copパラメーター

これらのパラメーターは、SharePlex sp_copプログラムによって使用されます。

- SP_COP_IDLETIME
- SP_COP_SCAN_TIMEOUT
- SP_COP_SO_RCVBUF
- SP_COP_TPORT
- SP_COP_UPORT

Systemパラメーター

これらのパラメーターは、システム関連のSharePlexプロパティを制御します。

- SP_SYS_HOST_NAME
- SP_SYS_IN_SYNC
- SP_SYS_JOB_HISTORY_RETENTION
- SP_SYS_JOB_HISTORY_SIZE
- SP_SYS_SUSPEND_ON_ERROR
- SP_SYS_TARGET_COMPATIBILITY
- SP_SYS_VAR_FULL
- SP_SYS_VAR_OK

5

一般的なSharePlexユーティリティ

SharePlexユーティリティは、SharePlex環境の設定、テスト、管理に役立ちます。

内容

構成スクリプト

PostgreSQLの設定スクリプト

cleanup.sql

clean_vardir.sh

create_ignore.sql

hana_cleansp

MSS_cleansp

MySQL_cleansp

ora_cleansp

pg_cleanspユーティリティ

snowflake_cleanspユーティリティ

SharePlexライセンスユーティリティ

OTS

provision

qview

show_scnユーティリティ

show_last_postedユーティリティ

socket_test

sp-bininfo

sp_wallet

sp_hsm

sp_security

トリガスクリプト

PostgreSQL用トリガスクリプト

データベース・セットアップ・ユーティリティ

構成スクリプト

説明

SharePlexには、Oracleソースオブジェクトを指定する設定ファイルの構築を自動化するための以下のスクリプトが用意されています。

- **config.sql**: データベース内のすべてのテーブルと、オプションですべてのシーケンスを設定します。
- **build_config.sql**: スキーマ内の複数の、またはすべてのテーブルを構成します。

サポート対象のソースおよびターゲットデータベース

OracleからOracleへ

config.sqlの使用

config.sqlスクリプトを使用すると、データベースの全スキーマのすべてのテーブル、およびオプションで全シーケンスをリストする設定を構築できます。このスクリプトによって、データベース全体を同一のセカンダリデータベースにレプリケートする高可用性レプリケーション戦略やその他のシナリオを確立する際の時間を節約できます。

config.sqlの使用条件

- ソースとターゲットのテーブル名は同じでなければなりません。
- このスクリプトでは、SYS、SYSTEM、およびSharePlexスキーマ内のオブジェクトは構成されません。これらのスキーマはシステムやインスタンスに固有であるため、複製することはできません。
- このスクリプトは、分割レプリケーションをサポートしていません。**copy config**コマンドを使用して、スクリプトにより構築される設定ファイルをコピーし、**edit config**コマンドを使用して、分割レプリケーションを使用するテーブルのエントリを追加できます。元のファイルではなく新しい設定ファイルをアクティベーションします。
- **edit config**コマンドを使用して、設定が構築された後、必要に応じてその他の変更を加えることができます。

config.sqlを実行するには:

1. SharePlexの変数データディレクトリの**config**サブディレクトリにディレクトリを変更します。**config.sql**スクリプトによって現在の作業ディレクトリに設定が置かれます。SharePlexの設定は**config**サブディレクトリになければなりません。

   cd */vardir*/**config**

2. SYSTEMとしてSQL*Plusにログオンします。
3. SharePlexの製品ディレクトリの**util**ディレクトリからフルパスを使用して**config.sql**を実行します。

   @ */proddir*/**util/config.sql**

プロンプトに従う際は以下の表を参照してください。

プロンプト	入力内容
Target	ターゲットマシンの名前（例: SystemB）。

プロンプト	入力内容
machine	
Source database SID	ソース（プライマリ）OracleインスタンスのORACLE_SID（例: oraA）。**o.**キーワードは含めないでください。ORACLE_SIDでは大文字と小文字を区別します。
Target database SID	ターゲット（デスティネーション）OracleインスタンスのORACLE_SID（例: oraB）。**o.**キーワードは含めないでください。ORACLE_SIDでは大文字と小文字を区別します。
Replicate sequences	シーケンスを複製する場合は**y**を、複製しない場合は**n**を入力します。
SharePlex oracle username	ソースデータベースのSharePlexユーザの名前。このエントリはSharePlexスキーマのレプリケーションを妨げるため、レプリケーションの問題を引き起こすことがあります。有効な名前を指定しないと、スクリプトは失敗します。

注意: SharePlexにより設定に割り当てられる名前は**config.file**です。スクリプトを再度実行して別の設定ファイルを作成すると、最初のファイルが上書きされます。元のファイルを保存するには、2つ目のファイルを作成する前にファイル名を変更してください。

次の手順:

- 大文字と小文字を区別するテーブルや所有者がある場合、**sp_ctrl**の**edit config**コマンドで設定ファイルを開き、テキストエディタを使用して、大文字と小文字を区別するテーブル名と所有者名を"scott". "emp"のように二重引用符で囲みます。このスクリプトは、大文字と小文字を区別するためにOracleが必要とする引用符を追加しません。

  ```
  sp_ctrl> edit config filename
  ```

- 設定が正しい場所にあることを確認するには、**list config**コマンドを実行します。設定の名前が表示されない場合は、間違ったディレクトリに作成されています。ファイルを見付けて、変数データディレクトリの**config**サブディレクトリに移動します。

  ```
  sp_ctrl> list config
  ```

build_config.sqlの使用

build_config.sqlスクリプトを使用すると、スキーマ内の複数（またはすべて）のテーブルを含む設定を構築できます。これは、設定の各コンポーネントを尋ねるプロンプトがステップごとに表示されるインタラクティブなスクリプトです。各オブジェクトとルーティングの情報を個別に入力する代わりに、ワイルドカードを使用して特定のテーブルを一度に選択したり、スキーマ内のすべてのテーブルを選択したりすることができます。

build_config.sqlの使用条件

- ソースとターゲットのテーブル名は同じでなければなりません。
- このスクリプトではシーケンスはサポートされていません。スクリプトが構築する設定をアクティベーションする前に、**sp_ctrl**の**edit config**コマンドを使用して、シーケンスのエントリを追加することができます。
- このスクリプトは、分割レプリケーションをサポートしていません。**copy config**コマンドを使用して、スクリプトにより構築される設定をコピーし、**edit config**コマンドを使用して、分割レプリケーションを使用するテーブルのエントリを追加できます。元の設定ではなく新しい設定をアクティベーションします。

- このスクリプトでは、SYS、SYSTEM、およびSharePlexスキーマ内のオブジェクトは構成されません。これらのスキーマはシステムやインスタンスに固有であるため、複製することはできません。
- **build_config.sql**を異なるスキーマに対して実行し、テキストエディタを使用してそれらの設定を1つの設定に結合することができます。ファイルの最初の非コメント行である**Datasource:o.SID**の1行を除き、すべてを削除します。**config**サブディレクトリからファイルを移動しないでください。
- **edit config**コマンドを使用して、設定が構築された後、必要に応じてその他の変更を加えることができます。

build_config.sqlを実行するには:

1. SharePlexの変数データディレクトリの**config**サブディレクトリにディレクトリを変更します。**build_config.sql**スクリプトによって現在の作業ディレクトリに設定が置かれます。SharePlexの設定は**config**サブディレクトリになければなりません。

   cd */vardir*/**config**

2. SYSTEMとしてSQL*Plusにログオンします。
3. SharePlexの製品ディレクトリの**util**ディレクトリからフルパスを使用して**build_config.sql**を実行します。

   @ */proddir*/**util/build_config.sql**

プロンプトに従う際は以下の表を参照してください。

プロンプト	入力内容
Target machine	ターゲットマシンの名前（例: SystemB）。
Source database SID	ソース（プライマリ）OracleインスタンスのORACLE_SID（例: oraA）。**o.**キーワードは含めないでください。ORACLE_SIDでは大文字と小文字を区別します。
Target database SID	ターゲット（デスティネーション）OracleインスタンスのORACLE_SID（例: oraB）。**o.**キーワードは含めないでください。ORACLE_SIDでは大文字と小文字を区別します。
Owner of the source database tables	ソーステーブルの所有者。
Owner of the target database tables	ターゲットテーブルの所有者。
Table name to include (blank for all)	以下のいずれかを実行します。 • **Enter**を押してデフォルトを受け入れます。ソース所有者に属するすべてのテーブルを選択します。 • ワイルドカード（%）文字と文字列を入力して、特定のテーブルを選択します（例: **%e_salary%**）。 • 個別のテーブル名を入力します。
Name of the output file to create	設定の名前。このスクリプトは、ファイルに**lst**という接尾辞を付けます（例: **Scott_config.lst**）。

次の手順:

- 大文字と小文字を区別するテーブルや所有者がある場合は、**sp_ctrl**の**edit config**コマンドで設定を開き、テキストエディタを使用して、大文字と小文字を区別するテーブル名と所有者名を"scott". "emp"のように二重引用符で囲みます。このスクリプトは、大文字と小文字を区別するためにOracleが必要とする引用符を追加しません。

```
sp_ctrl> edit config filename
```

- 設定が正しい場所にあることを確認するには、**list config**コマンドを実行します。設定の名前が表示されない場合は、間違ったディレクトリに作成されています。ファイルを見付けて、変数データディレクトリの**config**サブディレクトリに移動します。

```
sp_ctrl> list config
```

PostgreSQLの設定スクリプト

説明

SharePlexには、PostgreSQLおよびPostgreSQL Database as a Serviceソースオブジェクトを指定する設定ファイルの構築を自動化するための以下のスクリプトが用意されています。

- **pg_config.sql**: データベース内のすべてのテーブルを設定します。
- **pg_build_config.sql**: スキーマ内の複数の、またはすべてのテーブルを設定します。

サポート対象のソースおよびターゲットデータベース

PostgreSQLからPostgreSQLへ

pg_config.sqlの使用

pg_config.sqlスクリプトを使用すると、データベースの全スキーマのすべてのテーブル、およびオプションで全シーケンスをリストする設定を構築できます。このスクリプトによって、データベース全体を同一のセカンダリデータベースにレプリケートする高可用性レプリケーション戦略やその他のシナリオを確立する際の時間を節約できます。

pg_config.sqlの使用条件

- ソースとターゲットのテーブル名は同じでなければなりません。
- このスクリプトは、分割レプリケーションをサポートしていません。**copy config**コマンドを使用して、スクリプトにより構築される設定ファイルをコピーし、**edit config**コマンドを使用して、分割レプリケーションを使用するテーブルのエントリを追加できます。元のファイルではなく新しい設定ファイルをアクティベーションします。
- **edit config**コマンドを使用して、設定が構築された後、必要に応じてその他の変更を加えることができます。

pg_config.sqlを実行するには:

1. SharePlexの変数データディレクトリの**config**サブディレクトリにディレクトリを変更します。**pg_config.sql**スクリプトによって現在の作業ディレクトリに設定が置かれます。SharePlexの設定は**config**サブディレクトリになければなりません。

   cd /*vardir*/**config**

2. PostgreSQLにログオンします。
3. SharePlexの製品ディレクトリの**util**ディレクトリからフルパスを使用して**pg_config.sql**を実行します。

   \i *proddir*/**util/pg_config.sql**

プロンプトに従う際は以下の表を参照してください。

プロンプト	入力内容
Target machine	ターゲットマシンの名前（例: SystemB）。
Source database name	ソース（プライマリ）PostgreSQLインスタンスのデータベース名（dbnameAなど）。**r.**キーワードは含めないでください。データベース名では大文字と小文字が区別されます。
Target database name	ターゲット（宛先）PostgreSQLインスタンスのデータベース名（dbnameBなど）。**r.**キーワードは含めないでください。データベース名では大文字と小文字が区別されます。
SharePlex PostgreSQL user name（SharePlex PostgreSQLユーザ名）	ソースデータベースのSharePlexユーザの名前。このエントリはSharePlexスキーマのレプリケーションを妨げるため、レプリケーションの問題を引き起こすことがあります。有効な名前を指定しないと、スクリプトは失敗します。

注意: SharePlexにより設定に割り当てられる名前は**pg_config.file**です。スクリプトを再度実行して別の設定ファイルを作成すると、最初のファイルが上書きされます。元のファイルを保存するには、2つ目のファイルを作成する前にファイル名を変更してください。

次の手順:

- 大文字と小文字を区別するテーブルやスキーマがある場合、**sp_ctrl**の**edit config**コマンドで設定ファイルを開き、テキストエディタを使用して、大文字と小文字を区別するテーブル名とスキーマ名を“SCOTT”.“EMP”のように二重引用符で囲みます。このスクリプトは、大文字と小文字を区別するためにPostgreSQLが必要とする引用符を追加しません。

  ```
  sp_ctrl> edit config filename
  ```

- 設定が正しい場所にあることを確認するには、**list config**コマンドを実行します。設定の名前が表示されない場合は、間違ったディレクトリに作成されています。ファイルを見付けて、変数データディレクトリの**config**サブディレクトリに移動します。

  ```
  sp_ctrl> list config
  ```

pg_build_config.sqlの使用

build_config.sqlスクリプトを使用すると、スキーマ内の複数（またはすべて）のテーブルを含む設定を構築できます。これは、設定の各コンポーネントを尋ねるプロンプトがステップごとに表示されるインタラクティブなスクリプトです。各オブジェクトとルーティングの情報を個別に入力する代わりに、ワイルドカードを使用して特定のテーブルを一度に選択したり、スキーマ内のすべてのテーブルを選択したりすることができます。

pg_build_config.sqlの使用条件

- ソースとターゲットのテーブル名は同じでなければなりません。
- このスクリプトは、分割レプリケーションをサポートしていません。**copy config**コマンドを使用して、スクリプトにより構築される設定をコピーし、**edit config**コマンドを使用して、分割レプリケーションを使用するテーブルのエントリを追加できます。元の設定ではなく新しい設定をアクティベーションします。
- **pg_build_config.sql**を異なるスキーマに対して実行し、テキストエディタを使用してそれらの設定を1つの設定に結合することができます。ファイルの最初の非コメント行である**Datasource:r.dbname**の1行を除き、すべてを削除します。**config**サブディレクトリからファイルを移動しないでください。
- **edit config**コマンドを使用して、設定が構築された後、必要に応じてその他の変更を加えることができます。

pg_build_config.sqlを実行するには:

1. SharePlexの変数データディレクトリの**config**サブディレクトリにディレクトリを変更します。**pg_build_config.sql**スクリプトによって現在の作業ディレクトリに設定が置かれます。SharePlexの設定は**config**サブディレクトリになければなりません。

   cd */vardir*/**config**

2. PostgreSQLにログオンします。
3. SharePlexの製品ディレクトリの**util**ディレクトリからフルパスを使用して**pg_build_config.sql**を実行します。

   \i *proddir*/**util/pg_build_config.sql**

プロンプトに従う際は以下の表を参照してください。

プロンプト	入力内容
Target machine	ターゲットマシンの名前（例: SystemB）。
Source database name	ソース（プライマリ）PostgreSQLインスタンスのデータベース名（dbnameAなど）。**r.**キーワードは含めないでください。データベース名では大文字と小文字が区別されます。
Target database name	ターゲット（宛先）PostgreSQLインスタンスのデータベース名（dbnameBなど）。**r.**キーワードは含めないでください。データベース名では大文字と小文字が区別されます。
Source database schema	ソース・データベース・スキーマの名前。
Target database schema	ターゲット・データベース・スキーマの名前。
Table name to include (blank for all)	以下のいずれかを実行します。 • **Enter**を押してデフォルトを受け入れます。ソース所有者に属するすべてのテーブルを選択します。 • ワイルドカード（%）文字と文字列を入力して、特定のテーブルを選択します（例: **%e_salary%**）。 • 個別のテーブル名を入力します。
Name of the output file to create	設定の名前。

次の手順:

- 大文字と小文字を区別するテーブルやスキーマがある場合、**sp_ctrl**の**edit config**コマンドで設定を開き、テキストエディタを使用して、大文字と小文字を区別するテーブル名とスキーマ名を“SCOTT”.“EMP”のように二重引用符で囲みます。このスクリプトは、大文字と小文字を区別するためにPostgreSQLが必要とする引用符を追加しません。

  ```
  sp_ctrl> edit config filename
  ```

- 設定が正しい場所にあることを確認するには、**list config**コマンドを実行します。設定の名前が表示されない場合は、間違ったディレクトリに作成されています。ファイルを見付けて、変数データディレクトリの**config**サブディレクトリに移動します。

  ```
  sp_ctrl> list config
  ```

cleanup.sql

説明

cleanup.sqlスクリプトを使用して、SHAREPLEX_PARTITIONテーブル(再度必要になる可能性のあるパーティションスキームが含まれている)を除く、SharePlex内部テーブルをすべて切り捨てます。

> **注意:** **cleanup.sql**スクリプトは、SharePlex Oracleユーザ、パスワード、デモオブジェクトをSharePlexテーブルから削除しません。

cleanup.sqlスクリプトは、SharePlex変数データディレクトリ内のレプリケーションデータを維持します。他のユーティリティも関連オプションを提供しています。

- SharePlexテーブルを切り捨てずに変数データディレクトリをクリーンアップするには、clean_vardir.sh」の530を参照してください。
- 変数データディレクトリをクリーンアップし、SharePlexテーブルを切り捨てるには、ora_cleansp」の540を参照してください。このユーティリティは、SharePlexを完全に初期状態に復元します。

cleanup.sqlを初めて実行する場合は、実行する前にQuestテクニカルサポートにご連絡ください。SharePlexドキュメントの手順で**clean_vardir.sh**の実行が必要とされる場合を除き、本番環境でこのユーティリティを使用することが適切であることはほとんどありません。このユーティリティは設定を非アクティベーションするので、不適切に使用するとレプリケーションの問題が発生し、データを再同期する必要が生じる可能性があります。通常は別の代替手法があります。

サポート対象データベース

Unix上のOracle

cleanup.sqlを実行するには:

1. SharePlexデータベースユーザとしてOracleにログインします。SharePlexテーブルはそのユーザに属しています。UnixおよびLinuxでは、**sp_cop**の複数のインスタンスを複数の変数データディレクトリで実行している場合は、SharePlexのOracleユーザがそれぞれに存在します。必ず正しいユーザとしてこのスクリプトを実行してください。
2. (UnixおよびLinux)SharePlex変数データディレクトリを指すようにSP_SYS_VARDIR環境変数を設定します。

   kshシェル:

   export SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

   cshシェル:

   setenv SP_SYS_VARDIR=*/full_path_of_variable-data_directory*
3. SharePlex管理者として**cleanup.sql**を実行します。スクリプトは、SharePlex製品ディレクトリの**bin**サブディレクトリにあります。以下の構文を使用します。ここで、*Oracle_version*は、SharePlexがサポートするOracleバージョンの1つです。

   SQL> **@***proddir*/**bin/cleanup.sql**

clean_vardir.sh

説明

clean_vardir.shスクリプトを使用して、変数データディレクトリをクリーンアップして初期状態に戻します。

clean_vardir.shスクリプトは、SharePlex内部テーブルの内容を保持します。

- 変数データディレクトリをクリーンアップせずにSharePlexテーブルを切り捨てるには、cleanup.sqlを参照してください。
- 変数データディレクトリをクリーンアップし、SharePlexテーブルを切り捨てるには、適切な*database*_**cleansp**ユーティリティ(databaseはデータベースのタイプ)を参照してください。このユーティリティは、SharePlexを完全に初期状態に復元します。

重要!clean_vardir.shを初めて実行する場合は、実行する前にQuestテクニカルサポートにご連絡ください。SharePlexドキュメントの手順で**clean_vardir.sh**の実行が必要とされる場合を除き、本番環境でこのユーティリティを使用することが適切であることはほとんどありません。このユーティリティは設定を非アクティベーションするので、不適切に使用するとレプリケーションの問題が発生し、データを再同期する必要が生じる可能性があります。通常は別の代替手法があります。

このユーティリティの機能

clean_vardir.shスクリプトは以下を削除します。

- **rim**サブディレクトリ内のキューファイル。
- **log**サブディレクトリ内のログファイル。イベントログは、**clean_vardir.sh**プロシージャを反映する1つのエントリを保持します。
- **data**サブディレクトリ内の**statusdb**ファイルの内容。
- **dump**および**state**サブディレクトリの内容。

clean_vardir.shスクリプトは、設定ファイル、コンフリクト解決ファイル、ヒントファイル、**paramdb**、**oramsglist**ファイルなど、ユーザが作成したファイルを維持します。

clean_vardir.shスクリプトは、設定を非アクティベーションします。**clean_vardir.sh**を実行した後にレプリケーションを開始するには、設定をアクティベーションする必要があります。

サポート対象データベース

UnixおよびLinux上のOracle

シェル要件

このユーティリティを使用するには、Korn(ksh)シェルがシステムにインストールされていなければなりません。ユーティリティは処理中にこのシェルを呼び出します。

clean_vardir.shを実行するには:

注意: このスクリプトは、UnixおよびLinuxシステムでのみ実行してください。

1. **sp_cop**を停止します。
2. SharePlex変数データディレクトリを指すようにSP_SYS_VARDIR環境変数を設定します。SP_SYS_VARDIRが設定されていない場合、**clean_vardir.sh**は、*proddir*/**data**/**default.yaml**ファイルに記載されているディレクトリに影響を与えます。ここで、*proddir*は、SharePlex製品ディレクトリの**bin**サブディレクトリです。

   kshシェル:

   export SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

   cshシェル:

   setenv SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

3. **clean_vardir.sh**をSharePlex管理者として実行します。スクリプトは、SharePlex製品ディレクトリの**bin**サブディレクトリにあります。以下の構文を使用します。ここで、*Oracle_version*は、SharePlexがサポートするOracleバージョンの1つです。

   proddir/**bin**/**clean_vardir.sh** *Oracle_version*

スクリプトの実行が終了すると、コマンドプロンプトに戻ります。

注意: スクリプトが**save_***SharePlex_version*ディレクトリを削除できないというエラーメッセージを生成した場合、そのディレクトリを手動で削除することができます。

create_ignore.sql

説明

create_ignore.sqlユーティリティスクリプトは、DMLトランザクションがターゲットシステムにレプリケートされないようにするために使用します。このスクリプトは、ソースデータベースにSHAREPLEX_IGNORE_TRANSというパブリックプロシージャを作成します。このプロシージャをトランザクションの開始時に実行すると、Captureプロセスは、実行時点から行われるDML操作をトランザクションがコミットまたはロールバックされるまで無視します。そのため、影響を受ける操作は複製されません。

SHAREPLEX_IGNORE_TRANSプロシージャの影響を受けるのはDML操作のみです。SharePlexにTRUNCATEを含むDDL操作をスキップさせることはありません。DDL操作はOracleによって暗黙的にコミットされるため、プロシージャは無効になります。

サポート対象データベース

Unix上のOracle

create_ignore.sqlの実行

SharePlexの製品ディレクトリの**util**サブディレクトリから**create_ignore.sql**スクリプトを実行します。プロシージャがSharePlexスキーマに作成されるように、SharePlex Oracleユーザとして実行します。

スクリプトは以下のPL/SQLを実行します。

```
CREATE OR REPLACE PROCEDURE SHAREPLEX_IGNORE_TRANS AS

TNUM NUMBER;

BEGIN

INSERT INTO SHAREPLEX_TRANS (TRANS_NUM, QUE_SEQ_NO_1, QUE_SEQ_NO_2,

COMBO, OP_TYPE) VALUES (-999,0,0,'DUMMY',0);

DELETE FROM SHAREPLEX_TRANS WHERE TRANS_NUM=-999 AND COMBO='DUMMY'

AND OP_TYPE=0;

END;

/

GRANT EXECUTE ON SHAREPLEX_IGNORE_TRANS TO PUBLIC;

/
```

SHAREPLEX_IGNORE_TRANSの実行

SHAREPLEX_IGNORE_TRANSは、レプリケートしたくない操作を含むトランザクションの開始時にのみ呼び出してください。このプロシージャをトランザクションの途中で呼び出すと、プロシージャの開始前にレプリケートされた操作は、postキューに無期限に残り、COMMITを待つことになります。SharePlexはCOMMITがなければメッセージをリリースしないからです。Captureは、プロシージャが呼び出された後のトランザクション内のすべての操作を無視するため、COMMITは到着しません。Readプロセスは、それらのレコードの不要なキャッシュ情報を無期限に保持することになります。

1. SQL*PlusにSharePlexユーザとしてログオンします。
2. スキップしたいトランザクションの先頭でSHAREPLEX_IGNORE_TRANSプロシージャを実行します。

   ```
   execute SharePlex_schema.SHAREPLEX_IGNORE_TRANS;
   ```

3. 1つのトランザクションで、レプリケートしたくない変更を行います。Captureはこれらを無視します。
4. COMMITまたはROLLBACKの後、後続のトランザクションのレプリケーションは正常に再開されます。

hana_cleansp

説明

SharePlexがHANAターゲットにレプリケートしているシステム上の現在のレプリケーション状態を削除するには、**hana_cleansp**ユーティリティを使用します。

警告: **hana_cleansp**の結果は元に戻すことはできません。レプリケーションを初期状態に戻さずに停止するには、**abort config**コマンド、または**purge config**コマンドを使用できる場合があります。

使用ガイドライン

- このユーティリティを使用するには、Korn（ksh）シェルがシステムにインストールされていなければなりません。ユーティリティは処理中にこのシェルを呼び出します。
- **hana_cleansp**は、レプリケーション設定内のすべてのHANAターゲットシステムで実行する必要があります。
- レプリケーション環境を完全に削除するには、**ora_cleansp**ユーティリティをレプリケーション設定内のすべてのOracleソースシステム上で実行します。**hana_cleansp**、または**ora_cleansp**がシステム上で実行されたかどうか、またいつ実行されたかを確認するには、そのシステムのSharePlexイベントログを表示します。

このユーティリティの機能

hana_cleanspは、ターゲットシステムで以下の処理を実行します。

- SharePlexスキーマ内のSHAREPLEX_OPEN_TRANS内部テーブルと、SharePlexがインストールしたその他のテーブル（該当する場合）を切り捨てます。
- 変数データディレクトリから、キューファイル、プロセス・ログ・ファイル、**statusdb**の内容（ただしファイルは削除しない）、**dump**および**state**ディレクトリの内容、イベントログの1つのエントリ（**hana_cleansp**のステータスエントリ）を除くすべてのエントリを削除します。

hana_cleanspは以下を保持します。

- SharePlexデータベース、アカウント、パスワード（SharePlexテーブルのデータのみを削除）
- **paramdb**やターゲットの設定など、ユーザが作成したファイル

サポート対象データベース

サポート対象プラットフォーム上のHANA

hana_cleanspを実行するには:

1. システム上のすべてのSharePlexプロセスを停止します。
2. **sp_cop**を停止します。
3. SharePlex製品ディレクトリの**bin**サブディレクトリから、**hana_cleansp**ユーティリティを以下の構文で実行します。

   ```
   C:\users\splex\bin> hana_cleansp [port number ] database_name/user_name
   ```

   ここで:

   - **port** *number*はSharePlexインスタンスのポート番号です(デフォルトの2100以外の場合)。
   - *database_name*はSharePlexデータベースの名前です。
   - *user_name*はSharePlexユーザアカウントの名前です。

4. プロンプトでパスワードを入力します。

   ```
   Enter the password for the SharePlex User :
   ```

5. 「**Y**」と入力し、このSharePlexインスタンスのクリーンアップを実行することを確認します。

   ```
   Are you sure you want to perform the clean for '/splex/vardir/var2200' and port
   2200? [N] : Y
   ```

MSS_cleansp

説明

SQL Serverシステムの現在のレプリケーション状態を削除するには、**mss_cleansp**ユーティリティを使用します。

警告:

- **mss_cleansp**を実行することは、テスト環境やデモ環境では許容できますが、本番環境ではほとんど適切とは言えません。クリーンアップの結果は元に戻すことができません。
- レプリケーション環境を完全に削除するには、ターゲットシステムで**mss_cleansp**ユーティリティを実行します。ソースシステムがOracleの場合は、**ora_cleansp**ユーティリティを実行します。**mss_cleansp**、または**ora_cleansp**がシステム上で実行されたかどうか、またいつ実行されたかを確認するには、そのシステムのSharePlexイベントログを表示します。
- **mss_cleansp**を実行しているSharePlexインスタンスが複数のデータベースにポストするように設定されている場合、**mss_cleansp**は、**それらすべてのデータベース**のレプリケーション設定とサポートデータを削除します。

mss_cleanspは、ターゲットシステムで以下の処理を実行します。

- SHAREPLEX_OPEN_TRANS内部テーブルと、SharePlexがインストールしたその他のテーブル(該当する場合)を切り捨てます。
- 変数データディレクトリから、キューファイル、プロセス・ログ・ファイル、**statusdb**の内容(ただしファイルは削除しない)、**dump**および**state**ディレクトリの内容、イベントログの1つのエントリ(**mss_cleansp**のステータスエントリ)を除くすべてのエントリを削除します。

mss_cleanspは以下を保持します。

- SharePlexデータベース、アカウント、パスワード(SharePlexテーブルのデータのみを削除)
- **paramdb**やターゲットの設定など、ユーザが作成したファイル

サポート対象データベース

Microsoft SQL Server

mss_cleanspを実行するには:

1. システム上のすべてのSharePlexプロセスを停止します。
2. **sp_cop**を停止します。
3. SharePlex製品ディレクトリの**bin**サブディレクトリから、**mss_cleansp**ユーティリティを以下の構文で実行します。

   ```
   C:\users\splex\bin> mss_cleansp [port number] database_name/user_name
   ```

   ここで:

   - **port** *number*はSharePlexインスタンスのポート番号です(デフォルトの2100以外の場合)。
   - *database_name*はSharePlexデータベースの名前です。
   - *user_name*はSharePlexユーザアカウントの名前です。

4. プロンプトでパスワードを入力します。

```
Enter the password for the SharePlex User :
```

5. 「**Y**」と入力し、このSharePlexインスタンスのクリーンアップを実行することを確認します。

```
Are you sure you want to perform the clean for 'C:\splex\vardir\var2200' and
port 2200? [N] :Y
```

MySQL_cleansp

説明

MySQLシステムの現在のレプリケーション状態を削除するには、**mysql_cleansp**ユーティリティを使用します。

警告: **mysql_cleansp**を実行することは、テスト環境やデモ環境では許容できますが、本番環境ではほとんど適切とは言えません。クリーンアップの結果は元に戻すことができません。

使用ガイドライン

- このユーティリティを使用するには、Korn（ksh）シェルがシステムにインストールされていなければなりません。ユーティリティは処理中にこのシェルを呼び出します。
- **mysql_cleansp**は、レプリケーション設定内のすべてのMySQLターゲットシステムで実行する必要があります。
- レプリケーション環境を完全に削除するには、**ora_cleansp**ユーティリティをレプリケーション設定内のすべてのOracleソースシステム上で実行します。**mysql_cleansp**、または**ora_cleansp**がシステム上で実行されたかどうか、またいつ実行されたかを確認するには、そのシステムのSharePlexイベントログを表示します。

このユーティリティの機能

mysql_cleanspは、ターゲットシステムで以下の処理を実行します。

- SHAREPLEX_OPEN_TRANS内部テーブルと、SharePlexがインストールしたその他のテーブル（該当する場合）を切り捨てます。
- 変数データディレクトリから、キューファイル、プロセス・ログ・ファイル、**statusdb**の内容（ただしファイルは削除しない）、**dump**および**state**ディレクトリの内容、イベントログの1つのエントリ（**mysql_cleansp**のステータスエントリ）を除くすべてのエントリを削除します。

mysql_cleanspは以下を保持します。

- SharePlexデータベース、アカウント、パスワード（SharePlexテーブルのデータのみを削除）
- **paramdb**やターゲットの設定など、ユーザが作成したファイル

サポート対象データベース

サポート対象のUNIXプラットフォーム上のMySQL

mysql_cleanspを実行するには:

1. システム上のすべてのSharePlexプロセスを停止します。
2. **sp_cop**を停止します。
3. SharePlex製品ディレクトリの**bin**サブディレクトリから、**mysql_cleansp**ユーティリティを以下の構文で実行します。

   ```
   C:\users\splex\bin> mysql_cleansp [port number ] database_name/user_name
   ```

   ここで:

   - **port** *number*はSharePlexインスタンスのポート番号です(デフォルトの2100以外の場合)。
   - *database_name*はSharePlexデータベースの名前です。
   - *user_name*はSharePlexユーザアカウントの名前です。

4. プロンプトでパスワードを入力します。

   ```
   Enter the password for the SharePlex User :
   ```

5. 「**Y**」と入力し、このSharePlexインスタンスのクリーンアップを実行することを確認します。

   ```
   Are you sure you want to perform the clean for '/splex/vardir/var2200' and port
   2200? [N] : Y
   ```

ora_cleansp

説明

設定の非アクティベーションを含む、システム上の現在のレプリケーション状態を削除するには、**ora_cleansp**ユーティリティを使用します。

警告:

- **ora_cleansp**を実行することは、テスト環境やデモ環境では許容できますが、本番環境ではほとんど適切とは言えません。クリーンアップの結果は元に戻すことができません。
- **ora_cleansp**を実行しているSharePlexインスタンスが、複数のデータベースからキャプチャしたり複数のデータベースにポストしたりするように設定されている場合、**ora_cleansp**は、**それらすべてのデータベース**のレプリケーション設定とサポートデータを削除します。
- **ora_cleansp**ユーティリティを実行しているときに、**ora_setup**の実行時に提供されたTNSエイリアスを入力します。

サポート対象データベース

UnixおよびLinux上のOracle

ora_cleanspの他の代替手法

ora_cleanspを実行する前に、以下の代替手法を検討してください。

- レプリケーションを初期状態に戻さずに設定のアクティビティを停止するには、**sp_ctrl**で**abort config**コマンドまたは**purge config**コマンドを使用できる場合があります。
- 変数データディレクトリのファイルを消去せずにSharePlexテーブルを切り捨てるには、製品ディレクトリの**bin**サブディレクトリ(UnixおよびLinuxシステム)から**cleanup.sql**スクリプトを実行できます。
- UnixおよびLinuxシステムでは、SharePlex製品ディレクトリの**bin**サブディレクトリから**cleanup.sql**スクリプトを実行することで、SharePlexテーブルを切り捨てることなく、変数データディレクトリのファイルをパージできます。

シェル要件

このユーティリティを使用するには、Korn(ksh)シェルがシステムにインストールされていなければなりません。ユーティリティは処理中にこのシェルを呼び出します。

ora_cleanspの仕組み

ora_cleanspは以下のように2段階でレプリケーションを初期状態に戻します。

内部テーブルの切り捨て: 最初に**ora_cleansp**は、安全に切り捨てることができるSharePlex内部テーブルを切り捨てます。SharePlex Oracleユーザ、パスワード、デモオブジェクトは削除しません。

変数データディレクトリのクリーンアップ: 次に**ora_cleansp**は、SharePlex変数データディレクトリから以下のコンポーネントを削除します。

- **rim**サブディレクトリ内のキューファイル。
- **log**サブディレクトリ内のログファイル。イベントログは、**ora_cleansp**プロシージャを反映する1つのエントリを保持します。
- dataサブディレクトリ内の**statusdb**ファイルの内容。
- **dump**および**state**サブディレクトリの内容。
- SP_OPO_SAVE_OOS_TRANSACTIONパラメーターが有効な場合、**oos**サブディレクトリにあるすべてのトランザクション。

ora_cleanspは、設定ファイル、コンフリクト解決ファイル、トランスフォーメーションファイル、ヒントファイル、paramdb、oramsglistファイルなど、ユーザが作成したファイルを維持します。

ora_cleanspは設定を非アクティベーションします。**ora_cleansp**を実行した後にレプリケーションを開始するには、設定をアクティベーションする必要があります。

ora_cleanspを実行するには:

重要! レプリケーションを初期状態に戻すには、すべてのソースシステムとターゲットシステムで**ora_cleansp**を実行する必要があります。**ora_cleansp**がシステム上で実行されたかどうか、またいつ実行されたかを確認するには、そのシステムのイベントログを表示します。

1. SharePlex管理者としてログインします。
2. [**UnixおよびLinuxのみ**] SharePlex変数データディレクトリを指すようにSP_SYS_VARDIR環境変数を設定します。複数の変数データディレクトリを使用して**sp_cop**の複数のインスタンスを実行している場合は、クリーンアップしたい変数データディレクトリに対してSP_SYS_VARDIRを設定します。

   kshシェル:

   $**export SP_SYS_VARDIR=**/*SharePlex_vardir_absolute_path*

   cshシェル:

   $**setenv SP_SYS_VARDIR** /*SharePlex_vardir_absolute_path*
3. SharePlex製品ディレクトリの**bin**サブディレクトリから**sp_ctrl**を実行します。
4. **sp_cop**を停止します。

   sp_ctrl> **shutdown**
5. UnixおよびLinuxシステムで、**ps -ef | grep sp_**コマンドのいずれかを使用して、すべてのSharePlexプロセスが停止していることを確認します。
6. UnixおよびLinuxシステムで、**kill -9**コマンドのいずれかを使用して、すべての孤立したSharePlexプロセスを強制終了します。
7. **[Unix および Linux のみ]** まだ実行していない場合は、**oratab**ファイルを表示して、インスタンスのORACLE_HOMEおよびORACLE_SIDの値を確認します。

   $**cd /etc**
   $ **more oratab**

8. [**UnixおよびLinuxのみ**] ORACLE_HOMEおよびORACLE_SID環境変数を設定します。

   kshシェル:

   $**export ORACLE_HOME=**[*oratabファイルからの値*]
   $**export ORACLE_SID=**[*oratabファイルからの値*]

   cshシェル:

   $ setenv ORACLE_HOME [*oratabファイルからの値*]
   $ setenv ORACLE_SID [*oratabファイルからの値*]

9. オペレーティングシステムのコマンドラインで、ディレクトリをSharePlex製品ディレクトリの**bin**サブディレクトリに変更します。

   注意: これは現在の作業ディレクトリである必要があります。

10. OracleインスタンスでSharePlexユーザアカウントとして**ora_cleansp**を実行します。*注意*: 別のユーザとして実行すると、**ora_cleansp**はSharePlexテーブルを切り捨てません。

    さらに、UnixとLinuxでは、connection.yamlファイルで定義されているデータベース/データソース名（Oracle_SID）を含めます。

    UnixおよびLinux:

    $ **./ora_cleansp** *database_name/SharePlex_username[/SharePlex_password]*

    （Oracle PDBの場合）: $**./ora_cleansp** *tns_alias for pdb/SharePlex_username[/SharePlex_password]*

11. ユーティリティを実行した後、イベントログを表示して結果を確認します。

    注意: ユーティリティが .../**save_SharePlex_version**ディレクトリを削除できないというエラーメッセージを生成した場合、メッセージを無視してこのディレクトリを削除することができます。

pg_cleanspユーティリティ

説明

SharePlexがPostgreSQL Database as a ServiceからPostgreSQLターゲットにデータをレプリケートしているシステム上の現在のレプリケーションの状態を削除するには、**pg_cleanspユーティリティ**を使用します。

> **警告:** **pg_cleanspの結果は元に戻すことはできません。**

使用ガイドライン

- このユーティリティを使用するには、Korn（ksh）シェルがシステムにインストールされていなければなりません。ユーティリティは処理中にこのシェルを呼び出します。
- **pg_cleanspは、レプリケーション設定内のすべてのPostgreSQLソースシステムとターゲットシステムで実行しなければなりません。**
- レプリケーション環境を完全に削除するには、**pg_cleanspユーティリティをレプリケーション設定内のすべての**PostgreSQLソースシステム上で実行します。**pg_cleanspがシステム上で実行されたかどうか、またいつ実行されたかを検証するには、そのシステムのSharePlexイベントログを表示します。**

このユーティリティの機能

pg_cleanspはソースシステム上で以下を行います。

- SharePlexスキーマ内のすべてのSharePlex Client内部テーブルと、該当するその他のSharePlexインストール済みテーブルを切り捨てます。
- 変数データディレクトリから、キューファイル、プロセス・ログ・ファイル、**statusdbファイル、dumpおよびstateディレクトリの内容、イベントログの1つのエントリ（pg_cleanspのステータスエントリ）を除くすべてを削除します。**
- レプリケーションスロット（物理/論理）を削除します。

pg_cleanspは以下を保持します。

- SharePlexデータベース、アカウント、パスワード（SharePlexテーブルのデータのみを削除）
- **paramdbやターゲットの設定など、ユーザが作成したファイル**

PostgreSQLのソースとターゲット上でpg_cleanspユーティリティを実行するには:

1. システム上のすべてのSharePlexプロセスを停止します。
2. **sp_cop**を停止します。
3. SharePlex製品ディレクトリの**bin**サブディレクトリから、**pg_cleansp**ユーティリティを以下の構文で実行します。

   `<installationDirectiory\ProductDirectory>/bin` **pg_cleansp** [**port***number*] *database_name*/*user_name*

   ここで:

   - **port***number*はSharePlexインスタンスのポート番号です(デフォルトの2100以外の場合)。
   - *database_name*はSharePlexデータベースの名前です。
   - *user_name*はSharePlexユーザアカウントの名前です。

4. プロンプトでパスワードを入力します。

   ```
   Enter the password for the SharePlex User :
   ```

5. 「**Y**」と入力し、このSharePlexインスタンスのクリーンアップを実行することを確認します。

   ```
   Are you sure you want to perform the clean for '/splex/vardir/var2' and port
   2200? [N] :Y
   ```

正常にセットアップされると、次のようなメッセージで終了します:

```
Clean port 3800

sh: /splex/atul/1100B51/opt/.app-modules//clean_vardir.sh: /bin/ksh: bad interpreter:
No such file or directory

postgress internal tables are cleaned up.
```

注意: SharePlex 11.1では、アクティベーションの際に作成された物理スロットまたは論理スロットは、クリーンアップ後にデータベースから削除されます。

snowflake_cleanspユーティリティ

説明

SharePlexがSnowflakeターゲットにレプリケートしているシステム上の現在のレプリケーション状態を削除するには、**snowflake_cleansp**ユーティリティを使用します。

> **警告:** **snowflake_cleansp**の結果は元に戻すことはできません。

使用ガイドライン

- このユーティリティを使用するには、Korn（ksh）シェルがシステムにインストールされていなければなりません。ユーティリティは処理中にこのシェルを呼び出します。
- **snowflake_cleansp**は、レプリケーション設定内のすべてのSnowflakeターゲットシステムで実行する必要があります。
- レプリケーション環境を完全に削除するには、レプリケーション設定内のすべてのSnowflakeターゲットシステムで**snowflake_cleansp**ユーティリティを実行します。**snowflake_cleansp**がシステム上で実行されたかどうか、またいつ実行されたかを確認するには、そのシステムのSharePlexイベントログを表示します。

このユーティリティの機能

snowflake_cleanspは、ターゲットシステムで以下の処理を実行します。

- SharePlexスキーマ内のすべてのSharePlexクライアント内部テーブルと、SharePlexがインストールしたその他のテーブル（該当する場合）を切り捨てます。
- 変数データディレクトリから、キューファイル、プロセス・ログ・ファイル、**statusdb**ファイル、**dump**および**state**ディレクトリの内容、イベントログの1つのエントリ（**snowflake_cleansp**のステータスエントリ）を除くすべてのエントリを削除します。

snowflake_cleanspは以下を保持します。

- SharePlexデータベース、アカウント、パスワード（SharePlex内部テーブルのデータのみが削除されます）
- **paramdb**やターゲットの設定など、ユーザが作成したファイル

Snowflakeターゲットでsnowflake_cleanspユーティリティを実行するには:

1. システム上のすべてのSharePlexプロセスを停止します。
2. **sp_cop**を停止します。
3. SharePlex製品ディレクトリの**bin**サブディレクトリから、**snowflake_cleansp**ユーティリティを以下の構文で実行します。

   ```
   <installationDirectiory\ProductDirectory>/bin snowflake_cleansp [portnumber ]
   database_name/ schema name / user_name
   ```

   ここで:

   - **port***number*はSharePlexインスタンスのポート番号です(デフォルトの2100以外の場合)。
   - *database_name*はSharePlexデータベースの名前です。
   - *schema_name*はShareplexスキーマの名前です。
   - *user_name*はSharePlexユーザアカウントの名前です。
4. プロンプトでパスワードを入力します。

   ```
   Enter the password for the SharePlex User :
   ```

   注意: このオプションは、RSAベースのユーザには表示されません。
5. 「**Y**」と入力し、このSharePlexインスタンスのクリーンアップを実行することを確認します。

   ```
   Are you sure you want to perform the clean for '/splex/vardir/var2' and port
   2200? [N] :Y
   ```

正常にセットアップされると、次のようなメッセージで終了します:

```
Clean port 5626

Cleaning /splex/aparopka/sf_datatypes/var subdirectories

DEMO_SCHEMA_20.SHAREPLEX_OPEN_TRANS table truncated.
```

SharePlexライセンスユーティリティ

説明

SharePlexレプリケーション環境内のホストのライセンスキーの表示、追加、削除を行うには、SharePlexライセンスユーティリティを使用します。SharePlexの各インストールには、有効なライセンスキーが必要です。

SharePlexのライセンスキーは、サポート対象のプラットフォームごとに3種類あります。

- トライアルライセンスキー
- 永久ライセンスキー（永続的）
- 期間ライセンスキー

以下は、ライセンスが利用可能なサポート対象プラットフォームのリストです。

- Oracle
- File
- JMS
- Kafka
- SQL Server
- Postgres
- MySQL
- Snowflake
- Event Hubs
- All Platforms

注意： SharePlexの評価版をインストールするには、SharePlexのインストール中または**splex_add_key**ユーティリティの実行中に、プロンプトが表示されたら、「**All Platforms（すべてのプラットフォーム）**」オプションを選択する必要があります。

SharePlexのライセンス情報は、『Quest Software Product Guide（Questソフトウェア製品ガイド）』に記載されています。ご不明な点は、担当のアカウントマネージャまでお問い合わせください。

有効なライセンスキーをお持ちでない場合は、Questテクニカルサポートまたは Questの販売代理店を通じて、上記のサポート対象プラットフォームリストの中から必要なプラットフォームのライセンスキーを入手できます。このドキュメントの適切な手順を使用して、ライセンスリクエストをサポートするために必要な情報を入手してください。

UnixおよびLinuxのライセンスユーティリティ

UnixシステムとLinuxシステムでは、以下を実行するために異なるユーティリティを使用します。

- ライセンスキーの追加
- ライセンスキーの表示

- ライセンスキーの削除

ライセンスキーの追加

splex_add_keyユーティリティを使用して、SharePlexのインストール中にマシンにライセンスキーを追加したり、インストール後にライセンスを異なるタイプのライセンスに変換したりできます。

splex_add_keyユーティリティは以下のように使用できます。

- クラスタのプライマリノードで**splex_add_key**を使用すると、クラスタ内のすべてのセカンダリノードにライセンスをインストールできます。これは、クラスタ内のすべてのセカンダリノードが1つの変数データディレクトリを共有しているからです。
- **splex_add_key**を使用して、あるマシンからクラスタ化されていないマシンのライセンスを追加することはできません。このユーティリティをクラスタ化されていない各レプリケーションシステムで実行して、ライセンス情報が各システムの変数データディレクトリに保存されるようにする必要があります。

splex_add_keyを実行するには:

1. SharePlex管理者としてシステムにログオンします。
2. ライセンスキーをインストールするマシンで**sp_ctrl**を実行します。
3. SharePlexが実行中であればシャットダウンします。

   ```
   sp_ctrl>  shutdown
   ```

4. SharePlex製品ディレクトリの**install**サブディレクトリから**splex_add_key**を実行します。

   $ /*proddir*/**install/splex_add_key**

5. **ライセンスキーを追加/更新するプラットフォームを選択します。**

   ```
   SharePlex License Utility

   1) Oracle

   2) File

   3) JMS

   4) Kafka

   5) SQL Server

   6) Postgres

   7) MySQL

   8) Snowflake

   9) Event Hubs

   10) All Platforms

   q) Quit License Utility

   Enter option:
   ```

   注意: SharePlexの評価版をインストールするには、「**All Platforms（すべてのプラットフォーム）**」オプションを選択する必要があります。

6. 上記のリストから適切な番号を入力してプラットフォームを選択します。
7. Questから受け取ったキーを手動で入力します。キーの入力が終わったら、**Enter**を押します。
8. 「**q**」と入力してユーティリティを終了します。
9. レプリケーションを再開する準備ができたら、SharePlexを起動します。

ライセンスキーの表示

SharePlexライセンスキーに関連する詳細を表示するには、splex_get_keyユーティリティを使用します。SharePlex製品ディレクトリの**install**ディレクトリからこのユーティリティを実行します。

$ /*proddir*/**install/splex_get_key**

以下のような情報が表示されます。

```
$ /splex/proddir/install/splex_get_key

Platform = All

Product Name = SharePlex

Product Version = 11

License Number = 123-456-789

License Key Type = Trial

License Expiry = Midnight of Jan 01, 2050

License Key = lxxjLny9CqMCqdPZKZGRXIjnz7vpbTPQANliJi7PXJ7+Q8=
```

ライセンスキーの削除

特定のプラットフォームにインストールされているSharePlexライセンスキーを削除するには、`splex_remove_key`ユーティリティを使用します。マシンにライセンスキーがインストールされていない場合、「`No license installed`(ライセンスがインストールされていません)」というメッセージが表示されます。

splex_remove_keyを実行するには:

1. SharePlex管理者としてシステムにログオンします。
2. SharePlex製品ディレクトリの**install**サブディレクトリから**splex_remove_key**を実行して、ライセンスキーを削除します。実行すると、以前に追加されたライセンスキーのリストが表示されます。

   $ /*proddir*/**install/splex_remove_key**
3. ライセンスキーを削除する適切なプラットフォームを選択します。

   ```
   SharePlex License Utility

   1) Oracle

   2) Postgres

   q) Quit License Utility

   Enter option: 1
   ```

ライセンスが正常に削除されると、次のようなメッセージが表示されて終了します。

```
The SharePlex for Oracle license has been successfully removed.
```

OTS

説明

OTSユーティリティ(ODBC Test Suite)を使用して、選択したODBCドライバを介してOpen Targetターゲットのデータベースに接続するSharePlexの機能をテストし、さまざまな列タイプにテストデータをポストします。

注意:

- **OTS**ユーティリティは、汎用のODBC接続とデータ型の処理方法を使用し、まだSharePlex認定されていないODBCデータベースの予備テストを行います(データベースがSharePlex を参照してください)。**OTS**ユーティリティは、認定済みのデータベースには使用する必要がありません。認定済みデータベースのリストについては、SharePlexリリースノートを参照してください。
- このユーティリティは、ODBCデータストアがSharePlexターゲットとして適切であることを決定的に証明するものではありません。認証プロセスの一環として、提案された本番環境の複製において実際のデータを使用した包括的なテストを必ず実施する必要があります。

サポート対象データベース

SharePlexでの使用が検討されているOpen Targetデータベース。

要件

1. テストするODBCドライバをインストールします。
2. (Linux)以下の環境変数を設定します。
    - LD_LIBRARY_PATHを以下の両方の場所に設定します。
        - ODBCライブラリが含まれているSharePlex **lib**ディレクトリ。
        - ODBCドライバの場所。
    - データベースのデータソース名(DSN)を設定した場合は、ODBCINIとODBCSYSINIを、**OTS**がインストールされているディレクトリを指すように設定します。DSNが存在せず、接続文字列を使用して接続する場合は、これらの変数を設定する必要はありません(OTSの実行方法を参照してください)。
3. **OTS**は、INSERT、UPDATE、およびDELETE操作をテストするためのテーブルを作成します。このテーブルは、データベースに接続したユーザによって作成され、所有されます(OTSの実行方法 を参照してください)。このユーザは、このテーブルを作成し所有するのに十分な権限を持っている必要があります。

スタンドアロンOTSのインストール方法

OTSは、SharePlexビルドで使用するほか、スタンドアロンバージョンとして使用することもできます。

スタンドアロンバージョンをインストールするには:

Linux: OTSパッケージの**SPOodbc_test_suite-*version*-Beta-*db_version*-rh-40-amd64-m64.tpm**をダウンロードします。インストール先ディレクトリの入力を求められます。

OTSの実行方法

以下の手順でOTSを実行します。

1. **OTS**を実行します。Linux上のSharePlexのインストール内から、またはスタンドアロンOTSパッケージから実行する場合は、**util**サブディレクトリから実行します。
2. 以下のいずれかを入力します。
   - ターゲットデータベースのデータソース名（DSN）。DSNは既に存在している必要があります。その後、名前とパスワードの入力を求められます。
   - 接続文字列。この文字列には、ターゲットデータベースへの接続を確立するための正しいパラメーターがすべて含まれている必要があります。

```
OTS requires connection information for the tested database.
Enter data source name (DSN) or connection string: mydsn
```

仕組み

OTSユーティリティは以下のテストを実行します。

1. ユーティリティの実行者が提供する名前とパスワード、または接続文字列を使用して、Open Targetターゲットに接続します。
2. サポートされている操作とデータ型のリストを、許容されるストレージ値と列がNULL可能かどうかを含めて、ターゲットデータベースにクエリします。
3. 各データ型について、SharePlexがサポートするDMLおよびDDL操作の完全なセットを発行します。
4. 画面上で結果を要約します（OTSを参照してください）。
5. デバッグおよびサポートケースをサポートするために、レポート、ダンプファイル、ログを作成します。

> **注意:** レポートファイルは、OTSがデータベースに接続できる場合にのみ生成されます。接続に失敗した場合、ファイルは空になります。

OTSの結果を解釈する方法

OTS画面の概要には以下の情報が表示されます。

- 使用した接続情報の概要
- テストしたデータ型の概要リスト
- テストの詳細

詳細結果は、例に示すように、文字データ型や整数データ型など、データ型のカテゴリごとに1つずつ表にまとめられます。

図1: 文字データ型の詳細な表の例

```
Test: Character data types
-------------------------------------------------------------------------------
Name     Size Nullable Insert Insert Update Delete Add Drop Trunc Pass/Fail
                       Null                       Col Col
-------------------------------------------------------------------------------
```

```
char      4096 Y        P       P       P       P       P    P    P      P
nchar     4096 Y        P       P       P       P       P    P    P      P
varchar   4096 Y        P       P       P       P       P    P    P      P
sysname     30 Y        P       P       P       P       P    P    P      P
nvarchar  4096 Y        P       P       P       P       P    P    P      P
```

各テーブルの最初の3列（Name、Size、Nullable）は、ODBCドライバへのクエリから取得したものです。データ型がNULLを含むことを許可されている場合、Nullable列に**Y**（はい）が表示されます。そうでない場合は、**N**（いいえ）が表示されます。

クエリ結果の列の後に、実行したテストを表す列が続きます。テストしたデータ型ごとに1行ずつ表示されます。

OTSがテストするSharePlex対応の操作は以下の通りです。

- NULL値を使用したINSERT
- INSERT
- UPDATE
- DELETE
- ALTER TABLEのADD COLUMN
- ALTER TABLEのDROP COLUMN
- TRUNCATE TABLE

結果には、各テストでCHARとVARCHARデータ型に使用されたストレージの最大値と最小値（該当する場合）も表示されます。

- 最大値しかないデータ型（VARCHARおよびCHAR）の場合、OTSは最大サイズより大きいデータ長を挿入して、ODBCデータベースドライバがエラーを返すか、エラーなしで自動的にデータを切り捨てるかを確認します。
- 最小値と最大値の両方を持つデータ型（数値データ型）の場合、両方の値について一連の操作テストを実行します。最小値より小さい数値と最大値より大きい数値がテストされ、ODBCデータベースドライバがオーバーフローエラーを返すか、自動的に丸めて、値を切り捨てるかどうかを確認します。

データ型と操作のテストごとに、**P**（合格）または**F**（不合格）の結果が与えられ、SharePlexが特定のデータ型で特定のDMLまたはDDL操作を実行できたかどうかが示されます。

最後の列である**Pass/Fail**には、特定のデータ型について、テストが*全体として*合格/不合格のいずれであったかが一目で分かるように示されます。ある行のいずれかの操作タイプに**F**が与えられると、**Pass/Fail**列に**F**が表示されます。これは、そのデータ型に対するテストの総合結果が不合格であることを意味します。

結果の利用方法

OTSユーティリティは、まだQuest SharePlex認定されていないOpen Targetデータベース用の*初期スクリーニングツール*です。その目的は、SharePlex自体のインストールやセットアップを必要とせずに、特定のメタデータのセットで、特定のターゲットにレプリケートしたい操作のタイプが成功する*可能性が高い*かどうかを判断することです。例えば、テストの結果が以下の場合、SharePlex *はおそらく*ターゲットにポストできると仮定することができます。

- DMLテストは1つを除いてすべて合格したが、不合格だったデータ型がユーザのデータでは定義されていない。
- DDLはすべて不合格だったが、DDLはレプリケートする必要がない。

OTSユーティリティは最初の一歩に過ぎません。SharePlexの完全なインストールで実際のデータを使用してテストし、ソースOracleアプリケーションが生成する実際の操作を使用し、テスト環境でそれらのレプリケートされた操作を実際のターゲットテーブルにポストすることの代用にはなりません。さらに、OTSでのテストが成功したからといって、データベースがSharePlexで完全にサポートされるとは限りません。OTSの実行方法を参照してください。

データベースがSharePlex
にサポートされるかどうかを判断する方法

SharePlexが特定のOpen Targetデータベースをサポートするかどうかを判断するには、以下を検討します。

1. OTSを通じて、SharePlexはデータベースの指定されたODBCドライバを介してトランザクションの完全なセットを実行できますか?
2. SharePlex製品管理部門は、データベースをサポートする技術的および市場的な実現可能性を承認していますか?
3. SharePlex QAラボは、SharePlexがデータベースをサポートできることをテストで検証しましたか?

質問1と2の回答が**はい**の場合、データベースは**ODBC認定**です。つまり、このデータベースは、質問3に答えるためにQAラボでテストする資格があるということです。

あるデータベースのあらゆるエディション、バージョン、プラットフォームをテストするのは現実的ではありません。市場と技術的な要因に基づいて、製品管理部門はデータベースのどのバリエーションをテストするかを決定します。これらのテストに合格した場合、製品管理部門はデータベースに**SharePlex認定**の指定を与えることを検討できます。

provision

説明

SharePlexの設定のホスト名またはIPアドレスを変更するには、**provision**ユーティリティを使用します。

SharePlexプロセスは、データを適切にルーティングするために、ソースマシンとターゲットマシンのホスト名またはIPアドレスに依存しています。**provision**ユーティリティを使用すると、新しい設定を再アクティベーションすることなく、アクティブなSharePlexインスタンス内でホスト名または IP アドレスを変更できます。

注意: provisionユーティリティは、データベース内の内容を何も変更しません。SharePlex内部オブジェクトにのみ影響します。

サポート対象データベース

サポート対象のすべてのプラットフォーム上のSharePlexでサポートされるすべてのデータベース

provision使用のガイドライン

- AIXマシン上でSharePlexを実行している場合は、**provision**を実行する前にEXTSHMを設定します。

  export EXTSHM=ON

- SharePlex設定内のすべてのマシンで**provision**を実行します。各マシンは、他のすべてのマシンのIPアドレスを参照できます。

provisionの実行

1. **sp_cop**を停止します。**sp_cop**が実行中の場合、**provision**は失敗します。

注意: **provision**は、実行中に**sp_cop**が開始されるのを防止します

2. オペレーティングシステムのコマンドラインで、以下の構文を使用してproductdirのSharePlex utilサブディレクトリから**provision**を実行します。

   provision -f *old_name*[**:***old_ipaddress*] **-t** *new_name*[**:***new_ipaddress*] [**-p** *port*] [**-n**]

   provision -h <new hostid> [**-p** <port>] **-n**
   provision -i [**-p** <port>]

引数	入力
-f *old_hostname* [**:***old_ipaddress*]	• **-f**は必須で、「from」を表します。 • *old_hostname*は古い(現在の)ホスト名。 • *old_IPaddress*は古いIPアドレスです。ネットワークからIPアドレスを取得できない場合に使用します。
-t *new_hostname* [**:***new_ipaddress*]	• **-t**は必須で、「to」を表します。 • *new_hostname*は新しいホスト名。 • *new_IPaddress*は新しいIPアドレスです。ネットワークからIPアドレスを取得できない場合に使用します。
-p *port*	Windowsシステムの場合、provisionを実行するSharePlexインスタンスのポートを指定します。 (**"-p"port** provisionは、Windowsシステムでのみ実行できます。)
-n	実際に変更を加えることなく、**provision**を実行します。**provision**が行う変更に関するレポートを生成します。 **重要!ベストプラクティス**は、まず**-n**を指定して**provision**を実行して、行われる変更内容を確認してから、**-n**なしで実行して実際に変更を加えることです。
-h*new_hostID*	• **-h**は必須で、ホストIDの変更またはホストIDの置き換えを表します。 • new_hostidは新しいIPアドレスです。
-i	**-i**は必須で、ホスト情報を表します。

例:

```
provision -h newid -n

provision -i

provision -f oldname -t newname -n
```

3. イベントログを表示して、行われたすべての変更を表示します。provisionの実行が失敗したり、行われた変更に同意できない場合は、**undo_provision**スクリプトを実行することで変更を取り消すことができます。provisionによる変更を取り消すを参照してください。

provisionによる変更を取り消す

provisionユーティリティは、ホスト名とIPアドレスを以前の状態に復元するために使用できる**undo_provision**スクリプトを作成します。SharePlex製品ディレクトリの**util**ディレクトリから**undo_provision**スクリプトを実行します。このスクリプトには入力引数がありません。

既知の問題

以下のような状況になる可能性がありますが、レプリケーション環境の完全性には影響しません。

- **provision**ユーティリティは、アクティブな設定ファイルを変更しません。これは、**provision**を実行した後、設定ファイルがレプリケーションの現在の状態を表さなくなることを意味します。**compare config**コマンドを実行する必要がある場合、または設定を再アクティベーションすることにした場合は、先に設定ファイルのホスト名またはIPアドレスを更新してください。
- 名前またはアドレスが変更される前にSharePlexをマシンに接続して、ExportまたはImportエラーが発生した場合、エラーステータスが持続し、クリアすることができません。
- 新しいマシンや変更されたマシンがソースマシンである場合、**provision**は新しいルーティング情報を生成しますが、Readプロセスのキャッシュにまだ古いルーティングが残っている可能性があります。**sp_cop**を起動すると、保存されているIPアドレスがマシンのものと一致しないという警告がReadにより生成される場合があります。このエラーは無視してかまいません。
- ソースホストに対して**provision**を実行した後、SHAREPLEX_ACTIDテーブルの "hostname"列が新しいホスト名の詳細で更新されない場合があります。この列が正しく更新されない場合は、SHAREPLEX_ACTIDテーブルを手動で更新して新しいホスト名を指定する必要があります。これは、名前の変更がソースマシンに影響する場合にのみ必要です。

qview

説明

qviewユーティリティを使用すると、キュー名を表示したり、古いキューファイルを削除したりできます。ここで説明する**qview**ツールは、設定を非アクティベーションしません。

重要! **qview**を初めて使用する場合は、必ずQuestテクニカルサポートの支援を得て使用してください。このユーティリティを適切に使用しないと、レプリケーション環境が損傷し、再同期と再アクティベーションが必要になる場合があります。

サポート対象データベース

サポート対象のすべてのプラットフォーム上のSharePlexがサポートするすべてのデータベース

qviewの実行

SharePlex管理者としてシステムにログオンし、オペレーティングシステムのコマンドラインを使用して、SharePlex製品ディレクトリの**bin** サブディレクトリから**qview**を実行します。このユーティリティはインタラクティブなコマンドセッションです。

qviewコマンドの概要

qviewユーティリティは以下のコマンドを提供します。

コマンド	説明
リスト	システム上のすべてのアクティブな設定のすべてのキューを一覧表示します。
trim	古いサブキューファイルをクリーンアップします。
fullrb	完全なロールバックメッセージを作成します。
otrans	Captureキュー内の指定された数のメッセージをスキャンします。

キューの一覧表示

listコマンドを使用して、システム上のすべてのアクティブな設定のすべてのキューを一覧表示します。

説明

qview listコマンドは、各キュー、それに書き込むレプリケーションプロセス、およびそれを読み込むレプリケーションプロセスを一覧表示します。例えば、captureキューの場合、CaptureプロセスとReadプロセスが表示されます。キューは以下のように指定します。

- captureキューは+Cで指定します。
- exportキューは+Xで指定します。
- postキューは+Pで指定します。

出力例:

この例では、captureキュー**o.ora11+C**へのライターは、名前文字列の**sp_ocap**によって示されているように、Captureプロセスです。リーダーは、名前文字列の**sp_ord**で示されているように、Readプロセスです。同じ命名論理が、出力に示された他のキューにも適用されます(exportキューの**expdsg+X**およびpostキューの**expdsg+P**)。

```
The following queues exist:
o.ora11+C
       WRITER +PA+o.ora11+sp_ocap+o.ora11
       READER +PR+o.ora11+sp_ordr+o.ora11
elliot+X
       WRITER +PR+o.ora11+sp_ordr+o.ora11
       READER +PX+elliot+sp_xport+0x0a01014e (11.1.1.78)
elliot+P+o.ora11-o.ora11
subqueues range from 2 to 6
       WRITER +PI+elliot+sp_mport+0x0a01014e (11.1.1.78)
       READER +PP+elliot+sp_opst_mt+o.ora11-o.ora11
```

構文

リスト

古いサブキューの削除

trimコマンドを使用して、ソースシステム上の古いサブキューファイルをクリーンアップします。

説明

SharePlexpostキューは、実際には複数のサブキューから構成されており、それぞれがソースシステム上のユーザセッションにほぼ対応しています。Postプロセスはサブキューを使用して、ターゲットインスタンスのセッションを確立します。ターゲットシステム上の特定の時刻に存在するサブキューの数は、レプリケーション開始以降のソースシステム上のピークアクティビティを反映しています。

SharePlexは、チェックポイント・リカバリ・システムの一環として、レプリケートされたデータをサブキューからディスク上の関連付けられたデータファイルに定期的に書き込みます。各サブキューには1つ以上のデータファイルを関連付けることができ、それぞれのデフォルトサイズは8 MBです。8 MBのサイズ全体が使用されない場合、データがポストされ、読み取り/リリースされた後も、データファイルはシステム上に残ります。その結果、ソースシステムのアクティビティレベルが高いほど、ディスク上のデータファイルが増加します。**qstatus**で表示されるpostキューのサイズ(単位: MB)は、データファイルが占める実際のディスク容量です。

例えば、ソースシステム上に100の同時セッションがあり、ターゲットシステムのpostキューに100のサブキューが作成されたとします。そして、アクティビティレベルが低下したときにデータファイルが完全に使用されなかった(例えば、8 MBのうち半分の4 MBのみが使用された)ため、削除されなかったとします。そのシステムのpostキューは、それぞれ4 MBの100個のデータファイルで構成され、合計400 MBのディスク領域となります。

qviewで**trim**コマンドを使用すると、まだターゲットデータベースにコミットされていないデータを含むサブキューファイルを保持したまま、読み取りリリースされた古いサブキューファイルを定期的に削除することができます。**trim**コマンドは、サブキュー0または1のキューファイルを削除しません。これらはサブキューに最もよく使用されているからです。

このコマンドの実行方法

このコマンドはターゲットシステムでのみ実行します。

qviewを実行してこのコマンドを発行する前に、ImportとPostを停止します。**sp_cop**は実行したままにしておくことができます。

> **注意**: これらのプロセスの一方または両方が停止していない場合、**qview**は次のエラーメッセージを返します: `que_INUSE: Que is already open`(キューが既に開いています)。

一度に削除できるキューは1つだけです。複数のpostキューがある場合、コマンドを適用するpostキューを1つ選択するように求められます。

```
Queue zac+P+o.ora920-o.ora920 (y/n) <n>? n
Queue elliot+P+o.ora920-o.ora920 (y/n) <n>? y
```

> **注意**: キューを選択しないと、**qview**は次のエラーメッセージを返します: `que_NOEXIST: Queue does not exist`(キューが存在しません)。

構文

trim

完全なロールバックの実行

otransコマンドおよび**fullrb**コマンドを使用して、完全なロールバックメッセージを作成します。

説明

otransコマンドを使用して、Captureキュー内の指定された数のメッセージを、読み取りリリースポイントから順にスキャンします。**qview**ユーティリティは、トランザクションID、操作（レコード）の数、DMLタイプの操作（1つしかない場合）、および変更されたオブジェクトID（1つしかない場合）を出力します。

otransから取得したトランザクションIDを使用して、**fullrb**を実行します。**qview**ユーティリティは、Captureキューを開き、帯域外の完全ロールバックメッセージをCaptureキューに書き込み、その後コミットを書き込みます。

このコマンドの実行方法

以下の手順でqviewを実行します。

1. Captureを停止します。
2. **qview**を実行します。
3. **otrans**コマンドを発行します。

   ```
   vqiew> otrans 500000
   ```

   出力は以下のようになります。

   ```
   Full rollback 8(7).752562-3(139) --- 99999 Update operations on object id 466857
   Open transaction 8(23).752700-2(14162) --- 2001 Update operations (1000 backward
   operations)
   on object id 466857
   ```

4. **otrans**で出力されたトランザクションIDを使用して、**fullrb**コマンドを発行します。

   ```
   vqiew> fullrb 8(7).752562-3(139)
   ```

   出力は以下のようになります。

   ```
   Current queue o.ora920+C        user +PA+o.ora920+sp_ocap+o.ora920
   Full rollback record written to capture queue at 378744, id 1102
   odr_magic        0x4f445235
   odr_op           ODR_FULL_ROLLBACK (50)
   odr_trans        8(7).752562-3(139)
   odr_time         01/01/88 00:00:00 (0)
   ```

5. Captureを開始します。

   ```
   sp_ctrl> start capture
   ```

構文

otrans *number*

ここで、*number*はキューでスキャンするメッセージの数です。

fullrb *transaction_ID*

ここで、*transaction_ID*は、**otrans**から返されたトランザクションIDです。

show_scnユーティリティ

show_scnユーティリティを使用して、レプリケーション再開リカバリ手順中に以下のコマンドで指定する正しいOracle SCN値を表示します。

- **activate config** *config_name* **scn=***scn_value*
- **reconcile queue** *queue_name* **for** *datasource-datadest* **scn** *scn_value*

サポート対象データベース

Oracleのソースとターゲット

show_scnの実行

show_scnユーティリティは、ソース、ターゲット、またはその両方に障害が発生した後、レプリケーション再開手順中に実行されます。レプリケーション再開と**show_scn**ユーティリティを使用するには、*レプリケーション開始時*に以下のものを用意しておく必要があります。

- *本番稼働環境ソースインスタンス*の物理的に同一のコピーと*本番稼働環境ターゲットインスタンス*の物理的に同一の別のコピーを提供するディザスタリカバリ(DR)ソリューション。Oracle Data Guardやディスクミラーリング、テープバックアップなどの方法は、この要件をサポートしています。
- SP_OPO_UPDATE_SCNパラメーターは値1に設定しなければなりません。このパラメーターは、処理するトランザクションのSCNの記録を保持するようにSharePlexに指示します。このパラメーターを1に設定すると、Post Enhanced Performance機能も無効になります。

重要: コンテキスト内でshow_scnを使用する方法の詳細については、『SharePlex管理ガイド』の「障害およびリカバリ後のレプリケーション再開」を参照してください。

show_scnを実行するには:

ターゲットシステムのコマンドラインで、SharePlexの製品ディレクトリの**bin**サブディレクトリから**show_scn**ユーティリティを実行します。*データソース*には、**ソース**データベースのORACLE_SIDを使用します。

$ /*productdir*/**bin/show_scn** *datasource*

出力

このユーティリティは、以下のような出力を提供します。

```
$> show_scn o.ora112

On source activate to scn=510012416

For resume replication from ora112

reconcile queue sp01 for o.ora112-o.ora112 scn 4517993781

reconcile queue sp02 for o.ora112-o.ora112 scn 4517994532

reconcile queue sp03 for o.ora112-o.ora112 scn 4517995633
```

show_last_postedユーティリティ

show_last_postedユーティリティを使用して、レプリケーション再開リカバリ手順中に以下のコマンドで指定する正しいPostgreSQL LSN値を表示します。

- **activate config** *config_name* **pglsn=**lsn_value
- **reconcile queue** *queue_name* **for** *datasource-datadest* **pglsn** *lsn_value*

サポート対象データベース

PostgreSQLソースおよびターゲット

show_last_postedの実行

show_last_postedユーティリティは、ソース、ターゲット、またはその両方に障害が発生した後、レプリケーション再開手順中に実行されます。レプリケーション再開と**show_last_posted**ユーティリティを使用するには、*レプリケーション開始時*に以下のものを用意しておく必要があります。

*本番稼働環境ソースインスタンス*の物理的に同一のコピーと*本番稼働環境ターゲットインスタンス*の物理的に同一の別のコピーを提供するディザスタリカバリ(DR)ソリューション。PostgreSQL Data Guardやディスクミラーリング、テープバックアップなどの方法は、この要件をサポートしています。

重要: コンテキスト内で`show_last_posted`を使用する方法の詳細については、『SharePlex管理ガイド』の「障害およびリカバリ後のレプリケーション再開」を参照してください。

show_last_postedを実行するには:

前提条件: `Show_last_posted`ユーティリティを実行する前に、**Post**プロセスを停止する必要があります。

- ターゲットシステムのコマンドラインで、SharePlexの製品ディレクトリの**bin**サブディレクトリから**show_last_posted**ユーティリティを実行します。データソースには、**ソース**データベースの`r.dbname`を使用します。

  ```
  $ /productdir/bin/show_last_posted datasource
  ```

- **sp_ctrl**からコマンドを実行することもできます。

  ```
  sp_ctrl > show last_posted
  ```

出力

このユーティリティは、以下のような出力を提供します。

```
$> show_last_posted r.dbname1

On source activate to pglsn=<LSN in hexadecimal format>

For resume replication from r.dbname1

reconcile queue sp01 for r.dbname1-r.dbname1 pglsn <LSN1 in hexadecimal format>

reconcile queue sp02 for r.dbname1-r.dbname1 pglsn <LSN2 in hexadecimal format>
```

```
reconcile queue sp03 for r.dbname1-r.dbname1 pglsn <LSN3 in hexadecimal format>
```

注意: まずソース データベースで**Activation**コマンドを実行し、次にターゲットで**Reconcile**コマンドを実行します。

socket_test

説明

socket_testユーティリティを使用して、compare/repair機能のネットワークとファイアウォールの問題をデバッグします。**socket_test**ユーティリティが成功する場合、ソースとターゲットの通信を妨げるネットワークやファイアウォールの問題は何もありません。

このセクションでは、ネットワークアドレス変換（NAT）が設定された環境でこのユーティリティを使用する方法について説明します。NATを使用すると、ローカルエリアネットワーク（LAN）で、プライベートな内部トラフィック用に1組のIPアドレスを使用し、パブリックな外部トラフィック用にもう1組のアドレスを使用することが可能になります。NATは、プライベートLANとパブリックインターネットや、セキュアなネットワークとセキュアでないネットワークなど、2つのネットワーク間のブリッジおよびインタープリタとして機能します。

システムがNATを使用しているかどうかを調べるには、ソースマシンから**nslookup**を実行し、ターゲットマシンからも実行します。すべてのテストで同じホスト名を使用します。結果が異なる値を示す場合は、システムがNATが設定された環境で動作している可能性が高くなります。

socket_testユーティリティは、SharePlex製品ディレクトリの**util**サブディレクトリにあります。サーバコンポーネントとクライアントコンポーネントの2つのコンポーネントがあります。

サポート対象データベース

UNIXおよびLinuxでサポートされているすべてのデータベース

socket_testの実行

以下の手順でsocket_testを実行します。

1. ソースマシンで、**socket_test**に**server**オプションを付けて実行し、サーバコンポーネントを実行します。以下の構文を使用します。

   $ *proddir*/**util**/**socket_test server**

   このユーティリティは、以下の例に示すように、プラットフォーム情報、ホスト名、ホストIPアドレス、およびポート番号を表示します。**socket_test**クライアントコンポーネント（ステップ2）が接続されるまで、サーバは待機状態のままです。

2. ターゲットマシンで、**socket_test**に**client**オプションを付けて実行し、クライアントコンポーネントを実行します。以下の構文を使用します。ソースマシンのNAT IPアドレスと、前のステップで**socket_test**サーバテストによって発行されたポート番号を指定します。

   $*proddir*/**util**/**socket_test client** *IP_address*

   テストが成功すると、以下のように通知されます。

   - ユーティリティが「`SUCCESSFULLY read/write messages from server`（サーバからのメッセージの読み取り/書き込みが成功しました）」というメッセージを表示します。
   - さらに、ステップ1のソースマシンの**socket_test server**の出力に、以下のようなテキストが追加されます。

     ```
     Connected to client
     ```

```
WB(512):ReadWriteCnt = 0, readCnt=0, readBytes=0, writeCnt=0, writeBytes=0
WE:ReadWriteCnt = 1, readCnt=0, totalRBytes=0, writeCnt=1, totalWBytes=512
SE:ReadWriteCnt = 1, readCnt=0, totalRBytes=0, writeCnt=1, totalWBytes=512
Server completed successfully, Thu Mar  9 20:00:07 2006
Program exiting..., Thu Mar  9 20:00:07 2006
```

テストが成功しなかった場合、ユーティリティは以下の例のようにエラーメッセージを表示し、**socket_test**サーバは待機状態のままになります。

```
socket_test - version: 1.2
SunOS irvlabu01 5.9 Generic_112233-12 sun4u sparc SUNW,Sun-Fire-880
Program executing as client..., Thu Mar  9 20:02:26 2006
SP_SYS_HOST_NAME was not set.
client host name: irvlabu01
client IP: 10.1.0.36

Client is try to connect to.....
Server Host: 10.1.0.146
Server Port: 57370
Error calling connect in connect_to_server
Error 146: Connection refused
Error 146 calling connect_to_server
CE:ReadWriteCnt = 0, readCnt=0, totalRBytes=0, writeCnt=0, totalWBytes=0
Error in client, Thu Mar  9 20:02:26 2006
```

ソケットテストが失敗した場合、ソースマシンで**Ctrl-C**コマンドを実行し、**socket_test**サーバを終了します。

sp-bininfo

説明

sp-binfoユーティリティを使用して、SharePlexの新しいリリースに、GAリリース間にサポートから受け取ったSharePlexの過去の単発ビルドが含まれていることを確認します。

出力には、現在のバージョンのSharePlexにインストールされている各単発ビルドについて、以下の情報が表示されます。

- Capture（sp_ocap）、Post（sp_opstまたはsp_xpst）などのSharePlexモジュール
- SharePlexのバージョンとビルド番号
- データベースおよびプラットフォーム
- 変更要求（CR）番号（複数可）。これは、ユーザの単発ビルドの要求が開発プロセスに入った時点で、ユーザのケースに割り当てられる一意の内部変更追跡番号です。
- 単発ビルドで更新されたSharePlexライブラリ

図2: sp-binfo出力のサンプル

```
(6) sp_ocap:
       build 171 of SharePlex_Oracle (ONEOFF-CR123456-CR654321-CR789102-oracle110)
8.6.3 for rh-40-amd64 by jdoe
       SharePlex shared lib(s):
         libspwildcard.so.8.6.3.47
         libsporacle.so.8.6.3.47
         libsporalog.so.8.6.3.47
         libspshareplex.so.8.6.3.47
         libspspo.uname.so.8.6.3.47
         libspdb.so.8.6.3.47
         libspodb.so.8.6.3.47
         libspspo.typecheck.so.8.6.3.47
         libspcore.so.8.6.3.47
         libspext.so.8.6.3.47
         libspspo.memory.impl.so.8.6.3.47
         libspspo.memory.stub.so.8.6.3.47
         libspspo.shim.so.8.6.3.47
         libspspymdb.so.8.6.3.47
```

サポート対象データベース

該当なし

サポート対象プラットフォーム

LinuxおよびUnix

sp-bininfoの実行

以下の手順でsp-bininfoを実行します。

1. 現在SharePlexがインストールされている製品ディレクトリの**util**サブディレクトリから**sp-bininfo**ユーティリティを実行します。

   $ **cd** *path_to_SharePlex_proddir*/**util**

   $ ./**sp-bininfo**

2. **sp-bininfo**の出力に表示される各ワンオフのCRと、SharePlexの新しいバージョンに同梱されているリリースノートの「Resolved Issues（解決済みの問題）」セクションのCRをcompareしてください。
3. お客様のワンオフのCRが「Resolved Issues（解決済みの問題）」にない場合、新しいバージョンにはその機能が含まれていないため、アップグレードを続行しないでください。不足している修正プログラムを入手するには、SharePlexのサポートにご連絡ください。

sp_wallet

説明

sp_walletユーティリティを使用して、SharePlexにOracleウォレットのパスワードを提供します。SharePlexはウォレットのパスワードを使用して、TDEプライマリ暗号化キーにアクセスします。SharePlexはTDE プライマリ暗号化キーを使用して、レプリケートする必要があるREDOログ内のTDEで保護されたデータを復号化します。

sp_walletを使用する前に、**dba**グループにOracleウォレットファイルの読み取り権限を付与してください。

サポート対象データベース

UnixおよびLinux上のOracle

sp_walletの実行

sp_walletを実行し、パスワードを手動で指定するには:

1. ソースシステムで、SharePlexの製品ディレクトリからSharePlexを起動します。**sp_wallet**の実行を求めるプロンプトが表示されます。

   ```
   *** To enable TDE replication, run sp_wallet and provide the wallet password ***
   ```

2. **sp_wallet**を実行します。

   ./sp_wallet [-r *port_number*]

   ./sp_wallet -r 9400

   ```
   wallet password: walletpw

   Wallet loaded into SharePlex
   ```

sp_walletをオートオープンモードで実行するには:

オートオープンウォレットを使用している場合、TDEウォレットを自動的に開くようにSharePlexを設定できます。これにより、SharePlexの起動時に**sp_wallet**を手動で実行する必要がなくなります。構文は以下の通りです。

./sp_wallet --auto-open [**-r** *port_number*]

> **重要!** オートオープンウォレット機能を使用するには、セキュリティ上の考慮事項が追加で発生します。詳細については、Oracleのドキュメントを参照してください。また、SharePlexの変数データディレクトリをOracleウォレットやOracleデータファイルと共にバックアップしないでください。

オートオープンモードをキャンセルにするには:

./sp_wallet --no-auto-open [**-r** *port_number*]

TDEプライマリ暗号化キーを変更するには:

SharePlexの設定がアクティブな間にTDEプライマリ暗号化キーを変更する必要がある場合は、SharePlexが変更後もTDEで保護されたデータをレプリケートし続けるように、以下の手順を実行します。

1. ソースデータベースを停止します。
2. REDOログに残っているデータの処理をCaptureが終了したことを確認します。
3. SharePlexをシャットダウンします。
4. TDEプライマリ暗号化キーを変更します。
5. SharePlexを再起動します。
6. **sp_wallet**ユーティリティを実行し、SharePlexに新しいTDEプライマリ暗号化キーを提供します。

   ./sp_wallet [**-r** *port_number*]

sp_hsm

説明

sp_hsmユーティリティを使用して、外部ストレージ（HSM）に格納されているOracleウォレットのパスワードをSharePlexに提供します。SharePlexはウォレットのパスワードを使用して、TDEプライマリ暗号化キーにアクセスします。SharePlexはTDE プライマリ暗号化キーを使用して、レプリケートする必要があるREDOログ内のTDEで保護されたデータを復号化します。

sp_hsmを使用する前に、**dba**グループにOracleウォレットファイルの読み取り権限を付与してください。

サポート対象データベース

Oracle

sp_hsmユーティリティの実行

sp_hsmユーティリティを実行するには:

1. コマンドプロンプトを開きます。
2. RAC構成[ORCL1]: ORCL1の場合は、**Oracle SID**または**TNSエイリアス**を入力します
3. HSMライブラリが正しくセットアップされていることを確認します。

   ```
   HSM opm_is_hsm_library_setup entered for sid o.ORCL1

   HSM opm_is_hsm_library_setup returned true
   ```

4. Oracleデータベースが構成されているHSMの**スロットID**を入力します。

   ```
   HSM Slot ID [0]: 1
   ```

5. HSMパスワードを入力します。

HSMパスワードがShareplexに正常に追加されると、ユーティリティは次のメッセージを表示します。

```
The HSM password has been successfully added to Shareplex.

Shareplex will automatically connect to HSM upon startup.

The sp_hsm utility will not need to be run again unless you want to change the
HSM password.

To change the HSM password, just run sp_hsm and enter the new password.

The old HSM password will replace the existing password.
```

sp_security

説明

SharePlexネットワーク通信のSSL/TLS設定を有効、無効、または表示するには、**sp_security**ユーティリティを使用します。

SSL/TLSを有効にする

重要! SSL/TLSは、共通のネットワークパスワードを使用して有効にするか、SharePlexのすべてのインストールで無効にする必要があります。

SSL/TLSを有効にするには:

`sp_security --setup`を実行し、**SSL/TLS**オプションを選択して、ネットワークパスワードを入力します。

```
% sp_security --setup

Security Setup Wizard
---------------------
This wizard will walk you through setting up the SharePlex network security.

Setup configuration for '/home/shareplex/var110/' and Port 2100 [N]: Y

Choose your network security model.  Please note the following:
         * Cop must be down when the security model is changed, or when the
network password is changed
         * The same model must be used among all SharePlex nodes replicating to
each other
         * For security model [1], the same network password must be set on all
SharePlex nodes replicating to each other

  [1] Use basic SSL/TLS connections
  [2] Use non-SSL/TLS connections (default prior to SharePlex 9.1.3)

Security model: 1

Please enter a network password that will be used for authentication
among the SharePlex nodes.  All SharePlex nodes that replicate data to each
other must have the same network password.

Network password:
Please re-enter the network password

Network password:

Security settings:
```

```
Configuration for '/home/shareplex/var110/' and Port 2100:

  Security model            : SSL/TLS
  Network password          : stored for unattended startup
  SSL key file password     : stored for unattended startup
  SSL key file              : key.pem
  SSL cert file             : cert.pem

Setup complete!
```

SSL/TLSを無効にする

重要! SSL/TLSは、共通のネットワークパスワードを使用して有効にするか、SharePlexのすべてのインストールで無効にする必要があります。

SSL/TLSを無効にするには:

「sp_security --setup」を実行して、非SSL/TLS接続を選択します。

```
% sp_security --setup

Security Setup Wizard
---------------------
This wizard will walk you through setting up the SharePlex network security.

Setup configuration for '/home/shareplex/var110/' and Port 2100 [N]: Y

Choose your network security model.  Please note the following:
         * Cop must be down when the security model is changed, or when the
network password is changed
         * The same model must be used among all SharePlex nodes replicating to
each other
         * For security model [1], the same network password must be set on all
SharePlex nodes replicating to each other

  [1] Use basic SSL/TLS connections
  [2] Use non-SSL/TLS connections (default prior to SharePlex 9.1.3)

Security model: 2

Security settings:

Configuration for '/home/shareplex/var110/' and Port 2100:

  Security model            : Un-encrypted

Setup complete!
```

現在のSSL/TLS設定を表示する

現在のSSL/TLS設定を表示するには:

「sp_security --show」を実行します。

```
% sp_security --show

Security settings:

Configuration for '/home/shareplex/var110/' and Port 210:

  Security model            : Un-encrypted
```

トリガスクリプト

説明

SharePlexには、SQL*Plusで実行できる3つの対話型スクリプトが用意されており、レプリケーションを妨げないようにトリガを管理することができます。スクリプトを実行する前に、以下の点を確認してください。

- トリガスクリプトを実行するには、SYSオブジェクトとDBAオブジェクトの両方へのアクセス権が必要です。SharePlexデータベースアカウントがインストール時に適切に設定されていれば、そのアカウントにはこのアクセス権が付与されています。
- Oracleのパッチ(トリガを作成または変更することがあります)を適用した後、トリガが無効になっていない場合は、ターゲットシステムで**sp_add_trigger.sql**を再実行してください。
- このスクリプトは、SharePlex製品ディレクトリの**util**サブディレクトリに保存されています。
- SharePlexトリガスクリプトは行レベルのトリガのみをサポートしています。ステートメントレベルのトリガは、SharePlexユーザに対して個別に無効/有効にする必要があります。
- レプリケートされないテーブルのデータを変更し、レプリケーション設定内のテーブルに影響を与えないトリガは、特別な処理をしなくてもターゲットシステムで起動できます。ただし、このスクリプトは、OracleまたはPostgreSQLインスタンス内のすべてのオブジェクトに対して、それらがレプリケーションの一部であるかどうかに関係なく動作します。
- SharePlexには、トリガを管理するための対話型スクリプトが他にも用意されています。
- スクリプトはターゲットオブジェクトに対して実行する必要があります。ピアツーピアレプリケーションを使用している場合を除き、ソースオブジェクトでは実行しないでください。
- このスクリプトは、ソースオブジェクトとターゲットオブジェクトの両方が同じマシン上にある場合は使用しないでください。このような設定の場合、ユーザはターゲットオブジェクトのトリガを無効にすることができます。

サポート対象データベース

Oracleおよびサポート対象プラットフォーム

sp_add_trigger.sql

ピアツーピア設定や高可用性設定など、ターゲットオブジェクトのトリガを無効にできない場合は、**sp_add_trigger.sql**を使用します。このスクリプトは、Postプロセスに関連付けられたSharePlex Oracleユーザを無視し、他のすべてのユーザに対して起動するようにトリガを変更します。これは、データベース内の各トリガに以下のWHEN句を挿入します。

```
when user != 'SharePlex_username' begin
```

スクリプトは、SharePlexユーザ名の入力を求め、トリガを変更します。SYS、SYSTEM、SCOTTに属するトリガは変更されません。

構文

@absolute_pathname/**sp_add_trigger.sql**

sp_change_trigger.sql

sp_change_trigger.sqlを使用して、**sp_add_trigger.sql**で作成したWHEN句のSharePlex ユーザ名を置き換えます。現在のSharePlexユーザ名と新しいユーザ名の入力を求められます。このスクリプトを実行する前に、OracleのSharePlexユーザを変更してください。

構文

@absolute_pathname/**sp_change_trigger.sql**

sp_remove_trigger.sql

sp_remove_trigger.sqlを使用して、**sp_add_trigger.sql**で作成したWHEN句を削除します。SharePlexユーザ名の入力が求められ、その後、すべてのトリガからWHEN句が自動的に削除されます。これらのテーブルにレプリケートする必要がなくなった場合は、このスクリプトを使用します。トリガを無効にせずにSharePlexを使用してこれらのテーブルにレプリケートし続けると、SharePlexがデータをポストしたときにトリガが起動し、SharePlexは同期外エラーを生成します。

構文

@absolute_pathname/**sp_remove_trigger.sql**

PostgreSQL用トリガスクリプト

説明

SharePlexには、SQL*Plusで実行できる3つの対話型スクリプトが用意されており、レプリケーションを妨げないようにトリガを管理することができます。

スクリプトを実行する前に、以下の点を確認してください。

- トリガスクリプトを実行するには、SYSオブジェクトとDBAオブジェクトの両方へのアクセス権が必要です。SharePlexデータベースアカウントがインストール時に適切に設定されていれば、そのアカウントにはこのアクセス権が付与されています。
- PostgreSQLのパッチ(トリガを作成または変更することがあります)を適用した後、トリガが無効になっていない場合は、ターゲットシステムで**sp_pg_add_trigger.sql**を再実行してください。
- このスクリプトは、SharePlex製品ディレクトリの**util**サブディレクトリに保存されています。
- SharePlexトリガスクリプトは行レベルのトリガのみをサポートしています。ステートメントレベルのトリガは、SharePlexユーザに対して個別に無効/有効にする必要があります。
- レプリケートされないテーブルのデータを変更し、レプリケーション設定内のテーブルに影響を与えないトリガは、特別な処理をしなくてもターゲットシステムで起動できます。ただし、このスクリプトは、PostgreSQLインスタンス内のすべてのオブジェクトに対して、それらがレプリケーションの一部であるかどうかに関係なく動作します。
- SharePlexには、トリガを管理するための対話型スクリプトが他にも用意されています。
- スクリプトはターゲットオブジェクトに対して実行する必要があります。ピアツーピアレプリケーションを使用している場合を除き、ソースオブジェクトでは実行しないでください。
- このスクリプトは、ソースオブジェクトとターゲットオブジェクトの両方が同じマシン上にある場合は使用しないでください。このような設定の場合、ユーザはターゲットオブジェクトのトリガを無効にすることができます。

制限: PostgreSQLデータベースはユーザとスキーマの間に1対多の関係、つまり、1人のユーザと複数のスキーマの関係を持っているため、これらのスクリプトはユーザを入力として、そのユーザのデータベース以下のすべてのトリガを有効または無効にします。

サポート対象データベース

サポート対象のプラットフォーム上のPostgreSQL

sp_pg_add_trigger.sql

ピアツーピア設定や高可用性設定など、ターゲットオブジェクトのトリガを無効にできない場合は、**sp_pg_add_trigger.sql**を使用します。このスクリプトは、Postプロセスに関連付けられたSharePlex PostgreSQLユーザを無視し、他のすべてのユーザに対して起動するようにトリガを変更します。これは、データベース内の各トリガに以下のWHEN句を挿入します。

```
when user != 'SharePlex_username' EXECUTE { FUNCTION | PROCEDURE } function_name (
arguments )
```

スクリプトは、SharePlexユーザ名の入力を求め、トリガを変更します。SYS、SYSTEM、SCOTTに属するトリガは変更されません。

構文

\i *absolute_pathname*/**sp_pg_add_trigger.sql**

sp_pg_change_trigger.sql

sp_pg_change_trigger.sqlを使用して、**sp_pg_create_trigger.sql**で作成されたWHEN句のSharePlexユーザ名を置き換えます。現在のSharePlexユーザ名と新しいユーザ名の入力を求められます。このスクリプトを実行する前に、PostgreSQLでSharePlexユーザを変更してください。

構文

\i *absolute_pathname*/**sp_pg_change_trigger.sql**

sp_pg_remove_trigger.sql

sp_pg_remove_trigger.sqlを使用して、**sp_pg_create_trigger.sql**で作成したWHEN句を削除します。SharePlexユーザ名の入力が求められ、その後、すべてのトリガからWHEN句が自動的に削除されます。これらのテーブルにレプリケートする必要がなくなった場合は、このスクリプトを使用します。トリガを無効にせずにSharePlexを使用してこれらのテーブルにレプリケートし続けると、SharePlexがデータをポストしたときにトリガが起動し、SharePlexは同期外エラーを生成します。

構文

\i *absolute_pathname*/**sp_pg_remove_trigger.sql**

データベース・セットアップ・ユーティリティ

データベース・セットアップ・ユーティリティは、SharePlex接続を許可し、レプリケーションをサポートする必要なデータベースコンポーネントを確立するために、ソースまたはターゲットデータベースを自動的に設定します。

各種データベースユーティリティのセットアップ方法の詳細については、『SharePlex インストールおよびセットアップガイド』の「データベース・セットアップ・ユーティリティ」のセクションを参照してください。

6

SharePlex for Oracleクラウドインフラストラクチャ

内容

Oracle Cloud-ExaCSとDBCSのデータベースセットアップ
要件

Oracle Cloud-ExaCSとDBCSのデータベースセットアップ

Oracle用データベース・セットアップ・ユーティリティ(**Ora_setup**)を使用して、SharePlexをOracleユーザとして確立し、必要なSharePlexデータベースオブジェクトを作成します。このセットアップユーティリティでは、以下が作成されます。

- SharePlexアカウント
- SharePlexが使用し、SharePlexアカウントが所有するテーブルとその他のオブジェクト
- SharePlexユーザのデフォルト接続

セットアップユーティリティを実行する前に、このトピックの内容をすべて確認することをお勧めします。

サポート対象データベース

サポート対象プラットフォーム上のOracleソースまたはターゲット

Oracleのセットアップを実行するタイミング

SharePlexのインストール時にこのユーティリティを実行するかどうかは、データベースがソース、中間、ターゲットのいずれであるか、またデータをどのように同期させるかによって異なります。初期同期手順については『SharePlex管理ガイド』を参照してください。

システムのタイプ	Oracleのセットアップを実行するタイミング
ソースシステム	SharePlexのインストール中
中間システム	中間システムはカスケード構成で使用されます。ここでは、SharePlexがデータを1つ(または複数の)リモートシステムにレプリケートし、そのデータを中間システムから最終ターゲットに送信します。中間システム上のデータベースにデータをポストするようにSharePlexを設定し、ホットバックアップを使用してそのシステムとターゲットでデータを確立する場合は、中間システムまたはターゲットシステムでデータベース・セットアップ・ユーティリティを実

システムのタイプ	Oracleのセットアップを実行するタイミング
	行しないでください。このユーティリティは、初期同期手順を実施するときに実行します。
ターゲットシステム	レプリケーションをアクティベーションする準備ができたときに、ソースとターゲットのデータを同期するために使用する方法によって異なります。 ◦ 移動可能なテーブルスペースまたはコールドコピー（export/import、テープからの格納/リストア、FTPなど）を使用する場合は、データベース・セットアップ・ユーティリティをSharePlexのインストール中に実行してください。 ◦ ホットバックアップを使用してターゲットデータを確立する場合は、データベース・セットアップ・ユーティリティを実行しないでください。このユーティリティは、初期同期手順を実施するときに実行します。 **注意:** バックアップとリカバリの前にデータベース・セットアップ・ユーティリティを実行すると、セットアップが上書きされるため、バックアップとリカバリの後に再度実行する必要があります。

サポート対象のOracle接続

セットアップユーティリティでは、SharePlexユーザがデータベースへの接続時に使用する以下のいずれかの接続を設定できます。

データベースのタイプ	connection
ASMを使用したPDB	PDBにはTNSエイリアス、ASMインスタンスにはTNSまたはBequeathのいずれかを使用します。

必要な実行の権限

データベース・セットアップ・ユーティリティを実行するユーザには以下の権限が必要です。

Amazon RDSのソースまたはターゲット

セットアップユーティリティを実行するユーザは、Oracle RDSインスタンスの作成時に作成されたプライマリユーザである必要があります。セットアップ中にこのユーザに対して入力が求められます。

非マルチテナント（標準）データベース:

セットアップユーティリティを実行するユーザにはDBA権限が必要です。

マルチテナントデータベース:

セットアップユーティリティを実行するユーザはSYSDBAの権限を持っている必要がありますが（推奨）、少なくとも**sys.users$**および**sys.enc$**の権限を持つDBAユーザでなければなりません。

SharePlexユーザには少なくとも以下の権限が必要です。

```
create user c##sp_admin identified by sp_admin;
```

```
grant dba to c##sp_admin container=ALL;

grant select on sys.user$ to c##sp_admin with grant option container=ALL;
```

SharePlexに付与される権限

データベース・セットアップ・ユーティリティは、SharePlexデータベースユーザに以下の権限を付与します。

付与される権限	説明
DBAロール	データベース・セットアップ・ユーティリティは、DBAロールと無制限のリソース権限、テーブルスペース権限、およびREDOログの読み取り権限を付与します。
デフォルトのOracleプロファイル	デフォルトでこのプロファイルには、Oracleによって最初に割り当てられた無制限のリソース権限があります。
付与	**SharePlexには以下の権限が付与されます。** • O7_DICTIONARY_ACCESSIBILITYがFALSEに設定されている場合、(DBAロールの外で)データ辞書にアクセスするには: **grant select any dictionary to***SharePlexUser***;** • DDLをレプリケートするには: **grant select any table to***SharePlexUser***with admin option;** **grant create any view to***SharePlexUser***with admin option;**

要件

- Oracleのセットアップを実行するシステムにデータベースクライアントをインストールします。データベースで使用する適切なクライアントのバージョンについては、Oracleのドキュメントを参照してください。
- SharePlexのレプリケーション設定のすべてのソースおよびターゲットOracleインスタンスで、データベース・セットアップ・ユーティリティを実行します。
- 統合レプリケーショントポロジ、または複数の変数データディレクトリを持つ他のトポロジでは、各変数データディレクトリに対してデータベース・セットアップ・ユーティリティを実行します。
- SharePlexは、ローカルBEQUEATH接続またはTNSエイリアスを使用したリモート接続をサポートしています。使用する接続に必要な接続値をOracleのセットアップで指定する準備をしてください。TNSを使用する場合は、セットアップを実行する前に**tnsnames.ora**ファイルを設定する必要があります。
- Oracleデータベースがマルチテナント・コンテナ・データベースの場合、レプリケーションシナリオに関連する各プラガブルデータベースについてデータベース・セットアップ・ユーティリティを実行します。SharePlexユーザおよびスキーマオブジェクトが各PDBに存在していなければなりません。
- アクティブな設定があるときにデータベース・セットアップ・ユーティリティを実行すると、SharePlexの内部テーブルをインストールまたは更新するためにセットアップによって実行されるDDLがターゲットにレプリケートされます。この問題を回避するには、ユーティリティを実行する前に**SP_OCT_REPLICATE_ALL_DDL**パラメーターを**0**に設定し、セットアップ完了後にパラメーターを以前の設定に戻します。このパラメーターは即座に有効になります。

SharePlexスキーマのストレージ要件

Oracleのデータベース・セットアップ・ユーティリティは、SharePlexで使用するためにいくつかのデータベースオブジェクトをインストールします。これらのオブジェクトのストレージ要件は、Oracleのセットアップを実行する前に満たしておく必要があります。以下の表を参照してください。

ストレージ	説明
SharePlexオブジェクトテーブルスペース	データベース・セットアップ・ユーティリティは、選択したテーブルスペースにいくつかのテーブルをインストールします。SHAREPLEX_LOBMAPテーブル以外は、テーブルスペースのデフォルトのストレージ設定を使用します。 SHAREPLEX_LOBMAPテーブルには、行の外に格納されたLOBのエントリが含まれています。これは、1 MBのINITIALエクステント、1 MBのNEXTエクステント、および10のPCTINCREASEで作成されます。MAXEXTENTSは120で、テーブルの最大許容値サイズは120 MBです。 **推奨アクション**: プライマリキーと一意キーのサプリメンタルロギングを有効にしている場合、SP_OCT_ENABLE_LOBMAPパラメーターを0に設定すると、SHAREPLEX_LOBMAPテーブルには何も保存されません。この場合、サイズが大きくなることを考慮する必要はありません。Readプロセスのパフォーマンスを最大化するために、プライマリキーと一意キーのサプリメンタルロギングを有効にすることを推奨します。 **代替アクション**: 通常、デフォルトのストレージはSHAREPLEX_LOBMAPに十分であり、400万以上のLOBエントリが許可されます。複製するOracleテーブルに、頻繁に挿入または更新される多数のLOB列がある場合は、SharePlexのテーブルスペースのサイズを適宜大きくすることを検討してください。このテーブルが、SharePlexの他のテーブルとテーブルスペースを共有していることを考慮してください。 データベースがコスト・ベース・オプティマイザ(CBO)を使用しており、SharePlexが処理するテーブルに多数のLOBが含まれる場合は、SHAREPLEX_LOBMAPテーブルを分析スケジュールに組み込みます。 **注意:** SharePlexを新規にインストールしても、以前のインストールからストレージパラメーターは変更されません。
SharePlex一時テーブルスペース	データベース・セットアップ・ユーティリティでは、**compare**コマンドで実行されるソートを含む、ソートやその他の操作に使用するSharePlex用の一時テーブルスペースの入力を求めるプロンプトが表示されます。デフォルトの一時テーブルスペースは、SharePlexのオブジェクトがインストールされている一時テーブルスペースです。**compare**コマンドを使用して大きなテーブル、特にプライマリキーや一意キーを持たないテーブルを比較する場合は、SharePlex専用の一時テーブルスペースを指定します。
SharePlexインデックステーブルスペース	データベース・セットアップ・ユーティリティは、SharePlexテーブルのインデックスを格納するテーブルスペースを要求します。デフォルトのインデックステーブルスペースは、SharePlexオブジェクトがインストールされているものです。I/Oの競合を最小化するには、テーブルがインストールされているテーブルスペースとは別のインデックステーブルスペースを指定します。 **注意:** 以前のバージョンのSharePlexのインデックスがSharePlexのオブジェクトテーブルスペースにインストールされている場合は、それらを別のテーブルスペースに移動し、セットアップユーティリティを実行するときにそのテーブルスペースを指定することができます。

Oracle cloud-ExaCSおよびDBCSのデータベースセットアップの実行

Oracle cloud-ExaCSとDBCSのデータベースセットアップを行うには、以下の手順を実行します。

重要! この手順を実行する前に、Oracleインスタンスをオープンしておく必要があります。

1. (UnixとLinuxのみ)複数の変数データディレクトリを使用している場合は、データベースセットアップを実行するSharePlexインスタンスの変数データディレクトリを指す環境変数をエクスポートします。

   kshシェル:

   export SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

   cshシェル:

   setenv SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

2. **sp_cop**を含む、すべての実行中のSharePlexプロセスをシャットダウンします。
3. オペレーティングシステムのコマンドプロンプトから、SharePlex**bin**サブディレクトリのフルパスを使用してデータベース・セットアップ・プログラムを実行します。
4. SharePlexの設定で、システムがソースシステムであるか、ターゲットシステムであるか、あるいはソースシステムとターゲットシステムの両方であるかを指定します。

   注意: このプロンプトは、このデータベースセットアップを最初に実行したときにのみ表示されます。

5. 接続タイプには**Oracle**を選択します。
6. 希望する接続のタイプに合わせてSharePlexを適切に設定するためのプロンプトと応答については、以下の表を参照してください。

表3: セットアップのプロンプトと対応

プロンプト	対応
`Will SharePlex install be using a BEQUEATH connection? (Entering 'n' implies a SQL*net connection):`	ローカルのBEQUEATH接続を使用する場合は「**Y**」を、TNSエイリアス接続を使用する場合は「**N**」を押します。 **注意**: 以下の場合、**TNSエイリアス**を使用するために「**N**」を押します。 • データベースがマルチテナントデータベースである • SharePlexが、Amazon RDS上のデータベースなど、リモートデータベースからのキャプチャや、リモートデータベースへのポストを行っている • データベースがクラスタ内にある(Oracle RACなど)
`Are you configuring SharePlex for an AWS RDS database?`	RDS上のOracleデータベース用にSharePlexを設定していない場合は、「**N**」を押します。

プロンプト	対応
	AWS RDSデータベース用にSharePlexを設定している場合は「**Y**」を押します。

以下のいずれかのプロンプトが表示されます。
• BEQUEATH= Yを選択した場合:
`Enter the Oracle SID for which SharePlex should be installed:`
• BEQUEATH = Nを選択した場合:
`Enter the TNS alias for which SharePlex should be installed:`
	非マルチテナントデータベース: デフォルトを受け入れるか、正しいSIDまたはTNSエイリアスを入力します。
	RACでは、TNSエイリアスはグローバルエイリアスでなければなりません。
	マルチテナントデータベース: PDBのTNSエイリアスを入力します。
	Amazon RDS: RDSデータベースのTNSエイリアスを入力します。

以下のいずれかのプロンプトが表示されます。
• データベースがRDS上にない場合:
`Enter a DBA user for SID:`
• データベースがRDS上にある場合:
`In order to create the SharePlex tables and user account, we must connect to the RDS database using the RDS primary user.`
	非マルチテナントデータベース: DBA権限を持つデータベースユーザの名前を入力します。
	マルチテナントデータベース: アカウントとオブジェクトをインストールするために必要な権限を持つ共通のユーザの名前を入力します。
	Amazon RDSデータベース: RDSプライマリユーザの名前を入力します。

以下のいずれかのプロンプトが表示されます。
• データベースがRDS上にない場合:
`Enter password for the DBA account, which will not echo:`
• データベースがRDS上にある場合:
`Enter the password for the RDS primary user, which will not echo.`
	非マルチテナントデータベース: DBAユーザのパスワードを入力します。
	マルチテナントデータベース: 共通ユーザのパスワードを入力します。@と残りの接続文字列は省略します。SharePlexは適切な形式で接続文字列を構築します。
	Amazon RDSデータベース: RDSプライマリユーザのパスワードを入力します。

`Current SharePlex user is user.` `Would you like to create a new SharePlex user?`	「**N**」を押して既存のSharePlexアカウントを更新するか、「**Y**」を押して新しいSharePlex アカウントを作成します。入力を求められたら、資格情報を入力します。 既存のSharePlexユーザの有効なパスワードを入力するには、5回試みることができます。パスワードは難読化されています。 **重要!**アクティブな設定があるときにSharePlexスキーマを変更した場合は、SharePlexオブジェクトを古いスキーマから新しいスキーマにコピーして、レプリケーション環境を維持します。

プロンプト	対応
`Do you want to enable replication of tables with TDE?`	TDEはRDSではサポートされていないため、「**N**」を押します。
`Enter the default tablespace for use by SharePlex:`	**Enter**キーを押してデフォルトを受け入れるか、別のテーブルスペースの名前を入力します。
`Enter the temporary tablespace for use by SharePlex:`	**Enter**キーを押してデフォルトを受け入れるか、別のテーブルスペースの名前を入力します。
`Enter the index tablespace for use by SharePlex:`	**Enter**キーを押してデフォルトを受け入れるか、別のテーブルスペースの名前を入力します。
`Will the current setup for sid: SID be used as a source (including cases as source for failover or primary-primary setups)?`	ソースシステムであれば「**Y**」を、ターゲットシステムであれば「**N**」を押します。 **重要:** プライマリ/プライマリ構成（ピアツーピア）および高可用性構成のすべてのシステムは、レプリケーションの双方向性によりソースシステムと見なされます。

注意:

- 以下のプロンプトは、データベースがASM上のソースである場合にのみ表示されます。
- これがOracleターゲットであれば、これでセットアップは完了です。

`ASM detected. Do you wish to connect to ASM using BEQUEATH connection?`	SharePlexでBEQUEATH接続を使用してASMインスタンスに接続する場合は「**Y**」を、TNSエイリアスを使用する場合は「**N**」を押します。 **重要!** データベースでASMを使用し、さらにSCAN IPを使用してデータベースのTNSエイリアスを設定している場合、SharePlexをASMインスタンスに接続するために、ASM TNSエイリアスを介した接続を指定しなければなりません。
BEQUEATH接続を選択していない場合は、次のプロンプトが表示されます。 `Do you wish to keep connecting using the same user/password?`	ログインユーザと同じユーザとパスワードを使用する場合は「**Y**」を、異なるユーザとパスワードを使用する場合は「**N**」を押します。 通常、SharePlexを実行しているユーザはOSASMグループのメンバーでなければなりません。これは、SP_OCT_OLOG_USE_OCIがデフォルト以外の値である1に設定されている場合は適用されません。 また、BEQUEATH接続を使用している場合、SharePlexを実行しているユーザはOSDBAグループのメンバーでなければなりません。

プロンプト	対応
注意: • BEQUEATH接続を使用してASMに接続することを選択した場合、データベースセットアップは完了です。この表の後に「Oracle Cloud-ExaCSとDBCSのデータベースセットアップ。 • 「**N**」を選択した場合はTNSエイリアスを入力する必要があり、プロンプトが続けて表示されます。	
`Enter the ASM tns alias to be used by SharePlex:`	TNSエイリアスの名前を入力します。
`Enter an ASM admin (has both sysdba and sysasm privilege) username for alias:`	ASMインスタンスにsysasmおよびsysdba権限を持つユーザの名前を入力します。
`Enter user password for user:`	ユーザのパスワードを入力します。
SharePlexは、Oracle Spatial and GraphオプションのSDO_GEOMETRYデータ型をサポートするパッケージを含む内部オブジェクトをインストールします。このオプションがデータベースにインストールされていない場合、以下のプロンプトが表示されます。 `The SharePlex object that supports replication of SDO_GEOMETRY cannot be installed because the Oracle Spatial and Graph feature is not installed. Do you want to continue with the setup without support for SDO_GEOMETRY? [n]:`	「**Y**」を押してSDO_GEOMETRYをサポートせずにデータベースセットアップを続行するか、「**N**」を押して**ora_setup**を終了します。

7. Oracleのセットアップが完了した後、ora_setup.configファイルが<Installation_directory>/var/data内に生成されます。

tnsnamesファイルについて

SharePlexを、TNSエイリアスを使用したデータベース接続と、BEQUEATH接続（OS認証）を使用したローカルでのASM接続（OS認証）用にセットアップした場合は、各ノードで**tnsnames.ora**ファイルを正しく設定することが重要です。SharePlexのデータベースアカウントがプライマリノードに存在するものと仮定すると、SharePlexは、常にプライマリASM_SIDに自動的に接続されます。これは、SharePlexがインストールされたときに提供されているからです。しかし、フェールオーバー時には、SharePlexはローカルの**v$asm_client**ビューにクエリを実行し、フェールオーバーインスタンスの正しいASM_SIDを取得する必要があります。したがって、指定されたノードのローカル**tnsnames.ora**ファイルでは、そのノードのIPアドレスが常に最初にリストされるようにしてください。

7

SharePlexの変数

SharePlexは以下の環境変数を使用します。状況によってはこれらの環境変数を設定する必要があります。通常、変数を設定する前または後に追加の手順を実行する必要があるため、SharePlexの変数を設定する前に推奨手順を参照してください。

環境変数	説明
EDITOR	**create config**コマンドなどで使用する、**sp_ctrl**コマンドのデフォルトのASCIIテキストエディタを設定します。
HOST	**sp_ctrl**のローカルで実行されるすべてのセッションのホスト名を設定します。
SP_COP_TPORT	SharePlexのインスタンスにデフォルト以外のポート番号を設定します。デフォルトのポート番号は2100です。以下のいずれかに当てはまる場合は、別のポート番号を設定しなければならないことがあります。 • **sp_cop**の追加のインスタンスをセットアップしている場合 • 2100以外のポート番号を使用しなければならない場合
SP_SYS_HOST_NAME	設定のアクティベーションの際にSharePlexがバインドするホスト名を設定します。この変数は以下のように使用されます。 • Oracle RACなどのクラスタ化システムで仮想IPアドレス(*グローバル・クラスタ・パッケージ名*とも呼ばれる)を設定します。この変数はすべてのクラスタノードで設定する必要があります。 • SP_SYS_HOST_NAMEがソースシステムのIPV6アドレスに設定されている場合、ターゲットシステムのSharePlexはバージョン9.0以降でなければなりません。
SP_SYS_VARDIR	**sp_cop**が設定データ、キュー、ログ、その他の情報を見つけられるように、SharePlexの変数データディレクトリへのフルパスを設定します。ローカルシステム上に**sp_cop**のインスタンスが1つしかない場合、この変数はデフォルトで設定されます*。ローカルシステム上に**sp_cop**のインスタンスが複数ある場合、そのインスタンスに他のSharePlex変数を設定する前に常にこの変数を設定し、そのインスタンスの正しい変数データディレクトリを指すようにします。
SP_SYS_SECURE_MODE	SharePlexの現在のインスタンスが実行されている間、すべてのcompareとrepairの実行に関するcompareとrepairのSQLログファイルの出力を抑制します。この変数はSharePlexを開始する前に設定しなければなりません。したがって、**sp_cop**プロセスが実行されている場合は、この変数を設定した後に再起動しなければなりません。この環境変数を指定して**sp_cop**を実行すると、compareコマンドとrepairコマンドによってSQLファイルにデータは保存されず、Post処理でもSharePlexのエラーログにデータは保存されません。

* UnixとLinuxでは、変数データディレクトリは*proddir*/**data**/**default.yaml**ファイルに設定されます。

UnixやLinuxで環境変数を設定するには:

kshシェル:

export *variable_name*=*value*

cshシェル:

setenv *variable_name value*

kshシェル:

export SP_SYS_VARDIR=*full_path_of_variable-data_directory*

cshシェル:

setenv SP_SYS_VARDIR *full_path_of_variable-data_directory*

会社概要

新たなQuestを目指して

当社は、お客様のビジネスでITをさらに活用できるよう、日々さまざまな取り組みを行っております。お客様のITの管理にかかる時間を減らし、ビジネスのイノベーションに注力できるよう、コミュニティを活用したソフトウェアソリューションを作り上げました。当社は、データセンターの刷新やクラウドの迅速な導入をお手伝いし、データ主導型ビジネスの成長に不可欠な専門知識、セキュリティ、アクセス機能を提供いたします。Questのイノベーションの一環であるグローバルコミュニティやお客様にご満足いただくためのさまざまな取り組みを通じて、現代のビジネスに大きな影響を与えるソリューションを今後も提供していくと共に、これまで培ってきた当社の伝統も維持してまいります。次世代のソフトウェア企業として、常に新しい分野へチャレンジしていきます。パートナーである皆様とも、さらに緊密な関係を維持し、ビジネスにとって最適なIT環境の構築をお手伝いします。このミッションの実現に向けて、一致団結して取り組んでまいります。新しいQuestにご期待ください。Join the Innovation™にご参加ください。

当社のブランドとビジョン、一致団結

当社のロゴは、イノベーション、コミュニティ、サポートという当社のストーリーを反映したものとなっています。こうしたストーリーを語るうえで重要な文字が、Qなのです。この美しい円形は、当社が誇る高精度で強力なテクノロジーを表しています。Q自体に含まれているスペースは、新しいQuestというコミュニティにとって今後欠かせなくなってくる存在、つまり「お客様」を表しています。

Questのお問い合わせ先

ご購入とその他のお問い合わせは、www.quest.com/contactをご覧ください。

テクニカルサポートのリソース

有効なメンテナンス契約を締結したQuestのお客様、および評価版をご利用のお客様は、テクニカルサポートをご利用いただけます。Questサポートポータルにはhttps://support.quest.com/からアクセスしてください。

サポートポータルには、24時間365日いつでもアクセスすることができ、セルフヘルプツールを使って、お客様ご自身で問題を迅速に解決できます。サポートポータルでは以下のことができます。

- サービスリクエストの提出と管理
- ナレッジベースの記事の閲覧
- 製品通知の登録
- ソフトウェアと技術文書のダウンロード
- ハウツービデオの視聴
- コミュニティへの参加
- サポートエンジニアとのオンラインチャット
- 製品に関するサービスの確認