Quest

SharePlex® 12.0

Reference Guide

© 2026 Quest Software Inc. ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

Refer to our Web site (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, SharePlex, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. All other trademarks and registered trademarks are property of their respective owners.

SharePlexReference Guide
Updated - 8/14/2026
Version - 12.0

Contents

Revision History **27**

About this Guide **28**

Other SharePlex Documentation 28

Conventions Used in this Guide **29**

SharePlex Commands for Oracle **30**

SharePlex Commands at a Glance 30

Configuration Commands 30

Connect commands 31

Encryption commands 32

Information commands 32

Maintenance commands 33

Parameter commands 34

Partitioning commands 34

Properties commands 34

Route commands 35

Run commands 35

Synchronization commands 35

Target configuration command 36

Miscellaneous commands 37

Alphabetical Reference for SharePlex Commands 38

Abort config 38

Usage 38

Syntax 38

Abort service 40

Usage 40

Syntax 40

Activate config 44

Guidelines for activation 44

Usage 45

Syntax 46

Add partition 48

Usage 48
Syntax 48
Analyze config 51
Usage 52
Syntax 53
Append status 55
Usage 55
Syntax 55
Authlevel 57
Usage 57
Syntax 57
Cancel 59
Usage 59
Syntax 59
Clear history 61
Usage 61
Syntax 61
Clear status 63
Usage 63
Syntax 63
Compare status 65
Usage 65
Syntax 65
Connection 68
Usage 68
Syntax 68
Copy / append 74
The copy command 74
The append command 74
The launcher process 75
Overview of copy/append process 76
Running concurrent copy processes 76
Identifying synchronization processes 76
Tuning parameters for the synchronization 77
Viewing copy status 77
Using a synchronization routing file 78

Copying dissimilar source and target tables 78
Using copy with partitioned replication 79
Controlling the number of processing threads 79
Using Oracle wildcards in command line syntax 79
Other considerations for using copy/append 80
Usage 80
Syntax 81
Copy cache 84
Usage 84
Syntax 84
Copy config 86
Usage 86
Syntax 86
Copy status 88
Usage 88
Syntax 88
Create config 90
Usage 90
Syntax 90
Create encryption key 92
Usage 92
Syntax 92
Deactivate config 94
Usage 94
Syntax 94
Delete capture queue 96
Usage 96
Syntax 96
Delete export process 98
Usage 98
Syntax 98
Delete export queue 100
Usage 100
Syntax 100
Delete post queue 102
Usage 102

Syntax 102
Disable jobs / enable jobs 104
Usage 104
Syntax 104
Drop partition 106
Usage 106
Syntax 106
Drop partition scheme 108
Usage 108
Syntax 108
Edit command 109
Usage 109
Syntax 109
Edit config 110
Usage 110
Syntax 110
Exit command 112
Usage 112
Syntax 112
Expand command 113
Usage 113
Syntax 113
Flush command 115
Starting Post again 115
Usage 115
Syntax 115
Help command 119
Usage 119
Syntax 119
Host command 121
Usage 121
Syntax 121
Hostinfo command 123
Usage 123
Syntax 123
Job status command 125

Usage 126
Syntax 126
List config 129
Usage 129
Syntax 129
List param 131
SharePlex parameter modules 131
Usage 132
Syntax 132
lstatus command 134
Understanding the SharePlex queues 134
Usage 135
Syntax 135
Modify partition 137
Usage 137
Syntax 137
Orainfo command 139
Usage 139
Syntax 139
Port command 141
Usage 141
Syntax 141
Purge config 143
Usage 143
Syntax 143
Qstatus command 145
Usage 145
Syntax 146
Quit command 147
Usage 147
Syntax 147
Reconcile command 148
Usage 148
Syntax 149
Redo command 151
Usage 151

Syntax 151
Remove config 152
Usage 152
Syntax 152
Remove log 154
Usage 154
Syntax 154
Rename config 156
Usage 156
Syntax 156
Repair status 158
Usage 158
Syntax 158
Report 160
Usage 160
Syntax 160
Reset encryption key 162
Usage 162
Syntax 162
Reset param 164
Using command options 164
Viewing current parameter settings 164
Usage 165
Syntax 165
Set encryption key 168
Usage 168
Syntax 168
Set log 170
Usage 170
Syntax 170
Set param 172
Using command options 172
Usage 173
Syntax for global settings 173
Syntax for process-specific settings 174
Show command 178

Process 178
Source 178
Target 179
State 179
PID 179
Usage 180
Syntax 180
Use of Show Capture 182
Basic command 182
Detailed statistics 182
Detailed statistics for Oracle Capture 182
Usage 184
Syntax 184
Show Config 186
Usage 188
Syntax 188
Show Encryption Key 190
Usage 190
Syntax 190
Show Export 192
Usage 192
Syntax 192
Show Import 195
Usage 195
Syntax 195
Show Log 197
Usage 197
Syntax 197
Show Post 201
Basic show post command 201
Detailed show post command 201
Usage 204
Syntax 204
Show Read 207
Basic Command 207
Detailed statistics 207

Usage 209
Syntax 209
Show SQL 211
Usage 211
Syntax 211
Show Statusdb 214
Usage 214
Syntax 214
Show SCN 216
Usage 216
Syntax 216
Show Sync 218
Usage 218
Syntax 218
Shutdown 220
Shutting down gracefully 220
Shutting down forcefully 220
Usage 220
Syntax 220
Start 222
Usage 222
Syntax 222
Startup 225
Usage 225
Syntax 225
Status 227
Usage 227
Syntax 227
Stop 229
Stopping the Capture process 229
Stopping Post at a specific point in time 229
Usage 230
Syntax 230
Target 234
Usage 234
Syntax 234

CDC category ... 237
Characterset category ... 237
file category ... 238
format category ... 239
JSON record format ... 245
AVRO record format ... 248
JMS category ... 249
JSON category ... 251
kafka category ... 256
Metadata category ... 260
Resources category ... 263
Rule | Filter category ... 264
Source category ... 266
SQL category ... 267
Trace Capture ... 269
Usage ... 269
Syntax ... 269
Output ... 270
Trace Post ... 272
Usage ... 272
Syntax ... 272
Output ... 274
Trace Read ... 276
Usage ... 276
Syntax ... 276
Output ... 277
Truncate Log ... 279
Usage ... 279
Syntax ... 279
Typemap ... 281
Usage ... 281
Syntax ... 281
Usage ... 283
Usage ... 283
Syntax ... 283
Verify Config ... 284

Usage ... 285
Syntax ... 285
Version ... 287
Usage ... 287
Syntax ... 287
View Config ... 289
Usage ... 289
Syntax ... 289
View Log options ... 291
Usage ... 291
Syntax ... 291
View Partitions ... 293
Usage ... 293
Syntax ... 293

SharePlex Parameters for Oracle ... 294
Descriptions of User-Configurable Parameters ... 294
Configuration activation parameters ... 295
SP_OCF_HASH_BY_BLOCK ... 295
SP_OCF_LOCK_WAIT_TIME ... 295
SP_OCF_THREAD_COUNT ... 295
Capture Parameters ... 296
SP_OCT_ALLOW_DP_DDL ... 296
SP_OCT_ARCH_LOC ... 296
SP_OCT_ASM_CACHE_AGE ... 297
SP_OCT_ASM_ECACHE_SIZE ... 297
SP_OCT_ASM_FLIST_SIZE ... 297
SP_OCT_ASM_MULTI_OCI_BLOCK_SIZE ... 297
SP_OCT_ASM_MULTI_OCI ... 298
SP_OCT_ASM_SLIST_SIZE ... 298
SP_OCT_ASM_SUPPORT ... 298
SP_OCT_AUTOADD_ENABLE ... 298
SP_OCT_AUTOADD_MVIEW ... 299
SP_OCT_AUTOADD_SEQ ... 300
SP_OCT_CHECKPOINT_LOG ... 300
SP_OCT_CHECKPOINT_TIME ... 300

SP_OCT_CK_LOC_FIRST 301
SP_OCT_DATA_OBJ_CACHE_SIZE 301
SP_OCT_DATE_ MSG 301
SP_OCT_DDL_LOGGING 301
SP_OCT_DDL_UPDATE_CONFIG 302
SP_OCT_DEF parameters 302
SP_OCT_DENIED_USERID 302
SP_OCT_DLOAD_MAX_COLS_PER_MSG 303
SP_OCT_ENABLE_LOBMAP 303
SP_OCT_INCLUDE_UNCHANGED_COL 303
SP_OCT_INSERT_INCLUDE_NULLS 304
SP_OCT_LOB_BUFFER_SIZE 304
SP_OCT_LOG_FILESIZE 304
SP_OCT_LOG_MEMBER 304
SP_OCT_LOG_NUMFILES 305
SP_OCT_LOG_READ_SIZE 305
SP_OCT_LOGWRAP_RESTART 305
SP_OCT_MIN_SESSIONS 306
SP_OCT_OLOG_DELAY 306
SP_OCT_OLOG_NO_DATA_DELAY 306
SP_OCT_OLOG_QUEUE_SIZE 306
SP_OCT_OLOG_RDS_MINER 307
SP_OCT_OLOG_REOPEN 307
SP_OCT_OLOG_USE_OCI 307
SP_OCT_ONELINE_DEBUG 308
SP_OCT_OPS_LOGREADER_DELAY 308
SP_OCT_OPS_LOGREADER_RETRY 308
SP_OCT_OPS_NO_DATA_DELAY 309
SP_OCT_PREFER_ARCHIVE 309
SP_OCT_READ_ARC 309
SP_OCT_REDOLOG_ENSURE 310
SP_OCT_REDUCED_KEY 310
SP_OCT_REPLICATE_ALL_DDL 311
SP_OCT_REPLICATE_COMMENT 312
SP_OCT_REPLICATE_CTAS 312
SP_OCT_REPLICATE_DDL 313

SP_OCT_REPLICATE_DLOAD ... 314
SP_OCT_REPLICATE_GRANT ... 314
SP_OCT_REPLICATE_MVIEW ... 314
SP_OCT_REPLICATE_POSTER ... 315
SP_OCT_REPLICATE_SEQUENCES ... 315
SP_OCT_REPLICATE_SYNONYM ... 315
SP_OCT_REPLICATE_TRIGGER ... 315
SP_OCT_REQUIRED_DATA_IS_LOGGED ... 316
SP_OCT_TRUNC_PARTITION_BY_ID ... 316
SP_OCT_USE_DST ... 317
SP_OCT_USE_SUPP_KEYS ... 317
Read parameters ... 318
SP_ORD_BATCH_ENABLE ... 318
SP_ORD_BATCH_MAX ... 318
SP_ORD_BATCH_MATCH_MIN ... 318
SP_ORD_CDA_LIMIT ... 318
SP_ORD_DATE_ MSG ... 319
SP_ORD_DELAY_RECORDS ... 319
SP_ORD_FIRST_FIND ... 319
SP_ORD_HP_HASH ... 320
SP_ORD_HP_IN_SYNC ... 320
SP_ORD_LDA_ARRAY_SIZE ... 321
SP_ORD_ONELINE_DEBUG ... 321
SP_ORD_MSGS_CK_FREQ and SP_ORD_RCM_SKIP_RATIO ... 322
SP_ORD_RMSG_LIMIT ... 323
SP_ORD_ROLLBACK_TXNS_MAX ... 323
SP_ORD_SEND_DDL_TO_FIRST ... 323
Export parameters ... 325
SP_XPT_AES_KEY_LENGTH ... 325
SP_XPT_ENABLE_AES ... 325
SP_XPT_ENABLE_COMPRESSION ... 325
SP_XPT_KEEPALIVE ... 326
SP_XPT_PORT_OVERRIDE ... 326
SP_XPT_SO_SNDBUF ... 326
SP_XPT_USE_LOCALHOST ... 327
Import parameters ... 328

SP_IMP_ENABLE_AES 328
SP_IMP_QUEUE_PAUSE 328
SP_IMP_QUEUE_RESUME 329
SP_IMP_WCMT_MSGCNT 329
SP_IMP_WCMT_TIMEOUT 329
Oracle Poster parameters 330
SP_OPO_CHANGE_ID_START_VALUE 330
SP_OPO_COMMIT_REDUCE_MSGS 330
SP_OPO_CONNECTION_POOL 330
SP_OPO_CONT_ON_ERR 331
SP_OPO_DEPENDENCY_CHECK 332
SP_OPO_DEPENDENCY_SIZE 332
SP_OPO_DISABLE_OBJECT_NUM 332
SP_OPO_HINTS_LIMIT 333
SP_OPO_LOG_CONFLICT 333
SP_OPO_MAX_CDA 334
SP_OPO_MAX_OEXN_TIME 334
SP_OPO_MAX_ROLLBACK_MESSAGES 335
SP_OPO_NLS_CONVERSION 335
SP_OPO_NLS_DEFAULT_COMPAT 336
SP_OPO_OBJID_DEBUG 336
SP_OPO_ONELINE_DEBUG 336
SP_OPO_OUT_OF_SYNC_SUSPEND 337
SP_OPO_POSTER_DELAY 338
SP_OPO_PRB_MISMATCH_SUSPEND 339
SP_OPO_READRELEASE_INTERVAL 339
SP_OPO_REDUCED_KEY 339
SP_OPO_RETRIES_MAX 340
SP_OPO_SAVE_OOS_TRANSACTION 341
SP_OPO_SQL_CACHE_DISABLE 342
SP_OPO_STOP_ON_DDL_ERR 342
SP_OPO_SUPPRESS_DELETE 343
SP_OPO_SUPPRESSED_OOS 343
SP_OPO_SYNC_LOG_FREQUENCY 343
SP_OPO_THREADS_MAX 344
SP_OPO_TRACK_COMMITS 344

SP_OPO_TRACK_OPERATIONS 344
SP_OPO_TRACK_PREIMAGE 345
SP_OPO_TRUSTED_SOURCE 345
SP_OPO_UPDATE_SCN 345
SP_OPO_USE_VARNUM 346
SP_OPO_WAIT_MSG_DELAY 346
SP_OPO_LOB_PERFORMANCE_ENABLE 346
Open Target Post parameters 347
SP_OPX_BATCH_ENABLE 347
SP_OPX_COMMIT_REDUCE_MSGS 347
SP_OPX_CONT_ON_ERR 347
SP_OPX_DISABLE_OBJECT_NUM 348
SP_OPX_MAX_BATCH_MBYTES 348
SP_OPX_MSGS_IN_MEMORY 349
SP_OPX_NLS_CONVERSION 349
SP_OPX_ONELINE_DEBUG 350
SP_OPX_OUT_OF_SYNC_SUSPEND 351
SP_OPX_READRELEASE_INTERVAL 351
SP_OPX_REDUCED_KEY 352
SP_OPX_SQL_CACHE_DISABLE 352
SP_OPX_STOP_ON_DDL_ERR 353
SP_OPX_THREADS 353
SP_OPX_TRUSTED_SOURCE 353
Queue parameters 354
SP_QUE_MAX_QUEUES 354
SP_QUE_POST_SHMSIZE 354
SP_QUE_Q_SHMSIZE 354
SP_QUE_SHMDBUF 354
SP_QUE_SYNC 355
SP_QUE_USE_SUBQUE_INDEX 356
SP_SHS_SHMSIZE 357
sp_cop parameters 358
SP_COP_IDLETIME 358
SP_COP_SCAN_TIMEOUT 358
SP_COP_SO_RCVBUF 358
SP_COP_TPORT 359

SP_COP_UPORT 359
Log parameters 360
SP_SLG_LOG_MAXSIZE 360
SNMP parameters 361
SP_SLG_SNMP_ACTIVE 361
SNMP agent parameters 361
Custom MIB parameters 361
SNMP trap parameters 362
System parameters 363
SP_SYS_HOST_NAME 363
SP_SYS_IN_SYNC 364
SP_SYS_JOB_HISTORY_RETENTION 364
SP_SYS_JOB_HISTORY_SIZE 364
SP_SYS_SUSPEND_ON_ERROR 365
SP_SYS_TARGET_COMPATIBILITY 365
SP_SYS_VAR_FULL 366
SP_SYS_VAR_OK 366
Compare/Repair Parameters 367
SP_DEQ_IGNORE_ORACLE_ERROR_NUM 367
SP_DEQ_IP_MAPPING 367
SP_DEQ_LOG_FILESIZE 368
SP_DEQ_MALLOC 368
SP_DEQ_PARALLELISM 368
SP_DEQ_PARTIAL_REPAIR_CLR_OOS 368
SP_DEQ_PART_TABLE_UPDATE 369
SP_DEQ_READ_BUFFER_SIZE 369
SP_DEQ_ROW_LOCK_THRESHOLD 370
SP_DEQ_SKIP_LOB 370
SP_DEQ_THREADS 370
SP_DEQ_TIMEOUT 371
SP_DEQ_USE_SP_CKSUM 371
SP_DEQ_ORA_DATE_FORMAT 372
Copy/Append command parameters 373
SP_OSY_COMPRESSION 373
SP_OSY_LOCK_TIMEOUT 373
SP_OSY_POST_TIMEOUT 373

SP_OSY_THREADS ... 373
Analyze config command parameters ... 374
SP_ANL_CHECKPOINT_FREQ ... 374
SP_ANL_CHECKPOINT_TIME ... 374
SP_ANL_RUN_TIME ... 374
Deprecated Parameters ... 375
Deprecated Capture parameters ... 375
Deprecated Read parameters ... 375
Deprecated Oracle Post parameters ... 376
Deprecated Open Target Post parameters ... 376
Deprecated Export parameters ... 376
Deprecated Compare parameters ... 377
Deprecated queue parameters ... 377
Other deprecated parameters ... 377

SharePlex Commands for PostgreSQL ... 378
SharePlex Commands at a Glance ... 378
Configuration Commands ... 378
Information commands ... 379
Partitioning commands ... 379
Synchronization commands ... 380
Maintenance commands ... 380
Alphabetical Reference for SharePlex Commands for PostgreSQL ... 381
Abort Config for PostgreSQL ... 381
Usage ... 381
Syntax ... 381
Activate Config for PostgreSQL ... 382
Guidelines for activation ... 382
View activation status and results ... 382
What to do if activation fails ... 383
Usage ... 383
Syntax ... 383
Add Partition for PostgreSQL ... 385
Usage ... 385
Syntax ... 385
Cancel command for PostgreSQL ... 388

Clear History command for PostgreSQL 390
Clear Status command for PostgreSQL 392
Compare status command for PostgreSQL 394
Copy config for PostgreSQL 397
Usage 397
Syntax 397
Create Config for PostgreSQL 398
Usage 398
Syntax 398
Deactivate config for PostgreSQL 400
Usage 400
Syntax 400
Drop Partition for PostgreSQL 401
Usage 401
Syntax 401
Drop Partition Scheme command for PostgreSQL 403
Usage 403
Syntax 403
Edit config for PostgreSQL 404
Usage 404
Syntax 404
Flush for PostgreSQL 405
Starting Post again 405
Usage 405
Syntax 405
Job status command for PostgreSQL 408
List config for PostgreSQL 412
Usage 412
Syntax 412
Modify Partition for PostgreSQL 413
Usage 413
Syntax 413
Purge config for PostgreSQL 415
Usage 415
Syntax 415
Reconcile for PostgreSQL 416

Usage 417
Syntax 417
Rename config for PostgreSQL 419
Usage 419
Syntax 419
Repair status command for PostgreSQL 420
Verify config for PostgreSQL 422
Usage 422
Syntax 423
View Partitions for PostgreSQL 424
Usage 424
Syntax 424
Show Capture for PostgreSQL 425
Basic command 425
Detailed statistics 425
Usage 428
Syntax 429
Show Post for PostgreSQL 430
Basic Show Post command 430
Detailed Show Post command 430
Usage 432
Syntax 432
show_last_posted 434
Usage 434
Syntax 434

SharePlex Parameters for PostgreSQL 435
Descriptions of User-Configurable Parameters 436
Capture Parameters 436
SP_CAP_INCLUDE_UNCHANGED_COL 436
SP_CAP_MIN_SESSIONS 436
SP_CAP_REDUCED_KEY 436
SP_CAP_ONELINE_DEBUG 437
SP_CAP_CHECKPOINT_FREQ 437
SP_CAP_CHECKPOINT_TIME 438
SP_CAP_DENIED_SESSION_PG 438

SP_CAP_DENIED_PORT ... 439
Open Target Poster parameters ... 440
SP_OPX_REDUCED_KEY ... 440
SP_OPX_XML_MARK_NULL ... 440
SP_OPX_CREATE_ORIGIN_PG ... 441
SP_OPX_LOG_CONFLICT ... 441
List of additional supported parameters for PostgreSQL Open Target Poster ... 442
Oracle Poster parameters ... 443
SP_OPO_HINTS_LIMIT ... 443
List of additional supported parameters for Oracle Poster process ... 443
Read parameters ... 444
SP_ORD_FIRST_FIND ... 444
SP_ORD_HP_IN_SYNC ... 444
List of additional supported parameters for PostgreSQL Reader process ... 445
Compare/Repair Parameters ... 446
SP_DEQ_PG_FORCE_PARALLEL ... 446
SP_DEQ_PG_PARALLEL_MAX ... 446
SP_DEQ_PG_PARALLEL_SETUP_COST ... 447
SP_DEQ_PG_PARALLEL_TUPLE_COST ... 447
SP_DEQ_BATCH_ENABLED ... 447
SP_DEQ_MAX_BATCH_MBYTES ... 447
SP_DEQ_COMMIT_FREQ ... 448
SP_DEQ_USE_SP_CKSUM ... 448
SP_DEQ_PG_DECLARE_FETCH_SIZE ... 448
SP_DEQ_READ_BUFFER_SIZE for PostgreSQL ... 449
Export parameters ... 449
Import parameters ... 449
Queue parameters ... 449
sp_cop parameters ... 450
System parameters ... 450

Heterogenous compare-repair commands ... 451
Compare / compare using ... 451
Supported operations ... 451
Non-supported operations for Oracle as a source ... 451
Non-supported operations for PostgreSQL as a source ... 452

Other conditions ... 452
Supported Charactersets ... 453
Special use cases ... 454
How the comparison works ... 456
Manage the SQL log file ... 457
Run multiple processes ... 458
Compare a subset of a configuration ... 459
Control which rows are compared ... 459
Identify processes ... 460
View status and results in sp_ctrl ... 460
View warnings and errors for Oracle to Oracle replication ... 460
View warnings and errors for Oracle to PostgreSQL and PostgreSQL to PostgreSQL replication ... 461
Cancel a compare job ... 461
Manage compare history and logs ... 462
Control the batch size ... 462
Usage ... 462
Syntax ... 462
Repair / repair using ... 475
Supported operations ... 475
Non-supported operations for Oracle as a source ... 475
Non-supported operations for PostgreSQL as a source ... 476
Other conditions ... 476
Supported Charactersets ... 477
How to use the repair commands ... 477
When to run the repair ... 477
Special use cases ... 477
How the repair works ... 480
The conditions for a repair ... 480
The process ... 481
Error handling ... 481
Run multiple processes ... 482
Repair a subset of a configuration ... 483
Control which rows are repaired ... 483
Identify processes ... 483
View status and results in sp_ctrl ... 483

View warnings and errors for Oracle to Oracle replication ... 483
View warnings and errors for Oracle to PostgreSQL and PostgreSQL to PostgreSQL replication ... 484
Cancel a repair job ... 485
Manage compare history and logs ... 485
Control the batch size ... 485
Usage ... 485
Syntax ... 485

General SharePlex Utilities ... 497
Configuration Scripts ... 498
Supported source and target database ... 498
Use config.sql ... 498
Use build_config.sql ... 499
Configuration Scripts for PostgreSQL ... 501
Supported source and target database ... 501
Use pg_config.sql ... 501
Use pg_build_config.sql ... 502
Cleanup.sql ... 505
Supported databases ... 505
Clean_vardir.sh ... 506
Supported databases ... 506
Shell requirement ... 506
Create_ignore.sql ... 508
Supported databases ... 508
Run create_ignore.sql ... 508
Execute SHAREPLEX_IGNORE_TRANS ... 509
MSS_cleansp ... 510
Supported databases ... 510
MySQL_cleansp ... 512
Usage guidelines ... 512
Supported databases ... 512
Ora_cleansp ... 514
Supported databases ... 514
Other alternatives to ora_cleansp ... 514
Shell requirement ... 514
How ora_cleansp works ... 514

pg_cleansp Utility 517
Usage guidelines 517
What this utility does 517
Snowflake_Cleansp Utility 519
Usage guidelines 519
What this utility does 519
SharePlex License Utilities 521
License utilities 521
splex utility 521
Add a license key 522
View a license key 523
Remove a license key 524
spUtil utility 524
Add a license key 524
View the license details 525
Remove a license key 525
Provision 527
Supported databases 527
Guidelines for using provision 527
Run provision 528
Undo changes made by provision 529
Known issues 529
Qview 530
Supported databases 530
Run qview 530
Overview of qview commands 530
List queues 531
Description 531
Syntax 531
Trim obsolete subqueues 531
Description 532
How to run this command 532
Syntax 532
Execute a full rollback 533
Description 533
How to run this command 533

Syntax ... 534
Show_scn utility ... 535
Supported databases ... 535
Run show_scn ... 535
Output ... 535
show_last_posted Utility ... 536
Supported databases ... 536
Run show_last_posted ... 536
Output ... 536
Socket_test ... 538
Supported databases ... 538
Run socket_test ... 538
SP-bininfo ... 540
Supported databases ... 540
Supported platforms ... 540
Run sp-bininfo ... 541
SP_wallet ... 542
Supported databases ... 542
Run sp_wallet ... 542
sp_hsm ... 545
Supported databases ... 545
Run the sp_hsm Utility ... 545
sp_security ... 546
Description ... 546
Enable SSL/TLS ... 546
Disable SSL/TLS ... 547
View current SSL/TLS configuration ... 548
Trigger Scripts ... 549
Supported databases ... 549
sp_add_trigger.sql ... 549
Syntax ... 549
sp_change_trigger.sql ... 550
Syntax ... 550
sp_remove_trigger.sql ... 550
Syntax ... 550
Trigger Scripts for PostgreSQL ... 551

Supported databases 551
sp_pg_add_trigger.sql 551
Syntax 552
sp_pg_change_trigger.sql 552
Syntax 552
sp_pg_remove_trigger.sql 552
Syntax 552
Database Setup Utilities 553

SharePlex for Oracle Cloud Infrastructure 554
Database Setup for Oracle Cloud-ExaCS and DBCS 554
Supported databases 554
When to run Oracle setup 554
Supported Oracle connections 555
Required run privileges 555
Privileges granted to SharePlex 556
Requirements 556
SharePlex schema storage requirements 557
Run database setup for Oracle cloud-ExaCS and DBCS 558

SharePlex Variables 562

About us 564
Contacting Quest 564
Technical Support Resources 564

Revision History

Document Version	Date	Change History
2	18th February 2026	Updated the Database Setup for Snowflake section with information regarding RSA-based authentication with a passphrase.
3	14th August 2026	Updated the maximum valid value for the SP_DEQ_MALLOC parameter to 2047 MB.

About this Guide

This guide is a reference for the following:

- **SharePlex commands** — an overview of the commands that control replication through sp_ctrl, an explanation of command authorization levels, and detailed documentation for each command.
- **SharePlex parameters** — an overview of the SharePlex tuning parameters, how they are set and stored, and detailed documentation for parameters that can be changed by SharePlex users.
- **SharePlex utilities** — the SharePlex utilities and their use.
- **SharePlex error messages** — common error codes, their cause, and the suggested resolution.

Users of the SharePlex Reference Guide should have a thorough understanding of SharePlex before issuing commands, changing parameters or attempting to solve problems.

Other SharePlex Documentation

For the complete SharePlex documentation set, go to https://support.quest.com/shareplex/technical-documents.

Conventions Used in this Guide

Conventions used in this manual

The following typographic conventions are used in this guide:

- **Bold** represents required components of a command or option that must be typed as shown.
- *Italics* represent variables defined, named or entered by the user.
- {Braces} enclose required arguments.
- [Brackets] represent optional command components and may also be used in example command strings to emphasize required user defined variables in long strings.

  Example:

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

- A vertical bar, or “pipe” character, (|) within brackets or braces indicates that you can use only one of the enclosed components.

  Example:

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

Names of commands, programs, directories and files are expressed in **Bold**.

Other names are expressed in capital letters using the default font.

Examples:

The **sp_ctrl** program is located in the **bin** directory.

Open the **oramsglst** file.

Find the value for ORACLE_HOME.

Click **Apply**.

System displays, such as prompts and command output, are expressed in a `monofaced` (fixed-space) font.

Examples:

```
sp_ctrl(sysA)>
```

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

Windows menu items, dialog boxes, and options within dialog boxes are expressed in **Bold**.

Example:

From the **File** menu, select **Print**.

System names are expressed generically or fictitiously. When necessary, the source system (or primary system) is referred to as *SysA*. Target systems (or secondary systems) are referred to as *SysB*, *SysC*, *SysD*, and so forth.

1

SharePlex Commands for Oracle

The SharePlex commands configure, start, stop, control, and monitor the replication process. SharePlex commands are issued through the **sp_ctrl** interface.

Contents

SharePlex Commands at a Glance

Alphabetical Reference for SharePlex Commands

SharePlex Commands at a Glance

This section provides an overview of all SharePlex commands, grouped according to the processes or functions they control.

For more information about SharePlex command authorization levels and security groups, see the SharePlex Administration Guide.

Configuration Commands

The configuration commands control all aspects of managing SharePlex configuration files.

List of configuration commands

Command	Auth. Level	Supported targets	Description
Abort config	1	All	Immediately terminates replication for the designated configuration file whether or not data has posted. Deactivates the configuration and deletes queues and processes.
Activate config	1	All	Begins replication of the data specified in a configuration.
Copy config	2	All	Duplicates a configuration to edit and save as a new file.
Create config	2	All	Creates a new configuration file in which you enter information about objects to be replicated.
Deactivate config	1	All	Gracefully terminates replication of the data specified in a configuration, allowing data in the queues to be posted to the target database before the queues are deleted.
Edit config	2	All	Modifies an existing configuration file using the default text editor.
List config	3	All	Displays all of the configuration files on a source system, both active and inactive.
Purge config	1	All	Removes data from queues generated by an active configuration, but

Command	Auth. Level	Supported targets	Description
			does not deactivate the configuration or remove the queues themselves.
Remove config	2	All	Deletes a configuration file from the system permanently.
Rename config	2	All	Assigns a different name to a configuration file.
Verify Config	3	All	Verifies that the objects in a configuration are valid and that the configuration is structured properly to ensure successful activation and replication.
View Config	3	All	Displays the contents of a configuration file.

Connect commands

The connect commands control connections made to replication systems through the **sp_ctrl** interface.

List of connect commands

Command	Auth. level	Supported targets	Description
Connection	2	All	Specifies connection settings to connect to a source or target database.
Exit command	3	All	Exits the **sp_ctrl** command-line session and disconnects the **TCP/IP** link. (Used interchangeably with **quit**.)
Host command	3	All	sets a default machine for the current **sp_ctrl** connection.
Port command	3	All	Sets a default **sp_cop** port number for the current **sp_ctrl** session.
Quit command	3	All	Exits the **sp_ctrl** command-line session and disconnects the **TCP/IP** link. (Used interchangeably with **exit**.)

Encryption commands

The encryption commands enable you to encrypt data that is being sent across the network.

List of encryption commands

Command	Auth. level	Supported targets	Description
Create encryption key	2	All	Creates an encryption key to encrypt data across the network.
Reset encryption key	2	All	Removes an encryption key.
Set encryption key	2	All	Sets an encryption key for an Export-Import pair.

Information commands

The information commands provide information about the replication environment. Use these commands when you are trying to resolve a replication problem or you want to view certain process conditions.

List of information commands

Command	Auth. level	Description
Append status	3	Displays status and results of the **append using** and **append** commands.
Copy status	3	Displays status and results of the **copy using** and **copy** commands.
Compare status	3	Displays the status and results of the **compare using** and **compare** commands.
Istatus command	3	Displays detailed information about the state of SharePlex replication.
Job status command	3	Displays current status and history for **append**, **compare**, **copy** and **repair** commands.
Orainfo command	3	Displays the Oracle database information.
Qstatus command	3	Displays the state of the capture, export and post queues.
Repair status	2	Disaplys the status and results of the **repair** and **repair using** commands.
Report	3	Displays **append**, **compare**, **copy** and/or **repair** history for a table.
Show	3	Displays the source and destination of the data being processed by each replication

Command	Auth. level	Description
command		process on a system, and displays the status of each process.
Use of Show Capture	3	Displays brief or detailed statistics for the Capture process for use in tuning and problem solving.
Show Config	3	Displays properties of the active configuration.
Show Export	3	Displays the number of messages sent to the target system(s).
Show Import	3	Displays the number of messages received from the source system(s).
Show Log	3	Displays the Even Log, Command Log, Verify Log, Trace Log, or a process log.
Show Post	3	Displays brief or detailed statistics for the Post process for use in tuning and problem solving.
Show Read	3	Displays brief or detailed statistics for the Read process for use in tuning and problem solving.
Show SQL	3	Displays the current or last SQL statement processed by the Post process.
Show Statusdb	3	Displays the Status Database, which contains records of important replication events.
Show Sync	3	Displays information about out-of-sync conditions.
Status	3	Displays an overview of the state of SharePlex replication.

Maintenance commands

The maintenance commands control command displays and files.

List of maintenance commands

Command	Auth. level	Supported targets	Description
Clear history	2	Oracle	Removes the job information and source logs for runs of the **compare**, **compare using**, **repair**, **copy** and **append** commands.
Clear status	2	All	Removes warning messages from the Status Database.
Remove log	3	Oracle	Removes old compare reports and logs from the system.
Set log	3	All	Sets the default values for the **show log** command.
Truncate Log	1	All	Clears the records from the Event Log, with the exception of logged commands.

Parameter commands

The parameter commands control the environment parameters that define how SharePlex operates.

List of parameter commands

Command	Auth. level	Supported targets	Description
List param	3	All	Displays a list of SharePlex parameters withcurrent and default values and set-at points.
Reset param	1	All	Restores the value of a SharePlex parameter to the default setting.
Set param	1	All	Changes the value of a SharePlex parameter on a global basis or for a specified replication process.

Partitioning commands

The partitioning commands create, manage, and display the partition schemes and row partitions that are being used for horizontally partitioned replication.

List of partitioning commands

Command	Auth. level	Supported targets	Description
Add partition	2	All	Creates partition schemes and row partitions.
Drop partition	2	All	Removes a row partition from a partition scheme.
Drop partition scheme	2	All	Removes a partition scheme.
Modify partition	2	All	Modifies a row partition of a partition scheme.
View Partitions	3	All	Shows existing partition schemes and their row partition specifications.

Properties commands

The properties commands display information about a user's authorization level, the local system, and the local SharePlex installation.

List of properties commands

Command	Auth. level	Supported targets	Description
Authlevel	3	All	Displays the SharePlex authorization level of a user who logs into **sp_ctl**.

Command	Auth. level	Supported targets	Description
Hostinfo command	3	All	Displays a machine's name and operating system.
Version	3	All	Displays the version of SharePlex installed on a machine.

Route commands

The route commands enable you to delete a queue or process left over from a deprecated replication route.

Command	Auth. level	Supported targets	Description
Delete capture queue	2	All	Deletes a capture queue.
Delete export process	2	All	Deletes an export process.
Delete export queue	2	All	Deletes an export queue.
Delete post queue	2	All	Deletes a post queue.

Run commands

The run commands control the operation of SharePlex with options for starting and stopping processes, starting and shutting down replication, re-synchronizing data, and enabling/disabling confirmation messages.

List of run commands

Command	Auth. level	Supported targets	Description
Abort service	2 or 1	All	Stops one or all replication processes immediately, without finishing processing. (Level 1 required for Capture.)
Clear status	3	All	Removes warning messages from the Status Database.
Shutdown	1	All	Shuts down **sp_cop** and all replication processes.
Start	2 or 1	All	Starts a replication process. (Level 1 required for Capture.)
Startup	1	All	Use the **startup** command to start all processes at once.
Stop	2 or 1	All	Stops a replication process gracefully after it is finished reading from or writing to its queues. (Level 1 required for Capture.)

Synchronization commands

The synchronization commands control aspects of data synchronization.

List of synchronization commands

Command	Auth. level	Supported targets	Description
Cancel	2	Oracle	Cancels a running **compare**, **repair**, **copy** or **append** command job.
compare / compare using	2	Oracle	Compares the contents of source and target tables to verify synchronization.
Copy / append	2	Oracle	Uses the Oracle Export/Import utility to allow users to set-up replication and/or (re)synchronize tables already in replication.
Disable jobs / enable jobs	2	Oracle	Control whether the synchronization commands are allowed to run on a secondary system in a high-availability or peer-to-peer configuration.
Flush command	2	All	Puts a marker in the data stream to establish a point of synchronization between the source and target data, then stops the Post process at that point.
Reconcile command	1	All	Synchronizes the results of a backup or copy applied to the target system with user transactions from the source system.
repair / repair using	2	Oracle	Repairs out-of-sync rows in target tables to bring source and targets back into synchronization.

Target configuration command

The target configuration commands configure the Post process to support heterogeneous replication.

Target command

Command	Auth. level	Supported targets	Description
Target	2	All	The properties stored in the target configuration control how Post processes, formats, and outputs replicated data.
Typemap	2	All	Shows how source data types are mapped to the target database.

Miscellaneous commands

These miscellaneous commands provide additional controls for using SharePlex.

List of miscellaneous commands

Command	Auth. level	Supported targets	Description
Target	2	All	The properties stored in the target configuration control how Post processes, formats, and outputs replicated data.
Typemap	2	All	Shows how source data types are mapped to the target database.
Command	Auth. level	Supported targets	Description
Target	2	All	The properties stored in the target configuration control how Post processes, formats, and outputs replicated data.
Typemap	2	All	Shows how source data types are mapped to the target database.
Command	Auth. level	Supported targets	Description
Target	2	All	The properties stored in the target configuration control how Post processes, formats, and outputs replicated data.

Alphabetical Reference for SharePlex Commands

This starts the detailed reference documentation for SharePlex commands in alphabetical order.

Abort config

Use the **abort config** command to deactivate a configuration instantly and delete all associated queues and replication processes at the same time. This command stops all replication activity for the configured datasource on the source and target systems, whether or not data in the queues has been posted. It is a forceful deactivation.

The **abort config** command is effective when there has been a system, configuration, or data problem and you need to prevent posting to the target system(s). Because you are deleting whatever data remains in the pipeline, the source and target instances will go out of synchronization after this command takes effect.

If any SharePlex replication process stops before or during the **abort config** command, the command also stops. When the process starts again, the command resumes working. This allows **abort config** to work even if the network is temporarily down — it remains in the queues until the connection is restored.

If there are multiple active configurations on the same source system, use the **abort config** command only if there are named export queues that separate the replication streams for each one. Without named export queues, SharePlex funnels all replicated data through one export queue, and an **abort config** command for one configuration deletes the data for all of them. To create named queues, see Chapter 5 of the *SharePlex Administrator's Guide*.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator
Issues for:	source system
Related commands:	**deactivate config**, **purge config**

Syntax

Basic command	**Remote options**
abort config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration that you want to abort. Configuration names are case-sensitive. Example: `sp_ctrl(sysA)>` **abort config sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Abort service

Use the **abort service** command to terminate a SharePlex replication process (service) immediately, whether or not that process has finished reading from, or writing to, its queues. This command is effective when you cannot wait for a process to stop gracefully, or in unusual circumstances when a process will not shut down when you issue the **stop** command.

This command does not affect replicated data. The data remains safely in the queues, and SharePlex resumes normal processing when the process is started again.

A process aborted with the **abort service** command remains stopped even if SharePlex is shut down and re-started. Only the **start** command can start it again.

The **abort service** command provides options with which you can abort a process for any (or all) systems, or for a named queue, without affecting replication for other routes.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2); Administrator (1) to abort Capture
Issues for:	**abort service** for Capture, Read and Export are issued for the source system. **abort service** for Import and Post are issued for the target system.
Related commands:	**start**, **stop**

Syntax

Basic command	Command options	Remote options
abort service *service*	[**to** *host*] [**from** *host*] [**for** *datasource*] [**for** *datasource-datadest*] [**queue** *queuename*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
service	Required. The SharePlex replication process you want to abort. Valid values are: capture **read** **export** **import** **post**
to *host*	This option aborts Export to a designated system, while allowing Export to other systems to continue. • host is the name of the target system. **Example**: sp_ctrl(sysA)> **abort service export to sysB** In this example, Export is aborted from sysA to sysB, but other Export processes from sysA to other target systems will continue.
from *host*	This option aborts Import from a designated system, while allowing Import from other systems to continue. • *host* is the name of the source system. **Example**: `sp_ctrl(sysD)>` **abort service import from sysC** In this example, Import from sys C is aborted on sysD, but other Import processes on sysD from other source systems will continue.
for *datasource*	This option aborts Capture or Read for a designated datasource, but other replicating datasources on the same system are unaffected. • *datasource* is expressed as o.SID, where SID is the ORACLE_SID of the source Oracle instance. **Example**: `sp_ctrl(sysA)>` **abort service read for o.oraA** In this example, Read is aborted on sysA for instance oraA, but other Read processes for other instances on sysA will continue.
for *datasource-datadest*	This option aborts Post for a designated source SID-target SID data stream, while allowing other Post activities to continue.

Component	Description
	• *datasource* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the source Oracle instance. • *datadest* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the target Oracle instance. Example: `sp_ctrl(sysC)>` **abort service post for o.oraA-o.oraC** In this example, Post is aborted on sysC for instance oraA replicating to oraC, but other Post processes on sysC will continue.
queue *queuename*	This option aborts the Export, Import or Post process associated with a named queue, while allowing replication through other named queues to continue. • *queuename* is the name of the queue. **Example:** `sp_ctrl(sysA)>` **abort service export queue QA** In this example, Export of data through named export queue QA is aborted on sysA, but Export will continue for data assigned to all other named queues.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**

Option	Description
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Activate config

Use the **activate config** command to activate a configuration. Replication begins immediately as soon as the activation process is complete. For more information about what happens when you activate a configuration, see the SharePlex Administration Guide.

The activation process reads the configuration file, from which it gets all of the information needed for SharePlex to:

- Identify the objects that are in replication
- Route the replicated changes to the appropriate target database
- Generate the SQL that Post uses to apply the changes to the target
- Activate all of the tables that have been added to replication

The process that **sp_cop** calls to activate a configuration is **sp_tconf**.

Activation creates asynchronous, parallel processing threads to activate multiple tables simultaneously. Each table is locked for a very short time, just long enough to activate the table. Replication of each table begins as soon as its activation is complete.

Should one or more table fail to activate, SharePlex continues with the activation of the other tables. If an application uses NOWAIT locking on tables in the replication configuration, it could fail if it attempts to obtain a lock on an object being activated.

Guidelines for activation

- To activate a configuration, the database containing the objects to be replicated must be mounted and open. The length of time that activation takes varies, depending on the size, number and structure of the configured objects.
- You can activate one configuration *per datasource* (Oracle instance) on each system. For example, if there are ConfigA, ConfigB and ConfigC for instance ora10, you can activate only one of them at a time. Activating another configuration for the same datasource automatically deactivates the first one.
- Do not perform DDL, including TRUNCATE, during activation. DML changes are the only permissible changes during activation.
- Activation requires that the applications have retry logic. NOWAIT locking on tables in the replication configuration may cause the application to fail if it attempts to obtain a lock on an object that is being activated.
- The activation process retains control of the **sp_ctrl** interface until the activation is finished. To activate multiple configurations for different datasources on the same system, activate the first one, then open another session of **sp_ctrl** to activate the second one. Open as many sessions of **sp_ctrl** as you have configurations to activate.
- Before you activate a configuration, use the **verify config** command to confirm that basic requirements for successful activation and replication have been satisfied. The command alerts you to potential problems that can cause the activation to fail.

Set the Number of Activation Threads

You can set the number of activation threads globally (for all activations) and you can override this setting for any activation.

To set the number of threads globally:

1. Run **sp_ctrl**
2. Issue the following command. You may use a value of up to 32 threads.

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

To set the number of threads for the current activation:

Use the [**threads**=*n*] option when you issue the **activate config** command.

View activation status and results

SharePlex activates objects according to their object ID, not their order in the configuration file, so there is no way to predict the order of activation.

Because SharePlex continues with activation whether or not individual tables fail to activate, it alerts you when tables fail to activate by displaying the following error message at the **sp_ctrl** prompt: `"WARNING, not all objects activated successfully. Check activation log."`

To view the results of activation:

Issue the **show config** command

What to do if activation fails

Many things can cause the activation of a table or the entire configuration to fail. For example, if one or more components in the configuration file were entered incorrectly, activation of the affected objects fails.

If you did not issue the **verify config** command before you activated, run it now, and correct any problems that it finds. Then, try activating again. For more troubleshooting advice, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle (all options)
Supported targets:	All
Authorization level:	Administrator (1)
Issues for:	source system
Related commands:	**abort config, copy config, create config, deactivate config, edit config, list config, purge config, remove config, rename config, show config, verify config, view config**

Syntax

Basic command	Command options	Remote options
activate config*filename*	[**threads=***n*] [**nolock**] **scn=***scn_value* **seqno=***log_sequence_number*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	Required. The name of the configuration that you want to activate. Configuration names are case-sensitive. Example: `sp_ctrl(sysA)>` **activate config sales**
threads=*n*	(Valid for Oracle) Use this option to set the number of analysis threads that the activation process generates. This option overrides the default value set by the SP_OCF_THREAD_COUNT parameter. The range of valid values for *n* is 1 to 32, but it is recommended that you use no more than 5 threads because the benefits of using threads generally diminish beyond that point. SharePlex will not start more threads than the number of tables to be analyzed. When used, this option must appear after the required command arguments. Example: `sp_ctrl(sysA)>` **activate config sales threads=3**
nolock	(Valid for Oracle) Use this option to activate without locking the tables being added to replication.
scn=*scn_value*	(Valid for Oracle) Use this option to activate the configuration to start replication at a specific SCN in the redo logs. Before activating the configuration, do the following: • If there was a previously active configuration, run the **ora_cleansp** utility on the source and all targets to restore the environment to a clean state. For more information, see Ora_cleansp on page 514.

Component	Description
	• Use the **show scn** command to get the SCNs of the last transactions that were posted from all the Post processes (if using named queues). Use the lowest of those SCN values for **activate config**. Do not use this option with the **nolock** option. Example: `sp_ctrl>` **activate config myconfig scn=123456**
seqno=*log_sequence_number*	(Valid for Oracle) Use this option to activate the configuration to start replication at a specific redo log sequence number. Do not use this option with the **nolock** option. If there was a previously active configuration, run the **ora_cleansp** utility on the source and all targets to restore the environment to a clean state. For more information, see Ora_cleansp on page 514. Example: **activate config myconfig seqno=98765**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Add partition

Use the **add partition** command to add a row partition to a partition scheme when configuring horizontally partitioned replication. Issue an **add partition** command for each row partition that you want to create.

This command captures all of the information required to create the partition, including the following required components:

- The partition scheme name. To create a new partition scheme, specify the name in the **add partition** command that creates the first row partition for that scheme. SharePlex automatically creates the partition scheme. Then, specify that name when adding additional row partitions to that partition scheme.
- The hash value or the column condition specification that creates the row partition.
- The routing for the rows that are specified in the row partition.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes. For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	Modify partition, Drop partition, Drop partition scheme, View Partitions

Syntax

Basic command	Command options	Remote options
add partition to *scheme_name* **set** {**condition =** *column_condition* | **hash =** *hash_value*} **and** **route =** *routing_map*	[**and name =** *partition_name*] [**and tablename =** *target_table*] [**and description =** *description*]	Not available

Syntax description

Component	Description
to *scheme_name*	**to** is a required keyword indicating the row partition is being added to *scheme_name*. *scheme_name* is the name of the partition scheme. The partition scheme is created by the first **add partition** command that you issue, which will also specify the first set of rows to partition. If you are making heavy use of horizontal partitioning, it may help to establish naming conventions for your partition schemes.
set	Required keyword that starts the definition of the row partition.
condition = *column_condition*	Creates a row partition based on a column condition. The condition must be in quotes. Use standard WHERE conditional syntax such as **((region_id = West) and region_id is not null)**.
hash = *value*	Creates a row partition based on a hash value. The specified value determines the number of row partitions in the partition scheme. The **condition** and **hash** components are mutually exclusive.
route = *routing_map*	The route for this partition. This can be one of the following: **Partition based on a column condition:** Specify any standard SharePlex routing map, for example: **sysB@o.myora** or **sysB:q1@o.myora** or **sysB@o.myora+sysC@o.myora** (compound routing map). If the target is JMS, Kafka, or a file, then the target should be specified as x.jms, x.kafka, or x.file, for example: **sysA:hpq1@x.kafka**. To route a partition to multiple target tables that have different names, do the following: • Issue a separate **add partition** command for each different target name. Use the **tablename** option to specify the name. • In the configuration file, specify any of these target tables as the target table in the entry that uses this partition scheme. SharePlex will detect the other names when the configuration is activated. • Set the SP_ORD_FIRST_FIND parameter to 0 so that SharePlex checks all of the column conditions in the partition scheme. By default SharePlex assumes that any given row change will satisfy only one column condition in the partition scheme. **Partition based on a hash:** Use the following format to direct SharePlex to create a named post queue for each partition:

Component	Description
	host:*basename*|#{**o.***SID* | **r.***database*} where: • *host* is the name of the target system. • *basename* is the base name that is assigned to all queues. • **|#** directs SharePlex to number the queues sequentially by appending the base name with an integer, starting with 1 to the value set with **hash**. • **o.***SID* for an Oracle target or **r.***database* for an Open Target target.
name = *name*	(Recommended) A short name for this partition. This option is only useful for partitions based on column conditions. A name eliminates the need to type out long column conditions in the event that you need to modify or drop the partition in the future.
tablename = *owner.table*	(Optional) Use this option when there are multiple target tables and one or more have different names. Issue a separate **add partition** command for each name. The table name must be fully qualified. If case-sensitive, the name must be specified in quotes. **Example:** `add partition to scheme1 set name = p1 and condition = "C1 > 200" and route = sysb:p1@o.orasid and tablename = myschema.mytable`
description = *description*	(Optional) Description of this partition.

Examples

Row partitions based on column conditions

Route different sets of rows through different post queues:

```
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
```

Route different sets of rows to different target systems and different table names from the source:

```
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
```

Row partitions based on a hash value

Divide rows into four partitions, each processing through a different post queue:

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

Analyze config

Use the **analyze config** command to run an analysis of the tables in a configuration file. This command gathers information about the activity of the tables.

IMPORTANT! Do not activate the configuration before you run the analysis, and make certain there are no other active configurations when you run it. The use of this command is similar to an actual activation.

The analyze process writes out its results based upon the data gathered at the time that was specified in the command, and then the replication stream cleans itself up.

The analysis is written to a file in the **log** subdirectory of the variable-data directory. The name of the file is:

`o.`*`datasource`*`-analysis.`*`actid`*

The analyze process maintains information about the activity of each object in replication, as well as transaction information. The transaction information can be used to identify groups of tables that are interrelated in such a way that they should be replicated in the same replication stream (same set of queues and processes).

The analysis lists each group of related tables, the total number of operations per table , and the total number of operations for the group.

For example:

```
>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
```

```
Tablename              Inserts         Updates         Deletes     Rollbacks     Total
"TEST"."SS2_TEST3"     116             158             32          0             306
"TEST"."SS2_TEST1"     75              114             29          14            232
"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
```

To view the current state of analysis:

Use the **show analyze** command to view the state of the analysis:

```
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
```

To terminate the analysis before completion:

To terminate the analysis before it is complete, use the **abort config** or **deactivate config** command, or modify the SP_ANL_RUN_TIME parameter.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	source system

Related commands:	**abort config, copy config, create config, deactivate config, edit config, list config, purge config, remove config, rename config, show config, verify config, view config**

Syntax

Basic command	Command options	Remote options
analyze config *filename*	*n* {**minutes** | **hours** | **days**}	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration file that you want to analyze. Configuration names are case-sensitive. Example: `sp_ctrl(sysA)>`**analyze config sales**
n {**minutes** | **hours** | **days**)	The number of minutes, hours, or days worth of activity to analyze.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax.

Option	Description
	Example:`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Example

```
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

Use the **append status** command to view the status of the last **copy** or **append** command job run. The **append status** command can be used to view detailed status on a copy or append job or a portion of a copy or append job, or to view status on all copy and append jobs for which SharePlex has history.

For details about using the **append status** command, refer to the below example:

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Viewer
Issued for:	source or target
Related commands:	copystatus

Syntax

Basic command	Command options	Remote options
append status	[**job_id**] [**Job_id.table_id**] [**all**] [**full**] [**detail**] [**status**]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**: `sp_ctrl(sysA)` **>append status 28282**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example**: `sp_ctrl(sysA)` **>append status 2828.HR.SRC_TEST3**
all	Displays a summary line for every job with history in the database. **Example**: `sp_ctrl(sysA)` **>append status all**

Component	Description
full	Displays the status of every object in the job. By default, the job status command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)`**>append status 2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)`**>append status detail**
[status]	Displays status history for previous jobs with the specified status. **Example**: `sp_ctrl(sysA)`**>append status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Authlevel

Use the **authlevel** command to determine your authorization level for issuing SharePlex commands on a system.

The following is an example of the display:

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

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	none

Syntax

Basic command	Remote options
authlevel	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**

Option	Description
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Cancel

Use the **cancel** command to cancel a running **compare**, **repair**, **copy** or **append** command job.

To cancel a job, you must supply its job ID. The job ID is reported back from the compare, repair, copy or append job when you issue the command that starts it:

```
sp_ctrl> repair using 1elliot
repairing 7 of 7 objects
repair started; job id 408
```

Alternatively, you can get the job ID of the most recently streamed job by issuing the **job status** command:

```
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
```

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**compare**, **repair**, **copy**, **append**

Syntax

Basic command	Remote options
cancel *job_id*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
jobID	The ID of the job to be canceled. **Example**: `sp_ctrl>`**cancel 407**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Clear history

Use the clean history command to remove the information and logs from old compare, repair, copy, and append command jobs.

These jobs generate log files on the source and target systems. By default, the job information and log files are cleaned when the job is older than the value set with the SP_SYS_JOB_HISTORY_RETENTION parameter. The **clear history** command can be used to clear the job information and logs on demand for a specific job or table, or for all jobs that are of a specific age.

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**compare**, **repair**, **copy**, **append**

Syntax

Basic command	Command options	Remote options
clear history { **all** | *source_owner.source_table* | **age days** | *jobID* }	[**for o.***source_sid*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
all	Causes all job to be removed. **Example**: `sp_ctrl(sysA)` **> clear history all**
source_owner.source_table	Causes history for a particular table to be removed. **Example**:

Component	Description
	`sp_ctrl(sysA)` **> clear history clear history user2.employee**
age days	Causes the job history older than the specified number of days to be removed. **Example**: `sp_ctrl(sysA)` **> clear history age 10**
jobID	Causes the history for the specified job id (obtained using the job status command) to be removed.
for o.*source_sid*	Optional. Can be used to employ the **clear history** command when there is no active configuration, or if there are more than one active configurations. In either case, the source SID must be specified using the for option. **Example**: `sp_ctrl(sysA)` **>clear history all for o.source_sid1**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Clear status

Use the **clear status** command to remove old warning messages from the Status Database.

To use this command:

1. Issue the **show statusdb detail** command to find out which messages can be cleared. Clearable messages have a `Yes` in the **Clear** column.
2. If you don't want to clear all messages, make a note of the status ID of each one that you want to clear.
3. Issue the **clear status** command for each status ID, or use the **all** argument to remove all clearable messages at once.

SharePlex puts a message in the Event Log identifying the messages that were cleared.

This command clears messages from the Status Database for the default system. To clear messages from a Status Database on a different system, use the [**on** *host*] option.

NOTE: Some messages cannot be cleared.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issued for:	source or target system
Related commands:	**show statusdb**

Syntax

Basic command	**Remote options**
clear status {*statusID*| **all**}	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
statusID	The SharePlex-assigned ID of an individual message (obtained using the show statusdb command) that you want to remove. **Example**: `sp_ctrl(sysA)`**>clear status 20**
all	This argument removes all clearable messages. **Example**: `sp_ctrl(sysA)`**>clear status all**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Compare status

Use the **compare status** command to view the status of the last compare or repair job run. The **compare status** command can be used to view detailed status on a compare or repair job or a portion of a compare or repair job, or to view status on all compare and repair jobs for which SharePlex has history.

NOTE: For details and examples about using the **compare status** command, see the **job status** command.

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Viewer (3)
Issues on:	source system
Related commands:	compare / compare using, repair / repair using

Syntax

Basic command	Command options	Remote options
job status	[**job_id**] [**Job_id.table_id**] [**all**] [**full**] [**detail**] [**status**]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
job status	Shows status of all compare and repair jobs for which SharePlex has history.
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**:

Component	Description
	`sp_ctrl(sysA)` **>job status 2828.2**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example**: `sp_ctrl(sysA)` **>job status 2828.HR.SRC_TEST3**
all	Displays a summary line for every job with history in the database. **Example**: `sp_ctrl(sysA)` **>job status all**
full	Displays the status of every object in the job. By default, the job status command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)` **>job status 2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)` **>job status detail**
status	Displays status history for previous jobs with the specified status. **Example**: `sp_ctrl(sysA)` **>job status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>` **status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the

Option	Description
	command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Connection

Use the **connection** command to configure SharePlex to connect to a database.

How to use the connection command

Use the connection command to establish connection properties for SharePlex to use when connecting to a database. Use this command only if there is no database setup utility available for the database. To determine whether a setup utility exists for the database, see Database Setup Utilities on page 553 .

Use the connection command in the following ways:

- With the **set** option to set connection attributes. Issue one **connection** command per option used. The command does not support the stringing of multiple keyword/value pairs with one command.
- With the **reset** option to clear connection settings.
- With the **show** option to view the current connection settings.

NOTES:

- The SharePlex-supported Open Target targets are listed in the SharePlex Release Notes.
- When a DSN exists for an ODBC database, the **dsn**, **user** and **password** keywords are the only required connection settings. If a DSN does not exist, the **user**, **password**, **server**, **driver**, and **port** keywords are all required.
- The **connect_string** keyword supports a user-defined ODBC connection string where preferred or in cases where the supplied ODBC keywords are not sufficient.

IMPORTANT: Make certain to stop and then restart the Post process after using this command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source or target system
Related commands:	**target**

Syntax

Basic command	***Keyword=value* options**	**Remote options**
connection {**o.** | **r.**}*database*	[**user**=*username*]	Not available

Basic command	*Keyword=value* options	Remote options

{
set *keyword=value* |
show [**all**] |
reset [*keyword*]
}
	[**password**=*password*]
	[**tns_alias**=*alias*]
	[**asm_sid**=*SID*]
	[**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]	

Syntax description

Component	Description
{**o.** | **r.**}*database*	The database for which you are configuring the connection. Use the following format: Oracle: **o.***SID* where: *SID* is the ORACLE_SID of the database. Open Target (ODBC): **r.***database* where: *database* is the name (not the DSN) of the database. **NOTE:** The database specification must match the database specification on the datasource line or in the routing map of the configuration file. For example, if the routing map is **myhost@r.mydb** then a **connection** command would include the same database specification, like this: **connection r.mydb set user=myuser**
set *keyword=value*	Sets a connection property. See Keywords.
show [**all**]	**show** displays the current connection settings for the specified *database*. **show all** displays connection settings for all local databases. Do not use the *database* specification with **show all**, as in this example:**connection show all**
reset [*keyword*]	**reset** clears all of the connection settings for the specified *database*. **reset** *keyword* clears the connection setting only for a specific connection property. See Keywords

Keywords

Keyword	Database type	Description of input value
user=*username*	All	The name of the database user that SharePlex will use to connect to the database specified with *database*.
password=*password*	All	The password for the user specified with *username*.
tns_alias=*alias*	Oracle	The TNS alias to which SharePlex will connect. Required if connections to the database are managed with a **tnsnames.ora** file.
asm_sid=*SID*	Oracle	The Oracle SID of the ASM instance. Required if Oracle Automatic Storage Management is being used by the database.
asm_user=*username*	Oracle	The name of the ASM user that SharePlex will use to

Keyword	Database type	Description of input value
		connect to the ASM instance.
asm_password=*password*	Oracle	The password of the ASM user.
asm_tns_alias=*alias*	Oracle	The TNS alias of the ASM instance. Required if connections to the ASM instance are managed with a **tnsnames.ora** file.
wallet_location = path	Oracle	If set, path to Oracle wallet file
wallet_automode = Y/N	Oracle	Set to Y if setup for SharePlex auto open
dsn=*DSN*	Open Target (ODBC)	The data source name (DSN) that is associated with the database. If a DSN exists for the database, this keyword plus the **user** and **password** keywords are the only required connection settings.
server=*servername*	Open Target (ODBC)	The name or IP address of the server that hosts the database.
driver=*path*	Open Target (ODBC)	The full path to the ODBC driver that supports ODBC connectivity to the database.
port=*portnumber*	Open Target (ODBC)	The port number of the database.
connect_string=*string*	Open Target (ODBC)	A user-defined connection string. When using your own connection string, make certain it includes all of the required elements to make a successful ODBC connection, *but* omit the user name and password. Use the **connection** command with the **user** and **password** options to supply user information.
plugin = pluginname		Name of plugin
plugin_version = versionnumber		Plugin version
plugin_direction = source/target/both		The plugin_direction value can be set as source, target or both.
sp_host_port = Hostname:port	Database (Non-Oracle)	SharePlex host: port that ran setup
admin_user = username		Administrator user SQL Server
database = SID		Database name
dstype = Datastorename		Datastore name
version = versionnumber		
hostaddr = IPaddress	Database (Non-Oracle)	The server IP address to which ODBC will connect
host = servername	Database (Non-Oracle)	The server name to which ODBC will connect
dbname = databasename	Database (Non-Oracle)	The database name to which ODBC will connect
db_host = IP	Database (Non-	The host of the database through DSN

Keyword	Database type	Description of input value
	Oracle)	
db_port = portnumber	Database (Non-Oracle)	The port used by the database through DSN
oracle_home = path	Oracle	The path to Oracle Home
service_name = servicename	Oracle	Oracle service name

Examples

Oracle examples

connection o.myora12 set user=myuser

connection o.myora12 set password=mypassword

connection o.myora12 set tns_alias=myora12

Open Target examples

DSN exists

connection r.mydb set user=myuser

connection r.mydb set password=mypassword

connection r.mydb set dsn=mydsn

DSN does not exist

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 does not exist, use connection string

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”

Remove a connection value

connection r.mydb reset port

connection r.mydb reset

View connection settings

connection r.mydb show

connection show all

Copy / append

The **copy/append** command uses the Oracle Export/Import utility to allow users to set up replication and/or (re)synchronize individual tables already in replication.

Users may **copy/append** individual tables to a single target, specify multiple targets, or all targets in the replication configuration.

IMPORTANT:

- This command invokes Export with CONSISTENT=Y. This requires an adequately sized RBS to avoid an ORA-01555 error.
- This command cannot be used in a bi-directional or high availability environment without additional steps. Refer to the Knowledge Base solution 48020.
- Before you use this command, review all of this documentation. **Improper use could result in lost data and out-of-sync conditions.**

The copy command

Use the **copy** command to synchronize or instantiate a table (or tables) from one host to any other host when you want the target table truncated prior to synchronizing data.

The append command

Use the **append** command to synchronize or instantiate a table (or tables) from one host to any other host when you do not want the target table truncated prior to synchronizing data.

When to use the copy/append command

Use the **copy/append** command in the following ways:

- To migrate a database.
- To sync a table that the Compare/Repair process is unable to repair.
- Place a table into replication while instantiating it.

What the copy/append command supports

The **copy/append** command supports the following:

- Objects supported by the **copy/append** command are those objects supported by Oracle's Import/Export utility.
- Copying from a lower version of Oracle to a higher version of Oracle, or between the same versions of Oracle.
- The copy/append of objects in a synchronization routing file that are included as a result of Oracle wildcard specification.
- The use of Oracle wildcard specification in command line syntax.

What the copy/append command does not support

The **copy/append** command does not support:

- Copying from a higher version of Oracle to a lower version of Oracle
- Copying from a table to a view
- UDTs or VARRAYs if the UDT or VARRAY type was not created on the target database using Oracle's Import/Export utility or Oracle's hot backup. This is a limitation in Oracle's import/export facility
- Tables in replication that are horizontally or vertically partitioned
- Sequences
- Tables for which transformation is being used, unless the transformation is being applied on the target, as well
- Network configurations in which a pass-through server is used to pass data between the source and target servers
- Column mapping
- Subset of columns
- Specifying a subset of rows within a table
- Indexes, constraints, triggers, and grants options
- Direct load
- Transparent Data Encryption (TDE)
- The **append command** does not support Oracle partitions. (Only the **copy** command supports Oracle partitions.)
- On Oracle Database 23c, the Copy/Append command isn't supported with encrypted table space.
- The BINARY DOUBLE and the BINARY FLOAT data types are not supported for the **copy/append** command.
- If a configuration file has different table names on the source and target, and the target contains another table with the same name as the source, using the **copy/append** command may compromise the consistency of the data already present in the target table. It is important to note that in such cases, the **copy/append** command will not rename the table on the target. The **copy status detail** command will display a message in sp_ctrl that "*Cannot rename table on target; source name exists*".

IMPORTANT:

- The **copy/append** command supports Oracle wildcards; however, if the set of objects indicated by the wildcard specification includes objects that are not presently in replication, the **copy/append** command will only act on those objects currently in replication.
- If the **force** option is employed and all the target information is specified, the tables in replication will be copied statically, as is the current behavior.

The launcher process

The launcher process must be running on the target system(s) prior to executing the **copy/append** command. To start the **launcher** process the user must execute the following command on each of the target systems:

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

While the launcher process is running the **show launcher** command may be used to view process details.

When the user has completed synchronization the launcher may be stopped on the target system(s). To stop the **launcher** process the user must execute the following command on each of the target systems:

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

Overview of copy/append process

The **copy/append** command initiates a process to synchronize individual tables between individual hosts.

The following is the sequence of events that illustrates simple synchronization process activity:

1. Log onto the target system and issue the **start launcher** command from **sp_ctrl**.
2. When the **copy/append** command is issued from a **sp_ctrl** session the **sp_cnc** spawns **sp_sync_svr** on the source system.
3. The **sp_sync_svr** connects to the target system and starts the **sp_sync_clnt** on the target system.
4. The **sp_sync_svr** sends the **sp_syn_clnt** a table list to verify information on the target table. a) If the object is being copied (versus an append) the target table is truncated.
5. For each object under copy the **sp_sync_svr** starts up an Oracle export process.
6. If the object under copy is in replication **sp_sync_svr** must wait for **sp_sync_clnt** to indicate that the Post process is ready.
7. The **sp_sync_clnt** process starts an Oracle import process for each of the objects that the **sp_sync_svr** process opened an Oracle export process for.
8. The **sp_sync_svr** process transfers data to the **sp_sync_clnt** process until the copy is complete.
9. If the object under copy is in replication the Post process re-initiates and resumes it's posting activities.
10. Log onto the target system and issue the stop launcher command from **sp_ctrl**.

Running concurrent copy processes

You can issue multiple **copy/append** commands at a time. Users should note that while the copy process is in progress the Post process stops all replication.

Identifying synchronization processes

The synchronization processes are not listed in the **sp_ctrl** console as they are not started by **sp_cop** process. To identify the process use your operating system to locate the executables.

- sp_sync_svr
- sp_sync_clnt
- sp_sync_lstnr
- sp_sync_launcher

Tuning parameters for the synchronization

The following is a list of parameters associated with synchronization using the **copy/ append** command. These parameters are all live and take effect the next time the command is issued.

- **SP_OSY_COMPRESSION**: This parameter adjusts the data compression from off (0) to full (9). The default integer value is set to six (6).
- **SP_OSY_EXP_BUFFER_SIZE**: This parameter adjusts the amount of data that is buffered before being sent to the target system. The default integer value is 1024 Kilobytes.
- **SP_OSY_IMP_BUFFER_SIZE**: This parameter adjusts the amount of data that is buffered before being applied on the target system. The default integer value is 1024 Kilobytes.
- **SP_OSY_LOCK_TIMEOUT**: This parameter set the number of seconds that the synchronization process will wait to obtain a table lock. The default integer value is 2 seconds.
- **SP_OSY_PORT**: This parameter sets the port number for the synchronization process. The default integer value is set to port number 2501.
- **SP_OSY_POST_TIMEOUT**: This parameter sets the number of seconds that the synchronization process will wait for the Post process to be ready and the synchronization to begin. The default integer value is set to 1800 seconds (30 minutes).
- **SP_OSY_TCP_TIMEOUT**: This parameter sets the number of seconds for the IPC time-out. The default integer value is set to 60 seconds.
- **SP_OSY_THREADS**: This parameter sets the number of synchronization processing threads. This thread parameter only affects a partitioned table. The default integer value is set to 5 threads. It has a maximum value of 32 threads.

Viewing copy status

The best way to view the status of one or more copy/append commands is to use the **copy status** command or the append status command. The basic command displays the results of all **copy/append** processes on a system.

The basic command displays the status of all **copy/append** jobs currently running on the system, and the most recently executed **copy/append** if it isn't included in the currently running list. This list can be filtered to show only a specified status.

The best way to view the results of one or more **copy/append** commands is to use the report command with the **copy** option in **sp_ctrl**.

In the example text that follows, the **copy/append** command was issued against a table not in replication. The **copy status** that follows displays basic information about the table and the status of the procedure.

```
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
```

To filter this list to only show the tables with a status of error, execute the following command:

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

Using a synchronization routing file

The synchronization routing file is very similar in structure to a configuration file used for activation, in that the file contents have the same formatting requirements. The difference between the two files is that some tables or objects in a configuration file may not be supported by the synchronization process and therefore should not be included in a synchronization routing file. For example, horizontally or vertically partitioned tables are not supported by the synchronization process and should not be included in a synchronization routing file.

Use of Oracle wildacrds in synchronization routing files is supported. The application of Oracle wildcards in a synchronization routing file is the same as in a config file. For information about how SharePlex supports wildcards, see the SharePlex Administration Guide.

Example routing file

```
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
```

NOTE:

- When using the **copy/append** command for a group of tables all the tables must be in replication or all the tables must not be in replication. If you have a combination of tables in replication and tables not in replication the user must create a separate synchronization routing file for each group and a separate synchronization process be started with each file.
- The **copy/append** command copies tables in random order and does not take into account referential integrity. Therefore, we recommend that users copy/append only one table at a time, or limit the **copy/append** to a very small number of tables.

Copying dissimilar source and target tables

The structure of the source and target table(s) should be identical, with the exception that the source tables may contain less columns than the target tables.

The **copy/append** command should not be used for source tables that contain *more* columns than their corresponding target tables.

Using copy with partitioned replication

When tables in a configuration use partitioned replication:

- The **copy/append** command will support the copying of individual tables and individual partitioned tables, but not individual partitions.
- Data is never copied to an individual partition, even when the target object specified is a partition.
- Specifying a partition as a target is not supported.
- The synchronization of data, for both, **copy** and **append**, is always done on the base table and never the partition.
- The **copy/append** command does not support vertically or horizontally partitioned tables currently in replication.
- Using the option, [singlethread], a partitioned table will be copied as one object. The default behavior is to copy each partition of a partitioned table separately so that the copy can be done in parallel.

Controlling the number of processing threads

To take advantage of machines with multiple processors, you can set the number of copy/append processing threads on the source system using the SP_OSY_THREADS parameter. This thread parameter only affects a partitioned table. The default value of this parameter is set to five (5) threads, and the range of valid values is from 1 to 32 threads.

Please note that synchronization process is single threaded when the tables are in replication. Multi-threaded processing only comes into play with tables that are not in replication.

Using Oracle wildcards in command line syntax

Oracle wildcards may be employed when specifying the source_table portion of the **copy/append** command. Additionally, you may use **not** notation to exclude specific objects.

You may specify all the tables in a specific schema using the following:

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

In the above example, all objects under the scott schema will be copied, provided the objects are currently in replication.

You can also set exclusions using the following:

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

In the above example, all objects under the scott schema, except objects whose name contain “foo”, will be copied, provided the objects are currently in replication.

For more information about how SharePlex supports wildcards, see the SharePlex Administration Guide.

Other considerations for using copy/append

- When using the **copy/append** command to sync a table that is out-of-sync or when using the **copy/append** command to place a table into replication while instantiating it the Post process will pause while the table is being copied. The Post process will resume when the **copy/append** command is complete. This does not apply to a table that is being added to replication through a new route.
- The **copy/append** command can only operate on one table at a time if the tables are currently in replication.
- When using the **copy/append** command for a table not in a known replication route the user must employ the **force** option. To employ **copy/append** on objects not in replication the user must employ the **force** option.
- When using the **copy/append** command for a table in replication, all SharePlex processes (Capture, Read, Export, Import, Post) must be running when you issue the **copy/append** command, and they must remain running throughout the processes.
- Do not use the **copy/append** command in a cascading replication environment.
- Tables with foreign keys should have those keys disabled and then re-enabled.
- Table indexes, constraints, triggers, and grants must be (re)enabled after the synchronization completes, as these options are not supported.
- When objects are in replication **copy/append** does not require locks or synchronization between the source table and target table since it is assumed that the source and target table will not undergo changes while the **copy/append** command is being executed.
- When a config file is specified, which contains multiple objects, the source and target schema name can only be different if *all* of the target objects have the same schema name.
- The **copy/append** command will only be able to copy or append data between two objects on the same database if the objects reside in *different* schemas.
- Activating a new config while the **copy/append** process is running may cause the synchronization process to fail.

NOTE: The syntax for the **copy** and **append** commands are exactly the same. The examples that appear below utilize the **copy** command, but the **append** command can be invoked by simply substituting **append** for **copy**.

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	None

Syntax

Basic command	Command options	Remote options
{**copy** | **append**} *source_owner.source_table*	**[not** (*list of exceptions*)] **[to***target_owner.target_table*[*.partition*]**]** **[at** *target_host*[*@target_SID*] [**for o.***source_SID*] **[singlethread]** **[force]** **[status]**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]
{**copy** | **append**} **using** *filename*	[**for o.***source_SID*] **[force]**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax Description

Component	Description
source_owner.source_table	Specifies an individual source owner and source table name or a group of objects using wildcards. Not valid if using {**copy** | **append**} with the **using***filename* syntax. When used without the **to***target_owner.target_table* option, this syntax assumes that the synchronization target is the same as the replication target. **Examples**: `sp_ctrl(SysA)>`**copy** s_user1.sync_single_tbl `sp_ctrl(sysA)>`**append** s_user1.sync_single_tbl
filename	The name of a synchronization routing file containing the tables to be synchronized. File names are case-sensitive. Use this option when using the {**copy** | **append**} **using** command. **Example**:

Component	Description
	`sp_ctrl(SysA)>`**copy using** sync_file_2.txt `sp_ctrl(sysA)>`**append using** sync_file_2.txt
[**not** (list of exceptions)]	Specifies an exception list of tables not to be copied/appended when the table specification includes wildcards. The `exception_list` is a list of table names that should not be copied/appended. Consider three tables: copytest1, copytest2, and copytest3. In the example below, only the data from the copytest3 table is copied to the target. `sp_ctrl > copy s_user1.% not (copytest1,copytest2)`
to *target_owner*.*target_table* [.*partition*]	Specifies the target table for synchronization. This option only valid with the *source_owner*.*source_table* option. If the table is not in replication, use this option in conjunction with the **at***target_host*@ [*target_sid*] option. **Examples**: `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*]	Specifies the location of the target table for synchronization. Use in conjunction with the **to** *target_owner*.*target_table* [.*partition*] option. **Examples**: `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*	Use to specify the datasource that contains the objects to be synchronized, if there is no active configuration, or if there are more than one active configurations. **Examples**: `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	Use this option to copy a partitioned table as one object. The default behavior is to copy each partition of a partitioned table separately so that the copy can be done in parallel.

Component	Description
force	Use this option when tables intended for synchronization are *not* in replication or when the table route does not match an existing or known route. **Examples**: `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	Displays status history for previous **copy**/**append** commands. `sp_ctrl(SysA)>`**copy status** `sp_ctrl(SysA)>`**append status**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Copy cache

Use the **copy cache** command to copy the object cache from the source to the target.

If you find the following message in the SharePlex Event Log, include in the **copy cache** command the target host and target sid for the Post process, as well as the **actid** (activation ID) from the error message. This is the activation ID of the activation for which the Post process is currently posting.

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

If the **copy cache** command is issued with no options, SharePlex will identify the current activation ID on the source and copy the object cache for that activation ID from the source to all targets.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**show log**

Syntax

Basic command	Command Options	Remote options
copy cache	[**actid** *actid*] [**at** *target_host*[**@***target_sid*]]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
copy cache	If you issue the **copy cache** command with no options, SharePlex determines the current activation ID (actid) on the source and copies the object cache for that actid to all targets.
actid *actid*	Specifies the activation ID for the object cache you want to copy to the specified target or to all targets (if no target is specified).
at*target_host*[**@***target_sid*]	Specifies the target to which you want to copy the object cache for the specified actid or for the current actid (if no actid is specified).

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Copy config

Use the **copy config** command to copy a configuration file and save the copy under a different file name.

Copying the configuration allows you to:

- Save a replica for backup purposes
- Use the copy as the basis for composing a new configuration using the **edit config** command
- Change an active configuration. You can make your changes to the copy with the **edit config** command while replication under the current configuration continues, and then activate the new configuration when you are ready.

This command copies the configuration to the same source system as the original configuration. To copy a configuration file to a *different* system, you can transfer the file via tape, CD, or FTP.

To see a list of configurations on a system, use the **list config** command. To view the contents of a configuration, use the **view config** command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**activate config, create config, edit config, list config, remove config, rename config, show config, verify config view config**

Syntax

Basic command	Remote options
copy config *filename* **to***newname*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename **to** *newname*	• *filename* is the name of the configuration that you want to copy. Configuration names are case-sensitive.

Component	Description
	• **to** is a required part of the syntax. • *newname* is the name you are giving to the copy. It must be unique among configurations on the system. **Example**: `sp_ctrl(sysA)>` **copy config sales to sales2**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Copy status

Use the **copy status** command to view the status of the last **copy** or **append** job run. The **copy status** command can be used to view detailed status on a copy or append job or a portion of a copy or append job, or to view status on all copy and append jobs for which SharePlex has history. For details and examples about using the **copy status** command, see the **job status** command.

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Viewer
Issued for:	source or target
Related commands:	**append status**

Syntax

Basic command	Command options	Remote options
job status	**[job_id]** **[job_id.table_id]** **[all]** **[full]** **[detail]** **[status]**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
job status	Without options, the command shows the status of all jobs for which SharePlex has history.
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**: `sp_ctrl(sysA)>` **job status 2828.2**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example**: `sp_ctrl(sysA)>` **job status 2828.HR.SRC_TEST3**

Component	Description
all	Displays a summary line for every job with history in the database. **Example**: `sp_ctrl(sysA)>` **job status all**
full	Displays the status of every object in the job. By default, the job status command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)>` **job status2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)>` **job status detail**
[status]	Displays status history for previous jobs with the specified status. `sp_ctrl(SysA)>` **job status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Create config

Use the **create config** command to create a new configuration file. This command runs the SharePlex default text editor and opens a new file. To create a configuration, follow the instructions in Chapter 5 of the *SharePlex Administrator's Guide.*

If you attempt to give a new configuration the same name as an existing configuration, SharePlex returns this error message "`The parameter for create config must be a new filename.`"

As an alternative to creating a configuration, you can use the **copy config** command to duplicate an existing configuration and then use the **edit config** command to edit the copy.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**activate config, copy config, edit config, list config, remove config, show config, verify config, view config**

Syntax

Basic command	Remote options
create config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration that you want to create. Configuration names are case-sensitive. The name must be unique among configurations on the system. **Example**: `sp_ctrl(sysA>` **create config sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Create encryption key

Use the **create encryption key** command to create an encryption key for SharePlex to use to encrypt data across the network. SharePlex uses Advanced Encryption Standard (AES) encryption.

The **create encryption key** command returns a randomly generated, 256-bit AES key. By default, SharePlex uses 128 bits of that length to encrypt the data.

To increase the key length that SharePlex uses, set the **SP_XPT_AES_KEY_LENGTH** parameter to 192 or 256 bits. When you increase the length, the key is harder to hack but requires more CPU power.

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

Example: **set param sp_xpt_aes_key_length 256**

The following is an example key:

```
E5F5D4CBA329D2C86B5D7ABA096C18600595490129F55A1422AAB0248B28D0E4.
```

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**set encryption key**, **reset encryption key**, **show encryption key**

Syntax

Basic command	Remote options
create encryption key	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Deactivate config

Use the **deactivate config** command to gracefully terminate replication for an active configuration. This command stops all Capture activity for the configuration, posts all data currently in the queues, and removes the associated SharePlex processes and queues.

The **deactivate config** command does not remove a configuration from the system, but only stops replication of its objects until you activate it again. When you deactivate a configuration, it can cause the target data to go out of synchronization if users continue making changes to the configured source objects. Deactivating one configuration on a system does not deactivate other active configurations on the same system.

Deactivating when multiple configurations are active

When replicating from multiple databases on the same source system, use the **deactivate config** command only if you have named (separate) export queues in each configuration that separate the data streams of those instances. Normally, the data from all replicated instances accumulates in one export queue. When you issue the **deactivate config** command, it deletes all the data in the export queue (along with the queue itself), including data for configuration(s) you might not want to deactivate. See Chapter 5 of the *SharePlex Administrator's Guide* for instructions on creating a configuration using named export queues.

To save time when activating a configuration that is only slightly different than the active one, you do not need to deactivate the active one first. While it is still active, you can copy the active configuration with the **copy config** command. Change the copy with the **edit config** command, and then activate the copy. Using this method saves time because SharePlex will not re-analyze the objects that are common to both configurations. SharePlex will only analyze the new objects. (If you deactivate the first configuration before activating the copy, SharePlex will analyze all of the objects. This process takes longer.) Be sure that the first config has completed the activation process before activating the copy.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issues on:	source system
Related commands:	**abort config, list config, purge config, remove config, view config**

Syntax

Basic command	**Remote options**
deactivate config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration that you want to deactivate. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)>` **deactivate config sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Delete capture queue

Use the **delete capture queue** command to remove a capture queue that is left over from a deprecated replication route.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**delete export queue**, **delete post queue**, **delete export process**

Syntax

Basic command	Command options	Remote options
delete capture queue for *datasource*	None	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description

datasource	The datasource of the capture queue as shown in the **Queue Name** field of the **lstatus** command output:
	`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`

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Delete export process

Use the **delete export process** command to remove an Export process that is left over from a deprecated replication route.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**delete capture queue**, **delete export queue**, **delete post queue**

Syntax

Basic command	Command options	Remote options
delete export process to *host*	**queue** *queuename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
host	The name of the target system.

queue *queuename*	Use if there are multiple Export processes on the system through the use of named export queues. For *queuename*, specify the name of the export queue that is linked to the Export you want to delete.
	To view the Export processes and their queues, use the **show export** command, as shown in this example:
	`sp_ctrl (sys1:8567)> show export`
	`Host : sysl`
	`Queue : expquel`
	`Target Status Kbytes Exported Since Total Backlog`
	`---------- --------------- ------------ ------------------ ---------- - ---------`

Component	Description
	`sys2 Idle 245690 05-Feb-16 18:17:39 245690 0` Deleting an Export process also deletes the post queue and Post process that are associated with it on the target system.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Delete export queue

Use the **delete export queue** command to remove an export queue that is left over from a deprecated replication route.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**delete capture queue**, **delete post queue**, **delete export process**

Syntax

Basic command	Command options	Remote options
delete export queue *queuename*	None	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description

queuename	The name of the export queue, as shown in the **Queue Name** field of the **lstatus** command:
	`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`

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Delete post queue

Use the **delete post queue** command to remove a post queue that is left over from a deprecated replication route.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**delete capture queue**, **delete export queue**, **delete export process**

Syntax

Basic command	Command options	Remote options
delete post queue *queuename* **for** *datasource-datadest*	**cleartrans**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description

queuename	The name of the post queue, as shown in the **Queue Name** field of the **lstatus** command output:
	`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*	The route specification, as shown in the **DataSrc-DataDst** field of the **lstatus** command output: `sp_ctrl (mysys)> lstatus on sys103`

Component	Description
	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: sysl03 **DataSrc-DataDst: o.ora112-o.ora112**
cleartrans	Delete the specified post queue and remove the related rows from the SHAREPLEX_TRANS table.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Disable jobs / enable jobs

Use the **disable jobs** and **enable jobs** commands to control whether jobs that are performed by the following synchronization commands are allowed to run:

- **copy and copy using**
- **append and append using**
- **compare and compare using**
- **repair and repair using**

The **disable jobs** command is a safeguard that ensures that the actions performed by those jobs do not get issued in a peer-to-peer or high-availability configuration. The command prevents the primary (trusted source) data from being overwritten with un-trusted data if a **copy**, **append**, **repair**, or **repair using** command is issued on the secondary system.

Use the **enable jobs** command to enable jobs on the primary (trusted) system when it becomes the active system again.

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**copy**, **append**, **compare**, **compare using**, **repair**, **repair using**

Syntax

Basic command	Command options	Remote options
disable jobs	**for o.***source_SID*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
for o.*source_SID*	Use this option if there is no active configuration or if there are more than one active configurations against different Oracle source instances. **Examples**: `sp_ctrl(SysA)>`**disable jobs for o.ora1** `sp_ctrl(SysA)>`**enable jobs for o.ora1**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Drop partition

Use the **drop partition** command to remove a row partition from a partition scheme in a horizontally partitioned replication configuration. Issue a **drop partition** command for each row partition that you want to remove.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	Add partition, Modify partition, Drop partition scheme, View Partitions

Syntax

Basic command	Remote options

drop partition from *scheme_name*
where
{
{**condition =** *column_condition* | **hash =** *hash_value* }|
route = *routing_map* |
name = *partition_name* |
tablename = *target_table* |
description = *description*
}
	Not available

Syntax description

Syntax Component	Description
scheme_name	The name of the partition scheme.
condition	Column condition that defines the rows of the partition that you want to remove. This option and the **hash** option are mutually exclusive.
hash	Hash value of the partition that you want to remove. This option and the **condition** option are mutually exclusive.
route	The routing map of the partition that you want to remove.
tablename	The fully qualified name of the target table in the partition that you want to remove.
name	The short name of the partition that you want to remove.
description	The description of the partition that you want to remove.

Examples:

```
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

Use the **drop partition scheme** command to remove a partition scheme from the SharePlex replication configuration. This command also removes all of the row partitions that are specified for the partition scheme.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	Add partition, Modify partition, Drop partition, View Partitions

Syntax

Basic command	Remote options
drop partition scheme *scheme_name*	Not available

Syntax description

Syntax Component	Description
scheme_name	The name of the partition scheme.

Example

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

Edit command

Use the **edit** command in the following ways:

- To make changes to a SharePlex command that you previously issued. This saves time when you need to issue a command again, but with slightly different syntax, or if you need to correct a syntax mistake.
- To execute commands that contain more than 255 characters, a limitation of the operating system.

The **edit** command opens a temporary text file in the default text editor from within the **sp_ctrl** interface. You enter the command in the file, and it automatically executes when you save the file.

To use the edit command:

1. If no commands were issued previously in the current session of **sp_ctrl**, type the first word of the command at the **sp_ctrl** prompt, then press **Enter**. Otherwise, skip this step. (Ignore the online help or error that is displayed.)
2. At the **sp_ctrl** prompt, issue the edit command to open a temporary text file.

   ```
   sp_ctrl(sysA)> edit
   ```
3. In the text file, use the editor's standard commands to edit out any unwanted characters and type the command string.
4. Save and exit out of the file using the editor's standard save command. The command executes immediately

NOTE: SharePlex provides keyboard shortcuts for the edit command in the for of the up arrow (⇞) and the forward slash (/) keys.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	The user must have the previous command's authorization level or higher.
Issued for:	The system affected by the previous command
Related commands:	none

Syntax

Basic command	Remote options
edit | ed	not available

Edit config

Use the **edit config** command to modify an inactive configuration file. This command opens the configuration in SharePlex's default text editor. Modify and save the configuration using the tools provided by the editor.

To modify an active configuration, it is recommended that you copy the active configuration with the **copy config** command, and then edit the copy with the **edit config** command. When you activate the copy, the original configuration deactivates.

To view a list of configurations on a system, use the **list config** command. To view the contents of a configuration, use the **view config** command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**activate config, copy config, create config, list config, remove config, rename config, verify config, view config**

Syntax

Basic command	**Remote options**
edit config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	**Description**
filename	The name of the configuration that you want to edit. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)>` **edit config sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Exit command

Use the **exit** command to close the current session of **sp_ctrl**. Closing **sp_ctrl** does not shut down replication; all replication processes continue without interruption unless they have been stopped by a user. This command merely discontinues your session with **sp_ctrl** on that system. To run sp_ctrl again, change to the directory containing the SharePlex binaries and enter the **./sp_ctrl** command (for Unix and Linux systems) or run **sp_ctrl** in administrator mode(Windows systems).

There are no [**on** *host*] options for the exit command. It must be issued on the system where you want to stop running **sp_ctrl**.

This command is the same as the **quit** command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**quit**

Syntax

Basic command	**Remote options**
exit | e	not available

Expand command

Use the **expand** command to expand a wildcard specification and display it to the screen. This is particularly useful if you are not sure whether a wildcard specification will produce the list of tables that you want to replicate. For more information on how to use wildcards in the configuration file, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**create config, edit config, show config, verify config**

Syntax

Basic command	Command options	Remote options
expand *filespec*	[**not** (*list of exclusions*)] [**for o.***source_sid*] [**on** *host*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filespec	The wildcard specification (including the owner) that the command will verify. Without the **for o.***source_sid* option, it is assumed that there is an active configuration file that contains the wildcarded objects. **Example**: `sp_ctrl(sysA)>` **expand scott.%** In this example, the result will display all objects in the **scott** schema.
not (*list of exclusions*)	Exclusions to the wildcard specification. Listed objects are excluded from the expansion. **Example**:

Component	Description
	`sp_ctrl(sysA)>` **expand scott.% for o.oraA not (%"sal"%)** In this example, the result displays all objects in the **scott** schema from the oraA instance, except for any objects with names that contain “sal.” Please note that in this example, "sal" is enclosed within quotation marks because it is case sensitive.
for o.*source_sid*	Use when there is no active configuration file to specify the Oracle instance against which the command will be executed. Example: `sp_ctrl(sysA)>` **expand scott**.% **for o.oraA** In this example, the result will display all objects in the “scott” schema from the oraA instance.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Flush command

Use the **flush** command to stop the Post process at a certain point. It puts a marker in the data stream and automatically stops the Post process at the marker point. You can issue this command at a certain time or date – when month-end reports need to be generated, for example – and the data on the target system will reflect what was on the source system at the time the command was issued.

When you are performing backups, or when following certain documented SharePlex procedures, you issue the **flush** command after you stop user access on the source system to ensure that the last transaction gets posted to the target database before Post is stopped.

Starting Post again

Changes generated after the **flush** command accumulate in the queues and are applied to the target instance after you start Post again. Post remains stopped until the **start post** command is issued. Post keeps a record of where it stopped and resumes posting from that point to maintain synchronization.

IMPORTANT: If users continue making changes on the source system while Post is stopped, data will accumulate in the post queue and possibly consume all available disk space. Remember to start Post as soon as permissible.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**stop**

Syntax

Basic command	Command options	Remote options
flush *datasource*	[**to** *host*] [**queue** *queuename*] [**to** *host* **queue** *queuename*] [**to** *host@target_SID*] [**to** *host@target_SID* **queue** *queuename*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
datasource	This argument specifies the source data that the command will affect. Without additional options, it flushes the data through all queues on all target systems and stops Post. *datasource* is expressed as **o**.*SID*, where SID is the ORACLE_SID of the *source* Oracle instance. **Example**: `sp_ctrl(sysA)>` **flush o.oraA** In this example, the data for instance oraA is flushed to all target systems.
to *host*	This option flushes all of the data replicating to a designated target system. Replication to other target systems is unaffected. **Example**: `sp_ctrl(sysA)>` **flush o.oraA to sysB** In this example, the data is flushed from sysA to sysB.
queue *queuename*	This option flushes data for a designated post queue. It flushes data flowing through all queues of that name on all target systems. This option is most useful when there are named post queues. **Example**: `sp_ctrl(sysA)>` **flush o.oraA queue sales** In this example, the data is flushed for post queue *sales* on all target systems that have a post queue of that name.
to *host* **queue** *queuename*	This option flushes data for a designated post queue on a designated target system. Other post queues on that and other target systems are unaffected. **Example**: `sp_ctrl(sysA)>` **flush o.oraA to sysB queue sales** In this example, the data is flushed for post queue *sales* on sysB.
to *host@datadest*	This option flushes data to a designated target instance on a designated target system. It does not affect: • Other target instances on that system. • Other target instances with the same ORACLE_SID on other target systems In the syntax: • *host* is the target system's name.

Component	Description
	• *datadest* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the target instance. **Example**: `sp_ctrl(sysA)>` **flush o.oraA to sysB@o.oraB** In this example, the data is flushed to instance oraB on sysB.
to *host@datadest* **queue** *queuename*	This option flushes data for a designated post queue and target instance on a designated target system. It does not affect: • Other post queues for that instance or any other target instance on that system. • Other post queues on any other target system. In the syntax: • *host* is the target system's name. • *datadest* is expressed as o.SID, where SID is the ORACLE_SID of the target instance. • *queuename* is the name of the post queue. **Example**: `sp_ctrl(sysA)>` **flush o.oraA to sysB@o.oraB queue sales** In this example, the data is flushed for post queue *sales* for target instance oraB on sysB.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last

Option	Description
	component of the command syntax. **Example:** `sp_ctrl(sysB)>` **status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>` **status on john/spot5489@SysA:8304**

Help command

Use the **help** command to get help for SharePlex commands while working within **sp_ctrl**.

- Using this command without any options produces a list of SharePlex command groups.
- Additional options enable you to drill down through each group to get help for a specific command.

There are no **[on** *host*] options for this command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**usage**

Syntax

Basic command	Command options	Remote options
help	{*group_name*} {*command_name*}	Not available

Syntax description

Component	Description
group_name	Displays the commands within each group for which specific help can be obtained. Valid values are: **config** **connect** **info** **maint** **misc** **param** **properties** **queues**

Component	Description

	run
	sync
	For a list of commands within each group, see “SharePlex commands at a glance.”
	Example:
	`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	This option provides help for a designated command. **Example**: `sp_ctrl(sysA)>` **help start** `Restart a SharePlex process that was stopped with a "stop" or "abort service" command.`

Host command

Use the **host** command to establish a default system for **sp_ctrl**. A default machine enables you to enter a series of commands without using the [**on** *host*] option for each one. The TCP/IP connection to the default system remains active until you exit **sp_ctrl** or issue another host command.

The default system name is displayed as part of the sp_ctrl prompt as shown below:

```
sp_ctrl(hostname)>.
```

The default system can be the machine where you are running **sp_ctrl** or any other system to which that machine can connect through TCP/IP.

To issue a command for a system other than the default, use the command's [on host] option to temporarily connect to the alternate system. When the command finishes, it returns you to the default **sp_ctrl** connection.

The **host** command can only be issued from within **sp_ctrl** at the **sp_ctrl** prompt.

TIP: To set a default port number for an sp_ctrl session, use the port command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**port**

Syntax

Basic command	Remote options
host *hostname*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
hostname	The name of the machine that you want to establish as the default. This name must reside in your system's /etc/hosts file (Unix and Linux) or

Component	Description
	hosts file (Windows). **Example**: `sp_ctrl(sysA)>` **host sysB**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Hostinfo command

Use the **hostinfo** command to view information about a system's hardware, including the operating system, machine name, OS version and revision.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	none

Syntax

Basic command	Remote options
hostinfo	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**

Option	Description
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Job status command

Use the **job status** command to view the status of the last **compare**, **compare using**, **repair**, **repair using**, **copy** or **append** job, and for any other jobs that are still running. The **job status** command can be used to view detailed status on a job or a portion of a job, or to view status on all jobs for which SharePlex has history. SharePlex retains the history for all jobs as specified by the SP_SYS_JOB_HISTORY_RETENTION parameter. See the **clear history** and **remove log** commands for information on actively removing history and/or job process logs.

```
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
```

To view a summary of all jobs for which SharePlex has history:

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


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

To view the status of a job with detail:

```
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
```

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**copy status, append status, compare status, repair status**

Syntax

Basic command	**Command options**	**Remote options**
job status	[*job_id*]	[**on** *host* |

Basic command	Command options	Remote options
	[*job_id.table_id*] [**all**] [**full**] [**detail**] [**status**]	**on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**: `sp_ctrl(sysA)>` **job status 2828.2**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example: `sp_ctrl(sysA)>` job status 2828.HR.SRC_TEST3**
all	Displays a summary line for every job with history in the database. **Example**: `sp_ctrl(sysA)>` **job status all**
full	Displays the status of every object in the job. By default, the job status command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)>` **job status2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)>` **job status detail**
status	Displays status history for previous jobs with the specified status. `sp_ctrl(SysA)>` **job status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

List config

Use the **list config** command to view a list of all active and inactive configurations on a source system.

The command displays the following information:

- **File Name**: The name of the configuration, the time and date that it was modified, and the size of the file.
- **State**: Whether the configuration is active or inactive. *Active* means the configuration file is currently involved in replication. *Inactive* means the configuration is not currently involved in replication.
- **Datasource**: The Oracle instance containing the objects being replicated by the configuration.
- **Internal Name**: The name of the internal copy of the active configuration. This is the file that SharePlex actually replicates from. Its name is displayed under the **State** heading wherever there is an active configuration.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**view config, show config**

Syntax

Basic command	Remote options
list config	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

List param

Use the **list param** command to display SharePlex tuning parameters. Without options, the command displays the basic (user configurable) parameters and their current settings.

Options are available for displaying:

- All SharePlex parameters, including those that should not be changed by a SharePlex user (internal parameters).
- Only the basic and internal parameters whose settings have been changed from their default values. (These settings are recorded in the SharePlex variable-data directory.)
- Either of the preceding options filtered according to the SharePlex module.

Parameters are grouped by *module*, each module representing a functional component of the software. The following table lists the modules that contain user-configurable parameters (which can be changed without guidance from Quest).

SharePlex parameter modules

Module	Naming convention	Function controlled
analyze	SP_ANL	**analyze config** command
cap	SP_CAP	Non-Oracle Capture
capture	SP_OCT or SP_CAP	Capture process
compare	SP_DEQ or SP_CMP	**compare** commands
config	SP_OCF	configuration activation process
cop	SP_COP	**sp_cop**
copy	SP_OSY or SP_CPY	the copy/ append commands
export	SP_XPT	Export process
import	SP_IMP	Import process
logging	SP_SLG	the SNMP feature
post	SP_OPO or SP_OPX	Post process
queue	SP_QUE	the SharePlex queues
read	SP_ORD	Read process
SNMP	SP_SNMP	SNMP support
system	SP_SYS	system-related items

Each **list param** display provides the parameter's:

- **Parameter Name**
- **Actual Value**: the parameter's current setting and its default value, if different.

NOTE: Only up to 36 characters are displayed for this field.

- **Units**: the parameter's unit of measurement, such as seconds, kilobytes, or operations.
- **Set-At point**: the point where the parameter becomes active when you issue the set param command, either immediately after the parameter is set, after the affected process is restarted, or after **sp_cop** is restarted.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**set param, reset param**

Syntax

Basic command	Command options	Remote options
list param	[**basic** | **all** | **modified**] [*module*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
basic	This option displays parameters that can be set by users without guidance from a Quest support engineer or technical specialist. **Example**: `sp_ctrl(sysA)>` **list param basic**
all	This option displays all of the SharePlex parameters, including basic (user-configurable) parameters and **internal** parameters (whose settings you should not change without guidance from Quest). **Example**: `sp_ctrl(sysA)>` **list param all**
modified	This option lists the user-configurable and internal parameters that have been modified from their default values.

Component	Description
	Example: `sp_ctrl(sysA)>` **list param modified**
module	This option constrains the output to parameters for a specific module. See SharePlex parameter modules. This option, if used, must appear after the **list param [all | modified]** syntax. **Example**: `sp_ctrl>` **list param all post** In this example, SharePlex displays all parameters for the Post process.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

lstatus command

Use the **lstatus** command to view detailed information about the status of replication on a source or target system. This command is the most comprehensive information command in **sp_ctrl**. To view a brief status of replication, use the **status** command.

The **lstatus** command displays:

- The status of the replication processes, including their associated queues.
- The operating system process ID of each process.
- When the process started.
- The machine where the process is running.
- The number of messages (SQL or SharePlex operations) in the queues and their age.
- Whether the machine is a source or target system.
- Events that took place since **sp_cop** started.
- A list of active configurations.

The **lstatus** command also provides information about the queues, including:

- **Type**: The kind of queue that it is (capture queue, export queue, post queue).
- **Queue name**: The name of the queue, either a default name or the user-assigned name of a named queue.
- **# Messages**: The number of messages in the queue. A message approximately corresponds to a SQL operation, but there can be multiple messages for one operation on a LONG or LOB column. Conversely, there could be one message for numerous operations in an array insert.
- **Size**: The current size of the queue. This varies with the amount of data.
- **Age**: The time difference between when the oldest and newest messages in the queue were written to that queue.
- **Oldest Msg Time**: The date and time the oldest message entered the queue.
- **Newest Msg Time**: The date and time the newest message entered the queue.

The **lstatus** command also provides other information, such as how the system is being used, its configuration activity, and errors that occurred.

Understanding the SharePlex queues

The following will help you understand the statistics for the SharePlex queues.

Queue size

Although SharePlex uses memory for the queues, the data is periodically written to disk as part of the checkpoint recovery system. The default size for SharePlex queue files is 8 MB. However, queue files are sparse files, meaning that from a system standpoint the file size is 8 MB, but the filesystem might only allocate part of the file for data written to the queues.

That is why the **qstatus** command can show a queue size of less than 8 MB, but what you see with **qstatus** is NOT the *true indication of the actual file size*. To predict disk space usage for the queue files, use the information from **qstatus** and not the filesystem.

Number of queues

It is normal for the capture and export queues to have fewer queue files than the post queue. Data that resides in one queue file on the source system is separated into multiple sub-queue files on the target system, each approximately corresponding to a user session on the source system.

Names of queues

Queue names are case-sensitive on all platforms. The following explains the naming conventions for SharePlex queues.

- Default capture queues are identified by their datasource, which is expressed as **o**.*SourceSID*, for example *o.ora64*.
- Default export queues are identified by the source system's name, for example *SysA*.
- Default post queues are identified by the name of the source system appended with *datasource-datadestination*, where *datasource* is **o**.*sourceSid* and *datadestination* is **o**.*targetSID*, for example *lajolla (o.ora10a-o.ora10b)*.
- Named export and post queues are identified by their user-assigned names, not the default name. For more information about named queues, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**qstatus, show, show statusdb, status**

Syntax

Basic command	Remote options
lstatus	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Modify partition

Use the **modify partition** command to modify a row partition of a partition scheme in a horizontally partitioned replication configuration.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	Add partition, Drop partition, Drop partition scheme, View Partitions

Syntax

Basic command	Remote options
modify partition in *scheme_name* **set** *keyword* **=** *value* [**and** *keyword* **=** *value*] [...] **where** *keyword* **=** *value* [**and** *keyword* **=** *value*] [...]	Not available

Syntax description

NOTE: See Add partition for additional descriptions of these options.

Syntax Component	Description
scheme_name	The name of the partition scheme. Do not modify this component, or the row partition will shift to a new partition scheme.
keyword	Any of the following syntax components except *scheme_name*.
condition	Column condition that defines a row partition. The **condition** and **hash** components are mutually exclusive.
hash	Hash count that specifies the number of row partitions that SharePlex will create based on the rowid hash-based partition scheme. The **hash** and **condition** components are mutually exclusive.
route	The routing map for this partition.
tablename	Fully qualified target table name.
name	Short name of this partition.
description	Description of this partition.

Examples

`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 command

Use the **orainfo** command to view Oracle database information including the SID. The following is an example of the display:

```
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
```

The following will be displayed if a database account was not created for SharePlex or the **statusdb** was deleted, or if the command is executed from a version of SharePlex that does not support it.

```
sp_ctrl (alvspxl11:2101)> orainfo


Oracle information not available
```

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	none

Syntax

Basic command	Remote options
orainfo	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* |

Basic command	Remote options
	on *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Port command

Use the **port** command to connect from **sp_ctrl** to an instance of **sp_cop** that is running on a TCP/IP port number other than the default. The default port number for **sp_co**p is either 2100 (the default port when SharePlex was installed) or a user-defined port that was set with the SP_COP_TPORT and SP_COP_UPORT parameters. All commands issued during the current session of **sp_ctrl** will affect the **sp_cop** running on the specified port until the **sp_ctrl** session is terminated. A new session of **sp_ctrl** connects to the default port number or the user-defined port, if one was defined for the instance of SharePlex.

When to use the port command

When you are running one instance of **sp_cop**, using the **port** command is unnecessary. On startup, **sp_ctrl** determines the port number by first checking for a user-defined port and, if none exists, defaulting to port 2100. Use the **port** command when you are running multiple instances of **sp_cop** on dedicated ports (such as in consolidated replication) and default ports for them were not set with the **SP_COP_TPORT** and **SP_COP_UPORT** parameters. To issue commands for multiple **sp_cop** instances, open a session of **sp_ctrl** for each one, and use the port command to set a port for each session. SharePlex notifies you as follows if the **port** command is required when you start **sp_ctrl**:

```
"Your tcp port is not set properly or 'sp_cop' is not running."
```

The warning indicates that **sp_cop** is not running or that it is running on a different port than the default. The **port** command cannot be issued from a remote machine, so there are no [**on** *host*] options.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**host**

Syntax

Basic command	Remote options
port *number*	Not available

Syntax description

Component	Description
number	The port number you want to use. There must be a space between **port** and the port number. **Example**: `sp_ctrl>` **port 2200**

Purge config

Use the **purge config** command to remove the data from all queues associated with a configuration without removing the queues themselves or deactivating the configuration. Avoiding a deactivation avoids the need for SharePlex to recalculate the configuration data. This saves time when the tables are large and numerous, enabling replication can start sooner.

Issue the **purge config** command on the source system to affect the source system and all target systems in the configured routes. Should any SharePlex process stop prior to or during the **purge config** activity, the command also stops working. When the process starts again, the command resumes working. Thus, **purge config** works even when the network is temporarily unavailable — the command remains in the queues until the connection is restored.

Cautions for using the purge config command:

- Do not activate a configuration and then follow the **activate config** command with a **purge config** command. You might be purging more than just queued data, including the configuration information that controls replication, thus rendering the activation invalid.
- When there are multiple active configurations on the same source system, use the **purge config** command only if there are named export queues that separate the replication streams for each one. Without named export queues, SharePlex funnels all replicated data through one export queue, and a **purge config** command for one configuration deletes the data for all of them. To create named queues, see Chapter 5 of the *SharePlex Administrator's Guide*.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	source system
Related commands:	**abort config, deactivate config**

Syntax

Basic command	Remote options
purge config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration that you want to purge. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)>` **purge config sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Qstatus command

Use the **qstatus** command to view statistics for the capture, post, and export queues on any system. It displays the number of messages in each queue, their age, and the current size of the queue. Typically, a *message* approximately corresponds to a SQL operation, but there can be multiple messages for one operation on a LONG or LOB column, and there could be just one record for numerous operations of an array insert. A message also can be an internal SharePlex operation.

When to use the qstatus command

Use the qstatus command to:

- Determine if there is data still waiting to be read by a replication process or posted to the target database.
- Estimate the speed at which SharePlex is processing by analyzing the rate at which messages accumulate.
- View the size of the queues to ensure that they do not exceed available disk space.
- Verify that all of the queues are empty when that is required for certain operational procedures or when you need to shut down replication for system maintenance, upgrades, and other administrative tasks.
- Determine if there is user activity on a target system that can cause data to go out of synchronization.

About the output

- The number of messages in a queue reflects the messages that have been read by the next SharePlex process, as well as those that have not been read. As part of its checkpoint recovery system SharePlex retains copies of messages that were sent to the next process. These messages are deleted when receipt by that process is acknowledged.
- The Backlog field indicates the number of messages yet to be read by the next SharePlex process.
- The Age field is the difference in time between when the oldest and newest messages in the queue were written to the queue.
- The presence of a Post queue on a system that also has capture and export queues indicates that this system is used both as a source system and as a target system.
- The Size field indicates the true size of a queue.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**lstatus, show**

Syntax

Basic command	Remote options
qstatus	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Quit command

Use the **quit** command to close the current session of **sp_ctrl**. Closing **sp_ctrl** does not shut down replication; all replication processes continue without interruption unless they have been stopped by a user. This command merely discontinues your session with **sp_ctrl** on that system. To run **sp_ctrl** again, change to the directory containing the SharePlex binaries and enter the ./**sp_ctrl** command (for Unix and Linux systems) or double-click the **sp_ctrl** shortcut (Windows systems).

There are no [**on** *host*] options for the **quit** command. It must be issued on the system where you want to stop running **sp_ctrl**. This command is the same as the **exit** command.

Usage

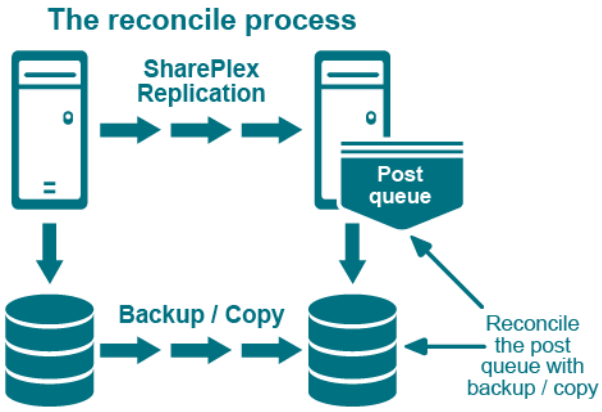
Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**exit**

Syntax

Basic command	Remote options
quit | q	Not available

Reconcile command

Use the **reconcile** command as part of a procedure to synchronize (instantiate) source and target data with minimal interruption to the database users. The **reconcile** command coordinates the results of ongoing replication with a copy of the source data that is applied to the target system, such as that applied by a hot-backup or a native copy utility. The reconcile function compares the replicated changes in the post queue with the state of the target database after the recovery process. It differentiates between the transactions that were applied during recovery from those that have not yet been applied (still waiting in the post queue), and it only posts the non-duplicated changes so that both systems are synchronized.



Although the **reconcile** command is designed for use in high-volume environments, it can be used in low-volume environments with an understanding that the reconcile process can, in some circumstances, seem to stall. This happens because the **reconcile** command depends on data continuing to arrive from the source system. If there is no replication activity on the source system after the hot backup or copy, the reconcile process waits until source activity resumes.

Considerations when using the reconcile command

The **reconcile** command should be used when following specific procedures for the initial synchronization of source and target data. It is not meant to be a standalone command. For initial synchronization procedures, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	target system
Related commands:	**flush**

Syntax

Basic command	Command options	Remote options
reconcile queue *queuename* **for** *datasource-datadest*	[**seq** *sequence_number*] [**scn** *scn_number*] [**to flush**]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description*
queue	**queue** is a required part of the command.
queuename	The post queue on the target system that you want to reconcile. Valid values are: • The name of the source system if using default queues • The name of the queue if using named queues When using named post queues, issue the **reconcile** command for each one. To determine the queue name, issue the **qstatus** command in **sp_ctrl**. Queue names are case-sensitive on all platforms.
for *datasource-datadest*	• *datasource* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the source instance. • *datadest* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the target instance. **Example**: `sp_ctrl (sysB)` > **reconcile queue SysA for o.oraA-o.oraB**
seq *sequence_number*	(Oracle) Use this option when an Oracle hot backup is used to establish Oracle target data in the synchronization procedure. It directs SharePlex to reconcile to the end of the same log that Oracle uses for its recovery. *sequence_number* is the sequence number of the log to which Oracle recovers. The syntax must appear after the syntax for the basic command. Do not use this option with the **to flush** option. **Example**: `sp_ctrl (sysB)` > **reconcile queue SysA for o.oraA-o.oraB seq 1234**
scn *scn_number*	(Oracle) Use this option when an Oracle hot backup is used to establish Oracle target data in the synchronization procedure. It directs SharePlex to reconcile to a specific Oracle System Change Number (SCN). *scn_number* is the SCN to which Oracle recovers.

Component	Description*
	The syntax must appear after the syntax for the basic command. Do not use this option with the **to flush** option. **Example**: `sp_ctrl (sysB)` **> reconcile queue SysA for o.oraA-o.oraB scn 0123456789**
to flush	Use this option to reconcile to a flush marker that is established with the **flush** command. Use it for synchronizing multiple Oracle databases in a peer-to-peer replication environment. The syntax must appear after the syntax for the basic command. Do not use this option with the **seq** *sequence_number* option. **Example**: `sp_ctrl (sysA)` **> reconcile queue SysA for o.oraA-o.oraB to flush**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Redo command

Use the **redo** command to execute the previous command again without having to retype it. This command is useful when you are making frequent status checks with the information commands, for example using the **qstatus** command to monitor changes in queue volume.

This command only can be issued from within **sp_ctrl**; it cannot be issued directly from the operating system's command line. There are no [**on** *host*] options for this command.

SharePlex also provides keyboard short-cuts that perform the same task as the **redo** command. Invoking the forward slash or the up arrow while in **sp_ctrl** will execute the previous command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	To "redo" the previous command, the user must have that command's authorization level or higher.
Issued for:	The system affected by the previous command.
Related commands:	none

Syntax

Basic command	Remote options
redo | **/** | **r**	Not available

Remove config

Use the **remove config** command to permanently delete a configuration file from the system. This command *does not* prompt for confirmation, and removing a configuration file cannot be undone. You cannot remove an active configuration. To remove an active configuration, deactivate it first.

TIP: You might be able to recover an accidentally deleted configuration if that configuration was previously active and you did not run **ora_cleansp** since it was activated. To recover the configuration, view the Event Log to determine the activation ID for that configuration file, then look in the **save** sub-directory of the SharePlex variable-data directory for a .**conf**.*actid* file, where actid is the activation ID you got from the Event Log.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**deactivate config, list config, show config, view config**

Syntax

Basic command	Remote options
remove config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration that you want to remove. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)` **> remove config sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Remove log

The **compare**, **compare using, repair**, **copy**, and **append** commands generate log files both on the source, and on the target. The job information and source log files are cleaned up when the job is older than SP_SYS_JOB_HISTORY_RETENTION, or if the clear history command is used. To remove the log files on the source without clearing job history from the database, or to remove log files from the target, use the remove log command.

For example:

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

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**compare, repair, copy, append**

Syntax

Basic command	Remote options
remove log {all | age *days* | **copy** | **compare**}	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
all	This argument causes all logs to be removed. **Example**: `sp_ctrl(sysA)> remove log all`
age *days*	This argument causes logs older than the specified number of days to be removed. **Example**:

Component	Description
	`sp_ctrl(sysA)` **> remove log age 10**
copy	This argument causes logs for the copy or append commands to be removed. **Example**: `sp_ctrl(sysA)` **> remove log copy**
compare	This argument causes logs for the compare and/or repair command to be removed. **Example**: `sp_ctrl(sysA)` **> remove log**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Rename config

Use the **rename config** command to give a configuration file a different name. Use a name that is unique among the configuration files on the system.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**copy config, edit config, list config, view config**

Syntax

Basic command	Remote options
rename config {*filename* **to** *newname*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename **to** *newname*	• *filename* is the name of the configuration that you want to rename. Configuration names are case-sensitive. • **to** is a required part of the syntax. • *newname* is the new name you are giving the configuration. **Example**: `sp_ctrl(sysA)` **> rename config sales to sales2**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Repair status

Use the **repair status** command to view the status of the last compare or repair job run. The **repair status** command can be used to view detailed status on a compare or repair job or a portion of a compare or repair job, or to view status on all compare and repair jobs for which SharePlex has history.

For details and examples about using the **repair status** command, see the **job status** command

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**copy status, append status, compare status, job status**

Syntax

Basic command	Command options	Remote options
repair status	[*job_id*] [*job_id.table_id*] [**all**] [**full**] [**detail**] [*status*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**: `sp_ctrl(sysA)` > **repair status 2828.2**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example**: `sp_ctrl(sysA)` > **repair status 2828.HR.SRC_TEST3**
all	Displays a summary line for every job with history in the database.

Component	Description
	Example: `sp_ctrl(sysA)` **> repair status all**
full	Displays the status of every object in the job. By default, the job status command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)` **> repair status2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)` **> repair status detail**
status	Displays status history for previous jobs with the specified status. **Example**: `sp_ctrl(SysA)` **> repair status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Report

Use this command to display the **copy/append** history for a table.

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**copy/append**

Syntax

Basic command	Command options	Remote options
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*]

Syntax description

Component	Description
source_owner.source_table	This option specifies a source table for the synchronization report. Allow no spaces between components of the source owner and table name. **Examples**: `sp_ctrl(SysA)` > **report s_user1.sync_single_tbl**
copy | append	This option returns results specific to either **copy** or **append**. **Examples**: `sp_ctrl(SysA)` > **report copy** s_user1.sync_single_tbl `sp_ctrl(SysA)` > **report append** s_user1.sync_single_tbl
toeditor	This option sets the target. **Examples**:

Component	Description
	`sp_ctrl(SysA)` > **report** s_user1.sync_single_tbl **toeditor** `sp_ctrl(SysA)` > **report copy** s_user1.sync_single_tbl **toeditor**
for o.*source_sid*	This option is used for specifying results specific to the provided source SID. **Examples**: `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

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Reset encryption key

Use the **reset encryption key** command to remove the AES encryption key that is being used by SharePlex to encrypt data.

For detailed instructions for configuring SharePlex encryption, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**create encryption key**, **set encryption key**, **show encryption key**

Syntax

Basic command	Remote options
reset encryption key	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**

Option	Description
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Reset param

Use the **reset param** command to restore a parameter to its default value.

The change takes effect based on the parameter's set-at point:

- For a set-at point of *Live*, the change takes effect immediately.
- For a set-at point of *Restart Process*, the change takes effect the next time the affected SharePlex process is started.
- For a set-at point of *Restart Cop*, the change takes effect the next time **sp_cop** is started (do *not* reboot the system).

Using command options

The following are guidelines for resetting SharePlex parameters:

- To reset a parameter that was set with the **set param** command on a global basis (without process-specific options), use the **reset param** command without options.
- To reset a parameter that has a *process-specific* value (set with one of the **set param** process-specific options), use one of the **reset param** process-specific options. Process- specific options can be used for parameters in the following modules:

Module	Naming convention	Function controlled
export	SP_XPT	Export process
import	SP_IMP	Import process
capture	SP_OCT	Capture process
post	SP_OPO	Post process
read	SP_ORD	Read process

Viewing current parameter settings

To view the names of SharePlex parameters, their values and set-at points, and whether or not process-specific values are in effect, use the **list param** command, as shown in the following example which shows both a global value for the Export process and a process-specific value for Export to sysB.

```
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
```

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	source or target system
Related commands:	**list param, set param**

Syntax

Basic command	Process-specific options	Remote options
reset param {*paramname*	[**to** *host*] [**from** *host*] [**for** *datasource*] [**for** *datasource-datadest*] [**queue** *queuename*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

IMPORTANT: Before you begin using this command, please review all of this documentation. Improper use could result in lost target data and out-of-sync conditions.

Syntax description

Component	Description
paramname	The name of the parameter that you want to restore to its default value. This argument is not case-sensitive. **Example**: `sp_ctrl(sysA)` **> reset param SP_XPT_SO_SNDBUF** This resets the parameter for all Export processes on the system.

Process-specific command options

Option	Description
to *host*	This option resets an Export parameter (SP_XPT_) for the Export processes to a designated target system. To restrict the command's effect to the Export process associated with a named export queue,

Option	Description
	follow this option with the [**queue** *queuename*] option. **Example**: `sp_ctrl(sysA)` **> reset param sp_xpt_so_sndbuf to sysB** In this example, the parameter is reset for Export to sysB, but other Export processes are not affected.
from *host*	This option resets an Import parameter (SP_IMP_) for the Import processes from a designated source system. To restrict the command's effect to the Import process associated with a named export queue, follow this option with the [**queue** *queuename*] option. **Example**: `sp_ctrl(sysD)` **> reset param sp_imp_wcmt_msgcnt from sysC** In this example, the parameter is reset for Import from sys C, but other Import processes are not affected.
for*datasource*	This option resets a Capture or Read parameter (SP_OCT_ or SP_ORD_) for a designated source Oracle instance. • datasource is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the source instance. **Example**: `sp_ctrl(sysA)` **> reset param sp_oct_replicate_dload for o.oraA** In this example, the parameter is reset for the Capture process for instance oraA, but other Capture processes are not affected.
for *datasource-datadest*	This option resets a Post parameter (SP_OPO_) for all Post processes posting data from one database to another. To restrict the command's effect to the Post process associated with a named post queue, follow this option with the [**queue** *queuename*] option. • *datasource* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the source instance. • *datadest* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the target instance. **Example**: `sp_ctrl(sysC)` **> reset param sp_opo_cont_on_err for o.oraA-o.oraC** In this example, the parameter is reset for the Post process for instance oraA replicating to oraC, but other Post processes are not affected.
queue *queuename*	This option resets a parameter for an Export, Import or Post process associated with a designated named queue.

Option	Description
	Example: `sp_ctrl(sysA)` **> reset param sp_opo_poster_delay queue QA** In this example, the parameter is reset for the Post process associated with named post queue QA, but other Post processes are not affected.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Set encryption key

Use the **set encryption key** command to configure the Export and Import processes to use the AES encryption key that is generated with the **create encryption key** command.

The encryption key must be set with this command on the source and target systems.

For more information about configuring AES encryption, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**create encryption key, show encryption key, reset encryption key**

Syntax

Basic command	Remote options
set encryption key *key_value*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
key	The encryption key that was generated by the **create encryption key** command. **Example:** `sp_ctrl>` **set encryption key E5F5D4CBA329D2C86B5D7ABA096C18600595490129F55A1422AAB0248B28D0E4**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Set log

Use the **set log** command to set the default parameters for the **show log** command. This command sets defaults for:

- The maximum number of lines to extract from the log.
- The maximum number of lines to display at once on the screen.
- The direction in which you want to view the entries (newest to oldest, or oldest to newest).

You can set any, or all, of those parameters at once with this command. There are no [on host] options for this command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**show log, view log options**

Syntax

Basic command	Remote options
set log {**maxlines=***number* | **lpp=***number* | **reverse** | **forward**}	Not available

Syntax description

Component	Description
filespec	This argument specifies the wildcarded Oracle object (including schema owner) that the command will execute against. Without the for datasource option it assumes that the schema and objects are in replication and are contained within the active configuration. **Example**: `sp_ctrl(sysA)>` **expand scott.%** In this example, the result will display all objects in the scott schema.
maxlines=*number*	Use this option to set the default for the maximum number of lines of a log that the **show log** command displays. This option can be used alone or in combination with any other **set log** command options.

Component	Description
	Examples: `sp_ctrl(sysA)` **> set log maxlines=50** `sp_ctrl(sysA)` **> set log maxlines=50 lpp=34 reverse**
lpp=*number*	Use this option to set the default for the number of lines per screen for the **show log** command. This option can be used alone or in combination with any other **set log** command options. **Examples**: `sp_ctrl(sysA)` **> set log lpp=34** `sp_ctrl(sysA)` **> set log maxlines=50 lpp=34 reverse**
reverse	Use this option to start the **show log** output with the most recent entries. This option can be used alone or in combination with any other **set log** command options. **Examples**: `sp_ctrl(sysA)` **> set log reverse** `sp_ctrl(sysA)` **> set log maxlines=50 lpp=34 reverse**
forward	Use this option to start the **show log** output with the oldest entries. This option can be used alone or in combination with any other **set log** command options. **Examples**: `sp_ctrl(sysA)` **> set log forward** `sp_ctrl(sysA)` **> set log maxlines=50 lpp=34 forward**

Set param

Use the set param command to change the value of a SharePlex parameter.

The new setting takes effect based on the parameter's set-at point:

- For a set-at point of *Live*, the change takes effect immediately.
- For a set-at point of *Restart Process*, the change takes effect the next time the affected SharePlex process is started.
- For a set-at point of *Restart Cop*, the change takes effect the next time that **sp_cop** is started. *Do* not reboot the system.

Changing a parameter from its default value

When you change a parameter from its *default* value, SharePlex records the change in the variable-data directory, where all user-defined parameter values are stored. Default values are stored in the **param-defaults** file in the product directory.

The default settings of most SharePlex parameters are adequate for most replication operations, but some can be changed by an authorized SharePlex user when necessary. Those parameters are viewed with the **list param basic** command.

The rest of the parameters are considered non-user-changeable. They are viewed with the **list param all** command. Do not change those parameters without the supervision of Quest Development or Technical Support team, or unless you are following a documented procedure in a SharePlex guide. Unless you are familiar with those parameters, changing them could adversely affect replication.

Using command options

The following explains how to use the set param command options:

- The basic **set param** command (without options) sets a parameter on a global basis for an instance of **sp_cop**. For example, the following command sets the parameter for all of the Export processes spawned by one instance of **sp_cop**.

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

- Parameters for the following replication processes can be *process-specific*, which enables you to set different values for the same parameter to accommodate multiple instances of a process running from one instance of **sp_cop**.

Module	Naming convention	Function controlled
export	SP_XPT	Export process
import	SP_IMP	Import process
capture	SP_OCT	Capture process
post	SP_OPO	Post process
read	SP_ORD	Read process

You can use a process-specific value in conjunction with a global value for the same parameter. For example, you could set SP_XPT_SO_SNDBUF globally for all Export processes except the one transmitting to SysC; for that one you could set a different value by using the [**to** *host value*] process-specific option.

Viewing current parameter settings

To view the names of SharePlex parameters, their values and set-at points, and whether or not process-specific values are in effect, use the **list param** command, as shown in the following example which shows both a global value for the Export process and a process-specific value for Export to sysB.

```
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
```

Changing a parameter back to its default value

To change the value of a parameter back to its default value, use the **reset param** command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	source or target system
Related commands:	**list param, reset param**

Syntax for global settings

Basic command	Remote options
set param *paramname value*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax for process-specific settings

Basic command	Command options	Remote options

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*]

Syntax description for global settings

Component	Description
paramname value	• *paramname* is the name of the parameter that you want to set. This argument is not case-sensitive. • *value* is the new value for the parameter. **Example**: `sp_ctrl(sysA)` > **set param SP_XPT_SO_SNDBUF 1024** In this example, the parameter is set globally for all Export processes on the system.

Syntax description for process-specific settings

Component	Description
to *host value*	This option sets an Export parameter (SP_XPT_) for all Export processes to a designated target system. To constrain the change to an Export process associated with a named export queue, use this option with the [**queue** *queuename*] option. • *host* is the target system's name. • *value* is the new value for the parameter. **Example**: `sp_ctrl(sysA)` > **set param sp_xpt_so_sndbuf to sysB 1024** In this example, the parameter is set for Export to sysB, but Export processes to other target systems are not affected.

Component	Description
from *host value*	This option sets an Import parameter (SP_IMP_) for all Import processes from a designated source system. To constrain the change to an Import process associated with a named export queue, use this option with the [**queue** *queuename*] option. • *host* is the source system's name. • *value* is the new value for the parameter. **Example**: `sp_ctrl(sysD)` > **set param sp_imp_wcmt_msgcnt from sysC 10000** In this example, the parameter is set for Import from sys C, but Import processes from other source systems are not affected.
for *datasource value*	This option sets a Capture or Read parameter (SP_OCT_ or SP_ORD_) for a designated datasource. • *datasource* is expressed as **o**.*SID*, where *SID* is the ORACLE_ SID of the source Oracle instance. • *value* is the new parameter value. **Example**: `sp_ctrl(sysA)` > **set param sp_oct_replicate_dload for o.oraA 1** In this example, the parameter is set for Capture for instance oraA, but other Capture processes for the same instance of **sp_cop** are not affected.
for *datasource-datadest value*	This option sets a Post parameter (SP_OPO_) for all Post processes posting data from one database to another. To constrain the change to a Post process associated with a named post queue, use this option with the [**queue** *queuename*] option. • *datasource* is expressed as **o**.*SID*, where *SID* is the ORACLE_ SID of the source Oracle instance. • *datadest* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the target Oracle instance. • *value* is the new parameter value. **Example**: `sp_ctrl(sysC)` > **set param sp_opo_cont_on_err for o.oraA-o.oraC 1** In this example, the parameter is set for the Post process for instance oraA replicating to oraC, but other Post processes are not affected.
queue *queuename*	This option restricts the command's effect to an Export, Import or Post

Component	Description
	process associated with a named queue. • *queuename* is the named queue's name. When used, the **queue** *queuename* option must appear after the required and optional syntax, but before the new value, as shown below: **Example 1: Export process** `sp_ctrl(sysA)` **> set param sp_xpt_so_sndbuf queue QA to sysB 1024** In this example, the parameter is set for the Export process associated with named export queue QA, but other Export processes are not affected. **Example 2: Import process** `sp_ctrl(sysD)` **> set param sp_imp_wcmt_msgcnt from sysC queue QA 10000** In this example, the parameter is set for the Import process associated with named export queue QA, but other Import processes are not affected. **Example 3: Post process** `sp_ctrl(sysC)` **> set param sp_opo_cont_on_err queue QA for o.oraA-o.oraC 1** In this example, the parameter is set for the Post process associated with named post queue QA for replication between oraA and oraC, but other Post processes are not affected. **Example 4: Post process** `sp_ctrl(sysC)` **> set param sp_opo_cont_on_err queue QA 1** In this example, the parameter is set for the Post process associated with named post queue QA, but other Post processes are not affected.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show command

Use the **show** command to view the replication processes for an instance of **sp_cop** and the path the data is taking.

The following describes each column in the show command display:

Process

The **Process** column shows the SharePlex replication processes (services) spawned by the instance of **sp_cop** to which the current **sp_ctrl** session is connected. The command does not show the command and control process (**sp_cnc**) or any compare processes that are running.

- If there are named export queues, there are multiple Export or Import processes, depending on whether it is a source or target **sp_cop**.
- If there are named post queues, there are multiple Post processes, one for each queue.
- The Post process is labeled `MTPost`.
- For an instance of **sp_cop** running peer-to-peer replication, or for those that serve as both source and target **sp_cop** instances, all SharePlex replication processes are present.

Source

The **Source** column shows the source of the data being processed by a SharePlex process. A source can be one of the following:

- *For a Capture or Read process*: the datasource (the source Oracle instance).
- *For an Export process*: the name of the associated export queue. Default export queues bear the name of the source system. Named export queues bear their userassigned names, for example exptq1.
- *For an Import process*: the name of the associated default export queue (for example, sysA), or the name of the associated named export queue, such as exptq1.
- *For a Post process*: There can be one of several displays for the source of a Post process, depending on the replication configuration:
    - When there is a *default* export queue and a default post queue, the Post source is the short name of the associated post queue. The short name consists of the datasource and the source system's name, in the format of o.ora10-elliot.
    - When there is a named export queue (which automatically creates a named post queue on the target system) the Post source consists of the datasource and the named export queue's name, in the format of o.ora10-exptq1.
    - When there is a default export queue and a named post queue, the Post source consists of the datasource and the named post queue's name, for example o.ora10- postq1.

Target

The **Target** column shows the destination, or *target*, of the data being processed by a SharePlex process.

A target can be one of the following:

- *For a Capture or Read process*: There is no target to show for these processes. The data captured by the Capture process and routed by the Read process can be routed to one or many destinations.
- *For an Export process*: the name of the target system.
- *For an Import process*: the name of the target system.
- *For a Post process*: the target.

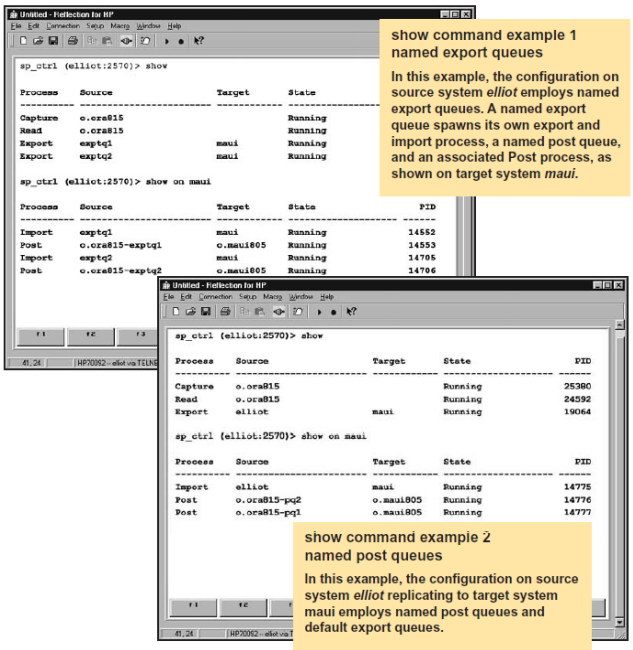
State

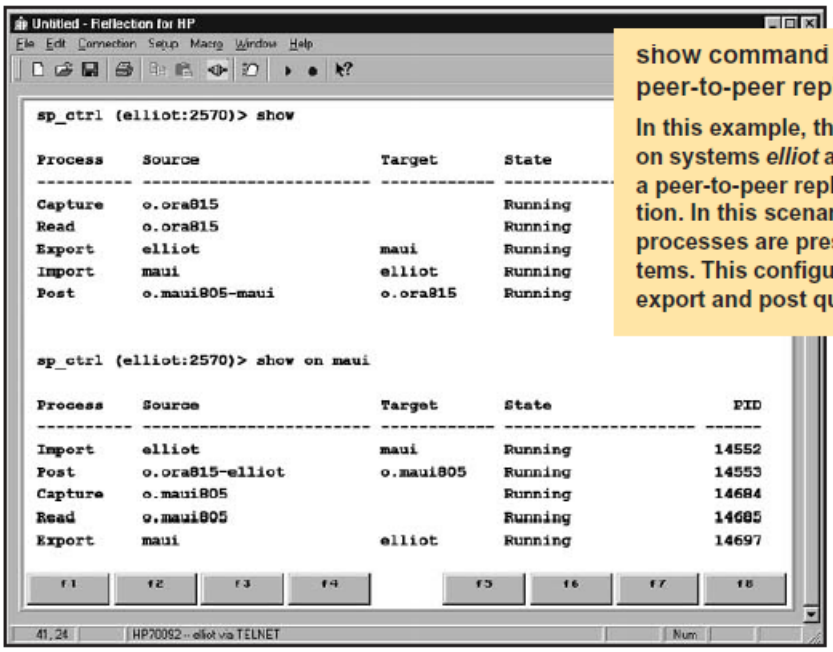
The **State** column shows the status of each process:

- Running
- Idle (waiting for data to process)
- Stopped by a user
- Stopped because of an error.

PID

This column lists the process ID number for each process.





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.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**lstatus, qstatus, status**

Syntax

Basic command	Remote options
show	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Use of Show Capture

Use the **show capture** command to view statistics for the Capture process.

Basic command

The basic **show capture** command shows an overview of the process, such as the datasource, whether the process is running or stopped, and other basic information.

Detailed statistics

To view detailed statistics for the Capture process, use the **show capture** command with the **[detail]** option. That option shows detailed statistics that can help you assess the performance of the process, decide whether tuning parameters need to be adjusted, and detect problems or bottlenecks.

Detailed statistics for Oracle Capture

Statistic	Description
Host	The name of the local machine (source system).
System time	The current time according to the system clock.
Source	The name of the source Oracle instance.
Status	The status of the Capture process (running or stopped).
Since	The time that Capture started.
Oracle current redo log	The sequence number of the redo log to which Oracle is writing.
Capture current redo log	The sequence number of the redo log that Capture is reading.
Capture log offset	The location in the redo log of the record being processed by Capture.
Last redo record processed	The record being processed by Capture or the last one processed if Capture is not currently replicating data.
Capture state	The state of the process, in relation to the replication work it performs **It can be one of the following:** • INITIALIZING: Capture is starting up. • WAITING: Capture is reading the redo log and waiting for records that need to be replicated. • WAITING FOR LOGFILE: Capture is configured to wait and then try again if it cannot access the required log. • PROCESSING: Capture is processing a redo log record for replication.

Statistic	Description
	• STOPPED DUE TO MISSING ARCHIVE LOG: Capture is configured to stop when it cannot access the log that it needs.
Activation ID	The internal identifying number of the configuration activation, which identifies the associated processes and queues.
Error count	The number of records that were skipped due to Oracle errors since Capture started. Data from skipped records is not reflected in the target database.
Operations captured	The number of redo records that Capture successfully processed for replication since it started.
Transactions captured	The number of committed Oracle transactions whose operations Capture successfully replicated since it started.
Concurrent sessions	The number of Oracle sessions being processed at the same time.
HWM concurrent sessions	The largest number of concurrent Oracle sessions since Capture started.
Checkpoints performed	The number of checkpoints to save the state of Capture since Capture started. Frequent checkpointing generates additional overhead on the system, but infrequent checkpoints cause SharePlex to recover less quickly from a system or instance failure. By default, Capture checkpoints every 2,000 messages, but it can be adjusted with the SP_OCT_CHECKPOINT_FREQ parameter.
Total operations processed	The number of all Oracle operations and SharePlex internal operations processed by Capture since it started, including records captured for replication and records for objects not in the configuration.
Total transactions completed	The number of committed Oracle transactions processed by Capture since it started, including transactions captured for replication and transactions for objects not in the replication configuration.
Total Kbytes read	The size in kilobytes of the data that was processed by Capture since it started.
Redo records in progress	The number of records that Capture is processing.
Redo records processed	The total number of redo records processed.
Redo records ignored	The number of records that Capture ignored because they are not associated with objects in the configuration.
Redo records - last HRID	The head row ID value for the last chained row processed by Capture.

NOTE: The **show capture detail** output for RAC systems will vary slightly from what is described here. The information is essentially the same, but is necessarily presented differently.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**show export, show import, show post, show read**

Syntax

Basic command	Command options	Remote options
show capture	[**detail**] [**for** *datasource*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
show capture	Shows the state of the Capture process and a summary of the operations captured.
detail	Shows detailed statistics that can help you tune Capture's performance and diagnose problems. **Example**: `sp_ctrl(sysA)` **> show capture detail**
for *datasource*	This option shows Capture statistics only for a specific datasource. *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID. **Examples**: `sp_ctrl(sysA)` **> show capture for o.oraA**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Config

Use the **show config** command to display statistics for the active configuration.

For example:

```
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
```

The following is displayed if this command is executed before Capture is up:

```
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 provides the following statistics:

- **Config**: The configuration name.
- **Datasource**: The Oracle instance containing the objects being replicated.
- **Activated**: The date and time that the configuration was activated.
- **Actid**: The activation ID number for the configuration (mostly for use by Quest Software. Technical Support).
- **Total Objects**: The number of objects in the configuration file.
- **Total Objects Replicating**: The number of objects in the configuration that are replicating.
- **Total Objects Not Replicating**: The number of objects in the configuration for which activation failed, and therefore are not replicating.
- List of the tables in replication and having a primary or unique key, in replication and having no key, and not in replication

For more information about the objects in the configuration, view the *SID*_**config_log**, where *SID* is the ORACLE_ SID of the instance. SharePlex prints activation results and error messages in this file, which resides in the log sub-directory of the SharePlex variable- data directory.

An example of the log file follows:

```
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:
```

NOTE: It may take a few seconds or longer to display the statistics.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**activate config, view config, verify config**

Syntax

Basic command	Remote options
show config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration for which you want to view statistics. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)` **> show config sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login

Option	Description
	credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Encryption Key

Use the **show encryption key** command to view the AES encryption key that is being used by SharePlex. For more information about SharePlex encryption, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**create encryption key**, **set encryption key**, **reset encryption key**

Syntax

Basic command	Remote options
show encryption key	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**

Option	Description
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Export

Use the **show export** command to view statistics about the Export process. This command keeps a record of the number of messages sent to target systems by all Export processes on the local machine, as well as optional statistic about Export performance. The message count begins over again whenever Export stops and starts again.

NOTE: Typically, a message approximately corresponds to a SQL operation, but there can be multiple messages for one operation on a LONG or LOB column, and there could be one record for numerous operations in an array insert. A message also can be an internal SharePlex operation.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**show capture, show import, show post, show read**

Syntax

Basic command	Command options	Remote options
show export	[**detail**] [**queue** *queuename*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
queue *queuename*	Constrains the output to a specific named Export queue.
detail	Shows the following statistics: • Highest rate since the Export process was activated and the date/time it occurred • Highest rate since Export was started and the date/time it occurred

Component	Description
	• Current rate reported as KB/sec and measured as the total KB in a 60 second time slice • Average packet size during the last 60 second time slice • Number of packets sent during the last 60 second time slice • Average TCP send time during the last 60 second time slice • Largest deviation from average during the last 60 second time slice • Number of packets that deviated more than 50% during the last 60 second time slice

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Example

```
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

Use the **show import** command to view statistics about the Import process. This command keeps a record of the number of messages received from source systems by all Import processes on the local machine. The message count begins over again whenever Import stops and starts again.

NOTE: Typically, a message approximately corresponds to a SQL operation, but there can be multiple messages for one operation on a LONG or LOB column, and there could be one record for numerous operations in an array insert. A message also can be an internal SharePlex operation.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**show capture, show export, show post, show read**

Syntax

Basic command	Command options	Remote options
show import	[**queue** *queuename*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
queue *queuename*	Constrains the output to a specific named import queue.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Log

Use the **show log** command to view the SharePlex Event log or user issued commands through the **sp_ctrl** interface, instead of opening the log file directly through the operating system. This command can be used for logs smaller than 2 MB in size; otherwise, view the log directly through the filesystem. The default command used without any options displays 60 lines from the Event Log at 15 lines per page on the screen, starting with the oldest entry.

- To view the current show log default parameters, use the **view log** options command.
- To change the defaults, use the **set log** command.

To override the set log defaults, you can use optional syntax to:

- Set the order of entries
- The number of lines extracted from the log
- The number of lines on the screen.
- Filter entries by keyword
- View the user issued commands instead of the full Event Log.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**set log, view log options**

Syntax

Basic command	Command options	Remote options
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*]

Syntax description

NOTE: To obtain the *datasource*, *datadest*, and *queuename* values in this command, use the **lstatus** command and view the **Queues** section of the output.

Component	Description
event	Use this option to display the Event Log. This option, if used, must appear in the syntax before any other option. It cannot be used with the **[command]** option. **Example**: `sp_ctrl(sysA)` **> show log event**
command	Use this option to view the user issued commands. This option, if used, must appear in the syntax before any other option. It cannot be used with the **[event]** option. **Example**: `sp_ctrl(sysA)` **> show log command**
post for *datadest* **queue** *queuename*	Use this option to view the Post Log. This option must specify the data destination and the queue name. **Example**: `sp_ctrl(sysA)` **> show log post for mydbqueue q1**
capture for *datasource*	Use this option to view the Capture Log. This option must specify the datasource. **Example**: `sp_ctrl(sysA)` **> show log capture for o.mySID**
read for *datasource*	Use this option to view the Read Log. This option must specify the datasource. **Example**: `sp_ctrl(sysA)` **> show log read for o.mySID**
maxlines=*number*	Use this option to specify the maximum number of lines to extract from the log. Without this option, show log defaults to 60 lines or the value set with the set log command. **Example**: `sp_ctrl(sysA)` **> show log maxlines=50**
lpp=*number*	Use this option to customize the output to fit your monitor screen. It specifies the number of lines to display on your screen. Without this option, **show log** defaults to 15 lines or the value set with the **set log** command.

Component	Description
	Example: `sp_ctrl(sysA)` **> show log lpp=34**
reverse	Use this option to order the display starting with the most recent entry. Without this option, **show log** defaults to the forward direction or the value set with the **set log** command. **Example**: `sp_ctrl(sysA)` **> show log reverse**
forward	Use this option to order the display starting with the oldest entry. Without this option, **show log** defaults to the forward direction or the value set with the **set log** command. **Example**: `sp_ctrl(sysA)` **> show log forward**

filter=*keyword*	Use this option to filter log entries based on a keyword. There are two ways to use this option:
	• To view only those lines *containing* the keyword, type **filter=***keyword*
	• To *exclude* lines containing the keyword, precede the keyword with an exclamation point (!), as in **filter=!***keyword*
	The keyword cannot contain blanks.
	Examples:
	`sp_ctrl(sysA)` **> show log filter=compare**
	The preceding example extracts only the compare related messages from the Event Log.
	`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**
	The preceding example *excludes* all Notice entries from the Event Log but shows all other types of entries.
	`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)`

Component	Description
	Info 08-07-08 22:47:22.334040 23710 1 Command server launched, pid = 23710 (connecting from 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

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Post

Use the **show post** command to view statistics for the Post process.

Basic show post command

The basic **show post** command shows global statistics for all sessions a Post process. It shows the status of the Post process and the number of messages posted since it started. To filter the output for a specific post queue or datasource (useful when you have multiple replicating data streams), use the **queue***queuename* or **for***datasourcedatadest* option.

Detailed show post command

To view detailed statistics for the Post process, use the **show post** command with the **detail** option. That option shows the most recent SQL statement processed, as well as other statistics that can help you assess Post's performance, decide whether tuning parameters need to be adjusted, and detect problems or bottlenecks.

The following explains the detailed statistics shown with **show post**. These statistics vary slightly depending on the type of source and target.

Statistic	Description
Host	The name of the local machine (target system).
Source	The source of the data being processed by Post.
Queue	The Post queue for this Post process. For a default Post queue, it is the name of the source system. For a named queue, it is the user-defined name.
Target	The name of the target of this Post process, for example the name of an Oracle instance or Open Target database.
Status	The status of the Post process (running or stopped). Possible statuses are: • Running • Stopping • Stopped by user • Stopped due to flush • Stopped due to error
Operations posted **Operations processed**	The number of transactional operations and SharePlex internal operations that this Post process processed since it was started.
Since	The time that Post started.
Total	The number of messages in the queue that have yet to be read-released. This number corresponds to the 'Number of messages' returned from running **qstatus**.

Statistic	Description
Backlog	The number of messages that are waiting in the queue to be processed by Post.
Last operation posted	Identifying information for the most current operation that is being posted to the target if Post is active, or the last operation posted if it is inactive. This information is specific to the type of datastore that originated the data. An operation can be: • INSERT • UPDATE • DELETE • TRUNCATE • COMMIT • DDL statement • INSERT_MULTIPLE or DELETE_MULTIPLE (array operations). • SharePlex internal operation.
Last transaction posted **Last transaction processed**	Identifying information for the last transaction that was posted. This information is specific to the type of datastore that originated the data.
Last file switch	Shown if the target is output to a file. Shows the record identifier for the record after which the active file was switched to a new one.
Post state	The state of the Post process, in relation to the replication work it performs. It can be one of the following: • **Waiting**: Post is waiting for messages to process. • **Active**: Post is posting changes to the database. • **Committed**: Post is committing the transaction. • **Idle**: Post has no open transactions to process. • **Rollback**: Post is processing a rollback. • **Recovery**: Post is in a crash-recovery mode.
Activation ID	The activation ID of the current configuration.
Number of messages read released	Shown for an Oracle target. Shows the difference between the number of operations posted and the actual number of operations removed from the queue. This field is used mainly by Technical Support staff when Post is stalled.
Number of threads	Shown for an Oracle target. Shows the number of processing threads in a multithreaded Post, including the main and timekeeper threads.
Number of Oracle connections	Shown for an Oracle target. Shows the number of connections that a multithreaded Post has open.

Statistic	Description
Concurrency (Active sessions)	Shown for an Oracle target. Shows the number of concurrent transactions that multithreaded Post is processing.
Peak number of sessions	Shown for an Oracle target. Shows the highest number of concurrent transactions that multithreaded Post processed since it was started.
Operations posted **Operations processed**	The number of SQL operations that Post applied to the target, whether or not the COMMIT was received.
Transactions posted **Transactions processed**	The number of committed transactions that Post applied to the target since it was started.
Full rollbacks	Shown for an Oracle target. Shows the number of rolled back transactions processed by Post.
Full rollback operations posted	Shown for an Oracle target. Shows the number of operations that Post applied before the transcation was rolled back. (Post does not wait for a commit to start applying operations that it has in the queue.)
Full rollback operations skipped	Shown for an Oracle target. Shows the number of operations that Post skipped because it received an early indication that the transaction was rolled back.
Transactions <= 2	Shown for an Oracle target. Shows the number of transactions that contain two or fewer operations.
Transactions > 10000	Shown for an Oracle target. Shows the number of transactions that contain more than 10,000 operations.
Largest transaction	Shown for an Oracle target. Shows the size of the largest transaction processed.
Insert operations	The number of INSERT operations processed by Post since it was started.
Update operations	The number of UPDATE operations processed by Post since it was started.
Delete operations	The number of DELETE operations processed by Post since it was started.
Insert batch operations / average	Shown for an Oracle target. Shows the number of INSERT_BATCH operations processed / average number of INSERT operations in one INSERT_BATCH statement.
Delete batch operations / average	Shown for an Oracle target. Shows the number of DELETE_BATCH statements processed / average number of DELETE operations in one DELETE_BATCH statement.
Other operations	Shown for an Oracle target. Shows the number of operations, other than INSERT, UPDATE or DELETE, processed by Post since it was started.
LOB changes	Shown for an Oracle target. Shows the number of LOB change operations.

Statistic	Description
Key cache hit count	Shown for an Oracle target. Shows the number of times that the SharePlex internal key cache was used by the Post process to generate UPDATE statements. SharePlex uses the key cache to post data to the target instance quickly.
SQL cache hit count	Shown for an Oracle target. Shows the ratio of the *total number of messages that were executed without parsing and binding* divided by the *total number of INSERT, UPDATE and DELETE operations*. For more information on the SQL Cache feature of SharePlex, see the SharePlex Administration Guide.
File switches	Shown if the target is output to a file. Shows the total number of file switches performed by Post.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	target system
Related commands:	**show capture, show read, show export, show import**

Syntax

Basic command	Command options	Remote options
show post	[**detail**] [**queue** *queuename*] [**for** *datasource-datadest*] [**sessions**] [**session=***n*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
show post	Shows the state of the Process process and a summary of the operations processed.
detail	This option displays detailed statistics for the Post process. **Example**:

Component	Description
	`sp_ctrl(sysB)` > **show post detail**

queue *queuename*	This option filters the **show post** display for a specific post queue.
	• **queue** is a required part of the syntax.
	• *queuename* is the post queue for which you want to see Post statistics. Valid values are:
	  • the name of the source system if using default queues.
	  • the user-defined queue name, if using named queues.
	If you are unsure what the queue name is, issue the **qstatus** command. Queue names are case-sensitive on all platforms.
	This option can appear in any order with other options.
	Example:
	`sp_ctrl(sysB)` > **show post queue sysA**

for *datasource-datadest*	This option filters the show post display for a specific data stream. • **for** is a required part of the syntax. • *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID. • *datadest* is expressed as one of the following, depending on the target: **o**.*ORACLE_SID* **r.***database_name* **x.kafka** **x.jms** **x.file** This option can appear in any order with other options. **Example**: `sp_ctrl(sysB)` > **show post for o.oraA-r.ssB**
sessions	For Oracle targets, this option displays statistics for all of the threads spawned by the Post process. For Open Target, which is single-threaded, this option can be used to view details for that thread. This option can appear in any order with other options. **Example**: `sp_ctrl(sysB)` > **show post sessions queue *queuename***

Component	Description
session=*n*	(Oracle only) This option displays statistics shown in the **show post sessions** output plus additional details about the state of the thread, the number of messages waiting to be processed by the thread, and the status of the overall transaction. • **session=** is a required part of the syntax. • *n* is any session number displayed with the **show post sessions** command. Leave no space between any of the components. This option can appear in any order with any other option. **Example**: `sp_ctrl(sysB)` **> show post session=1234 queue *queuename***

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Read

Use the **show read** command to view statistics about the Read process.

Basic Command

The basic **show read** command shows an overview of the process, such as the data source, whether the process is running or stopped, and other basic information.

Detailed statistics

To view detailed statistics for the Read process, use the **show read** command with the **[detail]** option. This option shows detailed statistics that can help you assess the performance of the process, decide whether tuning parameters need to be adjusted, and detect problems or bottlenecks.

Detailed statistics for Oracle Read

Statistic	Description
Host	The name of the local machine (source system).
Source	The name of the source Oracle instance.
Status	The status of the Read process (running or stopped).
Operations forwarded	The number of operations that Read sent to the Export queue.
Since	The time that Read started.
Total	The number of messages in the capture queue that have yet to be read-released. This number corresponds to the 'Number of messages' returned from running **qstatus**.
Backlog	The number of messages that are waiting in the capture queue to be processed by Read.
Last operation forwarded	Identifying information about the most recent operation that Read sent to the export queue, or the last operation it sent if Read is inactive. An operation can be data relating to a DML or DDL operation or an internal SharePlex operation.
Read state	The state of the process, in relation to the replication work it performs: It can be one of the following: • **IDLE**: Read is waiting to collect and process the next batch of records from the capture queue. If Read is idle too long, it could mean that Capture is running behind or that it is idle because there is no new data in the capture queue. If that is not the case, and you want to reduce the latency of the Read process, you can adjust the SP_ORD_DELAY_RECORDS parameter to have Read collect and process records faster. • **PASS1**: Read is in the first phase of processing the data and packaging it for routing.

Statistic	Description
	• **PASS2**: Read is in its second phase of processing. The **Read State** field is useful if Read appears to be taking too long to process an operation.
Activation ID	The internal identifying number of the configuration activation, which identifies the associated processes and queues.
Operations forwarded	The number of transactional and internal SharePlex operations sent by Read to the export queue since it started.
Transactions forwarded	The number of committed transactions sent by Read to the export queue since it started. Comparing this value with the value for the *operations forwarded* indicates whether the transactions tend to be small or large. You can use that information to determine why Read appears to be falling behind and why the queues are not emptying (large transactions without a COMMIT).
Full rollbacks	The number of rolled back transactions processed by Read.
Full rollback operations skipped	The number of operations that Read does not forward to the export queue because the transaction was rolled back.
Cursor cache hit count	The number of times Read used a cached cursor.
Cursor cache miss count	The number of times Read could not use a cached cursor.
Number of open cursors	The number of open cursors reserved by Read to access Oracle if necessary.
Number of active batches	The number of transactions currently active that are being processed as a batch transaction. If enabled, Read will combine batch operations so that they can be posted more quickly.
Batch message total	The number of operations that the Read process combines into batch operations.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**show capture, show export, show import, show post**

Syntax

Basic command	Command options	Remote options
show read	[**detail**] [**for** *datasource*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
show read	Shows the state of the Read process and a summary of the operations processed.
detail	This option displays detailed statistics for the Read process. **Example**: `sp_ctrl(sysA)` **> show read detail**
for *datasource*	This option shows Read statistics for a specific datasource. *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID. **Examples**: `sp_ctrl(sysA)` **> show read for o.oraA**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show SQL

Use the **show SQL** command to view the SQL statement being written by the Post process to post data to the target database. If the post queue is empty, or if Post is not processing a statement, the **show SQL** command shows the most recent SQL statement processed.

Use this command when you think replication is taking too long, or when Post stops on an error. Knowing which SQL statement is at fault can help you determine what is wrong and whether or not further action is required.

The following is an example of the show sql display:

```
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)
```

Issue this command for the target system.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	target system
Related commands:	none

Syntax

Basic command	Command options	Remote options
show sql	[**queue** *queuename*] [**for** *datasource-datadest*] [**thread=***n*] [**session=***n*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
show sql	Without options, shows all SQL statements.
queue *queuename*	Use this option to show the SQL statement for a specific named queue. • **queue** is a required part of the syntax.

Component	Description
	• *queuename* is the name of the queue. If you are unsure what the queue name is, issue the qstatus command. Queue names are case-sensitive on all platforms. **Example**: `sp_ctrl(sysA)` > **show sql queue q1**
for *datasource-datadest*	Use this option to show the SQL for a specific replication stream when you are replicating to or from more than one Oracle instance. • **for** is a required part of the syntax. • *datasource* is expressed as **o**.*SID*, where *SID* is the ORACLE_ SID of the source Oracle instance. • *datadest* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the target Oracle instance. **Example**: `sp_ctrl(sysA)` > **show sql for o.oraA-o.oraB**
thread=*n*	Use this option to show the SQL for a post processing thread. • **thread=** is a required part of the syntax. • *n* is a thread number displayed with the show post threads command. Leave no spaces between the components. This option can appear in any order with other options. **Example**: `sp_ctrl(sysB)` > **show sql thread=1234**
session=*n*	Use this option to show the SQL for a user session. • **session=** is a required part of the syntax. • *n* is a session number displayed with the **show post threads** command. Leave no space between the components. This option can appear in any order with other options. **Example**: `sp_ctrl(sysB)` > **show sql session=1234**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Statusdb

Use the **show statusdb** command to view the Status Database. Each machine involved in replication has its own Status Database that contains records of key replication events, including those that did not generate an error message or warning at the user interface. This information can alert you to potential problems and help you resolve existing ones.

The show statusdb display includes the following information:

- **Level**: whether the entry is there only for information purposes or whether it is the result of an error or warning condition.
- **Details**: the reason for the event.

When appropriate, the Status Database refers you to the Event Log if there is more information about an entry.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**status, lstatus**

Syntax

Basic command	Command options	Remote options
show statusdb	**detail**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
detail	This option displays a higher level of detail for the Status Database. **Example**: `sp_ctrl(sysB)` **> show statusdb detail**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show SCN

Use the **show scn** command to view the Oracle SCN of the last transaction that a Post process applied to the target. The command shows all of the last SCNs when using multiple post queues.

NOTE: This command stops all Post processes in order to obtain the SCN information.

```
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
```

Usage

Supported sources:	Oracle
Supported targets:	Oracle
Authorization level:	Viewer (3)
Issued for:	target system
Related commands:	**activate config**

Syntax

Basic command	Remote options
show scn	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Show Sync

The **show sync** command shows any out-of-sync issues that Post may have encountered. The information is extracted from the Status Database.

For each out-of-sync object, the command displays:

- **Count**: the number of statements that failed
- **Detail**: contains the table name, queue name, date and time

To see which SQL statement caused the error, view the **SID_errlog.sql** log file. The Event Log also will contain a record of the problem.

If nothing is out of synchronization when you issue the **show sync** command, you will be returned to the **sp_ crtl** prompt.

Issue this command for the target system.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	target system
Related commands:	**show statusdb**

Syntax

Basic command	Remote options
show sync	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where

Option	Description
	the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Shutdown

Use the **shutdown** command to shut down replication. Upon completion of the shutdown command, SharePlex shuts down **sp_ctrl** automatically.

Shutting down gracefully

The basic **shutdown** command shuts down **sp_cop** (the Sp_Copsrv on Windows systems) and all other SharePlex processes gracefully, saving the state of each process, performing a checkpoint to disk, read-releasing buffered data, and cleaning up child processes. Data in the queues remains safely in place, ready for processing when an authorized user starts **sp_cop** again.

Shutting down forcefully

The **shutdown** command with the **[force]** option kills the SharePlex processes immediately, whether or not the normal shutdown procedures were completed. SharePlex recovers from a forced shutdown when you start **sp_cop**, but replication processing resumes from a previous save point, which means startup can be slightly longer.

Use the **[force]** option only if you tried to shut down SharePlex gracefully, but child processes remained running. You can verify whether SharePlex processes are still running by issuing the **ps -ef | grep sp_** command in the command shell or by viewing the Processes tab in Windows Task Manager.

IMPORTANT: If you shut down replication and users continue changing the objects in the active configuration(s), it is possible for the Oracle redo logs to wrap before SharePlex starts again. If the delay is long and the archive logs become unavailable, you will need to re-synchronize the data. The redo logs should be large and numerous enough to accommodate the time SharePlex will be shut down. For more information about the proper sizing of the redo logs for replication, see the SharePlex Installation and Setup Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	source or target system
Related commands:	**startup**

Syntax

Basic command	Command options	Remote options
shutdown	[**force**]	[**on** *host* |

Basic command	Command options	Remote options
		on *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
force	This option shuts down all replication processes, including **sp_cop**, immediately, whether or not they have finished reading from or writing to a queue. Use this option if the default **shutdown** command fails or you cannot wait for a graceful shutdown. **Example**: `sp_ctrl(sysA)` **> shutdown force**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Start

Use the **start** command to start a replication process after it was stopped using the **stop** or **abort** [*service*] command, or after Post was stopped by the **flush** command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2); Administer (1) required for Capture
Issued for:	source or target system
Related commands:	**stop**

Syntax

Basic command	Command options	Remote options
start *service*	[**to** *host*] [**from** *host*] [**for** *datasource*] [**for** *datasource-datadest*] [**queue** *queuename*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
service	The SharePlex process you want to start. Valid values are: • Capture • Read • Export • Import • Post **Example**: `sp_ctrl(sysA)` **> start export**

Component	Description
to *host*	This option starts Export to a designated target system only. • **to** is a required part of the syntax. • *host* is the name of the target system. **Example**: `sp_ctrl(sysA)` > **start export to sysB** In this example, Export is started from sysA to sysB.
from *host*	This option starts Import from a designated source system only. • **from** is a required part of the syntax. • *host* is the name of the source system. **Example**: `sp_ctrl(sysD)` > **start import from sysA** In this example, Import is started on sysD from sysA.
for *datasource*	This option starts Capture or Read for a designated datasource. • **for** is a required part of the syntax. • *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID. **Example**: `sp_ctrl(sysA)` > **start read for o.oraA** In this example, Read is started on sysA for instance oraA.
for *datasource-datadest*	This option starts Post for a designated source SID-target SID data stream only. • **for** is a required part of the syntax. • *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID. • *datadest* is expressed as **o**.*SID* or **r.***database*, where *SID* is an ORACLE_SID or *database* is the name of an Open Target database. **Example**: `sp_ctrl(sysC)` > **start post for o.oraA-r.mssA** In this example, Post is started on sysC for instance oraA replicating to SQL Server database mssA.
queue *queuename*	This option starts the Export, Import or Post process associated with a designated named queue.

Component	Description
	• **queue** is a required part of the syntax. • *queuename* is the named of the queue as written in the configuration file. **Example**: `sp_ctrl(sysA)` **> start export queue QA** In this example, Export of data through named export queue QA is started on sysA.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Startup

Use the **startup** command to start all processes at once. All of the processes will start unless they were stopped with the **stop** command prior to shutting down **sp_cop**.

When there is an active configuration, replication normally starts automatically when you start **sp_cop** from the command line.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	source or target system
Related commands:	**shutdown**

Syntax

Basic command	Remote options
startup	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax.

Option	Description
	Example: `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Status

Use the **status** command to view a summary of the status of replication on a system, to ensure that processes are running and to check for errors, warnings or notices. For a more detailed status report, use the **lstatus** command.

The status display shows:

- **Process**: The name of the process.
- **State**: The status of each process, either *running, idle, stopped due to error, or stopped by user*
- **PID**: The operating-system process ID number of the process
- **Running Since**: The date and time that the process was started
- **Other information**: such as how the system is being used, if there is an active configuration on the system, and if replication errors occurred.

SharePlex updates the status display at intervals determined by the SP_COP_IDLETIME parameter.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**lstatus, qstatus, show, show statusdb**

Syntax

Basic command	Remote options
status	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Stop

Use the **stop** command to stop a SharePlex replication process gracefully, allowing it to finish reading from, or writing to, its associated queues. When you stop a replication process, data accumulates safely in the queues.

A process stopped with the **stop** command remains stopped even when SharePlex is shut down and restarted. It can only be started with the **start** command. When the process starts again, it resumes processing at the point where stopped, so the data remains synchronized.

NOTE: The **stop** command differs from the **abort service** command in that it stops a replication process after the process has finished reading from, or writing to, its associated queue. The **abort service** command immediately terminates the process, whether or not the process is processing data.

Stopping the Capture process

Be careful when you stop the Capture process.

The source and target data can go out of synchronization if:

- Users continue changing the source data while Capture is stopped.

  and...

- The Oracle redo logs wrap during that time.

  and...

- The archive logs become unavailable.

Stopping Post at a specific point in time

Use one of the [**at sourcetime**] options to control when the Post process stops. Either of those options automatically stops Post when it receives the first message stamped with a designated time, or time and date. Subsequent messages accumulate in the post queue until Post is restarted.

You can use an [**at sourcetime**] option when a job is scheduled on the source system, and you want to be certain that the operations were successful there before they are applied to the target database. Set the option to stop Post just before the job is scheduled to run.

You also can use this option when you are running reports on the target system and do not want Post operations competing for overhead.

Because SharePlex is not synchronous, the actual time on the target system when Post stops probably will be later than the operation's timestamp. The timing depends on how long it takes to post preceding messages in the queue.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2); Administrator (1) required to stop Capture
Issued for:	• **stop** for Capture, Read and Export is issued for the source system. • **stop** for Import and Post is issued for the target system.
Related commands:	**abort service, start**

Syntax

Basic command	Command options	Remote options
stop *service*	[**to** *host*] [**from** *host*] [**for** *datasource*] [**for** *datasource-datadest*] [**queue** *queuename*] [**at sourcetime** *hh:mm:ss*] [**at sourcetime** *mm/dd/yyyy hh:mm:ss*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
service	The SharePlex process you want to stop. Valid values are: • Capture • Read • Export • Import • Post **Example**: `sp_ctrl(sysA)` **> stop export**

Component	Description
to *host*	This option stops Export to a designated target system, while allowing Export to other systems to continue. • **to** is a required part of the syntax. • *host* is the name of the target system. **Example**: `sp_ctrl(sysA)` > **stop export to sysB** In this example, Export is stopped from sysA to sysB, but Export from sysA to other target systems continues.
from *host*	This option stops Import from a designated source system, while allowing Import from other systems to continue. • **from** is a required part of the syntax. • *host* is the name of the source system. **Example**: `sp_ctrl(sysD)` > **stop import from sysC** In this example, Import from sys C is stopped on sysD, but Import on sysD from other systems continues.
for *datasource*	This option stops Capture or Read for a designated datasource. • **for** is a required part of the syntax. • *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID. **Example**: `sp_ctrl(sysA)` > **stop read for o.oraA** In this example, Read is stopped on sysA for instance oraA, but other Read processes for other instances on sysA continue processing.
for *datasource-datadest*	This option stops Post for a designated source-target data stream. • **for** is a required part of the syntax. • *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID. • *datadest* is expressed as **o**.*SID* or **r.***database*, where *SID* is an ORACLE_SID or *database* is the name of an Open Target database. **Example**: `sp_ctrl(sysC)` > **stop post for o.oraA-o.oraC**

Component	Description
	In this example, Post is stopped on sysC for instance oraA replicating to oraC, but other Post processes on sysC continue posting.
queue *queuename*	This option stops the Export, Import or Post process associated with a named queue. Replication through other named queues continues unaffected. • **queue** is a required part of the syntax. • *queuename* is the user-defined name of the queue, as shown in the configuration file. **Example**: `sp_ctrl(sysA)` > **stop export queue QA** In this example, Export of data through named export queue QA is stopped on sysA, but Export continues for data assigned to all other named queues.
at sourcetime *hh:mm:ss*	This option stops the Post process when it receives the first message stamped with the designated time. Subsequent messages accumulate in the post queue until Post is restarted. • **at sourcetime** is a required part of the syntax. • *hh:mm:ss* is the time stamp at which Post stops, which must be specified using a 24-hour clock. Pad single-digit components with a zero (0). Allow no spaces. **Example**: `sp_ctrl(sysC)` > **stop post at sourcetime 24:00:00** In this example, Post stops on sysC when it receives the first message with the timestamp of midnight. **NOTE:** This component supports only Oracle-to-Oracle replication.
at sourcetime *mm/dd/yyyy hh:mm:ss*	This option stops the Post process when it receives the first message stamped with the designated time and date. It has the same purpose and functionality as **stop post at sourcetime** *hh:mm:ss*, with an added date option. • **at sourcetime** is a required part of the syntax. • *mm/dd/yyyy* is the date on which you want Post to stop at the designated time. The date component must precede the time component in the syntax. The year must include all four digits. Pad single-digit components with a zero (0). Allow no spaces. Separate this component from the time component with a space.

Component	Description
	• *hh:mm:ss* is the time stamp at which Post stops, which must be specified using a 24-hour clock. Pad single-digit components with a zero (0). **Example**: `sp_ctrl(sysC)` **> stop post at sourcetime 01/31/2002 24:00:00** In this example, Post stops on sysC when it receives the first message stamped midnight, January 31, 2002. **NOTE:** This component supports only Oracle-to-Oracle replication.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Target

Use the **target** command to configure Post to support a specific type of target. This is known as the *target configuration*. The properties stored in the target configuration control how Post processes, formats, and outputs replicated data.

The **target** command also provides options to reset or view target configuration settings.

The **target** command can be used to control a target configuration both at the global level or, if the **queue** option is used, at the per-process level.

IMPORTANT: Make certain to stop and then restart the Post process after using this command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	target system
Related commands:	**connection**

Syntax

Basic command	Command options	Remote options

target {**o.** | **r.** | **x.** | **c.**}*target*
[**queue** *queuename*]
{*command_option*}
	{
	{ **set** *category property* |
	reset [*category* [*property*]] |
	show [*category*] } |
	switch
	}
		Not available

Syntax description

Component	Description
{**o.** | **r.** | **x.** | **c.**}*target*	The target for which you are configuring Post. The letter preceding the name of the target identifies the type of target, whether an Oracle database, Open Target database, JMS, Kafka broker, file, or change-

Component	Description
	history target. **Possible targets are:** **o.***SID* where: *SID* is the ORACLE_SID of an Oracle database. **r.***database* where: *database* is the name (not the DSN) of an Open Target database.* **x.jms** where: **jms** is a Java Message Service (JMS) queue or topic. **x.kafka** where: **kafka** is an Apache Kafka broker. **x.file** where: **file** is a structured file that contains data in one of the supported formats (see format category). **c.***SID* where: *SID* is the ORACLE_SID of a target Oracle database that is configured as a change-history target. * **NOTE:** The SharePlex-supported datastore types, data types, and operations are listed in the SharePlex Release Notes.
queue *queuename*	Constrains the action of the command to the SharePlex Post process that is associated with the specified queue. This action overrides any global settings for the target property you are setting. Without the **queue** option, the target command affects all Post processes in the local SharePlex instance. Example: the following sets metadata properties for the Post process that is associated with the **myjms1** queue. **target x.jms queue myjms1 set metadata time, userid, trans, size**

{
set *category property* |
reset [*category* [*property*]] |
show [*category*] |
switch
}
	• **set** applies a configuration property to the target configuration.
	• **reset** clears configuration settings back to their defaults (if one exists). It can be constrained to a specific category and (where applicable) a specific property.
	NOTE: **reset** does not reset one property value to another. Use another **set** command for that.
	• **show** displays configuration settings.

Component	Description
	• *category* is any category of configuration properties. The properties are grouped by category for efficient management through commands. See Category descriptions. • *property* is the configuration property that you want to set, reset, or show. There are different properties per category. • **switch** applies only to file targets (**x.file**). Use this option to create a new active output file and save the current one as a serially numbered aged file. After issuing this command, the switch occurs after Post processes a new record. This option takes no input and is not associated with any configuration categories.

Category descriptions

Category	Description
cdc	Overrides the global settings for an Oracle change-history target that are set with the SP_OPO_TRACK_PREIMAGE parameter for a specific table. See CDC category.
characterset	Directs the Post process to convert replicated data to the specified character set before being posted to an Open Target target database. See Characterset category.
file	Sets the properties of a file target. See file category for more information.
format	Sets the format of the data that is written to the target. See format category for more information.
json	Sets the properties of JSON output when **format record=json**. See JSON category for more information.
jms	Sets the properties of a JMS target. See JMS category for more information.
kafka	Sets the properties of a Kafka target. See kafka category for more information.
metadata	Adds metadata properties to the data that is replicated by SharePlex. See Metadata category for more information.
resources	Sets parameters that affect resources on the target system. See Resources category
rule | filter	Specifies a conditional statement that applies a tracking rule or filter to a specific table in an Oracle change-history target. See Rule | Filter category
source	Sets source identifier properties for an Oracle change-history target. See Source category for more information.
sql	Sets the properties of SQL output when **format record=sql**. See SQL category for more information.

CDC category

Supported targets

Oracle change-history target

Command options

target c.*SID* [**queue** *queuename*]

table *tablename*

{

set cdc preimage={**yes** | **no**}

reset cdc

show cdc

}

Usage

Overrides the global setting for updates that is set with the SP_OPO_TRACK_PREIMAGE parameter for a specific table. This controls whether the before image of update operations are tracked in the history table. Valid values are **yes** to track before images of updates, or **no** to exclude update before images.

Characterset category

Supported targets

Open Target databases

Command options

target r.*database* [**queue** *queuename*]

{

set characterset *character_set* |

reset characterset *character_set* |

show characterset *character_set*

}

Usage

Directs the Post process to send replicated data to the local Oracle client to be converted to the specified character set before being posted to the target database. The specified character set overrides the default character set of Unicode that is used by Post when posting character data to Open Target targets.

UTF8 to UTF8 emoji replication works fine for Oracle to Open Poster. However, when you are inserting emoji characters into an Oracle source database that is not in UTF8 format (like CESU8), in such scenarios, to replicate

emoji characters to a non-Oracle target, the following configuration settings need to be applied to the target SharePlex instance:

1. Install the Oracle client on the target for data conversion.
2. Set the **SP_OPX_NLS_CONVERSION** parameter to 1 (default).
3. Set the target character set to AL32UTF8 using the target command for converting data to the valid UTF8 byte sequence required by the target DB.

NOTE: To use this option, there must be an Oracle client on the target system. For more information, see the Open Target Checklist section in the SharePlex Installation and Setup Guide.

Property	Input Value	Default
character_set	The Oracle equivalent of the target character set, *specified by its Oracle name*.	Unicode

Example

To post the data in the GB2312 character set on the target, specify the Oracle ZHS16GBK character set.

target r.mymss set characterset ZHS16GBK

file category

Supported targets

File

Command options

target x.file [**queue** *queuename*]

{

set file *property=value* |

reset file [*property*] |

show file

}

Usage

Sets the properties of a file target.

Property	Input Value	Default
location=*pathname*	Path name under the SharePlex variable-data directory where you want the file to be created	opx

Property	Input Value	Default
max_records=*number*	Maximum size of the active file, measured by the number of records, before switching files.	50,000
max_size=*megabytes*	Maximum size of the file, measured in megabytes, before switching files.	50
max_time=*seconds*	Maximum number of seconds to wait before switching files.	300
record_length=*number*	Maximum size of a record, in number of characters	132

Example

target x.file set file max_size=320

format category

Supported targets

File

Kafka

Command options

target x.{**file** | **kafka**} [**queue** *queuename*]

{

set format *property=value* |

reset format [*property*] |

show format

}

Usage

Sets the format of the data that is written to the target.

Property	Input Value	Default
date=*format*	Date format	yyyy-MM-dd HH:mm:ss
decimal=*character*	Decimal character	**.** (period)
enotation=*notation*	Exponential notation	14
record=*record_format*	Format of the output records. Valid values are **sql**, **xml**, **json** or **avro**.	**xml**
timestamp=*format*	Timestamp format	yyyy-MM-ddTHH:mm:ss.fffffffff

Examples

target x.file set format record=sql

target x.kafka set format record=json

target x.kafka set format record=avro

Descriptions of output formats

The following are descriptions of the output formats provided by **format record**=*record_format*.

SQL record format

Every transaction in a SQL-formatted file is headed by a comment that includes the transaction sequence within the SQL file and a unique transaction ID. A comment line at the end of the SQL file has the number of lines in the file. For example, the following is a SQL file with one transaction. In this example the transaction id is `2-113319`. The file has nine lines.

```
/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 record format

The XML format is separated into operation and schema "types" for easier consumption. They are actually the same when viewed from an XSD perspective and are not distinct types. The template XML represents all possible attributes and elements. The individual XML represents the bare minimum output for each supported operation.

After startup, the first time that Post writes a change record for any given table, it first writes a schema record for that table. Each schema record contains the table name and details of interest for each columns. A schema record is written only once for each table during a Post run, unless there is a change to that schema, and then a new schema record is written. If Post stops and starts, schema records are written again, once for each table as Post receives a change record for it.

Schema record template

```
<?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>
```

Table 1: Explanation of schema template (* = optional)

Element	Attribute	Description
txn		Transaction metadata
	id	ID of current transaction
	oracleTxnId *	Oracle transaction ID
	commitTime*	Transaction commit timestamp
tbl		Table metadata
	name	Fully qualified name of the table
	utcOffset	UTC offset in the log
cmd		Operation metadata (In the case of a schema, there are no operations.)
	ops	Type of record generated for this table. For a schema, the value is **schema**.
schema		Column metadata
col		Metadata for a column (One of these elements appears for every record in the table.)
	name	Name of the column
	xmlType	XML data type

Element	Attribute	Description
	key	Key flag (true, false)
	nullable	Nullable flag
	length	Length of the column

Operation record template

```
<?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>
```

Table 2: Explanation of operation template (* = optional)

Element	Attribute	Description
txn		Transaction metadata for the operation
	id	ID of current transaction
	msgIdx	Index of current record in the transaction
	msgTot*	Total number of messages in transaction
	oracleTxnId *	Oracle transaction ID, taken from the System Change Number (SCN)
	commitTime*	Transaction commit timestamp
	userId *	User ID that performed the operation
tble		Table metadata
	name	Fully qualified table name

Element	Attribute	Description
cmd		Operation metadata
	ops	Operation type (insert, update, delete, truncate)
row		Metadata of the row that changed in the operation
	id	Oracle ROWID
col		Change data for a column (One of these elements appears for every changed column in the operation.)
	name	Column name with the after value for that column
lkup		Before image for use in update and delete operations
col		Before image of column (One of these elements appears for every changed column in the operation.)
	name	Column name with the before value or the key value (depending on the operation) for that column

NOTE: The **id** and **msgIdx** attributes together uniquely identify an operation.

Supported data types

See the SharePlex Release Notes for a chart that shows how Oracle data types are converted to XML.

Sample XML records

Source table

This is the table for which the sample operations are generated.

```
SQL> desc products

 Name                  Null?                Type

 PRODUCT_ID            NOT NULL             NUMBER

 DESCRIPTION                                VARCHAR2(600)

 PRICE                                      NUMBER
```

Source DML operations

```
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;
```

Schema record

```
<?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>
```

Insert record

```
<?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>
```

Update record

```
<?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>
```

Delete record

```
<?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>
```

Truncate record

```
<?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 record format

NOTE: This is default output with the exception that **target x.kafka set json indent=2** was used to make the output more readable.

This is the table for which the sample operations are generated.

```
SQL> desc products

 Name                  Null?                Type

 PRODUCT_ID            NOT NULL             NUMBER

 DESCRIPTION                                VARCHAR2(600)

 PRICE                                      NUMBER
```

Source DML operations

```
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;
```

Schema record

```
{
    "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"
       }
    }
}
```

Insert record

```
{
    "meta":{
       "op":"ins",
       "table":"BILL.PRODUCTS"
    },
    "data":{
       "PRODUCT_ID":230117,
       "DESCRIPTION":"Hamsberry  vintage  tee,  cherry",
       "PRICE":4099
    }
 }
```

Update record

```
{
    "meta":{
       "op":"upd",
       "table":"BILL.PRODUCTS"
    },
    "data":{
       "PRICE":3599
    },
    "key":{
       "PRODUCT_ID":230117,
       "DESCRIPTION":"Hamsberry  vintage  tee,  cherry",
       "PRICE":4099
    }
 }
```

Delete record

```
{
    "meta":{
       "op":"del",
       "table":"BILL.PRODUCTS"
    },
    "data":{
       "PRODUCT_ID":230117,
       "DESCRIPTION":"Hamsberry  vintage  tee,  cherry",
       "PRICE":3599
    }
 }
```

Truncate record

```
{
    "meta":{
       "op":"truncate",
       "table":"BILL.PRODUCTS"
    }
 }
```

AVRO record format

NOTE: Parameters for AVRO formatting can be set using command **target x.kafka set avro <propertyname>=<property value>** and reset using **target x.kafka reset avro <propertyname>**.

This is the table for which the sample operations are generated:

```
CREATE TABLE TEST.ORDERS (ID INT PRIMARY KEY, PRODUCT VARCHAR(50), QUANTITY INT,
PRICE BINARY_FLOAT);
```

Message Key Schema: This schema is optional in Kafka but mandatory for the initial SharePlex implementation. It includes key columns of a table as fields.

```
{

"fields": [

{

"name": "ID",

"type": "int"

}

],

"name": "TEST.ORDERS",

"type": "record"

}
```

Message Value Schema: This schema includes non-key columns of a table as fields.

```
{

"fields": [

{

"name": "PRODUCT",

"type": [

"null",

"string"

],

"default" = "null"

},

{

"name": "QUANTITY",
```

```
"type": [
"null",
"int"
],
"default" = "null"
},
{
"name": "PRICE",
"type": [
"null",
"float"
],
"default" = "null"
}
],
"name": "TEST.ORDERS",
"type": "record"
}
```

JMS category

Supported targets

JMS

Command options

target x.jms [**queue** *queuename*]

{

set jms *property=value* |

reset jms [*property*] |

show jms

}

Usage

Sets the properties of a JMS target.

Property	Input Value	Default
factory_class=*factory_class*	*Required* Fully qualified class name of the factory class. Sets the JNDI environmental property **java.naming.factory.initial** to specify the class name of the initial context factory for the provider.	None
provider_url=*url*	*Required* RMI URL with no object name component. This sets the JNDI environmental property **java.naming.provider.url** to specify the location of the registry that is being used as the initial context. Use the correct format depending your JMS Provider and type of URL. For example, if using LDAP your URL might be similar to the following: **ldap://***hostname.company.com/contextName* Ask your JMS Provider Administrator for the JMS Provider URL.	None
lib_location=*path*	*Required* Path to the directory where you installed the client library files.	None
destination={**queue** | **topic**}	Messaging domain. Valid values are **queue** (port-to-port) or **topic** (publisher-subscriber model).	**queue**
factory_name=*factory_name*	Name of a JNDI connection factory lookup. You can specify multiple names with a comma-separated list, for example: (jndi.name1, jndi.name2).	None
user=*user*	Name of the user that is attaching to JMS. If authentication is not required, omit this and the **password** option.	None
password=*password*	Password of the JMS user.	None
queuename=*JMS_topic_queuename*	Name of the JMS queue or topic.	**OpenTarget**
persistent={**yes** | **no**}	**yes** logs messages to disk storage as part of send operations. **no** prevents logging.	**yes**
session_transacted ={**yes** | **no**}	**no** directs Post to issue a JMS commit for every replicated message, making each one immediately visible and consumable. This is the default. **yes** directs Post to operate in a transactional manner. In this mode, Post issues a JMS commit (to make messages visible and consumable) at intervals based on the following:	**no**

Property	Input Value	Default
	• Issue a commit before the data is read-released from the Post queue, as controlled by the value set for the SP_OPX_READRELEASE_ INTERVAL parameter. • Issue a commit every JMS write, as controlled by the value set for the SP_OPX_CHECKPOINT_ FREQ parameter, until the SP_OPX_ READRELEASE_INTERVAL is reached.	
properties	Use this option if the JMS provider that you are using cannot consume messages that contain the default set of properties supplied by SharePlex. It enables you to remove or add properties. Supply the properties as a comma-delimited list. • To add a property, specify it as *name*=*value*. • To remove a SharePlex property, prefix the name with a dash. For example this string removes two SharePlex properties: -JMSXDeliveryCount,-JMSXGroupSeq.	None
client_id	Use this option if the JMS provider that you are using cannot consume messages that contain the default SharePlex client ID. Set this value to the client ID that your provider accepts.	None
commit_frequency	Use this option when Post is configured to post to a JMS server in *transactional* style (issue a JMS commit at intervals, rather than after every message as directed by the **session_transacted** property). This parameter specifies the interval between JMS commits. It works in conjunction with the SP_OPX_READRELEASE_ INTERVAL parameter. Valid values are 1 to any positive integer.	1500

Example

target x.jms set jms queuename=SharePlexJMS

JSON category

Supported targets

File (**format record=json**)

Kafka (**format record=json**)

Command options

target x.{file | **kafka** } [**queue** *queuename*]

{

set json *property=value* |

reset json [*property*] |

show json

}

Usage

Sets the properties of JSON output when **format record** =**json**.

Property	Input Value	Default

before	• **yes** includes the before image of the changed row. It produces a record similar to the following, with the before image at the end of the line:
	`{"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** excludes the before image of the changed row. It roduces a record similar to the following:
	`{"meta":{"op":"upd","table":"BILL.JSON"},"data":{"C2":"Bye"}}`
	`{"meta":{"op":"upd","table":"BILL.JSON"},"data":{"C1":"2"}}`
	DISCLAIMER: In case of the **Update** operations, use of "before" with "yes" value is required in order to view the key(s) for the updated record(s).
		yes

commit	• **yes** includes the commit records. It produces a record similar to the following: `{"meta":{"op":"commit","table":""}}` • **no** omits commit records. **IMPORTANT!** When **commit** is set to **yes**, there can only be one target topic.	**yes**
ddl	• **yes** includes ALTER TABLE commands. It produces a record similar to the following: `{"meta":{"op":"ddl","table":"BILL.JSON"},"sql":{"ddl":"alter table \"BILL\".\"JSON\""}` • **No** omits ALTER TABLE commands.	**yes**
eol	**yes** includes an end-of-line character at the end of the JSON document.	**yes**

Property	Input Value	Default
	no excludes an end-of-line character.	

indent	Controls the level of indentation of the records.
	• A value of **0** places the entire record on one line, as follows:
	<pre>{"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"}}</pre>• A value greater than 0 indents the metadata (if included) and data lines by the specified number of characters.
	It splits the record logically onto multiple lines, as in the following example (indent= 4) :
	<pre>{
	"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"
	}
	}</pre>
		0

meta	• **yes** includes the metadata section of the JSON record. It produces a record similar to the following:
	<pre>{"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":""}}</pre>
		yes

Property	Input Value	Default

	• **no** omits the metadata section and produces a record similar to the following:
	
	`{"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"}}`
	`{}`
	
	NOTE: To specify which metadata fields appear in the record, use the **set metadata** *property* command. See Metadata category.
	
	Example:
	
	target x.kafka set metadata time,userid,op,scn	

schema	• **yes** includes schema records.	**yes**

Property	Input Value	Default

	• **no** omits schema records.
	
	A schema record for objects in replication is produced when a new SharePlex configuration is activated or when a table has a DDL change.
	A value of yes (with indent=4) produces a record similar to the following:
	
	`{`
	`"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"`
	`}`
	
	`}`
	`}`	

Example

target x.kafka set json meta=no

kafka category

Supported targets

Kafka

Command options

target x.kafka [**queue** *queuename*]

{

set kafka *property=value* |

reset kafka [*property*] |

show kafka

}

Usage

Sets the properties of a Kafka target.

Property	Input Value	Default
broker=*broker*	Required. The host and port number of the Kafka broker, or a comma delimited list of multiple brokers. This list is the bootstrap into the Kafka cluster. So long as Post can connect to one of these brokers, it will discover any other brokers in the cluster.	**localhost:9092**
client_id=*ID*	Optional. A user-defined string that Post will send in each request to help trace calls.	None
compression.code={**none**, **gzip**, **snappy**}	Optional. Controls whether data is compressed in Kafka. Options are **none**, **gzip** or **snappy**.	None
partition={*number* | **rotate** | **rotate trans**| **messagekey**}	**Required. One of the following:** • A fixed partition number: Directs Post to post messages only to the specified partition number. For example, setting it to 0 directs Post to post only to partition 0. This option is suitable for use in testing or if the target has multiple channels of data posting to the same Kafka topic. • The keyword **rotate**: Directs Post to apply messages to all of the partitions of a topic in a round-robin fashion. The partition changes with each new message. For example if a topic has three partitions, the messages are posted to partitions 0,1,2,0,1,2, and so on in that order.	**0**

Property	Input Value	Default
	• The keyword **rotate trans**: This is similar to the **rotate** option, except that the partition is incremented with each transaction rather than with each message. For example, if a topic has three partitions, the messages are posted to partition 0 until the commit, then to partition 1 until the commit, and so on in that order. This option is suitable if you are replicating multiple tables to a single topic. It allows you to distribute data across several partitions, while still preserving all of the operations of a transaction together in a single partition. This enables a consumer that reads from a single partition to receive a stream of complete transactions. • The keyword **messagekey**: Directs Post to post messages to partitions. The Kafka topics are divided into several partitions. These partitions are selected based on the default partition hash function. The hash value is calculated based on messagekey. Use the messagekey partition to place all messages with the same key values in the same partition. **NOTES:** • The LOB and CLOB columns are not considered Kafka partition keys. • For a table without a primary key, unique key, composite key, or unique index, all columns (except LOB and CLOB columns) will be considered key columns. When performing an alter query on such a table, the DDL statement will be replicated to all partitions, and subsequent DML statements will be sent to specific partitions based on the existing columns. • If the replication table has no key defined, SharePlex will consider all table columns as Kafka messagekey. For non-key tables, it is recommended to use SharePlex user-defined keys. For more information, see the **Define a Unique Key: PostgreSQL to PostgreSQL** section in the SharePlex Admin Guide.	

Property	Input Value	Default

	• In cases where multiple tables are involved in replication, if we want a specific table to have a different partition type, while the remaining tables are partitioned based on the messagekey, we can define a named post queue for those specific tables.
	
	For example:
	
	`target x.kafka queue <queue_name> set kafka partition= {number/rotate/rotate trans}`
	
	For the rest of the tables, use the below command:
	
	`target x.kafka set kafka partition=messagekey`
	
	IMPORTANT:
	
	When partitioning is based on the messagekey, messages that do not contain key information will be mapped according to Kafka's internal hash function. These messages may include commit, schema, rollback, savepoint, and DDL statements.
	
	During replication, if the number of partitions is increased, the existing mapping of keys to partitions will no longer remain valid.
	
	For tables with a few columns serving as indexes and no other constraints defined, use those indexes as unique keys in the SharePlex config file.
	
	For example, the following table has a unique index defined on two columns: ID and NAME.
	
	`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);`
	
	In the SharePlex config file, define the index columns as a unique key.
	
	datasource:o.SID
	src.mytable !key(ID,NAME) host
	For more information, see the **Define a Unique Key: Oracle to Oracle** section in the SharePlex Admin Guide.	

Property	Input Value	Default
	For tables with no constraints or indexes defined, users can define unique keys during configuration in SharePlex. For a table that has a composite key, if any of the key values are modified, the modification message will be placed in the current partition, and subsequent messages may or may not be assigned to the same partition.	
request.required.acks=*value*	Optional. This is a Kafka client parameter. By default it is set to a value of -1, which means **all**. Consult the Kafka documentation about this subject, because **all** really means *all in-sync replicas*. This parameter can be used in conjunction with the **min.insync.replicas** broker parameter to tune behavior between availability and data consistency. **IMPORTANT:** It is possible for data to be lost between a Kafka producer (SharePlex in this case) and a Kafka cluster, depending on these settings.	**-1**
topic=*topic_name*	Required. The name of the target Kafka topic. This string may contain the special sequences **%o** or **%t**. The **%o** sequence is replaced by the owner name of the table that is being replicated. The **%t** sequence is replaced by the table name of the table that is being replicated. This feature may be used in conjunction with a Kafka server setting of **auto.create.topics.enabled** set to '**true**'. Also view your server settings for **default.replication.factor** and **num.partitions** because these are used as defaults when topics are auto created. **IMPORTANT!** If using multiple topics, you must also set the following properties with the **target** command: • The output must be in JSON. Set the **record** property of the **format** category to **json**: **target x.kafka set format record=json** • Commits must be disabled. Set the **commit** property of the **json** category to **no**: **target x.kafka set json commit=no**	**shareplex**

* To avoid latency, if Post detects no more incoming messages, it sends the packet to Kafka immediately without waiting for the threshold to be satisfied.

Example

```
sp_ctrl> target x.kafka set kafka broker=host1:9092,host2:9092,host3:9092
sp_ctrl> target x.kafkaset kafka topic=shareplex
```

Metadata category

Supported targets

Oracle replication targets

Oracle change-history targets

File

JMS

Kafka

Command options

target {**o.***SID* | **c.***SID* | **r**.*database* | **x.file | x.jms** | **x.kafka**} [**queue** *queuename*]

{

set metadata [*colname***:**]*property*[, ...] |

reset metadata |

show metadata

}

Usage

The **target** command with **set metadata** can be used to override the default metadata properties that are set for a database target, an XML or JSON file target, a JMS target, or a Kafka target. All of these target types provide metadata in the Post output.

- **Oracle change-history target:** If no **target** specification is made, Post updates the default metadata columns, which must exist in their default names before starting replication. In the following table, the default column names are listed if a property is included in the change-history target by default. Use **set metadata** only to add additional metadata or to assign a custom name to a column (use the *colname* option).
- For targets that support XML input (JMS, Kafka, file) the metadata is formatted per the XML schema. Certain metadata is included in the output by default, as shown in the following table, without the need to use **set metadata**. Use **set metadata** only to add additional metadata.
- For targets that support JSON input (Kafka), the **op** and **table** properties are included by default unless you set **json meta=no**.

For more information about configuring SharePlex to support a change history target database, see the SharePlex Administrator Guide.

To set multiple metadata properties, specify them as a comma-separated list.

Property	Value inserted by Post	Valid for
time	Time the operation was applied on the source, as supplied by Oracle. **IMPORTANT!** In Oracle version 11g and later, the time appears only in transactional boundary records, for example the start of a transaction. Based on the linear design of both the Oracle redo logs and the way that SharePlex replicates operations in transactional order, SharePlex applies the timetamp contained in the originating record to subsequent records until it encounters a record with a different timestamp. **Default column name**: SHAREPLEX_SOURCE_TIME **data type**: TIMESTAMP This property is included by default in XML output and change-history tables.	Oracle replication target Change-history target JSON, AVRO, and XML output
userid	User ID that performed the operation **Default column name**: SHAREPLEX_SOURCE_USERID **data type**: NUMBER This property is included by default in XML output and change-history tables.	Oracle replication target Change-history target JSON, AVRO, and XML output
op	Type of operation (INSERT, UPDATE, DELETE, TRUNCATE, DROP COLUMN, UPDATE BEFORE, UPDATE AFTER). For JMS implementations, this field is always added to the XML output as a field named **SPOps**. To add it to the JMS properties, issue an explicit **target** command with the **op** option. **Default column name**: SHAREPLEX_SOURCE_OPERATION **data type**: VARCHAR2 This property is included by default in XML, AVRO, and JSON output and change-history tables.	Oracle replication target Change-history target JSON, AVRO, and XML output
scn	Source SCN for when the operation was applied Default column name: SHAREPLEX_SOURCE_SCN data type: NUMBER This property is included by default in change-history tables.	Oracle replication target Change-history target JSON output
rowid	ROWID of the row that changed Default column name: SHAREPLEX_SOURCE_ROWID data type: ROWID This property is included by default in change-history tables.	Oracle replication target Change-history target JSON output

Property	Value inserted by Post	Valid for
trans	Transaction ID for the operation. For JMS implementations, this field is added to both the XML output and the JMS properties as a field named **SPTxnId**. Default column name: SHAREPLEX_SOURCE_TRANS data type: VARCHAR2 This property is included by default in change-history tables.	Oracle replication target Change-history target JSON output
seq	Order of the operation within the transaction Default column name: SHAREPLEX_OPERATION_SEQ data type: NUMBER This property is included by default in change-history tables.	Oracle replication target Change-history target JSON output
host	Name or IP address of the source host Default column name: SHAREPLEX_SOURCE_HOST data type: VARCHAR2 This property is included in change-history tables when metadata for the host property is set.	Oracle replication target Change-history target JSON output
posttime	The time that the operation was posted to the target.	JSON output
queue	Name of the SharePlex queue Default column name: SHAREPLEX_QUEUENAME data type: VARCHAR2 This property is included in change-history tables when metadata for the host property is set.	Oracle replication target Change-history target JSON output
source	User-defined source identifier that was set with the **set source** option of **target**. See Source category. Default column name: SHAREPLEX_SOURCE_ID data type: VARCHAR2 This property is included in change-history tables when metadata for the host property is set.	Oracle replication target Change-history target JSON output
changeid	Unique sequential ID of the operation Default column name: SHAREPLEX_CHANGE_ID data type: NUMBER This property is included in change-history tables when metadata for the host property is set.	Oracle replication target Change-history target
size	Number of operations in the transaction.	JSON output

Property	Value inserted by Post	Valid for
		Kafka
table	The name of the target table. The name of the target table will be added to the JMS properties as a field named **SPTblName**. This property is included by default in JSON output.	JMS JSON, AVRO, and XML output
idx	The index of this operation within the transaction. This will be added to the JMS properties as a field named **SPMsgIdx**. It will be formatted in the JMS properties as *m*/*n* where *n* is the total number of operations in the transaction and *m* is the operation number within the transaction.	JMS

JMS and Kafka set metadata example

The following sets all Post processes to add the IP address or hostname of the source, name of the SharePlex queue, and the number of operations in the transaction to the output in the JMS target.

target x.jms set metadata host, queue, size

or...

target x.kafka set metadata host, queue, size

Oracle set metadata example

The following example sets the Post process associated with the **myqueue1** queue to add the time of the operation and the userid that executed it to the **Timestamp** and **User** columns, both of which are user-defined names that are case-sensitive.

target c.targSID queue myqueue1 set metadata time:"Timestamp", userid:"User"

Resources category

Supported targets

All targets

Command options

target {**o.** | **r.** | **x.** | **c.**} *target* [**queue** *queuename*]

{

set resources *property=value* |

reset resources [*property*] |

show resources [**for** *datastore*]

}

Usage

Sets parameters that affect resources on the target system.

Property	Input Value	Default
commit_frequency=*number_of_operations*	Specifies a maximum number of operations after which Post issues a commit. Can be any integer greater than 1. Can be used to divide a large transaction into two or more smaller ones, each containing, at most, the specified *number_of_operations*. This option can work around resource limits that affect large transactions, such as the number of row locks permitted per transaction. **NOTE:** The "commit_frequency" property is supported only for the JMS target.	50000
max_active_statements=*number_of_cursors*	Specifies the number of concurrent active SQL statements that Post can process to a target database in one session. Can be any integer up to, and including, the maximum number of active statements permitted by the database or client driver. It is only used by Post when the SP_OPX_SQL_CACHE_DISABLE parameter is set to 0 (enabled).	16

Examples

target r.mydb queue q1 set resources commit_frequency=10000

target r.mydb queue q1 set resources max_active_statements=10

Rule | Filter category

Supported targets

Oracle change-history target

Command options

target c.*SID* [**queue** *queuename*]

table *tablename*

{

set {**rule** | **filter**} { '*column_condition*' | !*filename* }

reset {**rule** | **filter**}

show {**rule** | **filter**}

}

Usage

Specifies a conditional statement that applies a tracking rule or filter to a specific table. Applies to UPDATE operations. Applies conditional logic that qualifies the data that you want to track to a target table, while excluding

data that is not of interest. There can be one **rule** or **filter** per target table, but you can combine nested expressions with parentheses and the AND, OR, and NOT logical connectives to create a wide variety of conditions.

Syntax element	Description
rule | **filter**	• **rule** directs Post to *insert* a row *only* if the UPDATE operation matches the *column_condition*. If the UPDATE does not match the column condition, the row is discarded. • **filter** directs Post to *insert* a row *only* if the UPDATE does not match the *column_condition*. If the UPDATE does match the column condition, the row is discarded.
table *tablename*	The name of the target table for which you are setting the rule or filter.
column_condition	The conditional logic that must be satisfied by the UPDATE operation. *column_condition* represents the entire conditional statement, which must be enclosed within single quotes. A column condition can be one or any combination of the following expressions, each enclosed within parentheses: (*column* **is changed**) (*column* **is not changed**) ([*column1*, *column2*, ...] **matches change_list**) ([*column1*, *column2*, ...] **contains change_list**) *Where*: • *column* is the name of a target column. Use a comma to separate column names in a list, and enclose a list within brackets. • **is changed** means that the specified column(s) are changed in the UPDATE operation. • **is not changed** means that the specified column(s) are not changed in the UPDATE operation. • **change_list** is a logical representation of the source columns that changed. • **matches change_list** is satisfied when the condition is an exact match to the change list (no fewer columns than the change list, no more columns than the change list, same names). • **contains change_list** is satisfied when the condition contains all of the columns in the change list. The change list can be a subset of the columns in the condition, but it cannot contain more columns than the change list. Names must match.
!*filename*	Enables you to store a large column condition in a text file, and then specify the file, rather than the entire *column condition* syntax, when you run the **target** command.

NOTE: Batch and direct load operations are not supported by the **rule**/**filter** feature.

Examples

These commands show some examples of how to set different rules by combining expressions:

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)'

The following table shows the different ways that **rule** and **filter** work to include or discard rows.

Rule	Updated columns	Row inserted?
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
Filter	**Updated columns**	**Row inserted?**
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 category

Supported targets

Oracle change-history target

Command options

target c.*SID* [**queue** *queuename*]

{

set source *sourceID* |

reset source |

show source

}

Usage

Sets the user-defined source identifier for the SHAREPLEX_SOURCE_ID metadata column. Use **target** with the **set metadata source** option to configure Post to populate this column. Specify any single alphanumeric string.

Example

The following shows the use of **set source** to support a configuration where the data from multiple source databases is being replicated to the same target database. This target configuration causes each Post process to update the SHAREPLEX_SOURCE_ID column with a different source ID, depending on the source database, to differentiate the data in the target database. This example uses four Post processes.

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

The Post process for queue “source718” will update column SHAREPLEX_SOURCE_ID with the value "east" for every insert or update to a table in its stream. The Post process for queue "source92" will update column SHAREPLEX_SOURCE_ID with the value "south", and so on for the other two Post processes.

SQL category

Supported targets

File (**format record**=**sql**)

Command options

target x.file [**queue** *queuename*]

{

set sql *property=value* |

reset sql [*property*] |

show sql

}

Usage

Sets the properties of SQL output when **formatrecord**=**sql**.

Property	Input Value	Default
add_rownum={**yes** | **no**}	**yes** to include or **no** to exclude row numbers	**yes**
begin_transaction={**yes** | **no**}	**yes** to include or **no** to exclude begin transaction records	**no**
comment=*character*	Character that marks a comment	**--**
concatenate=*character*	Character that concatenates strings	**||**
end_transaction={**yes** | **no**}	**yes** to include or **no** to exclude end transaction records	**no**
legacy={**yes** | **no**}	*Required in SharePlex 8.6.* Use legacy SQL date and timestamp format of: `MMDDYYYYHH24MISS` and `MMDDYYYYHH24MISS.FFFFFF`(`yes`/`no`)	**no**
name_delimiter=*character*	Character that delimits SID, table, owner, column names. This value overrides the value that is returned when Post queries the ODBC driver for the setting of SQL_DELIMITER.	**none**
record_terminator=*character*	Character that terminates the SQL	**;**

Example

target x.file set sql comment=#

Trace Capture

Use the **trace capture** command to output Capture processing and performance statistics to a trace file.

The Capture trace includes:

- Details of the trace itself
- Time that Capture spent on various tasks overall and per thread

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues for:	source system
Related commands:	**trace read**, **trace post**

Syntax

Basic command	Command options	Remote options
trace capture	[*minutes*] [**for** *datasource*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
minutes	The number of minutes for which you want to run the trace. The default is 15 minutes.
for *datasource*	The datasource for which you want to run the Capture trace. Only required if running multiple Captures in the same SharePlex instance. Must follow *minutes*, if used.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Output

When the trace is complete, the process writes the trace data to a file in the **log** subdirectory of the variable-data directory in the following format:

*process_id*_trace_*time*

For example:

orcl_ocap_trace_Sep30_15_24_2014

where: `orcl` is the datasource and `ocap` is the name for the Capture process.

Example

The following command runs the trace for datasource **myora** for one minute.

trace capture 1 for myora

This command produces trace output similar to the following:

```
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

Use the trace post command to output Post processing and performance statistics to a trace file.

The Post trace includes:

- Details of the trace itself
- Statistics on transaction activity
- Detail statistics per operation type

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues for:	target system
Related commands:	**trace capture**, **trace read**

Syntax

Basic command	Command options	Remote options
trace post	[*minutes*] [**for** *datasource-datadest*] [**queue** *queuename*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
minutes	The number of minutes for which you want to run the trace. The default is 15 minutes.
for *datasource-datadest*	Constrains the trace to a specific Post process, as identified by the source and target datastores. • Required if running multiple Post processes in the same SharePlex instance (replication from different sources) or if running multiple Post processes to different target databases. Must follow *minutes*, if used. • *datasource* is expressed as **o**.*SID* where *SID* is an ORACLE_SID.

Component	Description
	• *datadest* is expressed as one of the following, depending on the target: **o**.*ORACLE_SID* **r.***database_name* **x.kafka** **x.jms** **x.file**
queue *queuename*	Constrains the trace to the Post process that is associated with the specified named queue.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Output

When the trace is complete, the process writes the trace data to a file in the **log** subdirectory of the variable-data directory in the following format:

*process_id*_trace_*time*

For example:

```
orcl_orcl2_opo_trace_Feb_5_17_24_2014
```

where: `orcl` is the source datasource and `orcl2` is the target datasource and `opo` is the name for the Post process.

Example

The following command runs the trace for datasources **myora** and **myora2** for one minute.

trace post 1 for myora-myora2

This command produces trace output similar to the following:

```
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

Use the **trace read** command to output Read processing and performance statistics to a trace file.

The Read trace includes:

- Details about queries that were made to obtain the values of keys
- Statistics to show the time that Read spend on various processing tasks

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues for:	source system
Related commands:	**trace capture**, **trace post**

Syntax

Basic command	Command options	Remote options
trace read	[*minutes*] [**for** *datasource*]	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
minutes	The number of minutes for which you want to run the trace. The default is 15 minutes.
for *datasource*	The datasource for which you want to run the trace. Only required if running multiple Capture and Read processes in the same SharePlex instance. Must follow *minutes*, if used.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Output

When the trace is complete, the process writes the trace data to a file in the **log** subdirectory of the variable-data directory in the following format:

*process_id*_trace_*time*

For example:

orcl_ord_trace_Sep30_13_07_2014

where: `orcl` is the datasource and `ord` is the name for the Read process.

Example

The following command runs the trace for datasource **myora** for one minute.

trace read 1 for myora

This command produces trace output similar to the following:

```
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

Use the **truncate log** command to truncate (remove all data from) the Event Log and trace log files in the **log** sub-directory in the SharePlex variable-data directory. These logs accumulate data and eventually can consume a large amount of disk space. The **truncate log** command allows you to truncate the logs after the data has outlived its usefulness. This command does not affect replication. You can issue it while replication is active and data is in the queues.

Truncating logs when sp_eventmon is running

When the **sp_eventmon** monitoring script is running, issue the **truncate log** command and then delete the *sp_cop_name*.**mrk** file, where *sp_cop_name* is the value used in the -s argument when the script was run. This file is in the **util** sub-directory of the SharePlex product directory.

IMPORTANT: The **truncate log** command deletes *all* entries from the logs. Quest Technical Support refers to the information in the logs when you make a support call, so try to retain the data as long as possible.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Administrator (1)
Issued for:	source or target system
Related commands:	none

Syntax

Basic command	Remote options
truncate log	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Typemap

Use the **typemap** command to show data type mappings. The data type map is used to map the data type of a column on the source to a corresponding data type on the target. SharePlex uses the data type map when replicating DDL and when creating tables on the target. For example, Post uses the data type map corresponding to the source-target combination to determine the data type of the target column when it replicates ALTER TABLE...ADD COLUMN DDL operations.

The data type map contains a list of rules for the source-target combination. Each rule has a pattern that Post uses to match against the source data type. If the pattern matches, the corresponding pattern is used to construct the target data type.

Example rules

Source	Target	
CHAR(n)	VARCHAR(n)	Any column defined as CHAR in DDL from the source is mapped to VARCHAR on the target with the same length (n).
VARCHAR (4000:)	CLOB	Any column defined as VARCHAR with a length greater than or equal to 4000 is mapped to a CLOB datatype on the target.

To show data type map: Use the **show** option to view the current typemaps.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	target system
Related commands:	**connection, target**

Syntax

Basic command	Command options	Remote options
typemap show	[*source-target*]	Not available

Syntax description

Component	Description
show	Use this option to display part or all of the current typemap information. Examples: **typemap show** displays all typemaps **typemap show oracle-sqlserver** displays the Oracle to SQL Server typemap.
source-target	*source-target* specifies the source-target map to show.

Usage

Use the **usage** command to view the syntax for a SharePlex command. You can enter the entire command or just the first few keywords. For example, type **usage compare** to view syntax for both the **compare using** and **compare** commands.

To view syntax plus an explanation of a command or group of commands, use the help command. There are no [**on** *host*] options for this command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**help**

Syntax

Basic command	Remote options
usage *commandname*	Not available

Syntax description

Component	Description
commandname	The command for which you want to view the syntax. **Example**: `sp_ctrl(sysA)` **> usage stop**

Verify Config

The **verify config** command is intended for use as a preventive measure to avoid certain activation and replication problems. It is intended to be used to test activation to ensure that it will complete successfully.

The **verify config** command verifies tables and sequences only.

This command can be used to:

- Verify the syntax of the entries in the configuration file.
- Report an error if the source object is not supported for replication by SharePlex.
- Report if a host name specified in a route is unreachable.
- Report if there are duplicate specifications for a single object.
- Report if an object specification will be skipped and the reason why.

What the verify config command does not support

The **verify config** command does not:

- Verify activation time.
- Verify target objects or the target SID.

Verifying added or changed objects in an active configuration

To verify objects that you want to add to an active configuration or objects that you want to change (such as routing changes), it is suggested that you copy and modify the active configuration and then run the **verify command** against that copy.

Using with partitioned replication

You can use the verify config command for configurations containing tables that use partitioned replication. While SharePlex supports the use of **verify config** with partitioned replication, it does not include any additional functionality for verifying partitions or their definition.

Viewing the results of the verification

The **verify config** command retains control of the sp_ctrl interface until the verification is completed.

The command will read the entire config file, logging errors and duplicates, and expanding objects falling under wildcards.

The results of the verify are displayed to the screen within **sp_ctrl**.

If you would like to view detailed results you may:

- Issue the verify **config** command in **sp_ctrl** using the **detail** option
- Navigate to the results file directly using the path displayed to the screen after issuing the **verify config** command.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**activate config, copy config, create config, edit config**

Syntax

Basic command	Command options	Remote options
verify config *filename*	**detail**	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	*filename* is the name of the configuration to be verified.
detail	This option will display a greater level of detail to the screen. **Example**: `sp_ctrl(sysA)` **> verify config myconfig detail** In this example, the **myconfig** file will be verified and the results will be displayed with a higher level of detail.

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax.

Option	Description
	Example: `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Version

Use the version command to view the version number of the SharePlex software on a system.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	none

Syntax

Basic command	Remote options
version	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login,

Option	Description
	password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

View Config

Use the **view config** command to view the contents of an active or inactive configuration file from within the **sp_ctrl** interface.

Use it when:

- You suspect that an incorrectly written configuration could be causing activation or replication problems.
- You want to know which objects are being replicated and how the routes are configured.

NOTE: To edit a configuration, use the **edit config** command. To change an active configuration, it is recommended that you copy it with the **copy config** command and then edit the copy.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**activate config, copy config, deactivate config, edit config, list config, remove config, rename config, verify config**

Syntax

Basic command	Remote options
view config *filename*	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
filename	The name of the configuration file that you want to view. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysB)` **> view config Sales**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

View Log options

Use the **view log** options command to view the default values for the **show log** command.

Example display:

```
sp_ctrl(sysA)> view log options
Log current display options:
     direction = reverse
     maxlines = 50
     lpp = 30
```

These parameters can be changed with the **set log** command, or with options within the **show log** command itself.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Viewer (3)
Issued for:	source or target system
Related commands:	**set log, show log**

Syntax

Basic command	Remote options
view log options	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login

Option	Description
	credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

View Partitions

Use the **view partitions** command to view the row partitions in one partition scheme or all partition schemes in a horizontally partitioned replication configuration.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported sources:	Oracle
Supported targets:	All
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	Add partition, Drop partition, Drop partition scheme, Modify partition

Syntax

Basic command	Remote options
view partitions for {*scheme_name* | **all**}	Not available

Syntax description

Component	Description
scheme_name	Show the row partitions for the specified partition scheme.
all	Show all row partitions, grouped according to the names of their partition schemes.

Examples

```
sp_ctrl> view partitions for scheme1
```

```
sp_ctrl> view partitions all
```

2

SharePlex Parameters for Oracle

SharePlex parameters control and tune various aspects of replication.

For instructions on setting parameters, see the SharePlex Administration Guide.

Contents

Descriptions of User-Configurable Parameters

Deprecated Parameters

Descriptions of User-Configurable Parameters

This section describes the **user-configurable** SharePlex parameters. Parameters not documented in this chapter are internal parameters that should only be modified under guidance of a Quest developer or Technical Support representative.

Parameters are grouped as follows:

- Configuration activation parameters
- Capture Parameters
- Read parameters
- Export parameters
- Import parameters
- Oracle Poster parameters
- Open Target Post parameters
- Queue parameters
- sp_cop parameters
- Log parameters
- SNMP parameters
- System parameters
- Compare/Repair Parameters
- Copy/Append command parameters
- Analyze config command parameters
- PostgreSQL Parameters

Configuration activation parameters

These parameters are used by the SharePlex activation process. They take effect at the time that you activate a configuration file.

SP_OCF_HASH_BY_BLOCK

This parameter controls whether the hash algorithm used in horizontally partitioned replication is based on the rowid or on the block where the row resides. The default is 0, which uses the hash based on rowid. Using a hash based on the block may improve the performance of the Post process when processing tables that are using horizontally partitioned replication. To enable a hash based on the block, set this parameter to 1 and then reactivate the configuration file.

Default: 0 (disabled)

Range of valid values: 0 or 1 (enabled block-based hash)

Takes effect: for the next activation

SP_OCF_LOCK_WAIT_TIME

This parameter tells SharePlex how long to wait before failing activation on a particular table. Since the table must be locked for activation, the logic is to retry the lock for a designated period of time, and this designated period of time is controlled by the SP_OCF_LOCK_WAIT_TIME parameter.

Default: 5 minutes

Range of valid values: Any positive integer

Takes effect: Immediately

SP_OCF_THREAD_COUNT

This parameter controls the default number of threads that SharePlex generates during configuration activation. It might be necessary to experiment with the number of threads to determine the optimal performance level. As an example, for a 32-CPU machine with a large disk array, 10 or more threads could show improved performance. The value for the thread count is independent of the number of tables to be analyzed.

Default: 3 threads

Range of valid values: 1 to 32

Takes effect: Immediately

Capture Parameters

These parameters are used by the SharePlex Capture process.

SP_OCT_ALLOW_DP_DDL

This parameter can be enabled if SharePlex fails to replicate DDL operations that occur when running an Oracle Data Pump export/import. Occasionally, SharePlex identifies DDL in a Data Pump load as recursive DDL that should be ignored. This parameter directs SharePlex to capture that DDL.

A setting of 1 enables this parameter. After the load is finished, set this parameter back to 0 and then restart Capture.

Default: 0

Range of valid values: 0 or 1 (flag)

Takes effect: upon restart of the Capture process

SP_OCT_ARCH_LOC

This parameter defines the path to the archive logs. When the redo logs wrap, SharePlex looks for the archive log in Oracle's archive log list. If SharePlex cannot find the archive log there, it looks in the directory or directories specified by this parameter. It also searches the sub-directories. The default is the **/home** directory. If the path for the archive logs is different, specify the correct full path name with this parameter. You can specify more than one directory path if you separate them with semicolons (;) as shown in the following example. In this example, Capture will search under both /**disk1/log** and /**disk2/log** paths to find the logs.

`sp_ctrl`**> set param SP_OCT_ARCH_LOC /disk1/log;/disk2/log**

To specify the directory path for SharePlex for RAC, use the following:

`sp_ctrl`> set param SP_OCT_ARCH_LOC !;<node1_oracle_sid>;<node1_arch_absolutepath>;<node2_oracle_sid>;<node_arch_absolute_path>

Note that the non-RAC format for path names allows a list of directory path names separated by a semicolon. The list can be any length desired up to the limit of 1023 bytes. However, the RAC format, which is !; followed by a list of sid;pathname pairs, does not allow the pathname to be a list of path names as the non-RAC format does. Each SID must be followed by exactly **one** directory pathname. If you want to specify more than one pathname for a single instance SID, you must put in two pairs of entries in which each of the entries has the same SID. Thus, you cannot specify

!;sid1;path2;path2;path3;sid2;path4

Rather, the proper format for the preceding example is

!;sid1;path2;sid1;path2;sid1;path3;sid2;path4

Default: empty string

Range of valid values: full directory path to the restored archive logs

Takes effect: immediately.

SP_OCT_ASM_CACHE_AGE

This parameter controls how long the current ASM cache is kept, before before a re-query of the ASM blocks is performed to update it.

Default: 40

Range of valid values: Any positive integer

Takes effect: immediately

SP_OCT_ASM_ECACHE_SIZE

The number of elements in the buffer to cache extents. Extents are the units of file storage in Oracle ASM. Increasing this number increases memory usage.

Default: 20

Range of valid values: Any value greater than 9

Takes effect: upon restart of the Capture process

SP_OCT_ASM_FLIST_SIZE

Maximum number of ASM files that will be read simultaneously. Increasing this number increases memory usage.

Default: 10

Range of valid values: Any value greater than 4

Takes effect: upon restart of the Capture process

SP_OCT_ASM_MULTI_OCI_BLOCK_SIZE

This parameter controls the size of the buffer that is used by each Capture thread when SP_OCT_ASM_MULTI_OCI is set to enable multi-threaded capture on Exadata systems. This parameter should be left to its default. Capture automatically adjusts its buffer size to the value of the AU_SIZE parameter that is set for the disk group where the logs reside. This is the recommended buffer size for best performance.

Default: 0

Range of valid values: Any positive integer

Takes effect: upon restart of the Capture process

SP_OCT_ASM_MULTI_OCI

This parameter controls the number of threads that Capture uses to read the redo logs. To use this parameter, you must also set SP_OCT_OLOG_USE_OCI to 1.

The value for this parameter must be set to at least 2 but no more than the number of disks in the redo log disk group.

A large number of threads is not required, and performance actually diminishes with too many threads. The more threads, the more memory Capture requires. Start with a small number of threads and monitor performance, then add threads if needed until you obtain an ideal balance between performance gain and memory usage.

Default: 0 (single threaded capture)

Range of valid values: Minimum value is 2, maximum value is the number of disks in the ASM disk group that stores the redo logs.

Takes effect: upon restart of the Capture process

SP_OCT_ASM_SLIST_SIZE

Maximum number of sessions that will be accessing files simultaneously.

Default: 10

Range of valid values: Any value greater than 4

Takes effect: upon restart of the Capture process

SP_OCT_ASM_SUPPORT

This parameter enables and disables support for redo and archive logs on ASM supported platforms. It is enabled automatically if the SharePlex database connection is configured to connect to ASM.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: upon restart of the Capture process

SP_OCT_AUTOADD_ENABLE

This parameter is a global parameter that enables newly created objects to be added automatically to an active replication configuration. By default it applies to tables and indexes. To auto-add other object types, additional parameter settings are required.

The functionality works as follows:

Auto-add of the following objects is enabled by default when you enable SP_OCT_AUTOADD_ENABLE:

- A table named in a CREATE TABLE operation is automatically added to replication if the name matches a wildcard specification in the active configuration file.
- An index created with a CREATE INDEX operation is automatically added to replication if the table on which it was created is specified in the configuration file.
- A table named in a CREATE TABLE AS SELECT operation is automatically added to replication if the name matches a wildcard specification in the active configuration file. SharePlex creates the new table on the target, followed by the replication of the data in the source table.
- ALTER and DROP on the added objects.

The following objects must be explicitly enabled for auto-add support in addition to enabling SP_OCT_AUTOADD_ENABLE:

The supporting table for a materialized view named in a CREATE MATERIALIZED VIEW operation is automatically added to replication if the name of the materialized view matches a wildcard in the active configuration file and the SP_OCT_AUTOADD_MVIEW parameter is set to 1. The SP_SYS_TARGET_COMPATIBILITY parameter must be set to at least 8.6.2 to support auto-add of new materialized views.

A sequence named in a CREATE SEQUENCE operation is automatically added to replication if the name matches a wildcard in the active configuration file and the SP_OCT_AUTOADD_SEQ parameter is set to 1. The SP_SYS_TARGET_COMPATIBILITY parameter must be set to at least 8.6.3 to support auto-add of new sequences.

NOTE: To replicate changes to the sequence, the SP_OCT_REPLICATE_SEQUENCES parameter must be set to 1 (the default).

Make sure DDL replication is fully enabled by checking that the parameter SP_OCT_REPLICATE_DDL is set to 1 or 3.

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OCT_AUTOADD_MVIEW

This parameter determines whether SharePlex automatically adds materialized views to replication when they are created after activation. When enabled, it causes SharePlex to add the underlying table of a new materialized view to replication if the name of the materialized view satisfies a wildcard in the active configuration file. SharePlex converts the CREATE MATERIALIZED VIEW into a CREATE TABLE statement, posts the CREATE TABLE to the target, and replicates the DML that populates the view. The table is maintained by replication through future supported DDL and DML changes.

To auto-add materialized views, both this parameter and the SP_OCT_AUTOADD_ENABLE parameter must be set to 1, and the version of SharePlex on the target must be at least 8.6.2.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: Process restart

SP_OCT_AUTOADD_SEQ

This parameter determines whether SharePlex automatically adds sequences to replication when they are created after activation. When enabled, it causes SharePlex to replicate the CREATE statement if the name of the sequence satisfies a wildcard in the active configuration file. SharePlex then maintains the object on the target throughout future DDL and DML changes. To auto-add sequences, both this parameter and the SP_OCT_AUTOADD_ ENABLE parameter must be set to 1, and the version of SharePlex on the target must be at least 8.6.3.

IMPORTANT! To replicate sequences, the supplemental logging of primary and unique keys must be enabled at the database level, or you must enable supplemental logging for primary keys on the **sys.seq$** table.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: Process restart

SP_OCT_CHECKPOINT_LOG

Sometimes, the Capture process does not checkpoint on a regular basis. Checkpointing saves the state of the process in case it is needed for recovery. If the Capture process terminates for some reason and the redo logs wrap, SharePlex attempts to recover to a checkpoint that no longer exists. SP_OCT_CHECKPOINT_LOG ensures that the checkpointing occurs before the logs switch.

The checkpoint is triggered when Capture lags a specified number of logs behind Oracle. For example, with the default of 2, Capture does a checkpoint when it falls 2 or more logs behind Oracle.

The range of permissible values for this parameter is from 2 (the default) to a value equal to the number of logs you are using. A value of 0 disables this feature. This parameter is useful in environments where frequent log switches can cause a switch to occur before SharePlex's internal checkpoint mechanism can be triggered.

Default: 2 logs

Range of valid values: 2 to the number of redo logs

Takes effect: immediately

SP_OCT_CHECKPOINT_TIME

This parameter works in conjunction with the SP_OCT_CHECKPOINT_FREQ parameter. It defines the time delay, in seconds, before the Capture process checkpoints. If the value set for this parameter is reached before the value set for SP_OCT_CHECKPOINT_FREQ, it triggers the checkpoint. (Checkpointing saves the state of the process in case it is needed for failure recovery.)

Default: 120 seconds

Range of valid values: any positive integer

Takes effect: immediately

SP_OCT_CK_LOC_FIRST

This parameter causes the Capture process to first search for an archived redo log in the location specified by SP_OCT_ARCH_LOC. If SP_OCT_ARCH_LOC is an empty string, Capture will first search in the Archive Log Destination as configured in Oracle.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: Process restart

SP_OCT_DATA_OBJ_CACHE_SIZE

This parameter specifies the number of entries/elements used in the Data Object ID to Table Object ID mapping cache. For certain operations, such as DLOADs or LOB data, the redo-record for such operations contains only the Data Object ID. The Capture process needs to map the Data Object ID to the Table Object ID to see if the operation is of interest. The Data Object ID to Table Object ID cache provides a look-up solution; otherwise, the Capture processmust query Oracle to map the Data Object ID to Table Object ID.

Default: 50

Range of valid values: any positive integer

Takes effect: immediately

SP_OCT_DATE_ MSG

This parameter can be set so that the Capture process prints a warning message to the Event Log and the Capture Log when it detects an invalid date column. A setting of 0 disables it, and a setting of 1 activates it. The error message generated by Capture is:

```
 Invalid DATE format detected in record with rowid=rowid, on obj object_id. See
capture log for detail.
```

Default: 0 (do not print messages)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OCT_DDL_LOGGING

This parameter controls the Capture logging levels for DDL activity. The logging of DDLs to the DDL log is not dependent upon the SP_OCT_REPLICATE_ALL_DDL setting. Even when SP_OCT_REPLICATE_ALL_DDL is 0, the DDL can still be logged. The following settings are available:

0 = no logging,

1 = replicated DDL only

2 = all DDL

Default: 2

Range of valid values: 0, 1, 2

Takes effect: immediately

SP_OCT_DDL_UPDATE_CONFIG

This parameter controls whether SharePlex updates the table name in the configuration file when processing ALTER TABLE RENAME. By default, SharePlex updates the table name when it processes this DDL operation. To prevent updates the configuration file, disable this parameter.

Default: 1 (enabled)

Range of valid values: 0 or 1

Takes effect: when Capture is restarted

SP_OCT_DEF parameters

The following parameters can be set so that SharePlex corrects the format of dates and times if they were incorrectly entered by a user or application and bypassed the database's validity check. These parameters take effect as soon as they are activated.

SP_OCT_DEF_MONTH	range of values is from 1 - 12
SP_OCT_DEF_DAY	range of values is from 1 - 31
SP_OCT_DEF_YEAR	range of values is from 1987 - 9999
SP_OCT_DEF_HOUR	range of values is from 0 - 23
SP_OCT_DEF_MIN	range of values is from 0 - 59
SP_OCT_DEF_SEC	range of values is from 0 – 59

SP_OCT_DENIED_USERID

This parameter can be used to specify an Oracle userid for which all DML and DDL transactions should be ignored or filtered by the Capture process.

IMPORTANT: Ignoring transactions on the source machine may lead to an out-of-sync condition.

SharePlex does not verify that the specified userid exists.

Default: 0x00000000

Range of valid values: integers (any valid Oracle userid)

Takes effect: immediately

SP_OCT_DLOAD_MAX_COLS_PER_MSG

This parameter controls the maximum size of a DLOAD message. The number of rows in a message is determined by dividing this parameter by the number of columns in each row. For example: If a table has 99 columns and there is a DLOAD with 90 rows and this parameter is set to 5000 then it would split the message into 2 DLOAD messages with 50 rows of 99 columns for a total of 4950 columns in the first message the rest in the second message.

Default: 10000

Range of valid values: any value greater than 99

Takes effect: immediately

SP_OCT_ENABLE_LOBMAP

This parameter controls whether or not SharePlex uses a LOB map when replicating tables that contain out-of-row LOB columns. The LOB map is used by the Capture process to map LOBIDs and rows when PK/UK logging is not enabled. LOB mapping is enabled by default. The SHAREPLEX_LOBMAP table stores these mappings.Transactions with numerous LOB operations can slow down Capture because it needs to maintain and refer to the mappings. If PK/UK logging is enabled on the database, you can disable LOB mapping by setting this parameter to 0.

Default: 1 (on)

Range of valid values: 0 or 1

Takes effect: Process restart

SP_OCT_INCLUDE_UNCHANGED_COL

This parameter controls whether or not unchanged columns are included in the after image of an UPDATE operation. By default, SharePlex only includes the changed values in the after image.

Default: 0 (off)

Range of valid values: 0 or 1

Takes effect: immediately

SP_OCT_INSERT_INCLUDE_NULLS

This parameter controls whether or not columns with NULL values are replicated for an INSERT statement. By default, SharePlex does not replicate null values. For example, a statement like INSERT INTO mytable (col1, col2, col3) VALUEs ('red', 'green', 2), where mytable has six columns, causes SharePlex to replicate only the explicit 'red", 'green', and 2 values, but not the implicit NULL values for the other three columns. If SP_OCT_INSERT_INCLUDE_NULLS is set to 1, SharePlex replicates 'red', 'green', 2, ' ', ' ', ' '.

Default: 0 (off)

Range of valid values: 0 or 1

Takes effect: immediately

SP_OCT_LOB_BUFFER_SIZE

This parameter controls the size of Capture's LOB buffers, which must be maintained until Capture can assemble the related transaction information. If SharePlex is unable to determine the transaction, it uses a special sub-queue outside of the ones containing the transactions. The headers for these sub-queues require more shared memory, which can require an increase in the SP_OCT_LOB_BUFFER_SIZE parameter.

The larger the LOB buffers, the more likely that system memory will run out, preventing buffering for subsequent LOB operations until memory becomes available again.

Therefore, you might need to increase the SP_QUE_Q_SHMSIZE parameter in conjunction with increasing the SP_OCT_LOB_BUFFER_SIZE parameter. Generally, only the LOB data for VARRAYs is buffered, so this parameter generally only impacts transactions involving VARRAYs.

Default: 5 MB of memory per LOB

Range of valid values: any positive integer

Takes effect: when Capture is restarted

SP_OCT_LOG_FILESIZE

This parameter sets the size of the Capture debug log file.

Default: 50000000 bytes

Range of valid values: any value greater than 9999 bytes

Takes effect: when Capture is restarted

SP_OCT_LOG_MEMBER

The SP_OCT_LOG_MEMBER parameter is used to augment queries of the data dictionary to specify redo log locations. You can use this parameter to specify exactly what log file you want to operate on.

Default: % (percent symbol)

Range of valid values: any correctly formed file name specification with wildcard characters

Takes effect: when Capture is restarted

SP_OCT_LOG_NUMFILES

This parameter controls the number of Capture debug log files that are allowed before the oldest one is deleted and a new one is created.

Default: 3

Range of valid values: any value greater than 1

Takes effect: when Capture is restarted

SP_OCT_LOG_READ_SIZE

The Capture process reads multiple redo-log file blocks in one pass. This parameter controls the number of blocks read at a time, enabling you to adjust the value according to the system's configuration and whether or not Capture is falling behind the pace of Oracle's processing. Keep in mind that when the value is higher than necessary, it incurs more system overhead.

Default: 64 blocks

Range of valid values: any positive integer

Takes effect: when Capture is restarted

SP_OCT_LOGWRAP_RESTART

This parameter controls how Capture behaves in response to the following situations:

- The archives are not available.
- Capture is denied permission to open a log.

When Capture cannot find an archive log, or when it cannot open a log, the process stops by default (a setting of 0). At this setting, when Capture cannot access a log, the **show capture** command returns a status of "`Stopped due to missing archive log`."

You can configure Capture to wait a certain amount of time (in seconds) and then start again automatically. This allows time to assign the correct read permissions or run any external log-management processing that moves the archives into the location expected by SharePlex. Capture waits, checks for the logs, stops if they are not yet available or cannot be opened, and continues checking and stopping until the logs are restored or opened. At this setting, the **show capture** command returns a status of "`waiting for logfile`."

Default: 0 seconds (do not restart automatically if logs are not available)

Range of valid values: 1 to any positive number of seconds

Takes effect: Immediately

SP_OCT_MIN_SESSIONS

This parameter controls the minimum number of subqueues that Capture creates and maintains to contain data from concurrent transactions. This parameter supports the Post Enhanced Performance (PEP) feature by allowing Post to increase its own concurrency. The PEP feature is controlled with the SP_OPO_DEPENDENCY_CHECK parameter.

For best results, set SP_OCT_MIN_SESSIONS to at least the number of CPU cores that you have on the target system.

Default: 8

Range of valid values: any positive integer

Takes effect: when Capture is restarted

SP_OCT_OLOG_DELAY

This parameter controls the amount of microseconds capture's redo log reader spends sleeping when it has detected that there is no more data to read. This parameter essentially comes into play only when capture is idle. When busy capture essentially won't be sleeping since there are records to read.

Default: 10000 (microseconds)

Range of valid values: any positive integer

Takes effect: Immediately

SP_OCT_OLOG_NO_DATA_DELAY

This parameter controls the length of time that the log reader will sleep before a retry when there is no data to process.

Default: 10000 (microseconds)

Range of valid values: any positive integer

Takes effect: when Read is restarted

SP_OCT_OLOG_QUEUE_SIZE

This parameter sets the size of the log reader queue.

Default: 16384

Range of valid values: 100 to any positive integer

Takes effect: Process restart

SP_OCT_OLOG_RDS_MINER

NOTE: Enabling the SP_OCT_OLOG_RDS_MINER parameter is deprecated and no longer supported starting with Oracle 19c.

This parameter controls whether or not Capture uses a second thread to help keep Capture from lagging behind Oracle when capturing from an Oracle RDS database.

Due to the processing load incurred by using this thread, it is disabled by default. To enable it, set this parameter to 1.

Default: 0 (off)

Range of valid values: 0 or 1 (enabled)

Takes effect: Process restart

SP_OCT_OLOG_REOPEN

This parameter controls whether or not the Capture process will close and re-open the online redo log when it reaches the end of the file. If this parameter is enabled, the user may control the length of time that the Capture process will wait before retry by employing the SP_OCT_OLOG_NO_DATA_DELAY parameter.

Default: 1 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: Process restart

SP_OCT_OLOG_USE_OCI

This parameter controls whether Capture captures the source change records by reading the redo logs directly from the files or by using OCI calls. The default of 0 directs Capture to read directly from the redo logs.

- If the source database is running on ASM, enabling this parameter is optional. Capture can read the redo logs directly or use OCI calls.
- If the source database is remote from the machine where Capture is running, such as on a cloud service, this parameter must be set to 1 so that Capture uses OCI calls to get the source change records.

Default: 0 (off, Capture reads directly from the redo logs)

Range of valid values: 0 or 1 (flag)

Takes effect: Process restart

SP_OCT_ONELINE_DEBUG

This parameter controls the oneline debug feature for the Capture process.

The oneline debug feature enables you to perform debugging for out-of-sync errors without consuming a large amount of the system resources. The oneline debug logs just enough information in one line to detect where in the data stream an out-of-sync condition occurred. Once oneline debug identifies the process that is causing the problem, you can then enable regular debugging for that process.

To enable this parameter, specify the objects that you want to debug by their object ID, and separate each one by a comma. An example is:

```
sp_ctrl> set sp_oct_oneline_debug 230230, 351626
```

The following items are logged:

- The transaction identifier used by SharePlex
- The type of DML or DDL operation that was affected
- The object ID of the affected source table
- The row identifier of the affected row. (Oracle rowid)
- The time when the operation occurred (Oracle)
- The log sequence number and offset within the log of the affected operation
- The Oracle SCN for the transaction
- An internal SharePlex code that prevents redundant operations

Default: Disabled

Range of valid values: a string that forms a list of objects listed by object ID, separated by commas.

Takes effect: Immediately

SP_OCT_OPS_LOGREADER_DELAY

This parameter controls the log reader delay time before retry when it encounters no_data.

Default: 50000 (microseconds)

Range of valid values: any positive integer

Takes effect: when Capture is restarted

SP_OCT_OPS_LOGREADER_RETRY

This parameter controls the number of times the log reader will retry when it encounters no_data.

Default: 3

Range of valid values: any positive integer

Takes effect: when Capture is restarted

SP_OCT_OPS_NO_DATA_DELAY

This parameter controls the length of the sleep that is enforced per node when no data is encountered in the sequencer.

Default: 5 (deciseconds)

Range of valid values: any positive integer

Takes effect: when Read is restarted

SP_OCT_PREFER_ARCHIVE

This parameter controls whether Capture reads the online version of a redo log or the archived version if it is available. By default, Capture reads only from the online logs and only reads an archived version if the online log is no longer available.

When this parameter is set and Capture finds the required log online, Capture makes an additional check to determine if the log is also archived. If the log is archived, Capture queries the archive path and reads the archived log instead of the online one. If the log is not archived, Capture reads the online log.

Normally, SharePlex provides the least latency when reading from the online redo logs, but this parameter may help improve Capture performance on Exadata. SharePlex can process higher volumes of data when reading from a multiplexed archive location outside of the Exadata ASM file system. Should a backlog develop with high volumes of data on Exadata, you can set SP_OCT_PREFER_ARCHIVE parameter to a value of 1 and configure SharePlex as directed in the Tune Capture on Exadata section in the SharePlex Administration Guide.

To use this parameter, multiplexing of the redo logs must be enabled, and the path to the archive logs must be set in SharePlex. For more information, see SP_OCT_ARCH_LOC on page 296.

Default: 0 (disabled)

Range of valid values: 0 or 1 (flag)

Takes effect: when Capture is restarted

SP_OCT_READ_ARC

This parameter controls whether SharePlex will read only from Oracle archive logs. The default setting of 0 means SharePlex will utilize both online and archived logs to read Oracle redo logs. Any non-zero value means SharePlex capture process will read only from archived Oracle redo logs. SharePlex will wait until an online log is archived before attempting to read that sequence of the Oracle redo log.

0 = default behavior SharePlex, read both online and archived redo logs

1 = read only archived redo logs. The search sequence for the archived logs will start from Oracle defined archived location followed by searching under SP_OCT_ARCH_LOC directory specification.

2 = read only archived redo logs. The search will only occur under SP_OCT_ARCH_LOC directory specification. * (Will* *NOT search under ORACLE defined archive locations).* SP_OCT_CK_LOC_FIRST value will be ignored with this setting. It will be assumed to be 1.

3 = read only archived redo logs. Read only from Oracle defined archive locations. SP_OCT_ARCH_LOC directory will be ignored. SP_OCT_CK_LOC_FIRST value will be ignored with this setting. It will be assumed as 0.

Best practice: When enabling SP_OCT_READ_ARC, a natural delay of at least the length of time to generate and copy an archive log from an online log will be apparent in data latency. While this latency could be acceptable, reader's performance in fetching keys is reliant on read consistent views. Reader key fetching is dependent on the consistent views being current (or close to it) but using the parameter delays the age of the view with at least the latency time. To circumvent the need for reader to fetch keys, it's highly recommended that customers using the mentioned capture parameter enable the source database's supplemental logging for primary keys and unique keys (PU/UK) for tables in replication.

Default: 0

Range of valid values: 0 - 3

Takes effect: when Capture is restarted

SP_OCT_REDOLOG_ENSURE

This parameter controls the amount of seconds that Capture waits before processing a change record from the online redo log. This wait is the difference between the Oracle timestamp of the record and the current system time. It prevents Capture from trying to read records that the Oracle logwriter is not finished writing, which would result in the capture of a partial record. It ensures that Capture has a complete block to process.

Capture disables this parameter automatically when capturing from a remote database.

Default: 2 seconds

Range of valid values: 0 to any positive integer

Takes effect: when Capture is restarted

SP_OCT_REDUCED_KEY

This parameter controls which parts of an operation in the redo log are sent by Capture to the target for use by Post. Different SharePlex features may require more or less data to be made available to the Post process.

- If set to 0, this parameter directs Capture to send all of the data that Oracle writes to the redo record other than LONGs and LOBs. This setting sends the most data across the network, but is required to support certain SharePlex features.
- If set to 1 (the default), this parameter directs Capture to send the key values and, for UPDATEs, the before and after values of columns that were changed in the operation. This setting is the default. This setting provides a good balance between replication performance and target data integrity, because it enables Post to perform a before-and-after comparison of the changed columns when constructing the WHERE clause. For more information about how SharePlex uses before and after values, see the SharePlex Administration Guide.

- If set to 2, this parameter directs Capture to send the key values and, for UPDATEs, the after values of the columns that changed to the target. This setting sends the least amount of data across the network and also enables Post to only use the key values when posting data, which improves performance.

See also SP_OPO_REDUCED_KEY.

Default: 1

Range of Valid Values: 0, 1, 2

Takes effect: Immediately

SP_OCT_REPLICATE_ALL_DDL

This parameter controls *expanded DDL replication*, which is the replication of DDL for objects that are not in the SharePlex replication configuration file. The objects must exist in the source database before configuration activation and also on the target (except in the case of CREATE). SharePlex replicates the DDL statements, but does not maintain the objects on the target throughout any future DML changes.

Setting the SP_OCT_REPLICATE_ALL_DDL requires identical source and target databases in order for the replicated DDL to execute successfully on the target system. Because this DDL is broadcast to all target machines, all target systems must be identical.

SharePlex replicates supported DDL for all objects in all schemas of the active datasource, so those components must exist in the target database for the operations to succeed.

To configure the Post process to stop for errors encountered when applying replicated DDL operations to the target database, set the SP_OPO_STOP_ON_DDL_ERR parameter.

If you are replicating the creation of packages or stored procedures, the name of the objects in the body of the package or stored procedure should be fully qualified, because other users or schemas may execute this package or stored procedure.

For a list of operations supported by SharePlex, see the Release Notes that correspond to your version of SharePlex.

NOTE: Expanded DDL replication supports not only tables and sequences but also a wide range of other objects such as procedures, functions, users, and views, which are not part of replication. Some of these objects may have underlying objects that *are in replication*. In those cases, Expanded DDL replication applies not only to the object that is outside the replication configuration, but also to the underlying objects that are in replication.

SharePlex does not support the Oracle Flashback Table feature. If the SP_REPLICATE_ALL_DDL parameter is enabled (value of 1), SharePlex may try to replicate the flashback DDL, which will return an error. To perform Flashback Table on a table that is in replication, use the following procedures in the SharePlex Administrator Guide to work around this issue:

1. Remove source objects from replication
2. Perform the flashback
3. Add or change objects in an active configuration

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OCT_REPLICATE_COMMENT

This parameter controls whether or not SharePlex replicates the following DDL operations on tables that are listed in the configuration file:

- ALTER TABLE to ADD COMMENT
- COMMENT ON

By default, the preceding DDL operations are replicated. To disable replication of these operations, set this parameter to 0. This can be useful if replication of these operations is impacting performance.

Default: 1 (enabled)

Range of valid values: 0 or 1

Takes effect: when Capture is restarted

SP_OCT_REPLICATE_CTAS

This parameter determines how SharePlex replicates CREATE TABLE AS SELECT (CTAS) statements when the name of the new table matches a wildcard specification in the SharePlex configuration file. It has options to specify the way that CTAS statements are replicated by SharePlex, depending on whether or not the table exists on the target.

Parameter Value	Use Case	Description
1 (Default)	SELECT FROM table exists on the target	This setting replicates only the original CTAS statement and requires a populated SELECT FROM table to exist on the target. For example, the following CTAS statement is replicated as-is to the target: **create table my_table as select * from table_a;** In this example, **table_a** must exist on the target in order for the SELECT to succeed and populate the new table **my_table**.
2	SELECT FROM table does not exist on the target	This setting causes SharePlex to: 1. Replicate a CREATE TABLE statement to create the new table on the target. 2. Replicate all of the DML that is returned by the source SELECT statement as INSERTS to populate the new target table. For example, if a source statement is:

Parameter Value	Use Case	Description
		create table my_table as select * from table_a; The statements posted by SharePlex on the target are the following, assuming **table_a** has two columns and two rows: **Create table my_table (c1 number, c2 varchar2(20));** **Insert into my_table values (1,'a');** **Insert into my_table values (2,'b');** **Commit;** This produces the same result as a CREATE TABLE AS SELECT, but does not require the SELECT FROM source table to exist on the target. **IMPORTANT**! The replicated DML data must fit into memory that is allocated by Capture. If the amount of data is very large, this could cause Capture to fail.

Default: 1

Range of valid values: 1 or 2

Takes effect: Process restart

SP_OCT_REPLICATE_DDL

This parameter controls whether SharePlex replicates the following operations on objects in the active configuration that are being replicated by SharePlex:

- ALTER TABLE to ADD COLUMN, MODIFY COLUMN, DROP COLUMN
- ALTER TABLE to ADD, MODIFY, DROP, SPLIT, COALESCE, MOVE, TRUNCATE, EXCHANGE PARTITION/SUBPARTITION
- TRUNCATE TABLE

You can control whether both operation types are replicated, or just one or the other, or none.

NOTE: SharePlex 11.1 with Oracle 21C as a source currently supports only DML operations. Support for DDL operations will be added in a future version.

See also SP_OCT_AUTOADD_ENABLE for additional default DDL replication functionality.

To replicate DDL on objects outside the configuration file, see the SP_OCT_REPLICATE_ALL_DDL parameter.

Default: 3 (replicate ALTER TABLE and TRUNCATE)

Range of valid values:

0 (disable replication of both ALTER TABLE and TRUNCATE)

1 (enable ALTER replication only)

2 (enable TRUNCATE replication only)

3 (enable replication of ALTER and TRUNCATE)

Takes effect: immediately

SP_OCT_REPLICATE_DLOAD

This parameter controls whether or not SQL*Loader direct-path loads are replicated. The default setting of 1 enables direct-path load replication. SharePlex supports replication *for non-parallel loads only* (PARALLEL=FALSE). The database must be in archive mode, and table logging must be enabled. To disable replication of direct-path loads, change this parameter to 0.

Default: 1 (replicate direct-path loads)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OCT_REPLICATE_GRANT

This parameter controls whether SharePlex replicates GRANT commands that are issued for tables that are listed in the configuration file.

Default: 0 (disabled)

Range of valid values: 0 or 1

Takes effect: when Capture is restarted

SP_OCT_REPLICATE_MVIEW

This parameter controls whether or not activation will put the existing materialized views found during activation into replication. By default, it is enabled. The materialized views must exist in the source and target before activation.

This parameter is used only during activation to decide whether or not to put the materialized view in replication. To configure SharePlex to add materialized views to replication when they are created *after* activation, see SP_OCT_AUTOADD_MVIEW.

Default: 1 (replicate materialized views)

Range of valid values: 0 or 1

Takes effect: Process restart

SP_OCT_REPLICATE_POSTER

This parameter controls whether or not the Capture process on a system replicates data posted by the Post process on that system. Leave this parameter at the default setting of 0, which tells Capture to ignore Post activities on the same system. When establishing certain replication configurations — primarily cascading replication — you might be instructed to set this parameter to 1, which causes Capture to replicate posted changes.

Default: 0 (do not replicate Post transactions)

Range of valid values: 0 or 1 (flag)

Takes effect: when Capture is restarted

SP_OCT_REPLICATE_SEQUENCES

This parameter controls whether SharePlex replicates changes made to sequences that are listed in the configuration file, either explicitly or by means of a wildcard. By default, it is enabled. The sequences must exist in the source and target before activation.

You can configure SharePlex to add sequences to replication when they are created *after* activation. For more information, see SP_OCT_AUTOADD_SEQ on page 300.

A setting of 0 excludes sequences from replication, even if they are listed in the configuration file or their names satisfy a wildcard specification.

IMPORTANT! To replicate sequences, the supplemental logging of primary and unique keys must be enabled at the database level, or you must enable supplemental logging for primary keys on the **sys.seq$** table.

Default: 1 (replicate sequences)

Range of valid values: 0 or 1

Takes effect: Process restart

SP_OCT_REPLICATE_SYNONYM

This parameter controls whether SharePlex replicates CREATE SYNONYM and DROP SYNONYM commands that are issued for tables that are listed in the configuration file.

Default: 0 (disabled)

Range of valid values: 0 or 1

Takes effect: when Capture is restarted

SP_OCT_REPLICATE_TRIGGER

This parameter controls whether SharePlex replicates CREATE TRIGGER and DROP TRIGGER commands that are issued for tables that are listed in the configuration file.

NOTE: Do not replicate triggers in an active-active replication scenario. For more information, see *Configure replication to maintain multiple peer databases* in the SharePlex Administration Guide.

Default: 0 (disabled)

Range of valid values: 0 or 1

Takes effect: when Capture is restarted

SP_OCT_REQUIRED_DATA_IS_LOGGED

This parameter enables the Capture process to gather additional information to improve rollback handling. With rollback handling, the Read process will require more resources to accomplish and improve performance in return. If system resources are scarce and the systems experiences lots of rollbacks, to lower the Read process resource consumption, you can disable this parameter. This parameter is enabled by default.

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: when Capture is restarted

SP_OCT_TRUNC_PARTITION_BY_ID

Use this parameter to ensure that replication of an ALTER TABLE to truncate or drop a system-generated partition affects the correct partition on the target.

Because the database generates the names of system-generated partitions, the names of those partitions on the source will not match the names of their corresponding partitions on the target. However, the partition positions for the same time frame will match if the target is an exact copy of the source.

To ensure that replication affects the correct partition on the target:

1. Make certain the source and target tables structures and partition definitions are identical. For partitions with the same high value, the partition position from **dba_tab_partitions** should be the same between source and target before replication starts.
2. Set the SP_OCT_TRUNC_PARTITION_BY_ID parameter to 1. This setting directs SharePlex to identify the partition by using the partition position, rather than by using the partition name that is specified in the original ALTER TABLE command. Post maps the partition position to the correct partition name in the target table.

When this parameter is enabled, SharePlex checks the SP_SYS_TARGET_COMPATIBILITY parameter to verify that the target Post process is version 8.6.4 or higher. This is the minimum version that supports ALTER TABLE to truncate or delete a system-generated partition by ID. If the target Post process is an earlier version, the ALTER TABLE is replicated by using the partition name, and a warning message is logged to the SharePlex **event_log** on the target.

Default: 0 (disabled)

Range of valid values: 0 or 1

Takes effect: immediately

SP_OCT_USE_DST

Use this parameter to tell SharePlex to disable the correction for daylight savings time in redolog to ensure logic.

Default: 0 (Disabled)

Range of valid values: 0 or 1

Takes effect: Process restart

SP_OCT_USE_SUPP_KEYS

Use this parameter to tell SharePlex to use the columns set by Oracle's supplemental logging as the key columns when a row is updated or deleted.

In a typical replication scheme, SharePlex chooses a set of key columns for Post to use when it updates or deletes a row. When Oracle's PK/UK supplemental logging is enabled, Oracle logs key columns for each update. These key columns may not always match what SharePlex chose as its keys, which causes unnecessary work for the Read process. Setting this parameter overrides the default behavior of SharePlex, allowing SharePlex to be faster and more efficient.

NOTES: The SP_OCT_USE_SUPP_KEYS parameter takes effect only if the supplemental logging is enabled for PK/UI.

If a table is configured with horizontal partitioning in the configuration file, you must include the horizontal partitioning column condition in a redo log group, unless that column condition is already part of the PK/UK for that table.

LIMITATION:

When a table or column name exceeds 30 characters, Oracle does not log key columns for each update operation in the Redo log, resulting insufficient key columns for SharePlex Poster to use, leading to an Out of Sync condition at the target.

It is recommended to set `SP_OCT_USE_SUPP_KEYS = 0`, which directs SharePlex to select the appropriate set of key columns for Poster to use.

This may cause additional overhead during the identification of key columns in the SharePlex Read process and at the target during update transactions, potentially resulting in latency at the target.

This parameter is not compatible with the SP_ORD_HP_IN_SYNC parameter. If both parameters are enabled, SP_OCT_USE_SUPP_KEYS overrides SP_ORD_HP_IN_SYNC.

Default: 0 (Disabled)

Range of valid values: 0 or 1 (Enabled)

Takes effect: Immediately

Read parameters

These parameters are used by the SharePlex Read process.

SP_ORD_BATCH_ENABLE

This parameter controls the enabling of the Batch Processing functionality. By default it is enabled. This allows the Read process to combine multiple identical records into a single record, or into a batch, for processing by the Post process.

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: Immediately

SP_ORD_BATCH_MAX

This parameter controls the maximum number of concurrent batch transactions.

Default: 50 (transactions)

Range of valid values: any integer between (and including) 5 and 250

Takes effect: when Read is restarted

SP_ORD_BATCH_MATCH_MIN

This parameter controls the minimum number of matched operations before processing a batch.

Default: 2

Range of valid values: any positive integer

Takes effect: when Read is restarted

SP_ORD_CDA_LIMIT

This parameter controls the number of cursors cached by each login of the Read process. You might need to increase its value if replication starts falling behind Oracle activity on the source system. An initial setting of 15 cursors is recommended if you have a large number of tables in replication.

Default: 5 cursors

Range of valid values: any positive integer

Takes effect: when Read is restarted

SP_ORD_DATE_ MSG

This parameter can be set so that the Read process prints a warning message to the Event Log and the Read log when it detects an invalid date column. A setting of 0 disables the parameter, and a setting of 1 activates it. The error message generated by Read is:

```
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.
```

Default: 0 (do not print warning message)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately.

SP_ORD_DELAY_RECORDS

The Read process processes records in batches. The size of the batches depends on the number of records in the capture queue. If the number of records is large, Read ignores the value of this parameter, and the batches are kept as small as possible. If Read is keeping pace with Capture, the size of the batch is approximately the value set by this parameter.

Set this parameter low if SharePlex is generating `rollback segment too old` messages in the Event Log. It instructs SharePlex to pass the data along sooner. This parameter is of use where only minimal latency can be tolerated.

IMPORTANT: Use this parameter with caution, because reducing the number of records increases the I/O that SharePlex must perform, which increases system overhead. It can also negatively impact smooth interaction between SharePlex processes.

Default: 200 records

Range of valid values: any positive integer

Takes effect: Immediately

SP_ORD_FIRST_FIND

This parameter controls how the Read process checks column conditions to determine whether or not a replicated row change satisfies them.

- At the default of 1, when a row change satisfies a column condition, SharePlex does not check any other column conditions to see if that row change also satisfies any of them.
- At a value of 0, SharePlex sends the data to all target systems where the column conditions are satisfied.

For more information about horizontal partitioning, see the SharePlex Administration Guide.

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: when Read is restarted

SP_ORD_HP_HASH

This parameter controls the number of slots in the hash table used for Horizontal Partitioning. By default this parameter is set to 16 slots to minimize memory usage (usage is the number of slots x 32 bytes for every transaction with an insert). If the user's system does a lot of insert operations followed by updates (in the same transaction) on a horizontally partitioned table then SharePlex will use this hash table a lot and this value may need to be increased for performance. Additionally, if the user system has a lot of long transactions with inserts on tables with horizontal partitioning the value of this parameter might need to be increased.

Default: 16 slots

Range of valid values: any positive integer

Takes effect: when Read is restarted

SP_ORD_HP_IN_SYNC

This parameter is used for horizontally partitioned replication to ensure that data is replicated properly when a value for a column in a column condition changes so that the row no longer satisfies the condition.

It enables SharePlex to automatically correct the following:

- UPDATEs that cause a row to meet a different column condition than the one created for that row, sending the changes to a different location. An example would be an UPDATE to a row for which the column condition is **region=East** that changes the value of the **region** column to WEST. Such operations will fail because the original INSERT statement for that row was replicated to the original location (the Eastern region), so the row does not exist in the new location (the Western region) when Post attempts the update there.
- UPDATEs that cause a row to meet a column condition (and be replicated) when the row was not supposed to be replicated. An example would be when the region column is updated from the value of HEADQUARTERS (for which a row is not replicated) to the value of WEST. Such operations will fail because the original INSERT statement for that row (into the headquarters system) was not replicated to the Western region's system, so Post cannot perform the update there.
- UPDATEs that cause a row to no longer meet any column condition. An example would be when the **region** column is updated from the value of WEST to the value of HEADQUARTERS. The original INSERT statement was replicated to the Western region's system, but the update to the new value is not replicated, because the new value does not meet a column condition (headquarters data is not shared). The rows are now out of synchronization, but there are no errors.

When this parameter is enabled, SharePlex automatically corrects rows for which UPDATEs cause the preceding conditions. SharePlex converts the UPDATE to a DELETE and, if needed, an INSERT.

To convert an UPDATE statement (which normally only uses the changed columns and the key) to an INSERT statement, SharePlex needs values for all of the columns. Enabling SP_ORD_HP_IN_SYNC directs SharePlex to

send all of the columns in a row to the Post process when there is an UPDATE to a table using horizontally partitioned replication, so that an INSERT can be constructed.

Set this parameter on the **source** system before you activate the configuration. If replication is active, set the parameter and then reactivate the configuration so that SharePlex can rebuild its object cache.

If you know that the columns in column conditions for tables using horizontally partitioned replication will never change, leave this parameter set to 0, because using it incurs processing overhead.

This parameter is not compatible with SP_OCT_REDUCED_KEY and SP_OPO_REDUCED_KEY [any value: 1 or 2] as it overrides the behavior of both the parameters.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: when Read is restarted

SP_ORD_LDA_ARRAY_SIZE

This parameter controls the number of logins made to the database for read consistency. If the Read process slows down, try increasing the value of this parameter. The maximum setting is determined by the MAX_PROCESSES parameter in the init_ora file.

Default: 5 logins

Range of valid values: any positive integer

Takes effect: when Read is restarted

SP_ORD_ONELINE_DEBUG

This parameter controls the oneline debug feature for the Read process.

The oneline debug feature enables you to perform debugging for out-of-sync errors without consuming a large amount of the system resources. The oneline debug logs just enough information in one line to detect where in the data stream an out-of-sync condition occurred. Once oneline debug identifies the process that is causing the problem, you can then enable regular debugging for that process.

To enable this parameter, specify the objects that you want to debug by their object ID, and separate each one by a comma. An example is:

```
sp_ctrl> set sp_ord_oneline_debug 230230, 351626
```

The following items are logged:

- The state of the operation, which can be **Dispatch** (read from queue), **processMessage** (process the SQL code), or **ExecSQL** (apply to target)
- The transaction identifier used by SharePlex
- The type of DML or DDL operation that was affected
- The owner and name of the target table

- The object ID of the affected source table
- The row identifier of the affected row. (Oracle rowid)
- The time when the operation occurred (Oracle)
- The log sequence number and offset within the log of the affected operation
- The Oracle SCN for the transaction
- An internal SharePlex code that prevents redundant operations
- The routing information

NOTE: This information is repeated for each target if the row is being routed to multiple targets.

Default: Disabled

Range of valid values: a string that forms a list of objects listed by object ID, separated by commas.

Takes effect: Immediately

SP_ORD_MSGS_CK_FREQ and SP_ORD_RCM_SKIP_RATIO

These parameters work together, so if one of them is set its default value of 0, then the other parameter, if set to a number other than 0, does not work. Both parameters support the automated process in which the Read process detects that too much time is being taken to process its queries, by checking the ratio of disk gets per executed queries. Once Read makes this determination, it gets rid of its existing read consistent views and replaces it with a new view.

The ORD_MSGS_CK_FREQ parameter is the frequency that the Read process checks if the ratio has been surpassed. For example, if you set this parameter to 100, the Read process will check every 100 queries to see if the ratio is still good. The recommended setting is 1,000 queries.

The ORD_RCM_SKIP_RATIO parameter specifies the number (the ratio of disk gets to executed queries) that when reached, causes the read consistent view to be replaced. The most sensitive setting is 1, which means one disk get per one executed query. The recommended setting is 2.

To turn off these parameters, set one of them to 0, which disables the other parameter.

Default:

SP_ORD_MSGS_CK_FREQ: 10,000

SP_ORD_RCM_SKIP_RATIO: 2

Range of valid values:

SP_ORD_MSGS_CK_FREQ: 0 to 100,000

SP_ORD_RCM_SKIP_RATIO: 0 to 1000 (not recommended to set this greater than 5)

Takes effect: Process restart

SP_ORD_RMSG_LIMIT

This parameter controls the frequency of the checkpoints performed by the Read process when it reads messages from the capture queue and determines the key values. A checkpoint saves the capture queue to a cache file, commits the outgoing queue messages (being passed to the export queue), and does a read release (delete) on the heldover records in the capture queue that have already been received by the export queue.

The higher the value of this parameter, the more records will be held in memory before the checkpoint is triggered. A very high value causes less I/O on the system and faster processing — but at the expense of increased memory usage and a longer recovery time should something unforeseen cause the Read process stop. A low value increases I/O, which increases the recovery speed but reduces throughput speed.

The default value of 100,000 records should establish a reasonable balance between the need for speed and the conservation of memory and process recovery, but you can adjust this parameter to suit your processing requirements. Adjustment options range from checkpointing after every record to holding as many records as the system and its memory can accommodate.

Default: 100,000 records

Range of valid values: any positive integer within system limitations

Takes effect: immediately

SP_ORD_ROLLBACK_TXNS_MAX

This parameter limits the number of entries the reader will keep in its transaction cache that indicate a transaction was completely rolled back

Default: 5,000 records

Range of valid values: any positive integer within system limitations

Takes effect: process restart

SP_ORD_SEND_DDL_TO_FIRST

This parameter helps improve Post performance when multiple post queues are in use and you are replicating DDL for objects that are not in the replication configuration (SP_OCT_REPLICATE_ALL_DDL=1).

This parameter directs Import to send DDL for objects that are *not* in the replication configuration to a specific post queue. Normally, DDL that is not related to objects in the replication configuration can be sent to any available queue. Large amounts of this DDL can block the DML of the objects that are in replication, causing data latency to increase.

This DDL, because it has no association with the objects in replication, does not have to be applied in any sequence relative to those objects. By routing it independently through a dedicated queue, you can free the other queues to process replicated data and its related DDL.

If you activate this parameter, you must specify the post queue through which to process the DDL for the non-replicating objects. This specification must be placed on the first line after the "Datasource:o.SID" line in the configuration file, as follows (the ! is a placeholder that replaces an actual object specification):

```
Datasource: o.dbprod
```

```
#Route for non-replication DDL

! sysmm:queddl1@o.dbprep

#The rest of the configuration entries

SCOTT.EMP SCOTT.EMP sysmm:que1@o.dbrep

SCOTT.FOO SCOTT.FOO sysmm:que2@o.dbrep
```

If you do not make an explicit designation in that manner, the DDL will be routed to the first route listed. For example, in the following configuration file, all DDL that is not associated with an object in replication will be sent to sysmm:que1@o.dbrep.

```
Datasource: o.dbprod

SCOTT.EMP SCOTT.EMP sysmm:que1@o.dbrep

SCOTT.FOO SCOTT.FOO sysmm:que2@o.dbrep
```

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately.

Export parameters

These parameters are used by the SharePlex Export process.

SP_XPT_AES_KEY_LENGTH

This parameter controls the size of the AES encryption key that is used by Export to encrypt data that is sent across the network from source to target. The key itself is generated by issuing the **create encryption key** command, which generates a full 256-bit key. The SP_XPT_AES_KEY_LENGTH parameter controls how much of that length is used as the key when Export encrypts the data. The default is 128 bits, the minimum length. A longer key is harder to hack but requires more CPU power. To configure SharePlex for AES encryption, see the SharePlex Administration Guide.

Default: 128-bits

Range of valid values: 128, 192 or 256 bits

Takes effect: when Export is restarted

SP_XPT_ENABLE_AES

This parameter controls whether Advanced Encryption Standard (AES) is enabled to encrypt replicated data that is sent by Export across the network. To configure SharePlex for AES encryption, see the SharePlex Administration Guide.

NOTE: SharePlex uses AES Cipher Block Chaining (CBC mode) for data encryption.

Default: 0 (disabled)

Range of valid values: 0 or 1 (enabled)

Takes effect: when Export is restarted

SP_XPT_ENABLE_COMPRESSION

This parameter controls the compression of data across TCP/IP.

You can enable compression to reduce the amount of data that SharePlex sends across the network. SharePlex uses LZIP lossless compression. Enabling compression on the source SharePlex instance automatically enables compression to all targets of the source SharePlex instance.

By default compression is disabled. You can enable compression by itself or in conjunction with encryption. For more information about encryption, see the SharePlex Administration Guide.

Default: 0 (disabled)

Range of valid values: 0 or 1 (enabled)

Takes effect: when Export is restarted

SP_XPT_KEEPALIVE

This parameter controls whether the Export process sends a "hello" message to Import at regular intervals to prevent TCP timeouts when replication activity is low. If the network times out because no replication packets are being sent, SharePlex alerts you with a message such as "Export cannot connect to import on server2: timeout waiting for ack." By setting SP_XPT_KEEPALIVE to 1, you can eliminate this message and keep the SharePlex network connection alive.

Default: 0 (disabled)

Range of valid values: 0 or 1 (enabled)

Takes effect: when Export is restarted

SP_XPT_PORT_OVERRIDE

This parameter works in conjunction with the SP_XPT_USE_LOCALHOST parameter to enable SharePlex to send data through a secure tunnel port created with SSH® Secure ShellTM software. This parameter sets the local connection, which overrides the default SharePlex port. From that port number, the SSH daemon directs the connection to a different system (the SharePlex target machine) using another port number and the SSH data encryption.

NOTE: The SP_XPT_PORT_OVERRIDE parameter at queue level is not supported for the Compare, Repair, Copy, or Append functionalities.

Default: 0 (disabled)

Range of valid values: 0 or any positive integer

Takes effect: when Export is restarted

SP_XPT_SO_SNDBUF

This parameter tunes the TCP/IP window size on the source machine. It works in conjunction with the SP_COP_SO_RCVBUF parameter on the target machine to establish the size of a packet sent across the network. SharePlex references both parameters when TCP/IP sockets are created at the startup of sp_cop and the Export process.

If SharePlex is replicating across a WAN and the export queue is continually backlogged, try adjusting both parameters. SP_XPT_SO_SNDBUF must be set equal to or greater than the value of SP_COP_SO_RCVBUF, in multiples of 1024 bytes. To size the parameters, determine the ping time between the source and target machines, then use the following formula for both parameters:

param_value / ping_time **= bytes per second**

For example, if ping time is 200 milliseconds, and the value for the two parameters is 64K, SharePlex will send five 64K-packets every second, totaling 320K per second.

Unless you observe a bandwidth problem, Quest recommends leaving both parameters set to their defaults, which use the system's setting. To change SP_XPT_SO_SNDBUF, set it on the source system, then stop and start Export on that system.

If transfer still is slow, try increasing the SP_IMP_WCMT_MSGCNT and SP_IMP_WCMT_TIMEOUT parameters on the target system. Set SP_IMP_WCMT_MSGCNT to at least 10,000.

Default: 0 (default is set by the operating system)

Range of valid values: positive integers, in bytes, using multiples of 1024. Maximum is set by the operating system.

Takes effect: when Export is restarted

SP_XPT_USE_LOCALHOST

This parameter enables SharePlex to send data through a secure tunnel port created with SSH® Secure Shell™ software. The Export process reads this parameter before making a TCP connection. If the parameter is enabled, the Export process connects to the local host through a local port number, where the SSH daemon directs the connection to a different system (the SharePlex target machine) using another port number and the SSH data encryption.

Default: 0 (disabled)

Range of valid values: 0 or 1 (flag)

Takes effect: when Export is restarted

Import parameters

These parameters are used by the SharePlex Import process.

SP_IMP_ENABLE_AES

This parameter enforces the use of data encryption between the source system and the target on which it is set. It prevents Import from accepting incoming data unless data encryption is enabled in the Export process. It must be enabled if SP_XPT_ENABLE_AES is enabled.

> **NOTE:** SharePlex uses AES Cipher Block Chaining (CBC mode) for data encryption.

To configure SharePlex for AES encryption, see the SharePlex Administration Guide.

Default: 0 (disabled)

Range of valid values: 0 or 1 (enabled)

Takes effect: when Import is restarted

SP_IMP_QUEUE_PAUSE

This parameter pauses the writing of data to the post queue when that queue contains the specified number of messages. Post stores queue messages in shared memory until it issues a checkpoint, after which it releases the data from memory.

If the post queue runs out of shared memory, the read and write functions will start incurring file IO to free up the memory buffers. By pausing the queue writing, this parameter helps Post maintain its performance by avoiding the need for disk storage and the resultant slowdown in IO.

When Import is requested to write to a post queue, and the backlog for that queue is equal to or greater than SP_IMP_QUEUE_PAUSE, Import checkpoints with it associated Export and stops writing to any post queue. Import does, however, remain running. A warning is written to the event log and the status of the process in the **show statusdb** command output is shown as "paused." The **status** command shows "Paused" instead of "Running."

Import resumes writing to the post queue(s) if Import is stopped and restarted or Import detects that the backlog is less than or equal to the SP_IMP_QUEUE_RESUME parameter.

Use the SP_IMP_QUEUE_RESUME parameter to set the number of messages at which Import resumes writing to the post queue.

To use this feature, both SP_IMP_QUEUE_PAUSE and SP_IMP_QUEUE_RESUME must be greater than zero, and SP_IMP_QUEUE_PAUSE must be greater than SP_IMP_QUEUE_RESUME.

Default: 0 messages (disabled)

Range of valid values: *n* thousand messages, where *n* is any positive integer

Takes effect: immediately

SP_IMP_QUEUE_RESUME

This parameter works in conjunction with SP_IMP_QUEUE_PAUSE. If the number of messages in the post queue is lower or equal to the value set with this parameter, Import resumes writing to the post queue.

To use this feature, both SP_IMP_QUEUE_PAUSE and SP_IMP_QUEUE_RESUME must be greater than zero, and SP_IMP_QUEUE_PAUSE must be greater than SP_IMP_QUEUE_RESUME.

Default: 0 messages (disabled)

Range of valid values: *n* thousand messages, where *n* is any positive integer

Takes effect: immediately

SP_IMP_WCMT_MSGCNT

This parameter works in conjunction with the SP_IMP_WCMT_TIMEOUT parameter. It defines the number of messages that are processed before the Import process checkpoints. Checkpointing saves the state of the process in case it is needed for failure recovery.

When Import checkpoints, it triggers the Export process to perform its own checkpoint. If the value for this parameter is reached before the value set for SP_IMP_WCMT_TIMEOUT, it triggers the checkpoint.

In a WAN environment, you can increase this parameter to as many as 10,000 messages, with the understanding that increasing the message interval between checkpoints can negatively affect SharePlex's fault tolerance. If you are using SharePlex in an environment where the network continually fails, you can decrease this parameter.

Default: 10,000 messages

Range of valid values: any positive integer

Takes effect: immediately

SP_IMP_WCMT_TIMEOUT

This parameter works in conjunction with the SP_IMP_WCMT_MSGCNT parameter. It defines the number of seconds that pass before the Import process checkpoints. Checkpointing saves the state of the process in case it is needed for failure recovery.

When Import checkpoints, it triggers the Export process to perform its own checkpoint. If the value for this parameter is reached before the value set for SP_IMP_WCMT_MSGCNT, it triggers the checkpoint.

Default: 30 seconds

Range of valid values: any positive integer

Takes effect: immediately

Oracle Poster parameters

These parameters are used by the SharePlex Post process when applying data to an Oracle target.

SP_OPO_CHANGE_ID_START_VALUE

This parameter sets a start value for the **changeid** metadata column that can be included in a change history target and is set with the **target** command. The changeid is generated by SharePlex to uniquely identify records and prevent duplicates.

Default: 0 (off)

Range of valid values: 0 to 9223372036854775807

Takes effect: when Post is restarted

SP_OPO_COMMIT_REDUCE_MSGS

Sets the threshold for the Commit Reduction component of the Post Enhanced Performance feature. The Commit Reduction feature enables you to configure Post to combine smaller transactions into larger ones. This reduces the number of commits and acknowledgments that must be processed. The smaller the transaction, the bigger the performance gain.

When the specified number of messages is reached, Post issues a commit. The commits of transactions whose transactional borders are within this span of messages are skipped, and those transactions are all committed as one transaction. Commit reduction is on by default. To disable it, set this parameter to a value of 1.

The parameter setting is not an absolute threshold. SharePlex will not break up a transaction across different combined transactions. Therefore, Post may need to exceed that threshold in order to include all of the operations and the commit of the last transaction in the group.

Default: 100 messages

Range of valid values: 1 or any positive integer.

Takes effect: when Post is restarted

SP_OPO_CONNECTION_POOL

This parameter controls how Post uses the connection pool.

When connection pooling is enabled (the default) and a subqueue needs a connection to Oracle, it will try to find a subqueue that is committed. If it finds one, it will take that connection rather than open a new connection to Oracle. This allows Post to operate with fewer connections to Oracle.

Without connection pooling, each subqueue has a separate connection to Oracle and will keep that connection for a number of seconds (determined by the parameter SP_OPO_IDLE_LOGOUT) until the subqueue is committed.

To use this parameter:

- Enabling connection pooling can be helpful if SharePlex is running with a large number of subqueues.
- If SharePlex is running with a small number of subqueues (less than 20) that get reused quickly, it may be more efficient to disable the pooling and not move the connections around.

Default: 1 (enabled)

Range of valid values: 0 or 1 (flag)

Takes effect: when Post is restarted.

SP_OPO_CONT_ON_ERR

This parameter controls whether or not Post stops when it encounters errors that can be corrected. When this flag is set to the default of 0, Post stops for all Oracle and SharePlex errors. To have Post continue posting despite certain SharePlex or Oracle errors, list them in the **oramsglist** file in the **data** subdirectory of the variable-data directory and set this parameter to 1. If this parameter is set to 2, Post will treat table errors the same as other errors and stop unless the error is listed in the **oramsglist** file.

Post will always continue to post despite the following errors, regardless of the parameter setting:

Default errors for which Post will not stop

- unique key violation
- operation interrupted
- no data found
- no such table
- invalid number
- non-numeric in date
- invalid rowid
- invalid hex number
- cannot update not-null to null
- Year must be between -4713 and +9999
- check constraint violated
- packet writer failure*
- sequence not found
- Oracle internal error

*Packet writer failure and resource busy with nowait will retry based on the SP_OPO_RETRIES_MAX parameter which defaults to 10 after which poster will exit

For more information about how to configure Post to continue on errors, see the SharePlex Administration Guide.

Default: 0 (stop on all errors)

Range of valid values: 0, 1, 2 (flag)

Takes effect: immediately

SP_OPO_DEPENDENCY_CHECK

This parameter controls the following features:

- The Transaction Concurrency component of the Post Enhanced Performance feature. For more information about how to tune the performance of Post, see the SharePlex Administration Guide.
- Post handling of enabled ON DELETE CASCADE constraints on target tables (apply the replicated parent delete and ignore the replicated cascaded deletes).

To support these features, do the following:

1. **Set SP_OPO_DEPENDENCY_CHECK to the appropriate value:**
    - To enable Commit Reduction and Transaction Concurrency, set this parameter to 1. To use this feature, supplemental logging for primary and unique keys must be enabled on the source.
    - To enable ON DELETE CASCADE, set this parameter to 2.

NOTE: The ON DELETE CASCADE feature is dependent on the Post Enhanced Performance feature, which is enabled with a setting of 2. In addition to setting SP_OPO_DEPENDENCY_CHECK, you must also enable the logging of primary keys, unique index columns, and foreign key columns on the source.

2. Set the SP_OCT_REDUCED_KEY parameter to 0, which sends all of the required data to the target for use by Post. See SP_OCT_REDUCED_KEY .
3. Set the SP_OPO_REDUCED_KEY parameter to 0, 1 (the default) or 2. See SP_OPO_REDUCED_KEY.

Default: 0 (off)

Range of valid values: 0, 1, 2 (flag)

Takes effect: when Post is restarted

SP_OPO_DEPENDENCY_SIZE

This parameter sets the size of the memory that is used by dependency checking in the Post Enhanced Performance feature. The default value should be sufficient, but you can increase it if needed.

Default: 7019

Range of valid values: Any valid memory value

Takes effect: when Post is restarted

SP_OPO_DISABLE_OBJECT_NUM

This parameter prevents Post from posting replicated DML and DDL operations to the target, based on the object ID of the **source** table. You can set this parameter if the data in a source table is invalid or corrupted, if the source table contains data types that are not supported on the target, or for any other reason that you do not want operations for a table to be reflected in the target database.

You can set this parameter for one or more tables. It prevents further replication activity on the target table(s) of a source table until you have time to resynchronize the data and reactivate the configuration file. The Post process discards all replicated messages for these tables from the post queue, and the messages do not accumulate in the queue.

Use the parameter with caution. If it is enabled and DDL or DML is executed for the source table(s), the target data will be out of date because the changes are not posted. If there are dependencies on the table(s), such as a foreign key in other tables outside the replication configuration, disabling posting will prevent the dependencies from being satisfied.

This parameter is disabled by default. To enable it, issue the following command on the target system, where the numbers shown are the object IDs of the source tables to exclude from posting.

```
sp_ctrl(sysB)> set param SP_OPO_DISABLE_OBJECT_NUM 12345,67890
```

Separate each object ID with a comma, and allow no spaces between them. To use spaces between entries, enclose the entire list within quotes, as in the following example:

```
sp_ctrl(sysB)> set param SP_OPO_DISABLE_OBJECT_NUM "498438, 1000, 497109"
```

Invalid object IDs will be ignored.

When you are ready to begin posting to the target table again, set SP_OPO_DISABLE_OBJECT_NUM to 0.

Default: 0 (off)

Range of valid values: list of valid Oracle Object IDs separated by commas

Takes effect: immediately

SP_OPO_HINTS_LIMIT

This parameter controls the maximum number of hints (table/index combinations) that can be listed in the hints file. Use hints only if you see that Post is doing full table scans on tables where there are defined indexes. Using a large number of hints can reduce the performance of the Post process. See the SharePlex Administration Guide for more information about the hints feature.

Default: 100 hints

Range of valid values: any positive integer

Takes effect: when Post is restarted

SP_OPO_LOG_CONFLICT

This parameter enables the logging of information about successful conflict resolution procedures to the **SHAREPLEX_CONF_LOG** table. This applies only to the prepared routines that are provided by SharePlex.

- A setting of 1 enables the logging of conflict resolution to the **SHAREPLEX_CONF_LOG** table.

  NOTE: A setting of 1 will not update the columns EXISTING_TIMESTAMP and TARGET_ROWID (when existing data is not replaced) in the SHAREPLEX_CONF_LOG table.

- A setting of 2 enables the logging of conflict resolution to the **SHAREPLEX_CONF_LOG** table with Post query for additional meta data.

  Using LeastRecentRecord or MostRecentRecord prepared routines Post will query the target database for the timestamp column of the existing record. The query result is logged into the EXISTING_TIMESTAMP column of the SHAREPLEX_CONF_LOG table.

  For any prepared routines, on rows that aren't replaced by the incoming record, Post will query the TARGET_ROWID of the existing row that could have been replaced. Otherwise the ROWID of the existing row will not be logged.

  NOTE: A setting of 2 may affect the performance of Post as a result of making the query.

Default: 0 (Disabled)

Range of valid values: 0, 1, or 2

Takes effect: when Post is restarted

SP_OPO_MAX_CDA

This parameter is for the SQL cache module and controls the max number of cursor caches that a session can concurrently open. This number must be smaller than the Oracle setting for OPEN_CURSORS in v$parameter view. It is only used when SP_OPO_SQL_CACHE_DISABLE is set to 0 (enabled).

Default: 50

Range of valid values: any positive integer

Takes effect: when Post is restarted

SP_OPO_MAX_OEXN_TIME

This parameter controls the maximum duration, in seconds, that a SQL thread may spend in a call to OCIStmtExecute ( ) before the timekeeper thread defaults to a deadlocked position and forces the Multi-threaded Post process to exit.

Default: 900 [seconds]

Range of valid values: any positive integer

Takes effect: when Post is restarted

SP_OPO_MAX_ROLLBACK_MESSAGES

This parameter is the maximum number of messages to which the backward count is incremented for a partial rollback. The main thread scans forward and counts the backward messages. Once the backward count reaches the maximum number set for this parameter, the main thread resolves the rollback. When the rollback is resolved, the main thread resumes dispatching messages. If there are more backward messages, the main thread again scans forward in the subqueue. The next group of 10,000 backward messages (if the value for the parameter is set to 10,000) is then resolved, using the same procedure.

Default: 10,000 messages

Range of valid values: 11 - 1,000,000

Takes effect: when Post is restarted.

SP_OPO_NLS_CONVERSION

This parameter controls character set conversion between an Oracle source and an Oracle target.

For SharePlex to replicate all characters within the Oracle character sets that you are using, one of the following must be true:

- The character sets are identical on the source and target
- The character set of the source database is a subset of the character set of the target database (all characters contained on the source exist in the character set of the target)

The following character sets are tested and supported for SharePlex:

US7ASCII

UTF8

WE8ISO8859P1

AL16UTF16

AL32UTF8

KO16KSC5601

By default, SharePlex allows an Oracle target database to perform character conversion. Post notifies Oracle of the character encoding of the source data and Oracle performs any required conversion.

Depending on the character sets involved, the Oracle conversion might lead to data loss. For example:

Example 1: The Japanese character for 'rice' in the JA16SJIS character set has no corresponding symbol in the US7ASCII character set. If you attempt to replicate this symbol into a US7ASCII database, Oracle converts it to a '?' character.

Example 2: According to Oracle, the WE8ISO8859P1 character set is a superset of the US7ASCII character set, so it is logical to assume that any character in US7ASCII is posted unconverted into a WE8ISO8859P1 target database. This is true for characters in the range 0x00 to 0x7F. However, Oracle strips off the top bit of characters in the range 0x80 to 0xFF. This "conversion" may result in data loss while replicating to a character set that is a superset of the source.

NOTE: Oracle does not convert characters if the character sets are identical. Thus, posting WE8ISO8859P1 data to a database with a character set of WE8ISO8859P1 bypasses the Oracle conversion process.

To apply data without conversion:

Set the SP_OPO_NLS_CONVERSION parameter to 1 to apply the data with conversion.

NOTE: SharePlex will always convert NVARCHAR and NCLOB data if the NLS_NCHAR_CHARACTERSET of the source database is not the same as that of the target database.

Default: 1

Range of valid values: 0 or 1 (flag)

Takes effect: when Post is restarted.

SP_OPO_NLS_DEFAULT_COMPAT

This parameter controls whether Post converts character data (CHAR, VARCHAR). By default no conversion is performed. When set to 7, Post will not convert character data, but will convert CLOB data. The compare/repair commands will compare and repair character data without conversion and CLOB data with conversion. The replication of NCHAR data is not affected.

SP_OPO_NLS_DEFAULT_COMPAT takes precedence over SP_OPO_NLS_CONVERSION.

Default: 0

Range of valid values: 0 or 7

Takes effect: when Post is restarted.

SP_OPO_OBJID_DEBUG

This parameter allows debugging for one specific object, such as a table, for Post. The SP_OPO_OBJID_DEBUG parameter is typically used for debugging out-of-syncs. To use the SP_OPO_OBJID_DEBUG parameter, set its value on the target system to the value of the object ID for the object from the source database.

Default: 0

Range of valid values: Any number greater than or equal to 0

Takes effect: immediately

SP_OPO_ONELINE_DEBUG

This parameter controls the oneline debug feature for the Post process.

The oneline debug feature enables you to perform debugging for out-of-sync errors without consuming a large amount of the system resources. The oneline debug logs just enough information in one line to detect where in the data stream an out-of-sync condition occurred. Once oneline debug identifies the process that is causing the problem, you can then enable regular debugging for that process.

To enable this parameter, specify the objects that you want to debug by their object ID, and separate each one by a comma.

An example is:

```
sp_ctrl> set sp_opo_oneline_debug 230230, 351626
```

The following items are logged:

- The state of the operation, which can be **Dispatch** (read from queue), **ProcessMessage** (process the SQL code), **Skipped** (commit if the Commit Reduction feature is enabled) or **ExecSQL** (apply to target)
- The transaction identifier used by SharePlex
- The type of DML or DDL operation that was affected
- The object ID of the affected source table
- The row identifier of the affected row. (Oracle rowid)
- The time when the operation occurred (Oracle)
- The log sequence number and offset within the log of the affected operation
- The Oracle SCN for the transaction
- An internal SharePlex code that prevents redundant operations

Default: Disabled

Range of valid values: a string that forms a list of objects listed by object ID, separated by commas.

Takes effect: Immediately

SP_OPO_OUT_OF_SYNC_SUSPEND

This parameter controls what SharePlex does when it encounters an out-of-sync condition, based on the following:

- INSERT: The row already exists, resulting in a Unique constraint violation error.
- UPDATE and DELETE: The row is not in the database.

Value of 0 (default)

The default Post behavior when a transaction contains an out-of-sync operation is to continue processing other valid operations in the transaction to minimize latency and keep targets as current as possible. Latency is the amount of time between when a source transaction occurs and when it is applied to the target. Different factors affect the amount of latency in replication, such as unusually high transaction volumes or interruptions to network traffic.

Post logs the SQL statement and data for the out-of-sync operation to the `ID_errlog.sql` log file, where *ID* is the database identifier. This file is in the **log** sub-directory of the variable-data directory on the target system.

Value of 1

A value of 1 directs Post to stop posting when it encounters an out-of-sync condition. Post logs the SQL statement and data for the out-of-sync operation to the *ID*_**errlog.sql** log file, and then stops.

When you set this parameter to a value of 1, check the status of the replication processes frequently. Stopping Post causes latency between source and target databases, and it causes data to accumulate in the replication queues, which could cause them to exceed available disk space. SharePlex provides several **sp_ctrl** commands for checking replication status, and it provides tools for unattended SharePlex monitoring. See the SharePlex Administration Guide for more information about how to monitor SharePlex.

See also SP_OPO_SAVE_OOS_TRANSACTION to configure Post to roll back the entire transaction if it contains any out-of-sync operations.

Default: 0 (do not stop for out-of-sync conditions)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OPO_POSTER_DELAY

This parameter controls the amount of time that the Post process delays before it posts records to the target instance. Normally, Post applies the changes immediately to minimize latency between the source and target systems, but you can delay posting for up to 5 days (7200 minutes).

Delaying posting helps protect the data against accidental loss caused by unwanted deletes or object drops on the source system. The delay gives you enough time to detect the mistake and retrieve the data from the target instance before the mistake is replicated. Running “what-if” analyses is another reason you could change the default of this parameter. Having the target database behind in time enables you to validate predictive modeling compared to the real thing.

The delay caused by SP_OPO_POSTER_DELAY is measured from the time the message first appears in the redo logs.

Things to consider when using this parameter include the following:

- Make certain that there is enough disk space for the data to collect in the post queue for the designated time interval until it is applied to the database.
- To determine when to start posting based on the parameter’s setting, SharePlex compares the target system’s current timestamp with the time that a record enters the redo log on the source system. Those machines could be in different locations, perhaps thousands of miles from each other. Consider any difference between time zones, and add that to the delay time.

  For example, there is an 18-hour time difference between Los Angeles, California, U.S.A., and Sydney, Australia. To delay posting for five hours, you would need to set SP_OPO_POSTER_DELAY to 23 hours (1380 minutes) to account for the desired five-hour delay plus the 18-hour time difference. Also consider whether or not a location observes Daylight Savings Time, which could change the time difference between two locations. The way that Daylight Savings Time is observed varies among, and even within, nations that use it.
- Do not use this parameter going back in time, for example when a source system is in Sydney and a target system is in Los Angeles. To SharePlex, this as a negative time difference when the time stamps on each system are compared. For example, 2:00 in the morning on April 2 in Sydney is 8:00 in the morning on April 1 in Los Angeles. The parameter does not accommodate settings of less than 0.

Default: 0 minutes

Range of valid values: 0 to 7200 minutes

Takes effect: immediately

SP_OPO_PRB_MISMATCH_SUSPEND

This parameter, when set to 1, will cause Post to stop when there is a mismatch on a partial rollback. This allows the problem to be investigated and resolved before Post resumes processing.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OPO_READRELEASE_INTERVAL

This parameter controls how often Post read/releases. Normally, Post performs a read/ release after it receives each COMMIT, which means it purges that transaction's data from the queue as part of the checkpoint recovery process. For smaller transactions, this causes excessive I/O on the target system and slows Post. If most transactions are small, you can set this parameter so that Post read/releases after a certain number of messages instead of after every COMMIT.

When you set a read/release interval, be aware that if new data does not follow a committed transaction (for example, if it was the last transaction of the day or there is no more user activity), Post processes the COMMIT and waits a certain amount of time, which is controlled by the internal SP_OPO_IDLE_LOGOUT parameter. If no data arrives, Post performs the read/release even though the SP_OPO_READRELEASE_INTERVAL interval is not satisfied. The number of messages in the post queue reduces to 0, indicating that Post is finished processing all messages from the queue.

NOTE: Because the Post process has multiple threads, the number of messages is associated with each thread instead of the queue. Thus, the read/release interval can be longer than expected, and you might need to lower the default value.

Default: 100

Range of valid values: any positive integer

Takes effect: immediately

SP_OPO_REDUCED_KEY

This parameter controls the content of the Post WHERE clause. Post uses a WHERE clause to find the row in the target that needs to be changed by a replicated UPDATE from the source. Different SharePlex features may require more or less data to be used in the Post WHERE clause.

- If set to 0, this parameter directs Post to construct a WHERE clause of all of the data that is sent by Capture. The data that Capture sends depends on the setting of the SP_CAP_REDUCED_KEY parameter. If you want the WHERE clause to include the values of the keys and all of the columns other than LONGs or LOBs, set both parameters to 0.

  IMPORTANT! In a peer-to-peer configuration, a setting of 0 is required.

- If set to 1, this parameter directs Post to build a WHERE clause with the key values and the before values of the columns that changed. This setting provides a good balance between replication performance and target data integrity, because it enables Post to perform a before-and-after comparison of the changed columns when constructing the WHERE clause. To be able to use this setting, the SP_CAP_REDUCED_KEY parameter must be set to 0 or 1. For more information about how SharePlex uses before and after values, see the SharePlex Administration Guide.
- If set to 2, this parameter directs Post to build a WHERE clause of only the key columns. This setting can be used to maximize posting performance. Because this setting omits the before-and-after comparison of the changed columns, you should ensure that no process or user can make changes to the target data except SharePlex. Additionally, it is recommended that you perform regular integrity verification by using the **compare** command.

  NOTE: When the SP_OPX_REDUCED_KEY parameter is set to 2 in the horizontally partitioned replication, Poster will prepare the WHERE clause with keys and before data of columns that is added for the UPDATE clause.

See also SP_CAP_REDUCED_KEY.

Default: 0

Range of Valid Values: 0, 1, 2

Takes effect: When Post is restarted

SP_OPO_RETRIES_MAX

This parameter controls the number of times that the Post process attempts to post a SQL statement that failed the first time.

Post will retry certain failed operations when there is the possibility that they will succeed with another attempt. The main operations that Post will retry are TNS write failures, connection failures, or locks on tables when Post needs to apply a TRUNCATE.

To increase the likelihood that the failed operations are successful, you can increase the SP_OPO_RETRIES_MAX parameter so that Post tries the operation more times. At the same time, increase the SP_OPO_RETRY_DELAY_TIME parameter to increase the time interval between the attempts. That gives the lock or other blocking operation enough time to be resolved between attempts.

If the Post process is set to continue on error (SP_SYS_SUSPEND_ON_ERROR=0) or if the error message is listed in the **oramsglist** file, Post moves on to the next transaction in the queue. In all other cases, Post stops after it reaches the maximum allowed attempts.

NOTE: for more information about the **oramsglist** file, see the SharePlex Administration Guide.

IMPORTANT: Reducing this parameter can cause the data to accumulate in the queues, possibly causing them to exceed the available disk space.

Default: 10 times

Range of valid values: 0 or any positive integer

Takes effect: immediately

SP_OPO_SAVE_OOS_TRANSACTION

This parameter controls whether or not Post rolls back and discards a transaction if it contains any out-of-sync operations. This functionality is controlled by the `SP_OPO_SAVE_OOS_TRANSACTION` parameter.

How to use this parameter

When this parameter is set to 1 and a transaction contains any operations that generate out-of-sync errors, Post discards the entire transaction and saves all the operations of that transaction to a SQL file. You can edit this file to repair the problem with the failed SQL and then run the file to apply the transaction to the target database.

When this parameter is set to 1, Post will continue to process valid transactions that follow a rolled back transaction by default. This is to prevent target latency. To configure Post to stop after rolling back a transaction, set the SP_OPO_OUT_OF_SYNC_SUSPEND parameter to 1.

IMPORTANT! This parameter should only be used if you know that your applications make all interdependent changes *within one transaction*.

All of the tables involved in the transaction that is being rolled back must contain only the following data types:

- CHAR – US7ASCII
- VARCHAR – US7ASCII
- NUMBER
- DATE
- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE
- INTERVAL
- ROWID
- RAW
- BINARY FLOAT
- BINARY DOUBLE

Each rolled back transaction has its own SQL file. The file name is *SCN_queue*.**sql**, where:

- *SCN* is the commit System Change Number (SCN) of the transaction.
- *queue* is the name of the Post queue that contains the transaction.

Example file name:

4346118046_postq1.sql

NOTE:

- Each SQL file is stored in the **oos** subdirectory of the SharePlex variable-data directory. You may need to change the location of the files if you use the **ora_cleansp** utility to re-initiate the SharePlex environment. This utility deletes the transaction files when you run it. You can modify the location, size, and number of SQL files by using any of the options of the **file** category of the **target** command.
- When this feature is enabled, Post writes the whole transaction to the *SCN_queue*.**sql** file, instead of writing only the out-of-sync portion of the transaction to the **errlog.sql** file.
- This feature supports Oracle targets only.
- Post will still generate out-of-sync messages in the **statusdb** which will show when the **show statusdb** command is issued in **sp_ctrl**. You can use the **clear status** command to clear these status messages after you apply the transaction using the SQL file.
- A rolled back transaction cannot be "un-rolled back."

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: when Post is restarted

SP_OPO_SQL_CACHE_DISABLE

This parameter enables or disables the SQL Cache feature. By default, SQL Cache is on, and this parameter is set to 0. A setting of 1 disables SQL Cache. A setting of 3 disables SQL Cache for batch operations to reduce the amount of memory that Post uses. (A setting of 2 is not available.)

If you disable SQL caching, SharePlex prints the following message to the Event Log: `SQL Cache disabled`.

For more information about SQL Cache, see the SharePlex Administration Guide.

Default: 0 (on)

Range of valid values: 0, 1, or 3

Takes effect: when Post is restarted

SP_OPO_STOP_ON_DDL_ERR

This parameter controls whether or not the Post process stops when there is an error applying DDL. The default of 1 directs Post to stop for errors. An error usually indicates that the source component for which the DDL was executed does not exist in the target database, indicating the likelihood that subsequent DML changes will also fail. Stopping Post prevents the DML failures and enables you to correct the problem to keep the databases synchronized.

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OPO_SUPPRESS_DELETE

This parameter controls whether or not Post applies DELETE operations to the target. Suppressing DELETE operations may be appropriate in situations such as a data warehouse, where a row must exist centrally even though it is deleted from its source table. You can set and unset this parameter without modifying or activating a configuration file. Set it to 1 to suppress DELETES.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: Process restart

SP_OPO_SUPPRESSED_OOS

This parameter controls whether Post returns an out-of-sync message when it detects that a target row already contains the changes that are being replicated from the source. If this condition is met, Post discards the replicated SQL and does not write an out-of-sync message to the Event Log or the **errlog.sql log** file.

This parameter supports INSERT, UPDATE, and DELETE operations in the following cases:

- All values (including the key value) of a replicated INSERT match the existing values of a row in the target.
- The key value of an UPDATE matches a row in the target and the existing values in that row match the *after* (change) values that were replicated from the source.

NOTE: SharePlex *will* return an out-of-sync message if the target values do not match the replicated after values.

- The target row of a DELETE operation does not exist.

By default this parameter is enabled. However, you should still use tools that verify all of the target data, such as the **compare** command, because out-of-sync values can exist that are not detected by Post.

Default: 1

Range of valid values: 0 (disabled) or 1

Takes effect: Immediately

SP_OPO_SYNC_LOG_FREQUENCY

This parameter controls how often Post logs out-of-sync messages to the event log (**event_log** file in the variable-data directory). When Post detects an out-of-sync condition, it generates an out-of-sync event. If target target table

is so out-of-sync that the event log is filling up with these messages, you can use this parameter to control how frequently these messages are logged.

When this parameter is greater than 0, Post logs the first out-of-sync error and then logs out-of-sync messages only at the specified interval. Post continues posting valid data until the out-of-sync table can be re-synchronized.

Default: 1 (log every out-of-sync message)

Range of valid values: Integer greater than 0

Takes effect: Immediately

SP_OPO_THREADS_MAX

This parameter is used primarily in testing to view the behavior of a SQL thread when it is handling several subqueues. The minimum number of threads is four, which enables the Post process to create the main thread, the timekeeper thread, the signal waiter thread, and one SQL thread.

Default: 128 threads

Range of valid values: 4 threads (minimum) to 1024 threads (maximum)

Takes effect: When the Post process is restarted

SP_OPO_TRACK_COMMITS

This parameter controls whether or not the Post process tracks commits. If it is 1, Post will also insert a row for every commit.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: When the Post process is restarted

SP_OPO_TRACK_OPERATIONS

This parameter controls which DML is tracked by Post when maintaining a change history target database. By default, Post tracks all DML types. For example, to configure Post so that only inserts and updates are tracked, set the parameter to I/U.

Default: I/U/D (inserts, updates and deletes)

Range of valid values: Any combination of I, U, or D separated by a slash (/)

Takes effect: When the Post process is restarted

SP_OPO_TRACK_PREIMAGE

This parameter controls whether or not the Post process tracks the before image of inserts and updates or the after image of deletes. The before image for an insert and the after image for a delete includes the key values, with all other columns set to null.

You can set this parameter to I, U, or D, or any combination of those values separated by slashes, for example: I/U. When this parameter is used, Post applies two rows for each operation of the specified type. One has the column values of the before image, and the other has the values of the after image.

Default: off

Range of valid values: **I**, **U**, **D**

Takes effect: When the Post process is restarted

SP_OPO_TRUSTED_SOURCE

This parameter specifies the *trusted source system*, which contains the data that is considered to be the primary set of data in a peer-to-peer replication environment. The trusted host can be used in custom conflict resolution routines and is also the basis of one of the prepared routines provided with SharePlex. For more information about conflict resolution and peer-to-peer replication, see the SharePlex Administration Guide.

Default: None

Range of valid values: Any valid host name in the SharePlex replication environment

Takes effect: When the Post process is restarted

SP_OPO_UPDATE_SCN

This parameter controls the tracking of Oracle SCNs and allows SharePlex to recover quickly when operating in a high-availability cluster. When this parameter is enabled, SharePlex tracks SCNs in its internal transaction table. This enables SharePlex to activate to the lowest applied transaction and then reconcile to the correct transaction when there is a failover.

NOTE: Enabling this parameter disables the commit reduction feature of the Post Enhanced Performance feature. That feature is not supported when tracking SCNs.

Default: 0

Range of valid values: 0 (disabled) or 1 (enabled)

Takes effect: when Post is restarted

SP_OPO_USE_VARNUM

This parameter causes Post to use a varnum to enter numbers into the target database. It allows non-standard high precision values to be entered into a number column in the database.

Default: 0 (Off)

Range of valid values: 0 or 1 (flag)

Takes effect: when Post is restarted

SP_OPO_WAIT_MSG_DELAY

This parameter controls how long Post waits before generating a message indicating that it is killing old or stalled Oracle sessions. Those messages occur at the startup of Post.

Default: 300 seconds

Range of valid values: any positive integer up to 86400

Takes effect: when Post is restarted

SP_OPO_LOB_PERFORMANCE_ENABLE

This parameter is introduced as part of the LOB performance improvement feature, where SharePlex tries to process up to 32k of data as InRow LOB instead of OutOfRow LOB to speed up the Poster process.

Default: 1

Range of valid values: 0, 1

Takes effect: When Poster is restarted

Open Target Post parameters

These parameters are used by the SharePlex Post process when applying data to an Open Target (non-Oracle) target.

SP_OPX_BATCH_ENABLE

To maximize performance when applying multiple transactions of the same operation, the Post process can group the transactions and apply as a batch. The SP_OPX_BATCH_ENABLE parameter controls this feature and is enabled by default. SP_OPX_MAX_BATCH_MBYTES controls the maximum size of the batch.

Default: 1 (Enabled)

Range of valid values: 0 or 1

Takes effect: Immediately

SP_OPX_COMMIT_REDUCE_MSGS

Sets the threshold for the Commit Reduction component of the Post Enhanced Performance feature. The Commit Reduction feature enables you to configure Post to combine smaller transactions into larger ones. This reduces the number of commits and acknowledgments that must be processed. The smaller the transaction, the bigger the performance gain.

When the specified number of messages is reached, Post issues a commit. The commits of transactions whose transactional borders are within this span of messages are skipped, and those transactions are all committed as one transaction. Commit reduction is on by default. To disable it, set this parameter to a value of 1.

The parameter setting is not an absolute threshold. SharePlex will not break up a transaction across different combined transactions. Therefore, Post may need to exceed that threshold in order to include all of the operations and the commit of the last transaction in the group.

Default: 100 messages

Range of valid values: Any positive integer.

Takes effect: when Post is restarted

SP_OPX_CONT_ON_ERR

This parameter controls whether or not Post stops when it encounters specific ODBC errors. When this flag is set to the default of 0, Post stops for all ODBC errors. To configure Post to ignore certain errors and continue processing, list them in one of the following files (depending on the database) and then set SP_OPX_CONT_ON_ERR to 1.

mysqlmsglist

postgresmsglist

sqlservermsglist

sybasemsglist

tdmsglist

These files are located in the **data** subdirectory of the variable-data directory. For more information about how to configure these files, see the SharePlex Administration Guide.

For each error that it ignores, Post will log an error in the SharePlex **errlog.sql** file and then resume posting. If Post receives an error that is not specified in the file, the process stops.

Default: 0 (stop on all errors)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OPX_DISABLE_OBJECT_NUM

This parameter prevents Post from posting replicated DML and DDL operations to the target, based on the object ID of the **source** table. You can set this parameter if the data in a source table is invalid or corrupted, if the source table contains data types that are not supported on the target, or for any other reason that you do not want operations for a table to be reflected in the target database.

You can set this parameter for one or more tables. It prevents further replication activity on the target table(s) of a source table until you have time to resynchronize the data and reactivate the configuration file. The Post process discards all replicated messages for these tables from the post queue, and the messages do not accumulate in the queue.

Use the parameter with caution. If it is enabled and DDL or DML is executed for the source table(s), the target data will be out of date because the changes are not posted. If there are dependencies on the table(s), such as a foreign key in other tables outside the replication configuration, disabling posting will prevent the dependencies from being satisfied.

This parameter is disabled by default. To enable it, issue the following command on the target system, where the numbers shown are the object IDs of the source tables to exclude from posting.

```
sp_ctrl(sysB)> set param SP_OPX_DISABLE_OBJECT_NUM 12345,67890
```

Separate each object ID with a comma, and allow no spaces between them. To use spaces between entries, enclose the entire list within quotes, as in the following example:

```
sp_ctrl(sysB)> set param SP_OPX_DISABLE_OBJECT_NUM "498438, 1000, 497109"
```

Invalid object IDs will be ignored.

When you are ready to begin posting to the target table again, set SP_OPX_DISABLE_OBJECT_NUM to 0.

Default: 0 (off)

Range of valid values: list of valid object IDs separated by commas

Takes effect: immediately

SP_OPX_MAX_BATCH_MBYTES

This parameter controls the maximum size of a batched DML operation. To maximize its performance when applying SQL operations to the target, the Post process groups smaller replicated operations into one larger batched operation. Post groups replicated operations when the operations are identical and the SQL is similar. Post

continues to add operations (with matching criteria) to the batch until the batch size specified by SP_OPX_MAX_BATCH_MBYTES is reached.

The SP_OPX_BATCH_ENABLE parameter enables this feature (enabled by default).

Default: 1 megabyte

Range of valid values: 1 to any positive integer

Takes effect: Immediately

SP_OPX_MSGS_IN_MEMORY

This parameter controls the amount of memory that the Post process uses. The Open Target Post process uses an internal queue to store transaction data temporarily. This parameter controls the maximum amount of memory that this internal queue uses.

Default: default 5000 messages

Range of valid values: integer, no maximum

Takes effect: when Post is restarted

SP_OPX_NLS_CONVERSION

This parameter controls character set conversion between an Oracle source and a non-Oracle target.

When replicating to an Open Target target (non-Oracle target), SharePlex supports replication from any Oracle Unicode character set and the US7ASCII character set. SharePlex posts data to Open Target in the Unicode character set, and therefore if the source data is Unicode or US7ASCII, no conversion on the target is required.

However, if the following are true, conversion is required on the target:

- If the character set of the source data is anything other than Oracle Unicode or US7ASCII, you must install an Oracle client on the target to perform the conversion to Unicode for posting to the target.
- If the data must be posted to the target database in any character set other than Unicode, you must install an Oracle client on the target to perform the conversion and use the **target** command to identify the target character set for Post to use.
- If you are replicating LOB data, conversion is required regardless of what the source character set is.

To perform conversion with an Oracle client on Linux:

1. Install an Oracle *Administrator* client on the target system. The client must be the Administrator installation type. The Instant Client and Runtime installation types are not supported.
2. Set ORACLE_HOME to the client installation. Set ORACLE_SID to an alias or a non-existing SID. SharePlex does not use them and a database does not have to be running.
3. SharePlex using the Linux/Unix installer for your operating system.
4. Make certain the SP_OPX_NLS_CONVERSION parameter is set to the default of 1.

To perform conversion with an Oracle client on Windows:

1. Install an Oracle Administrator client on the target system. The client must be the Administrator installation type. The Instant Client and Runtime installation types are not supported.
2. In the `SharePlex Registry key \HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\QuestSoftware\SharePlex\port_number`, set ORACLE_HOME to the Oracle client installation location, and set ORACLE_SID to an alias or a non-existing SID. No Oracle database is required. SharePlex only needs to use the client libraries.
3. Install SharePlex using the Windows installer.
4. Make certain the SP_OPX_NLS_CONVERSION parameter is set to the default of 1.

To apply Unicode and US7ASCII data without conversion:

If the source data is Unicode or US7ASCII and you are not replicating LOB data, no conversion or Oracle client is needed. Set the SP_OPX_NLS_CONVERSION parameter to 0 to disable conversion, and then restart Post if it is running.

Default: 1

Range of valid values: 0 or 1 (flag)

Takes effect: when Post is restarted.

SP_OPX_ONELINE_DEBUG

This parameter controls the oneline debug feature for the Open Target Post process.

The oneline debug feature enables you to perform debugging for out-of-sync errors without consuming a large amount of the system resources. The oneline debug logs just enough information in one line to detect where in the data stream an out-of-sync condition occurred. Once oneline debug identifies the process that is causing the problem, you can then enable regular debugging for that process.

To enable this parameter, specify the objects that you want to debug by their object ID, and separate each one by a comma.

An example is:

```
sp_ctrl> set sp_opo_oneline_debug 230230, 351626
```

The following items are logged:

- The state of the operation, which can be **Process** (read from queue), **Skipped** (commit if the Commit Reduction feature is enabled) or **Applied** (apply to target)
- The transaction identifier used by SharePlex
- The type of DML or DDL operation that was affected
- The object ID of the affected source table
- The row identifier of the affected row. (Oracle rowid)
- The time when the operation occurred (Oracle)
- The log sequence number and offset within the log of the affected operation

- The Oracle SCN for the transaction
- An internal SharePlex code that prevents redundant operations

Default: Disabled

Range of valid values: a string that forms a list of objects listed by object ID, separated by commas.

Takes effect: Immediately

SP_OPX_OUT_OF_SYNC_SUSPEND

This parameter controls what SharePlex does when it encounters an out-of-sync condition, based on the following:

- INSERT: The row already exists, resulting in a Unique constraint violation error.
- UPDATE and DELETE: The row is not in the database.

Value of 0 (default)

The default Post behavior when a transaction contains an out-of-sync operation is to continue processing other valid operations in the transaction to minimize latency and keep targets as current as possible. Latency is the amount of time between when a source transaction occurs and when it is applied to the target. Different factors affect the amount of latency in replication, such as unusually high transaction volumes or interruptions to network traffic.

Post logs the SQL statement and data for the out-of-sync operation to the `ID_errlog.sql` log file, where *ID* is the database identifier. This file is in the **log** sub-directory of the variable-data directory on the target system.

Value of 1

A value of 1 directs Post to stop posting when it encounters an out-of-sync condition. Post logs the SQL statement and data for the out-of-sync operation to the *ID*_**errlog.sql** log file, and then stops.

When you set this parameter to a value of 1, check the status of the replication processes frequently. Stopping Post causes latency between source and target databases, and it causes data to accumulate in the replication queues, which could cause them to exceed available disk space. SharePlex provides several **sp_ctrl** commands for checking replication status, and it provides tools for unattended SharePlex monitoring. See the SharePlex Administration Guide for more information about how to monitor SharePlex.

Default: 0 (do not stop for out-of-sync conditions)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OPX_READRELEASE_INTERVAL

This parameter is used when Post is configured to post to a JMS server in *transactional* style (issue a JMS commit at intervals, rather than after every message as directed by the **session_transacted** property of the **target** command).

This parameter controls the read/release interval, which is how often Post purges the data of a transaction from the queue.

Default: 1,000

Range of valid values: any positive integer

Takes effect: when Post is restarted

SP_OPX_REDUCED_KEY

This parameter controls the content of the Post WHERE clause. Post uses a WHERE clause to find the row in the target that needs to be changed by a replicated UPDATE from the source. Different SharePlex features may require more or less data to be used in the Post WHERE clause.

- If set to 0, this parameter directs Post to construct a WHERE clause of all of the data that is sent by Capture. The data that Capture sends depends on the setting of the SP_CAP_REDUCED_KEY parameter. If you want the WHERE clause to include the values of the keys and all of the columns other than LONGs or LOBs, set both parameters to 0.

  IMPORTANT! In a peer-to-peer configuration, a setting of 0 is required.

- If set to 1, this parameter directs Post to build a WHERE clause with the key values and the before values of the columns that changed. This setting provides a good balance between replication performance and target data integrity, because it enables Post to perform a before-and-after comparison of the changed columns when constructing the WHERE clause. To be able to use this setting, the SP_CAP_REDUCED_KEY parameter must be set to 0 or 1. For more information about how SharePlex uses before and after values, see the SharePlex Administration Guide.
- If set to 2, this parameter directs Post to build a WHERE clause of only the key columns. This setting can be used to maximize posting performance. Because this setting omits the before-and-after comparison of the changed columns, you should ensure that no process or user can make changes to the target data except SharePlex. Additionally, it is recommended that you perform regular integrity verification by using the **compare** command.

  NOTE: When the SP_OPX_REDUCED_KEY parameter is set to 2 in the horizontally partitioned replication, Poster will prepare the WHERE clause with keys and before data of columns that is added for the UPDATE clause.

See also SP_CAP_REDUCED_KEY.

Default: 0

Range of Valid Values: 0, 1, 2

Takes effect: When Post is restarted

SP_OPX_SQL_CACHE_DISABLE

This parameter enables or disables the SQL Cache feature. By default, SQL Cache is on, and this parameter is set to 0. A setting of 1 disables SQL Cache. A setting of 3 disables SQL Cache for batch operations to reduce the amount of memory that Post uses. (A setting of 2 is not available.)

If you disable SQL caching, SharePlex prints the following message to the Event Log: `SQL Cache disabled`.

For more information about SQL Cache, see the SharePlex Administration Guide.

Default: 0 (on)

Range of valid values: 0, 1, or 3

Takes effect: when Post is restarted

SP_OPX_STOP_ON_DDL_ERR

This parameter controls whether or not the Post process stops when there is an error applying DDL. The default of 1 directs Post to stop for errors. An error usually indicates that the source component for which the DDL was executed does not exist in the target database, indicating the likelihood that subsequent DML changes will also fail. Stopping Post prevents the DML failures and enables you to correct the problem to keep the databases synchronized.

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_OPX_THREADS

This parameter controls the Transaction Concurrency component of the Post Enhanced Performance feature. When set to a value of 2 or greater, Post runs with the specified number of threads (concurrent operations).

This parameter supports SQL Server and PostgreSQL databases. Typically, it should be set to the number of cores in the processors on the system.

For more information about how to tune the performance of Post, see the SharePlex Administration Guide.

Default: 1 thread (disabled)

Range of valid values: Less than the Max_connection value

Takes effect: when Post is restarted

SP_OPX_TRUSTED_SOURCE

This parameter specifies the *trusted source system*, which contains the data that is considered to be the primary set of data in a peer-to-peer replication environment. This parameter is used in a prepared conflict resolution routine provided by SharePlex. The SP_OPX_TRUSTED_SOURCE parameter should be set to same value on both servers. For more information about conflict resolution and peer-to-peer replication, see the SharePlex Administration Guide.

Default: None

Range of valid values: Any valid host name in the SharePlex replication environment

Takes effect: When the Post process is restarted

Queue parameters

These parameters control properties of the SharePlex queues.

SP_QUE_MAX_QUEUES

This parameter determines the maximum number of queues allowed per instance of **sp_cop**. By default, each capture and export queue uses 8 MB of shared memory and each post queue uses 32MB. If the system has enough memory, you can increase the value of this parameter. On startup, SharePlex displays the amount of shared memory that is required for the number of queues in the active configuration files.

Default: 25

Range of valid values: 9 - 254

Takes effect: When **sp_cop** is restarted

SP_QUE_POST_SHMSIZE

This parameter determines the specific shared-memory size for the post queue.

IMPORTANT: Increasing this parameter might require an increase in the maximum shared memory segment size in your system settings.

Default: 32 megabytes

Range of valid values: 1 to 2047

Takes effect: When **sp_cop** is restarted

SP_QUE_Q_SHMSIZE

This parameter determines the specific shared-memory size for the capture and export queues.

Default: 8 megabytes

Range of valid values: 1 to 2047

Takes effect: When **sp_cop** is restarted

SP_QUE_SHMDBUF

This parameter controls the shared-memory buffer size, which determines how much information the SharePlex processes read from, or write to, the queues at one time. The buffer is part of the SharePlex checkpoint recovery

process that facilitates fast, safe, asynchronous transport of data. The default setting is an optimal value and should not be changed without the guidance of a Technical Support representative or documented SharePlex procedure. However, it is acceptable to change this parameter if the majority of transactions are large. Try an initial setting of 65536. All values set for this parameter must be a power of 2.

IMPORTANT: Increasing this parameter to a value greater than 64k might require an increase in the maximum shared memory segment size in your system settings.

Default: 32,768 bytes

Range of valid values: any positive integer in powers of 2

Takes effect: when SharePlex is restarted

SP_QUE_SYNC

SP_QUE_SYNC instructs the queue module to verify that the writes of queue data to disk have been written to the disk media before returning. This is not the standard disk write model in Unix and Linux.

In the default OS disk I/O procedure, disk writes are written to an internal OS buffer cache and then are written to disk later. This action distributes the overhead of writing to disk so processes do not have to wait for the data to be written to disk. When the data blocks are written to the buffer cache, the applications accessing the data cannot distinguish the data written to the buffer cache from data written to disk. Unlike the Oracle COMMIT, all processes that have access to the file also have access to the data in the buffer cache. If a system crash should occur between the time the data blocks are written to the buffer cache and to the physical media, any data not written to the media is at risk for being lost upon the system recovery.

The SP_QUE_SYNC parameter is implemented as follows:

SP_QUE_SYNC=0

The OS default case described in the second paragraph is the disk write algorithm used for the SharePlex queue data.

SP_QUE_SYNC=1

Setting SP_QUE_SYNC to **1** causes the O_SYNC flag to be set upon opening each queue data and header file. This flag tells the OS not to return a write call until the data has been successfully written to disk. Without the sync flag ("normal" I/O), space would be allocated for the file data but the file data might not be written due to a system crash, cluster failover, or other critical problem that causes the OS to stop executing.

SP_QUE_SYNC=2 (SharePlex default setting)

Setting the SP_QUE_SYNC parameter to **2** does not turn on the O_SYNC flag. Instead, normal writes are done until a queue write COMMIT is called. As part of the write COMMIT, the queue module executes the system call fsync on each queue data file and then on the queue header files. This eliminates redundant sync operations on data that may be rewritten later by a later write COMMIT.

Impact of setting SP_QUE_SYNC

Setting SP_QUE_SYNC to a value that causes disk writes to complete before returning might have an impact on SharePlex performance and may slightly increase I/O processing for non-SharePlex processes. The amount of overhead is dependent upon the amount of data in the queue, the filesystem types, and the types of disk drive and disk controller in use.

What SP_QUE_SYNC does not do

There are many functions that the SP_QUE_SYNC parameter does not do. Among the most common issues:

- SP_QUE_SYNC does not choose between shared memory and disk. It also does not influence when the decision is made to write to disk. It only tells the OS to guarantee that the disk write is complete to the physical disk before returning.
- It does not prevent an “out of subqueues” error.
- It does not address the “out of disk space” error - the space for file data is allocated before the write or sync to physical media is performed. If an out-of-space error occurs, it would occur before the sync is performed. Data loss is likely, but the queue is expected to be usable once free space becomes available.
- It does not cause or prevent “out of shared memory” errors or “out of virtual memory” errors. Any message indicating loss of shared memory is most likely a failure in the queue module’s demand paging system. If an “out of virtual memory” error occurs, verify the OS parameters for data segment limits. If the usual or expected limits are not enough, contact Technical Support.

Using SP_QUE_SYNC in a cluster

In a cluster, setSP_QUE_SYNC on the primary node of the cluster (the node to which the shared variable-data directory is mounted) and on each source and target machine outside the cluster that is part of the same replication environment.

Default: 2

Range of valid values: 0 - 2

Takes effect: When SharePlex is restarted

SP_QUE_USE_SUBQUE_INDEX

This parameter improves queue performance. If the source has a high amount of concrrency, the post queue may contain numerous subqueues, each of which represents a transaction session. This may reduce the performance of the Post process. This parameter directs SharePlex to use improved indexing to access the subqueue structures. A message "Subqueue index enabled *queuename*" is written to the Event Log for every Post queue for which this parameter is enabled.

This parameter does not support VARRAYs. If you are replicating VARRAYs and this parameter is enabled, the parameter is ignored.

Default: 0

Range of valid values: 0 (disabled) or 1 (enabled)

Takes effect: When SharePlex is restarted

SP_SHS_SHMSIZE

This parameter determines the size of the shared memory allocated for storing statistics related to commands such as show post, show capture, and others.

Default: 32 megabytes

Range of valid values: 1 to 2047

Takes effect: When **sp_cop** is restarted

sp_cop parameters

These parameters are used by the SharePlex **sp_cop** program.

SP_COP_IDLETIME

This parameter determines the time interval during which **sp_cop** is inactive. The purpose of idle time is to conserve a machine's resources. In general, **sp_cop** should not be idle for long, because it plays a central role in replication. This parameter enables you to increase the idle time if **sp_cop** is using too much CPU time for overhead activities. A recommended value is 600 seconds to reduce the overhead.

Default: 60 seconds

Range of valid values: any positive integer

Takes effect: immediately

SP_COP_SCAN_TIMEOUT

This parameter sets the initial time interval sp_cop uses to scan its services table to determine if a process needs to be stopped or restarted.

As SharePlex continues to run, the interval might change, depending upon the history of each process.

Default: 300 seconds

Range of valid values: any positive integer

Takes effect: immediately

SP_COP_SO_RCVBUF

This parameter tunes the TCP/IP window size on the target machine. It works in conjunction with the SP_XPT_SO_SNDBUF parameter on the source machine to establish the size of a packet sent across the network. SharePlex references both parameters when TCP/IP sockets are created at the startup of sp_cop and the Export process.

If SharePlex is replicating across a WAN and the export queue is continually backlogged, try adjusting both parameters. SP_COP_SO_RCVBUF must be set equal to or greater than the value of SP_XPT_SO_SNDBUF, in multiples of 1024 bytes. To size the parameters, determine the ping time between the source and target machines, then use the following formula for both parameters:

```
param_value / ping_time= bytes per second
```

For example, if ping time is 200 milliseconds, and the value for the two parameters is 64K, SharePlex sends five 64K-packets every second, totaling 320K per second.

Unless you observe a bandwidth problem, Quest recommends leaving both parameters set to their defaults, which use the system's setting. To change SP_COP_SO_RCVBUF, set it on the target system, then stop and start sp_cop on that system.

If transfer still is slow, try increasing the SP_IMP_WCMT_MSGCNT and SP_IMP_WCMT_TIMEOUT parameters on the target system. Set SP_IMP_WCMT_MSGCNT to at least 10,000.

Default: 0 (default is set by the operating system)

Range of valid values: positive integers, in bytes, using multiples of 1024. Maximum is set by the operating system.

Takes effect: when SharePlex is restarted

SP_COP_TPORT

This parameter sets the TCP/IP port number for TCP/IP communications among Share- Plex sp_cop processes. All **sp_cop** processes replicating among each other must use the same TCP/IP port number. The default setting for SP_COP_TPORT is 2100; however, you may set a different port number during installation. After installation you may set a different port number by using the **set port** command. Thereafter, **sp_cop** uses the new port number.

Default: 2100

Range of valid values: any positive integer

Takes effect: when SharePlex is restarted

SP_COP_UPORT

This parameter sets the UDP port number for communication among SharePlex sp_cop processes. All **sp_cop** processes replicating among each other must use the same UDP port number. The default setting for SP_COP_ UPORT is 2100; however, you may set a different port number during installation. After installation you may set a different port number by using the **set port** command. Thereafter, **sp_cop** uses the new port number.

Default: 2100

Range of valid values: any positive integer

Takes effect: when SharePlex is restarted

Log parameters

These parameters are used to control properties of the SharePlex logs.

SP_SLG_LOG_MAXSIZE

This parameter controls the size of the SharePlex Event and Trace logs.

When the file reaches its maximum size, the following happens, depending on the log:

- The Event Log is renamed to **event_log_***n*, where *n* is the next number in the sequence of files, and then a new file is created as the active event log. For example, when the first event log reaches its maximum size, it is renamed to event_log_0000000001 and then a new file is opened as the active log. When that file reaches its maximum size, it is renamed to event_log_0000000002, and so forth.
- The Trace Log is truncated. Before writing to this log, SharePlex checks its size. If the size is larger than the value of this parameter, SharePlex truncates the file before writing to it.

Before increasing this parameter, make certain that you have enough disk space to accommodate the new file size.

Default: 500 MB

Range of valid values: 1 to 2048 MB in increments of 1 MB

Takes effect: when SharePlex is restarted

SNMP parameters

These parameters are used to control properties of SharePlex SNMP support.

SP_SLG_SNMP_ACTIVE

This parameter is a flag that enables or disables SNMP support.

SharePlex provides agent support for Simple Network Management Protocol (SNMP) on all Unix, Linux, and Windows platforms supported by SharePlex replication.

NOTE: SharePlex provides only **agent** support for SNMP. It only sends SNMP traps. SharePlex *does not* provide an SNMP signal daemon (SNMP manager) to intercept the traps. Use the SharePlex SNMP feature only if you have a Network Management Station (NMS) to manage SNMP signals. The SharePlex SNMP agent is named **snmptrap** and is installed with SharePlex in the **bin** sub-directory of the SharePlex product directory. Do not run this program.

Default: 0 (off)

Range of valid values: 0 or 1 (on)

Takes effect: when SharePlex is restarted

SNMP agent parameters

The following parameters configure the SNMP agent to communicate with the NMS. Each parameter must have a value if the SP_SLG_SNMP_ACTIVE parameter is enabled.

Parameter	Value
SP_SLG_SNMP_HOST	The name of the system (host) to which the traps will be sent
SP_SLG_SNMP_COMMUNITY	The community security string
SP_SLG_SNMP_MJR_ERRNUM	The major error number to be used by the traps
SP_SLG_SNMP_MNR_ERRNUM	The minor error number to be used by the traps

Custom MIB parameters

The following parameters specify required information for a custom MIB.

Parameter	Value
SP_SLG_SNMP_ENTERPRISE_OID	The enterprise object identifier to send with the trap. The default is 1.3.6.1.4.1.3.1.1 .
SP_SLG_SNMP_TRAP_OID	A custom object identifier to bind to the trap. The default is 1.3.6.1.2.1.1.1.0.
SP_SLG_SNMP_TRAP_PROGRAM	The name of the trap program. The default is iwsnmptrap.

SNMP trap parameters

The following parameters configure the SNMP agent to send traps for specific replication events. The message or error text for the event is included in the trap and is the same error that appears in the Event Log.

To enable an SNMP trap for an event, set the corresponding parameter to a value of 1. By default all traps are disabled (parameter value of 0).

Parameter	SharePlex Event
SP_SLG_SNMP_INT_ERROR	SharePlex logic errors and errors that cause processes to exit
SP_SLG_SNMP_SYS_ERROR	System-related errors encountered by SharePlex
SP_SLG_SNMP_ERROR	Other SharePlex errors
SP_SLG_SNMP_OUT_OF_SYNC	Replication is out of synchronization
SP_SLG_SNMP_STARTUP	SharePlex starts up
SP_SLG_SNMP_SHUTDOWN	SharePlex shuts down
SP_SLG_SNMP_LAUNCH	A SharePlex process starts
SP_SLG_SNMP_EXIT	A SharePlex process stops

System parameters

These parameters control system-related SharePlex properties.

SP_SYS_HOST_NAME

This parameter is for use in environments where one SharePlex variable-data directory is shared among multiple nodes, such as clusters and systems with alternate network interfaces. It provides SharePlex the correct logical host name (global cluster package name) when any of its processes issues a name lookup, superseding the local system name. The name set with this parameter enables SharePlex to migrate properly during failover.

- Set SP_SYS_HOST_NAME on the primary node and all secondary (adoptive) nodes before you activate the configuration, and configure it into the package that fails over.
- Set this parameter in the **.profiles** file on Unix and Linux systems and in the Windows Registry on Windows systems.
- On Unix and Linux systems, set this parameter through **sp_ctrl** in the shared variable-data directory. Setting SP_SYS_HOST_NAME through **sp_ctrl** sets it permanently in the variable-data directory, which is part of the failover, and it ensures that the package name exports before **sp_cop** starts when you are ready to begin replication.
- Set SP_SYS_HOST_NAME only on machines within the cluster.

When you issue **sp_ctrl** commands from a system affecting a clustered machine, use the name set with SP_SYS_HOST_NAME as the host in the [**on***host*] option, or set it as the default for sp_ctrl by using the host command.

For more information about how to set up SharePlex on clustered systems, see the SharePlex Installation and Setup Guide.

Default: none

Range of valid values: the character string of the package name

Takes effect: when SharePlex is restarted

SP_SYS_IN_SYNC

This parameter controls the way that SharePlex replicates tables that have key columns where a NULL value is allowed.

- When this parameter is set to the default of 0 (off), SharePlex uses the key to locate rows on the target system, whether or not any of those columns contain NULLs. This can cause an out-of-sync condition for UPDATEs and DELETEs if the key for a row contains NULLs. The row's uniqueness cannot be assured, and it is possible for SharePlex to change the wrong row on the target system.
- When this parameter is set to 1 (on), it directs SharePlex to use the key, but only if there are no NULLs in the row's key. If the key contains a NULL value, SharePlex uses all of the columns in the row as a simulated key to ensure that it locates the correct target row.

If you know that the keys in your application will never contain NULLs (although NULLs were declared), leave this parameter off, because it incurs more overhead with the added logic that it uses. Set this parameter on the source system, and set it before you activate the configuration.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: when Capture and Read are restarted

SP_SYS_JOB_HISTORY_RETENTION

This parameter defines how many days information about past **compare**, **compare using**, **repair**, **copy** or **append** jobs will be kept for the **job status** and/or **report** commands to display.

Default: 90 days

Range of valid values: 0 or any positive integer

Takes effect: immediately

SP_SYS_JOB_HISTORY_SIZE

This parameter controls the size of the jobs status database, which resides in the data sub-directory of the SharePlex variable-data directory. The value specified determines the maximum size in kilobytes (kB) allowed for the database. When the database size exceeds the value specified, entries are deleted, oldest first. The default value is 0, specifying unlimited.

Default: 0 (unlimited)

Range of valid values: 0 or any positive integer

Takes effect: immediately

SP_SYS_SUSPEND_ON_ERROR

This parameter controls whether or not the Capture or Post process stops when it encounters a system or internal SharePlex error. The default of 1 directs the process to stop when there is an error. A setting of 0 directs the process to continue but generate a message in the Event Log.

IMPORTANT: If Capture is stopped for too long, it can lose pace with Oracle, and the redo logs can wrap. If that happens, Capture reads the archive logs, but if the archive logs become unavailable, you will need to resynchronize the data.

Default: 1 (stop on errors)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately

SP_SYS_TARGET_COMPATIBILITY

This parameter enables you to preserve compatibility between different versions of SharePlex to allow for smoother migrations and upgrades. As SharePlex continues to evolve to satisfy new user requirements, features or functionality added in a new version are not always compatible with previous versions. This parameter enables replication from a higher version of SharePlex on the source system to a lower version on a target system.

By default, this parameter is set to the version of the installed SharePlex software.

- When replicating from a higher version on a source system to a lower version of SharePlex on a target, set this parameter to a value that most closely matches the version of SharePlex on the target. For example, if SharePlex is version 8.6.3 on the source and 8.6.2 on the target, set this parameter to 8.6.2.
- When replicating between identical versions, leave this parameter set to the default on both systems.

Default: The version of the installed SharePlex software

Range of valid values: SharePlex release versions from 6.0.0 to the current version, up to three numbers long (for example 8.6.3).

Takes effect: when Capture is restarted

SP_SYS_VAR_FULL

This parameter sets a threshold for available space on the disk where the SharePlex variable- data directory is installed. It prevents the queues from exceeding available disk space. If an operation being processed by Capture, Read or Import will cause available disk space to drop below the value set for this parameter, the process stops. Processing resumes when available disk space reaches the threshold set with the SP_SYS_VAR_OK parameter.

Default: 30 MB

Range of valid values: any positive integer greater than the value of SP_SYS_VAR_OK

Takes effect: immediately

SP_SYS_VAR_OK

This parameter sets the amount of available disk space at which Capture, Read or Import resumes processing after it stopped because the value for SP_SYS_VAR_FULL was reached.

Default: 50 MB

Range of valid values: any positive integer

Takes effect: immediately

Compare/Repair Parameters

These parameters control properties of the SharePlex **compare** command.

SP_DEQ_IGNORE_ORACLE_ERROR_NUM

This parameter allows the user to specify an Oracle Error Number to ignore. The specified errror number will be ignored up to 500 times. This parameter is mainly for debugging purposes and should not be use with the **repair** or **repair1** option.

Default: 0 (none)

Range of valid values: Any valid Oracle Error Number, e.g. 1406

Takes effect: immediately available for the next comparison

SP_DEQ_IP_MAPPING

This parameter maps the IP addresses when Network Address Translation (NAT) is implemented between the source and target systems. When NAT is implemented, the target IP address on the source machine (for example, 192.168.32.10) and the IP address on the target machine (for example, 213.18.123.103) can be different, which can cause confusion when the **compare** and **repair** commands are issued. When you set the parameter SP_DEQ_IP_MAPPING on the target machine, the IP address 192.168.32.10 is considered equivalent to the IP address 213.18.123.103, thus allowing Poster to process the compare message and the compare **sp_declt** client process to finish the Compare process.

Set the SP_DEQ_IP_MAPPING parameter on the target system as in the following example:

```
sp_ctrl> set param SP_DEQ_IP_MAPPING "192.168.32.10:213.18.123.103"
```

NOTE: Use the appropriate IP addresses for your source and target machines. The IP addresses listed above are only an example.

In addition, the SP_DEQ_IP_MAPPING parameter accepts multiple mappings, separated by a comma, as in the following example:

```
sp_ctrl> set param SP_DEQ_IP_MAPPING "192.168.32.10:213.18.123.103,10.2.3.4:50.64.70.88"
```

Default: None

Range of valid values: one or more valid IP addresses, separated with a comma

Takes effect: When the next compare command is issued

SP_DEQ_LOG_FILESIZE

This parameter controls the size of the SQL log file that is generated by the compare server and the compare client.

Default: 50 MB

Range of valid values: Any size greater than 0.

Takes effect: Immediately available for the next comparison.

SP_DEQ_MALLOC

This parameter controls the fetch batch size. The batch size controls the number of rows that SharePlex selects at once for comparison. Larger batch sizes increase processing speed but require more memory. The value is divided equally by the number of compare threads to be used, and then the batch size is recalculated based on all column sizes added together.

Default: 500 (MB)

Range of valid values: 50 to 2047 (MB)

Takes effect: Immediately available for the next comparison.

SP_DEQ_PARALLELISM

This parameter manages the select statement Degree of Parallelism hint.

When SP_DEQ_PARRALLISM is set to zero, no parallel hint will be used. A user-provided hint from the command line will have higher precedence.

Default: 2

Range of valid values: 0 to 500

Takes effect: immediately available for the next comparison.

SP_DEQ_PARTIAL_REPAIR_CLR_OOS

This parameter enables the repair process to clear out-of-sync messages from the **statusdb** for a table after a compare-repair of only a subset of rows of that table, such as row subsets specified with a WHERE clause or a Oracle [sub]partition name (including horizontally partitioned replication).

IMPORTANT! Set this parameter on the target system. It has no effect on the source system.

This parameter works as follows:

- Value of 0 (disabled, the default): The out-of-sync messages are retained in the **statusdb** after a subset of the rows is compared and repaired. The messages are retained so that users are aware there may be out-of-sync rows that were not qualified by the selection criteria.
- Value of 1 (enabled): The out-of-sync messages are cleared after a subset of the rows is compared and repaired. Enable this parameter only if you know that the rows in the subset are the only ones in the table that are out-of-sync.

NOTE: To be certain all of the out-of-sync rows in a table are repaired, run the compare-repair for the entire table.

Default: 0 (disabled)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately available for the next comparison.

SP_DEQ_PART_TABLE_UPDATE

This parameter affects the behavior of the repair using and repair commands when they are issued for Oracle partitioned target tables.

- When this parameter is set to the default of 0, partitioned Oracle target tables are repaired using INSERTs and DELETEs only. Repairs requiring UPDATEs are converted to a DELETE followed by an INSERT to prevent errors when an UPDATE could cause a row to change partitions and row movement is not enabled for the table.
- When this parameter is set to 1, partitioned Oracle tables are repaired using INSERTs, UPDATEs, and DELETEs as appropriate. Use this mode only when you know UPDATEs will not result in a row changing partitions in the target table or when row movement is enabled for the target table.

Set this parameter on the **target** system.

Default: 0 (do not repair with UPDATEs)

Range of valid values: 0 or 1 (flag)

Takes effect: immediately available for the next comparison

SP_DEQ_READ_BUFFER_SIZE

This parameter applies only to LOB and LONG columns during a compare/repair operation. It adjusts the size of the buffer that holds the fetched LOB or LONG data when those columns are being compared and repaired. The value of the parameter should be adjusted based on the available system memory and data size.

Default: 1 MB

Range of valid values: any integer between 1 and 100 (values in MB)

Takes effect: Immediately available for the next comparison

SP_DEQ_ROW_LOCK_THRESHOLD

This parameter controls whether the repair process locks the entire target table or only the out-of-sync rows when performing a repair. When the number of out-of-sync rows is less than the specified value, only the out-of-sync rows are locked for the repair. This enables the Poster process to continue posting to other rows in the table. If the number of out-of-sync rows on the source table is less than this value, they are locked; otherwise, a brief lock on the table is taken to ensure a read-consistent value.

Default: 1000 out of sync rows

Range of valid values: Any positive integer

Takes effect: Immediately available for the next comparison

SP_DEQ_SKIP_LOB

This parameter determines whether or not LOBs are included in the compare/repair processing.

- When the parameter is set to the default of 0, the compare processes include LOBs in their processing.
- When the parameter is set to 1, only non-LOB columns are compared and repaired. If LOBs are not modified once inserted, you can speed up processing by setting this parameter to 1.

Set this parameter on the **source** system.

Default: 0

Range of valid values: 0 or 1 (flag)

Takes effect: Immediately available for the next comparison

SP_DEQ_THREADS

This parameter controls the number of processing threads used by the compare commands on the source system. Each thread creates a **sp_declt** process on the target machine.

The default of 2 threads has proven to be the best number for UP machines, but you might obtain performance improvements with up to 15 threads. Too many threads causes diminishing benefits, especially if it causes large tables to be compared at the same time.

The value for the thread count is independent of the number of tables to be compared, and SharePlex will not generate more threads than there are tables to be compared. If the machine has only one processor, set this parameter to a value of 1. Do not set it to more than the number of processors on the system.

Default: 2 threads

Range of valid values: 1 to 15

Takes effect: immediately available for the next comparison

SP_DEQ_TIMEOUT

This parameter controls the connection time out for the compare and repair commands. Because the **sp_desvr** process relies on the replication queues to instantiate the **sp_declt** process(es) on the target system, the **sp_declt** process cannot start until all previous messages in the queues have been processed. If the delay is longer than the time out specified with this parameter, **sp_desvr** exits and returns an error. If you know the queues are more than 30 minutes backlogged, you can increase this parameter as needed.

Default: 1800 seconds (30 minutes)

Range of valid values: 121 seconds or greater

Takes effect: immediately available for the next comparison

SP_DEQ_USE_SP_CKSUM

This parameter determines whether or not Oracle ora_hash() is used in compare.

- When this parameter is set to the default value of 0, database hashing functions are used.
- When this parameter is set to 1, the SharePlex internal checksum method is used.

NOTE: If compare and repair operations involve Oracle DB tables with BINARY_FLOAT, BINARY_DOUBLE, or LONG data types, comparisons are automatically done by setting the SP_DEQ_USE_SP_CKSUM parameter to 1. This can lead to increased CPU utilization, depending on the table structure and row size. To avoid this, users must exclude these columns from the comparison.

Default: 0

Range of valid values: 0 or 1

Takes effect: immediately available for the next comparison

SP_DEQ_ORA_DATE_FORMAT

Set the parameter to 1 at both the Oracle source and PostgreSQL target when you have the Oracle DATE data type mapped to the PostgreSQL TIMESTAMP(0) data type, as the Oracle DATE can store the timestamp portion. In this case, values will be read in the 'YYYY-MM-DD HH24:MI' format.

Set the parameter to 0 when you have the Oracle Date data type mapped to the PostgreSQL DATE data type, as only the date portion is stored by PostgreSQL. In this case, values will be read in the 'YYYY-MM-DD' format from the Oracle database, and the timestamp portion will be skipped during the compare/repair process.

If the same table has more than one date column, with one Oracle Date mapped to the PostgreSQL Timestamp(0) and another to the PostgreSQL Date, data with mismatched column formats will be shown as out of sync with either of the set column formats.

For example, when the parameter is set to 1 and the Oracle Date data type is mapped to the PostgreSQL Date data type, the column will be shown as out of sync. If the Oracle Date data type is mapped to the PostgreSQL Timestamp (0) data type, the column will be shown as in sync.

Default: 1

Valid Values: 0 or 1 (flag)

Takes effect: Immediately available for the next comparison

Copy/Append command parameters

These parameters are used by the **sp_sync** processes associated with the SharePlex **copy**/**append** command.

SP_OSY_COMPRESSION

This parameter adjusts the data compression level.

Default: 6

Range of valid values: 0 to 9 (9 being the highest level of compression, 0 being no compression)

Takes effect: immediately available for the next sync

SP_OSY_LOCK_TIMEOUT

This parameter set the number of seconds that the synchronization process will wait to obtain a table lock.

Default: 2

Range of valid values: 0 to 900

Takes effect: immediately available for the next sync

SP_OSY_POST_TIMEOUT

This parameter sets the number of seconds that the synchronization process will wait for the Post process to be ready and the synchronization to begin.

Default: 1800

Range of valid values: 0 to 36000 (seconds)

Takes effect: immediately available for the next sync

SP_OSY_THREADS

This parameter sets the number of pairs of synchronization processing threads, export/import, between the source and target systems. This parameter is utilized by SharePlex sync processes, e.g. the copy/append commands.

Default: 5

Range of valid values: 1 to 32

Takes effect: immediately available for the next sync

Analyze config command parameters

These parameters control properties of the SharePlex **analyze config** command.

SP_ANL_CHECKPOINT_FREQ

This parameter is one of the parameters that can be used to define the frequency of the checkpoints that are issued by the **analyze config** command process. The process saves its state to disk at each checkpoint to allow for fast recovery in the event that the process stops or fails before it is finished with the analysis.

Default: 20000 messages

Range of valid values: any positive integer

Takes effect: immediately

SP_ANL_CHECKPOINT_TIME

This parameter is one of the parameters that can be used to define the frequency of the checkpoints that are issued by the **analyze config** command process. The process saves its state to disk at each checkpoint to allow for fast recovery in the event that the process stops or fails before it is finished with the analysis.

Default: 120 seconds

Range of valid values: any positive integer

Takes effect: immediately

SP_ANL_RUN_TIME

This parameter controls how long the **analyze config** command runs. The amount of time that the command runs determines the amount of transaction activity that is analyzed. This parameter sets a default that you can override with the *n* {**minutes** | **hours** | **days**} option at runtime.

Default: 4320 minutes

Range of valid values: 1 to 432000 minutes

Takes effect: immediately

Deprecated Parameters

This section contains descriptions of SharePlex parameters that are deprecated.

How SharePlex handles deprecated parameters

After an upgrade, SharePlex will continue to support the functionality of a deprecated parameter if it is set in your environment. If there is a replacement parameter or other action that you should take to complete the upgrade, you are notified in Event Log. To stop using a deprecated parameter, issue the **reset param** command in **sp_ctrl**.

Deprecated Capture parameters

Parameter	Version deprecated	Notes
SP_OCT_ASM_CONNECT_DESC	8.6	Due to enhancements made in database setup.
SP_OCT_ASM_SID	8.6	Due to enhancements made in database setup.
SP_OCT_ASM_USE_OCI	9.1	Replaced by SP_OCT_OLOG_USE_OCI.
SP_OCT_OLOG_THREAD	8.0.7	
SP_OCT_REPLICATE_DR_DDL	8.6.6	Renamed to SP_OCT_REPLICATE_TRIGGER. The old parameter is still functional for backward compatibility.
SP_OCT_TARGET_COMPATIBILITY	9.0	Renamed to SP_SYS_TARGET_COMPATIBILITY.

Deprecated Read parameters

Parameter	Version deprecated	Notes
SP_ORD_NLS_LANG	8.0	
SP_ORD_LOGIN_O	8.6	Due to the enhancements made in database setup.
SP_ORD_OWNER_O	8.6	Due to the enhancements made in database setup.
SP_ORD_SKIP_OBJECT	8.0.7	This functionality is now configured in the source configuration file through the use of the "**c.**" designation in the routing map. For more information about this feature, see the

Parameter	Version deprecated	Notes
		SharePlex Administration Guide.

Deprecated Oracle Post parameters

Parameter	Version deprecated	Notes
SP_OPO_CHARACTERSET	8.0	
SP_OPO_CONF_LOG	8.6.6	Renamed to SP_OPO_LOG_CONFLICT. The old parameter is still functional for backward compatibility.
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	

Deprecated Open Target Post parameters

Parameter	Version deprecated	Notes
SP_OPX_CHECKPOINT_FREQ	8.6.4	
SP_OPX_MAX_CDA	9.1	Now handled with the **target** command.

Deprecated Export parameters

Parameter	Version deprecated	Notes
SP_XPT_ALTERNATE_HOST	9.0	Run provision to change the shareplex artifacts if a machine name or ip address has changed

Deprecated Compare parameters

Parameter	Version deprecated	Notes
SP_DEQ_CHARACTERSET	8.0	
SP_DEQ_BATCHSIZE	8.5	
SP_DEQ_CONFIG_LINE_SIZE	8.5	
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	

Deprecated queue parameters

Parameter	Version deprecated	Notes
SP_QUE_SHMSIZE	8.0.7	

Other deprecated parameters

Parameter	Version deprecated	Notes
SP_CFP_DEBUG	8.0.7	
SP_COP_WALLET_PATH	8.6.3	

3

SharePlex Commands for PostgreSQL

The SharePlex commands configure, start, stop, control, and monitor the replication process. SharePlex commands are issued through the `sp_ctrl` interface.

SharePlex Commands at a Glance

This section provides an overview of all SharePlex commands, grouped according to the processes or functions they control.

For more information about SharePlex command authorization levels and security groups, see the SharePlex Administration Guide.

Configuration Commands

The configuration commands control all aspects of managing SharePlex configuration files.

List of configuration commands

Command	Auth. Level	Supported targets	Description
abort config	1	All	Immediately terminates replication for the designated configuration file whether or not data has posted. Deactivates the configuration and deletes queues and processes.
activate config	1	All	Begins replication of the data specified in a configuration.
copy config	2	All	Duplicates a configuration to edit and save as a new file.
create config	2	All	Creates a new configuration file in which you enter information about objects to be replicated.
deactivate config	1	All	Gracefully terminates replication of the data specified in a configuration, allowing data in the queues to be posted to the target database before the queues are deleted.
edit config	2	All	Modifies an existing configuration file using

Command	Auth. Level	Supported targets	Description
			the default text editor.
list config	3	All	Displays all of the configuration files on a source system, both active and inactive.
purge config	1	All	Removes data from queues generated by an active configuration, but does not deactivate the configuration or remove the queues themselves.
rename config	2	All	Assigns a different name to a configuration file.
verify config	3	All	Verifies that the objects in a configuration are valid and that the configuration is structured properly to ensure successful activation and replication.

Information commands

The information commands provide information about the replication environment. Use these commands when you are trying to resolve a replication problem or you want to view certain process conditions.

List of information commands

Command	Auth. level	Description
show capture	3	Displays brief or detailed statistics for the Capture process for use in tuning and problem solving.
show post	3	Displays brief or detailed statistics for the Post process for use in tuning and problem solving.
job status	3	Displays current status and history for **compare** and **repair** commands.
compare status	3	Displays the status and results of the **compare using** and **compare** commands.
repair status	2	Disaplys the status and results of the **repair** and **repair using** commands.

Partitioning commands

The partitioning commands create, manage, and display the partition schemes and row partitions that are being used for horizontally partitioned replication.

List of partitioning commands

Command	Auth. level	Supported targets	Description
add partition	2	All	Creates partition schemes and row partitions.

Command	Auth. level	Supported targets	Description
drop partition	**2**	All	Removes a row partition from a partition scheme.
drop partition scheme	**2**	All	Removes a partition scheme.
modify partition	**2**	All	Modifies a row partition of a partition scheme.
view partitions	**3**	All	Shows existing partition schemes and their row partition specifications.

Synchronization commands

The synchronization commands control aspects of data synchronization.

List of synchronization commands

Command	Authorization level	Supported targets	Description
Compare / compare using	2	PostgreSQL	Compares the contents of source and target tables to verify synchronization.
Cancel	2	PostgreSQL	Cancels a running **compare** or **repair** command job.
Repair / repair using	2	PostgreSQL	Repairs out-of-sync rows in target tables to bring source and targets back into synchronization.
flush	2	All	Puts a marker in the data stream to establish a point of synchronization between the source and target data, then stops the Post process at that point.
reconcile	1	All	Synchronizes the results of a backup or copy applied to the target system with user transactions from the source system.

Maintenance commands

The maintenance commands control command displays and files.

List of maintenance commands

Command	Authorization level	Supported targets	Description
Clear History	2	PostgreSQL	Removes the job information and source logs for runs of the **compare**, **compare using**, and **repair** commands.
Clear Status	2	PostgreSQL	Removes warning messages from the Status Database.

Alphabetical Reference for SharePlex Commands for PostgreSQL

This starts the detailed reference documentation for SharePlex commands in alphabetical order.

Abort Config for PostgreSQL

Use the **abort config** command to deactivate a configuration instantly and delete all associated queues and replication processes at the same time. This command stops all replication activity for the configured datasource on the source and target systems, whether or not data in the queues has been posted. It is a forceful deactivation.

The **abort config** command is effective when there has been a system, configuration, or data problem and you need to prevent posting to the target system(s). Because you are deleting whatever data remains in the pipeline, the source and target instances will go out of synchronization after this command takes effect.

If any SharePlex replication process stops before or during the **abort config** command, the command also stops. When the process starts again, the command resumes working. This allows **abort config** to work even if the network is temporarily down — it remains in the queues until the connection is restored.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issues for:	source and target systems
Related commands:	**deactivate config**, **purge config**

Syntax

Basic command
abort config *filename*

Syntax description

Component	Description
filename	The name of the configuration that you want to abort. Configuration names are case-sensitive.

Component	Description
	Example: `sp_ctrl(sysA)>`**abort config sales**

Activate Config for PostgreSQL

Use the **activate config** command to activate a configuration. Replication begins immediately as soon as the activation process is complete.

The activation process reads the configuration file, from which it gets all of the information needed for SharePlex to:

- Identify the objects that are in replication
- Route the replicated changes to the appropriate source and target database
- Generate the SQL that Post uses to apply the changes to the target
- Activate all of the tables that have been added to replication

The process that **sp_cop** calls to activate a configuration is **sp_tconf**.

NOTES:

- While activating the configuration using logical replication, if any uncommitted transactions are being executed on the PostgreSQL instance, you may experience lag in the config activation.
- For partitioned tables, users must set the replica identity to full for all its sub-partitions. If not set, verify config will display a message as "Object may not be replicated because replica identity is not full for its partitions." Activation will not add the partitioned table in replication. It will log all the sub-partition table names that do not have replica identity set to full in the activation log.

Guidelines for activation

- To activate a configuration, the database containing the objects to be replicated must be mounted and open. The length of time that activation takes varies, depending on the size, number and structure of the configured objects.
- You can activate one configuration *per PostgreSQL server* on each system. For example, if there are ConfigA, ConfigB and ConfigC for instance `dbname1`, you can activate only one of them at a time. Activating another configuration for the same datasource automatically deactivates the first one.
- Before you activate a configuration, use the **verify config** command to confirm that basic requirements for successful activation and replication have been satisfied. The command alerts you to potential problems that can cause the activation to fail.

View activation status and results

SharePlex activates objects according to their object ID, not their order in the configuration file, so there is no way to predict the order of activation.

To view the results of activation, issue the **show config** command

What to do if activation fails

Many things can cause the activation of a table or the entire configuration to fail. For example, if one or more components in the configuration file were entered incorrectly, activation of the affected objects fails.

If you did not issue the **verify config** command before you activated, run it now, and correct any problems that it finds. Then, try activating again.

Usage

Supported sources:	PostgreSQL (pglsn option only applicable for physical replication.) , Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Authorization level:	Administrator (1)
Issues for:	source system
Related commands:	**abort config, copy config, create config, deactivate config, edit config, list config, purge config, remove config, rename config, show config, verify config, view config**

Syntax

Basic command	Command option
activate config *filename*	`pglsn=lsn_value`

Syntax description

Component	Description
filename	Required. The name of the configuration that you want to activate. Configuration names are case-sensitive. **Example:** `sp_ctrl(sysA)>`**activate config sales**
pglsn=lsn_number	Use this option to activate the configuration to start replication at a specific LSN in the WAL files. Before activating the configuration, do the following: If there was a previously active configuration, run the pg_cleansp utility on the source and all targets to restore the environment to a clean state. For more information, see pg_cleansp. Use the `show last_posted` command to get the LSNs of the last transactions that were posted from all the Post processes (if using named queues). Use the lowest of those LSN

Component	Description
	values for activate config. **Example:** `sp_ctrl> activate config myconfig pglsn=6/555FAE0` **NOTE:** Activate with LSN is not supported for logical replication. If database goes out of sync, user need to sync the data manually using external utilities.

NOTE: After successful activation physical and logical slots will get created in the database.

IMPORTANT!

Steps to follow when performing the alter table add/drop column operation which is part of ongoing replication.

CAUTIONS:

- Do not stop the Capture process while altering the table.
- No DML operations should be performed on the table for which column is being added or dropped.

Follow the below steps:

1. Execute the query for alter table add/drop column without performing any DML on the table.
2. Reactivate the config file again so that the latest details are fetched and stored in the object cache of all the processes.
3. Once reactivation is done, then DML operations can be performed on the table.

Steps to follow when performing the alter table add/drop partition operation which is part of ongoing replication.

CAUTIONS:

- Do not stop the Capture process while altering the table. (applicable only if reactivation is being done)
- No DML operations should be performed on the table for which partition is being added or dropped.

Follow the below steps:

1. Execute the query for alter table add/drop partition without performing any DML on the table.
2. Reactivate the config file again or restart the Capture process so that the latest partition details are fetched and used by the Capture process.
3. Once reactivation or restart Capture process is done, then DML operations can be performed on the table.

Add Partition for PostgreSQL

Use the **add partition** command to add a row partition to a partition scheme when configuring horizontally partitioned replication. Issue an **add partition** command for each row partition that you want to create.

This command captures all of the information required to create the partition, including the following required components:

- The partition scheme name. To create a new partition scheme, specify the name in the **add partition** command that creates the first row partition for that scheme. SharePlex automatically creates the partition scheme. Then, specify that name when adding additional row partitions to that partition scheme.
- The routing for the rows that are specified in the row partition.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issues on:	source system
Related commands:	modify partition, drop partition, drop partition scheme, view partitions

Syntax

Basic command	Command options
add partition to*scheme_name* **set** {**condition =** *column_condition* | **route =** *routing_map*	[**and name =***partition_name*] [**and tablename =***target_table*] [**and description =***description*]

Syntax description

Component	Description
to *scheme_name*	**to** is a required keyword indicating the row partition is being added to *scheme_name*. *scheme_name* is the name of the partition scheme. The partition scheme is created by the first **add partition** command that you issue, which will also specify the first set of rows to partition. If you are making heavy use of horizontal partitioning, it may help to establish naming conventions for your partition schemes.
set	Required keyword that starts the definition of the row partition.
condition = *column_condition*	Creates a row partition based on a column condition. The condition must be in quotes. Use standard WHERE conditional syntax such as **((region_id = West) and region_id is not null)**.
route = *routing_map*	The route for this partition. This can be one of the following: **Partition based on a column condition:** Specify any standard SharePlex routing map, for example: **sysB@r.dbname** or **sysB:q1@r.dbname** or **sysB@r.testdb+sysC@r.testdb** (compound routing map). **To route a partition to multiple target tables that have different names, do the following:** • Issue a separate **add partition** command for each different target name. Use the **tablename** option to specify the name. • In the configuration file, specify any of these target tables as the target table in the entry that uses this partition scheme. SharePlex will detect the other names when the configuration is activated. • Set the SP_ORD_FIRST_FIND parameter to 0 so that SharePlex checks all of the column conditions in the partition scheme. By default SharePlex assumes that any given row change will satisfy only one column condition in the partition scheme.
name = *name*	(Recommended) A short name for this partition. This option is only useful for partitions based on column conditions. A name eliminates the need to type out long column conditions in the event that you need to modify or drop the partition in the future.
tablename = *owner.table*	(Optional) Use this option when there are multiple target tables and one or more have different names. Issue a separate **add partition** command for each name. The table name must be fully qualified. If case-sensitive, the name must be specified in quotes. **Example:**

Component	Description
	add partition to scheme1 set name = p1 and condition = "C1 > 200" and route = sysb:p1@r.dbname and tablename = myschema.mytable
description = *description*	(Optional) Description of this partition.

Examples

Row partitions based on column conditions

Route different sets of rows through different post queues:

```
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
```

Route different sets of rows to different target systems and different table names from the source:

```
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
```

Cancel command for PostgreSQL

Use the **cancel** command to cancel a running **compare** and **repair** command job.

To cancel a job, you must supply its job ID. The job ID is reported back from the compare and repair job when you issue the command that starts it:

```
sp_ctrl>repair using 1elliot
repairing 7 of 7 objects
repair started; job id 408
```

Alternatively, you can get the job ID of the most recently streamed job by issuing the **job status** command:

```
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
```

Usage

Supported sources:	PostgreSQL
Supported targets:	PostgreSQL
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**compare**, **repair**

Syntax

Basic command	Remote options
cancel *job_id*	[**on** *host* |
	on *host:portnumber* |
	on *login/password@host* |
	on *login/password@host:portnumber*]

Syntax description

Component	Description
jobID	The ID of the job to be canceled. **Example**: `sp_ctrl>`**cancel 407**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Clear History command for PostgreSQL

Use the clean history command to remove the information and logs from old compare, and repair command jobs.

These jobs generate log files on the source and target systems. By default, the job information and log files are cleaned when the job is older than the value set with the **SP_SYS_JOB_HISTORY_RETENTION** parameter. The **clear history** command can be used to clear the job information and logs on demand for a specific job or table, or for all jobs that are of a specific age.

Usage

Supported sources:	PostgreSQL
Supported targets:	PostgreSQL
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**compare**, **repair**

Syntax

Basic command	Command options	Remote options
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*]

Syntax description

Component	Description
all	Causes all jobs and log history to be removed. **Example**: `sp_ctrl(sysA)` **> clear history all**
source_owner.source_table	Causes history for a particular table to be removed. History gets deleted as per **SP_SYS_JOB_HISTORY_RETENTION** parameter. **Example**: `sp_ctrl(sysA)` **> clear history clear history user2.employee**
age days	Causes the job history older than the specified number of days to be removed.

Component	Description
	Example: `sp_ctrl(sysA)` **> clear history age 10**
jobID	Causes the jobs for the specified job id (obtained using the job status command) to be removed. History gets deleted as per **SP_SYS_JOB_HISTORY_ RETENTION** parameter. Example: sp_ctrl(sysA)> **clear history jobID**
for r.*dbid*	Optional. Can be used to employ the **clear history** command when there is no active configuration, or if there are more than one active configurations. In either case, the source SID must be specified using the for option. **Example**: `sp_ctrl(sysA)` **>clear history all for r.dbid**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Clear Status command for PostgreSQL

Use the **clear status** command to remove old warning messages from the Status Database.

To use this command:

1. Issue the **show statusdb detail** command to find out which messages can be cleared. Clearable messages have a `Yes` in the **Clear** column.
2. If you don't want to clear all messages, make a note of the status ID of each one that you want to clear.
3. Issue the **clear status** command for each status ID, or use the **all** argument to remove all clearable messages at once.

SharePlex puts a message in the Event Log identifying the messages that were cleared.

This command clears messages from the Status Database for the default system. To clear messages from a Status Database on a different system, use the [**on***host*] option.

NOTE: Some messages cannot be cleared.

Usage

Supported sources:	PostgreSQL
Supported targets:	PostgreSQL
Authorization level:	Operator (2)
Issued for:	source or target system
Related commands:	**show statusdb**

Syntax

Basic command	Remote options
clear status {*statusID*| **all**}	[**on** *host* | **on** *host:portnumber* | **on** *login/password@host* | **on** *login/password@host:portnumber*]

Syntax description

Component	Description
statusID	The SharePlex-assigned ID of an individual message (obtained using the show statusdb command) that you want to remove. **Example**:

Component	Description
	`sp_ctrl(sysA)`**>clear status 20**
all	This argument removes all clearable messages. **Example**: `sp_ctrl(sysA)`**>clear status all**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Compare status command for PostgreSQL

Use the **compare status** command to view the status of the last compare or repair job run. The **compare status** command can be used to view detailed status on a compare or repair job or a portion of a compare or repair job, or to view status on all compare and repair jobs for which SharePlex has history.

NOTE: For details and examples about using the **compare status** command, see the **job status** command.

Usage

Supported sources:	PostgreSQL
Supported targets:	PostgreSQL
Authorization level:	Viewer (3)
Issues on:	source system
Related commands:	compare / compare using, repair / repair using

Syntax

Basic command	Command options	Remote options
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**]	

Syntax description

Component	Description
job status	Shows status of all compare and repair jobs for which SharePlex has history.
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**: `sp_ctrl(sysA)` **>job status 2828.2**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example**:

Component	Description
	`sp_ctrl(sysA)`**>job status 2828.HR.SRC_TEST3**
all	Displays a summary line for every job with history in the database. **Example**: `sp_ctrl(sysA)`**>job status all**
full	Displays the status of every object in the job. By default, the job status command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)`**>job status 2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)`**>job status detail**
status	Displays status history for previous jobs with the specified status. **Example**: `sp_ctrl(sysA)`**>job status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax.

Option	Description
	Example: `sp_ctrl(sysB)>` **status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>` **status on john/spot5489@SysA:8304**

Copy config for PostgreSQL

Use the **copy config** command to copy a configuration file and save the copy under a different file name.

Copying the configuration allows you to:

- Save a replica for backup purposes
- Use the copy as the basis for composing a new configuration using the **edit config** command
- Change an active configuration. You can make your changes to the copy with the **edit config** command while replication under the current configuration continues, and then activate the new configuration when you are ready.

This command copies the configuration to the same source system as the original configuration. To copy a configuration file to a *different* system, you can transfer the file via tape, CD, or FTP. To see a list of configurations on a system, use the **list config** command. To view the contents of a configuration, use the **view config** command.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issues on:	source system
Related commands:	**activate config, create config, list config, show config, verify config, view config**

Syntax

Basic command
copy config *filename* **to** *newname*

Syntax description

Component	Description
filename **to** *newname*	• *filename* is the name of the configuration that you want to copy. Configuration names are case-sensitive. • **to** is a required part of the syntax. • *newname* is the name you are giving to the copy. It must be unique among configurations on the system.

Component	Description
	Example: `sp_ctrl(sysA)>`**copy config sales to sales2**

Create Config for PostgreSQL

Use the **create config** command to create a new configuration file. This command runs the SharePlex default text editor and opens a new file. To create a configuration, follow the instructions in Chapter 5 of the *SharePlex Administrator's Guide.*

If you attempt to give a new configuration the same name as an existing configuration, SharePlex returns this error message "`The parameter for create config must be a new filename.`"

As an alternative to creating a configuration, you can use the **copy config** command to duplicate an existing configuration and then use the **edit config** command to edit the copy.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	**activate config, copy config**

Syntax

Basic command
create config *filename*

Syntax description

Component	Description
filename	The name of the configuration that you want to create. Configuration names are case-sensitive. The name must be unique among configurations on the system. **Example**: `sp_ctrl(sysA>`**create config sales**

Deactivate config for PostgreSQL

Use the **deactivate config** command to gracefully terminate replication for an active configuration. This command stops all Capture activity for the configuration, posts all data currently in the queues, removes the associated SharePlex processes and queues, and drops replication slots (physical/logical).

The **deactivate config** command does not remove a configuration from the system, but only stops replication of its objects until you activate it again. When you deactivate a configuration, it can cause the target data to go out of synchronization if users continue making changes to the configured source objects. Deactivating one configuration on a system does not deactivate other active configurations on the same system.

NOTE: In SharePlex 11.1, physical or logical slots created at activation will get dropped from the database after successful deactivation.

Usage

Supported sources:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Authorization level:	Administrator (1)
Issues on:	source system
Related commands:	**list config**

Syntax

Basic command
deactivate config *filename*

Syntax description

Component	Description
filename	The name of the configuration that you want to deactivate. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)>`**deactivate config sales**

Drop Partition for PostgreSQL

Use the **drop partition** command to remove a row partition from a partition scheme in a horizontally partitioned replication configuration. Issue a **drop partition** command for each row partition that you want to remove.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issues on:	source system
Related commands:	add partition, modify partition, drop partition scheme, view partitions

Syntax

Basic command

drop partition from *scheme_name*
where
{
{**condition =***column_condition*
route = *routing_map* |
name = *partition_name* |
tablename = *target_table* |
description = *description*
}

Syntax Description

Syntax Component	Description
scheme_name	The name of the partition scheme.
condition	Column condition that defines the rows of the partition that you want to remove.
route	The routing map of the partition that you want to remove.
tablename	The fully qualified name of the target table in the partition that you want to remove.
name	The short name of the partition that you want to remove.
description	The description of the partition that you want to remove.

Examples

```
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
```

Drop Partition Scheme command for PostgreSQL

Use the **drop partition scheme** command to remove a partition scheme from the SharePlex replication configuration. This command also removes all of the row partitions that are specified for the partition scheme.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issues on:	source system
Related commands:	add partition, modify partition, drop partition, view partitions

Syntax

Basic command
drop partition scheme *scheme_name*

Syntax Description

Syntax Component	Description
scheme_name	The name of the partition scheme.

Example

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

Edit config for PostgreSQL

Use the **edit config** command to modify an inactive configuration file. This command opens the configuration in SharePlex's default text editor. Modify and save the configuration using the tools provided by the editor.

To modify an active configuration, it is recommended that you copy the active configuration with the **copy config** command, and then edit the copy with the **edit config** command. When you activate the copy, the original configuration deactivates.

To view a list of configurations on a system, use the **list config** command. To view the contents of a configuration, use the **view config** command.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issued for:	source system
Related commands:	**activate config, copy config, create config, list config, rename config, verify config, view config**

Syntax

Basic command
edit config *filename*

Syntax description

Component	Description
filename	The name of the configuration that you want to edit. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)>`**edit config sales**

Flush for PostgreSQL

Use the **flush** command to stop the Post process at a certain point. It puts a marker in the data stream and automatically stops the Post process at the marker point. You can issue this command at a certain time or date – when month-end reports need to be generated, for example – and the data on the target system will reflect what was on the source system at the time the command was issued.

When you are performing backups, or when following certain documented SharePlex procedures, you issue the **flush** command after you stop user access on the source system to ensure that the last transaction gets posted to the target database before Post is stopped.

Starting Post again

Changes generated after the **flush** command accumulate in the queues and are applied to the target instance after you start Post again. Post remains stopped until the **start post** command is issued. Post keeps a record of where it stopped and resumes posting from that point to maintain synchronization.

IMPORTANT: If users continue making changes on the source system while Post is stopped, data will accumulate in the post queue and possibly consume all available disk space. Remember to start Post as soon as permissible.

NOTE: The **Status** command for post will show "stopped due to flush"

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issued for:	source system

Syntax

Basic command	Command options
flush*datasource*	[**to***host*] [**queue***queuename*] [**to***host***queue***queuename*] [**to***host@target_SID*] [**to***host@target_SID***queue***queuename*]

Syntax description

Component	Description
datasource	This argument specifies the source data that the command will affect. Without additional options, it flushes the data through all queues on all target systems and stops Post. datasource is expressed as **r.**dbname, where dbname is the database name of the source PostgreSQL instance. **Example**: `sp_ctrl(sysA)>`**flush r.dbname1** In the above example, the data for instance **r.dbname1** is flushed to all target systems.
to*host*	This option flushes all of the data replicating to a designated target system. Replication to other target systems is unaffected. **Example**: sp_ctrl(sysA)> **flush r.dbname1 to sysB** In the above example, the data is flushed from **dbname1** to **sysB**.
queue*queuename*	This option flushes data for a designated post queue. It flushes data flowing through all queues of that name on all target systems. This option is most useful when there are named post queues. **Example**: sp_ctrl(sysA)> **flush r.dbname1 queue sales** In the above example, the data is flushed for post queue *sales* on all target systems that have a post queue of that name.
to*host***queue** *queuename*	This option flushes data for a designated post queue on a designated target system. Other post queues on that and other target systems are unaffected. **Example**: sp_ctrl(sysA)> **flush r.dbname1 to sysB queue sales** In the above example, the data is flushed for post queue sales on sysB.
to*host*@*datadest*	This option flushes data to a designated target instance on a designated target system. It does not affect: • Other target instances on that system. • Other target instances with the same ORACLE_SID / database name on other target systems **In the syntax:** • *host* is the target system's name. • *datadest* is expressed as **o**.*SID*, where *SID* is the ORACLE_SID of the target instance. • **datadest** is expressed as **r.**dbname, where dbname is the database name of the PostgreSQL target instance.

Component	Description
	Examples: sp_ctrl(sysA)> **flush r.dbname1 to sysB@r.dbname2** (PostgreSQL to PostgreSQL implementation) In the above example, the data is flushed to database dbname2 on sysB. sp_ctrl(sysA)> **flush r.dbname1 to sysB@o.oraB** (PostgreSQL to Oracle implementation) In the above example, the data is flushed to Oracle instance oraB on sysB.
to*host@datadest***queue***queuename*	This option flushes data for a designated post queue and target instance on a designated target system. It does not affect: • Other post queues for that instance or any other target instance on that system. • Other post queues on any other target system. **In the syntax:** • *host* is the target system's name. • *datadest* is expressed as o.SID, where SID is the ORACLE_SID of the target instance. • *queuename* is the name of the post queue. **Examples**: sp_ctrl(sysA)> **flush r.dbname1 to sysB@r.dbname2 queue sales** (PostgreSQL to PostgreSQL implementation) In the above example, the data is flushed for post queue sales for target PostgreSQL database dbname2 on sysB. sp_ctrl(sysA)> **flush r.dbname1 to sysB@o.oraB queue sales** (PostgreSQL to Oracle implementation) In the above example, the data is flushed for post queue sales for target instance oraB on sysB.

Job status command for PostgreSQL

Use the **job status** command to view the status of the last **compare**, **compare using**, **repair** or **repair using** job, and for any other jobs that are still running. The **job status** command can be used to view detailed status on a job or a portion of a job, or to view status on all jobs for which SharePlex has history. SharePlex retains the history for all jobs as specified by the SP_SYS_JOB_HISTORY_RETENTION parameter. See the **clear history** and **remove log** commands for information on actively removing history and/or job process logs.

```
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      %Comp    Total
                                                  rows                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
```

To view a summary of all jobs for which SharePlex has history:

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


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

To view the status of a job with detail:

```
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
```

Usage

Supported sources:	PostgreSQL
Supported targets:	PostgreSQL
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**compare status, repair status**

Syntax

Basic command	Command options	Remote options
job status	[*job_id*]	[**on** *host* |
	[*job_id.table_id*]	**on** *host:portnumber* |
	[**all**]	**on** *login/password@host* |
	[**full**]	**on** *login/password@host:portnumber*]

Basic command	Command options	Remote options
	[**detail**]	
	[**status**]	

Syntax description

Component	Description
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**: `sp_ctrl(sysA)>`**job status 2828.2**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example: `sp_ctrl(sysA)>`job status 2828.HR.SRC_TEST3**
all	Displays a summary line for every job with history in the database. **Example**: `sp_ctrl(sysA)>`**job status all**
full	Displays the status of every object in the job. By default, the job status command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)>`**job status2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)>`**job status detail**
status	Displays status history for previous jobs with the specified status. `sp_ctrl(SysA)>`**job status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

List config for PostgreSQL

Use the **list config** command to view a list of all active and inactive configurations on a source system.

The command displays the following information:

- **File Name**: The name of the configuration, the time and date that it was modified, and the size of the file.
- **State**: Whether the configuration is active or inactive. *Active* means the configuration file is currently involved in replication. *Inactive* means the configuration is not currently involved in replication.
- **Datasource**: The PostgreSQL instance containing the objects being replicated by the configuration.
- **Internal Name**: The name of the internal copy of the active configuration. This is the file that SharePlex actually replicates from. Its name is displayed under the **State** heading wherever there is an active configuration.

Usage

Supported sources:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**view config, show config**

Syntax

Basic command
list config

Modify Partition for PostgreSQL

Use the **modify partition** command to modify a row partition of a partition scheme in a horizontally partitioned replication configuration.

Reactivate the configuration file if the command affects a table that is already being replicated. SharePlex will only lock tables for which there are configuration changes.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issues on:	source system
Related commands:	add partition, drop partition, drop partition scheme, view partitions

Syntax

Basic command
modify partition in *scheme_name* **set** *keyword*=*value* [**and** *keyword*=*value*] [...] **where** *keyword*=*value* [**and** *keyword*=*value*] [...]

Syntax description

NOTE: See add partition for additional descriptions of these options.

Syntax Component	Description
scheme_name	The name of the partition scheme. Do not modify this component, or the row partition will shift to a new partition scheme.
keyword	Any of the following syntax components except *scheme_name*.
condition	Column condition that defines a row partition.
route	The routing map for this partition.
tablename	Fully qualified target table name.
name	Short name of this partition.
description	Description of this partition.

Examples

`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"**

Purge config for PostgreSQL

Use the **purge config** command to remove the data from all queues associated with a configuration without removing the queues themselves or deactivating the configuration. Avoiding a deactivation avoids the need for SharePlex to recalculate the configuration data. This saves time when the tables are large and numerous, enabling replication can start sooner.

Issue the **purge config** command on the source system to affect the source system and all target systems in the configured routes. Should any SharePlex process stop prior to or during the **purge config** activity, the command also stops working. When the process starts again, the command resumes working. Thus, **purge config** works even when the network is temporarily unavailable — the command remains in the queues until the connection is restored.

Cautions for using the purge config command:

Do not activate a configuration and then follow the **activate config** command with a **purge config** command. You might be purging more than just queued data, including the configuration information that controls replication, thus rendering the activation invalid.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issued for:	source and target systems
Related commands:	**abort config, deactivate config**

Syntax

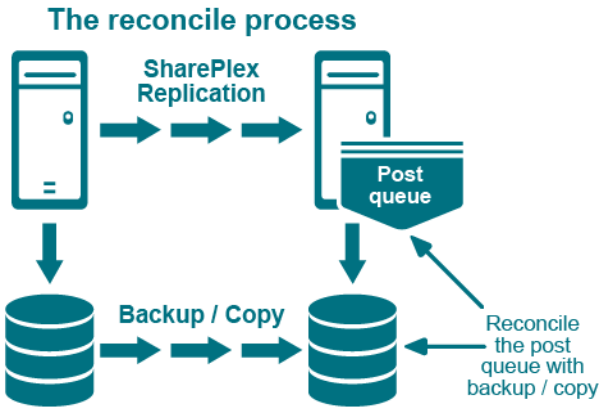
Basic command
purge config *filename*

Syntax description

Component	Description
filename	The name of the configuration that you want to purge. Configuration names are case-sensitive. **Example**: `sp_ctrl(sysA)>`**purge config sales**

Reconcile for PostgreSQL

Use the **reconcile** command as part of a procedure to synchronize (instantiate) source and target data with minimal interruption to the database users. The **reconcile** command coordinates the results of ongoing replication with a copy of the source data that is applied to the target system, such as that applied by a hot-backup or a native copy utility. The reconcile function compares the replicated changes in the post queue with the state of the target database after the recovery process. It differentiates between the transactions that were applied during recovery from those that have not yet been applied (still waiting in the post queue), and it only posts the non-duplicated changes so that both systems are synchronized.



Although the **reconcile** command is designed for use in high-volume environments, it can be used in low-volume environments with an understanding that the reconcile process can, in some circumstances, seem to stall. This happens because the **reconcile** command depends on data continuing to arrive from the source system. If there is no replication activity on the source system after the hot backup or copy, the reconcile process waits until source activity resumes.

Considerations when using the reconcile command

The **reconcile** command should be used when following specific procedures for the initial synchronization of source and target data. It is not meant to be a standalone command. For initial synchronization procedures, see the SharePlex Administration Guide.

NOTES:

- Ensure that the transactions committed before the LSN are replicated at the target
- Transactions that are not committed will not be removed and will be in the post queue
- If ROLLBACK / COMMIT is before the LSN, then transaction will be removed from post queue
- If ROLLBACK / COMMIT is after the LSN, then transaction will remain in the post queue

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issued for:	target system
Related commands:	**flush**

Syntax

Basic command	Command options
reconcile queue*queuename***for***datasource-datadest*	[**to flush**] [**pglsn** Log Sequence Number]

Syntax description

Component	Description*
queue	**queue** is a required part of the command.
queuename	The post queue on the target system that you want to reconcile. Valid values are: • The name of the source system if using default queues • The name of the queue if using named queues When using named post queues, issue the **reconcile** command for each one. To determine the queue name, issue the **qstatus** command in **sp_ctrl**. Queue names are case-sensitive on all platforms.
for*datasource-datadest*	• datadest is expressed as **o.**SID, where SID is the ORACLE_SID of the target instance (for Oracle target). • datasource is expressed as **r.**dbname, where dbname is the database name of the source instance (for PostgreSQL source). • datadest is expressed as **r.**dbname, where dbname is the database name of the target instance (for PostgreSQL target). **Examples**: sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-r.dbname2**

Component	Description*
	(PostgreSQL to PostgreSQL implementation) sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-o.oraB** (PostgreSQL to Oracle implementation)
pglsn LSN number	Use LSN number in case of PostgreSQL implementation. We can provide LSN number in decimal or hexadecimal format. **Query to find current LSN:** `select pg_current_wal_lsn();` **Examples**: sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-r.dbname2 pglsn 0/B817B360** (PostgreSQL to PostgreSQL - hexadecimal format LSN) sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-r.dbname2 pglsn 3088560992** (PostgreSQL to PostgreSQL - decimal format LSN) sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-o.oraB pglsn 3088560991** (PostgreSQL to Oracle decimal format LSN) sp_ctrl (sysB)> **reconcile queue SysA for r.dbname1-o.oraB pglsn 0/B817B361** (PostgreSQL to Oracle - hexadecimal format LSN)
to flush	Use this option to reconcile to a flush marker that is established with the **flush** command. Use it for synchronizing multiple Oracle databases in a peer-to-peer replication environment. The syntax must appear after the syntax for the basic command. **Example**: sp_ctrl (sysA)> **reconcile queue SysA for r.dbname1-r.dbname2 to flush** **NOTE:** Before executing the Reconcile command on target, issue the Flush command at source, which creates a flush marker. Even if we issue the Reconcile command first, it will wait for the flush marker to be created on the source machine.

Rename config for PostgreSQL

Use the **rename config** command to give a configuration file a different name. Use a name that is unique among the configuration files on the system.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issued for:	source system
Related commands:	**copy config, edit config, list config, view config**

Syntax

Basic command
rename config {*filename* **to** *newname*

Syntax description

Component	Description
filename **to** *newname*	• *filename* is the name of the configuration that you want to rename. Configuration names are case-sensitive. • **to** is a required part of the syntax. • *newname* is the new name you are giving the configuration. **Example**: `sp_ctrl(sysA)` **> rename config sales to sales2**

Repair status command for PostgreSQL

Use the **repair status** command to view the status of the last compare or repair job run. The **repair status** command can be used to view detailed status on a compare or repair job or a portion of a compare or repair job, or to view status on all compare and repair jobs for which SharePlex has history.

For details and examples about using the **repair status** command, see the **job status** command

Usage

Supported sources:	PostgreSQL
Supported targets:	PostgreSQL
Authorization level:	Operator (2)
Issued for:	source system
Related commands:	**compare status, job status**

Syntax

Basic command	Command options	Remote options
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*]	

Syntax description

Component	Description
job_id	Displays status history for the job with the specified SharePlex-assigned job ID. **Example**: `sp_ctrl(sysA)` > **repair status 2828.2**
job_id.table_id	Displays status history for the job with the specified SharePlex-assigned job ID and table. **Example**: `sp_ctrl(sysA)` > **repair status 2828.HR.SRC_TEST3**
all	Displays a summary line for every job with history in the database. **Example**: `sp_ctrl(sysA)` > **repair status all**
full	Displays the status of every object in the job. By default, the job status

Component	Description
	command displays the status of those objects not completed, or completed with an exceptional status. **Example**: `sp_ctrl(sysA)` **> repair status2828 full**
detail	Displays detail information for every object reported upon. By default, the job status command displays a summary line for every object reported upon. Note that the detail information is the same as is displayed for the job_id.table_id option. **Example**: `sp_ctrl(sysA)` **> repair status detail**
status	Displays status history for previous jobs with the specified status. **Example**: `sp_ctrl(SysA)` **> repair status "Error"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Verify config for PostgreSQL

The **verify config** command is intended for use as a preventive measure to avoid certain activation and replication problems. It is intended to be used to test activation to ensure that it will complete successfully.

The **verify config** command verifies tables only.

This command can be used to:

- Verify the syntax of the entries in the configuration file.
- Report an error if the source object is not supported for replication by SharePlex.
- Report if a host name specified in a route is unreachable.
- Report if there are duplicate specifications for a single object.
- Report if an object specification will be skipped and the reason why.

What the verify config command does not support

The **verify config** command does not:

- Verify activation time.
- Verify target objects or the target database name.

Verifying added or changed objects in an active configuration

To verify objects that you want to add to an active configuration or objects that you want to change (such as routing changes), it is suggested that you copy and modify the active configuration and then run the **verify command** against that copy.

Viewing the results of the verification

The **verify config** command retains control of the sp_ctrl interface until the verification is completed.

The command will read the entire config file and logging errors.

The results of the verify are displayed to the screen within **sp_ctrl**. If you would like to view detailed results you may:

- Issue the verify **config** command in **sp_ctrl** using the **detail** option
- Navigate to the results file directly using the path displayed to the screen after issuing the **verify config** command.

Usage

Supported sources:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for

	PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Authorization level:	Viewer (3)
Issued for:	source system
Related commands:	**activate config**

Syntax

Basic command	Command options
verify config *filename*	**detail**

Supported wildcard syntax

SharePlex supports the following SQL wildcards for tablename

- Percent (%) wildcard to specify a string.
- Underscore (_) wildcard to specify a single-character.

For more information on supported wildcard syntax, see Use Wildcards to Specify Multiple Tables.

Syntax description

Component	Description
filename	*filename* is the name of the configuration to be verified.
detail	This option will display a greater level of detail to the screen. **Example**: `sp_ctrl(sysA)` **> verify config myconfig detail** In this example, the **myconfig** file will be verified and the results will be displayed with a higher level of detail.

View Partitions for PostgreSQL

Use the **view partitions** command to view the row partitions in one partition scheme or all partition schemes in a horizontally partitioned replication configuration.

For more information about how to configure horizontally partitioned replication, see the SharePlex Administration Guide.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issues on:	source system
Related commands:	add partition, drop partition, drop partition scheme, modify partition

Syntax

Basic command
view partitions for {*scheme_name* | **all**}

Syntax description

Component	Description
scheme_name	Show the row partitions for the specified partition scheme.
all	Show all row partitions, grouped according to the names of their partition schemes.

Examples

```
sp_ctrl> view partitions for scheme1
sp_ctrl> view partitions all
```

Show Capture for PostgreSQL

Use the **show capture** command to view statistics for the Capture process.

Basic command

The basic **show capture** command shows an overview of the process, such as the datasource, whether the process is running or stopped, and other basic information.

Detailed statistics

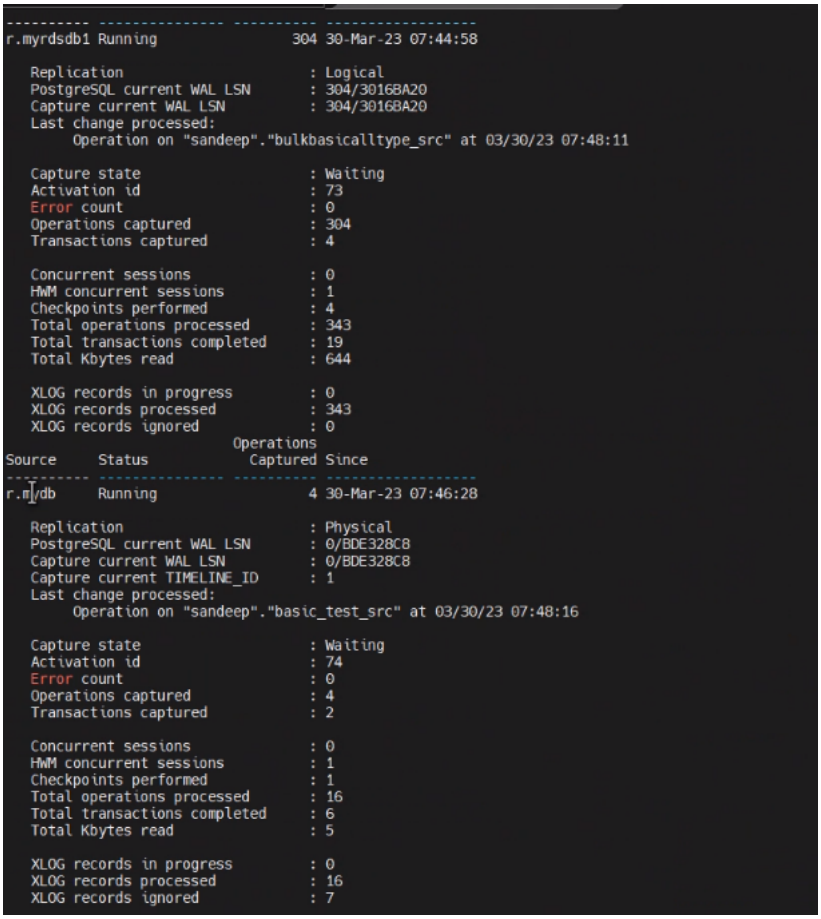
To view detailed statistics for the Capture process, use the **show capture** command with the **[detail]** option. That option shows detailed statistics that can help you assess the performance of the process, decide whether tuning parameters need to be adjusted, and detect problems or bottlenecks.

Detailed statistics for PostgreSQL Capture

Statistic	Description
Host	The name of the local machine (source system).
System time	The current time according to the system clock.
Source	The name of the source PostgreSQL database.
Status	The status of the Capture process (running or stopped).
Since	The time that Capture started.
PostgreSQL current WAL LSN	The LSN (Log Sequence Number) number of the WAL file log to which PostgreSQL is writing.
Capture current WAL LSN	The LSN (Log Sequence Number) number of the WAL file log that Capture is reading. This value needs to show the latest LSN value read by Capture, independent of whether data is coming from a replicated table or not. In idle condition, it should match PostgreSQL's current WAL LSN.
Last WAL file record processed	The record being processed by Capture or the last one processed if Capture is not currently replicating data. **NOTE:** The time displayed in this field is the time when the Capture processed the record, not the time when the record was logged in the WAL file. This is a limitation of the PostgreSQL WAL file.
Capture state	The state of the process, in relation to the replication work it performs: It can be one of the following: • WAITING: Capture is waiting for records from WAL sender

Statistic	Description
	• PROCESSING: Capture is processing a WAL file log record for replication. • STOPPED BY ERROR: Capture is stopped by error and error is mentioned in the EVENT logs
Activation ID	The internal identifying number of the configuration activation, which identifies the associated processes and queues. This value needs to be displayed immediately after activation even before DML replication starts.
Error count	The number of records that were skipped due to PostgreSQL errors since Capture started. Data from skipped records is not reflected in the target database.
Operations captured	The number of DML operations that Capture successfully processed for replication since it started.
Transactions captured	The number of committed PostgreSQL transactions whose operations Capture successfully replicated since it started.
Concurrent sessions	The number of PostgreSQL sessions being processed at the same time.
HWM concurrent sessions	The largest number of concurrent PostgreSQL sessions since Capture started.
Checkpoints performed	The number of checkpoints to save the state of Capture since Capture started. Frequent checkpointing generates additional overhead on the system, but infrequent checkpoints cause SharePlex to recover less quickly from a system or instance failure. By default, Capture checkpoints every 40,000 messages or 120 seconds, but it can be adjusted with the `SP_OCT_CHECKPOINT_FREQ` and `SP_OCT_CHECKPOINT_TIME` parameters.
Total operations processed	The number of all PostgreSQL operations and SharePlex internal operations processed by Capture since it started, including records captured for replication and records for objects not in the configuration (both replicated and not-to-be replicated records)
Total transactions completed	The number of committed PostgreSQL transactions processed by Capture since it started, including transactions captured for replication and transactions for objects not in the replication configuration (both replicated and not-to-be replicated transactions)
Total Kbytes read	The size in kilobytes of the data that was processed by Capture since it started.
XLOG records in progress	The number of records that Capture is processing.
XLOG records processed	The total number of XLOG records processed.
XLOG records ignored	The number of records that Capture ignored because they are not associated with objects in the configuration.
Replication	Type of replication (Physical or Logical)
Capture current TIMELINE_ID	Displays the current Timeline ID (applicable only for Physical replication)

For an example of the sample statistics for PostgreSQL Capture, see the example below:



Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issued for:	source and target systems
Related commands:	**show post**

Syntax

Basic command	Command options
show capture	[**detail**] [**for***datasource*]

Syntax description

Component	Description
show capture	Shows the state of the Capture process and a summary of the operations captured.
detail	Shows detailed statistics that can help you tune Capture's performance and diagnose problems. **Example**: `sp_ctrl(sysA)` > **show capture detail**
for *datasource*	This option shows Capture statistics only for a specific datasource. *datasource* is expressed as **r**.*database* where *database* is a dbname. **Example**: `sp_ctrl(sysA)` > **show capture for r.dbname**

Show Post for PostgreSQL

Use the **show post** command to view statistics for the Post process.

Basic Show Post command

The basic **show post** command shows global statistics for all sessions a Post process. It shows the status of the Post process and the number of messages posted since it started. To filter the output for a specific post queue or datasource (useful when you have multiple replicating data streams), use the **queue** *queuename* or **for** *datasource-datadest* option.

Detailed Show Post command

To view detailed statistics for the Post process, use the **show post** command with the **detail** option. That option shows the most recent SQL statement processed, as well as other statistics that can help you assess Post's performance, decide whether tuning parameters need to be adjusted, and detect problems or bottlenecks.

The following explains the detailed statistics shown with **show post**. These statistics vary slightly depending on the type of source and target.

Statistic	Description
Host	The name of the local machine (target system).
Source	The source of the data being processed by Post.
Queue	The Post queue for this Post process. For a default Post queue, it is the name of the source system. For a named queue, it is the user-defined name.
Target	The name of the target of this Post process, for example the name of an PostgreSQL instance or Open Target database.
Status	The status of the Post process (running or stopped). Possible statuses are: • Running • Stopping • Stopped by user • Stopped due to error
Operations posted **Operations processed**	The number of transactional operations and SharePlex internal operations that this Post process processed since it was started.
Since	The time that Post started.
Total	The number of messages in the queue that have yet to be read-released. This number corresponds to the 'Number of messages' returned from running **qstatus**. (The TOTAL value goes down with time and shows same value as Number of messages in QSTATUS)
Backlog	The number of messages that are waiting in the queue to be

Statistic	Description
	processed by Post.
Last operation posted	Identifying information for the most current operation that is being posted to the target if Post is active, or the last operation posted if it is inactive. This information is specific to the type of datastore that originated the data. An operation can be: • INSERT • UPDATE • DELETE • COMMIT • INSERT_MULTIPLE or DELETE_MULTIPLE (array/bulk operations). • SharePlex internal operation.
Last transaction posted	Identifying information for the last transaction that was posted. This information is specific to the type of datastore that originated the data.
Post state	The state of the Post process, in relation to the replication work it performs. It can be one of the following: • **Waiting**: Post is waiting for messages to process. • **Active**: Post is posting changes to the database. • **Committed**: Post is committing the transaction. • **Idle**: Post has no open transactions to process. • **Rollback**: Post is processing a rollback. • **Recovery**: Post is in a crash-recovery mode.
Activation ID	The activation ID of the current configuration. This value needs to be displayed immediately after activation even before DML replication starts.
Operations processed	The number of SQL operations that Post applied to the target, whether or not the COMMIT was received.
Transactions processed	The number of committed transactions that Post applied to the target since it was started.
Insert operations	The number of INSERT operations processed by Post since it was started.
Update operations	The number of UPDATE operations processed by Post since it was started.
Delete operations	The number of DELETE operations processed by Post since it was started.
Latency	The time taken to process the replication (excluding the time taken by database on source)

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Issued for:	target system
Related commands:	**show capture**

Syntax

Basic command	Command options
show post	[**detail**] [**queue** *queuename*] [**for***datasource-datadest*] [**sessions**]

Syntax description

Component	Description
show post	Shows the state of the Process process and a summary of the operations processed.
detail	This option displays detailed statistics for the Post process. **Example**: `sp_ctrl(sysB)` **> show post detail**

queue*queuename*	This option filters the **show post** display for a specific post queue.
	• **queue** is a required part of the syntax.
	• *queuename* is the post queue for which you want to see Post statistics. Valid values are:
	  • the name of the source system if using default queues.
	  • the user-defined queue name, if using named queues.
	If you are unsure what the queue name is, issue the **qstatus** command. Queue names are case-sensitive on all platforms.

Component	Description
	This option can appear in any order with other options. **Example**: `sp_ctrl(sysB)` **> show post queue sysA**
for*datasource-datadest*	This option filters the show post display for a specific data stream. • **for** is a required part of the syntax. • *datasource* is expressed as **r**.*database* where *database* is a dbname. • *datadest* is expressed as one of the following, depending on the target: **r.***database_name* **x.kafka** **x.jms** **x.file** This option can appear in any order with other options. **Example**: `sp_ctrl(sysB)` **> show post for r.dbnameA-r.ssB**
sessions	For PostgreSQL targets, this option displays statistics for all the threads spawned by the Post process. For Open Target, which is single-threaded, this option can be used to view details for that thread. This option can appear in any order with other options. **Example**: `sp_ctrl(sysB)` **> show post sessions queue *queuename***

show_last_posted

Use the **show_last_posted** command to view the PostgreSQL LSN of the last transaction that a Post process applied to the target. The command shows all of the last LSNs when using multiple post queues.

```
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>
```

NOTE: This command stops all Post processes in order to obtain the LSN information.

Usage

Supported source:	PostgreSQL (on-prem), Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Supported targets:	PostgreSQL, Oracle, SQL Server, Kafka, Amazon RDS for PostgreSQL, Amazon Aurora for PostgreSQL, Azure Database for PostgreSQL Flexible Server, and Google Cloud SQL for PostgreSQL
Authorization level:	Viewer (3)
Issued for:	target system
Related commands:	**activate config**

Syntax

Basic command
show last_posted

4

SharePlex Parameters for PostgreSQL

PostgreSQL parameters control and tune various aspects of replication for PostgreSQL databases.

Contents

- Descriptions of User-Configurable Parameters
  - Capture Parameters
  - Open Target Poster parameters
  - Oracle Poster parameters
  - Read parameters
  - Compare/Repair Parameters
  - Export parameters
  - Import parameters
  - Queue parameters
  - sp_cop parameters
  - System parameters

Descriptions of User-Configurable Parameters

This section describes the user-configurable SharePlex parameters for the PostgreSQL database. Parameters not documented in this chapter are internal parameters that should only be modified under guidance of a Quest developer or Technical Support representative.

Parameters are grouped as follows:

Capture Parameters

These parameters are used by the SharePlex Capture process.

SP_CAP_INCLUDE_UNCHANGED_COL

This parameter controls whether or not unchanged columns are included in the after image of an UPDATE operation. By default, SharePlex only includes the changed values in the after image.

Default: 0 (off)

Range of valid values: 0 or 1

Takes effect: immediately

SP_CAP_MIN_SESSIONS

This parameter controls the minimum number of subqueues that Capture creates and maintains to contain data from concurrent transactions. This parameter supports the Post Enhanced Performance (PEP) feature by allowing Post to increase its own concurrency. The PEP feature is controlled with the SP_OPO_DEPENDENCY_CHECK (Oracle Poster) or SP_OPX_THREADS (Open Poster) parameter.

For best results, set SP_CAP_MIN_SESSIONS to at least the number of CPU cores that you have on the target system.

Default: 8

Range of valid values: any positive integer

Takes effect: when Capture is restarted

SP_CAP_REDUCED_KEY

This parameter controls which parts of an operation in the WAL file are sent by Capture to the target for use by Post. Different SharePlex features may require more or less data to be made available to the Poster process.

- If set to 0, this parameter directs Capture to send all of the data that PostgreSQL writes to the WAL files other than TEXT. This setting sends the most data across the network, but is required to support certain SharePlex features.

- If set to 1 (the default), this parameter directs Capture to send the key values and, for UPDATEs, the before and after values of columns that were changed in the operation. This setting is the default. This setting provides a good balance between replication performance and target data integrity, because it enables Posterto perform a before-and-after comparison of the changed columns when constructing the WHERE clause. For more information about how SharePlex uses before and after values, see the SharePlex Administration Guide.
- If set to 2, this parameter directs Capture to send the key values and, for UPDATEs, the after values of the columns that changed to the target. This setting sends the least amount of data across the network and also enables Posterto only use the key values when posting data, which improves performance.

Default: 1

Range of Valid Values: 0, 1, 2

Takes effect: Immediately

SP_CAP_ONELINE_DEBUG

This parameter controls the oneline debug feature for the Capture process.

The oneline debug feature enables you to perform debugging for out-of-sync errors without consuming a large amount of the system resources. The oneline debug logs just enough information in one line to detect where in the data stream an out-of-sync condition occurred. Once oneline debug identifies the process that is causing the problem, you can then enable regular debugging for that process.

To enable this parameter, specify the objects that you want to debug by their object ID, and separate each one by a comma. An example is:

```
sp_ctrl>set sp_cap_oneline_debug 230230, 351626
```

The following items are logged:

- The transaction identifier used by SharePlex
- The type of DML or DDL operation that was affected
- The object ID of the affected source table
- The time when the operation occurred (PostgreSQL)
- The LSN value for the transaction
- An internal SharePlex code that prevents redundant operations

Default: Disabled

Range of valid values: a string that forms a list of objects listed by object ID, separated by commas.

Takes effect: Immediately

SP_CAP_CHECKPOINT_FREQ

This parameter is one of the parameters that can be used to define the frequency of the checkpoints that are issued by the show capture command. This parameter works in conjunction with the `SP_CAP_CHECKPOINT_TIME` parameter.

Default: 40000 messages

Range of valid values: any positive integer

Takes effect: immediately

SP_CAP_CHECKPOINT_TIME

This parameter works in conjunction with the `SP_CAP_CHECKPOINT_FREQ` parameter. It defines the time delay, in seconds, before the Capture process checkpoints. If the value set for this parameter is reached before the value set for `SP_CAP_CHECKPOINT_FREQ`, it triggers the checkpoint. (Check pointing saves the state of the process in case it is needed for failure recovery.)

Default: 120 seconds

Range of valid values: any positive integer

Takes effect: immediately

SP_CAP_DENIED_SESSION_PG

This parameter is applicable for SharePlex PostgreSQL Capture only. It controls whether all transactions for a user session should be ignored or filtered by the Capture process.

If set to 0, Capture will not ignore any transactions, whether or not a user session is bound to the origin name.

If set to 1, Capture will ignore all transactions for a user session which are bound to the origin name in the format `'sp_deny_anyString'`. The `sp_deny_` prefix must be used, and you can insert a string in place of 'anyString'. **For example:** `sp_deny_1`.

The user needs to create (if it does not exist) and bind the PostgreSQL user session to the origin name by calling the PostgreSQL functions `'pg_replication_origin_create ()'` and `'pg_replication_origin_session_setup ()'`.

Examples:

- select `pg_replication_origin_create('sp_deny_1');`
- select `pg_replication_origin_session_setup('sp_deny_1');`

IMPORTANT: Ignoring transactions on the source machine may lead to an out-of-sync condition.

Default: 0

Range of valid values: 0 or 1

Takes effect: immediately

How to use this parameter:

Before the designated user performs the transaction in the source database, issue the following command on source sp ctrl: (For PostgreSQL as a source and target).

```
sp_ctrl>set param SP_CAP_DENIED_SESSION_PG 1
```

NOTE: The parameter is live. After setting this parameter, users need to bind the PostgreSQL session to the origin by calling the `pg_replication_origin_session_setup()` function. All operations executed on source database after this will be ignored until the parameter is unset. The `SP_CAP_DENIED_SESSION_PG` is a live parameter and can be changed any time.

To unset the parameter, run the below command:

```
sp_ctrl>reset param SP_CAP_DENIED_SESSION_PG
```

Perform the following steps to bind more than 10 PostgreSQL sessions simultaneously to the different origins:

1. Navigate to PostgreSQL data directory (`cd /var/lib/pgsql/13/data/`).
2. Edit the `postgresql.conf` file by uncommenting the `max_replication_slots` Parameter (It is by default commented, user need to uncomment it & set it to more than maximum number of simultaneous sessions user want to bind to different origins).
3. Restart the database.

NOTES:

- The syntax for adding a valid origin is `sp_deny_name`, the `sp_deny_` prefix is required before the user-defined name of origin.
- The `max_replication_slots` value should be more than simultaneous user sessions who are going to do session setup. [For eg, if user is going to do origin setup for 100 sessions, set the `max_replication_slots` value greater than 100]

SP_CAP_DENIED_PORT

This parameter is applicable only for PostgreSQL Capture. It can be used to ignore or filter operations performed on specific SharePlex ports by the Capture process when multiple SharePlex instances are running on the same machine. It is useful in a cascading replication setup of SharePlex running on different ports.

The SP_OPX_CREATE_ORIGIN_PG parameter must be enabled on the specified ports for this parameter to work.

IMPORTANT: Ignoring transactions on the source machine may lead to an out-of-sync condition.

Default: (empty)

Range of valid values: list of valid SharePlex ports separated by commas

Takes effect: when Capture is restarted

Open Target Poster parameters

These parameters are used by the SharePlex Poster process when applying data to an Open Target (non-Oracle) target.

SP_OPX_REDUCED_KEY

This parameter controls the content of the Poster WHERE clause. Posteruses a WHERE clause to find the row in the target that needs to be changed by a replicated UPDATE from the source. Different SharePlex features may require more or less data to be used in the Poster WHERE clause.

- If set to 0, this parameter directs Poster to construct a WHERE clause of all of the data that is sent by Capture. The data that Capture sends depends on the setting of the SP_OCT_REDUCED_KEY parameter. If you want the WHERE clause to include the values of the keys and all of the columns other than TEXT, set both parameters to 0.
- If set to 1, this parameter directs Poster to build a WHERE clause with the key values and the before values of the columns that changed. This setting provides a good balance between replication performance and target data integrity, because it enables Poster to perform a before-and-after comparison of the changed columns when constructing the WHERE clause. To be able to use this setting, the SP_OCT_REDUCED_KEY parameter must be set to 0 or 1. For more information about how SharePlex uses before and after values, see the SharePlex Administration Guide.
- If set to 2, this parameter directs Poster to build a WHERE clause of only the key columns. This setting can be used to maximize posting performance. Because this setting omits the before-and-after comparison of the changed columns, you should ensure that no process or user can make changes to the target data except SharePlex.

Default: 0

Range of Valid Values: 0, 1, 2

Takes effect: When Poster is restarted

SP_OPX_XML_MARK_NULL

This parameter controls how Null values are displayed when the user selects XML as the target output format for Kafka. JSON format should always be NULL. Set the SP_OPX_XML_MARK_NULL parameter to 1 on the Kafka target to display the null value as `xsi:nil="true"`.

Default: 1

Range of Valid Values: 0 or 1

Takes effect: Process restart

SP_OPX_CREATE_ORIGIN_PG

This parameter is applicable for Open Poster only when the target is PostgreSQL. It is used to create a replication origin in the PostgreSQL database and bind the Poster user session with the created origin. It is useful in **bi-directional replication** (peer to peer replication) to prevent operations posted by the Poster process from looping back. It is also useful in a cascading replication setup of SharePlex running on different ports to ignore operations performed on ports set in the SP_CAP_DENIED_PORT parameter. If enabled, the Capture process of the same port ignores such operations posted by the Poster process. By default, it is disabled.

NOTE: If this parameter is enabled, each Poster process creates an origin, and the number of origins created depends upon the **max_replication_slots** parameter of the PostgreSQL database. As a result, if the number of post-processes exceeds this parameter value, increase it and restart the database.

Default: 0 (Disabled)

Range of valid values: 0 or 1

Takes effect: When Poster is restarted

SP_OPX_LOG_CONFLICT

This parameter enables the logging of information about successful conflict resolution procedures to the **shareplex_conf_log** table. This applies only to the prepared routines that are provided by SharePlex.

- A setting of 1 enables the logging of conflict resolution to the **shareplex_conf_log** table.

  NOTE: A setting of 1 will not update the column `existing_timestamp` (when existing data is not replaced) in the `shareplex_conf_log` table.

- A setting of 2 enables the logging of conflict resolution to the **shareplex_conf_log** table with Poster query for additional meta data.

  Using **LeastRecentRecord** or **MostRecentRecord** prepared routines Poster will query the target database for the timestamp column of the existing record. The query result is logged into the `existing_timestamp` column of the `shareplex_conf_log` table.

  NOTE: A setting of 2 may affect the performance of Poster as a result of making the query.

Default: 0 (Disabled)

Range of valid values: 0, 1, or 2

Takes effect: when Poster is restarted

List of additional supported parameters for PostgreSQL Open Target Poster

- 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 parameters

These parameters are used by the SharePlex Poster process when applying data to an Oracle target.

SP_OPO_HINTS_LIMIT

This parameter controls the maximum number of hints (table/index combinations) that can be listed in the hints file. Use hints only if you see that Poster is doing full table scans on tables where there are defined indexes. Using a large number of hints can reduce the performance of the Poster process.

Default: 100 hints

Range of valid values: any positive integer

Takes effect: when Poster is restarted

List of additional supported parameters for Oracle Poster process

- SP_OPO_DISABLE_OBJECT_NUM
- SP_OPO_COMMIT_REDUCE_MSGS
- SP_OPO_DEPENDENCY_CHECK
- SP_OPO_SUPPRESSED_OOS
- SP_OPO_READRELEASE_INTERVAL

Read parameters

These parameters are used by the SharePlex Reader process.

SP_ORD_FIRST_FIND

This parameter controls how the Read process checks column conditions to determine whether or not a replicated row change satisfies them.

- At the default of 1, when a row change satisfies a column condition, SharePlex does not check any other column conditions to see if that row change also satisfies any of them.
- At a value of 0, SharePlex sends the data to all target systems where the column conditions are satisfied.

For more information about horizontal partitioning, see the SharePlex Administration Guide.

Default: 1 (on)

Range of valid values: 0 or 1 (flag)

Takes effect: when Read is restarted

SP_ORD_HP_IN_SYNC

This parameter is used for horizontally partitioned replication to ensure that data is replicated properly when a value for a column in a column condition changes so that the row no longer satisfies the condition. It enables SharePlex to automatically correct the following:

- UPDATEs that cause a row to meet a different column condition than the one created for that row, sending the changes to a different location. An example would be an UPDATE to a row for which the column condition is **region=East** that changes the value of the **region** column to WEST. Such operations will fail because the original INSERT statement for that row was replicated to the original location (the Eastern region), so the row does not exist in the new location (the Western region) when Poster attempts the update there.
- UPDATEs that cause a row to meet a column condition (and be replicated) when the row was not supposed to be replicated. An example would be when the region column is updated from the value of HEADQUARTERS (for which a row is not replicated) to the value of WEST. Such operations will fail because the original INSERT statement for that row (into the headquarters system) was not replicated to the Western region's system, so Poster cannot perform the update there.
- UPDATEs that cause a row to no longer meet any column condition. An example would be when the **region** column is updated from the value of WEST to the value of HEADQUARTERS. The original INSERT statement was replicated to the Western region's system, but the update to the new value is not replicated, because the new value does not meet a column condition (headquarters data is not shared). The rows are now out of synchronization, but there are no errors.

When this parameter is enabled, SharePlex automatically corrects rows for which UPDATEs cause the preceding conditions. SharePlex converts the UPDATE to a DELETE and, if needed, an INSERT.

To convert an UPDATE statement (which normally only uses the changed columns and the key) to an INSERT statement, SharePlex needs values for all of the columns. Enabling `SP_ORD_HP_IN_SYNC` directs SharePlex to

send all of the columns in a row to the Poster process when there is an UPDATE to a table using horizontally partitioned replication, so that an INSERT can be constructed.

Set this parameter on the **source** system before you activate the configuration. If replication is active, set the parameter and then reactivate the configuration so that SharePlex can rebuild its object cache.

If you know that the columns in column conditions for tables using horizontally partitioned replication will never change, leave this parameter set to 0, because using it incurs processing overhead.

This parameter is not compatible with `SP_OCT_REDUCED_KEY` and `SP_OPO_REDUCED_KEY` [any value: 1 or 2] as it overrides the behavior of both the parameters.

Default: 0 (off)

Range of valid values: 0 or 1 (flag)

Takes effect: when Read is restarted

List of additional supported parameters for PostgreSQL Reader process

- 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 Parameters

These parameters control properties of the SharePlex Compare and Repair commands.

SP_DEQ_PG_FORCE_PARALLEL

This parameter enables the use of parallel queries in a running session for the PostgreSQL database.

Based on this parameter value, the following value is assigned to PostgreSQL database parameter `force_parallel_mode (DB version < 16.0)/ debug_parallel_query (DB version >= 16.x)`

0 - off

1 - on

2 - regress

Default value: 0

Range of valid values: 0 to 2

Takes effect: Immediately available for the next comparison.

NOTES:

- SharePlex cannot control the number of worker threads assigned for parallelism in a SELECT query by the PostgreSQL database optimizer.
- This parameter is not supported with the `pg_hint_plan` extension.

SP_DEQ_PG_PARALLEL_MAX

This parameter manages the maximum number of parallel worker threads allowed for a query in a running session for the PostgreSQL database.

Value of this parameter is assigned to PostgreSQL database parameter - `max_parallel_workers_per_gather`

Default value: 6

Range of valid values: 0 to 1024

Takes effect: Immediately available for the next comparison.

NOTES:

- SharePlex cannot control the number of worker threads assigned for parallelism in a SELECT query by the PostgreSQL database optimizer.
- This parameter is not supported with the `pg_hint_plan` extension.

SP_DEQ_PG_PARALLEL_SETUP_COST

This parameter is utilized to configure the `parallel_setup_cost` parameter for the PostgreSQL database in a running session.

Default value: 10

Range of valid values: 0 to 2147483647

Takes effect: Immediately available for the next comparison.

NOTES:

- SharePlex cannot control the number of worker threads assigned for parallelism in a SELECT query by the PostgreSQL database optimizer.
- This parameter is not supported with the `pg_hint_plan` extension.

SP_DEQ_PG_PARALLEL_TUPLE_COST

This parameter is used to configure the `parallel_tuple_cost` parameter for the PostgreSQL database in a running session.

Default value: 0.001

Range of valid values: 0.0001 to 3.4E+38

Takes effect: Immediately available for the next comparison.

NOTES:

- SharePlex cannot control the number of worker threads assigned for parallelism in a SELECT query by the PostgreSQL database optimizer.
- This parameter is not supported with the `pg_hint_plan` extension.

SP_DEQ_BATCH_ENABLED

When set to 1, this parameter enables batch DML operations during the repair process. If set to 0, out-of-sync (OOS) records will be corrected by firing one transaction at a time during the repair.

Default value: 1

Range of valid values: 0 or 1

Takes effect: Immediately available for the next comparison.

SP_DEQ_MAX_BATCH_MBYTES

This parameter controls the maximum size of a batched DML operation to be executed at once during the repair process. The repair process continues to add operations (I/U/D) until the batch size specified by this parameter value is reached. This parameter is applicable on the target side where the client process runs.

Default: 1 megabyte

Range of valid values: 1 to any positive integer

Takes effect: Immediately available for the next comparison.

SP_DEQ_COMMIT_FREQ

This parameter is applicable for repair when batch mode is off. It determines the number of single transactions that should be committed.

Default value: 1000

Range of valid values: 0 to 10000

Takes effect: Immediately available for the next comparison.

SP_DEQ_USE_SP_CKSUM

This parameter determines whether SHA256 hash is used in the comparison. When set to the default value of 0, SHA256 hash is utilized. When set to 1, SharePlex selects the exact table data as it is, using an internal checksum method.

NOTES:

- Setting this parameter to 1 for larger tables may affect performance, as `hash_values` take significantly less time than processing exact table data.
- If the table has no key columns and has NULL values in any of its columns, the Compare and Repair operation should be run by setting the parameter `SP_DEQ_USE_SP_CKSUM` to 1.

Default value: 0

Range of valid values: 0 or 1

Takes effect: Immediately available for the next comparison.

SP_DEQ_PG_DECLARE_FETCH_SIZE

This parameter determines whether the driver attempts to return a result set in a single fetch or across multiple fetches.

If this parameter is set to 0, the driver will perform a single fetch that can improve overall response times but at the cost of additional memory.

If this parameter is set to a non-zero number, the driver may perform either a single fetch or multiple fetches, depending on the size set for this parameter. The driver will perform a single fetch when the size set by the user for this parameter is greater than the actual size of the result set returned by a select query. Otherwise, it will perform multiple fetches. If multiple fetches are used, the size of the fetch is determined by the value of this parameter. For large result sets, retrieving rows in multiple fetches can improve response time and thereby prevent possible timeouts.

Additionally, this parameter can be adjusted to trade-off between throughput and response time.

Default value: 100

Range of valid values: Between 0 to 2147483647, applicable to both Server and Client processes.

Takes effect: Immediately after running the Compare or Repair command.

SP_DEQ_READ_BUFFER_SIZE for PostgreSQL

This parameter applies only to advanced data type columns during a compare/repair operation. It adjusts the size of the buffer that holds the fetched data related to advanced data types when those columns are being compared and repaired. The value of the parameter should be adjusted based on the available system memory and data size.

Default: 1 MB

Range of valid values: any integer between 1 and 100 (values in MB)

Takes effect: Immediately available for the next comparison

Export parameters

These parameters are used by the SharePlex Export process.

- 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 parameters

These parameters are used by the SharePlex Import process.

- SP_IMP_ENABLE_AES
- SP_IMP_QUEUE_PAUSE
- SP_IMP_QUEUE_RESUME
- SP_IMP_WCMT_MSGCNT
- SP_IMP_WCMT_TIMEOUT

Queue parameters

These parameters control properties of the SharePlex queues.

- 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 parameters

These parameters are used by the SharePlex sp_cop program.

- SP_COP_IDLETIME
- SP_COP_SCAN_TIMEOUT
- SP_COP_SO_RCVBUF
- SP_COP_TPORT
- SP_COP_UPORT

System parameters

These parameters control system-related SharePlex properties.

- 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

Heterogenous compare-repair commands

The SharePlex commands configure, start, stop, control, and monitor the replication process. SharePlex commands are issued through the **sp_ctrl** interface.

Compare / compare using

Use the **compare** and **compare using** commands (collectively known as the *comparison commands*) to verify that a target table is in synchronization with its source table.

- The **compare** command compares individual source and target tables or any number of tables in a given schema through the use of wildcards. When comparing individual tables, you can make use of column-based filtering to control the rows that are selected for comparison.
- The **compare using** command compares all of the tables listed in the active configuration or in another file that contains a subset of the tables in the active configuration.

After running a comparison, you can run the **repair** or **repair using** command to repair any out-of-sync rows.

NOTE: A running comparison or repair does not affect the source tables in any way. SharePlex logs into the database only to query for read consistency, and the locks on the source tables are brief. SharePlex briefly locks the target tables during the processing, but users can continue accessing them with little or no awareness of the lock.

Supported operations

- SharePlex can detect and repair out-of-sync rows in a target table that are caused by **DML** operations: INSERT, UPDATE, DELETE.
- Compare/Repair can be executed on single or multiple instances that have the same Oracle client version, i.e., the same ORACLE_HOME.

Non-supported operations for Oracle as a source

SharePlex does not support (and will skip) the comparison and repair of the following:

- If a table has a hash partition, comparing or repairing using the partition name is not supported. The user can use the table name instead. However, if the sub-partition is hash-based and the root partition is not, the user can use either the table name or the root partition name. **(This limitation is applicable only for Oracle to PostgreSQL replication)**
- UDT inside VARRARY. Other UDTs are supported
- Sequences
- Tables for which transformation was used

- Network configurations in which a pass-through server is used to pass data between the source and target servers.
- XML type with different storage on the source and target (CLOB on source, BINARY on target, or the other way around). Like to like Compare/Repair is supported.
- VARRAY types not inside SDO_GEOMETRY or ST_GEOMETRY.
- Comparisons and repairs are not supported in a cascading replication environment.
- Do not perform DDL on a table that is being compared or repaired. A comparison does not detect out-of-sync conditions caused by DDL operations, including those that SharePlex supports. If the DDL changes the table definition, it invalidates the SELECT statement that is built by the comparison process to get the rows that need to be compared.

  Once you correct an out-of-sync condition caused by DDL, you can use the **repair** command to resynchronize the data in the rows.
- Comparison and repair command strings longer than 255 characters are not supported. This is an operating system limitation. To work around this limitation, use the **edit** command on the source system. You can type the command string within a text file, and then the command automatically executes the file.
- If multiple `ORACLE_HOME` directories with different Oracle versions are installed on the same machine, users can perform compare/repair operations only for the Oracle version that was used during the SharePlex configuration.

See the SharePlex Release Notes for additional information about data types that are supported by **compare** and **compare using**.

Non-supported operations for PostgreSQL as a source

- Network configurations in which a pass-through server is used to pass data between the source and target servers.
- Comparisons and repairs are not supported in a cascading replication environment.
- Comparison and repair command strings longer than 255 characters are not supported. This is an operating system limitation. To work around this limitation, use the edit command on the source system. You can type the command string within a text file, and then the command automatically executes the file.

Other conditions

- The tables that you want to compare or repair must be part of an active configuration file.
- Uncommitted transactions on a source table prevent the comparison and repair processes from obtaining the brief locks they need to obtain read consistency. Make certain that all transactions are committed before you run a comparison or repair.
- Replication latency reduces the performance of compare and repair processing. The message from the source that spawns the comparison and repair processes on the target is sent through the queues with the replicated data. Delays caused by a data backlog also delay the spawn message and can cause the source process to lose its read consistency. If possible, perform comparisons and repairs during off-peak hours.

- Do not truncate a table that is being compared or repaired. The comparison commands take a snapshot of each source table when they start. If a table is truncated, the table view in the snapshot is truncated, and this can cause the command to return invalid out-of-sync conditions.
- To repair a view, the following must be true:
  - The view table must contain all of the constraint columns of the underlying table.
  - The names of the constraint columns in the view must be identical to the names of the constraint columns in the underlying table.
- **(Applicable only for PostgreSQL source**) To enhance the performance of the Compare process, SharePlex supports parallel hints via the SP_DEQ_PARALLELISM parameter, allowing you to specify the degree of parallelism, i.e., the number of worker processes to be launched.

  By default, the PostgreSQL database optimizer selects the best-suited query execution plan for SQL statements, which may not be directly related to the degree of parallelism specified in the parallel hints. pg_hint_plan makes it possible to adjust execution plans using parallel hints each time a query is executed. SharePlex internally utilizes the pg_hint_plan extension if it is installed or configured in the database.

  The example syntax to use parallel queries is as follows:

  The pg_hint_plan extension supports the Parallel hint: `Parallel(table <# of workers> [soft|hard])`

  SharePlex uses Parallel hints with hard parsing support: `Parallel(table <# of workers> [hard])`

  When performing table-level partitioning on DATE or TIMESTAMP data types in Oracle to PostgreSQL comparisons, the NLS_CALENDAR parameter must be set to GREGORIAN on the Oracle database.

  NOTE: The number of workers should be less than the CPU VCore of the database system.
- If the Oracle table is set for replication to multiple targets that include Oracle and PostgreSQL (or other non-Oracle) targets, the compare command will stop with the message: "Oracle to multiple targets detected! Use 'at target' or 'to table' Compare/Repair options to select only one of its targets." (Applicable only for Oracle as a source)
- It is expected that characters in the Oracle database are stored with the correct byte encoding, as per the character set specified in the `NLS_CHARACTERSET` (or `NLS_NCHAR_CHARACTERSET` for `nchar`, etc.). If the byte values are corrupted, Shareplex Compare/Repair will not be supported. For more information on setting `NLS_LANG` at the client or session level while inserting data into the DB, refer to https://docs.oracle.com/en/database/oracle/oracle-database/21/nlspg/setting-up-globalization-support-environment.html#GUID-372A5104-DC0A-41CA-B449-6AEC93F09C2E (Applicable only for Oracle as a source)

Supported Charactersets

The following character sets are tested and supported for Oracle database as a source:

- US7ASCII
- UTF8
- WE8ISO8859P1

- AL16UTF16
- AL32UTF8

Special use cases

The following scenarios require special handling when running a comparison.

Use case	Compare support
Consolidated replication	Consolidated replication is supported if the target database and Post processes are configured to add the ID of the source host to each row. To compare or repair the correct rows in the central target table, use the **targetwhere** option and base the **where** clause on the source ID value. For example, to compare a table in the database at the Eastern headquarters of a company to the correct rows in the central corporate database, you could use a source ID of "East" for the Eastern database and then base the **targetwhere** clause on that value. Use the same **targetwhere** clause in the **repair** command. The comparison and repair processes can use the source ID value to select only the rows that are valid for the Eastern database. To use the comparison or repair commands for any implementation of consolidated replication, other than one that identifies a source ID, may result in the unwanted deletion of target rows. For more information about this configuration, see the SharePlex Administration Guide. You may need to combine the **targetwhere** option with the standard **where** option to ensure that the target rows are selected accurately.
Peer-to-peer replication	In a peer-to-peer configuration, you must decide which system is the *trusted source system* and which is the *secondary*, or target, system. The secondary system is the one where any repairs will be performed. Before you run a comparison or repair in a peer-to-peer environment, follow these steps: 1. Stop user access on the secondary system and wait for replicated operations from that system to post to the trusted source database. Users can continue to access the source database. 2. Issue the **qstatus** command on the source and secondary systems. 3. When there are 10 or fewer messages in the queues, run the comparison from the source system. 4. During a comparison, you can permit user access to the source and secondary databases after the **sp_xdesvr** and **sp_xdeclt** start. 5. Use the **repair** command with the **where** option to repair selected rows in a target table without locking users out of the table. For more information about this configuration, see the SharePlex Administration Guide.
Tables without keys	The comparison and repair commands issue a SELECT statement with an ORDER BY clause on the source and target systems. The ordering is faster if large tables have a primary key or a unique, non-null key and an index (preferably a unique index). Otherwise, all of the columns are used as a key.

Use case	Compare support
	If a table has no unique row identifier, but does have one or more columns that can identify a row as unique, you can use the **compare** command with the **orderby** option. When this option is used, SharePlex prints a notice to the **sp_xdesvr** log on the source system that the command used those columns as a key. **LIMITATION**: Data inconsistency is likely to occur in replication or compare-repair processes when SharePlex replication involves tables containing non-key columns with duplicate data.
Tables with extra source or target columns	Use the **compare** command with the **sourcewhere** or **targetwhere** clause if the source or target table contains extra columns and those columns contain unique values on which to base the sorting. See Control which rows are compared.
Compare operation on the XML data (applicable only for Oracle to Oracle)	SharePlex displays the "`ORA-04036: PGA memory used by the instance exceeds PGA_AGGREGATE_LIMIT`" error while performing the Compare operation on the XML data. **WORKAROUND:** 1. Upgrade Oracle to version 19.19 DBRU

Use case	Compare support

	2. Set an appropriate PGA_AGGREGATE_LIMITusing the Oracle suggested script provided below :
	`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. Upgrade to or install the SharePlex version 11.1.

Oracle table having LONG, BINARY_FLOAT, BINARY_ DOUBLE, and NUMBER without any size.	If SP_DEQ_USE_SP_CKSUM is set to 1, then the comparison is done by fetching values instead of calculating hashes using SQL queries, which is performance-taxing. CPU utilization is expected to spike depending on the table structure, row size, and row count. If an Oracle DB table contains LONG, BINARY_FLOAT, BINARY_DOUBLE, or NUMBER without precision data types, the comparison is done by fetching values by default due to limitations in calculating hashes for these data types in SQL queries. To avoid high resource utilization in such cases, users will need to skip these columns from the comparison.

How the comparison works

A comparison detects out-of-sync conditions in a target table that are caused by DML operations:

- extra or missing rows
- rows whose values do not match

When you run the **compare** or **compare using** command on the source system, SharePlex initiates the following events:

1. The **sp_cop** process spawns a **sp_xdesvr** (server) process on the source system.
2. The **sp_desvr** process returns control and use of the **sp_ctrl** interface to the user, and replication continues while the comparison proceeds.
3. The **sp_xdesvr** process sends a message to the Post process to initiate a **sp_xdeclt** (client) process on the target system.
4. The server and client processes establish direct communication with each other.
5. The processes perform a *sanity check* by comparing the number of rows in the source and target tables. If the difference in the number of rows between a source and target table is larger than 20 percent, the command terminates. The sanity check prevents wasting time to compare or repair tables that could be fixed more quickly by reloading the target data. You can use the **copy** or **append** command to reload the target data or use another method of your choosing. To override the sanity check, reissue the compare command with the override option.
6. To override the sanity check, reissue the **Compare** command with **Override** option.
7. If the row count passes the sanity check, the tables are compared as follows:
    - If the **compare** command is being used, **sp_xdesvr** selects the source rows, and **sp_xdeclt** selects the target rows, then the rows are sorted and compared.
    - If the **compare using** command is being used, the **sp_xdesvr** process creates a number of processing threads on the target system. The value set by the SP_DEQ_THREADS parameter controls the number of threads created. Each thread spawns an **sp_xdeclt** (client) process. The server and client processes establish direct communication with each other. The processing load is divided among the client processes. The rows from each source and target table are selected, sorted and compared.
8. When finished, the processes write their log files, and you can view the results with the **show compare** command.

Manage the SQL log file

The compare and repair commands write the SQL that is needed to repair any out-of-sync rows to a SQL file in the same location as the log files. If only a compare command is issued, SharePlex does not execute these SQL statements. If a repair command is issued, the command works identically to the compare commands except that it executes the SQL statements to repair the out-of-sync rows.

You can suppress the output of the SQL log file. Some reasons to suppress this file are:

- The data contains sensitive information. The SQL log file is written in clear text. By not producing a SQL log file, the sensitive data is not persisted to disk, which may satisfy security requirements for data at rest, such as those required to meet PCI compliance standards.
- The compared or repaired tables have a very large number of out-of-sync rows. A log file of this size can consume a large amount of disk space.

To suppress the SQL log file, use the **nosqllog** option with the **compare** or **repair** command.

To suppress the output of the SQL log file for all compare and repair runs while the current instance of SharePlex is running, set the SP_SYS_SECURE_MODE environment variable to 1. This variable must be set before starting SharePlex, so if the **sp_cop** process is running it must be restarted after setting this variable. When **sp_cop** is run

with this environment variable, the compare and repair commands will not put data into SQL files and the Post process will not put data into the SharePlex error log.

Run multiple processes

All of the compare and repair commands enable you to run multiple processes concurrently.

- Multiple **compare** and **repair** commands can operate concurrently, each processing a pair of source and target tables, or you can use one command with Oracle wildcards to specify multiple sets of tables. See the (missing or bad snippet) for more information about how SharePlex supports wildcards. (applicable only for Oracle as a source)
- The **compare using** and **repair using** commands operate on an entire file. For example, you can compare or repair the tables in an entire configuration file, or you can create one or more *compare files* or *repair files* to affect a subset of the target tables, and then run one or more of them concurrently. See the command syntax for instructions.

A maximum of 20 SharePlex processes can use the post queue at the same time, including the replication processes and the comparison and repair processes. It is recommended that you allow a maximum of five comparison and repair processes to run at any given time. By using the **compare using** and **repair using** commands, you can work around the 20-process limit by comparing more tables per process.

If a comparison or repair fails because the limit is reached, SharePlex logs a message to the Event Log.

NOTE: You can run multiple commands more easily by using the **edit** command to edit a previous command to create a new one.

Compare a subset of a configuration

You can compare subsets of an active configuration file in the following ways.

- To compare all of the tables in replication that belong to one schema, use the **compare** command with a wildcard:

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

- To compare all of the tables in a configuration file, use the **compare using** command:

  ```
  sp_ctrl> compare using myconfig
  ```

- To compare all of the tables in replication to one target route, use the **compare using** command with the **at** option:

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

  PostgreSQL: `sp_ctrl>` **compare using config.active at prodsys@r.database_name**

- To compare a custom subset of the tables in a configuration, specify them in a *compare file*. This is a plain text file that lists only the source tables that you want to compare. The target tables are taken from the configuration file at the time of comparison. You can create a compare file by using the **create config** or **copy config** command. Make certain to give this file a name that makes it clear it is not a configuration file. See the command syntax for more information.

Control which rows are compared

The **compare** and **repair** commands have **where** options that enable you to filter the rows that are selected for processing. By default, these commands affect all rows of a table and ignore columns in the target table that are not contained in the source table.

- Use the **where** option to filter rows based on identically named columns in the source and target tables.
- Use the **where** option for a table that uses vertically partitioned replication. The source and target columns can have different names. Base the **where** selection on the source columns. SharePlex reads the column mappings from the configuration file to build the correct WHERE clause for the target table.
- Use the **sourcewhere** and **targetwhere** options if one or more extra columns exists in either the source or target table and those rows contain values that determine row uniqueness.
  - Use **sourcewhere** if the source table contains the extra columns.
  - Use **targetwhere** if the target table contains the extra columns.

  To use this option correctly:

  - Use a **sourcewhere** or **targetwhere** option *only* for the extra columns.
  - Use the standard **where** option for the other columns that have the same name on both source and target.
  - SharePlex combines the **where** option with the **sourcewhere** or **targetwhere** option to create the complete WHERE clause.

IMPORTANT! If you plan to run both a comparison and repair for a target table that has extra rows, only use **targetwhere** to compare for UPDATEs and DELETEs. The **repair** command cannot determine the correct values

for INSERTs. To work around this issue, set a default value for the extra columns or manually update the inserted rows.

Identify processes

Every time that a comparison or repair command is issued, the job ID is shown in the **sp_ctrl** display. If the **sp_ctrl** display is not available, you can view the job ID by running the **compare status** command.

View status and results in sp_ctrl

To view the status or results of a comparison, use the **compare status** command in **sp_ctrl**.

- The basic command displays the processing status of the most recently started comparison job, as well as other comparison jobs that are still running.
- Additional options can be used to display a status summary for comparison jobs for which there is history, or to display detailed information about a comparison job.

For more information, see Compare status.

View warnings and errors for Oracle to Oracle replication

The **sp_desvr** and **sp_declt** processes write a log file on the system where they run. The logs are stored in the **log** sub-directory of the SharePlex variable-data directory.

The name of the log written by the **sp_desvr** process is `xdesvr_<jobid>_r.<dbid>_p<process id>.log`, where:

- *JobID* is the SharePlex-assigned job ID.
- *SID* is the ORACLE_SID of the Oracle instance where the source table resides.
- *ProcessID* is the process ID of the **sp_desvr** process.

The names of the files written by the **sp_declt** process are `xdeclt_<jobid>-<tableid>_r.<dbid>_p<processid>.log` appended with either **.log** or **.sql**, where:

- *JobID* is the SharePlex-assigned job ID for the job.
- *TableID* is the SharePlex-assigned table ID for the table in the job.
- *SID* is the ORACLE_SID of the Oracle instance where the source table resides.
- *SourceHost* is the name or IP address of the source host.
- *ProcessID* is the process ID of the **sp_declt** process.

Example log file names:

```
xdesvr_7_r.aparopka_p4970.log

xdeclt_7-1_r.aparopka_p25095.log

xdeclt_7-1_r.aparopka_p25095_01.sql
```

To control disk usage, the logs are aged in a circular fashion. SharePlex generates a new log file when the current log reaches the size limit. New logs are created up to a maximum number of logs, and then SharePlex starts overwriting the oldest log.

View warnings and errors for Oracle to PostgreSQL and PostgreSQL to PostgreSQL replication

The **sp_xdesvr** and **sp_xdeclt** processes write a log file on the system where they run. The logs are stored in the **log** sub-directory of the SharePlex variable-data directory.

The name of the log written by the **sp_xdesvr** process is `xdesvr_<jobid>_r.<dbid>_p<process id>.log`, where:

- *JobID* is the SharePlex-assigned job ID.
- *DBID* is the database ID of the PostgreSQL instance.
- *ProcessID* is the process ID of the **sp_xdesvr** process.

The names of the files written by the **sp_xdeclt** process are `xdeclt_<jobid>-<tableid>_r.<dbid>_p<processid>.log` appended with either **.log** or **.sql**, where:

- *JobID* is the SharePlex-assigned job ID for the job.
- *TableID* is the SharePlex-assigned table ID for the table in the job.
- *DBID* is the database ID of the PostgreSQL instance.
- *SourceHost* is the name or IP address of the source host.
- *ProcessID* is the process ID of the **sp_declt** process.

Example log file names:

```
xdesvr_7_r.aparopka_p4970.log

xdeclt_7-1_r.aparopka_p25095.log

xdeclt_7-1_r.aparopka_p25095_01.sql
```

To control disk usage, the logs are aged in a circular fashion. SharePlex generates a new log file when the current log reaches the size limit. New logs are created up to a maximum number of logs, and then SharePlex starts overwriting the oldest log.

NOTE: For the **compare using** command, there rarely is more than one log file.

Cancel a compare job

Use the **cancel** command to stop a running comparison or repair job.

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

For more information, see Cancel.

Manage compare history and logs

SharePlex retains a history of each finished job in the database on the source system. The SP_SYS_JOB_HISTORY_RETENTION parameter controls how long history is retained.

To clear this history on demand, use the **clear history** command. When SharePlex removes the history of a job, it also removes the log file that was the source of the history.

To remove the log files from the source system without clearing the job history from the database, use the **remove log** command. You can also use this command to remove old log files from the target system.

To control the size of the log files, set the SP_DEQ_LOG_FILESIZE parameter.

To produce separate SQL files for INSERT, UPDATE, and DELETE operations, rather than use just one file for all operation types, use the **log split** option.

Control the batch size

You can control the size of the block of rows that is fetched when the process makes its SELECT query. The block size is calculated based on the value set with the **SP_DEQ_MALLOC** parameter. The value is divided equally by the number of comparison threads to be used, and then it is recalculated based on the size of all of the columns added together.

NOTE: To repair out-of-sync rows found by the comparison, use the **repair** or **repair using** command. See repair / repair using.

Usage

Supported source and target combinations:	Oracle to Oracle, Oracle to PostgreSQL, PostgreSQL to PostgreSQL
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	repair / repair using

Syntax

Command	Command options	Remote options
compare *owner.source_table [.partition]*	[**at** *target_host*@**o.***target_sid*] | (applicable only for Oracle to Oracle) [**for o.***source_sid*] | (applicable only for Oracle to Oracle) [**at** *target_host*@**r.***target_db*] | (applicable only for PostgreSQL to PostgreSQL) [**for r.***source_db*] | (applicable only for PostgreSQL to PostgreSQL)	(missing or bad snippet)

Command	Command options	Remote options
	[**hint** "*hint*"] | [**for o.***source_sid*] | (applicable only for Oracle to Oracle) [{**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*] | [**port** *port_number*] | [**quickcheck**] | [**sourcewhere** "*clause*"] | [**threads** *thread_count*] | [**targetwhere** "*clause*"] | [**to***target_owner.target_table*[*.partition*]] | [**where** "*clause*"]	
compare using *filename*	[**key**] | [**log rowdata**] | [**log split**] | [**parallelism** *degree*] | [**port** *port_number*] | [**quickcheck**] | [**threads** *threads_count*]	(missing or bad snippet)

Syntax description

Required command components

Component	Description
compare *owner.source_table[.partition]*	The basic command compares all of the source rows with all of the target rows. *owner.source_table* is the owner and name of the source table. Use double quotes to enforce case-sensitivity or spaces within a name, for example **"HR".emp**. Wildcarded table names (but not owner names) are supported. To be compared, tables that satisfy a wildcard in this command must be listed (explicitly or by wildcard) in the active replication configuration. For more information about how SharePlex handles wildcards, see the SharePlex Administration Guide. **Examples** `sp_ctrl(sysA)>`**compare scott.emp** `sp_ctrl(sysA)>`**compare scott.emp.west** When running a comparison from the command line of the operating system, if the table name is case-sensitive, quoted strings must include an extra set of escaped double quotes. For example, if the table name is `Dt_TOtalS1`, the compare command should be: `splex_install/bin/./sp_ctrl compare splex."\"Dt_TOtalS1\""`
compare using *filename*	The basic command compares all of the source rows with all of the target rows in the tables listed in *filename*. *filename* is the name of the file that contains the names of the source tables that you want to compare. **Example** `sp_ctrl(sysA)>`**compare using sales**

Optional command components

Component	Description
at *target_host@o.target_sid* (applicable only for Oracle to Oracle)	**Valid for compare** Compares the source table to only one of its targets. Use when the source table replicates to multiple target systems. *target_host* is the name of the target system. *target_sid* is the ORACLE_SID of the target Oracle instance.

Component	Description
	Example `sp_ctrl(SysA)>`**compare scott.emp at prod@o.prodsid**
for o.*SID* (applicable only for Oracle to Oracle)	**Valid for compare** Specifies the Oracle instance that contains the source table. Use when the same source table is in multiple Oracle instances on a system. *SID* is the ORACLE_SID of the source instance. It is case-sensitive and must be typed as it appears in the **oratab** file or V$PARAMETER table, or Windows Registry. When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**compare scott.emp for o.oraA**
at *target_host@r.target_database_name* (applicable only for PostgreSQL to PostgreSQL)	**Valid for compare** Compares the source table to only one of its targets. Use when the source table replicates to multiple target systems. *target_host* is the name of the target system. *target_db* is the name of the target database name. **Example** `sp_ctrl(SysA)>`**compare scott.emp at prod@r.database_name**
for r.*DBID* (applicable only for PostgreSQL to PostgreSQL)	**Valid for compare** Specifies the PostgreSQL instance that contains the source table. Use when the same source table is in multiple PostgreSQL instances on a system. *DBID* is the database name of the PostgreSQL source instance. It is case-sensitive and must be typed as it appears in the **config** file. When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**compare scott.emp for r.database_name**
hint "*hint*" (applicable only for Oracle to Oracle)	**Valid for compare**

Component	Description
	Includes an Oracle hint in the SELECT statement. The hint is used on the source and target systems. "*hint*" is a standard Oracle hint no longer than 2000 characters. Enclose the entire hint within double quotes. Omit the leading /*+ and trailing */ in the hint string. They are added by SharePlex. When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**compare scott.emp where "file >001005" hint "emp(salary)"** When running a comparison from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes as in this example: */productdir/***bin/sp_ctrl compare scott.emp hint** "\"**emp (salary)**\"" **NOTE:** The Hint component is not supported on PostgreSQL by default. To enable this feature on PostgreSQL, users must configure the following parameters: • SP_DEQ_PG_FORCE_PARALLEL • SP_DEQ_PG_PARALLEL_MAX • SP_DEQ_PG_PARALLEL_SETUP_COST • SP_DEQ_PG_PARALLEL_TUPLE_COST
{**include** | **exclude**} "(*column_list"*)	**Valid for compare** Filters the columns to be compared. • Use **include** to specify columns that you want to compare. No other columns are compared. You must include all of the key columns in an **include** clause. • Use **exclude** to compare all columns except those specified with **exclude**. Do not exclude any key columns. (*column_list*) is the list of columns to include or exclude. • Separate each name with a comma. No spaces are permitted in the list unless the name of a column contains spaces. • Enclose the column list within double quote marks.

Component	Description
	• List columns in any order. The sort is performed in ascending order. • Column names are not case-sensitive. • When used, this option must appear after the required arguments of the command, but it can appear in any order with other options. **NOTE:** There could still be rows that are out-of-sync in the columns that were not compared. **Example** `sp_ctrl (SysA)>`**compare scott.emp exclude "color, weight"**
key	**Valid for compare and compare using** Performs a fast comparison of large tables. This command does not compare all of the data values. It compares one of the following: • Only the PRIMARY key or non-null UNIQUE key columns. Or... • The columns specified with the **orderby** option. Use this option if the tables have no keys. **IMPORTANT:** Even if the keys or the **orderby** values match, the tables remain out of synchronization if values in other columns do not match. When used, this option must appear after the required command arguments. It can appear in any order with other options. Do not use this option to base a comparison on a *SharePlex key definition*. For more information about SharePlex key definitions, see the SharePlex Administration Guide. **Example** `sp_ctrl (SysA)>`**compare scott.emp key** `sp_ctrl(sysA)>`**compare using sales key**
log rowdata	**Valid for compare and compare using** Directs the client process to produce a SQL file that logs the actual row data rather than hash values. The file is produced

Component	Description
	on the target system. If the value of any of these data types is very large, SharePlex may not be able to log the actual data. **IMPORTANT!** The purpose of this SQL file is to provide a view of the data that was compared. Do not use it to repair target tables. The data values that were captured when the command was run may be different from those currently in the database. Use the **repair** command to repair any out-of-sync rows. **Example** `sp_ctrl(sysA)>`**compare scott.emp log rowdata** `sp_ctrl(sysA)>`**compare using sales log rowdata**
log split	**Valid for compare and compare using** Directs the client process to split its SQL file into three different files based on the operation type: one for INSERTs, one for UPDATEs, and one for DELETEs. **Example** `sp_ctrl(sysA)>`**compare scott.emp log split** `sp_ctrl(sysA)>`**compare using sales log split**
{**nolocktarget** | **nolocksource**}	**Valid for compare** Prevents the comparison process from locking tables during the comparison phase of a run that includes a repair. Normally, SharePlex locks the tables momentarily during a comparison to get a read-consistent view, and then unlocks them immediately, but SharePlex always locks target tables during a repair. SharePlex locks source tables momentarily during a repair to get a read-consistent view. **Example** `sp_ctrl(sysA)>`**compare scott.emp nolocksource** `sp_ctrl(sysA)>`**compare scott.emp nolocktarget**
nosqllog	Suppresses output of the SQL log file. This file contains the SQL that is needed to repair out-of-sync rows. Some reasons not to output this file include: • The data contains sensitive information. The SQL log file is written in clear text. By not producing a SQL log file, the sensitive data is not persisted to disk, which may satisfy security requirements for data at rest, such as those required to meet PCI compliance standards.

Component	Description
	• The compared or repaired tables have a very large number of out-of-sync rows. A log file of this size can consume a large amount of disk space.
not "*exception_list*"	**Valid for compare** Specifies an exception list of tables not to compare when the table specification includes wildcards. "*exception_list*" is a list of names of the tables not to compare. • Use the *owner.tablename* format. • Separate each name with a comma. No spaces are permitted in the list. • Enclose the list within double quote marks. • List the tables in any order. • When used, this option must appear after the required arguments of the command, but it can appear in any order with other options. **Example** `sp_ctrl(SysA)>`**compare scott.% not (%temp%)**
orderby "*column_list*"	**Valid for compare** Specifies columns for the comparison process to use in its ORDERBY clause when it sorts rows to be compared. This option enables comparisons to be performed on tables that have no primary or unique key. "*column_list*" is the names of the columns to use in the ORDERBY clause. • Separate each name with a comma. No spaces are permitted in the list unless the name of a column contains spaces. • Enclose the column list within double quote marks. • List columns in any order. The sort is performed in ascending order. • Column names are not case-sensitive. • When used, this option must appear after the required arguments of the command, but it can appear in any order with other options. **Example** `sp_ctrl(SysA)>`**compare scott.emp where "file >001005" orderby "Last Name,Division"**

Component	Description
	When running a comparison from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes: */productdir/***bin/sp_ctrl compare scott.emp orderby** "\"**Last Name**,**Division**\""
override	**Valid for compare and compare using** Overrides the sanity check and allows the comparison of tables that are out-of-sync beyond the sanity check threshold.Use an alternative method to re-sync the target table, similar to the initial process used for instantiating the target database/table. For more information about sanity check, see how the comparison works. **Example:** SP_ctrl (sysA) > **compare scott.emp override** SP_ctrl (sysA) > **compare using sales override**
port *port_number* (applicable only for Oracle to Oracle)	Valid for compare and compare **using** Available for backward compatibility if the version of SharePlex is earlier than 8.0 on the source or target system. Specifies a port on the source system for the client process to use for communication with the server process. In earlier versions of SharePlex, the communication is two-way, and a random port number is selected by default for client-to-server communication. This option overrides the random port selection with a specific port number, such as that required by a firewall. **Example** `sp_ctrl(sysA)>`**compare scott.emp port 1234**
quickcheck	**Valid for compare and compare using** Performs an initial check to determine if any rows are out of synchronization. As soon as one out-of-sync row is detected, the process stops. No further information is returned. No out-of-sync row information is logged to the compare SQL file. Do not use this option with the **key** option or any of the **where** options. This option does not support LONG columns. **Example** `sp_ctrl(sysA)>`**compare scott.emp quickcheck**

Component	Description
	`sp_ctrl(sysA)>`**compare using sales quickcheck**
parallelism *degree*	**Valid for compare and compare using** Adds a parallel hint to the SELECT statement. For *degree*, set the degree of parallelism. **Example** `sp_ctrl(sysA)>`**compare scott.emp parallelism 4** `sp_ctrl(sysA)>`**compare using sales parallelism 4**
sourcewhere "*clause*"	**Valid for compare** Bases the comparison on one or more columns in the source table when those columns do not exist in the target table. Rows filtered by this condition on the source table will be locked. • Enclose *clause* within double quote marks and refer to tables by their fully qualified names, for example **scott.emp**. • Use double quote marks to enforce case sensitivity or spaces within a table name. • Dates must be in the format of 'YYYYSMMDDHH24MISS'. To convert a date to that format, use the PostgreSQL TO_DATE function. For example if c1 is a DATE column, the WHERE clause `"c1 > '10-SEP-2001'"` will not work, but `"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"` will work. • When running a comparison from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes, like the following example: `sp_ctrl>`**compare scott.emp sourcewhere "\"file >001005\""** • When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example #1**: `sp_ctrl(sysA)>`**compare scott.emp sourcewhere "file >001005"** **Example #2**: The following example shows how the **sourcewhere** and **where** options are combined to get the desired result. Only the

Component	Description
	source comparison process will use the **sourcewhere** clause, but both the source and target comparison processes will use the **where** clause. `sp_ctrl(SysA)>`**compare scott.emp sourcewhere "deptno = 200" where "mgr = 'SMITH'"**
targetwhere "*clause*"	**Valid for compare** Bases the comparison on one or more columns in the target table when those columns do not exist in the source table. Rows filtered by this condition on the target table will be locked. • Enclose the clause within double quote marks and refer to tables by their fully qualified names, for example **scott.emp**. • Use double quote marks to enforce case sensitivity or spaces within a table name. • Dates must be in the format of 'YYYYSMMDDHH24MISS'. To convert a date to that format, use the PostgreSQL TO_DATE function. For example if c1 is a DATE column, the WHERE clause `"c1 > '10-SEP-2001'"` will not work, but `"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"` will work. • When running a comparison from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes: */productdir/***bin/sp_ctrl compare scott.emp targetwhere "\"file >001005\""** • When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example #1**: `sp_ctrl(SysA)>` **compare scott.emp targetwhere "file >001005"** **Example #2**: The following example shows how the **targetwhere** and **where** options are combined to get the desired result. Only the target comparison process will use the **targetwhere** clause, but both the source and target comparison processes will use the **where** clause. `sp_ctrl(SysA)>`**compare scott.emp where "deptno =**

Component	Description
	200" targetwhere "mgr = 'SMITH'" repair
threads *thread_count*	**Valid for compare and compare using** Sets the number of processing threads that are used by the comparison process. **Example** `sp_ctrl(sysA)>`**compare scott.emp threads 4** `sp_ctrl(sysA)>`**compare using sales threads 4**
to *target_owner.target_table*[.*partition*]	**Valid for compare** Compares the source table to only one of its targets. Use when the source table replicates to multiple target systems and the target tables have different names. This option can also be used to specify a target partition. `compare source_owner.source_table.[source_partition] to target_owner.target_table.[target_partition]` **Example** (Compares a partition) `sp_ctrl(SysA)>`**compare scott.emp.east to scott.allemp.alleast**
where "*clause*"	**Valid for compare** Include a WHERE clause in the SELECT statement on both the source and target systems. The WHERE clause acts as a filter to compare specific rows. Rows filtered by this condition will be locked. For "*clause*" specify a standard WHERE clause that does not include subqueries. • Enclose the clause within double quote marks and refer to tables by their fully qualified names, for example **scott.emp**. • Use double quote marks to enforce case sensitivity or spaces within a table name.

Component	Description
	• Dates must be in the format of 'YYYYSMMDDHH24MISS'. To convert a date to that format, use the PostgreSQL TO_DATE function. For example if c1 is a DATE column, the WHERE clause `"c1 > '10-SEP-2001'"` will not work, but `"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"` will work. • When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**compare scott.emp where "region=4"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:** `sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

Repair / repair using

Use the **repair** and **repair using** commands (collectively known as the *repair commands*) to repair out-of-sync rows in a target table or tables.

- The **repair** command repairs individual target tables or any number of target tables in a given schema through the use of wildcards. When repairing individual tables, you can make use of column-based filtering to control the rows that are selected for repair.
- The **repair using** command repairs all of the target tables listed in the active configuration or in another file that contains a subset of the tables in the active configuration.

The repair commands first perform a comparison to identify the rows that need to be repaired, and then they perform the repair. For more information about how tables are compared, see compare / compare using.

NOTE: A running comparison or repair does not affect the source tables in any way. SharePlex logs into the database only to query for read consistency, and the locks on the source tables are brief. SharePlex briefly locks the target tables during the processing, but users can continue accessing them with little or no awareness of the lock. Target table (or Out-Of-Sync rows if less than threshold value) is locked during the repair process to correct any Out-of-Sync rows.

Supported operations

- SharePlex can detect and repair out-of-sync rows in a target table that are caused by **DML** operations: INSERT, UPDATE, DELETE.
- Compare/Repair can be executed on single or multiple instances that have the same Oracle client version, i.e., the same ORACLE_HOME.

Non-supported operations for Oracle as a source

SharePlex does not support (and will skip) the comparison and repair of the following:

- If a table has a hash partition, comparing or repairing using the partition name is not supported. The user can use the table name instead. However, if the sub-partition is hash-based and the root partition is not, the user can use either the table name or the root partition name. **(This limitation is applicable only in Oracle to PostgreSQL replication)**
- UDT inside VARRAY. Other UDTs are supported.
- Sequences.
- Tables for which transformation was used.
- Network configurations in which a pass-through server is used to pass data between the source and target servers.
- XML type with different storage on the source and target (CLOB on source, BINARY on target, or the other way around). Like to like compare/repair is supported.
- VARRAY types not inside SDO_GEOMETRY or ST_GEOMETRY.
- Comparisons and repairs are not supported in a cascading replication environment.

- Do not perform DDL on a table that is being compared or repaired. A comparison does not detect out-of-sync conditions caused by DDL operations, including those that SharePlex supports. If the DDL changes the table definition, it invalidates the SELECT statement that is built by the comparison process to get the rows that need to be compared.

  Once you correct an out-of-sync condition caused by DDL, you can use the **repair** command to resynchronize the data in the rows.
- Comparison and repair command strings longer than 255 characters are not supported. This is an operating system limitation. To work around this limitation, use the **edit** command on the source system. You can type the command string within a text file, and then the command automatically executes the file.
- If multiple `ORACLE_HOME` directories with different Oracle versions are installed on the same machine, users can perform compare/repair operations only for the Oracle version that was used during the SharePlex configuration.

Non-supported operations for PostgreSQL as a source

- Network configurations in which a pass-through server is used to pass data between the source and target servers.
- Comparisons and repairs are not supported in a cascading replication environment.
- Comparison and repair command strings longer than 255 characters are not supported. This is an operating system limitation. To work around this limitation, use the edit command on the source system. You can type the command string within a text file, and then the command automatically executes the file.

See the (missing or bad snippet) for additional information about data types that are supported by **compare** and **compare using**

Other conditions

- The tables that you want to compare or repair must be part of an active configuration file.
- Uncommitted transactions on a source table prevent the comparison and repair processes from obtaining the brief locks they need to obtain read consistency. Make certain that all transactions are committed before you run a comparison or repair.
- Replication latency reduces the performance of compare and repair processing. The message from the source that spawns the comparison and repair processes on the target is sent through the queues with the replicated data. Delays caused by a data backlog also delay the spawn message and can cause the source process to lose its read consistency. If possible, perform comparisons and repairs during off-peak hours.
- Do not truncate a table that is being compared or repaired. The comparison commands take a snapshot of each source table when they start. If a table is truncated, the table view in the snapshot is truncated, and this can cause the command to return invalid out-of-sync conditions.
- To repair a view, the following must be true:
  - The view table must contain all of the constraint columns of the underlying table.
  - The names of the constraint columns in the view must be identical to the names of the constraint columns in the underlying table.

- If the Oracle table is set for replication to multiple targets that include Oracle and PostgreSQL (or other non-Oracle) targets, the compare command will stop with the message: "Oracle to multiple targets detected! Use 'at target' or 'to table' compare/repair options to select only one of its targets."
- When performing table-level partitioning on DATE or TIMESTAMP data types in Oracle to PostgreSQL comparisons, the NLS_CALENDAR parameter must be set to GREGORIAN on the Oracle database.
- It is expected that characters in the Oracle DB are stored with the correct byte encoding, as per the character set specified in the `NLS_CHARACTERSET` (or `NLS_NCHAR_CHARACTERSET` for `nchar`, etc.). If the byte values are corrupted, Shareplex compare/repair will not be supported. For more information on setting `NLS_LANG` at the client or session level while inserting data into the DB, refer to https://docs.oracle.com/en/database/oracle/oracle-database/21/nlspg/setting-up-globalization-support-environment.html#GUID-372A5104-DC0A-41CA-B449-6AEC93F09C2E

Supported Charactersets

The following character sets are tested and supported for Oracle database as a source:

- US7ASCII
- UTF8
- WE8ISO8859P1
- AL16UTF16
- AL32UTF8

How to use the repair commands

(missing or bad snippet)

When to run the repair

(missing or bad snippet)

Special use cases

The following scenarios require special handling when running a comparison.

Use case	Compare support
Consolidated replication	The target table in a central database has more rows than any of its contributing source databases, and often it has more columns than the source databases. Special consideration is required when using the repair commands in this environment. **repair using command** Consolidated replication is not supported by the **repair using** command. The **repair using** command will cause unwanted deletion of target rows that do not exist in those source tables.

Use case	Compare support
	As a workaround, create a subset of the configuration that excludes the tables that are involved in consolidated replication, and repair the subset configuration instead. You can use the **repair** command to repair the tables that are involved in consolidated replication. **repair command** (missing or bad snippet)
Peer-to-peer replication	In a peer-to-peer configuration, you must decide which system is the *trusted source system* and which is the *secondary*, or target, system. The secondary system is the one where any repairs will be performed. Before you run a comparison or repair in a peer-to-peer environment, follow these steps: 1. Stop user access on the secondary system and wait for replicated operations from that system to post to the trusted source database. Users can continue to access the source database. 2. Issue the **qstatus** command on the source and secondary systems. 3. When there are 10 or fewer messages in the queues, run the comparison from the source system. 4. During a comparison, you can permit user access to the source and secondary databases after the **sp_xdesvr**(PostgreSQL) or **sp_desvr**(Oracle) and **sp_xdeclt** (PostgreSQL) or **sp_declt**(Oracle) start. 5. Use the **repair** command with the **where** option to repair selected rows in a target table without locking users out of the table. For more information about this configuration, see the SharePlex Administration Guide.
Tables without keys	The comparison and repair commands issue a SELECT statement with an ORDER BY clause on the source and target systems. The ordering is faster if large tables have a primary key or a unique, non-null key and an index (preferably a unique index). Otherwise, all of the columns are used as a key. If a table has no unique row identifier, but does have one or more columns that can identify a row as unique, you can use the **compare** command with the **orderby** option. When this option is used, SharePlex prints a notice to the **sp_xdesvr** (PostgreSQL) or the **sp_desvr** (Oracle) log on the source system that the command used those columns as a key. The columns considered as keys should not contain null values if a repair needs to be done for a single row; it may result in a repair error. In this case, use the **orderby** option with columns that uniquely identify that row and do not contain null values. **LIMITATION**: Data inconsistency is likely to occur in replication or compare-repair processes when SharePlex replication involves tables containing non-key columns with duplicate data.
Target tables with more columns than the source table	The **repair** and **repair using** commands ignore target columns that are not contained in the source table. A repair does the following:

Use case	Compare support
	• An INSERT inserts values into target columns that have corresponding columns in the source table, but not into the extra columns. Columns with NOT NULL constraints but no default values cause PostgreSQL errors. Default values are recommended for extra columns in target tables. • An UPDATE resynchronizes values in target columns that have corresponding columns in the source table, but not the values in the extra columns. • A DELETE is not affected by extra columns in the target table, because the repair command selects rows based on column data in the matching columns.
Tables with a UNIQUE constraint	Columns defined with a UNIQUE constraint can cause the **repair** or **repair using** command to return unique-constraint violation errors. The following example shows source and target tables with two columns each. The first column is the primary key, and the second column has the UNIQUE constraint. When SharePlex attempts to repair row 1 of the target table to match the source, the UNIQUE constraint on column 2 returns an error because the value 'ABC' already exists in row 2. The same thing happens for row 2 of the target table, because 'XYZ' already exists in row 1. Workarounds are: • Disable UNIQUE constraints on the target table before you run a **repair** command, then enable them again after the repair is finished. • Delete the target rows for which the unique constraint violations occurred, then run the repair again so that SharePlex inserts those rows with the correct data.
Tables with LOB columns	Repairs take longer if any target tables have LOB columns. For a faster repair, you can set the SP_DEQ_SKIP_LOB parameter to 0 so that the LOB columns are skipped in the comparison and repair. For more information, see SP_DEQ_SKIP_LOB on page 1.
Repair operation on the XML data (valid only for Oracle to Oracle replication)	SharePlex displays the "`ORA-04036: PGA memory used by the instance exceeds PGA_AGGREGATE_LIMIT`" error while performing the Repair operation on the XML data. **WORKAROUND:** 1. Upgrade Oracle to version 19.19 DBRU

Use case	Compare support

	2. Set an appropriate PGA_AGGREGATE_LIMITusing the Oracle suggested script provided below :
	
	`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. Upgrade to or install the SharePlex version 11.1.

How the repair works

A repair repairs out-of-sync conditions in a target table that are caused by **DML** operations:

- extra or missing rows
- rows whose values do not match

The conditions for a repair

The **repair** and **repair using** commands issue the following corrective SQL statements:

- If a row exists on the source side but not the target side, SharePlex issues an INSERT statement.
- If a row exists on the target side but not the source side, SharePlex issues a DELETE statement.

- If a target row differs from the source row, and the key columns match, SharePlex issues an UPDATE statement based on the source values.
- If you are using the **repair** or **repair using** command for an Oracle partitioned table, the default behavior of the repair process is to repair rows by using INSERTs and DELETEs only. Repairs that require UPDATEs are converted to a DELETE followed by an INSERT to prevent errors should an UPDATE cause a row to change partitions and row movement is not enabled for the table. You can change the repair behavior so that SharePlex repairs partitioned tables by using INSERTs, UPDATEs, and DELETEs as appropriate. For more information, see SP_DEQ_PART_TABLE_UPDATE on page 1.

The process

A repair always includes a comparison to locate the out-of-sync conditions in a target table.

When you run the **repair** or **repair using** command, SharePlex initiates the following sequence of events:

1. The **sp_cop** process spawns a **sp_xdesvr** (PostgreSQL) or **sp_desvr** (Oracle) server process on the source system.
2. The **sp_xdesvr** (PostgreSQL) or the **sp_desvr** (Oracle) process returns control and use of the **sp_ctrl** interface to the user, and replication continues while the comparison proceeds.
3. The **sp_xdesvr** (PostgreSQL) or the **sp_desvr** (Oracle) process sends a message to the Post process to initiate a **sp_xdeclt** (PostgreSQL) or **sp_declt** (Oracle) client process on the target system.
4. The server and client processes establish direct communication with each other.
5. The row selection and repair proceeds as follows:
    - If the **repair** command is being used, the **sp_xdesvr** (PostgreSQL) or the **sp_desvr**(Oracle) selects the rows from the source tables, and **sp_declt** selects the rows from the target tables. The rows are sorted, compared and repaired.
    - If the **repair using** command is being used, the **sp_xdesvr** (PostgreSQL) or the **sp_desvr** (Oracle) process creates a number of processing threads on the target system. The value set by the SP_DEQ_THREADS parameter controls the number of threads created. Each thread spawns an **sp_xdeclt** (PostgreSQL) or **sp_declt** (Oracle) client process. The server and client processes establish direct communication with each other. The processing load is divided between the client processes. The rows from each source and target table are selected, sorted, compared, and repaired. The target tables are locked when it is their turn to be repaired, and then the lock is released.
    - Target tables are locked when it is their turn to be repaired, and then the lock is released.
6. When finished, the processes write their log files, and you can view the results with the **show repair** command.

Error handling

If SharePlex encounters a database error when it applies a repair SQL statement, it stops the repair from that statement forward and commits only the previously applied valid statements. Thus, the table is partially repaired, but it still could be out of synchronization. The **repair status** command alerts you to this situation.
(missing or bad snippet)

Run multiple processes

All of the compare and repair commands enable you to run multiple processes concurrently.

- Multiple **compare** and **repair** commands can operate concurrently, each processing a pair of source and target tables, or you can use one command with PostgreSQL wildcards to specify multiple sets of tables. See the (missing or bad snippet) for more information about how SharePlex supports wildcards.
- The **compare using** and **repair using** commands operate on an entire file. For example, you can compare or repair the tables in an entire configuration file, or you can create one or more *compare files* or *repair files* to affect a subset of the target tables, and then run one or more of them concurrently. See the command syntax for instructions.

A maximum of 20 SharePlex processes can use the post queue at the same time, including the replication processes and the comparison and repair processes. It is recommended that you allow a maximum of five comparison and repair processes to run at any given time. By using the **compare using** and **repair using** commands, you can work around the 20-process limit by comparing more tables per process.

If a comparison or repair fails because the limit is reached, SharePlex logs a message to the Event Log.

NOTE: You can run multiple commands more easily by using the **edit** command to edit a previous command to create a new one.

Repair a subset of a configuration

You can repair subsets of an active configuration in the following ways.

- To repair all of the target tables in replication that belong to one schema, use the **repair** command with a wildcard:

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

- To repair all of the target tables in a configuration file, use the **repair using** command:

  ```
  sp_ctrl> repair using myconfig
  ```

- To repair all of the target tables in one target route, use the **repair using** command with the **at** option:

  ```
  sp_ctrl> repair using config.active at prodsys@o.ora112 (Oracle source)
  ```

  ```
  sp_ctrl> repair using config.active at prodsys@r.dbid (PostgreSQL source)
  ```

- To repair a custom subset of the tables in a configuration, specify them in a *repair file*. This is a plain text file that lists only the source tables whose targets you want to repair. The target tables are taken from the configuration file at the time that the command is issued. You can create a repair file by using the **create config** or **copy config** command. Make certain to give this file a name that makes it clear it is not a configuration file. See the command syntax for more information.

Control which rows are repaired

(missing or bad snippet)

Identify processes

(missing or bad snippet)

View status and results in sp_ctrl

To view the status or results of a repair, use the **repair status** command in **sp_ctrl**.

- The basic command displays the processing status of the most recently started repair job, as well as any other jobs that have not yet finished.
- Additional options can be used to display a summary status for all repair jobs for which there is history, or to display detailed information about one job.

For more information, see repair status.

View warnings and errors for Oracle to Oracle replication

The **sp_desvr** and **sp_declt** processes write a log file on the system where they run. The logs are stored in the **log** sub-directory of the SharePlex variable-data directory.

The name of the log written by the **sp_desvr** process is **desvr_***JobID_SID*_**p***ProcessID*.**log**, where:

- *JobID* is the SharePlex-assigned job ID.
- *SID* is the ORACLE_SID of the Oracle instance where the source table resides.
- *ProcessID* is the process ID of the **sp_desvr** process.

The names of the files written by the **sp_declt** process are **declt_***JobIDTableID_ SID_SourceHost_***p***ProcessID* appended with either **.log** or **.sql**, where:

- *JobID* is the SharePlex-assigned job ID for the job.
- *TableID* is the SharePlex-assigned table ID for the table in the job.
- *SID* is the ORACLE_SID of the Oracle instance where the source table resides.
- *SourceHost* is the name or IP address of the source host.
- *ProcessID* is the process ID of the **sp_declt** process.

Example log file names:

```
desvr_606_ora112_p14610.log

declt_606-1_ora112_prodsys_p6528.log

declt_606-1_ora112_prodsys_p6528.sql
```

To control disk usage, the logs are aged in a circular fashion. SharePlex generates a new log file when the current log reaches the size limit. New logs are created up to a maximum number of logs, and then SharePlex starts overwriting the oldest log.

View warnings and errors for Oracle to PostgreSQL and PostgreSQL to PostgreSQL replication

The **sp_xdesvr** and **sp_xdeclt** processes write a log file on the system where they run. The logs are stored in the **log** sub-directory of the SharePlex variable-data directory.

The name of the log written by the **sp_xdesvr** process is `xdesvr_<jobid>_r.<dbid>_p<process id>.log`, where:

- *JobID* is the SharePlex-assigned job ID.
- *DBID* is the database ID of the PostgreSQL instance.
- *ProcessID* is the process ID of the **sp_xdesvr** process.

The names of the files written by the **sp_xdeclt** process are `xdeclt_<jobid>-<tableid>_r.<dbid>_p<processid>.log` appended with either **.log** or **.sql**, where:

- *JobID* is the SharePlex-assigned job ID for the job.
- *TableID* is the SharePlex-assigned table ID for the table in the job.
- *DBID* is the database ID of the PostgreSQL instance.
- *SourceHost* is the name or IP address of the source host.
- *ProcessID* is the process ID of the **sp_xdeclt** process.

Example log file names:

```
xdesvr_7_r.aparopka_p4970.log
```

```
xdeclt_7-1_r.aparopka_p25095.log

xdeclt_7-1_r.aparopka_p25095_01.sql
```

To control disk usage, the logs are aged in a circular fashion. SharePlex generates a new log file when the current log reaches the size limit. New logs are created up to a maximum number of logs, and then SharePlex starts overwriting the oldest log.

Cancel a repair job

(missing or bad snippet)

Manage compare history and logs

(missing or bad snippet)

Control the batch size

(missing or bad snippet)

Usage

Supported source and target combinations:	Oracle to Oracle, PostgreSQL to PostgreSQL, Oracle to PostgreSQL
Authorization level:	Operator (2)
Issues on:	source system
Related commands:	compare / compare using

Syntax

Basic commands	Command options	Remote options
repair *owner.source_table [.partition]*	[**at** *target_host*@**o.***target_sid*] | (applicable only for Oracle to Oracle) [**for o.***source_sid*] | (applicable only for Oracle to Oracle) [**hint** "*hint*"] | (applicable only for Oracle to Oracle) [**at** *target_host*@**r.***target_dbid*] | (PostgreSQL command option) [**for r.***source_dbid*] | (PostgreSQL command option) [{**include | exclude**} "*column_list*"] | [**insertonly**] | [**key**] |	(missing or bad snippet)

Basic commands	Command options	Remote options
	[**nosqllog**] | [**not** "*exception_list*"] | [**onepass**] | [**orderby** "*column_list*"] | [**parallelism** *degree*] | [**port***port_number*] | (applicable only for Oracle to Oracle) [**sourcewhere** "*clause*"] | [**threads** *thread_count*] | [**targetwhere** "*clause*"] | [**to** target_owner.target_table[.partition]] | [**where** "*clause*"]	
repair using *filename*	[**key**] | [**onepass**] | [**port** *port_number*] | [**threads** *threads_count*] | [**parallelism** *degree*]	(missing or bad snippet)

Syntax description

Required command components

Component	Description
repair *owner.source_table[.partition]*	The basic command repairs all of the source rows with all of the target rows. *owner.source_table* is the owner and name of the source table. Use double quotes to enforce case-sensitivity or spaces within a name, for example **"HR".emp**. Wildcarded table names (but not owner names) are supported. To be repaired, tables that satisfy a wildcard in this command must be listed (explicitly or by wildcard) in the active replication configuration. For more information about how SharePlex handles wildcards, see the (missing or bad snippet). **Example** `sp_ctrl(sysA)>`**repair scott.emp**

Component	Description
	When running a repair from the command line of the operating system, if the table name is case-sensitive, quoted strings must include an extra set of escaped double quotes. For example, if the table name is `Dt_TOtalS1`, the repair command should be: `splex_install/bin/./sp_ctrl repair splex."\"Dt_TOtalS1\""`
repair using *filename*	The basic command repairs all of the out-of-sync rows in the target tables listed in *filename*. *filename* is the name of the file that contains the names of the source tables whose targets you want to repair. **Example** `sp_ctrl(sysA)>`**repair using sales**

Optional command components

Component	Description
at *target_host@o.target_sid* (applicable only for Oracle to Oracle)	**Valid for repair** Repairs only one of the target tables in a configuration where the source table replicates to multiple target systems. *target_host* is the name of the target system. *target_sid* is the ORACLE_SID of the target Oracle instance. **Example** `sp_ctrl(SysA)>`**repair scott.emp at prod@o.prodsid**
for o.*SID* (applicable only for Oracle to Oracle)	**Valid for repair** Specifies the Oracle instance that contains the source table. Use when the same source table is in multiple Oracle instances on a system. *SID* is the ORACLE_SID of the source instance. It is case-sensitive and must be typed as it appears in the **oratab** file or V$PARAMETER table, or Windows Registry. When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**repair scott.emp for o.oraA**
at *target_host@r.target_database_name* (applicable only for PostgreSQL to	**Valid for repair**

Component	Description
PostgreSQL)	Repairs only one of the target tables in a configuration where the source table replicates to multiple target systems. *target_host* is the name of the target system. *target_db* is the name of the target database name. **Example** `sp_ctrl(SysA)>`**compare scott.emp at prod@r.database_name**
for r.*DBID* (applicable only for PostgreSQL to PostgreSQL)	**Valid for compare** Specifies the PostgreSQL instance that contains the source table. Use when the same source table is in multiple PostgreSQL instances on a system. *DBID* is the database name of the PostgreSQL source instance. It is case-sensitive and must be typed as it appears in the **config** file. When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**compare scott.emp for r.database_name**
{**include** | **exclude**} "(*column_list*)"	**Valid for repair** Filters the columns to be repaired. • Use **include** to specify columns that you want to repair. No other columns are repaired. You must include all of the key columns in an **include** clause. • Use **exclude** to repair all columns except those specified with **exclude**. Do not exclude any key columns. (*column_list*) is the list of columns to include or exclude. • Separate each name with a comma. No spaces are permitted in the list unless the name of a column contains spaces. • Enclose the column list within double quote marks. • List columns in any order. The sort is performed in ascending order. • Column names are not case-sensitive. • When used, this option must appear after the required arguments of the command, but it can appear in any order with other options.

Component	Description
	NOTE: There could still be rows that are out-of-sync in the columns that were not repaired. **Example** `sp_ctrl (SysA)>`**repair scott.emp exclude "color, weight"**
hint "*hint*" (applicable only for Oracle to Oracle)	**Valid for repair** Includes an Oracle hint in the SELECT statement. The hint is used on the source and target systems. "*hint*" is a standard Oracle hint no longer than 2000 characters. Enclose the entire hint within double quotes. Omit the leading /*+ and trailing */ in the hint string. They are added by SharePlex. When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**repair scott.emp where "file >001005" hint "emp(salary)"** When running a repair from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes: */productdir/***bin/sp_ctrl repair scott.emp hint** "\"**emp(salary)**\"" **NOTE:** The Hint component is not supported in PostgreSQL by default. To enable this feature on PostgreSQL, users must configure the following parameters: • SP_DEQ_PG_FORCE_PARALLEL • SP_DEQ_PG_PARALLEL_MAX • SP_DEQ_PG_PARALLEL_SETUP_COST • SP_DEQ_PG_PARALLEL_TUPLE_COST
insertonly	**Valid for repair** Repairs the target table for INSERT statements only. **Example:** `sp_ctrl(SysA)>`**repair scott.emp insertonly**
key	**Valid for repair and repair using** Performs a fast compare and repair of large tables. This command does not compare all of the data values, but only compares one of the following:

Component	Description
	• Only the PRIMARY key or non-null UNIQUE key columns. Or... • The columns specified with the **orderby** option. Use this option if the tables have no keys. If the key or **orderby** columns do not match, SharePlex repairs the entire row by deleting it and then inserting it again based on the source values. **IMPORTANT!** Use this option with caution. Even if key values match, it is possible for values in non-key columns to be out of synchronization. When used, this option must appear after the required command arguments. It can appear in any order with other options. Do not use this option to base a repair on a SharePlex key definition. For more information about SharePlex key definitions, see the (missing or bad snippet). **Example** `sp_ctrl (SysA)>`**repair scott.emp key** `sp_ctrl(sysA)>`**repair using sales key**
nosqllog	Suppresses output of the SQL log file. This file contains the SQL that is needed to repair out-of-sync rows. Some reasons not to output this file include: (missing or bad snippet)
not "*exception_list*"	**Valid for repair** Specifies an exception list of tables not to repair when the table specification includes wildcards. "*exception_list*" is a list of names of the tables not to repair. • Use the *owner.tablename* format. • Separate each name with a comma. No spaces are permitted in the list. • Enclose the list within double quote marks. • List the tables in any order. • When used, this option must appear after the required arguments of the command, but it can appear in any order with other options. **Example** `sp_ctrl(SysA)>`**repair scott.% not (%temp%)**

Component	Description
onepass	**Valid for repair and repair using** Use this option to run a compare and repair concurrently. Use it for large out-of-sync tables. Normally, a repair runs in two passes: first a compare, then a repair, which locks the target table. Both passes require a consistent view. With **onepass**, the target table is locked and repaired as soon as the compare client receives the consistent view marker. **Example** `sp_ctrl(SysA)>`**repair scott.emp onepass** **NOTE:** For PostgreSQL, if a table has more than one key defined, using the 'onepass' command option may result in a duplicate constraint violation error during the repair process. It is recommended not to use the 'onepass' command option for such tables, as repair will work correctly without it. For tables with a single key defined, repairing with the 'onepass' option will also work correctly. In the case of a table with no key defined in a PostgreSQL database, the 'onepass' option should be used for repair if there are duplicate records. Additionally, UPDATE or DELETE statements may not work accurately because the WHERE clause can affect more rows than intended.
orderby "*column_list*"	**Valid for repair** Use the ORDERBY clause command option in conjunction with the KEY command option to compare the columns specified with the ORDERBY option. If the key or ORDERBY columns do not match, SharePlex repairs the entire row by deleting it and then inserting it again based on the source values. Specifies columns for the repair process to use in its ORDERBY clause when it sorts rows to be compared. This option enables repairs to be performed on tables that have no primary or unique key. "*column_list*" is the names of the columns to use in the ORDERBY clause. • Separate each name with a comma. No spaces are permitted in the list unless the name of a column contains spaces. • Enclose the column list within double quote marks. • List columns in any order. The sort is performed in ascending order.

Component	Description
	• Column names are not case-sensitive. • When used, this option must appear after the required arguments of the command, but it can appear in any order with other options. • When running a repair from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes: */productdir/***bin/sp_ctrl repair scott.emp orderby** "\"**Last Name**,**Division**\"" **Example** `sp_ctrl(SysA)>`**repair scott.emp where "file >001005" orderby "Last Name,Division"**
parallelism *degree*	**Valid for repair and repair using** Adds a parallel hint to the SELECT statement. For *degree*, set the degree of parallelism. **Example** `sp_ctrl(sysA)>`**repair scott.emp parallelism 4** `sp_ctrl(sysA)>`**repair using sales parallelism 4**
port *port_number* (applicable only for Oracle to Oracle)	Valid for **repair** and **repair using** Available for backward compatibility if the version of SharePlex is earlier than 8.0 on the source or target system. Specifies a port on the source system for the client process to use for communication with the server process. In earlier versions of SharePlex, the communication is two-way, and a random port number is selected by default for client-to-server communication. This option overrides the random port selection with a specific port number, such as that required by a firewall. **Example** `sp_ctrl(sysA)>`**compare scott.emp port 1234**
sourcewhere "*clause*"	**Valid for repair** Bases the repair on one or more columns in the source table when those columns do not exist in the target table. Rows filtered by this condition on the source table will be locked. • Enclose *clause* within double quote marks and refer to tables by their fully qualified names, for example **scott.emp**.

Component	Description
	• Use double quote marks to enforce case sensitivity or spaces within a table name. • Dates must be in the format of 'YYYYSMMDDHH24MISS'. To convert a date to that format, use the PostgreSQL TO_DATE function. For example if c1 is a DATE column, the WHERE clause "`c1 > '10-SEP-2001'`" will not work, but "`c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')`" will work. • When running a repair from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes, like the following example: `sp_ctrl>`**repair scott.emp sourcewhere "\"file >001005\""** • When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example #1**: `sp_ctrl(sysA)>`**repair scott.emp sourcewhere "file >001005"** **Example #2**: The following example shows how the **sourcewhere** and **where** options are combined to get the desired result. Only the source repair process uses the **sourcewhere** clause, but both the source and target repair processes use the **where** clause. `sp_ctrl(SysA)>`**repair scott.emp sourcewhere "deptno = 200" where "mgr = 'SMITH'"**
targetwhere "*clause*"	**Valid for repair** Bases the repair on one or more columns in the target table when those columns do not exist in the source table. Rows filtered by this condition on the target table will be locked. • Enclose the clause within double quote marks and refer to tables by their fully qualified names, for example **scott.emp**. • Use double quote marks to enforce case sensitivity or spaces within a table name. • Dates must be in the format of 'YYYYSMMDDHH24MISS'. To convert a date to that format, use the PostgreSQL TO_DATE function. For example if c1 is a DATE column, the WHERE clause "`c1 > '10-SEP-2001'`" will not work, but "`c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')`" will work.

Component	Description
	• When running a comparison from the command line of the operating system, quoted strings must have an extra set of *escaped* double quotes: */productdir/***bin/sp_ctrl repair scott.emp targetwhere "\"file >001005\""** • When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example #1**: `sp_ctrl(SysA)>` **repair scott.emp targetwhere "file >001005"** **Example #2**: The following example shows how the **targetwhere** and **where** options are combined to get the desired result. Only the target repair process will use the **targetwhere** clause, but both the source and target repair processes will use the **where** clause. `sp_ctrl(SysA)>`**repair scott.emp where "deptno = 200" targetwhere "mgr = 'SMITH'" repair**
threads *thread_count*	**Valid for repair and repair using** Sets the number of processing threads that are used by the repair process. **Example** `sp_ctrl(sysA)>`**repair scott.emp threads 4** `sp_ctrl(sysA)>`**repair using sales threads 4**
to *target_owner.target_table [.partition*]	**Valid for repair** Repairs only one of the targets of a source table. Use when the source table replicates to multiple target systems and the target tables have different names. This option can also be used to specify a target partition. `compare source_owner.source_table.[source_ partition] to target_owner.target_table.[target_ partition]` **Example** (Repairs a partition) `sp_ctrl(SysA)>`**repair scott.emp.east to scott.allemp.alleast**
where "*clause*"	**Valid for repair**

Component	Description
	Include a WHERE clause in the SELECT statement on both the source and target systems. The WHERE clause acts as a filter to repair specific rows. Rows filtered by this condition will be locked. For "*clause*" specify a standard WHERE clause that does not include subqueries. • Enclose the clause within double quote marks and refer to tables by their fully qualified names, for example **scott.emp**. • Use double quote marks to enforce case sensitivity or spaces within a table name. • Dates must be in the format of 'YYYYSMMDDHH24MISS'. To convert a date to that format, use the PostgreSQL TO_DATE function. For example if c1 is a DATE column, the WHERE clause `"c1 > '10-SEP-2001'"` will not work, but `"c1 > to_date('10- SEP-2001', 'DD-MON-YYYY')"` will work. • When used, this option must appear after the required command arguments, but it can appear in any order with other options. **Example** `sp_ctrl (SysA)>`**repair scott.emp where "region=4"**

Remote options

These options enable you to issue the command on a remote machine and to script commands that include a login name, password, port number, or combination of those items.

Option	Description
on *host*	Execute the command on a remote system (one other than the one where the current **sp_ctrl** session is running). You are prompted for login credentials for the remote system. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on SysA**
on *host:portnumber*	Execute the command on a remote system when a remote login and port number must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on SysA:8304**
on *login/password@host*	Execute the command on a remote system when a remote login, password, and host name must be provided. If used, must be the last component of the command syntax.

Option	Description
	Example:`sp_ctrl(sysB)>`**status on john/spot5489@SysA**
on *login/password@host:portnumber*	Execute the command on a remote system when a remote login, password, host name, and port number must be provided. If used, must be the last component of the command syntax. **Example:**`sp_ctrl(sysB)>`**status on john/spot5489@SysA:8304**

5

General SharePlex Utilities

The SharePlex utilities help you configure, test, and manage the SharePlex environment.

Contents

- Configuration Scripts
- Configuration Scripts for PostgreSQL
- Cleanup.sql
- Clean_vardir.sh
- Create_ignore.sql
- MSS_cleansp
- MySQL_cleansp
- Ora_cleansp
- pg_cleansp Utility
- Snowflake_Cleansp Utility
- SharePlex License Utilities
- Provision
- Qview
- Show_scn utility
- show_last_posted Utility
- Socket_test
- SP-bininfo
- SP_wallet
- sp_hsm
- sp_security
- Trigger Scripts
- Trigger Scripts for PostgreSQL
- Database Setup Utilities

Configuration Scripts

Description

SharePlex provides the following scripts to automate the building of a configuration file to specify Oracle source objects.

- **config.sql**: configure all tables and optionally all sequences in the database.
- **build_config.sql**: configure multiple or all tables in a schema

Supported source and target database

Oracle to Oracle

Use config.sql

The **config.sql** script enables you to build a configuration that lists all of the tables, and optionally all of the sequences, in all of the schemas of a database. This script saves time when establishing a high-availability replication strategy or other scenario where you want the entire database to be replicated to an identical secondary database.

Conditions for using config.sql

- Source and target table names must be the same.
- The script does not configure objects in the SYS, SYSTEM, and SharePlex schemas. These schemas cannot be replicated since they are system and/or instance-specific.
- The script does not support partitioned replication. You can use the **copy config** command to copy the configuration file that the script builds, then use the **edit config** command to add entries for tables that use partitioned replication. Activate the new configuration file, not the original one.
- You can use the **edit config** command to make any other changes as needed after the configuration is built.

To run config.sql:

1. Change directories to the **config** sub-directory of the SharePlex variable-data directory. The **config.sql** script puts configurations in the current working directory, and SharePlex configurations must reside in the **config** sub-directory.

   cd */vardir*/**config**

2. Log onto SQL*Plus as SYSTEM.
3. Run **config.sql** using the full path from the **util** sub-directory of the SharePlex product directory.

   @ */proddir*/**util/config.sql**

Refer to the following table when following the prompts:

Prompt	What to enter
Target machine	The name of the target machine, for example SystemB.
Source database SID	The ORACLE_SID of the source (primary) Oracle instance, for example oraA. Do not include the **o.** keyword. The ORACLE_SID is case-sensitive.
Target database SID	The ORACLE_SID of the target (destination) Oracle instance, for example oraB. Do not include the **o.** keyword. The ORACLE_SID is case-sensitive.
Replicate sequences	Enter **y** to replicate sequences or **n** not to replicate sequences.
SharePlex oracle username	The name of the SharePlex user in the source database. This entry prevents the SharePlex schema from being replicated, which would cause replication problems. If a valid name is not provided, the script fails.

NOTE: The name assigned by SharePlex to the configuration is **config.file**. If you run the script again to create another configuration file, it overwrites the first file. To preserve the original file, rename it before you create the second one.

Next steps:

- If any tables or owners are case-sensitive, open the configuration file with the **edit config** command in **sp_ctrl**, then use the text editor to enclose case-sensitive table and owner names within double-quote marks, for example “scott”.“emp”. The script does not add the quote marks required by Oracle to enforce case-sensitivity.

  ```
  sp_ctrl> edit config filename
  ```

- To ensure that the configuration is in the correct location, issue the **list config** command. If the name of the configuration is not shown, it was created in the wrong directory. Find the file and move it to the **config** sub-directory of the variable-data directory.

  ```
  sp_ctrl> list config
  ```

Use build_config.sql

The **build_config.sql** script enables you to build a configuration that contains multiple (or all) tables in a schema. It is an interactive script that prompts for each component of the configuration step by step. Instead of entering the information for each object and the routing individually, you can use a wildcard to select certain tables at once, or you can select all of the tables in the schema.

Conditions for using build_config.sql

- Source and target table names must be the same.
- The script does not support sequences. Before you activate the configuration that the script builds, you can use the **edit config** command in **sp_ctrl** to add entries for sequences.
- The script does not support partitioned replication. You can use the **copy config** command to copy the configuration that the script builds, then use the **edit config** command to add entries for the tables that use partitioned replication. Activate the new configuration, not the original.

- The script does not configure objects in the SYS, SYSTEM, and SharePlex schemas. These schemas cannot be replicated since they are system and/or instance-specific.
- You can run **build_config.sql** for different schemas, then combine those configurations into one configuration by using a text editor. Make certain to eliminate all but one **Datasource:o.SID** line, which is the first non-commented line of the file. Do not move the file out of the **config** sub-directory.
- You can use the **edit config** command to make any other changes as needed after the configuration is built.

To run build_config.sql:

1. Change directories to the **config** sub-directory of the SharePlex variable-data directory. The **build_config.sql** script puts configurations in the current working directory, and SharePlex configurations must reside in the **config** sub-directory.

   cd /*vardir*/**config**

2. Log onto SQL*Plus as SYSTEM.
3. Run **build_config.sql** using the full path from the **util** sub-directory of the SharePlex product directory.

   @ /*proddir*/**util/build_config.sql**

Refer to the following table when following the prompts.

Prompt	What to enter
Target machine	The name of the target machine, for example SystemB.
Source database SID	The ORACLE_SID of the source (primary) Oracle instance, for example oraA. Do not include the **o.** keyword. The ORACLE_SID is case-sensitive.
Target database SID	The ORACLE_SID of the target (destination) Oracle instance, for example oraB. Do not include the **o.** keyword. The ORACLE_SID is case-sensitive.
Owner of the source database tables	The owner of the source tables.
Owner of the target database tables	The owner of the target tables.
Table name to include (blank for all)	Do one of the following: • Press **Enter** to accept the default, which selects all tables that belong to the source owner. • Enter a wildcard (%) character and a string to select certain tables, for example **%e_salary%**. • Enter an individual table name.
Name of the output file to create	A name for the configuration. The script gives the file a **.lst** suffix, for example **Scott_config.lst**.

Next steps:

- If any tables or owners are case-sensitive, open the configuration with the **edit config** command in **sp_ctrl**, then use the text editor to enclose case-sensitive table and owner names within double-quote marks, for example “scott”.“emp”. The script does not add the quote marks required by Oracle to enforce case-sensitivity.

  ```
  sp_ctrl> edit config filename
  ```

- To ensure that the configuration is in the correct location, issue the **list config** command. If the name of the configuration is not shown, it was created in the wrong directory. Find the file and move it to the **config** sub-directory of the variable-data directory.

  ```
  sp_ctrl> list config
  ```

Configuration Scripts for PostgreSQL

Description

SharePlex provides the following scripts to automate the building of a configuration file to specify on-prem PostgreSQL and PostgreSQL Database as a Service source objects.

- **pg_config.sql**: configure all tables in the database.
- **pg_build_config.sql**: configure multiple or all tables in a schema

Supported source and target database

PostgreSQL to PostgreSQL

Use pg_config.sql

The **pg_config.sql** script enables you to build a configuration that lists all of the tables, and optionally all of the sequences, in all of the schemas of a database. This script saves time when establishing a high-availability replication strategy or other scenario where you want the entire database to be replicated to an identical secondary database.

Conditions for using pg_config.sql

- Source and target table names must be the same.
- The script does not support partitioned replication. You can use the **copy config** command to copy the configuration file that the script builds, then use the **edit config** command to add entries for tables that use partitioned replication. Activate the new configuration file, not the original one.
- You can use the **edit config** command to make any other changes as needed after the configuration is built.

To run pg_config.sql:

1. Change directories to the **config** sub-directory of the SharePlex variable-data directory. The **pg_config.sql** script puts configurations in the current working directory, and SharePlex configurations must reside in the **config** sub-directory.

   cd /*vardir*/**config**

2. Log onto PostgreSQL.
3. Run **pg_config.sql** using the full path from the **util** sub-directory of the SharePlex product directory.

   \i *proddir*/**util/pg_config.sql**

Refer to the following table when following the prompts:

Prompt	What to enter
Target machine	The name of the target machine, for example SystemB.
Source database name	The database name of the source (primary) PostgreSQL instance, for example dbnameA. Do not include the **r.** keyword. The database name is case-sensitive.
Target database name	The database name of the target (destination) PostgreSQL instance, for example dbnameB. Do not include the **r.** keyword. The database name is case-sensitive.
SharePlex PostgreSQL user name	The name of the SharePlex user in the source database. This entry prevents the SharePlex schema from being replicated, which would cause replication problems. If a valid name is not provided, the script fails.

NOTE: The name assigned by SharePlex to the configuration is **pg_config.file**. If you run the script again to create another configuration file, it overwrites the first file. To preserve the original file, rename it before you create the second one.

Next steps:

- If any tables or schemas are case-sensitive, open the configuration file with the **edit config** command in **sp_ctrl**, then use the text editor to enclose case-sensitive table and schema names within double-quote marks, for example “SCOTT”.“EMP”. The script does not add the quote marks required by PostgreSQL to enforce case-sensitivity.

  ```
  sp_ctrl> edit config filename
  ```

- To ensure that the configuration is in the correct location, issue the **list config** command. If the name of the configuration is not shown, it was created in the wrong directory. Find the file and move it to the **config** sub-directory of the variable-data directory.

  ```
  sp_ctrl> list config
  ```

Use pg_build_config.sql

The **build_config.sql** script enables you to build a configuration that contains multiple (or all) tables in a schema. It is an interactive script that prompts for each component of the configuration step by step. Instead of entering the information for each object and the routing individually, you can use a wildcard to select certain tables at once, or you can select all of the tables in the schema.

Conditions for using pg_build_config.sql

- Source and target table names must be the same.
- The script does not support partitioned replication. You can use the **copy config** command to copy the configuration that the script builds, then use the **edit config** command to add entries for the tables that use partitioned replication. Activate the new configuration, not the original.
- You can run **pg_build_config.sql** for different schemas, then combine those configurations into one configuration by using a text editor. Make certain to eliminate all but one **Datasource:r.dbname** line, which is the first non-commented line of the file. Do not move the file out of the **config** sub-directory.
- You can use the **edit config** command to make any other changes as needed after the configuration is built.

To run pg_build_config.sql:

1. Change directories to the **config** sub-directory of the SharePlex variable-data directory. The **pg_build_config.sql** script puts configurations in the current working directory, and SharePlex configurations must reside in the **config** sub-directory.

   cd /*vardir*/**config**

2. Log onto PostgreSQL.
3. Run **pg_build_config.sql** using the full path from the **util** sub-directory of the SharePlex product directory.

   \i *proddir*/**util/pg_build_config.sql**

Refer to the following table when following the prompts.

Prompt	What to enter
Target machine	The name of the target machine, for example SystemB.
Source database name	The database name of the source (primary) PostgreSQL instance, for example dbnameA. Do not include the **r.** keyword. The database name is case-sensitive.
Target database name	The database name of the target (destination) PostgreSQL instance, for example dbnameB. Do not include the **r.** keyword. The database name is case-sensitive.
Source database schema	Name of the source database schema.
Target database schema	Name of the target database schema.
Table name to include (blank for all)	Do one of the following: • Press **Enter** to accept the default, which selects all tables that belong to the source owner. • Enter a wildcard (%) character and a string to select certain tables, for example **%e_salary%**. • Enter an individual table name.
Name of the output file to create	A name for the configuration.

Next steps:

- If any tables or schemas are case-sensitive, open the configuration with the **edit config** command in **sp_ctrl**, then use the text editor to enclose case-sensitive table and schema names within double-quote marks, for example “SCOTT”.“EMP”. The script does not add the quote marks required by PostgreSQL to enforce case-sensitivity.

  ```
  sp_ctrl> edit config filename
  ```

- To ensure that the configuration is in the correct location, issue the **list config** command. If the name of the configuration is not shown, it was created in the wrong directory. Find the file and move it to the **config** sub-directory of the variable-data directory.

  ```
  sp_ctrl> list config
  ```

Cleanup.sql

Description

Use the **cleanup.sql** script to truncate all of the SharePlex internal tables except the SHAREPLEX_PARTITION table (which contains partition schemes that might be needed again).

NOTE: The **cleanup.sql** script does not remove the SharePlex Oracle user, password, or demonstration objects from the SharePlex tables.

The **cleanup.sql** script preserves the replication data in the SharePlex variable-data directory. Other utilities provide related options:

- To clean out the variable-data directory without truncating the SharePlex tables, see Clean_vardir.sh on page 506.
- To clean out the variable-data directory and truncate the SharePlex tables, see Ora_cleansp on page 514. This utility completely restore SharePlex to an initial state.

Contact Quest Technical Support before running **cleanup.sql** for the first time. Unless a procedure in the SharePlex documentation requires running **clean_vardir.sh**, this utility rarely is appropriate in a production environment. It deactivates the configuration, and using it improperly can result in replication problems and the need to resynchronize the data. Usually, there is another alternative.

Supported databases

Oracle on Unix and Windows

To run cleanup.sql:

1. Log into Oracle as the SharePlex database user. The SharePlex tables belong to that user. On Unix and Linux, If you are running multiple instances of **sp_cop** with multiple variable-data directories, there is a SharePlex Oracle user for each one. Make certain to run this script as the correct one.
2. (Unix and Linux) Set the SP_SYS_VARDIR environment variable to point to the SharePlex variable-data directory.

   ksh shell:

   export SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

   csh shell:

   setenv SP_SYS_VARDIR=*/full_path_of_variable-data_directory*
3. Run **cleanup.sql** as a SharePlex Administrator. The script is in the **bin** sub-directory of the SharePlex product directory. Use the following syntax, where *Oracle_version* is one of the SharePlex-supported Oracle versions.

   `SQL>` **@***proddir*/**bin/cleanup.sql**

Clean_vardir.sh

Description

Use the **clean_vardir.sh** script to clean out the variable-data directory to restore it to an initial state.

The **clean_vardir.sh** script preserves the contents of the SharePlex internal tables.

- To truncate the SharePlex tables, without cleaning out the variable-data directory, see Cleanup.sql.
- To clean out the variable-data directory and truncate the SharePlex tables, see the appropriate *database*_**cleansp** utility, where database is the type of database. This utility completely restore SharePlex to an initial state.

IMPORTANT! Contact Quest Technical Support before running **clean_vardir.sh** for the first time. Unless a procedure in the SharePlex documentation requires running **clean_vardir.sh**, this utility rarely is appropriate in a production environment. It deactivates the configuration, and using it improperly can result in replication problems and the need to resynchronize the data. Usually, there is another alternative.

What this utility does

The **clean_vardir.sh** script removes the following:

- the queue files in the **rim** sub-directory.
- the log files in the **log** sub-directory. The Event log retains one entry reflecting the **clean_vardir.sh** procedure.
- the contents of the **statusdb** file in the **data** sub-directory.
- the contents of the **dump** and **state** sub-directories.

The **clean_vardir.sh** script preserves user-created files such as configuration files, conflict-resolution files, hint files, the **paramdb**, and the **oramsglist** file.

The **clean_vardir.sh** script deactivates configurations. To start replication after running **clean_vardir.sh**, you must activate a configuration.

Supported databases

Oracle on Unix and Linux

Shell requirement

To use this utility, the Korn (ksh) shell must be installed on the system. The utility calls this shell during processing.

To run clean_vardir.sh:

NOTE: Run this script on Unix and Linux systems only.

1. Shut down **sp_cop**.
2. Set the SP_SYS_VARDIR environment variable to point to the SharePlex variable-data directory. If SP_SYS_VARDIR is not set, **clean_vardir.sh** affects the directory listed in the *proddir*/**data**/**default.yaml** file, where the *proddir* is the **bin** sub-directory of the SharePlex product directory.

   ksh shell:

   export SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

   csh shell:

   setenv SP_SYS_VARDIR=*/full_path_of_variable-data_directory*
3. Run **clean_vardir.sh** as a SharePlex Administrator. The script is in the **bin** sub-directory of the SharePlex product directory. Use the following syntax, where *Oracle_version* is one of the SharePlex-supported Oracle versions.

   proddir/**bin**/**clean_vardir.sh** *Oracle_version*

When the script is finished running, you are returned to the command prompt.

NOTE: If the script generates an error message stating that it cannot remove the **save_***SharePlex_version* directory, you can remove that directory manually.

Create_ignore.sql

Description

Use the **create_ignore.sql** utility script to prevent DML transactions from being replicated to the target system. This script creates a public procedure named SHAREPLEX_IGNORE_TRANS in the source database. When executed at the start of a transaction, the procedure makes the Capture process ignore DML operations that occur from the point of execution until the transaction is either committed or rolled back. Thus, the affected operations are not replicated.

Only DML operations are affected by the SHAREPLEX_IGNORE_TRANS procedure. It does not cause SharePlex to skip DDL operations, including TRUNCATE. DDL operations are implicitly committed by Oracle, so they render the procedure invalid.

Supported databases

Oracle on Unix and Windows

Run create_ignore.sql

Run the **create_ignore.sql** script from the **util** sub-directory in the SharePlex product directory. Run it as the SharePlex Oracle user so that the procedure is created in the SharePlex schema.

The script executes the following PL/SQL:

```
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;

/
```

Execute SHAREPLEX_IGNORE_TRANS

Call SHAREPLEX_IGNORE_TRANS only at the beginning of a transaction containing operations that you do not want replicated. If it is called in the middle of a transaction, replicated operations preceding the start of the procedure will remain in the post queue indefinitely awaiting a COMMIT, because SharePlex does not release messages without one. The COMMIT will not arrive because Capture ignores all operations in the transaction after the procedure is called. The Read process will retain unwanted cache information on those records indefinitely.

1. Log onto SQL*Plus as the SharePlex user.
2. Execute the SHAREPLEX_IGNORE_TRANS procedure at the beginning of the transaction that you want to skip.

   ```
   execute SharePlex_schema.SHAREPLEX_IGNORE_TRANS;
   ```

3. In one transaction, make the changes that you do not want replicated. They will be ignored by Capture.
4. After the COMMIT or ROLLBACK, replication of subsequent transactions resumes normally.

MSS_cleansp

Description

Use the **mss_cleansp** utility to remove the current replication state on a SQL Server system.

CAUTION:

- Running **mss_cleansp** can be acceptable in a test or demonstration environment, but it rarely is appropriate in a production environment. The effects of the cleanup are not reversible.
- To fully remove the replication environment, run the **mss_cleansp** utility on the target system. If the source system is Oracle, run the **ora_cleansp** utility. To verify if and when **mss_cleansp** or **ora_cleansp** was run on a system, view the SharePlex event log on that system.
- If the SharePlex instance for which you are running **mss_cleansp** is configured to post to multiple databases, **mss_cleansp** will remove the replication configuration and supporting data **for all of those databases**.

mss_cleansp does the following on the target system:

- Truncates the SHAREPLEX_OPEN_TRANS internal table and any other SharePlex-installed tables as applicable.
- Removes the following from the variable-data directory: the queue files, the process log files, the contents of the **statusdb** (but not the file), the contents of the **dump** and **state** directories, and all but one entry in the event log (the status entry for **mss_cleansp**)

mss_cleansp preserves the following:

- The SharePlex database, account, and password, and only cleans the data in the SharePlex tables.
- User-created files such as the **paramdb** and the target configuration settings

Supported databases

Microsoft SQL Server on Windows

To run mss_cleansp:

1. Stop all SharePlex processes on the system.
2. Shut down **sp_cop**.
3. Run the **mss_cleansp** utility from the **bin** sub-directory of the SharePlex product directory with the following syntax:

   `C:\users\splex\bin>` **mss_cleansp** [**port** *number*] *database_name*/*user_name*

   where:

   - **port** *number* is the port number of the SharePlex instance, if other than the default of 2100.
   - *database_name* is the name of the SharePlex database.
   - *user_name* is the name of the SharePlex user account.

4. Enter the password at the prompt.

```
Enter the password for the SharePlex User :
```

5. Type **Y** to confirm that you want to run cleanup for this SharePlex instance.

```
Are you sure you want to perform the clean for 'C:\splex\vardir\var2200' and
port 2200? [N] :Y
```

MySQL_cleansp

Description

Use the **mysql_cleansp** utility to remove the current replication state on a MySQL system.

CAUTION: Running **mysql_cleansp** can be acceptable in a test or demonstration environment, but it rarely is appropriate in a production environment. The effects of the cleanup are not reversible.

Usage guidelines

- To use this utility, the Korn (ksh) shell must be installed on the system. The utility calls this shell during processing.
- **mysql_cleansp** must be run on all MySQL target systems in the replication configuration.
- To fully remove the replication environment, run the **ora_cleansp** utility on all Oracle source systems in the replication configuration. To verify if and when **mysql_cleansp** or **ora_cleansp** was run on a system, view the SharePlex event log on that system.

What this utility does

mysql_cleansp does the following on the target system:

- Truncates the SHAREPLEX_OPEN_TRANS internal table and any other SharePlex-installed tables as applicable.
- Removes the following from the variable-data directory: the queue files, the process log files, the contents of the **statusdb** (but not the file), the contents of the **dump** and **state** directories, and all but one entry in the event log (the status entry for **mysql_cleansp**)

mysql_cleansp preserves the following:

- The SharePlex database, account, and password, and only cleans the data in the SharePlex tables.
- User-created files such as the **paramdb** and the target configuration settings

Supported databases

MySQL on supported UNIX platforms

To run mysql_cleansp:

1. Stop all SharePlex processes on the system.
2. Shut down **sp_cop**.
3. Run the **mysql_cleansp** utility from the **bin** sub-directory of the SharePlex product directory with the following syntax:

   ```
   C:\users\splex\bin> mysql_cleansp [port number ] database_name/user_name
   ```

   where:

   - **port** *number* is the port number of the SharePlex instance, if other than the default of 2100.
   - *database_name* is the name of the SharePlex database.
   - *user_name* is the name of the SharePlex user account.

4. Enter the password at the prompt.

   ```
   Enter the password for the SharePlex User :
   ```

5. Type **Y** to confirm that you want to run cleanup for this SharePlex instance.

   ```
   Are you sure you want to perform the clean for '/splex/vardir/var2200' and
   port 2200? [N] : Y
   ```

Ora_cleansp

Description

Use the **ora_cleansp** utility to remove the current replication state on a system, including deactivating configurations.

CAUTION:

- Running **ora_cleansp** can be acceptable in a test or demonstration environment, but it rarely is appropriate in a production environment. The effects of the cleanup are not reversible.
- If the SharePlex instance for which you are running **ora_cleansp** is configured to capture from or post to multiple databases, **ora_cleansp** will remove the replication configuration and supporting data **for all of those databases**.
- While running the **ora_cleansp** utility, enter the TNS alias provided while performing **ora_setup**.

Supported databases

Oracle on Unix, Linux and Windows

Other alternatives to ora_cleansp

Before running **ora_cleansp**, consider the following alternatives:

- To stop activity for a configuration without restoring replication to an initial state, you might be able to use the **abort config** or **purge config** command in **sp_ctrl**.
- To truncate the SharePlex tables without purging files in the variable-data directory, you can run the **cleanup.sql** script from the **bin** sub-directory (Unix and Linux systems) or the util sub-directory (Windows systems) of the product directory.
- On Unix and Linux systems, you can purge the files in the variable-data directory, without truncating the SharePlex tables, by running the **cleanup.sql** script from the **bin** sub-directory of the SharePlex product directory.

Shell requirement

To use this utility, the Korn (ksh) shell must be installed on the system. The utility calls this shell during processing.

How ora_cleansp works

ora_cleansp works in two stages to restore replication to an initial state, as follows:

truncate internal tables: First **ora_cleansp** truncates any SharePlex internal tables that can be truncated safely. It does not remove the SharePlex Oracle user, password, or demonstration objects.

clean the variable-data directory: Next, **ora_cleansp** removes the following components from the SharePlex variable-data directory:

- The queue files in the **rim** sub-directory.
- The log files in the **log** sub-directory. The Event log retains one entry reflecting the outcome of the **ora_cleansp** procedure.
- The contents of the **statusdb** file in the data sub-directory.
- The contents of the **dump** and **state** sub-directories.
- Any transactions in the **oos** sub-directory, if the SP_OPO_SAVE_OOS_TRANSACTION parameter is enabled.

ora_cleansp preserves user-created files such as configuration files, conflict-resolution files, transformation files, hint files, the paramdb, and the oramsglist file.

ora_cleansp deactivates configurations. To start replication after running **ora_cleansp**, you must activate a configuration.

To run ora_cleansp:

IMPORTANT! **ora_cleansp** must be run on all source and target systems to restore replication to an initial state. To verify if and when **ora_cleansp** was run on a system, view the Event Log on that system.

1. Log on as a SharePlex Administrator.
2. [**Unix and Linux ONLY**] Set the SP_SYS_VARDIR environment variable to point to the SharePlex variable-data directory. If you are running multiple instances of **sp_cop** using multiple variable-data directories, set SP_SYS_VARDIR for the variable-data directory that you want to clean up.

   ksh shell:

   ```
   $export SP_SYS_VARDIR=/SharePlex_vardir_absolute_path
   ```

   csh shell:

   ```
   $setenv SP_SYS_VARDIR /SharePlex_vardir_absolute_path
   ```
3. Run **sp_ctrl** from the **bin** sub-directory of the SharePlex product directory.
4. Shut down **sp_cop**.

   ```
   sp_ctrl> shutdown
   ```
5. Verify that all SharePlex processes are stopped by using either the **ps -ef | grep sp_** command on Unix and Linux systems or **Taskmgr** from the **SpUtil** program on Windows systems.
6. Kill all orphan SharePlex processes by using either the **kill -9** command on Unix and Linux systems or the KillSharePlex.bat program on Windows systems.
7. **[Unix and Linux ONLY]** If you have not done so already, view the **oratab** file to determine the values for ORACLE_HOME and ORACLE_SID for the instance.

   ```
   $cd /etc
   $ more oratab
   ```

8. [**Unix and Linux ONLY**] Set the ORACLE_HOME and ORACLE_SID environment variables.

   ksh shell:

   $**export ORACLE_HOME=**[*value from oratab file*]
   $**export ORACLE_SID=**[*value from oratab file*]

   csh shell:

   $ setenv ORACLE_HOME [*value from oratab file*]
   $ setenv ORACLE_SID [*value from oratab file*]

9. On the command line of the operating system, change directories to the **bin** sub-directory of the SharePlex product directory

   NOTE: This must be the current working directory.

10. Run **ora_cleansp** as the SharePlex user account in the Oracle instance. *Note*: If run as a different user, **ora_cleansp** does not truncate the SharePlex tables.

    Additionally, on Unix and Linux, include the database/datasource name (Oracle_SID) as defined in the connections.yaml file.

    Unix and Linux:

    $ **./ora_cleansp** *database_name/SharePlex_username[/SharePlex_password]*

    (For an Oracle PDB): $**./ora_cleansp** *tns_alias for pdb/SharePlex_username[/SharePlex_password]*

    Windows:

    ```
    C:\users\splex\bin>ora_cleansp SharePlex_username[/SharePlex_password]
    ```

    (For an Oracle PDB): C:\users\splex\bin>ora_cleansp SharePlex_username[/SharePlex_password@tns_alias for pdb]

    IMPORTANT! On Windows, if you installed SharePlex on any port other than the default of 2100, use the -r option to specify the port number. For example, in the following command the port number is 9400:

    ```
    C:\users\splex\bin>ora_cleansp SharePlex_username[/SharePlex_password] -r9400
    ```

11. View the Event Log after you run the utility to confirm its results.

    Note: If the utility generates an error message stating that it cannot remove the .../ **save_SharePlex_version** directory, you can ignore the message and remove this directory.

pg_cleansp Utility

Description

Use the **pg_cleansp** utility to remove the current replication state on a system where SharePlex is replicating data from PostgreSQL Database as a Service to a PostgreSQL target.

CAUTION: The effects of **pg_cleansp** are not reversible.

Usage guidelines

- To use this utility, the Korn (ksh) shell must be installed on the system. The utility calls this shell during processing.
- **pg_cleansp** must be run on all PostgreSQL source and target systems in the replication configuration.
- To fully remove the replication environment, run the **pg_cleansp** utility on all PostgreSQL source systems in the replication configuration. To verify if and when **pg_cleansp** was run on a system, view the SharePlex event log on that system.

What this utility does

pg_cleansp does the following on the source system:

- Truncates all the SharePlex Client internal tables in the SharePlex schema and any other SharePlex-installed tables as applicable.
- Removes the following from the variable-data directory: the queue files, the process log files, the **statusdb** file, the contents of the **dump** and **state** directories, and all but one entry in the event log (the status entry for **pg_cleansp**)
- Drops replication slots (physical/logical)

pg_cleansp preserves the following:

- The SharePlex database, account, and password, and only cleans the data in the SharePlex tables.
- User-created files such as the **paramdb** and the target configuration settings

To run the pg_cleansp utility on PostgreSQL source and target:

1. Stop all SharePlex processes on the system.
2. Shut down **sp_cop**.
3. Run the **pg_cleansp** utility from the **bin** sub-directory of the SharePlex product directory with the following syntax:

   `<installationDirectiory\ProductDirectory>/bin` **pg_cleansp** [**port***number*] *database_name*/*user_name*

   where:

   - **port***number* is the port number of the SharePlex instance, if other than the default of 2100.
   - *database_name* is the name of the SharePlex database.
   - *user_name* is the name of the SharePlex user account.
4. Enter the password at the prompt.

   ```
   Enter the password for the SharePlex User :
   ```
5. Type **Y** to confirm that you want to run cleanup for this SharePlex instance.

   ```
   Are you sure you want to perform the clean for '/splex/vardir/var2' and port
   2200? [N] :Y
   ```

A successful setup terminates with a message similar to the following:

```
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.
```

NOTE: In SharePlex 11.1, physical or logical slots created at activation will get dropped from the database after cleanup.

Snowflake_Cleansp Utility

Description

Use the **snowflake_cleansp** utility to remove the current replication state on a system where SharePlex is replicating to a Snowflake target.

> **CAUTION:** The effects of **snowflake_cleansp** are not reversible.

Usage guidelines

- To use this utility, the Korn (ksh) shell must be installed on the system. The utility calls this shell during processing.
- **snowflake_cleansp** must be run on all Snowflake target systems in the replication configuration.
- To fully remove the replication environment, run the **snowflake_cleansp** utility on all Snowflake target systems in the replication configuration. To verify if and when **snowflake_cleansp** was run on a system, view the SharePlex event log on that system.

What this utility does

snowflake_cleansp does the following on the target system:

- Truncates all the SharePlex Client internal tables in the SharePlex schema and any other SharePlex-installed tables as applicable.
- Removes the following from the variable-data directory: the queue files, the process log files, the **statusdb** file, the contents of the **dump** and **state** directories, and all but one entry in the event log (the status entry for **snowflake_cleansp**)

snowflake_cleansp preserves the following:

- The SharePlex database, account, and password, and only cleans the data in the SharePlex internal tables.
- User-created files such as the **paramdb** and the target configuration settings

To run the snowflake_cleansp utility on Snowflake target:

1. Stop all SharePlex processes on the system.
2. Shut down **sp_cop**.
3. Run the **snowflake_cleansp** utility from the **bin** sub-directory of the SharePlex product directory with the following syntax:

   `<installationDirectiory\ProductDirectory>/bin` **snowflake_cleansp** [**port***number*] *database_name*/ *schema name* / *user_name*

   where:

   - **port***number* is the port number of the SharePlex instance, if other than the default of 2100.
   - *database_name* is the name of the SharePlex database.
   - *schema_name* is the name of the Shareplex schema.
   - *user_name* is the name of the SharePlex user account.

4. Enter the password at the prompt.

   ```
   Enter the password for the SharePlex User :
   ```

   NOTE: This option will not be displayed for the RSA based user.

5. Type **Y** to confirm that you want to run cleanup for this SharePlex instance.

   ```
   Are you sure you want to perform the clean for '/splex/vardir/var2' and port
   2200? [N] :Y
   ```

A successful setup terminates with a message similar to the following:

```
Clean port 5626

Cleaning /splex/aparopka/sf_datatypes/var subdirectories

DEMO_SCHEMA_20.SHAREPLEX_OPEN_TRANS table truncated.
```

SharePlex License Utilities

Description

Use the SharePlex license utilities to view, add, and remove license keys to hosts in the SharePlex replication environment. Each installation of SharePlex requires a valid license key.

There are three types of SharePlex license keys for each supported platforms:

- Trial license keys
- Perpetual License keys (Permanent)
- Term license keys

Following is the list of supported platforms for which licenses are available:

- Oracle
- File
- JMS
- Kafka
- SQL Server
- PostgreSQL
- MySQL
- Snowflake
- MariaDB
- Event Hubs
- All Platforms

NOTE: To install a trial version of SharePlex, users need to select the **All Platforms** option when prompted during installation of SharePlex or while running the **splex_add_key** utility.

SharePlex licensing information can be found in the Quest Software Product Guide. Please contact your account manager if you have questions.

If you do not have a valid license key, you may obtain one from Quest Technical Support or your Quest sales representative for the required platform from the above list of supported platforms. Use the appropriate procedure in this documentation to obtain the necessary information to support your license request.

License utilities

splex utility

This utility can be used on the Linux, Unix, and Windows platforms to:

- Add a license key
- View a license key
- Remove a license key

Add a license key

Use the **splex_add_key** utility to add a license key to a machine during the installation of SharePlex or afterward to convert from one type of license to another.

You can use the **splex_add_key** utility as follows:

- You can use **splex_add_key** on the primary node of a cluster to install licenses for all secondary nodes in the cluster, because they all share one variable-data directory.
- You cannot use **splex_add_key** to add licenses for non-clustered machines from one machine. It must be run on each non-clustered replication system so that the license information is stored in the variable-data directory on each system.

To run splex_add_key:

1. Log on to the system as the SharePlex Administrator.
2. Run **sp_ctrl** on the machine where you want to install a license key.
3. If SharePlex is running, shut it down.

   ```
   sp_ctrl>  shutdown
   ```

4. Run **splex_add_key** from the **install** sub-directory of the SharePlex product directory.

   ```
   $ /proddir/install/splex_add_key
   ```

5. **Choose a platform to add/update license key:**

```
SharePlex License Utility

1) Oracle

2) File

3) JMS

4) Kafka

5) SQL Server

6) Postgres

7) MySQL

8) Snowflake

9) MariaDB

10) Event Hubs

11) All Platforms

q) Quit License Utility

Enter option:
```

> **NOTE:** To install a trial version of SharePlex, users need to select the **All Platforms** option.

6. Enter the appropriate number from the above list to choose the platform.
7. Enter the key manually as received from Quest. Press **Enter** when finished entering the key.
8. Enter **q** to exit the utility.
9. Start SharePlex when you are ready for replication to resume.

View a license key

Use the splex_get_key utility to view the SharePlex license key related details. Run this utility from the **install** sub-directory of the SharePlex product directory.

$ /*proddir*/**install/splex_get_key**

The information is similar to the following example:

```
$ /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=
```

Remove a license key

Use the `splex_remove_key` utility to remove already installed SharePlex license key for a particular platform. If no license key is installed on the machine, it will display the "`No license installed`" message.

To run splex_remove_key:

1. Log on to the system as the SharePlex Administrator.
2. Run **splex_remove_key** from the **install** sub-directory of the SharePlex product directory to remove a license key. Remove key will display a list of previously added licence keys.

   `$ /`*proddir*`/`**install/splex_remove_key**
3. Choose the appropriate platform to remove a license key:

```
SharePlex License Utility

1) Oracle

2) Postgres

q) Quit License Utility

Enter option: 1
```

A successful removal of the license terminates with a message similar to the following:

```
The SharePlex for Oracle license has been successfully removed.
```

spUtil utility

This utility can be used only on the Windows platform to:

- Add a license key
- View a license key
- Remove a license key

Add a license key

Users can add a license key using the following steps.

To add a key:

1. Log on to the system as the SharePlex Administrator.
2. Right-click the **spUtils** shortcut from the desktop and select **Run as administrator**.

   The **SharePlex Utilities window** appears.

3. Navigate to the **License Key** tab.
4. Choose a platform from the **Platform** dropdown to add a license key.
5. Click **Add License**.
6. Enter the key provided by the licensing team in the **Enter Oracle <platform name> key** field and click **OK**.

A successful addition of the license key terminates with a message similar to the following:

```
The SharePlex license has been successfully added.
```

View the license details

Users can view the license details using the following steps.

To view a license details:

1. Log on to the system as the SharePlex Administrator.
2. Right-click the **spUtils** shortcut from the desktop and select **Run as administrator**.

   The **SharePlex Utilities** window appears.
3. Navigate to the **License Key** tab.
4. Choose a platform from the **Platform** dropdown to add a license key.
   - If the license for the chosen platform is not installed, the **License Status** section displays the message `No license installed`.
   - If a license is already installed, the **License Status** section will show the license details as follows. For example:

```
Platform           = All
Product Name       = SharePlex
Product Version    = 11
License Number     = 123-456-789
License Key Type   = Trial
License Expiry     = Midnight of Jan 01, 2050
```

Remove a license key

Users can add a license key using the following steps. If no license key is installed on the machine, it will display the "`No license installed`" message.

To run splex_remove_key:

1. Log on to the system as the SharePlex Administrator.
2. Right-click the **spUtils** shortcut from the desktop and select **Run as administrator**.

   The **SharePlex Utilities window** appears.
3. Navigate to the **License Key** tab.
4. To remove a license for any platform, click **Remove License**.
5. In the **Select Platform to remove license** window, select the relevant platform from the drop-down and click **OK**.

A successful removal of the license key terminates with a message similar to the following:

```
The SharePlex for Oracle <platform name> license has been successfully removed.
```

Provision

Description

Use the **provision** utility to change a host name or IP address in the SharePlex configuration.

The SharePlex processes rely on the host names or IP addresses of the source and target machines to route data properly. The **provision** utility enables you to change host names or IP addresses within an active SharePlex instance, without reactivating a new configuration.

NOTE: The provision utility does not change anything in the database. It only affects SharePlex internal objects.

Supported databases

All databases supported by SharePlex on all supported platforms

Guidelines for using provision

- If running SharePlex on an AIX machine, set EXTSHM before running **provision**.

  export EXTSHM=ON

- Run **provision** on all of the machines in the SharePlex configuration. Each machine can reference the IP addresses of all the other machines.

Run provision

1. Stop **sp_cop**. If **sp_cop** is running, **provision** will fail.

NOTE: **provision** prevents **sp_cop** from being started while it is running.

2. Using the command line of the operating system, run **provision** from the SharePlex util sub-directory of productdir with the following syntax:

   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>]

Argument	Input
-f *old_hostname* [**:***old_ipaddress*]	• **-f** is required and represents "from." • *old_hostname* is the old (current) host name. • *old_IPaddress* is the old IP address. Use if the IP address cannot be obtained from the network.
-t *new_hostname* [**:***new_ipaddress*]	• **-t** is required and represents "to." • *new_hostname* is the new host name. • *new_IPaddress* is the new IP address. Use if the IP address cannot be obtained from the network.
-p *port*	For Windows systems, specifies the port of the SharePlex instance for which provision is being run. (You can run the **"-p"port** provision only on the Windows system.)
-n	Runs **provision** without actually making any changes. Generates a report on the changes that **provision** will make. **IMPORTANT!** The **best practice** is to run **provision** with **-n** first, to make certain you agree with the potential changes, then run it without **-n** to make the changes.
-h*new_hostID*	• **-h** is required and represents changing host ID or replacing host ID • new_hostid is the new IP address
-i	**-i** is required and represents host information

Example:

```
provision -h newid -n

provision -i

provision -f oldname -t newname -n
```

3. View the event log to view every change that was made. If the provision run fails or you do not agree with the changes that were made, you can undo them by running the **undo_provision** script. See Undo changes made by provision .

Undo changes made by provision

The **provision** utility creates an **undo_provision** script that can be used to restore the host names and IP addresses to their previous state. Run the **undo_provision** script from the **util** subdirectory of the SharePlex product directory. There are no input arguments to this script.

Known issues

The following may occur but *do not* affect the integrity of the replication environment:

- The **provision** utility does not change the active configuration file. This means that the configuration file no longer represents the current state of replication after **provision** is run. If you need to run the **compare config** command, or if you decide to reactivate the configuration, update the host name or IP address in the configuration file first.
- If an Export or Import error occurred when SharePlex connected to a machine before the name or address was changed, the error status persists and cannot be cleared.
- If the new or changed machine is a source machine, **provision** generates new routing information, but the Read process may still have the old routing in its cache. When you start **sp_cop**, Read might generate a warning that the stored IP address does not match the one for the machine. You can ignore this error.
- After **provision** is run for a source host, it might not update the "hostname" column in SHAREPLEX_ACTID table with the new host name details. If that column is not correctly updated, you must update the SHAREPLEX_ACTID table manually to specify the new host name. This is only required if the name change affected a source machine.

Qview

Description

Through the **qview** utility, you can view queue names and remove old queue files. The **qview** tools described here do not deactivate the configuration.

IMPORTANT! Do not use **qview** for the first time without the assistance of Quest Technical Support. If this utility is not used properly, it can damage the replication environment and require resynchronization and reactivation.

Supported databases

All SharePlex-supported databases on all supported platforms

Run qview

Log on to the system as a SharePlex Administrator, and use the command line of the operating system to run **qview** from the **bin** sub-directory of the SharePlex product directory. The utility is an interactive command session.

To run **qview** on the Windows platform, log onto the system as a member of the Administrators group. Run qview from a command prompt or powershell which was run by right clicking on its icon and selecting run as administrator.

On Windows, run **qview** with the **-p** option to specify the port number of the SharePlex instance for which you want to view queues.

qview-p*port***list**

Overview of qview commands

The qview utility provides the following commands:

Command	Description
list	Lists all queues for all active configurations on a system.
trim	Clean up obsolete subqueue files.
fullrb	Create a full rollback message.
otrans	Scans for a specified number of messages in the Capture queue.

List queues

Use the **list** command to list all queues for all active configurations on a system.

Description

The **qview list** command lists each queue, the replication process that writes to it, and the replication process that reads it. For example, for the capture queue, it lists the Capture process and the Read process. The queues are designated as follows:

- A capture queue is designated with a +C.
- An export queue is designated with a +X.
- A post queue is designated with a +P.

Example output:

In this example, the writer to the capture queue **o.ora11+C** is the Capture process, as indicated by the **sp_ocap** in its name string. The reader is the Read process, as indicated by the **sp_ord** in its name string. The same naming logic applies to the other queues shown in the output (export queue **expdsg+X** and post queue **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
```

Syntax

list

Trim obsolete subqueues

Use the **trim** command to clean up obsolete subqueue files on the source system.

Description

The SharePlex post queue actually consists of a number of subqueues, each approximately corresponding to a user session on the source system. The Post process uses the subqueues to establish Oracle sessions for the target instance. The number of subqueues that exist at a given time on a target system reflects the peak activity on the source system since replication started.

SharePlex routinely writes replicated data from the subqueues to associated datafiles on disk as part of its checkpoint recovery system. Each subqueue can have one or more datafiles associated with it, each with a default size of 8 MB. If the entire 8 MB file size is not consumed, a datafile remains on the system even though the data was posted and read/released. Consequently, the higher the activity level on the source system, the more datafiles on disk. The size in megabytes (MB) for the post queue in a **qstatus** display is the actual disk space that the datafiles occupy.

For example, suppose there were 100 concurrent sessions on the source system, creating 100 subqueues in the post queue on the target system. And, suppose the datafiles were only partially full when the activity level dropped—half full, for example, or 4 MB of 8 MB used—and thus were not deleted. The post queue on that system would consist of 100 datafiles at 4 MB each, totalling 400 MB of disk space.

Using the **trim** command in **qview**, you can routinely eliminate obsolete subqueue files that were read-released, while preserving the ones containing data not yet committed to the target database. The **trim** command does not eliminate queue files for subqueues 0 or 1, because those are the most heavily used subqueues.

How to run this command

Run this command on the target system only.

Stop Import and Post before running **qview** to issue this command. You can leave **sp_cop** running.

> **NOTE**: If one or both of those processes is not stopped, **qview** returns this error message: `que_INUSE: Que is already open.`

You can only trim one queue at a time. If there are more than one post queue, you are prompted to select the one you want the command to affect:

```
Queue zac+P+o.ora920-o.ora920 (y/n) <n>? n
Queue elliot+P+o.ora920-o.ora920 (y/n) <n>? y
```

> **NOTE**: If you do not select a queue, **qview** returns this error message: `que_NOEXIST: Queue does not exist.`

Syntax

trim

Execute a full rollback

Use the **otrans** and **fullrb** commands to create a full rollback message.

Description

Use the **otrans** command to scan a specified number of messages in the Capture queue, starting at the read release point. The **qview** utility then prints the transaction id, the number of operations (records), the DML type operation (if there is only one) and the object id modified (if there is only one).

Use the transaction id obtained from **otrans** to execute **fullrb**. The **qview** utility opens the Capture queue, writes an out-of-band full rollback message to the Capture queue, and then writes a commit.

How to run this command

Perform the follwing steps to to run Qview:

1. Stop Capture.
2. Run **qview**.
3. Issue the **otrans** command.

   ```
   vqiew> otrans 500000
   ```

   The output is similar to the following:

   ```
   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. Issue the **fullrb** command using the transaction ID from the **otrans** output.

   ```
   vqiew> fullrb 8(7).752562-3(139)
   ```

   The output is similar to the following:

   ```
   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. Start Capture.

   ```
   sp_ctrl> start capture
   ```

Syntax

otrans *number*

where: *number* is the number of messages to scan in the queue.

fullrb *transaction_ID*

where: *transaction_ID* is the transaction ID that was returned from **otrans**.

Show_scn utility

Use the **show_scn** utility to view the correct Oracle SCN values to supply with the following commands during a Resume Replication recovery procedure:

- **activate config** *config_name* **scn=***scn_value*
- **reconcile queue** *queue_name* **for** *datasource-datadest* **scn** *scn_value*

Supported databases

Oracle source and target

Run show_scn

The **show_scn** utility is run during the Resume Replication procedure after the source, target or both have failed. To use Resume Replication and the **show_scn** utility, there must be the following in place *at the onset of replication*:

- A disaster recovery (DR) solution that provides a physically identical copy of the *production source instance* and another physical copy of the *production target instance*. Methods such as Oracle Data Guard or disk mirroring, tape backups and other methods support this requirement.
- The SP_OPO_UPDATE_SCN parameter must be set to a value of 1. This parameter directs SharePlex to keep a record of the SCNs of the transactions that it processes. When you set this parameter to 1, it also disables the Post Enhanced Performance feature.

IMPORTANT: For more information about how to use show_scn in context, see "Resume replication after failure and recovery" in the SharePlex Administration Guide.

To run show_scn:

From the command line of the target system, run the **show_scn** utility from the **bin** subdirectory of the SharePlex product directory. For *datasource*, use the ORACLE_SID of the **source** database.

$ /*productdir*/**bin**/**show_scn** *datasource*

Output

The utility provides output similar to the following:

```
$> 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 Utility

Use the **show_last_posted** utility to view the correct PostgreSQL LSN values to supply with the following commands during a Resume Replication recovery procedure:

- **activate config** *config_name* **pglsn=**lsn_value
- **reconcile queue** *queue_name* **for** *datasource-datadest* **pglsn** *lsn_value*

Supported databases

PostgreSQL source and target

Run show_last_posted

The **show_last_posted** utility is run during the Resume Replication procedure after the source, target or both have failed. To use Resume Replication and the **show_last_posted** utility, there must be the following in place *at the onset of replication*:

A disaster recovery (DR) solution that provides a physically identical copy of the *production source instance* and another physical copy of the *production target instance*. Methods such as PostgreSQL Data Guard or disk mirroring, tape backups and other methods support this requirement.

IMPORTANT: For more information about how to use `show_last_posted` in context, see "Resume replication after failure and recovery" in the SharePlex Administration Guide.

To run show_last_posted:

PRE-REQUISITE: Users need to stop the **Post** process before running the `Show_last_posted` utility.

- From the command line of the target system, run the **show_last_posted** utility from the **bin** subdirectory of the SharePlex product directory. For *datasource*, use the `r.dbname` of the **source** database.

  ```
  $ /productdir/bin/show_last_posted datasource
  ```

- You can also run the command from **sp_ctrl**:

  ```
  sp_ctrl > show last_posted
  ```

Output

The utility provides output similar to the following:

```
$> 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>
```

NOTE: On the source database, first run the **Activation** command, and then run the **Reconcile** command on the target.

Socket_test

Description

Use the **socket_test** utility to debug networking and firewall issues for the Compare/Repair feature. If the **socket_test** utility is successful, there are no network or firewall issues to prevent the source and target from communicating.

This section explains how to use this utility in an environment configured with Network Address Translation (NAT). NAT enables a local-area network (LAN) to use one set of IP addresses for private internal traffic and another set of addresses for public external traffic. NAT acts as a bridge and an interpreter between the two networks, for example a private LAN and the public Internet, or a secure network and an insecure network.

To determine whether your system is using NAT, execute **nslookup** from the source machine, and then again from the target machines. Use the same host name for all tests. If the results reflect different values, then it is likely that the systems are working in a NAT configured environment.

The **socket_test** utility is in the **util** sub-directory of the SharePlex product directory. There are two components: a server component and a client component.

Supported databases

All databases supported on UNIX and Linux

Run socket_test

Perform the follwing steps to run socket_test:

1. On the source machine, use the following syntax to execute **socket_test** with the **server** option to run the server component.

   $ *proddir*/**util**/**socket_test server**

   The utility displays platform information, the host name, the host IP address and the port number as shown in the following example. The server remains in a waiting state until the **socket_test** client component (step 2) connects to it.

2. On the target machine, use the following syntax to execute **socket_test** with the **client** option to run the client component. Provide the NAT IP address of the source machine and the port number issued by the **socket_test** server test in the previous step.

   $*proddir*/**util**/**socket_test client** *IP_address*

   If the test is successful, two things happen:

   - The utility displays the message "`SUCCESSFULLY read/write messages from server`":
   - Additionally, the **socket_test server** output on the source machine from step 1 is appended with text similar to the following :

     ```
     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
```

If the test is not successful, the utility displays error messages as in the following example, and the **socket_test** server remains in a waiting state.

```
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
```

If the socket test fails, execute the **Ctrl-C** command on the source machine to exit out of the **socket_test** server.

SP-bininfo

Description

Use the **sp-bininfo** utility to verify that a new release of SharePlex includes past one-off builds of SharePlex that you received from Support between GA releases.

The output shows the following for each one-off that is installed for your current version of SharePlex:

- SharePlex module, such as Capture (sp_ocap) or Post (sp_opst or sp_xpst)
- SharePlex version and build number
- Database and platform
- Change Request (CR) number(s). This is a unique internal change tracking number that is assigned to your case at the time your one-off request enters the development process.
- The SharePlex libraries that were updated in the one-off

Figure 1: Sample sp-bininfo output

```
(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
```

Supported databases

Not applicable

Supported platforms

Linux and Unix

Run sp-bininfo

Perform the follwing steps to run sp-bininfo:

1. Run the **sp-bininfo** utility from the **util** subdirectory of the product directory of your current SharePlex installation.

   $ **cd** *path_to_SharePlex_proddir*/**util**

   $ ./**sp-bininfo**

2. Compare the CRs of each one-off shown in the **sp-bininfo** output with the CRs in the Resolved Issues section of the Release Notes that are included with the new SharePlexversion.
3. If any CRs of your one-offs are not shown in the Resolved Issues, the new version does not include that functionality, and you should not proceed with the upgrade. Contact SharePlex support to obtain the missing fixes.

SP_wallet

Description

Use the **sp_wallet** utility to provide the Oracle Wallet password to SharePlex. SharePlex uses the wallet password to access the TDE primary Encryption Key. SharePlex uses the TDE primary Encryption Key to decrypt TDE-protected data in the redo log that must be replicated.

Grant read permission on the Oracle Wallet file to the **dba** group before using **sp_wallet**.

Supported databases

Oracle on Unix, Linux, and Windows

Run sp_wallet

To run sp_wallet and manually supply the password:

1. On the source system, start SharePlex from the SharePlex product directory. You are prompted to run **sp_wallet**.

   ```
   *** To enable TDE replication, run sp_wallet and provide the wallet
   password ***
   ```

2. Run **sp_wallet**.

   ./sp_wallet [-r *port_number*]

   IMPORTANT! On Windows, if you installed SharePlex on any port other than the default of 2100, use the -r option to specify the port number. For example, in the following command the port number is 9400:

   ./sp_wallet -r 9400

   ```
   wallet password: walletpw
   Wallet loaded into SharePlex
   ```

To run sp_wallet in auto-open mode:

If you are using an auto-open wallet, you can configure SharePlex to open the TDE wallet automatically. This eliminates the need to run **sp_wallet** manually at SharePlex startup. The syntax is:

./sp_wallet --auto-open [**-r** *port_number*]

IMPORTANT! Using the auto-open wallet feature has additional security considerations. See the Oracle documentation for more information. In addition, do not back up the SharePlex variable-data directory together with the Oracle wallet and the Oracle data files.

To cancel auto-open mode:

./sp_wallet --no-auto-open [**-r** *port_number*]

To change the TDE primary encryption key:

If you need to change the TDE primary Encryption Key while a SharePlex configuration is active, take the following steps to ensure that SharePlex continues to replicate the TDE-protected data after the changes.

1. Quiesce the source database.
2. Make sure that Capture finishes processing the remaining data in the redo log.
3. Shut down SharePlex.
4. Change the TDE primary Encryption Key.
5. Restart SharePlex.
6. Run the **sp_wallet** utility to provide SharePlex with the new TDE primary Encryption Key.

   ./sp_wallet [**-r** *port_number*]

sp_hsm

Description

Use the sp_hsm utility to provide the Oracle Wallet password to SharePlex that is stored at external storage (HSM). SharePlex uses the wallet password to access the TDE primary Encryption Key. SharePlex uses the TDE primary Encryption Key to decrypt TDE protected data in the redo log that must be replicated.

Grant read permission on the Oracle Wallet file to the **dba** group before using **sp_hsm**.

NOTES:

- The sp_hsm utility supports HSM physical devices that are compatible with the PKCS 11 library.
- The sp_hsm utility is not supported on the Windows platform.

Supported databases

Oracle

Run the sp_hsm Utility

Run the sp_hsm utility:

1. Open command prompt.
2. Enter the **Oracle SID** or **TNS alias** if it is an RAC configuration [ORCL1]: ORCL1
3. Ensure the HSM library is set up correctly:

   ```
   HSM opm_is_hsm_library_setup entered for sid o.ORCL1

   HSM opm_is_hsm_library_setup returned true
   ```

4. Enter the **Slot ID** for the HSM on which the Oracle database is configured.

   ```
   HSM Slot ID [0]: 1
   ```

5. Enter the HSM password.

After successfully adding the HSM password to Shareplex, the utility displays the following message:

```
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

Description

Use the **sp_security** utility to enable, disable or view the SSL/TLS settings for SharePlex network communication.

Enable SSL/TLS

IMPORTANT! SSL/TLS must be either enabled with a common network password or disabled on all SharePlex installations.

To enable SSL/TLS:

Run `sp_security --setup`, select the **SSL/TLS** option, and then enter a network password.

```
% 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!
```

Disable SSL/TLS

IMPORTANT! SSL/TLS must be either enabled with a common network password or disabled on all SharePlex installations.

To disable SSL/TLS:

Run “sp_security --setup” and select non-SSL/TLS connections.

```
% 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!
```

View current SSL/TLS configuration

To view the current SSL/TLS configuration:

Run “sp_security --show”.

```
% sp_security --show

Security settings:

Configuration for '/home/shareplex/var110/' and Port 210:

  Security model            : Un-encrypted
```

Trigger Scripts

Description

SharePlex provides three interactive scripts that you can run in SQL*Plus to manage triggers so that they do not interfere with replication. Review the following points before you run the scripts.

- To run the trigger scripts, you must have access to both the SYS and DBA objects. If the SharePlex database account was set up properly during installation, that account has this access.
- After applying an Oracle patch (which sometimes creates or modifies triggers), re-run **sp_add_trigger.sql** on the target system if triggers are not disabled.
- The scripts reside in the **util** sub-directory of the SharePlex product directory.
- The SharePlex trigger scripts support row-level triggers only. Statement-level triggers must be disabled/enabled for the SharePlex user separately.
- Triggers that modify data in tables NOT being replicated and do not affect tables in the replication configuration can fire on the target system without special treatment. However, the scripts operate on all objects in the Oracle or PostgreSQL instance, whether or not they are part of replication.
- SharePlex provides other interactive scripts for managing triggers.
- The scripts must be run for target objects. Do not run them on the source objects unless you are using peer-to-peer replication.
- The scripts should not be used if source and target objects are both on the same machine. In this configuration, the user can disable triggers on target objects.

Supported databases

Oracle and supported platforms

sp_add_trigger.sql

Use **sp_add_trigger.sql** when you cannot disable triggers on target objects, such as for peer-to-peer and high-availability configurations. This script changes the triggers so that they ignore the SharePlex Oracle user associated with the Post process, but fire for all other users. It inserts the following WHEN clause into each trigger in the database.

```
when user != 'SharePlex_username' begin
```

The script prompts for the SharePlex user name, and then it modifies the triggers. It does not modify triggers belonging to SYS, SYSTEM, and SCOTT.

Syntax

@*absolute_pathname*/**sp_add_trigger.sql**

sp_change_trigger.sql

Use **sp_change_trigger.sql** to replace the name of the SharePlex user in the WHEN clause that was created with **sp_add_trigger.sql.** It prompts for the current SharePlex user name and for the new user name. Change the SharePlex user in Oracle before you run this script.

Syntax

@*absolute_pathname*/**sp_change_trigger.sql**

sp_remove_trigger.sql

Use **sp_remove_trigger.sql** to remove the WHEN clause created with **sp_add_trigger.sql**. It prompts for the SharePlex user name and then automatically removes the WHEN clause from all of the triggers. Use this script if you no longer will be replicating to those tables. If you continue to use SharePlex to replicate to those tables without disabling the triggers, the triggers will fire when SharePlex posts data, and SharePlex will generate out-of-sync errors.

Syntax

@*absolute_pathname*/**sp_remove_trigger.sql**

Trigger Scripts for PostgreSQL

Description

SharePlex provides three interactive scripts, that you can run in SQL*Plus to manage triggers so that they do not interfere with replication.

Review the following points before you run the scripts:

- To run the trigger scripts, you must have access to both the SYS and DBA objects. If the SharePlex database account was set up properly during installation, that account has this access.
- After applying a PostgreSQL patch (which sometimes creates or modifies triggers), re-run **sp_pg_add_trigger.sql** on the target system if triggers are not disabled.
- The scripts reside in the **util** sub-directory of the SharePlex product directory.
- The SharePlex trigger scripts support row-level triggers only. Statement-level triggers must be disabled/enabled for the SharePlex user separately.
- Triggers that modify data in tables NOT being replicated and do not affect tables in the replication configuration can fire on the target system without special treatment. However, the scripts operate on all objects in the PostgreSQL instance, whether or not they are part of replication.
- SharePlex provides other interactive scripts for managing triggers.
- The scripts must be run for target objects. Do not run them on the source objects unless you are using peer-to-peer replication.
- The scripts should not be used if source and target objects are both on the same machine. In this configuration, the user can disable triggers on target objects.

LIMITATION: As the PostgreSQL database has a one-to-many relationship between users and schemas, i.e., one user and many schemas, these scripts will take the user as input and enable or disable all triggers under that user's database.

Supported databases

PostgreSQL on supported platforms

sp_pg_add_trigger.sql

Use **sp_pg_add_trigger.sql** when you cannot disable triggers on target objects, such as for peer-to-peer and high-availability configurations. This script changes the triggers so that they ignore the SharePlex PostgreSQL user associated with the Post process, but fire for all other users. It inserts the following WHEN clause into each trigger in the database.

```
when user != 'SharePlex_username' EXECUTE { FUNCTION | PROCEDURE } function_name ( arguments )
```

The script prompts for the SharePlex user name, and then it modifies the triggers. It does not modify triggers belonging to SYS, SYSTEM, and SCOTT.

Syntax

\i *absolute_pathname*/**sp_pg_add_trigger.sql**

sp_pg_change_trigger.sql

Use **sp_pg_change_trigger.sql** to replace the name of the SharePlex user in the WHEN clause that was created with **sp_pg_create_trigger.sql.** It prompts for the current SharePlex user name and for the new user name. Change the SharePlex user in PostgreSQL before you run this script.

Syntax

\i *absolute_pathname*/**sp_pg_change_trigger.sql**

sp_pg_remove_trigger.sql

Use **sp_pg_remove_trigger.sql** to remove the WHEN clause created with **sp_pg_create_trigger.sql**. It prompts for the SharePlex user name and then automatically removes the WHEN clause from all of the triggers. Use this script if you no longer will be replicating to those tables. If you continue to use SharePlex to replicate to those tables without disabling the triggers, the triggers will fire when SharePlex posts data, and SharePlex will generate out-of-sync errors.

Syntax

\i *absolute_pathname*/**sp_pg_remove_trigger.sql**

Database Setup Utilities

The Database Setup utilities automatically configure a source or target database to allow SharePlex connections and to establish required database components that support replication.

For detailed information on how to set up various database utilities, refer to the Database Setup Utilities section in the SharePlex Installation and Setup Guide.

6

SharePlex for Oracle Cloud Infrastructure

Contents

Database Setup for Oracle Cloud-ExaCS and DBCS

Requirements

Database Setup for Oracle Cloud-ExaCS and DBCS

Use the Database Setup utility for Oracle (**ora_setup**) to establish SharePlex as an Oracle user and create the required SharePlex database objects. This setup utility creates the following:

- A SharePlex account
- Tables and other objects for use by SharePlex and owned by the SharePlex account
- Default connection for the SharePlex user

It is recommended that you review all of the content in this topic before running the setup utility.

Supported databases

Oracle source or target on supported platforms

When to run Oracle setup

Whether or not to run this utility at the time of SharePlex installation depends on whether the database is a source, intermediary, or target database, and on how you intend to synchronize the data. To view the initial synchronization procedures, see the SharePlex Administration Guide.

System Type	When to run Oracle Setup
Source system	During installation of SharePlex
Intermediary system	An intermediary system is used in a cascading configuration, where SharePlex replicates data to a remote system (or systems) and then sends that data from the intermediary system to the final target. If you intend to configure SharePlex to post data to a database on an intermediary system, and you intend to use a hot backup to establish the data on that system and the target, do not run the Database Setup

System Type	When to run Oracle Setup
	utility on the intermediary or target systems. You will run it when you perform the initial synchronization procedure.
Target system	Depends on the method that you will use to synchronize the source and target data when you are ready to activate replication: ◦ If you intend to use transportable tablespaces or a cold copy (such as export/import, store/restore from tape, FTP), run the Database Setup utility during SharePlex installation. ◦ If you intend to use a hot backup to establish the target data, *do not* run the Database Setup utility. You will run it when you perform the initial synchronization procedure. **NOTE:** If you run the Database Setup utility before the backup and recovery, the setup gets overwritten, and you will need to re-run it again after the backup and recovery.

Supported Oracle connections

The setup utility can configure any of the following connections for the SharePlex user to use when connecting to the database.

Database type	Connection
PDB with ASM	TNS alias for the PDB and either TNS or bequeath for the ASM instance.

Required run privileges

The user who runs the Database Setup utility must have the following privileges:

Amazon RDS source or target

The user who runs the setup utility must be the primary user that was created when the Oracle RDS instance was created. You are prompted for this user during the setup.

Non-multitenant (standard) database:

The user who runs the setup utility must have DBA privileges.

Multitenant database:

The user who runs the setup utility should have SYSDBA privileges (recommended), but at minimum the user should be a DBA user with privileges for **sys.users$** and **sys.enc$**.

The minimum following grants are required for the SharePlex user:

```
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;
```

Privileges granted to SharePlex

The Database Setup utility grants to the SharePlex database user the following privileges.

Privilege granted	Description
DBA role	The Database Setup utility grants DBA role and unlimited resource privileges, tablespace privileges, and read privileges to the redo logs.
Default Oracle profile	By default this profile has the unlimited resource privileges originally assigned by Oracle.
Grants	**The following grants are issued to SharePlex:** • To access the data dictionary (outside the DBA roles) if O7_DICTIONARY_ACCESSIBILITY is set to FALSE: **grant select any dictionary to***SharePlexUser***;** • To replicate DDL: **grant select any table to***SharePlexUser***with admin option;** **grant create any view to***SharePlexUser***with admin option;**

Requirements

- Install the database client on the system where you are running Oracle Setup. Consult the Oracle documentation for the appropriate client version to use with the database.
- Run the Database Setup utility for all source and target Oracle instances in the SharePlex replication configuration.
- For a consolidated replication topography, or other topology with multiple variable-data directories, run the Database Setup utility for each variable-data directory.
- SharePlex supports local BEQUEATH connections or remote connections using a TNS alias. Be prepared to supply Oracle Setup the needed connection values for whichever connection you want to use. If using TNS, the **tnsnames.ora** file must be configured prior to running setup.
- If the Oracle database is a multitenant container database, run the Database Setup utility for each pluggable database involved in a replication scenario. A SharePlex user and schema objects must exist in each PDB.
- If you run the Database Setup utility when there is an active configuration, the DDL that the setup performs to install or update the SharePlex internal tables will be replicated to the target. To work around this issue, set the **SP_OCT_REPLICATE_ALL_DDL** parameter to **0** before running the utility, then return the parameter to its previous setting after the setup is complete. This parameter takes effect immediately.

SharePlex schema storage requirements

The Database Setup utility for Oracle installs some database objects for use by SharePlex. The storage requirements for these objects should be satisfied before running Oracle Setup. See the following table.

Storage	Description
SharePlex objects tablespace	The Database Setup utility installs some tables into a tablespace of your choosing. All but the SHAREPLEX_LOBMAP table use the default storage settings of the tablespace. The SHAREPLEX_LOBMAP table contains entries for LOBs stored out-of-row. It is created with a 1 MB INITIAL extent, 1 MB NEXT extent, and PCTINCREASE of 10. The MAXEXTENTS is 120, allowing the table to grow to 120 MB. **Preferred action**: If you enable supplemental logging for primary and unique keys, you can set the SP_OCT_ENABLE_LOBMAP parameter to 0, and nothing will be stored in the SHAREPLEX_LOBMAP table. In this case, you do not have to consider its size growth. It is recommended that you enable supplemental logging for primary and unique keys to maximize the performance of the Read process. **Alternate action**: The default storage usually is sufficient for SHAREPLEX_LOBMAP, permitting more than 4 million LOB entries. If the Oracle tables to be replicated have numerous LOB columns that are inserted or updated frequently, consider increasing the size the SharePlex tablespace accordingly. Take into account that this table shares the tablespace with other SharePlex tables. If the database uses the cost-based optimizer (CBO) and the tables that SharePlex processes include numerous LOBs, incorporate the SHAREPLEX_LOBMAP table into the analysis schedule. **NOTE:** A new installation of SharePlex does not change storage parameters from a previous installation.
SharePlex temporary tablespace	The Database Setup utility prompts for a temporary tablespace for SharePlex to use for sorts and other operations, including sorts performed by the **compare** commands. The default temporary tablespace is the one where the SharePlex objects are installed. If you plan to use the **compare** commands to compare large tables, especially those without a primary or unique key, specify a dedicated temporary tablespace for SharePlex.
SharePlex index tablespace	The Database Setup utility prompts for a tablespace to store the indexes for the SharePlex tables. The default index tablespace is the one where the SharePlex objects are installed. To minimize I/O contention, specify a different index tablespace from the one where the tables are installed. **NOTE:** If indexes from a previous version of SharePlex are installed in the SharePlex objects tablespace, you can move them to a different tablespace and then specify that tablespace when you run the setup utility.

Run database setup for Oracle cloud-ExaCS and DBCS

Perform the follwing steps to run database setup for Oracle cloud-ExaCS and DBCS:

IMPORTANT! The Oracle instance must be open before this procedure is performed.

1. (Unix and Linux only) If you are using multiple variable-data directories, export the environment variable that points to the variable-data directory for the SharePlex instance for which you are running Database Setup.

   ksh shell:

   export SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

   csh shell:

   setenv SP_SYS_VARDIR=*/full_path_of_variable-data_directory*

2. Shut down any SharePlex processes that are running, including **sp_cop**.
3. Run the Database Setup program from the command prompt of the operating system, using the full path from the SharePlex **bin** subdirectory.

   IMPORTANT! On Windows, if you installed SharePlex on any port other than the default of 2100, use the -p option to specify the port number. For example, in the following command the port number is 9400:

   ```
   C:\users\splex\bin>ora_setup -p9400
   ```

4. Specify whether the system is a source system, a target system, or both a source and target system in the SharePlex configuration.

   NOTE: This prompt only appears the first time that you run setup for this database.

5. For connection type, select **Oracle**.
6. Refer to the following table for the prompts and responses to configure SharePlex correctly for the desired connection type.

Table 3: Setup prompts and response

Prompt	Response
`Will SharePlex install be using a BEQUEATH connection? (Entering 'n' implies a SQL*net connection):`	Press **Y** to use a local BEQUEATH connection, or press **N** to use a TNS alias connection. **NOTE**: Press **N** to use a **TNS alias** if: • the database is a multitenant database • SharePlex is capturing from, or posting to, a remote database, such as one on Amazon RDS. • the database is in a cluster (such as Oracle RAC)

Prompt	Response
`Are you configuring SharePlex for an AWS RDS database?`	Press **N** if you are not configuring SharePlex for an Oracle database on RDS. Press **Y** if you are configuring SharePlex for an Amazon AWS RDS database.

One of the following prompts is shown:
 	
• If you selected BEQUEATH= Y:
 	
`Enter the Oracle SID for which SharePlex should be installed:`
 	
• If you selected BEQUEATH = N:
 	
`Enter the TNS alias for which SharePlex should be installed:`
	Non-multitenant database: Accept the default or type the correct SID or TNS alias.
	
	On RAC, the TNS alias must be a global alias.
	
	Multitenant database: Type the TNS alias of the PDB.
	
	Amazon RDS: Type the TNS alias of the RDS database.

One of the following prompts is shown:
 	
• If the database is not on RDS:
 	
`Enter a DBA user for SID:`
 	
• If the database is on RDS:
 	
`In order to create the SharePlex tables and user account, we must connect to the RDS database using the RDS primary user.`
	Non-multitenant database: Type the name of a database user that has DBA privileges.
	
	Multitenant database: Type the name of a common user who has the required privileges to install the account and objects.
	
	Amazon RDS database: Type the name of the RDS primary user.

One of the following prompts is shown:
 	
• If the database is not on RDS:
 	
`Enter password for the DBA account, which will not echo:`
 	
• If the database is on RDS:
 	
`Enter the password for the RDS primary user, which will not echo.`
	Non-multitenant database: Type the password of the DBA user.
	
	Multitenant database: Type the password of the common user. Omit the @ and the rest of the connect string. SharePlex constructs the connect string in the proper format.
	
	Amazon RDS database: Type the password of the RDS primary user.

`Current SharePlex user is user.`
 	
`Would you like to create a new SharePlex user?`
	Press **N** to update an existing SharePlex account or **Y** to create a new SharePlex account. Type the credentials when prompted.
	
	You are allowed five attempts to type a valid password for an existing SharePlex user. Passwords are obfuscated.
	
	IMPORTANT!If there is an active configuration and

Prompt	Response
	you changed the SharePlex schema, copy the SharePlex objects from the old schema to the new one to preserve the replication environment.
`Do you want to enable replication of tables with TDE?`	Press **N** since TDE is not supported for RDS.
`Enter the default tablespace for use by SharePlex:`	Press **Enter** to accept the default or type the name of a different tablespace.
`Enter the temporary tablespace for use by SharePlex:`	Press **Enter** to accept the default or type the name of a different tablespace.
`Enter the index tablespace for use by SharePlex:`	Press **Enter** to accept the default or type the name of a different tablespace.
`Will the current setup for sid: SID be used as a source (including cases as source for failover or primary-primary setups)?`	Press **Y** if this is a source system or press **N** if this is a target system. **IMPORTANT:** All systems in a primary-primary configuration (peer-to-peer) and in a high-availability configuration are considered to be source systems due to the bidirectional nature of the replication.

NOTES:

- The following prompts are only shown if the database is a source on ASM.
- If this is an Oracle target, the setup is now complete.

`ASM detected. Do you wish to connect to ASM using BEQUEATH connection?`	Press **Y** for SharePlex to use a BEQUEATH connection to connect to the ASM instance, or press **N** to use a TNS alias. **IMPORTANT!** If the database uses ASM *and* the database TNS alias is configured with a SCAN IP, then you must specify connection through an ASM TNS alias in order for SharePlex to connect to the ASM instance.
The following prompt is displayed if you did not select a BEQUEATH connection: `Do you wish to keep connecting using the same user/password?`	Press **Y** to use the same user and password as the login user, or press **N** to be prompted for a different user and password. Normally the user running SharePlex must be a member of the OSASM group. This does not apply if SP_OCT_OLOG_USE_OCI is set to the non-default

Prompt	Response
	value of 1. Also if you are using a BEQUEATH connection, the user running SharePlex must be a member of the OSDBA group.

NOTES:

- If you selected to use a BEQUEATH connection to connect to ASM, the database setup is complete. Continue to Database Setup for Oracle Cloud-ExaCS and DBCS.
- If you selected **N**, you need to supply a TNS alias, and the prompts continue.

`Enter the ASM tns alias to be used by SharePlex:`	Type the name of the TNS alias.
`Enter an ASM admin (has both sysdba and sysasm privilege) username for` *`alias`*`:`	Type the name of a user with sysasm and sysdba privileges to the ASM instance.
`Enter user password for` *`user`*`:`	Type the password of the user.
SharePlex installs internal objects that include a package to support the SDO_GEOMETRY data type of the Oracle Spatial and Graph option. If this option is not installed in the database, the following prompt is shown: `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]:`	Press **Y** to continue the database setup without support for SDO_GEOMETRY, or press **N** to terminate **ora_setup**.

7. After completion of the Oracle setup, the ora_setup.config file will be generated inside <Installation_ directory>/var/data.

Note about the tnsnames file

When you set up SharePlex for database connection through a TNS alias and ASM connection locally through a BEQUEATH connection (through OS authentication), it is important to set up the **tnsnames.ora** file correctly on each node. Assuming a SharePlex database account exists on the primary node, SharePlex will always connect to the primary ASM_SID automatically because it was provided when SharePlex was installed. However, upon failover, SharePlex must query the local **v$asm_client** view to get the correct ASM_SID of the failover instance. Therefore, ensure that the IP address of a given node is always listed first in the local **tnsnames.ora** file on that node.

7

SharePlex Variables

SharePlex uses the following environment variables, which you may need to set in certain situations. Usually you must perform additional steps before or after setting a variable, so refer to the recommended instructions before setting a SharePlex variable.

Environment Variable	Description
EDITOR	Sets the default ASCII text editor for **sp_ctrl** commands that use one, for example the **create config** command.
HOST	Sets a host name for all locally run sessions of **sp_ctrl**.
SP_COP_TPORT	Sets a non-default port number for an instance of SharePlex. The default port number is 2100. You may need to set a different port number if one of the following is true: • You are setting up additional instances of **sp_cop**. • A different port number than 2100 must be used.
SP_SYS_HOST_NAME	Sets the host name that SharePlex binds to during configuration activation. This variable is used for the following: • Sets the virtual IP address (also known as the *global cluster package name*) on a clustered system, such as Oracle RAC. This variable must be set on all cluster nodes. • In the case where there are multiple network cards on Windows, it must be set to the network card or IP address that you want SharePlex to use as the local host. Otherwise, if the server reboots after the SharePlex configuration is activated, the IP address may bind to a different network card from the one that is bound in the replication configuration. • If SP_SYS_HOST_NAME is set to an IPV6 address on the source system, SharePlex on the target system must be version 9.0 or later.
SP_SYS_VARDIR	Sets the full path to the SharePlex variable-data directory so that **sp_cop** can locate the configuration data, queues, logs and other information. If there is only one instance of **sp_cop** on the local system, this variable is set by default*. If there are multiple instances of **sp_cop** on the local system, always set this variable to point to the correct variable-data directory of an instance before setting any other SharePlex variables for that instance.
SP_SYS_SECURE_MODE	Suppresses the output of the compare and repair SQL log file for all compare and repair runs while the current instance of SharePlex is running. This variable must be set before starting SharePlex, so if the **sp_**

Environment Variable	Description
	cop process is running it must be restarted after setting this variable. When **sp_cop** is run with this environment variable, the compare and repair commands will not put data into SQL files and the Post process will not put data into the SharePlex error log.

* On Unix and Linux, the variable-data directory is set in the *proddir*/**data**/**default.yaml** file. On Windows, it is set in the Windows Registry.

To set an environment variable in Unix or Linux:

ksh shell:

export *variable_name*=*value*

csh shell:

setenv *variable_name value*

ksh shell:

export SP_SYS_VARDIR=*full_path_of_variable-data_directory*

csh shell:

setenv SP_SYS_VARDIR *full_path_of_variable-data_directory*

To set an environment variable on Windows

Set the environment variable by adding it to the SharePlex node of the Windows Registry.

IMPORTANT! Incorrect Registry settings can adversely affect the operation of SharePlex. Consult a System Administrator if you need help with this procedure.

1. Shut down the SharePlex service.
2. Open the **Run** dialog. The location varies with the Windows version.
3. In the **Run** dialog, type **regedit** to run the Registry Editor.
4. Expand the SharePlex node:
   `\HKEY_LOCAL_MACHINE\Software\Wow6432node\Quest Software\SharePlex`
5. Right click the port number of the SharePlex instance to which you want to add a variable, then select **New**, then **String Value**.
6. Under the **Name** column, right click the new variable, then select **Rename**.
7. Type the correct name.
8. Double click the new variable.
9. Under **Value Data**, enter the string for the new variable and then click **OK**.
10. Exit the Registry.

About us

We are More than Just a Name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our Brand, our Vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece — you — to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical Support Resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product