Quest

SharePlex® 11.4

Installation and Setup Guide

© 2024 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.

SharePlexInstallation and Setup Guide
Updated - 8/23/2024
Version - 11.4

Contents

About this Guide **13**

Conventions Used in this Guide **14**

Revision History **15**

Installing and Setting up SharePlex on Oracle Source Database **16**

SharePlex Pre-installation Checklist for Oracle 17

Network checklist 17

Installer checklist for Oracle 19

Unix/Linux system checklist for Oracle 21

Oracle checklist 25

Open target checklist 29

Amazon EC2 checklist 31

Download the SharePlex Installer 32

SharePlex Installers 32

Linux and Unix 32

Where to get SharePlex installers 32

Install SharePlex 33

Install SharePlex on Linux/Unix for Oracle database 33

Run the installer in interactive mode 33

Install SharePlex on Linux/Unix for open target databases 36

Run the installer in interactive mode 36

Run the installer in unattended mode 38

Set up an Oracle Environment for Replication 40

Set up Oracle logging to support SharePlex 40

Enable archive logging 40

Configure the online logs 41

Set the appropriate logging level 42

Set up Oracle database objects for replication 42

Ensure row uniqueness 42

Prevent triggers from firing on the target 46

Configure integrity constraints 46

Prevent access to target objects 47

Configure sequences 47
Set up an Oracle database to support SharePlex 49
Adjust OPEN_CURSORS to support Post cursors 49
Adjust the PROCESSES parameter to support connections 49
Adjust the log buffer size to improve posting 50
Adjust the SharePlex transaction table based on user volume 50
Control conversion of character sets 51
Set up SharePlex to support Oracle data 52
LOBs, LONGs, VARRAYs, and XML 52
Set system process priority 53
Enable Oracle direct path loads 53
Use compression 54
Configure support of Data Pump exports 54
Set up TDE Support 54
Required privilege to capture TDE-protected data 54
Configure SharePlex to capture TDE-protected data 55
Configure SharePlex to capture the TDE wallet key stored on external storage 56
Set up Replication from Oracle to a Supported Target Type 57
Set up replication from Oracle to MySQL or Aurora 58
Install SharePlex 58
Review column names 58
Configure SharePlex on the source 58
Configure SharePlex on the target 59
Set up replication from Oracle to SQL Server 60
Review column names 60
Configure SharePlex on the source 60
Configure SharePlex on the target 60
Set up replication from Oracle to PostgreSQL 61
Review column names 61
Configure SharePlex on the source 61
Configure SharePlex on the target 61
Set up replication from Oracle to SAP HANA 62
Review column names 62
Configure SharePlex on the source 62
Configure SharePlex on the target 62
Set up replication from Oracle to other open targets 63

Review column names 63
Configure SharePlex on the source 63
Configure SharePlex on the target 63
Map data types 66
Set up Replication from Oracle to JMS 67
Configure SharePlex on the source 67
Configure SharePlex on the target 67
Configure SharePlex on the JMS target 68
Enable logs for JMS replication 70
View and change JMS settings 70
Set recovery options 72
Control memory settings 73
Configure debugging 73
View the XML format 73
Set up replication from Oracle to Kafka 74
Guidelines for posting to Kafka 74
Configure SharePlex on the source 74
Configure SharePlex on the target 74
Set the Kafka record format 75
View and change Kafka settings 75
Set recovery options 79
Set up Replication from Oracle to a SQL or XML File 81
Configure SharePlex on the source 81
Configure SharePlex on the target 82
File storage and aging 83
Installation and Setup for Cloud-Hosted Databases for Oracle 84
Overview of SharePlex Setup on Cloud 84
Post to PaaS Cloud from the Source Server for Oracle 86
Source Oracle to cloud Oracle 86
Source Oracle to cloud Open Target 87
Post to PaaS Cloud from an Intermediary Server for Oracle 88
Source Oracle to cloud Oracle 88
Source Oracle to cloud Open Target 89
Installation and Setup for Remote Capture 91
Installation and Setup for HA Cluster 92
Overview of SharePlex Oracle cluster support 92

Set up SharePlex in an Oracle cluster ... 93
Basic SharePlex Demonstration for Oracle ... 97
What you will learn ... 97
Prework for the demonstrations ... 98
Tables used in the demonstrations ... 98
INSERT scripts ... 98
Start SharePlex ... 98
Create and activate a configuration for Oracle ... 99
Create a configuration file ... 99
Activate the configuration ... 100
Troubleshooting tips ... 100
Demonstration of replication ... 101
Verify replication startup ... 101
Verify replication of large data volumes ... 101
Verify queuing and continuity of replication ... 101
Verify SharePlex capture recovery ... 102
Demonstration of data compare and repair ... 104
Demonstration of named post queues ... 105
Clean up the replication environment ... 105
Create a configuration file ... 105
Activate the configuration ... 106
Generate data ... 107
View the post queues ... 107
Advanced SharePlex Demonstrations for Oracle ... 108
Prework for the demonstrations ... 109
Verify that DDL replication is enabled ... 109
Install the demonstration objects ... 109
Start SharePlex ... 111
Configure, activate, and verify replication on the Oracle source system ... 111
Create a configuration file ... 111
Activate the configuration ... 112
Generate transactions to the source table ... 113
Compare source and target tables ... 113
Repair an out-of-sync condition ... 113
Demonstration of horizontally partitioned replication ... 115
Prepare the tables ... 115

Configure the row partition ... 116
Specify the partition scheme in the configuration file ... 116
Activate the configuration ... 117
Replicate data ... 117
Demonstration of vertically partitioned replication ... 118
Prepare the tables ... 118
Configure the column partition ... 118
Activate the configuration ... 120
Replicate data ... 120
Demonstration of transformation ... 121
Prepare the objects ... 121
Configure SharePlex ... 122
Activate and start replication ... 123
View the transformed data ... 123
Demonstration of conflict resolution for Oracle ... 124
Prepare the objects ... 124
Configure SharePlex ... 125
Activate and start replication ... 126
Demonstrate trusted-source priority ... 126
Demonstrate timestamp priority ... 127
View the conflict resolution results ... 128
Demonstration of Oracle DDL replication ... 129
Verify that DDL replication is enabled ... 129
Test DDL replication ... 129
Database Setup Utilities ... 130
Database setup for HANA ... 131
Supported databases ... 131
Guidelines for use ... 131
Required privileges ... 131
Run database setup for HANA ... 131
Grant privileges to SharePlex ... 133
Database setup for MySQL ... 134
Supported databases ... 134
Guidelines for use ... 134
Required privileges ... 134
Run database setup for MySQL ... 135

Configure SharePlex to support MySQL database on Azure platform 137
Database setup for Oracle 139
Supported databases 139
When to run Oracle setup 139
Supported Oracle connections 140
Required run privileges 140
Privileges granted to SharePlex 141
Requirements 142
SharePlex schema storage requirements 143
Run database setup for Oracle 144
Run database setup for Oracle remote capture 148
Run database setup for Oracle in silent mode 151
Database setup for SQL Server 152
Supported databases 152
Guidelines for use 152
Required privileges 152
Run database setup for SQL server 153
Configure remote posting from a Linux to a SQL Server database 154
Database setup for Snowflake 155
Guidelines for use 156
Privileges granted to SharePlex 157
Run database setup for Snowflake 157
Solve Installation Problems for Oracle 161
Solve database setup problems for Oracle 161
Oracle setup issues 161
Solve database connection problems 163
Solve SharePlex startup problems for Oracle 163
How to find the ORACLE_SID and ORACLE_HOME 164

Installing and Setting up SharePlex on a PostgreSQL Database as Source and Service 165
SharePlex Pre-installation Checklist for PostgreSQL 166
Network checklist for PostgreSQL 167
Installer checklist for PostgreSQL 168
Linux system checklist for PostgreSQL 170
Amazon EC2 checklist 172
Download the SharePlex installer for PostgreSQL 173

SharePlex Installer ... 173
Linux ... 173
Install SharePlex on Linux for PostgreSQL as a Source ... 174
Set up Replication from PostgreSQL to Supported Target Types ... 175
Set up replication from PostgreSQL to PostgreSQL ... 176
Configure SharePlex on the source and target ... 176
Set up replication from PostgreSQL to Oracle ... 178
Configure SharePlex on the source and target ... 178
Set up replication from PostgreSQL to Kafka ... 179
Kafka category ... 179
Guidelines for posting to Kafka ... 179
Configure SharePlex on the source ... 179
Configure SharePlex on the target ... 180
Set the Kafka Record Format ... 180
View and change Kafka settings ... 181
Set up replication from PostgreSQL to SQL Server ... 183
Configure SharePlex on the source ... 183
Configure SharePlex on the target ... 184
Installation and Setup for Cloud-Hosted Databases for PostgreSQL ... 185
Overview of SharePlex Setup on Cloud for PostgreSQL ... 185
Post to PaaS Cloud from the Source Server for PostgreSQL ... 185
Source PostgreSQL to cloud PostgreSQL ... 186
Source PostgreSQL to cloud Open Target ... 186
Post to PaaS Cloud from an Intermediary Server for PostgreSQL ... 187
Source PostgreSQL to cloud Open Target ... 187
Installation and Setup for Remote Capture for PostgreSQL ... 188
Install SharePlex on PostgreSQL High Availability Cluster ... 189
Configure SharePlex on PostgreSQL Azure Flexible Server with High Availability Using Logical Replication ... 190
Generic SharePlex Demonstration for PostgreSQL ... 191
Prework for the demonstrations ... 192
Tables used in the demonstrations ... 192
INSERT scripts ... 192
Start SharePlex ... 192
Create and activate a configuration for PostgreSQL ... 193
Create a configuration file ... 193

Activate the configuration ... 194
Troubleshooting tips ... 194
Demonstration of replication for PostgreSQL ... 194
Demonstration of named post queues for PostgreSQL ... 195
Clean up the replication environment ... 195
Create a configuration file ... 195
Activate the configuration ... 196
Generate data ... 196
View the post queues ... 197
Advanced SharePlex Demonstrations for PostgreSQL ... 198
Demonstration of conflict resolution - PostgreSQL to PostgreSQL ... 199
Pre-work for the demonstrations ... 199
Prepare the objects ... 201
Configure SharePlex ... 201
Activate and start replication ... 202
Demonstrate trusted-source priority ... 202
Demonstrate timestamp priority ... 203
Demonstrate no conflict resolution ... 203
View the conflict resolution results ... 204
Demonstration of conflict resolution - PostgreSQL to Oracle ... 205
Pre-work for the demonstrations ... 205
Prepare the objects ... 207
Configure SharePlex ... 208
Activate and start replication ... 209
Demonstrate trusted-source priority ... 209
Demonstrate timestamp priority ... 210
Demonstrate no conflict resolution ... 210
View the conflict resolution results ... 211
Sample configuration for PostgreSQL to PostgreSQL replication ... 212
Configure, activate, and verify replication on the PostgreSQL source ... 214
Create a configuration file ... 214
Syntax configuration ... 215
Activate the configuration ... 215
Database Setup for PostgreSQL ... 216
Guidelines for use ... 217
Required Privilege ... 218

Run PG setup for PostgreSQL ... 219
Support for hyphen, dot, and plus (-, ., and +) special characters ... 222
Configure SharePlex to support PostgreSQL database on Azure platform ... 223
Database Setup for PostgreSQL Database as a Service ... 225
Guidelines for Use ... 225
Required Privilege ... 227
Run PG setup for PostgreSQL ... 228
Installation of pg_hint_plan extension ... 230
Solve Installation Problems for PostgreSQL ... 231
Solve database setup problems for PostgreSQL ... 231
PostgreSQL setup issues ... 231
Solve database connection problems for PostgreSQL ... 232

Installing SharePlex on a Docker container ... 233
Build and run SharePlex container ... 236

Assign SharePlex Users to Security Groups ... 237
About the SharePlex Security Groups ... 238
Description of the SharePlex security groups ... 238
Create and Populate SharePlex Groups on Unix and Linux ... 239

Solve Installation Problems for SharePlex ... 240
Solve Installer Problems ... 240
Solve SharePlex Startup Problems ... 241
Solve sp_ctrl Problems ... 241
Solve Host Connection Problems ... 241

Remove SharePlex from a System ... 242
Remove SharePlex from Unix/Linux ... 242

Advanced SharePlex Installer Options ... 244

Install SharePlex as Root ... 246

SharePlex Installed Items ... 247
SharePlex directories ... 247
SharePlex installed objects ... 249

About us ... 251
Contacting Quest ... 251

Technical Support Resources 251

About this Guide

This manual provides instructions for installing and setting up SharePlex on an Oracle and PostgreSQL database(s) as a source and all the supported target types.

This manual also contains some demonstration tutorials that show how certain features of SharePlex work.

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.

Revision History

Document Version	Date	Change History
2	17th May 2024	Added a query to grant privileges to existing objects after completing the Snowflake setup in the Run database setup for Snowflake section.
3	21st August 2024	Added a prerequisite in the Database Setup for PostgreSQL section to set the value of the `logical_decoding_work_mem` parameter to 4096MB.

1

Installing and Setting up SharePlex on Oracle Source Database

The following installation and setup configurations apply to the Oracle source database.

Contents

SharePlex Pre-installation Checklist for Oracle

Download the SharePlex Installer for Oracle

Install SharePlex

Set up an Oracle Environment for Replication

Set up Replication from Oracle to a Supported Target Type

Installation and Setup for Cloud-Hosted Databases for Oracle

Installation and Setup for Remote Capture

Installation and Setup for HA Cluster

Basic SharePlex Demonstration for Oracle

Advanced SharePlex Demonstrations for Oracle

Database Setup Utilities

Solve Installation Problems for Oracle

SharePlex Pre-installation Checklist for Oracle

Contents

- Network checklist
- Installer checklist for Oracle
- Unix/Linux system checklist for Oracle
- Oracle checklist
- Open target checklist
- Amazon EC2 checklist

Overview

Review and satisfy all of the requirements in this checklist before installing SharePlex.

NOTE: The requirements in this checklist apply to all source and target systems where SharePlex will be installed unless otherwise noted.

Network checklist

Requirement	Completed? (Y/N)
Add SharePlex users and groups to the nameserver. If you are installing SharePlex in a network managed by a name server such as NIS or NISPLUS, do the following before you install SharePlex: • Add SharePlex users to the nameserver. • Add the SharePlex groups to the nameserver. The SharePlex security groups **spadmin** (administrator), **spopr** (operator), and **spview** (viewer) control access to SharePlex processes. Add each SharePlex user to one of these groups on the nameserver. For more information, see Assign SharePlex Users to Security Groups on page 237. **To add the user groups:** 1. For NIS add the groups to the **group.byname** and **group.bygid** maps. For NISPLUS, add them to the **group.org_dir** table. 2. Add the SharePlex Administrator user to the **spadmin** group on the nameserver. 3. Create the **spadmin** group in the **/etc/group file** (on Unix and Linux) and then add the SharePlex Administrator user to the group.	

Requirement	Completed? (Y/N)
To add SharePlex groups to the local system after you install SharePlex, see Assign SharePlex Users to Security Groups on page 237.	

Ensure that SharePlex can resolve host names.
 	
If you find that SharePlex cannot connect to a host, try mapping the host name to an alphanumeric alias in the following locations:
 	
• Network: The NIS and DNS servers
• Unix and Linux: Local **/etc/hosts** file
 	
In these files, put each entry on an individual line.
 	
The following is an example, where **sysA** and **sysB** are aliases:
 	
`111.22.33.44  sysA.company.com  sysA  # source system`
`55.66.77.88  sysB.company.com  sysB  # target system`	

Allow access to Amazon EC2 instance For an EC2 instance, make certain that the appropriate network group is set up to allow access to EC2 instances.	
Verify the SharePlex port number. By default SharePlex uses the port number 2100 (hex equivalent is 834) for both TCP/IP and UDP. If port 2100 is available to SharePlex, no further action is needed. You will need to enter the SharePlex port number during the installation procedure, at which time you can specify a different port number if needed. **IMPORTANT!** The SharePlex port number must be the same one on all machines in the replication configuration so that they can communicate through TCP/IP connections. Make certain the SharePlex port number is open for TCP/IP on the firewall.	
Verify TCP/IP settings for Oracle: SharePlex replicates over any TCP/IP network connection. Typically, if all of the tables in a database are being replicated, SharePlex replicates about 33 percent of the redo log volume, the remainder being mostly maintenance data used by the Oracle software. The following is a formula for determining bandwidth. (*size of a redo log*) x (*number of log switches in 1 hour*) x 1/3 = *volume of data per hour* For example, if the redo logs are 20 MB in size, and they switch six times an hour, SharePlex will replicate approximately 40 MB each hour: 20 x 6 x 1/3 = 40 MB/hour	

Requirement	Completed? (Y/N)
Verify TCP protocol SharePlex has been tested on IP v6 networks, but it is impossible to test all scenarios. See the System Requirements in the SharePlex Release Notes for more informationon.	

Installer checklist for Oracle

Requirement	Completed? (Y/N)
Assign a directory to store the downloaded SharePlex installation package. This directory requires approximately the following disk space: • Unix and Linux: 200 MB It can be removed after SharePlex is installed.	
Plan the SharePlex product directory. You can create a directory for the SharePlex software files or let the SharePlex installer create it. **This directory requires approximately the following disk space:** • Unix and Linux: 600 MB **Install this directory on the following:** • Unix and Linux: a separate filesystem from the one that contains the database. Do not install SharePlex on a raw device.	
Plan the SharePlex variable-data (working) directory. This directory is installed by the SharePlex installer with a name of your choosing. It contains the working data and varies greatly in size in correlation to the volume of data being generated. Install this directory on a separate file system from the one that contains the database, but not on a raw device. **To estimate the required disk space:** 1. Estimate the longest time that a replication outage can be tolerated.	

Requirement	Completed? (Y/N)
2. Use the following formula to estimate the amount of data SharePlex would replicate during that amount of time. [*size of a redo log*] x [*number of log switches per hour*] x .333 x [*number of hours downtime*] = required disk space For example: [500 MB redo log] x [5 switches per hour] x [.333] x [8 hours] = 6.5 GB disk space To replicate data from more than one database on a system, use a variable-data directory for each one. Ideally they should be on different file systems. Do not install the variable-data directory within the SharePlex product directory. Both directories contain identically named files, and SharePlex utilities that clean up the environment (if this becomes necessary) could remove the wrong files. You can install both directories under one parent directory if desired. **NOTE**: Always monitor disk usage when there is an active SharePlex configuration, especially when there are peaks in transaction activity.	
Install in a cluster (including Oracle RAC) Most shared storage solutions can be used to house SharePlex. Such file systems include, but are not limited to: • Oracle Cluster File System (OCFS2) • Oracle Automatic Storage Management (ASM) Cluster File System (ACFS) • Oracle DataBase File System (DBFS) **NOTE**: This file system must be mounted **nointr**. Both SharePlex and Oracle report interrupt errors if **nointr** is not specified. • Most general purpose cluster file systems See Installation and Setup for HA Cluster on page 92 for SharePlex installation requirements in a cluster. Many of those steps must be performed before you install SharePlex, while others are performed after installation.	
Create the SharePlex security groups. SharePlex provides three security groups to enable access control through **sp_ctrl**. On Unix and Linux, unless you install SharePlex as a root user, the SharePlex Administrator user and the SharePlex **admin** group must exist prior to installation. For more information, see Assign SharePlex Users to Security Groups on page 237. **NOTE:** If you install as root, you are prompted by the installer to create these groups.	

Requirement	Completed? (Y/N)
Choose a DBA-privileged operating system group to own SharePlex. The SharePlex Administrator user must be in the Oracle **dba** group. For Oracle RAC and ASM 11*g*R2 and above, the user must also be in the Oracle Inventory group. For example: **$ useradd –g spadmin –G dba,oinstall**. The membership in Oracle Inventory group must be listed explicitly in the **etc/group** file.	
Get a valid SharePlex license key. You must have a valid permanent, term, or trial license key from Quest to run SharePlex. SharePlex licenses have validity and usage limits according to specific platforms. For example, you must have a PostgreSQL license to use a PostgreSQL database and a Kafka license to use the Kafka platform. Additionally, SharePlex supports multiple keys for situations where customers need two platforms on one server. For example, if a user is replicating data from an Oracle source to a Kafka target, where one SharePlex instance is serving as both the source and target, the SharePlex server would require both Oracle and Kafka licenses. **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.	

Unix/Linux system checklist for Oracle

Requirement	Completed? (Y/N)
Confirm that the platform is supported. Review the SharePlex Release Notes to make certain your operating system is supported.	
Allocate at least 4 GB of memory for SharePlex processes. Plan for per-process memory up to 256 MB. This recommendation enables the Post and Read processes to allocate larger sets of memory when necessary.	
Disable the disk cache option. (Source system) Place the redo logs, archive logs, and SharePlex files on a file system that does not have a cache option. Disk caching may interfere with the capture process. For more information, see the SharePlex Knowledge Base article 30895.	
Set the number of semaphores per process.	

Requirement	Completed? (Y/N)
Semaphores help ensure the stability of the SharePlex processes. The required SharePlex settings depend on the platform, as follows: **Oracle Solaris:** • semmni: 70 • semmns: 255 • semmnu: 255 • semmsl: 128 • semume: 255 • shmmax: 60 MB • shmmni: 100 **Red Hat Linux:** • semmni*: 70 • semmns*: 255 • semmnu: 255 • semmsl: 128 • semopm: 64 • semume: 255 • shmmax: 60 MB • shmmin: 1MB • shmmni: 100 • shmseg: 26 *These are additive. Add the database minimum values to the SharePlex minimum values to determine the correct setting. An alternative is to set the value to the number of queues you will be using plus 2. For more information about SharePlex queues, see SharePlex Administrator Guide.	
Set the ulimit (number of system file descriptors) to as close to 1024 as possible. The ulimit can be set either as a system *hard limit* or a session-based *soft limit*, as follows: • Set a hard limit: (Recommended) A root user and system restart are required to change the hard limit, but the value remains fixed at the correct level to support SharePlex. Consult your System Administrator for assistance. • Set a soft limit: A soft limit setting stays in effect only for the duration of the **sp_cop** session for which it was set, and then it reverts back to a default value that may be lower than the hard limit and too low for SharePlex.	

Requirement	Completed? (Y/N)
Set soft and hard limits for the SharePlex user Set a soft limit and a hard limit for **nproc** and **nofile** for the SharePlex O/S user in the **/etc/security/limits.conf** file. • **shareplex O/S user soft nproc 2048** • **shareplex O/S user hard nproc 16384** • **shareplex O/S user soft nofile 1024** • **shareplex O/S user hard nofile 65536** As an alternative, you can simply use the setting for the Oracle O/S user.	
Set core file parameters. • Set the system core dump block size as large as system resources can accommodate, at minimum 1.5 million blocks. The default is usually 0. Core files help Quest support representatives resolve SharePlex support cases. Higher size settings ensure that enough data is captured to be useful. • Set the core file output location to the **dump** sub-directory of the SharePlex variable-data directory. • Set the naming convention of core files to either **core** or **core.***pid*. **NOTE:** SharePlex renames all core files named **core** to **core.***pid*, except for those generated by **sp_cop**. If these requirements are not met, the SharePlex event log might report that a core file was not generated, even though a file exists.	
Install the ksh shell. Install the **ksh** shell before you install SharePlex. The SharePlex monitoring scripts and other features required this shell. A version of **ksh** called **pdksh** is included with the Red Hat Linux builds. Refer to the Red Hat Linux documentation for more information.	
Install Native POSIX Threading Library (NPTL) Quest recommends using the Native POSIX Threading Library (NPTL) on Linux. NPTL is faster and behaves more like other Unix operating systems than LinuxThreads. Although LinuxThreads can be enabled per process by using the LD_ASSUME_KERNEL environment variable, setting that variable adversely affects the performance of SharePlex. If LD_ASSUME_KERNEL is employed, use a setting of 2.4.21.	
Set the UNIX account that installs SharePlex to -rwsr-s--x using set-user-id. The value of **-rwsr-s--x** enables the Database Setup utility to connect to an Oracle database	

Requirement	Completed? (Y/N)
through SQL*Plus to install the SharePlex database account and objects during installation. The UNIX account that installs SharePlex owns this program.	
(Debian Linux) **Create a symbolic link to the /lib64/libc.so.6 library path.** On Debian Linux, the **/libc.so.6** library files are not in the **/lib64/libc.so.6** location that SharePlex expects. Before installing SharePlex on Debian Linux, create a symbolic link to **/lib64/libc.so.6**.	
Configure the oratab file. Make sure that the correct ORACLE_SID and ORACLE_HOME values are explicitly listed in the **oratab** file. SharePlex refers to this file to set its environment. On Sun machines, SharePlex only uses the **oratab** file that is in the **/var/opt/oracle** directory. If there is a copy of the **oratab** file in the **/etc** directory ensure that this file is identical to the one in the **/var/opt/oracle** directory.	

Oracle checklist

Requirement	Completed? (Y/N)
Perform any required database upgrades. Perform any required database upgrades before you install SharePlex. This ensures that SharePlex gets the most current object definitions when you run Database setup during the installation and setup steps.	
Confirm the Oracle release version and processor type. **(source and target databases)** Verify that the Oracle release version is supported by SharePlex. **NOTE:** SharePlex does not support 32-bit Oracle versions.	
Set up Oracle online and archive logging. (**source databases**) • Enable archive logging to avoid the need to resynchronize data after a log wrap. • Enable minimum supplemental logging. • Enable supplemental logging of primary and unique keys. • Configure the logs to support replication. For more information, see Set up Oracle logging to support SharePlex on page 40.	
Set up database objects to support replication. Set up Oracle objects to support replication properly, including: • Log primary and unique keys (source database only). • Handle tables that do not have keys. • Handle indexes, triggers, constraints, and sequences. For more information, see Set up Oracle database objects for replication on page 42.	
Configure database properties to support replication (**source databases**) Make the recommended database settings to support SharePlex processes: • Verify character sets are compatible with SharePlex. • Adjust log buffer size. • Adjust open cursors and processes parameters.	

Requirement	Completed? (Y/N)
• Adjust **initrans** setting of SHAREPLEX_TRANS table. For more information, see Set up an Oracle database to support SharePlex on page 49.	
Set privileges to capture TDE-protected data. (**source databases**) To decrypt TDE-protected data from the redo log, the SharePlex Administrator must open the Oracle Wallet with the wallet password. By default, only the Oracle Wallet owner-user has read and write permissions for this file. To enable SharePlex to open the wallet, you can either of the following: Grant read permission to the wallet file to the **dba** group, because the SharePlex Administrator user is a member of that group. Or... Have the owner of the wallet start SharePlex. For more information, see Set up TDE Support on page 54.	

Plan the SharePlex Oracle account.
 	
(**source and target databases**)
 	
During SharePlex installation, you will run the Database Setup utility to create a database account (user and schema) for SharePlex. The following is a list of privileges required for the database user who runs the this utility:
 	
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;`	

Plan the SharePlex objects tablespace. (**source and target databases**) The Database Setup utility installs some tables into a tablespace of your choosing. All but	

Requirement	Completed? (Y/N)
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.	
Plan the SharePlex temporary tablespace. (**source and target databases**) 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.	
Plan for theSharePlex index tablespace. (**source and target databases**) 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.	
Plan for required queue space	

Requirement	Completed? (Y/N)
SharePlex stores replicated data that is being processed in queues on the local system. There must be enough disk space for these queues to grow in the event that a SharePlex process stops (for example, a target system is down) and replicated data accumulates in the queues. **To determine the amount of queue space on each system:** 1. Use a worst-case scenario and estimate the length of downtime to be tolerated. 2. Determine how much data is replicated in that time, using the following formula as a guide. This formula multiplies the redo log value by one-third because only about one-third of the redo log is actually data that is replicated. The rest is data used by Oracle to maintain and operate the instance itself. *[size of a redo log* x *number of log switches in 1 hour]* x 1/3 x *[number of hours downtime]* = **amount of disk space needed for the queues on each system** **Example:** Suppose you expect to recover from 8 hours of downtime and your redo logs are 60 MB. They switch five times an hour. According to the formula, you need 800 MB of space on the source machine and on the target machine for the SharePlex queues. [60 MB redo log] x [5 switches/hour] x [1/3] x [8 hours] = 800 MB disk space	
Install the Oracle client. (**on-premises databases**) The Oracle client libraries are needed both for installation and setup as well as for the operation of SharePlex.	
Consider Case if replicating to an Open Target database To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.	

Open target checklist

All open targets

Requirement	Completed? (Y/N)
Perform any required database upgrades. Perform any required database upgrades before you install SharePlex. This ensures that SharePlex gets the most current object definitions when you run Database setup during the installation and setup steps.	
Confirm the database release version. Verify that the release version of the database is supported by SharePlex.	
Consider character sets 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. Install 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 apply Unicode and US7ASCII data without conversion:** If the source data is Unicode or US7ASCII and you are not replicating LOB data, no	

Requirement	Completed? (Y/N)
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.	
Consider Case To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.	
Install the appropriate ODBC driver. Verify that the appropriate ODBC driver is installed for your target, and install one if there is not one present. To test a driver with SharePlex you can use the OTS utility. You can use a standalone version of this utility before you install SharePlex, or you can run the utility from the installation directory after you install SharePlex. See the OTS documentation in the SharePlex Reference Guide for more information.	

Enable case sensitivity on key columns
Enable case-sensitivity for the data of any character-based primary key columns or columns that form a unique index. This ensures that Post compares the correct source and target key values so that it updates the correct target row and prevents unique constraint errors. Unless the key values are case sensitive, cases like the following can happen:
`Create table Sales (CustName varchar(20) primary key);`
`insert into Sales values ('abc company');`
(Succeeds)
`insert into Sales values ('ABC Company');`
(Fails with unique constraint violation error)	

Disable triggers, cascade deletes, and foreign keys on the target tables. Triggers, cascaded DELETES, and foreign keys must be disabled on Open Target tables. DML changes resulting from triggers, cascaded DELETES, and foreign keys on the source system enter the transaction log and are replicated to the target database by SharePlex. If the same mechanisms are allowed to occur on the target parent table, they initiate changes to the child tables that are duplicated through replication. These duplicate operations cause out-of-sync errors. All tables with foreign keys to one another must all be included in the replication configuration for accurate replication of the source foreign key results. All tables with referential constraints must exist in the target database. If you leave one or more out, the referential integrity could become corrupted.	

Postgres target

Requirement	Completed? (Y/N)
Add a source entry to the PostgreSQL pg_hba.conf file For a PostgreSQL target (non-cloud), make certain that the PostgreSQL **pg_hba.conf** file has an entry for the SharePlex source server IP address, which gives that server access to PostgreSQL target server and its databases. **Example entry:** `host all all sourceip md5`	

Amazon EC2 checklist

Requirement	Completed? (Y/N)
Install on multiple EBS volumes • Install the database and SharePlex on Amazon Elastic Block Storage (EBS). An EBS volume is persistent storage, whereas the default Amazon storage is non-persistent and data is lost when the instance shuts down. EBS volumes also provide better performance. Minimum size for a volume is 1GB. • To optimize disk performance, create multiple EBS volumes and combine them by using software RAID. According to benchmarks, the optimal number of EBS volumes is 8.	
Assign Elastic IP addresses Amazon Elastic IP addresses are static, which satisfy SharePlex requirements. An Elastic IP must be created and assigned to both the source and target machines that will be used with SharePlex.	

Download the SharePlex Installer

SharePlex Installers

There are different installers for SharePlex based on the type of operating system. This topic helps you understand the differences and the naming conventions used.

Linux and Unix

The SharePlex installer on Linux and Unix, as applicable, is a self-extracting installation file with the extension **.tpm**.

Oracle and Open Target:

There is a separate SharePlex installer build for each supported platform.

`SharePlex-release#-build#-platform-chipset.tpm`

For Oracle Solaris (Sun), there is a separate SharePlex installer build for each supported operating system version. If the operating system version that you have is not listed, choose the highest number that is *below* your version.

`SharePlex-release#-build#-platform-version-chipset.tpm`

The installer creates a temporary target directory, within the current directory, for extraction. This temporary target directory is removed upon installation completion. You can extract the files to a file system that is separate from the SharePlex installation location by using the -t option when running the **.tpm** file.

NOTE: The installation file (**.tpm**) file for the PostgreSQL database is available only for the Linux platform.

Where to get SharePlex installers

Download the SharePlex installation package that matches the operating system you are using.

Additionally, download any SharePlex patches, so that you can install them after you install the base software.

Perform the below steps to download the SharePlex patches:

1. Go to the Quest Software Support page: http://support.quest.com/
2. Click **Download Software.**
3. In the search box, type **SharePlex**and press **Go**.
4. Click the arrow in the **Download** column for the version you need. You can also click the file name for access to more information and to download the file(s).
5. Transfer the file to system where you are installing SharePlex.
6. You are ready to begin the installation process. Be sure to thoroughly read the version specific Release Notes *prior* to running the installer.

Install SharePlex

Install SharePlex on Linux/Unix for Oracle database

Read this before you begin:

- **These instructions assume that you understand and satisfied all requirements in the Installing and Setting up SharePlex on Oracle Source Database on page 16.**
- Perform the installation steps on all Unix and Linux machines involved in SharePlex replication. In a cluster, install on the primary node, which is the one to which the shared disk is mounted.
- These instructions assume installation as non-root. To install as root, see Install SharePlex as Root on page 246.
- The SharePlex security groups and SharePlex Administrator must exist on the system prior to installation. See Installer checklist for Oracle on page 19 for more information.
- **You can run the installer in either of the following ways:**

  Run the installer in interactive mode on page 33

  Run the installer in unattended mode on page 38

Run the installer in interactive mode

In interactive mode, you are prompted for each part of the installation information.

Perform the following steps to run the installer in an interactive mode:

1. Log in to the system as the user that will be named as the SharePlex Administrator during this installation. This user will own the installation files and binaries.
2. If **sp_cop** is running, shut it down.

   ```
   sp_ctrl>  shutdown
   ```
3. Copy the installation file to a temporary directory where you have write permissions.
4. Grant executable permissions to the file.

   #**chmod 555***installation_file*
5. Run the .tpm file. If installing SharePlex in a cluster, run the installer from the primary node (the one to which the shared disk is mounted)

   # ***./installation_file***
6. Verify that the information shown on the first screen corresponds to the Oracle version and platform you are upgrading.
7. You are prompted for the following:

Prompt	Input
Installation type	Select <**New Installation**>.
Product directory location (path)	Enter the path to the SharePlex installation directory. If the specified directory does not exist, the installer creates it. If the directory exists, it must be empty. The installer quits if the directory contains prior SharePlex installations or other files. In a cluster, install on the shared disk. For more information, see Installation and Setup for HA Cluster on page 92.
Variable data directory location	Specify an empty directory. The installer creates the specified directory if it does not exist. **IMPORTANT!** Do not install this directory into the SharePlex product directory. In a cluster, install the variable-data directory on the shared disk. For more information, see Installation and Setup for HA Cluster on page 92.
SharePlex Admin group	Enter the DBA-privileged group to which the SharePlex Administrator user belongs, which will own the SharePlex binaries. If the default group of the SharePlex Administrator is **oinstall**, select any option, and make certain this user is listed under **oinstall** in the **etc/group** file. For more information, see Installer checklist for Oracle on page 19.
TCP/IP port for SharePlex	Enter the port number to use for SharePlex TCP/IP communications.
License key (do you have?)	Press **Enter** to accept the default of **Y** (yes). If you do not have a license, enter **no**. For licensing on a cluster, see Installation and Setup for HA Cluster on page 92 At any point before you run SharePlex you can add the license key with the **splex_add_key** utility. For more information, see the SharePlex License Utilities section in the SharePlex Reference Guide.

Please specify platform for license key (select a number):	Please select the appropriate number from the following list of platforms to install the license key:
	`1) Oracle`
	`2) File`
	`3) JMS`
	`4) Kafka`
	`5) SQL Server`
	`6) Postgres`
	`7) MySQL`
	`8) Snowflake`
	`9) Event Hubs`

Prompt	Input
	`10) All Platforms` **NOTE:** To install a trial version of SharePlex, users need to select the **All Platforms** option.
License key	Enter the license key you received from Quest.

The installer displays the location of the install log file and then quits.

See Next steps.

Next steps

The following setup tasks must be finished before you start up replication:

Task	Description
Patch SharePlex	If you downloaded patches for this version of SharePlex, apply them now.
Run sp_security	Run “sp_security –setup” to specify whether or not to enable SSL/TLS for all network traffic. When enabled, SSL/TLS is used for network communications between the SharePlex instances on the source and target systems.
Run Database Setup	Run the Database Setup utility for Oracle (**ora_setup**) to establish a database account and connection information for SharePlex. For more information, see Database setup for Oracle on page 139.
(Heterogeneous replication)	Install SharePlex on the Open Target system and perform database setup operations. See: • Install SharePlex on Linux/Unix for open target databases on page 36 • Set up Replication from Oracle to a Supported Target Type on page 57
Assign users to security groups	For more information, see Assign SharePlex Users to Security Groups on page 237.
Multi-instance configurations	To install multiple instances of SharePlex on this system, such as to support consolidated replication, see the *Run SharePlex* section in SharePlex Administrator’s Guide.

Install SharePlex on Linux/Unix for open target databases

Read this before you begin:

- **These instructions assume that you understand and satisfied all preinstallation requirements that apply to your environment. See the Installing and Setting up SharePlex on Oracle Source Database on page 16.**
- For pre-installation information about installing SharePlex for target databases hosted on Amazon cloud services, see Installation and setup for cloud-hosted databases
- These instructions assume installation as non-root. To install as root, see Install SharePlex as Root on page 246.
- The SharePlex security groups and SharePlex Administrator must exist on the system prior to installation. See Installer checklist for Oracle on page 19 for more information.
- **You can run the installer in either of the following ways:**

  Run the installer in interactive mode on page 36

  Run the installer in unattended mode on page 38

Run the installer in interactive mode

In interactive mode, you are prompted for each part of the installation information.

Perform the following steps to run the installer in interactive mode:

1. Log in to the system as the user that will be named as the SharePlex Administrator during this installation. This user will own the installation files and binaries.
2. (Reinstallations) If **sp_cop** is running, shut it down.
3. Copy the installation file to a temporary directory where you have write permissions.
4. Grant executable permissions to the file.

   # **chmod 555** *installation_file*
5. Run the .tpm file. If installing SharePlex in a cluster, run the installer from the primary node (the one to which the shared disk is mounted)

   # *./installation_file*
6. You are prompted for the following:

Prompt for:	Input
Installation type	Select <**New Installation**>.
Product directory location (path)	Enter the path to the SharePlex installation directory. If the specified directory does not exist, the installer creates it. If the

Prompt for:	Input
	directory exists, it must be empty. The installer quits if the directory contains prior SharePlex installations or other files.
Variable data directory location	Specify an empty directory. The installer creates the specified directory if it does not exist. **IMPORTANT!** Do not install this directory into the SharePlex product directory.
SharePlex Admin group	Enter the DBA-privileged group to which the SharePlex Administrator user belongs, which will own the SharePlex binaries. If the default group of the SharePlex Administrator is **oinstall**, select any option, and make certain this user is listed under **oinstall** in the **etc/group** file. For more information, see Installer checklist for Oracle on page 19.
TCP/IP port for SharePlex	Enter the port number to use for SharePlex TCP/IP communications.
License key (do you have?)	Press **Enter** to accept the default of **Y** (yes). If you do not have a license, enter **no**. At any point before you run SharePlex you can add the license key with the **splex_add_key** utility. For more information, see the SharePlex License Utilities section in the SharePlex Reference Guide.

Please specify platform for license key (select a number):	Please select the appropriate number from the following list of platforms to install the license key:
	
	`1) Oracle`
	`2) File`
	`3) JMS`
	`4) Kafka`
	`5) SQL Server`
	`6) Postgres`
	`7) MySQL`
	`8) Snowflake`
	`9) Event Hubs`
	`10) All Platforms`
	
	NOTE: To install a trial version of SharePlex, users need to select the **All Platforms** option.

License key	Enter the license key you received from Quest.

The installer displays the location of the install log file and then quits.

See Next Steps.

Next Steps

The following setup tasks must be finished before you start up replication:

Task	Description
Patch SharePlex	If you downloaded patches for this version of SharePlex, apply them now.
Run sp_security	Run “sp_security –setup” to specify whether or not to enable SSL/TLS for all network traffic. When enabled, SSL/TLS is used for network communications between the SharePlex instances on the source and target systems.
Assign users to security groups	For more information, see Assign SharePlex Users to Security Groups on page 237.
Perform Database Setup and other setup tasks	**See:** Set up Replication from Oracle to a Supported Target Type on page 57
Repeat	Repeat all of the installation procedures for all Unix and Linux machines that will be involved in SharePlex replication.
Multi-instance configurations	To install multiple instances of SharePlex on this system, such as to support consolidated replication, see the SharePlex Administrator’s Guide for the correct setup.

Run the installer in unattended mode

SharePlex can be installed unattended through the use of a response file. This installation method speeds the installation of multiple SharePlex instances. The file supplies responses to the standard installer prompts, while providing on screen status information.

NOTE: When running in unattended mode, the installation process does not call the system password utility. If you create a new SharePlex user during the installation, that user will remain locked until the password is set manually.

Response files that you can edit are located in the **install** subdirectory of the SharePlex product (installation) directory:

```
/productdir/install
```

Enter responses in the File

IMPORTANT! The response file contains two sections. ***Only the top section is user configurable.*** Do not edit the bottom section. The bottom section begins with the line "`Do not change settings that appear below.`"

Edit the top section of the response file to provide the responses for the installation. ***Only edit the values to the right of the colon,*** and make certain there is a space between the colon and the response.

The following example is for non-root installation:

```
# Please modify the following settings for your particular system to

# install the SharePlex. Only values to the right of a

# colon may be editted. Incorrect changes on the left side may make
```

```
# the installer output questions and wait for answeres, thus becoming

# interactive instead of silent.

#

the SharePlex Admin group: spadmin

product directory location: /home/splex/proddir

variable data directory location: /home/splex/vardir

# not required for Open Target installations #

ORACLE_SID that corresponds to this installation: oracledb

# not required for Open Target installations #

ORACLE_HOME directory that corresponds to this ORACLE_SID:
/home/oracle/products/version

TCP/IP port number for SharePlex communications: 2100

# To specify the platform for license key [ Oracle, File, JMS, Kafka, SQL Server,
Postgres, MySQL, Event Hubs, All Platforms ]

the platform for license key: Oracle

the License key: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

# only change the current version number of SharePlex in place of 11.0.0

# do not change the value from yes to no

valid SharePlex v. 11.0.0 license: yes

# Do not change settings below.

#

Proceed with installation: yes

Proceed with upgrade: no

OK to upgrade: no
```

To run the response file:

From the command shell of the operating system, run the **.tpm** installation file with the **-r** option followed by the full path to the response file.

```
# ./installation_file -r /users/shareplex/product.rsp
```

Next steps:

For additional setup requirements, see the **Next Steps** section in one of the following depending on the platform:

Install SharePlex on Linux/Unix for Oracle database on page 33

Install SharePlex on Linux/Unix for open target databases on page 36

Set up an Oracle Environment for Replication

Overview

This chapter contains instructions for preparing an Oracle source or target database environment for replication. The tasks outlined in this chapter must be performed before you perform initial synchronization to start replication in your environment. Perform all tasks that apply to your data and to your replication objectives.

Set up Oracle logging to support SharePlex

SharePlex captures from the online and archived Oracle redo logs. SharePlex supports redo logs and datafiles that are stored on raw devices, on filesystem devices, and ASM instances.

Enable archive logging

If the Capture process stops (or is stopped by a SharePlex user) while replication is active, Capture records its location in the redo log and continues from that point when started again. However, Capture may need to read the archive logs instead of the redo logs if the following conditions occur:

- There is a long delay between when Capture stops and starts again, and the redo logs wrap during that time. When the archive logs are available, Capture reads through them to locate the records rhR it missed.
- Capture loses pace with Oracle transaction activity, and the redo logs wrap before Capture catches up to Oracle.

To support uninterrupted capture when an online log is not available, archive logging must be enabled on the source system and any other system from which SharePlex will capture data, such as an intermediate system in a cascading replication strategy. Otherwise, if the online logs wrap before Capture is finished processing them, a resynchronization of the source and target data will be necessary.

To avoid Capture issues, configure archive logging to support faster and uninterrupted replication as follows:

Requirement	Description
Time compression and removal properly	Do not compress or remove the archive logs until SharePlex is finished processing them. Otherwise, SharePlex returns a "log wrap detected" message and will stop because it cannot process the data. To determine the current log for SharePlex, issue the **show capture** command with the **detail** option in **sp_ctrl** on the source system. You can compress any logs that were generated before the current one.
Specify a non-default archive location	If you are storing the archive logs in a location other than the Oracle default, set the `SP_OCT_ARCH_LOC` parameter to the full path name of the directory where the archive logs are located. If the redo logs wrap, SharePlex searches for the archive logs in Oracle's archive log list. If SharePlex does not find the archive log there, it searches the directory or directories specified by the `SP_OCT_ARCH_LOC` parameter. To have Capture go directly to the `SP_OCT_ARCH_LOC` location and skip the read of the Oracle log list, set the `SP_OCT_CK_LOC_FIRST` to 1.
Configure Capture to wait	If using `SP_OCT_ARCH_LOC` and you are using an automated method to move the logs to that location, you can configure Capture to wait a certain amount of time for the move to be

Requirement	Description
for log-management processes	completed. This prevents Capture from stopping because a log it needs is not yet available. Capture waits, checks for the logs, stops if they are not yet available, and continues checking and stopping until the logs are available. To configure Capture to wait, set the SP_OCT_LOGWRAP_RESTART parameter to the number of seconds that you want Capture to wait. Monitor these processes regularly to prevent replication latency.
Disable archive logging on the target	You can disable archive logging on the target systems to eliminate unnecessary Oracle activity on that system, except for high availability or peer-to-peer strategies.
Do not place logs at the root ASM location	If the database uses ASM, the Oracle redo logs (online and archive) cannot be located under the ASM root directory. SharePlex will not be able to read them at that location.
ASM raw device permission	The ASM 'oracle' user must have permission to access the raw device. For example, when the raw device permission default is u:root g:disk add the 'oracle' user group 'disk'. Giving permission only to the 'grid' user is insufficient.

Configure the online logs

Ideally, the redo logs should be configured so that SharePlex can avoid reading the archive logs. In most cases, reading the online logs is faster than reading the archives. Ensure that the online redo logs are large and numerous enough to minimize processing from the archive logs. At minimum, there should be enough redo log capacity to hold several hours worth of data without wrapping.

To test an appropriate online log configuration:

In pre-production testing, you can determine if Capture is reading the archive logs by doing the following:

1. Determine the log that SharePlex is processing by querying the SHAREPLEX_ACTID table.

   SQL> **select seqno from splex.shareplex_actid**

2. Determine the log that Oracle is writing to by querying Oracle's V$LOG table.

   SQL> **select sequence# from v$log where status='CURRENT'**

3. Subtract the **seqno** value from the **sequence#** value. This reveals how many logs Capture lags behind Oracle.
4. Subtract the number of online redo logs from that value. If the number is negative, SharePlex is processing archive logs. For example, if you have 10 redo logs and SharePlex is 11 logs behind, it is processing archive logs. You can then use this result to adjust the online logging configuration.

IMPORTANT: If Capture falls behind the pace at which Oracle generates redo volume, the following may apply:

- It might be more practical to resynchronize the data instead of waiting for SharePlex to restore parity by capturing from the archive logs.
- It is possible to run out of disk space on the source system while Capture processes and queues the operations that it missed.
- There is the potential for SharePlex to lose information needed for Post to construct SQL statements, especially if the needed archive logs no longer are available. Always monitor disk space and latency while SharePlex is running.

Set the appropriate logging level

- Minimal supplemental logging must be set before you activate the SharePlex replication configuration.
- In addition to minimal supplemental logging, it is strongly recommended that you enable both primary key and unique key supplemental logging, or that you create a supplemental log group on unique columns for every table in replication. When the key column values for a row update are in the redo log, SharePlex does not have to fetch them from the database. On a busy system, this greatly improves the performance of the Read process. Some SharePlex features require primary key and unique key logging to be enabled.

  NOTE:
  ALTER TABLE DDL commands that change the rowid of a table can affect subsequent DML operations if the primary or unique keys of the tables in replication are not being logged. When the keys are not logged, SharePlex fetches their values based on the rowid. Any operation that changes the rowid, such as ALTER TABLE...MOVE, can cause the wrong key values to be used for subsequent DML operations.

- If you are using vertically partitioned replication for any table, you can use table-level logging to log only the columns that you want to replicate, plus any other columns that may be referenced by them, such as foreign keys. If you are using horizontally partitioned replication for the same table, make certain that you log the columns that you specified as the filters.

Set up Oracle database objects for replication

This topic provides information about how to configure certain characteristics of Oracle database objects that you will be replicating with SharePlex.

Ensure row uniqueness

SharePlex must have a way to ensure that the row it is changing on the target is the correct one that matches the source row. This is accomplished by ensuring a one-to-one relationship through the use of keys and indexes.

The role of keys

SharePlex works fastest when there is a primary key or a unique key on all source and target tables being replicated, especially large tables and tables containing LONG columns. When choosing a key to use, SharePlex uses the best available key column in the following order of priority.

- Primary key
- Unique key with the fewest columns, where at least one of the columns is NOT NULL
- Unique key with the fewest columns.

For best performance, it is recommended that you enable primary and unique key supplemental logging.

If a table does not have a primary or unique key, or if Oracle logs the wrong unique key for SharePlex, you can specify columns for SharePlex to use as a key when you create the configuration file. This is known as a *key definition* and is specified in the configuration file. For more information, see Define a unique key in the SharePlex Administration Guide.

An alternative to a key definition is to create or use a unique index based on one or more columns that establish uniqueness.

Ensure that the correct key is logged

When primary and unique key supplemental logging is enabled and a table does not have a primary key, Oracle has to decide what type of unique key to log. If the table has multiple unique keys, Oracle determines the best key to use and logs those column values for every UPDATE. If a table does not have any type of key, Oracle logs all columns that are not a LONG or a LOB.

SharePlex must also identify a key to use to replicate data. Like Oracle, SharePlex chooses a key in this order:

- primary key if there is one
- the best (or only) unique key if there is one
- all the columns

If a table being replicated by SharePlex does not have a primary key, but does have multiple unique keys, it is possible that the key columns that Oracle logs are not those that SharePlex requires.

Tables without a key or unique index

If SharePlex cannot detect a key or unique index on a table, it constructs a key using all of the columns except LONGs and LOBs. This key is maintained internally and is not created on the table itself.

This is not a desirable option, because the resultant WHERE clause causes Oracle to perform a full table scan on the target table to find the row, and that significantly slows replication. Additionally, row uniqueness cannot be enforced.

For example, if there is a possibility that non-LONG columns in different rows could contain identical values, but the LONG columns could have different values, the table can be out of synchronization without being detected by you or SharePlex. The following example illustrates the problem. The rows in the table are identical except for the LONG columns, and there is no primary or unique key.

COLUMN A	COLUMN B	COLUMN C (LONG)
10	20	100
10	20	200
10	20	300

Suppose a user on the source system changes Column A to 15 in the first row. When constructing the SQL statement to apply the change to the target table, SharePlex constructs a key using Columns A and B (UPDATE *tablename* SET Column A to 15 WHERE Column A = 10 and Column B = 20) to locate the row to change. There are three rows that meet this criteria, so SharePlex could post the change to the wrong row.

Keys with nulls

If a key permits NULLs, SharePlex cannot ensure the uniqueness of a row for UPDATEs and DELETES, so there is a possibility it could change the wrong row on the target system. To control how SharePlex addresses keys that permit NULLs, set the SP_SYS_IN_SYNC parameter. See the SharePlex Reference Guide for more information.

Changes to key values

SharePlex handles changes to the value of a key column without any special settings. However, if sequences are used for keys and if there is a possibility that those values will be updated, create the sequences so that the updates cannot result in a duplication of keys on the target system. Otherwise, if a new value is used to apply an operation,

and that value already exists as a key in another row in the target table, SharePlex will return unique-key constraint violations and out-of-sync errors. This type of error can happen when you update values using an "*x* +*n*" formula, where *n* is an incremental increase. It is possible for one of the "*x* +*n*" values to equal an existing value.

The following is an example, where the values in the key column are increased by 1:

```
Key_Col

1

4

5

7

SQL> update table X set a=a+1; commit
```

The new values become the following, and they are replicated to the target system:

```
Key_Col

2

5

6

8
```

SharePlex performs the update in the order that the operations enter the redo log:

`update x set a=2 where a=1;` (Succeeds)

`update x set a=5 where a=4;` (Fails because a value of `5` already exists.)

`update x set a=6 where a=5;` (Succeeds)

`update x set a=8 where a=7;` (Succeeds)

The pre-image value that Post uses for the target sequence is the same as the increased value replicated from the source. Oracle rejects the operation as a unique constraint violation. Another example would be a transaction that updates A to B and then updates B to C.

IMPORTANT! If you plan to use peer-to-peer replication, there are additional requirements for keys. For more information, see the *Configure Replication to Maintain Multiple Peer Databases* section in the SharePlex Administration Guide.

Indexes

The correct use of indexes is important in a replication environment. An index maintains the integrity of the target data.

- When replicating a source table that has a unique index, the target table also should have a unique index.
- All large tables should have a unique index on the target system. If there is no unique index, Oracle scans the entire table in order to locate the rows to be changed by Post.

- Some applications do not use primary key constraints, and therefore a unique index is not created by default. Often, however, there will be an index that was created, but not named, as a unique index (not using CREATE UNIQUE INDEX command), although it was created on one or more columns populated with unique values, such as a person's name and employee identification number. If no unique index exists for a table, it is recommended that you create one or specify a user-defined key when you create the configuration file. For more information, see the *Define a Unique Key* section in the SharePlex Administration Guide.
- Once you have identified or created a unique index, you can ensure that Oracle uses it by employing the hints feature of SharePlex. For more information, see the *Use Oracle Index Hints* section in the SharePlex Administration Guide.
- If tables have foreign keys, make sure the appropriate columns are indexed so that modifications to the foreign keys do not cause full table scans.
- Keep your indexes up-to-date, or it can slow the Post process. Rebuild the ones that are fragmented.

If there are too many indexes on a target table, Oracle must update them all as rows are added and deleted. This slows the entire system, including replication. Consider limiting the number of indexes to the ones that have the most utility.

For applications that perform mostly one type of DML, consider the following:

- INSERTs: Use just a few indexes, to limit their maintenance.
- UPDATEs: Use an index on a column that does not change after an INSERT statement.
- DELETEs: Remove as many indexes as you can.

If you run large batch jobs that do millions of SQL operations, remove unnecessary indexes before the batch job, then rebuild them at the end. This makes SharePlex run faster, and you will have more organized indexes afterward.

Bitmap indexes

For performance purposes, avoid the use of bitmap indexes while the Post process is applying the data. These indexes can adversely affect the performance of the Post process.

If you must use bitmap indexes on target tables, weigh their benefit for queries against the their impact on the transactions applied by Post:

- When Oracle adds, updates or deletes a bitmap entry, it effectively locks all the rows associated with the bitmap segment.
- A bitmap segment can contain references to hundreds of rows. As a result, changes made by different Post sessions (there is a Post session for every session on the source system) can block each other if their work updates bitmap entries in the same bitmap segment.
- To proceed, Post must detect and resolve the blocking, which delays posting significantly if the number of locks is high.
- In general, frequent inserts by multiple concurrent sessions into a table with bitmap indexes will incur lock conflicts, but random update and delete activity on such a table will not. SharePlex follows the Oracle recommendation to have a bitmap index on a more static table.

NOTE: Replicating bitmap indexes is not recommended. Every time you change a table with a bitmap index, the index is rebuilt. This associated cost (Oracle time and resources) to rebuild is added to your SQL UPDATE statement.

Prevent triggers from firing on the target

DML changes resulting from a trigger firing on the source system enter the redo log and are replicated to the target database by SharePlex. Consequently, if the same trigger fires on the target system and initiates the same DML changes (which already were made through replication) there will be out-of-sync errors.

For example, if an INSERT to TableA on the source system triggers an INSERT into TableB, SharePlex replicates both INSERTs to the target system. The Post process applies the first INSERT to TableA on the target system, triggering an INSERT into TableB. Thus, when Post attempts to post the replicated INSERT to TableB, there will be a unique-key violation. The row already exists because the trigger fired for TableA.

Triggers can be handled as follows, depending on your replication strategy:

Replication strategy	How to handle triggers on the target
High availability and Peer-to-peer	1. Enable triggers for users other than SharePlex, either in preparation for failover or because transactions are performed on multiple source systems. 2. Disable triggers for the SharePlex user by running the **sp_add_trigger.sql** script. This script puts a WHEN clause into each trigger's procedural statement that tells it to ignore operations posted by the SharePlex user.
Reporting, data sharing, other basic one-way replication	Either disable triggers entirely on the target system or run the **sp_add_trigger.sql** script to ignore operations posted by the SharePlex user.

Triggers on objects not in the replication configuration can remain active.

For important information about how to use the trigger scripts, see the SharePlex Reference Guide.

Configure integrity constraints

Integrity constraints have a effect on replication. Follow these guidelines to ensure that they are handled.

Foreign key constraints

Foreign key constraints must be disabled on the target tables. SharePlex replicates the results of the source foreign key constraints. Tables with foreign keys to one another must all be included in the replication configuration for accurate replication of the source foreign key results. All tables with referential constraints must exist in the target database. If you leave one or more out, the referential integrity could become corrupted.

NOTE: If constraints are DEFERRED on the target tables, the Post transaction can fail on the constraint validation. To work around this problem, enable the `SP_OPO_DISABLE_OBJNUM` parameter to allow the transaction to succeed. The underlying target table still remains out-of-sync until it is resynchronized.

ON DELETE CASCADE constraints

SharePlex provides a feature that allows ON DELETE CASCADE constraints to remain enabled on the target tables, but it must be enabled explicitly through a parameter setting. Post detects the

ON DELETE CASCADE dependencies and suppresses the posting of any replicated cascaded deletes to the child tables.

If you do not enable this support through SharePlex you must disable these constraints manually on the target. Otherwise SharePlex replicates both the primary delete and the cascaded deletes, which results in conflicts and errors when the deletes cascade on the target.

To enable ON DELETE CASCADE support:

1. Enable the logging of primary keys, unique index columns, and foreign key columns on the source.
2. Set the following SharePlex parameters:
    - SP_OPO_DEPENDENCY_CHECK parameter to 2
    - SP_OCT_REDUCED_KEY parameter to 0
    - SP_OPO_REDUCED_KEY parameter to 0, 1 or 2

NOTE: In peer-to-peer replication, SP_OPO_REDUCED_KEY must be set to 0.

Check constraints

Disable check constraints on the target system. They add unnecessary overhead. These checks are redundant in a well maintained, synchronized replication environment because they are satisfied on the source system. For high-availability purposes, you can build a script that re-enables the constraints as part of the failover procedure.

Prevent access to target objects

In all scenarios except peer-to-peer replication, the SharePlex database user must be the only user allowed to perform DML or DDL on the target objects. If DML or DDL changes are made to target objects by other individuals, jobs or applications, the target data may no longer reflect the state of the data on the source system. For more information, see the *Understand the Concept of Synchronization* section in the SharePlex Administration Guide.

Configure sequences

SharePlex replicates changes to Oracle sequences that are made in an ALTER SEQUENCE and DROP SEQUENCE commands and during DML transactions. It may not be necessary to replicate sequences in certain replication strategies.

- **High availability**: **Yes**

  The way that SharePlex replicates sequences allows users to begin using the failover database immediately without worrying about incrementing the sequences or reusing one.

- **Reporting, data sharing, other basic one-way replication**: **No**

  If sequences are unnecessary on the target system, do not replicate them. It can slow down replication. Even if a sequence is used to generate keys in a source table, the sequence values are part of the key columns when the replicated rows are inserted on the target system. The sequence itself does not have to be replicated.

- **Peer-to-peer**: **No**

  SharePlex does not support peer-to-peer replication of identical sequences. For more information, see the *Configure Replication to Maintain Multiple Peer Databases* section in the SharePlex Administration Guide.

To configure sequences for replication:

1. 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.
2. Use caching and set the cache to increments of at least 20. When sequences are cached, SharePlex can replicate the values as a group. When sequences are not cached, SharePlex must go to disk every time a value is obtained from the sequence, which slows replication of more critical data.
3. To ensure the uniqueness of sequences on the target system, the start value of the target sequence must be larger than the start value of the source sequence. Use the following formula to determine the target START_WITH value:

   *source_current_value***+** (*source_INCREMENT_BY_value* **x** *source_CACHE_value*) **=***target_START_WITH_value*

   IMPORTANT! (*source_INCREMENT_BY_value* **x** *source_CACHE_value*) should not exceed 2 GB, or replication of sequences will fail.
4. Specify sequences in the configuration by owner and name, as you would a table.
5. Because changes to sequences are DDL commands, the Post process pauses all posting until the sequence update is finished. For that reason, it is recommended that you process sequences through a separate post queue from the tables, especially if the sequences are not cached. For more information, see the *Configure SharePlex to Replicate Data* section in the .SharePlex Administration Guide.

SharePlex uses the ALTER SEQUENCE command to update sequences in a target database as follows:

- Change the increment value to:

  *source_INCREMENT_BY_value***x***source_CACHE_value*
- Set to NOCACHE.
- UPDATE the sequence(s).
- ALTER the sequence(s) again by setting the following values:

  *Increment_value***=***source_INCREMENT_BY_value*

  *Cache_value***=***source_CACHE_value*

SharePlex treats ALTER SEQUENCE operations like a simple SELECT (UPDATE) to a sequence because the redo log records make no distinctions between the two operations.

Set up an Oracle database to support SharePlex

Certain Oracle database settings have an affect on replication and must be set appropriately.

Adjust OPEN_CURSORS to support Post cursors

SharePlex requires the value of the Oracle OPEN_CURSORS parameter to be set correctly on the target database. To view the OPEN_CURSORS value, query the database using the following SQL statement:

select value from V$PARAMETER where name = 'open_cursors';

The Post process reserves 10 cursors for routine calls that are closed once they finish, plus a minimum of 50 cursors per transaction when the SQL Cache feature is enabled (this is the default). For more information, see Tune SQL Caching in the SharePlex Administration Guide.

If you intend to disable SQL caching, then estimate the peak number of concurrent updating transactions (sessions) that your applications generate, and follow this formula:

10 + (*peak number of concurrent transactions* x 2) = *minimum open cursors needed*

The OPEN_CURSORS value can be modified or added if absent. Consult the Oracle documentation prior to making changes to Oracle parameters.

Adjust the PROCESSES parameter to support connections

For the PROCESSES and SESSIONS parameters, 65 is the minimum value required by the SharePlex Post process so that it can open enough SQL connections to the target database to handle current transaction volume. This value is determined by the default setting of the SP_OPO_THREADS_MAX parameter, plus one for the main Post thread.

The PROCESSES parameter in the **init.ora** file must be set to accommodate connections created by SharePlex and the database users. Its value depends on whether a database is a source database, a target database, or one that serves as both a source and target database.

Database is a source only

If a database serves only as a source, the following formula takes into account logins made by the Read process.

(*peak number of source database sessions*) + (*background Oracle processes*) + (*value of SP_ORD_LDA_ARRAY_SIZE parameter* +3) = *setting for PROCESSES*

Database is a target only

The Post process creates as many connections on the target system as there are sessions on the source system in order to maintain transaction consistency.

The PROCESSES parameter on the target system must be set high enough to accommodate all of those connections, plus:

- The background Oracle processes those connections generate
- The peak number of users expected to access the target database for queries

Use the following formula as a guide:

(*peak number of source database sessions*) + (*peak number of target database sessions*) + (*background Oracle processes*) = *setting for PROCESSES*

Database is a source and target

If a database serves as both a source and a target, the following formula takes into account connections made by:

- The Read process
- The Post process
- Background Oracle processes
- User connections

(*peak number of source database sessions*) + (*peak number of target database sessions*) + (*background Oracle processes*) + (*value of SP_ORD_LDA_ARRAY_SIZE parameter* +3) = *setting for PROCESSES*

Adjust the log buffer size to improve posting

The number of database writers has an impact on replication, especially when there are numerous concurrent transactions. Whenever a transaction is committed, its buffered data is flushed to disk. If most transactions are small, but the buffer is large, this can cause slow posting. When a large transaction is committed while another, more normal-sized transaction is committed, the second COMMIT must wait while the entire buffer is flushed to disk.

Reducing the size of the buffer that is flushed to disk can speed the Post process. Try decreasing the size of the log buffer to 1024 KB, or even 512 KB if possible.

Adjust the SharePlex transaction table based on user volume

SharePlex updates the SHAREPLEX_TRANS table to maintain read consistency for the target database. You might need to adjust the **initrans** setting of this table to improve replication performance and reduce contention on that table:

- If the production database has between 500 and 1,000 concurrent users, rebuild the SHAREPLEX_TRANS table to have an initrans of 30.
- If the production database has more than 1,000 concurrent users, rebuild the SHAREPLEX_TRANS table to have an initrans value of 40.

Control conversion of character sets

This topic explains how SharePlex handles character set conversion between an Oracle source and target, and between an Oracle source and a non-Oracle target.

Replication between an Oracle source and Oracle targets

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.

Replication between an Oracle source and non-Oracle targets

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. Install 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 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.

Set up SharePlex to support Oracle data

This topic contains setup guidelines that apply to specific Oracle data types. These guidelines should be addressed before you start replication for the first time.

LOBs, LONGs, VARRAYs, and XML

- Tables that contain a LOB or LONG should have a primary key or unique key defined on them. If a table does not have a key, SharePlex builds its own key from all of the columns except LONGs or LOBs. If a LOB or LONG is the only difference between two rows that otherwise satisfy the Post WHERE clause, then Post may update the wrong row.
- Dedicate one or more named export queues to tables that contain LOBs. This automatically creates separate Export processes and named post queues with their own Post processes. By separating the processing of LOB data types from that of other data, you can improve the overall speed of replication. For more information, see the Configure Named Export Queues section in the SharePlex Administration Guide.
- To ensure that SharePlex has enough shared memory when replicating LOBs, increase the SP_QUE_POST_SHMSIZE parameter to an initial setting of 60 MB. If SharePlex generates shared memory segment errors such as "`Error: sp_cop process sp_mport/sp_opst_mt killed due to SIGSEGV`," increase the setting.

NOTE: A larger shared memory segment can result in a large amount of swap space being used on the system, so make sure enough disk space is available.

Manage SharePlex LOB storage

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.

Set system process priority

If Oracle or other processes are assigned resource priority, SharePlex can be left with a default setting and little resource allocation. Oracle increases its CPU utilization during peak processing. If SharePlex loses pace with Oracle, you can try increasing its process priority.

To set process priority on Unix:

Use the **nice** command. Consult with the System Administrator to select an appropriate value based on the requirements of all software running on the system. A root user can modify the niceness value of any process. The SharePlex Administrator user can adjust the niceness value of SharePlex.

Enable Oracle direct path loads

By default SharePlex replicates changes made to tables through a SQL*Loader direct-path load (DIRECT=TRUE keyword parameter). There can be only one load per table (PARALLEL=FALSE), although there can be simultaneous loads on different tables. The database must be in archive mode, and table logging must be enabled.

If you expect the direct-path load to be sustained for a long time on the source system, it might be more efficient to load the data to the target database directly, instead of relying on replication. A large direct-path load can cause Capture to lose pace with changes that enter the redo logs from user application activity.

After the load, you should disable check constraints. You can leave `ON DELETE CASCADE` constraints enabled.

The `SP_OCT_REPLICATE_DLOAD` parameter controls whether or not direct-path loads are replicated. To disable replication of direct-path loads, change this parameter to 0. For more information, see the SharePlex Reference Guide.

Use compression

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.

To enable compression

Set the SP_XPT_ENABLE_COMPRESSION parameter to 1.

```
sp_ctrl> set param SP_XPT_ENABLE_COMPRESSION 1
```

To activate the parameter after you set it, stop and start Export.

Configure support of Data Pump exports

When replicating Oracle Data Pump export operations, set the SP_OCT_ALLOW_DP_DDL parameter to 1, and then restart Capture.

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.

Set up TDE Support

SharePlex uses the TDE primary Encryption Key to decrypt TDE-protected data that must be replicated. SharePlex uses the Oracle wallet password to access the TDE primary Encryption Key. If the wallet password is stored on external hardware security modules (HSM), then use the sp_hsm utility.

If the wallet opens successfully, Capture connects to the decryption module and processes the data. If the wallet does not open, Capture remains in the initialization state until either the wallet is opened or the process is stopped. The initialization state that is displayed in the **show capture** command is "`Capture state: Waiting for open wallet`."

NOTE: The SharePlex **copy/append** command does not support TDE. For full information on the Oracle features that SharePlex supports, see SharePlex Release Notes.

Required privilege to capture TDE-protected data

To decrypt TDE-protected data from the redo log, the SharePlex Administrator must open Oracle Wallet using the wallet password. By default, only the Oracle Wallet owner-user has read and write permissions for this file. You can

either start as the owner of the wallet, or you can grant read permission to the file to the **dba** group, because the SharePlex Administrator user is a member of that group.

Configure SharePlex to capture TDE-protected data

To configure SharePlex to support TDE-protected data, two setup tools must be run:

- (If this was not done during installation) Run Database Setup.When prompted to enable TDE replication, type "y" and then enter the fully qualified path to the TDE wallet file, including the wallet file name, when prompted. For more Information, see For more information, see Database Setup Utilities on page 130.
- Run the **sp_wallet** utility to provide the Oracle Wallet password to SharePlex. This utility can be run in manual or auto-open mode.

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*]

   ./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*]

Configure SharePlex to capture the TDE wallet key stored on external storage

To configure SharePlex to support TDE wallet key stored on HSM (external storage) , two setup tools must be run:

- (If this was not done during installation) Run the Database Setup. When prompted to enable TDE replication, type **y**. If asked whether the TDE keys are stored in the Oracle wallet, type **n**. Next, you will be prompted to enter the fully qualified path name of the HSM PKCS 11 library, including the library name. For more Information, see For more information, see Database Setup Utilities on page 130.
- Run the **sp_hsm** utility to provide the Oracle wallet key stored on external storage (HSM). This utility can be run in manual mode.

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_hsm**.

   ```
   *** To enable TDE replication for ORACLE SID, run sp_hsm to setup the password
   for the HSM ***
   ```

2. Run **sp_hsm**.

Set up Replication from Oracle to a Supported Target Type

Contents

- Set up replication from Oracle to MySQL or Aurora
- Set up replication from Oracle to SQL Server
- Set up replication from Oracle to PostgreSQL
- Set up replication from Oracle to SAP HANA
- Set up replication from Oracle to other open targets
- Set up Replication from Oracle to JMS
- Set up replication from Oracle to Kafka
- Set up Replication from Oracle to a SQL or XML File

About these instructions

This chapter contains instructions for configuring SharePlex to replicate from Oracle to a different type of target. This is known as heterogeneous replication.

These instructions highlight specific tasks that are pertinent to the flow of data between source and target. Refer to other topics in the SharePlex documentation as needed to complete the configuration, deploy any optional features that apply, and monitor and maintain the environment.

For additional information, see:

- For the SharePlex-supported datastores, data types and operations that are supported by SharePlex, see the "System Requirements " section of SharePlex Release Notes.
- For additional configuration options, activation steps, and monitoring information, see SharePlex Administration Guide.
- For reference documentation on SharePlex commands, parameters and utilities, see SharePlex Reference Guide.

Set up replication from Oracle to MySQL or Aurora

Overview

SharePlex can post replicated Oracle data to a MySQL or Aurora target database through an Open Database Connectivity (ODBC) interface.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Install SharePlex

IMPORTANT! If replicating to MySQL or Aurora on a PaaS cloud server (no access to the operating system), see the installation instructions in Installation and setup for cloud-hosted databases.

Review column names

To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.

See SharePlex Administration Guide for more information about column mapping with SharePlex.

Configure SharePlex on the source

Set up SharePlex and the database on the Oracle source system as follows.

Make keys available to SharePlex

To replicate from an Oracle source to an Open Target target, you must make key information available to SharePlex.

Enable Oracle supplemental logging

Enable PK/UK supplemental logging in the Oracle source database. SharePlex must have the Oracle key information to build an appropriate key on the target.

Set SP_OCT_USE_SUPP_KEYS parameter

This parameter directs SharePlex to use the columns set by Oracle's supplemental logging as the key columns when a row is updated or deleted. When both supplemental logging and this parameter are set, it ensures that SharePlex can always build a key and that the SharePlex key will match the Oracle key.

See the SharePlex Reference Guide for more information about this parameter.

Configure replication

On the source, create a SharePlex configuration file that specifies capture and routing information.

Datasource:o.*SID*		
src_owner.table	*tgt_owner.table*	*host***@r.***database_name*

where:

- *SID is the Oracle SID of the source Oracle database..*
- *src_owner.table is the owner and name of the source table.*
- *tgt_owner.table is the owner and name of the target table.**
- *host* is the name of the target system.
- *database_name* is the name of the target database.

*** IMPORTANT!** If target owner or table name is defined in the database as anything other than UPPERCASE, be certain to:

- Type the name in the correct case.
- Enclose the name in quotation marks, for example "MySchema"."MyTable".
- To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.

NOTE: This is a basic one-source, one-target configuration using no additional SharePlex configuration features. See "Configure SharePlex to replicate data" in the SharePlex Administration Guide for important information about creating a configuration file and for additional setup instructions for more complex replication scenarios.

Source configuration example

The following configuration file replicates table `HR.Emp` from Oracle instance `ora112` to target table `region1.emp` in database `mydb` on target system `sysprod`. The source table is case-sensitive.

```
Datasource:o.ora112

HR."Emp"  region1.emp  sysprod@r.mydb
```

Configure SharePlex on the target

Perform the following steps to configure SharePlex on the target:

1. Make certain that the database setup meets all of the requirements in Open target checklist on page 29 .
2. Run Database Setup for MySQL (**mysql_setup**) to establish a database account and connection information for SharePlex. For more information, see Database setup for MySQL on page 134.

Set up replication from Oracle to SQL Server

Overview

SharePlex can post replicated source data to a Microsoft SQL Server target database through an Open Database Connectivity (ODBC) interface.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Review column names

To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.

See SharePlex Administration Guide for more information about column mapping with SharePlex.

Configure SharePlex on the source

You need to setup SharePlex and the database on the Oracle source system. For detailed setup steps, see Configure SharePlex on the source on page 58.

Configure SharePlex on the target

Perform the following steps to configure SharePlex on the target:

1. Make certain that the database setup meets all of the requirements in Open target checklist on page 29 .
2. Run Database Setup for SQL Server (**mss_setup**) to establish a database account and connection information for SharePlex. For more information, see Database setup for SQL Server on page 152.

Set up replication from Oracle to PostgreSQL

NOTE: These instructions apply to all implementations of PostgreSQL unless otherwise noted.

Overview

SharePlex can post replicated Oracle data to a PostgreSQL target database through an Open Database Connectivity (ODBC) interface. SharePlex supports all implementations of the PostgreSQL open-source database.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

IMPORTANT! Greenplum does not allow UPDATE operations to primary keys, but Oracle does. To replicate an UPDATE to a key from Oracle to Greenplum, Post deletes the target row and then inserts a new row based on the source row, including the new key value(s).

Review column names

To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.

See SharePlex Administration Guide for more information about column mapping with SharePlex.

Configure SharePlex on the source

You need to setup SharePlex and the database on the Oracle source system. For detailed setup steps, see Configure SharePlex on the source on page 58.

Configure SharePlex on the target

Perform the following steps to configure SharePlex on the target:

1. Make certain that the database setup meets all of the requirements in Open target checklist on page 29 .
2. Run Database Setup for PostgreSQL (**pg_setup**) to establish a database account and connection information for SharePlex. For more information, see Database Setup for PostgreSQL on page 216 .

Set up replication from Oracle to SAP HANA

Overview

SharePlex can post replicated Oracle data to a HANA target database through an Open Database Connectivity (ODBC) interface.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Review column names

To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.

See SharePlex Administration Guide for more information about column mapping with SharePlex.

Configure SharePlex on the source

You need to setup SharePlex and the database on the Oracle source system. For detailed setup steps, see Configure SharePlex on the source on page 58.

Configure SharePlex on the target

Perform the following steps to configure SharePlex on the target:

1. Make certain that the database setup meets all of the requirements in Open target checklist on page 29 .
2. Run Database Setup for HANA (**hana_setup**) to establish a database account and connection information for SharePlex. See Database setup for HANA on page 131
3. See the **target** and **typemap** commands for additional configuration options.

Set up replication from Oracle to other open targets

Overview

SharePlex can connect to a database that supports Open Database Connectivity (ODBC).

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Review column names

To support replication between a source of one database type and a target of another type, the letter case of the names of the source and target columns must be the same, for example the column names on both sides in lower case or both sides in upper case. If the case differs between the source and target column names, use the column mapping feature to map the column names in the configuration file.

See SharePlex Administration Guide for more information about column mapping with SharePlex.

Configure SharePlex on the source

You need to setup SharePlex and the database on the Oracle source system. For detailed setup steps, see Configure SharePlex on the source on page 58.

Configure SharePlex on the target

Configure SharePlex and the database on the target system as follows.

NOTE: Make certain that the database setup meets all of the requirements in Open target checklist on page 29 .

Install the ODBC driver

Make certain to download the correct ODBC driver for your database.

Set up ODBC

1. Install and test the ODBC driver that is appropriate for your target database. SharePlex provides the OTS utility for testing the ability of SharePlex to connect to and post through ODBC. For more information, see the OTS documentation in the SharePlex Reference Guide.

2. On the target system, configure ODBC connection information for use by Post to connect to the target database. You have the following options for configuring this connection information:

    - **On Unix and Linux, you can do either of the following:**

        - Configure a user or system DSN on the target system according to the instructions provided with the database. Test the DSN by using it to connect to the target database. If the connection is successful, copy the ODBC configuration files to the **odbc** subdirectory of the SharePlex variable-data directory. Set the LD_LIBRARY_PATH environment variable to the location of the database ODBC driver.

          or...

        - Set the ODBC connection information in the Post configuration. See Set connection information with the connection command.

Set connection information with the connection command

Use the **connection** command to:

- Set the Post user name and password if you created a DSN.
- Set all of theODBC connection information if a DSN does not exist.

To set connection information:

1. Create a user account for SharePlex in the target database. This account must be granted the privileges to connect, query the metadata structures of the database, create and update tables in the SharePlex database or schema, and perform full DML and supported DDL operations. Make certain that this user can connect successfully to the database through ODBC outside SharePlex.
2. Run **sp_ctrl**.
3. Execute the **connection** command with the **set** option, once for each keyword.

    connection r.*database_name* **set** *keyword=value*

Option 1: Input when a DSN exists

Keyword	Value to enter
user	The database user assigned to SharePlex
password	The password for the SharePlex user
dsn	The DSN of the database. **IMPORTANT! user**, **password**, and **dsn** are the only required keywords if a DSN exists.

Option 2: Input when a DSN does not exist (Unix and Linux)

Keyword	Value to enter
user	The database user assigned to SharePlex
password	The password for the SharePlex user
port	The database port number.
server	The name or IP address of the database server.
driver	The full path to the ODBC driver on the database server.

Option 3: Connect string when a DSN does not exist (Unix and Linux)

Keyword	Value to enter
user	The database user assigned to SharePlex
password	The password for the SharePlex user
connect_string	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.

Connection command 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

Use **connection** with the **reset** option to remove SharePlex connection settings.

To remove a specific connection value:

connection r.*database_name* **reset** *keyword*

To remove all connection values:

connection r.*database_name* **reset**

Examples:

connection r.mydb reset port

connection r.mydb reset

View connection values

Use **connection** with the **show** option to view SharePlex connection settings.

To view connection values for a database:

connection r.*database_name* **show**

To view connection settings for all local databases:

connection show all

Map data types

To support the accurate posting of replicated Oracle ALTER TABLE commands to the correct data types in an Open Target target, SharePlex maps Oracle data types to default data types for the target database. To view and change this mapping, use the **typemap** command.

> **NOTE:** For DML operations, SharePlex queries the target database to determine the appropriate data type mapping for replicated data.

For more information about the **typemap** command, see SharePlex Reference Guide.

Set up Replication from Oracle to JMS

Overview

The SharePlex Post process can connect and write to a JMS (Java Messaging Service) queue or topic. The data is written as XML records that include the data definitions, the operation type, and the changed column values. This data is written as a sequential series of operations as they occurred on the source, which can then be posted in sequential order to a target database or consumed by an external process or program.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Configure SharePlex on the source

You need to setup SharePlex and the database on the Oracle source system. For detailed setup steps, see Configure SharePlex on the source on page 58.

Configure SharePlex on the target

These instructions establish the Java provider on the target and configure the SharePlex Post process to connect to a JMS provider.

Prerequisites

Java Runtime Environment 1.8 or later. JRE 1.8 (or later) is required and installed along with SharePlex except on the following operating systems:

- CentOS Linux 7.x and 8.x
- Oracle Linux 7.x and 8.x
- RHEL Linux 7.x and 8.x

Prior to installing SharePlex on one of these systems, ensure JRE 1.8 (or later) is installed. Then after installing SharePlex, modify the **bridge** configuration file to point to the JRE binary. See the "Custom JRE" steps in each of the following procedures.

Using a JRE version later than 1.8. If you are using JMS Provider libraries compiled with a version of JRE later than 1.8, use that later version of JRE and modify the **bridge** (Linux/Unix) configuration file to point to that JRE binary. See the "Custom JRE" steps in each of the following procedures.

Configure SharePlex on the JMS target

These instructions configure the SharePlex Post process to post to a JMS queue or topic.

NOTES:

- The JMS Provider does not necessarily need to be installed on the same machine where SharePlex is running, but you will need to copy the Provider's JMS libraries to a location in the SharePlex installation directory according to these instructions.
- Ensure that you are using a compatible version of client libraries for your JMS Provider.
- SharePlex is distributed with JRE 1.8. If the JMS libraries for the JMS Provider you are using are compiled with a JRE version later than 1.8, modify the bridge (or bridge.ini) configuration file as described here. Otherwise, SharePlex returns an error similar to one of the following.

```
INFO 2016-05-10 01:43:42.000860 [x.jms+jsd-olinux-01]
com.quest.shareplex.openbridge.stomp.internal.Connector - Connector
x.jms+jsd-olinux-01 at tcp://localhost:61613 stopped.
```

```
ERROR 2016-05-10 01:43:42.000861 [x.jms+jsd-olinux-01]
com.quest.shareplex.openbridge.stomp.internal.Connector -
org/apache/activemq/jndi/ActiveMQInitialContextFactory : Unsupported
major.minor version 51.0
```

```
java.lang.UnsupportedClassVersionError:
org/apache/activemq/jndi/ActiveMQInitialContextFactory : Unsupported
major.minor version 51.0
```

The following directory naming conventions are used in the examples in this procedure:

Name	Directory
SharePlex	The SharePlex installation directory

To configure post to the JMS target:

1. Create a directory for your Provider under *SharePlex*/**lib/providers**. For example:

   ```
   shareplex/lib/providers/provider_name
   ```

   If your Provider is Active MQ, the path may look similar to the following:

   ```
   shareplex/lib/providers/amq
   ```

2. Copy the client library files you require to the *provider_name* directory.

   For example, if your Provider is Active MQ, you might copy the following JAR files to the amq directory.

   ```
   shareplex/lib/providers/amq/activemq-all.jar
   ```

   ```
   shareplex/lib/providers/amq/slf4j.jar
   ```

3. (Custom JRE) If you are using a JRE version later than 1.8, set the JAVA_HOME and JRE_HOME variable values to the path to your JRE home directory. In addition, add the JAVA_HOME path to the PATH variable.

4. (Custom JRE) If you are using a JRE other than the one installed with SharePlex which is 1.8.0_312 (e.g., SharePlex did not install JRE on your operating system or you must use a JRE version later than 1.8), modify the **bridge.ini** or **bridge** file to point to the JRE binary you intend to use.

   Use the following methods:

   - **On Linux or Unix:**

     a. The **bridge** file is located in the following directory:

        `SharePlex/.app-modules/openbridge`

     b. If a JRE was installed during the SharePlex installation, it is located in the following directory:

        `SharePlex/.app-modules/openbridge/jre`

     c. In the **bridge** file, locate the _jvm option (for example< _jvm = "jre/bin/java") and replace the current path with the full path to the JRE you intend to use. You must specify the absolute path to your JRE, not the relative path.

5. Start **sp_cop**. (Do not activate the configuration yet.)
6. Run **sp_ctrl**.
7. Issue the **target** command to configure Post to post to the JMS queue or topic. The values you set using the target command tell SharePlex how to instantiate on the target. In the following examples, user-defined values are shown in italics. Replace these with values specific to your JMS Provider and your provider-specific JMS libraries.

   Issue the following commands, as necessary. For detailed descriptions of the properties and values, see View and change JMS settings. For more information about the **target** command, see the *SharePlex Reference Guide*.

   If posting to a JMS queue:

   `sp_ctrl>` **target x.jms set jms factory_class**=*factory_class*

   `sp_ctrl>` **target x.jms set jms provider_url**=*url*

   `sp_ctrl>` **target x.jms set jms lib_location**=*path*

   For example, if using Active MQ you might set the following:

   `sp_ctrl>` **target x.jms set jms lib_location=shareplex/lib/providers/amq**

   `sp_ctrl>` **target x.jms set jms queuename**=*JMS_topic_queuename*

   If posting to a JMS topic:

   `sp_ctrl>` **target x.jms set jms factory_class**=*factory_class*

   `sp_ctrl>` **target x.jms set jms provider_url**=*url*

   `sp_ctrl>` **target x.jms set jms lib_location**=*path*

   For example, if using Active MQ you might set the following:

   `sp_ctrl>` **target x.jms set jms lib_location=shareplex/lib/providers/amq**

   `sp_ctrl>` **target x.jms set jms destination=topic**

   `sp_ctrl>` **target x.jms set jms queuename**=*JMS_topic_queuename*

Enable logs for JMS replication

We need to implement the logging mechanism with a different approach than the one we used earlier for legacy framework. In the current version, we have removed the `-OB_debugToLog, -OB_debugToConsole` parameters. Now, users get more control to generate and retain the log details by resetting properties inside the `log4j2.properties` file.

To enable the logs for JMS, follow the below steps:

1. Navigate to `<Prod/Opt_Directory>/.app-modules/openbridge/configuration/` location and update the required parameters in the `log4j2.properties` file.
2. Restart SharePlex.
3. To enable full debug, set `rootLogger.level` to **All**. There are other possible values for `rootLogger.level` which are mentioned in the form of comments in the `log4j2.properties` file.
4. To enable console logging, set `appender.console.filter.threshold.level` to **All**. There are other possible values for `appender.console.filter.threshold.level`which are mentioned in the form of comments in the `log4j2.properties` file.

NOTE: You can change other parameters as per standard log4j2.properties syntax. Refer to the below link for information.
https://logging.apache.org/log4j/2.x/manual/configuration.html#Properties

View and change JMS settings

To view current property settings for output to JMS, use the following command:

target x.jms show

To change a property setting, use the following command.

target x.jms [**queue** *queuename*] **set jms** *property*=*value*

where:

- queue queuename is the name of a Post queue. Use this option if there are multiple Post processes.
- *property* and *value* are shown in the following table.

Table 1: JMS target properties

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	None

Property	Input Value	Default
	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.	
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: • 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.	**no**

Property	Input Value	Default
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

Set recovery options

If the JMS process aborts suddenly, or if the machine that it is running on aborts, row changes may be written twice to the JMS target. The consumer must manage this by detecting and discarding duplicates.

Every record of every row-change operation in a transaction has the same transaction ID and is also marked with a sequence ID. These attributes are **id** and **msgIdx**, respectively, under the **txn** element in the XML output (see View the XML format).

The transaction ID is the SCN at the time the transaction was committed and the sequence ID is the index of the row change in the transaction. These two values are guaranteed to be the same if they are re-written to the JMS queue in a recovery situation.

If desired, you can configure the target to include additional metadata with every row-change record by using the following command:

target *target* [**queue** *queuename*] **set metadata** *property*[, *property*]

Table 2: Optional JMS metadata properties

Property	Description
time	The time the operation was applied on the source.
userid	The ID of the database user that performed the operation.

Property	Description
trans	The ID of the transaction that included the operation.
size	The number of operations in the transaction.

Example

target x.jms set metadata time, userid, trans, size

To reset the metadata:

target x.jms [**queue** *queuename*] **reset metadata**

To view the metadata:

target x.jms [**queue** *queuename*] **show metadata**

Control memory settings

If you configured Post to process large amounts of transactions to JMS through multiple named Post queues, you may need to allocate more memory to the Post JMS bridge. The default settings are 256MB and 512MB.

To increase the JMS bridge memory:

Open the openbridge.ini file and set the following parameters in the format shown:

```
-Xms=512m

-Xmx=1024m

-XX:PermSize=512m

-XX:MaxPermSize=1024m
```

Configure debugging

You can configure Post to log whether each write to a queue or topic succeeded or failed. This is controlled with the debugging setting.

To enable debugging:

Open the **openbridge.ini** file and set the **-OB_debugToLog** parameter to true.

```
-OB_debugToLog=true
```

View the XML format

To view the XML formatting of the records that SharePlex sends to JMS, see the **format** category of the **target** command documentation in the SharePlex Reference Guide.

Set up replication from Oracle to Kafka

PRE-REQUISITE: When replicating data to the Kafka (Apache and Confluent) target, target SharePlex should be installed only on the Linux platform.

Overview

The SharePlex Post process can connect and write to a Kafka broker. The data can be written in JSON or XML output as a sequential series of operations as they occurred on the source, which can then be consumed by a Kafka consumer.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Guidelines for posting to Kafka

- A SharePlex Post process acts as a Kafka producer. A SharePlex Post process can write to one or more topics that have one or more partitions.
- The SharePlex Post process does not create a topic itself, but you can configure the Kafka broker to auto-create topics.

Configure SharePlex on the source

You need to setup SharePlex and the database on the Oracle source system. For detailed setup steps, see Configure SharePlex on the source on page 58.

Configure SharePlex on the target

These instructions configure the SharePlex Post process to connect to Kafka. You must have a running Kafka broker.

To configure post to Kafka:

1. Create a Kafka topic.
2. Start **sp_cop**. (Do not activate the configuration yet.)
3. Run **sp_ctrl**.
4. Issue the **target** command to configure posting to a Kafka broker and topic. The following are example commands.

   ```
   sp_ctrl> target x.kafka set kafka broker=host1:9092,host2:9092,host3:9092
   sp_ctrl> target x.kafka set kafka topic=shareplex
   ```

   See View and change Kafka settings for command explanations and options.

NOTE: Specify more than one broker so that SharePlex will attempt to connect to the other brokers in the list if any one of them is down.

Set the Kafka record format

SharePlex can output to either XML or JSON format as input to Kafka. XML is the default. To set the input format and specify format options, use one of the following **target** commands:

target x.kafka set format record=json

or:

target x.kafka set format record=xml

To view samples of these formats, see the **format** category of the **target** command documentation in the SharePlex Reference Guide.

NOTE: When replicating data from Oracle to Kafka in JSON format, SharePlex does not support the varray data type or the varray type inside the **SDO_GEOMETRY** data type.

View and change Kafka settings

To view current property settings for output to Kafka, use the following target command:

target x.kafka show

To change a property setting, use the following command.

target x.kafka [**queue** *queuename*] **set kafka** *property*=*value*

where:

- queue *queuename* is the name of a Post queue. Use this option if there are multiple Post processes.
- *property* and *value* are shown in the following table.

Table 3: Kafka target properties

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

Property	Input Value	Default
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. • 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.	**0**

Property	Input Value	Default

	• 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.
	• 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.	

Property	Input Value	Default

	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.
	
	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	**shareplex**

Property	Input Value	Default
	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**	

* 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.

Set recovery options

If the Kafka process aborts suddenly, or if the machine that it is running on aborts, row changes may be written twice to the Kafka topic. The consumer must manage this by detecting and discarding duplicates.

Every record of every row-change operation in a transaction has the same transaction ID and is also marked with a sequence ID. These attributes are **id** and **msgIdx**, respectively, under the **txn** element in the XML output (see Set up replication from Oracle to Kafka).

The transaction ID is the SCN at the time the transaction was committed, and the sequence ID is the index of the row change in the transaction. These two values are guaranteed to be the same if they are re-written to the Kafka topic in a recovery situation.

If desired, you can configure Post to include additional metadata with every row-change record by using the following command:

target x.kafka [**queue** *queuename*] **set metadata** *property*[, *property*]

Table 4: Optional metadata properties

Property	Description
time	The time the operation was applied on the source.
userid	The ID of the database user that performed the operation.
trans	The ID of the transaction that included the operation.
size	The number of operations in the transaction.

Example:

target x.kafka set metadata time, userid, trans, size

To reset the metadata:

```
target  x.kafka  [queue queuename] reset metadata
```

To view the metadata:

target x.kafka [**queue** *queuename*] **show metadata**

Set up Replication from Oracle to a SQL or XML File

Overview

SharePlex can post replicated Oracle data to a file formatted as SQL or XML. This data is written as a sequential series of operations as they occurred on the source, which can then be posted in sequential order to a target database or consumed by an external process or program.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Configure SharePlex on the source

On the source, create a SharePlex configuration file that specifies capture and routing information. The structure that is required in a configuration file varies, depending on your replication strategy, but this shows you the required syntax for routing data to a SQL or XML file.

Datasource:o.*SID*		
src_owner.table	**!file**[:*tgt_owner.table*]	*host*

where:

- *SID* is the Oracle SID of the source Oracle database.
- *src_owner.table* is the owner and name of the source table.
- **!file** is a required keyword that directs Post to write to a file.
- *tgt_owner.table* is optional and specifies the owner and name of the target table. Use if either component is different from that of the source table.
- *host* is the name of the target system.

NOTE: For more information, see *Configure SharePlex to Replicate Data* in the SharePlex Administration Guide.

Source configuration example:

The following example replicates the `parts` table in schema `PROD` from Oracle instance `ora112` to a file on target system `sysprod`.

```
Datasource:o.ora112

PROD.parts   !file   sysprod
```

Configure SharePlex on the target

By default, SharePlex formats data to a file in **XML** format, and there is no target setup required unless you want to change properties of the output file (see Set up Replication from Oracle to a SQL or XML File.) To output in SQL format, use the **target** command to specify the SQL output as follows.

To output data in SQL format:

1. Start **sp_cop**.
2. Start **sp_ctrl**.
3. Issue the following required **target** commands to output the records in SQL.

   NOTE: Use all lower-case characters.

   target x.file [**queue** *queuename*] **set format record=sql**

   target x.file [**queue***queuename*] **set sql legacy=yes**

   where: **queue** *queuename* constrains the action of the command to the SharePlex Post process that is associated with the specified queue.

   See Set up Replication from Oracle to a SQL or XML File for descriptions of these settings and other optional properties that you can set.

To view samples of the SQL and XML formats, see the **target** command documentation in the SharePlex Reference Guide.

View and change target settings

To view current property settings for output to a file, use the following command:

target x.file show

To change a setting, use the following target command.

target x.file [**queue** *queuename*] **set** [*category*] *property*=*value*

For more information, see the *target* command in the SharePlex Reference Guide.

File storage and aging

Post writes to a series of files. The active working file is prepended with the label of **current_** and is stored in the **opx/current** subdirectory of the variable-data directory.

Output Format	Name of Current File
SQL	`current_legacy.sql`
XML	`current_prodsys.XML`

IMPORTANT: Do not open or edit the **current_** file.

Post uses the **max_records**, **max_size** and **max_time** parameters to determine the point at which to start a new active file. When this switch occurs, Post moves the processed data to a sequenced file in the **opx** subdirectory of the variable-data directory. The file names include the name of the post queue, the time and date, and an incrementing ID.

SQL files:

```
/installed/vardir> ls -1 opx

0000000000_20140305130858_legacy.sql

0000000001_20140305131130_legacy.sql

0000000002_20140305131212_legacy.sql

0000000003_20140305133835_legacy.sql

0000000004_20140305134028_legacy.sql
```

XML files:

```
/installed/vardir> ls -1 opx

0000000000_20140305130858_prodsys.XML

0000000001_20140305131130_prodsys.XML

0000000002_20140305131212_prodsys.XML

0000000003_20140305133835_prodsys.XML

0000000004_20140305134028_prodsys.XML
```

To force a file switch:

The **current** file cannot be viewed or consumed without stopping Post. To access the data in the current file, you can use the **target** command with the **switch** option to move the data to a sequenced file, from which it can then be consumed or viewed. After issuing this command, the switch occurs after Post processes a new record.

target x.file [**queue** *queuename*] **switch**

Installation and Setup for Cloud-Hosted Databases for Oracle

Contents

- Overview of SharePlex Setup on Cloud
- Post to PaaS Cloud from the Source Server for Oracle
  - Source Oracle to cloud Oracle
  - Source Oracle to cloud Open Target
- Post to PaaS Cloud from an Intermediary Server for Oracle
  - Source Oracle to cloud Oracle
  - Source Oracle to cloud Open Target

Overview of SharePlex Setup on Cloud

SharePlex supports databases installed as services of Amazon Web Services (AWS) and Microsoft Azure. To view the cloud databases that SharePlex supports, refer to the **Supported Cloud Platforms** section in the SharePlex Release Notes document for the respective databases.

There are some differences in the way that SharePlex installs in an IaaS cloud environment, a PaaS cloud environment, and a SaaS cloud environment. These differences are only in the installation and configuration of SharePlex. Once installed and configured, SharePlex operates in the cloud the same way that it operates in on-premise installations.

Installation in an IaaS (accessible) environment

If your cloud database service is a true IaaS virtual computing environment, you can install and run a custom application environment, access the operating system, and manage access permissions and storage. In this environment, SharePlex is installed directly on the cloud server just as you would install it locally, without any special setup requirements.

In this environment, the following applies:

- SharePlex can capture from an Oracle source database in an IaaS cloud.
- SharePlex can Post to any supported target database in an IaaS cloud.
- You can proceed to the standard installation instructions in this manual.

  Install SharePlex on Linux/Unix for Oracle Database

  Install SharePlex on Linux/Unix for Open Target Databases

Installation in a PaaS (non-accessible) environment

If your cloud database is installed in a true PaaS environment, you do not have access to the underlying operating system, and you must install SharePlex on a server that is external to the cloud deployment. You then configure SharePlex to interact with the target database through a remote connection.

SharePlex can capture data from supported sources and post it to databases in PaaS environment using remote capture and remote post capabilities.

You can install SharePlex for a PaaS source and target in one of the following ways:

- With remote post, you can use your on-premise production source server to run all of the SharePlex replication components. In this setup, both source and target replication processes (and their queues) are installed on one server. The SharePlex Post process connects through a remote connection to the target cloud database.

  For more information, see Post to PaaS cloud from the source system.

  > **NOTE:** In a high-volume transactional environment, the buildup of data in the post queues and the presence of multiple Post processes may generate unacceptable overhead for a production system. In that case, you should use an intermediary server.

- You can use an on-premise intermediary server to run the Import and Post components (and the post queues). Post connects to the cloud target through a remote connection. This method removes most of the replication overhead from the source server. For more information, see Post to PaaS cloud from an intermediary server.

Installation in a SaaS (non-accessible) environment

If your cloud database is hosted in a true SaaS environment, you do not have access to the underlying operating system, and you must install SharePlex on a server that is external to the cloud deployment. You then configure SharePlex to interact with the source and target databases through a remote connection.

You can install SharePlex for a SaaS target in one of the following ways:

- With remote capture, you can utilize an on-premise or cloud VM intermediary server to install and configure SharePlex for running the Capture and Export processes. Capture establishes a remote connection to the source SaaS database, while Export communicates with Import and Post on the target system.
- With remote Capture and remote Post, you can employ an on-premise or cloud VM intermediary server to install and configure SharePlex. Both Capture and Post processes will run on the same server. Capture establishes a remote connection to the source SaaS database, and Post establishes a remote connection to the target SaaS database.

  Install SharePlex on Linux/Unix for PostgreSQL Database as a Service.

Post to PaaS Cloud from the Source Server for Oracle

These instructions help you set up SharePlex to replicate to a target cloud database directly from the source server. All of the SharePlex processes will run on this server. In a high-volume transactional environment, this may generate unacceptable overhead on the production system. In that case, you should use an intermediary server.

See also:

Post to PaaS cloud from an intermediary server

Overview of SharePlex setup on cloud databases

Source Oracle to cloud Oracle

All steps are performed on the source server:

1. Complete the SharePlex pre-installation checklist.
2. Install SharePlex. See Install SharePlex on Linux/Unix for Oracle Database.
3. Run **ora_setup** to establish connection information for the source Oracle database. See Database Setup for Oracle.
4. Add an entry in the local **tnsnames.ora** file that points to the target cloud database. This enables you to query the target database from the source server.
5. Run **ora_setup** to establish connection information for the target cloud database. Run the utility as the master user that was created when the service was created. This user has the required DBA privileges.

   IMPORTANT:

   a. When asked whether this is a bequeath connection, enter N. This directs SharePlex to use TNS to connect to the target cloud database.

   b. When prompted to specify the tns_alias, specify the one that you created for the target cloud database.

   c. When asked if the current SID will be used as a source, enter N.
6. Specify the following in the routing map of the SharePlex configuration file:

   - The name of the source host to specify the target system.
   - The ORACLE_SID of the target cloud database.

   In the following example, source1 is the source system and orcldb1 is the tns_alias of the target cloud database.

```
datasource:o.orcldbprim
#source tables          target tables                    routing map
splex.demo_src          splex.demo_dest                   source1@r.orcldb1
```

For more information, see Configure SharePlex to replicate data in the SharePlex Administration Guide.

Source Oracle to cloud Open Target

All the below steps are performed on the source server:

1. Complete the SharePlex preinstallation checklist.
2. Install SharePlex. See:
   - Install SharePlex on Linux/Unix for Oracle Database.
3. Run the appropriate database setup utility for the source database. See:
   - Database Setup for Oracle.

   IMPORTANT:

   a. (Oracle setup only) When asked whether this is a bequeath connection, enter Y to use bequeath, unless this system is RAC.

   b. Reply Y when asked if the database or SID will be used as a source.
4. Install the appropriate ODBC client of the target cloud database.
5. Run the appropriate database setup utility for the target cloud database. See Database Setup Utilities.
6. Specify the following in the routing map of the SharePlex configuration file:
   a. the name of the source server as the target host.
   b. the name of the cloud database as the target database

For more information about how to specify configuration components, see **Configure SharePlex to replicate data** in the SharePlex Administration Guide.

Post to PaaS Cloud from an Intermediary Server for Oracle

These instructions help you set up SharePlex to replicate to a target cloud database from an intermediary server that runs the SharePlex Import and Post components.

See also:

Post to PaaS cloud from the source server.

Overview of SharePlex setup on cloud databases.

Source Oracle to cloud Oracle

1. Complete the SharePlex preinstallation checklist.
2. On the source server, install SharePlex for the source database. See: Install SharePlex on Linux/Unix for Oracle Database.
3. On the source server, run **ora_setup** to establish connection information for the source database. See Database Setup for Oracle.

   IMPORTANT:

   a. When asked whether this is a bequeath connection, enter Y, unless this system is RAC.

   b. When asked if the current SID will be used as a source, enter Y.
4. On the intermediary server, install the Oracle binaries. Do not create an Oracle instance.
5. On the intermediary server, add an entry in the oratab file that points to the local Oracle home, and create a dummy entry for the ORACLE_SID. Make certain to specify N so that the startup scripts do not attempt to start the dummy instance.

   ```
   orcldb:/u01/app/oracle/product/12.1.0/dbhome_1:N
   ```
6. On the intermediary server, add an entry in the tnsnames.ora file that points to the target cloud database.
7. On the intermediary server, install SharePlex for the target cloud database. When prompted for the ORACLE_SID, specify the tns_alias that you created for the target cloud database. See: Install SharePlex on Linux/Unix for Oracle Database.

8. On the intermediary server, run ora_setup to establish connection information for the target cloud database. Run it as the master user that was created when the cloud service was created. This user has the required DBA privileges to establish connection information for the target database.

   IMPORTANT:

   a. When asked whether this is a bequeath connection, enter N. This directs SharePlex to use TNS to connect to the target cloud database.

   b. When prompted to specify the `tns_alias`, specify the one that you created for the target cloud database.

   c. When asked if the current SID will be used as a source, enter N.

9. Specify the following in the routing map of the SharePlex configuration file:

   - the name of the intermediary server as the target host.
   - the ORACLE_SID of the target cloud database.

In the following example, intermediary1 is the intermediary server and orcldb1 is the **tns_alias** of the target cloud database.

datasource:o.orcldbprim

```
#source tables              target tables               routing map
splex.demo_src               splex.demo_dest           intermediary1@o.orcldb1
```

For more information, see **Configure SharePlex to replicate data** in the SharePlex Administration Guide.

Source Oracle to cloud Open Target

1. Complete the SharePlex preinstallation checklist.
2. On the source server, install SharePlex for the source database. See: Install SharePlex on Linux/Unix for Oracle Database.
3. On the source server, run the appropriate database setup utility for the source database. See: Database Setup for Oracle.

   IMPORTANT:

   a. (Oracle setup only) When asked whether this is a bequeath connection, enter Y to use bequeath, unless this system is RAC.

   b. Reply Y when asked if the database or SID will be used as a source.

4. On the intermediary server, install the appropriate ODBC client of the target cloud database.
5. On the intermediary server, install SharePlex for the target cloud database. See: Install SharePlex on Linux/Unix for Oracle Database.

6. On the intermediary server, run the appropriate database setup utility for the target cloud database. See Database Setup Utilities.

   IMPORTANT:

   - If the target is Aurora, use the mysql_setup utility.
   - If the target is SQL Server, reply N when asked if this database will be used as a source.
   - (All targets) Specify the full target database name when prompted for the connection string.

7. Specify the following in the routing map of the SharePlex configuration file:

   a. the name of the intermediary server as the target host.

   b. the name of the cloud database as the target database.

In the following example using a MySQL target, intermediary3 is the intermediary system and sptest3 is the target cloud database.

datasource:o.orcldbprim

```
#source tables       target tables      routing map

HR.EMP               "sptest3"."emp"    intermediary3@r.sptest3
```

For more information, see Configure SharePlex to replicate data in the SharePlex Administration Guide.

Installation and Setup for Remote Capture

SharePlex can capture from an Oracle database that is running on a system other than the system where SharePlex is running.

This support covers Oracle databases running on Linux. This support applies only when SharePlex and the remote capture database are running on the same operating system type (i.e., if SharePlex is on Linux, the remote database must be on Linux).

SharePlex does not support remote capture from Oracle on RAC unless the database is running on ASM.

NOTES:

- If configuring SharePlex to capture from a remote database that is on Amazon AWS, see the *SharePlex Installation and Setup Guide for Oracle on RDS*.
- Remote capture using more than one SharePlex instance to the same Oracle instance is not supported.

To set up SharePlex for remote capture:

1. Install an Oracle client on the system where SharePlex is running. No database is required on this system.
2. Set the `ORACLE_HOME` system variable on the SharePlex system to the Oracle client installation folder.
3. Set the `ORACLE_SID` to a dummy value, as a placeholder for the non-existent local database.
4. Create a **tnsnames.ora** file on the SharePlex system and make an entry for the remote source database. Place the file in the path specified in the `TNS_ADMIN` environment variable.
5. Install SharePlex on the remote system in the normal manner.
6. Run Oracle Setup on the SharePlex system according to the directions in Database setup for Oracle.
    - Reply **NO** to "Will SharePlex install be using a BEQUEATH connection?"
    - Reply **NO** to "Are you setting up SharePlex for an RDS database?"
    - Reply **YES** to "Will the current setup for sid: *SID* be used as a source (including cases as source for failover or primary-primary setups)?"
7. Complete all other installation and setup tasks according to this document.
8. After setup is complete, start **sp_cop** and **sp_ctrl**.
9. Set the `SP_OCT_OLOG_USE_OCI` parameter to 1 to enable it. When capturing from a remote database, Capture must read the redo logs through OCI calls because it does not have access to the log files on disk.
10. See the SharePlex Administration Guide for help with configuring and starting replication.

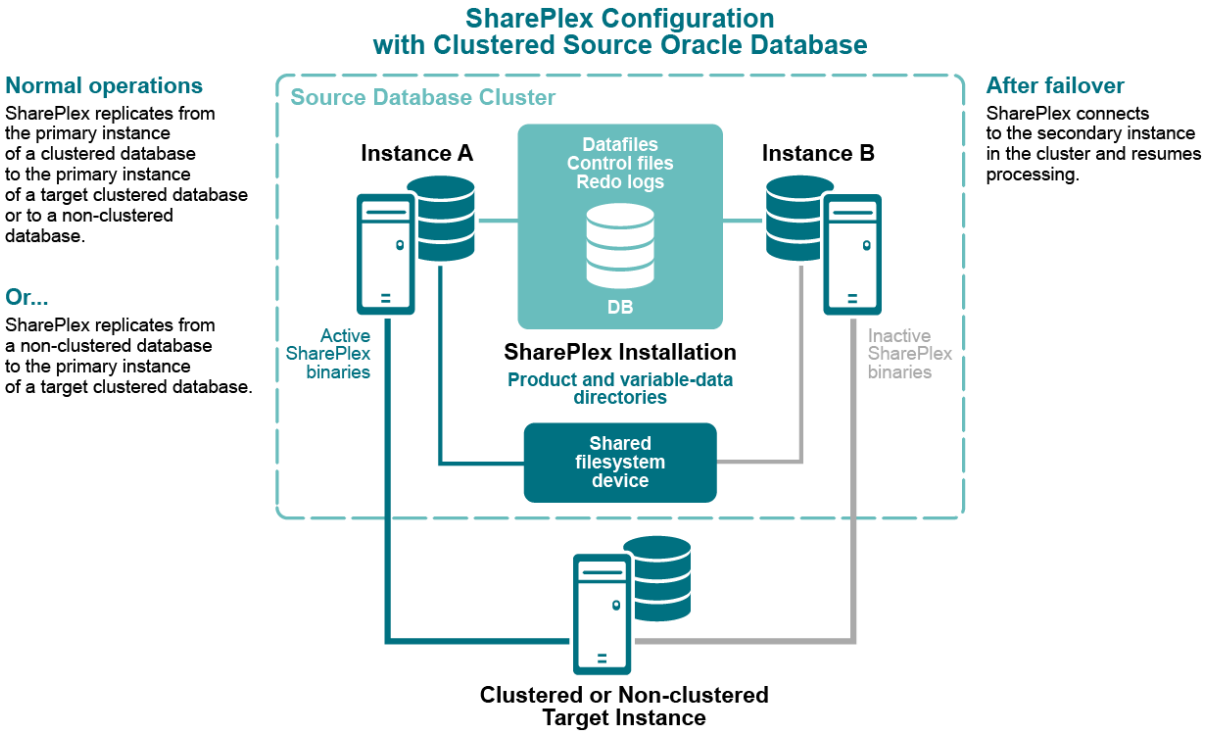
Installation and Setup for HA Cluster

Overview of SharePlex Oracle cluster support

SharePlex integrates with Oracle Clusterware cluster hardware and software to maintain the high availability of data capture and uninterrupted replication to your targets. If the node where SharePlex is running fails or must be taken out of the cluster for maintenance, SharePlex can be started on another server by the cluster software. SharePlex start and stop is controlled through the cluster.

These instructions assume that the cluster solution is already installed according to the cluster documentation, tested, and is functioning, and they are not a substitute for the documentation. Additional steps that are specific to your cluster installation may be required.

The following diagram shows SharePlex installed into an Oracle RAC cluster:



Set up SharePlex in an Oracle cluster

These instructions are for setting up SharePlex in an Oracle cluster.

IMPORTANT! These instructions cover the parts of installing and setting up a cluster that pertain to the setup of SharePlex in the cluster. See the Oracle documentation for complete instructions for the platform that you are using.

Configure the cluster

Perform these steps on a source cluster and a target cluster, if applicable:

1. On one node of the cluster, create a static application Virtual IP (VIP) address for SharePlex. This VIP must point to the node where SharePlex will run and fail over to the secondary node if the primary node fails.

   NOTES:

   - See the Oracle documentation for more information about creating a Virtual IP address on your specific platform.
   - The VIP establishes a consistent host name across all nodes.
   - The cluster software maps the VIP to the SharePlex server and migrates it during a failover.
   - Alternatively, this mapping can be done as a host alias in the **/etc/hosts** file (on Unix and Linux) on both nodes. The alias is exported in the SharePlex user profile and used in the SharePlex configuration parameters.

     Example: `123.456.0.78 splex`

2. Create a **TNS alias** for SharePlex to use to connect to the database on each node. **Use the same alias on each node of a cluster.** Set load balance to **off** and set failover to **on**. For example:

   Node 1:

   ```
   ora_a_sp =
          (DESCRIPTION =
          (ADDRESS_LIST =
          (ADDRESS = (PROTOCOL = TCP)(HOST = RAC1)(PORT = 1521))
          (ADDRESS = (PROTOCOL = TCP)(HOST = RAC2)(PORT = 1521))
          )
          (LOAD_BALANCE = OFF)
          (FAILOVER = ON)
            (CONNECT_DATA =
            (SERVER = DEDICATED)
            (SERVICE_NAME = ora_a)
            (INSTANCE_NAME = ora_a1)
            )
          )
   ```

   Node 2:

   ```
   ora_a_sp =
        (DESCRIPTION =
        (ADDRESS_LIST =
   ```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = RAC2)(PORT = 1521))
(ADDRESS = (PROTOCOL = TCP)(HOST = RAC1)(PORT = 1521))
)
(LOAD_BALANCE = OFF)
(FAILOVER = ON)
  (CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = ora_a)
  (INSTANCE_NAME = ora_a2)
  )
)
```

NOTE: A **TNS alias** establishes global connection information that supercedes local instance names and enables SharePlex to connect to the failover instance without requiring a configuration reactivation. SharePlex identifies the correct Oracle instance from the configuration file.

3. (UNIX and Linux only) Add the **TNS alias** to the **oratab** file on each node that SharePlex is expected to start on during a failover.

   Example (all nodes):

   ```
   ora_a:/u01/app/oracle/product/11.2.0/dbhome_1:N
   ora_a_sp:/u01/app/oracle/product/11.2.0/dbhome_1:N
   ```

4. If the Oracle instances in the cluster have different ORACLE_HOMEs, edit the **oratab** file on each node and on the DNS nameserver, if applicable, to use a symbolic link in place of the actual ORACLE_HOME path:

   SID:/*path_to_symbolic_link*:N

Install SharePlex on the Cluster

In a cluster, SharePlex is installed on Unix and Linux.

To install on Unix and Linux:

On Linux and Unix, the best practice is to install both the SharePlex variable-data and product directories on a shared drive, rather than on each node. This configuration enables more efficient failover and faster upgrades of SharePlex.

If you do install these directories locally on both nodes of the cluster, do the following:

- Install each SharePlex instance on the same port number, with identical path names. When installing on the secondary node, specify a temporary variable-data directory and then remove it after installation. Alternatively, you can use a shared drive for the variable-data directory, and then specify that directory when installing SharePlex on each node. You will configure SharePlex to use the correct variable-data directory in this procedure.
- When prompted by the installer for the ORACLE_SID, specify the **TNS alias** .
- For more information, see Install SharePlex on Linux/Unix for Oracle database on page 33.

Run database setup

After you install SharePlex, run the Database Setup utility. The following applies in a cluster:

- **Source cluster:** Run Database Setup and specify the **TNS alias** as the connection type.
- **Target system or target cluster:**Run Database Setup and specify the **TNS alias** as the connection type, *except if you are populating the target with a hot backup*. If you are populating the target with a hot backup, *do not* run Database Setup now. You will run Database Setup during the activation procedure.
- For more information, see Database setup for Oracle on page 139.

Persist the SharePlex licenses

If you are using shared drive for variable data directory, license key added for primary node will be applicable for all other secondary nodes. If you are using local variable data directory, license need to be installed on the secondary nodes as well using the **splex_add_key** utility.

Set the SharePlex environment parameters

Perform these steps on a source cluster and on a target cluster, if applicable.

Set the SharePlex environment to point to the VIP alias and SharePlex variable-data directory.

- SP_SYS_HOST_NAME directs SharePlex to use the VIP alias when any of its processes issues a name lookup, superseding the local system name. It ensures that **sp_ctrl** commands are directed to the correct host, in this case the cluster name, and it enables SharePlex to migrate properly during failover.
- SP_SYS_VARDIR points to the variable-data directory that you installed on the shared disk. This is the active variable-data directory. Setting SP_SYS_VARDIR ensures that the current replication environment continues to be used by SharePlex after failover.

To set the environment on UNIX and Linux:

EXPORT SP_SYS_HOST_NAME="splex"

SP_SYS_PRODDIR=/home/shareplex

SP_SYS_VARDIR="/app/shareplex/vardir"

Configure SharePlex

When you create the configuration file that directs SharePlex replication, you specify the **TNS alias** rather than an actual ORACLE_SID, as follows.

To capture from a source cluster:

Specify the **TNS alias** on the **o.***datasource* line of the file. This is the first line of the configuration file.

For example:

```
datasource:o.ora_a_sp
```

To post to a target cluster:

Specify the **TNS alias** as the destination in the routing map.

For example (assumes the use of wildcards to specify objects):

```
expand hr.%     hr.%    inst_c@ora_b_sp
```

Activate replication

Once a source and target cluster (if applicable) are configured, you can populate a standalone or clustered target with a copy of the source data, while replication keeps track of transactional changes made by users on the source database. Use any of the Oracle-specific copy methods for activating replication from an active source. See the procedures listed in Activate replication in your production environment in the SharePlex Administration Guide for more information.

Add SharePlex to the cluster software

Incorporate SharePlex as a resource in the cluster software, and include it in the cluster failover routines so that it migrates with the other applications during failover. This ensures that the **sp_cop** process is started on the adoptive node by the cluster software. At minimum, this includes creating a startup script for SharePlex and a cluster script for SharePlex to handle failover.

NOTES:

- The **sp_cop** program is the *only* process that the cluster software should start. The **sp_cop** process must be allowed to start the other SharePlex processes. All SharePlex processes, except **sp_cop**, can be controlled through the **sp_ctrl** interface.
- Do not attempt to start or stop **sp_cop** yourself through the command interface; otherwise the cluster software will attempt to restart it. If you need to stop **sp_cop**, use the cluster software commands.
- If possible, configure SharePlex and Oracle into a single global cluster package. The combination of SharePlex and Oracle in the same package allows the cluster software to start and stop SharePlex and Oracle in the proper sequence if any component of the package fails. Configure Oracle to start before SharePlex.
- Assistance for creating startup and cluster scripts is available through SharePlex Professional (PSO) Services.

System maintenance

Make certain your systems administrators understand that any changes or upgrades they perform to the operating system on any node in the cluster must be implemented on all nodes in the cluster so that SharePlex fails over to an identical environment.

Basic SharePlex Demonstration for Oracle

Contents

Prework for the demonstrations

Start SharePlex

Create and activate a configuration for Oracle

Demonstration of replication

Demonstration of data compare and repair

Demonstration of named post queues

Overview

This chapter demonstrates the basics of SharePlex replication. This demonstration can be run on Unix or Linux from Oracle source to supported target databases.

NOTES:

- These demonstrations are for use with databases. They do not support replication to a file or a messaging container.
- These are only demonstrations. Do not use them as the basis for deployment in a production environment. To properly implement replication in your environment, follow the instructions in the SharePlex Installation and Setup Guide and the SharePlex Administration Guide.
- For more information about the commands used in the demonstrations, see the SharePlex Reference Guide.
- The demonstrations assume that SharePlex is fully installed on a source system and one target system, and that any pre- and post-installation setup steps were performed.

What you will learn

- How to activate a configuration
- How SharePlex replicates smoothly from source to target systems
- How SharePlex quickly and accurately replicates large transactions
- How SharePlex queues the data if the target system is unavailable
- How SharePlex resumes from its stopping point when the target system is recovered
- How SharePlex recovers after a primary instance interruption
- How SharePlex replicates an Oracle TRUNCATE command
- How SharePlex verifies synchronization and repairs out-of-sync rows
- How to use named queues to spread the processing of different tables across parallel Post processes

Prework for the demonstrations

Before you run the basic demonstrations, have the following items available.

Tables used in the demonstrations

You will replicate **splex.demo_src** from the source system to **splex.demo_dest** on the target system. These tables are installed by default into the SharePlex schema, which in these demonstrations is "**splex**." Your SharePlex schema may be different. Verify that these tables exist.

Description of the demo tables.

Column Name	Data Type	Null?
NAME	varchar2(30)	
ADDRESS	verchar2(60)	
PHONE	varchar2(12)	

INSERT scripts

- Create a SQL script named **insert_demo_src** that inserts and commits 500 rows into the **splex.demo_src** table. You will run this script during some of the demonstrations.
- If you will be using the demonstration of named post queues, create a SQL script named **insert_demo_dest** that inserts and commits 500 rows into the **splex.demo_dest** table. You will run this script during some of the demonstrations.

Start SharePlex

The following are instructions for starting SharePlex and the **sp_ctrl** command-line interface on Unix and Linux. Start SharePlex on the source and target systems.

To start SharePlex on Unix and Linux systems:

1. Log onto the system as a SharePlex Administrator (a member of the ***SharePlex Admin group***).
2. From the **bin** sub-directory of the SharePlex product directory (*productdir* in the following syntax), run **sp_cop** and **sp_ctrl**.

   $**cd** /*productdir*/**bin**

   $**./sp_cop &**

   $**. /sp_ctrl**

Create and activate a configuration for Oracle

SharePlex gets its replication instructions from a configuration file, which defines the objects that are to be replicated. The file specifies following:

- The *datasource* (source database) — the identifier of the source database.
- The *source* objects — the names of the objects that contain the data to be replicated.
- The *target* objects — the names of the target objects that will receive the replicated data.
- The *routing map* — the name of the target system and, if the target is a database, its identifier.

NOTE: This demonstration demonstrates replication from one database to another. It does not cover replication to a file or a messaging container.

Create a configuration file

Perform these steps on the **source** system. The demonstration objects are assumed to be in the schema named **splex**.

Perform the following steps to create a configuration file:

1. Run **sp_ctrl**.
2. Issue the following command to create a configuration file named **sample_config** in the default text editor.

   `sp_ctrl(source)>` **create config sample_config**
3. In the text editor, build your configuration file based on the appropriate template, as shown in the Configuration templates. Allow no spaces between the characters in a component (source specification, target specification, routing map), and place at least one space between each component.
4. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory of the variable-data directory.
5. In **sp_ctrl**, verify that the configuration file will activate successfully.

   `sp_ctrl(source)>`**verify config sample_config**

Configuration templates

Oracle source to Oracle target

datasource:o.*source_SID*

splex.demo_src **splex.demo_dest** *target_system***@o.***target_SID*

where:

- *source_SID* is the ORACLE_SID of the source database.
- *target_system* is the name or IP address of the target system.
- *target_SID* is the ORACLE_SID of the target database.

Oracle source to Open Target

datasource:o.*source_SID*		
splex.demo_src	**splex.demo_dest**	*target_system***@r.***database_name*

where:

- *source_SID* is the ORACLE_SID of the source database.
- *target_system* is the name or IP address of the target system.
- *database_name* is the name of the Open Target database.

Activate the configuration

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

Perform the following steps to activate a configuration file:

1. Activate the configuration.

   `sp_ctrl(source)` **> activate config sample_config**

   NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **sample_config** should appear under **File Name**, and the word **Active** should appear under **State**.

   `sp_ctrl(source)` **> list config**

Troubleshooting tips

If your configuration activation fails, isue the **view config sample_config** command to view the file. Compare it to the template and make sure all of the information you entered is correct. Make certain you specified the correct database identifier. Check your syntax for extra spaces or missing components.

Because the configuration file is not active, you can edit it directly with the following command:

`sp_ctrl(source)` **> edit config sample_config**

Save the changes, then re-try the activation.

NOTE: To change an active configuration, you must copy it to a new file first, and then edit and activate the copy. For more information, see *Add or Change Table Specifications in an Active Configuration* in the SharePlex Installation and Setup Guide.

Demonstration of replication

This section demonstrates the speed and accuracy of SharePlex replication. It can also be used to verify that SharePlex was installed and configured correctly.

Verify replication startup

This test verifies that replication is working properly.

Perform these steps in the native SQL interface of the database:

1. On the **source**, TRUNCATE **splex.demo_src** to make certain it is empty. SharePlex replicates the TRUNCATE to the target.

   truncate table splex.demo_src;

2. Insert and commit a record into **splex.demo_src**:

   insert into splex.demo_src values ('Jim', '123 Main Street', '123-456-7890');

   commit;

3. Verify that the record exists in **splex.demo_dest** on the target system:

   select * from splex.demo_dest;

   The query should show the replicated record, and only that record.

Verify replication of large data volumes

This test verifies that SharePlex replicates large volumes of data quickly and accurately.

Perform these steps in the native SQL interface of the database:

1. On the **source**, TRUNCATE **splex.demo_src** to make certain it is empty. SharePlex replicates the TRUNCATE to the target.

   truncate table splex.demo_src;

2. Run the **insert_demo_src** script to insert 500 rows into the **splex.demo_src** table.
3. Verify that all of the record exist in **splex.demo_dest** on the target system:

   select count (*) from splex.demo_dest;

   The count should match the number of records inserted by the **Inserts** script.

Verify queuing and continuity of replication

This test shows you how:

- SharePlex queues replicated data on the source system if the target system is unavailable.
- SharePlex resumes replication from where it left off when the target becomes available.

1. On the **source**, TRUNCATE **splex.demo_src** to make certain it is empty. SharePlex replicates the TRUNCATE to the target.

   truncate table splex.demo_src;

2. On the **target**, shut down SharePlex to simulate that this system is unavailable.

   `sp_ctrl(source)>` **shutdown**

3. On the **source**, run the **insert_demo_src** script to insert records into **splex.demo_src**.

4. On the **source**, issue the following command to show the status of the local SharePlex queues.

   `sp_ctrl(source)>` **qstatus**

   The output should show a backlog of messages.

   NOTES:

   - If SharePlex on the target were running instead of stopped, there would be no messages in the queue. Replication of 500 rows (and the clearing of the queues) would typically happen faster than the time it took to run the script and then issue the **qstatus** command. To confirm that the queues are storing the records, continue to issue the command. The backlog value should be the same each time.
   - If your environment permits it, you can perform the same test with the same results by unplugging the network connection to the source system before you run the script.

5. On the **target**, start **sp_cop** to allow replication to resume.

6. On the **target**, run **sp_ctrl**.

7. On the **target**, verify that the SharePlex processes started.

   `sp_ctrl(target)>` **status**

8. On the **target**, verify that the records inserted by the script on the source system now exist in the target database.

   select count (*) from splex.demo_dest;

   The count should match the number of records inserted by the **Inserts** script.

Verify SharePlex capture recovery

This test shows how SharePlex recovers after an interruption to data capture:

1. On the **source**, TRUNCATE **splex.demo_src** to make certain it is empty. SharePlex replicates the TRUNCATE to the target.

   truncate table splex.demo_src;

2. On the **source,** stop the Capture process.

   `sp_ctrl(source)>` **stop capture**

3. On the **source**, run the **Insert** script to generate a transaction to **splex.demo_src**.

4. On the **source**, start the Capture process.

   ```
   sp_ctrl(source)> start capture
   ```

5. On the **source**, verify that Capture is running and that it processed the data. Look at **Capture state** and **Operations captured**.

   ```
   sp_ctrl(source)> status
   ```

6. On the **target**, use the native SQL interface to verify that all records inserted by the script on the source now exist in the target database.

   select count (*) from splex.demo_dest;

Demonstration of data compare and repair

This demonstration shows you how to use the SharePlex **compare** command to compare the source and target data, and then repair rows that are out of synchronization.

1. On the **source**, TRUNCATE **splex.demo_src** to make certain it is empty. SharePlex replicates the TRUNCATE to the target.

   truncate table splex.demo_src;

2. On the **source**, use the **insert_demo_src** script to insert rows into **splex.demo_src.**
3. On the **target**, verify that all of the data posted to **splex.demo_dest**. If the command shows that data is still in the queue, issue the command again.

   `sp_ctrl(target)>`**qstatus**

4. On the **source**, issue the **compare** command in **sp_ctrl**.

   `sp_ctrl (source)>`**compare splex.demo_src**

   Note: The command determines the correct target table from the configuration file.

5. On the **source**, view the results of the comparison. There should be no rows out of synchronization.

   `sp_ctrl(source)>`**compare status**

6. On the **target**, UPDATE **splex.demo_dest** to change the values of the NAME column in two or more rows. The UPDATE causes the source and target tables to be out of synchronization.
7. On the **source**, issue the **compare** command again. It should show that the rows you updated in **splex.demo_dest** are out of synchronization.

   `sp_ctrl(source)>`**compare splex.demo_src**

8. On the **source**, issue the **repair** command to repair the rows that are out of synchronization.

   `sp_ctrl(source)>` **repair splex.demo_src**

9. On the **source**, verify that the repair was performed.

   `sp_ctrl(source)>`**repair status**

10. On the **target**, you can manually verify that the repair was accurate by using a SELECT statement to view all rows in both tables.

    select * from splex.demo_src;

    select * from splex.demo_dest;

Demonstration of named post queues

This demonstration shows you how to use named post queues to process different tables through different, parallel Post processes to improve performance.

Clean up the replication environment

NOTE: The demonstration objects are assumed to be in the schema named **splex**.

Perform the following steps to clean up the replication environment:

1. **If you ran previous demonstrations, do the following:**
   a. On the **source** and **target**, run **sp_ctrl** and issue the following command to shut down **sp_cop**.

      `sp_ctrl(source)` **shutdown**

      `sp_ctrl(target)` **shutdown**

   b. On the **source** and **target**, run the **ora_cleansp** or **mss_cleansp** utility according to the instructions in SharePlex utilities. This removes the queues from the previous demonstrations and deactivates the previous configuration.
2. On the **source**, TRUNCATE the **splex.demo_src**. SharePlex replicates the TRUNCATE to the target.

   truncate table splex.demo_src;

Create a configuration file

Perform these steps on the **source** system. The demonstration objects are assumed to be in the schema named **splex**.

You will configure the following:

- **demo_src** on the source replicates through a post queue named **q1** to target **demo_dest**.
- **demo_dest** on the source replicates through a post queue named **q2** to target **demo_src**.

1. Run **sp_ctrl**.
2. Issue the following command to create a configuration file named **postq_config** in the default text editor.

   `sp_ctrl(source)>`**create config postq_config**
3. In the text editor, build your configuration file based on the appropriate template, as shown in the Configuration templates. Allow no spaces between the characters in a component (source specification, target specification, routing map), and place at least one space between each component.
4. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory of the variable-data directory.
5. In **sp_ctrl**, verify that the configuration file will activate successfully.

   `sp_ctrl(source)>`**verify config postq_config**

Configuration templates

Oracle source to Oracle target

datasource:o.*source_SID*		
splex.demo_src	**splex.demo_dest**	*target_system*:**q1@o.***target_SID*
splex.demo_dest	**splex.demo_src**	*target_system*:**q2@o.***target_SID*

where:

- *source_SID* is the ORACLE_SID of the source database.
- *target_system* is the name or IP address of the target system.
- *target_SID* is the ORACLE_SID of the target database.

Oracle source to Open Target

datasource:o.*source_SID*		
splex.demo_src	**splex.demo_dest**	*target_system*:**q1@r.***database_name*
splex.demo_dest	**splex.demo_src**	*target_system*:**q2@r.***database_name*

where:

- *source_SID* is the ORACLE_SID of the source database.
- *target_system* is the name or IP address of the target system.
- *database_name* is the name of the open target database.

Activate the configuration

IMPORTANT! Make certain you created the scripts described in Prework for the demonstrations on page 192**.**

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

Perform the following steps to activate the configuration:

1. Activate the configuration.

   ```
   sp_ctrl(source)>activate config postq_config
   ```

   NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **postq_config** should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

Generate data

On the **source**, run the following scripts to insert data into the two local demonstration tables (in this demonstration, the **demo_dest** table is used as a source):

- Run the **insert_demo_src** script to insert 500 rows into the **splex.demo_src** table.
- Run the **insert_demo_dest** script to insert 500 rows into the **splex.demo_dest** table.

View the post queues

Perform the following steps to view the post queues:

1. On the **target**, run **sp_ctrl**.
2. On the **target**, issue the show post command with the detail option. This command shows statistics about the number of messages processed and the queues that were used.

   ```
   sp_ctrl(target)> show post detail
   ```

   In the output, the **Queue** field shows the name of each post queue, in this case **q1** and **q2**, and the **Operations Posted** field shows the number of operations that were posted. Each queue should show 500 operations posted.

Advanced SharePlex Demonstrations for Oracle

Overview

This chapter demonstrates selected features of SharePlex. These exercises can be run on Unix and Linux systems to demonstrate:

- How to build and verify a replication configuration
- How to use the **compare** command to verify synchronization
- How to use partitioned replication to replicate subsets of data
- How to use transformation to manipulate replicated data
- How to use generic conflict resolution in peer-to-peer replication

NOTES:

- These are only demonstrations. Do not use them as the basis for deployment in a production environment. To properly implement replication in your environment, follow the instructions in the SharePlex Installation and Setup Guide and the SharePlex Administration Guide.
- For more information about the commands used in the demonstrations, see the SharePlex Reference Guide.
- The demonstrations assume that SharePlex is installed on a source system and one target system, and that any pre- and post-installation setup steps were performed. For more information, see SharePlex Installation and Setup Guide.

Prework for the demonstrations

Perform these tasks before you start the demonstrations.

Verify that DDL replication is enabled

On the **source**, verify that the `SP_OCT_REPLICATE_DDL` parameter is set to the default of 3. This parameter controls basic DDL replication, including the TRUNCATE command.

```
sp_ctrl(source)>list param modified capture
```

The `SP_OCT_REPLICATE_DDL` parameter should not be listed in the output. If it is, issue the following command:

```
sp_ctrl(source)>reset param SP_OCT_REPLICATE_DDL
```

Install the demonstration objects

The **p2p.sql** and **od.sql** scripts install the demonstration objects that you will use during these demonstrations. These scripts are stored in the **util** sub-directory of the SharePlex product directory.

To run the scripts:

- Run **od.sql** first, then run **p2p.sql**.
- Run the scripts on the source and target systems that you will be using for the demonstrations.
- Run them in SQL*Plus as an existing user with the DBA role and SELECT ANY TABLE privileges.

The scripts prompt for the following items:

- The schema where you want the demonstration objects to be installed.
- The tablespace for the demonstration objects.
- Whether or not you want old demonstration objects from a previous version of SharePlex to be removed. To remove the old objects, supply the name of their schema.

Description of the demonstration objects

od_employee

Name	Null?	Type
EMP_NO	NOT NULL	NUMBER
EMP_FIRST_NAME		VARCHAR2
EMP_LAST_NAME		VARCHAR2
EMP_DOB		DATE
EMP_DEPT_NO		NUMBER
EMP_TIMESTAMP		DATE

od_timesheet

Name	Null?	Type
TS_EMP_NO		NUMBER
TS_IN_TIME		DATE
TS_OUT_TIME		DATE
TS_MOD_TIME		DATE

od_department

Name	Null?	Type
DEPT_NO	NOT NULL	NUMBER
DEPT_NAME		VARCHAR2
DEPT_CODE		VARCHAR2

od_salary

Name	Null?	Type
SALE_EMP_NO		NUMBER
SAL_VALUE		NUMBER
SAL_CHANGED		DATE

od_sales_emp_data

Name	Null?	Type
EMP_NO_KEY	NOT NULL	NUMBER
EMPLOYEE_NAME		VARCHAR2 (70)
SALARY		NUMBER
DEPARTMENT		VARCHAR2 (50)

oxc_table

Name	Null?	Type
EXC_NO	NOT NULL	NUMBER
EXC_TYPE		VARCHAR2 (6)
EXC_TARGET_TABLE		VARCHAR2 (66)
EXC_FIXED		VARCHAR2 (3)

EXC_INFO		VARCHAR2 (500)
EXC_TIMESTAMP		DATE

Start SharePlex

The following are instructions for starting SharePlex and the **sp_ctrl** command-line interface on Unix and Linux. Start SharePlex on the source and target systems.

To start SharePlex on Unix and Linux systems:

1. Log onto the system as a SharePlex Administrator (a member of the ***SharePlex Admin group***).
2. From the **bin** sub-directory of the SharePlex product directory (*productdir* in the following syntax), run **sp_cop** and **sp_ctrl**.

```
$cd /productdir/bin
$./sp_cop &
$. /sp_ctrl
```

Configure, activate, and verify replication on the Oracle source system

NOTE: Before proceeding, make certain the SharePlex demonstration objects are installed. See Prework for the demonstrations on page 109.

In this demonstration, you will create and activate a replication configuration, load data to the source table, and then confirm that the data was replicated successfully to the target table. You will also repair a table that is out of synchronization.

Create a configuration file

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

Perform these steps on the source system:

1. Run **sp_ctrl** .
2. Issue the following command to create a configuration file named **od_config** that replicates the **od_department**, **od_salary**, **od_timesheet**, and **od_employee** tables to target tables of the same names on the target system.

```
sp_ctrl(source)>create config od_config
```

3. In the text editor, build your configuration file based on the following template. Allow no spaces between the characters in a component (source specification, target specification, routing map), and place at least one space between each component.

# od_config configuration file		
datasource:o.*source_SID*		
demo.od_department	**demo.od_department**	*target_system***@o.***target_SID*
demo.od_salary	**demo.od_salary**	*target_system***@o.***target_SID*
demo.od_timesheet	**demo.od_timesheet**	*target_system***@o.***target_SID*
demo.od_employee	**demo.od_employee**	*target_system***@o.***target_SID*

where:

- # denotes a comment.
- *source_SID* is the ORACLE_SID of the source database.
- *target_system* is the name or IP address of the target system.
- *target_SID* is the ORACLE_SID of the target database.

4. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory.
5. In **sp_ctrl**, verify that the configuration file will activate successfully.

```
sp_ctrl(source)>verify config sample_config
```

Activate the configuration

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

Perform the following steps to activate the configuration:

1. Activate the configuration.

```
sp_ctrl(source)>activate config od_config
```

NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **config od_config** should appear under **File Name**, and the word **Active** should appear under **State**.

```
sp_ctrl(source)>list config
```

NOTE: If you ran the basic SharePlex demonstration using the same datasource , this activation deactivates the configuration file that you used for that demonstration.

Generate transactions to the source table

On the **source**, log in as the demo schema owner and execute the **od_add_emps** procedure to populate the **od_employee** and **od_salary** tables.

This procedure has one IN parameter that specifies the number of employees to insert per department:

- The default number of departments is 5.
- Use an IN parameter of 100 to create 500 new employees in the **od_employee** table and 500 entries in the **od_salary** table.

```
SQL> exec od_add_emps(100);
```

Compare source and target tables

In this section, you will compare the source and target tables to verify that the data is synchronized.

1. On the **source**, run **sp_ctrl**.
2. On the **source**, compare the **od_employee** and **od_salary** tables to their targets.

   ```
   sp_ctrl(source)> compare demo.od_employee
   sp_ctrl(source)> compare demo.od_salary
   ```

   Note: The command determines the correct target table from the configuration file.
3. On the **source**, view the results of the comparisons.

   ```
   sp_ctrl(source)> compare status
   ```

   Continue issuing this command until both compare processes have completed their work. The command output should show no out-of-sync rows.

Repair an out-of-sync condition

In this section, you will repair an out-of-sync condition to restore data synchronization.

Perform the following steps to repair an out-of-sync condition:

1. On the **target**, run SQL*Plus and delete some rows from **od_employee**. This causes that table to go out of synchronization with its source table.
2. On the **source**, compare the **od_employee** tables.

   ```
   sp_ctrl(source)> compare demo.od_employee
   ```
3. On the **source**, view the results of the comparisons.

   ```
   sp_ctrl(source)>compare status
   ```

   The command output should show out-of-sync rows, the number of which should equal the number of rows you deleted from **od_employee** on the target.

4. On the source, repair the **od_employee** table.

   ```
   sp_ctrl(source)>repair demo.od_employee
   ```

5. On the **source**, view the results of the repair.

   ```
   sp_ctrl(source)>repair status
   ```

   The command output should show no out-of-sync rows, because the **repair** command inserted the rows that that you deleted, because they were still present in the source table.

Demonstration of horizontally partitioned replication

NOTE: Before proceeding, make certain the SharePlex demonstration objects are installed. See Prework for the demonstrations on page 109.

In this demonstration you will:

1. Define a *partition scheme* and one or more *row partitions* for the partition scheme.
   - A row partition is a defined subset of the rows in a source table that you want to replicate to the target table.
   - A partition scheme is a logical container for row partitions.
2. Specify the name of the partition scheme in the SharePlex configuration file to include the partitions in replication.
3. Activate the configuration.
4. Load data into the source table and verify replication of the specified rows to the target.

Prepare the tables

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

Perform the following steps to prepare the table:

1. If you ran previous demonstrations, do the following:
   a. On the **source** and **target**, run **sp_ctrl** and issue the following command to shut down **sp_cop**.

      ```
      sp_ctrl (source) shutdown
      ```

      ```
      sp_ctrl (target) shutdown
      ```

   b. On the **source** and **target**, run the **ora_cleansp** utility according to the instructions in "ora_cleansp" in the *SharePlex Reference Guide*. This removes the queues from the previous demonstrations and deactivates the previous configuration.
2. On the **source** and **target**, TRUNCATE the **od_employee** and **od_salary** tables.

   ```
   SQL> truncate table demo.od_employee;
   ```

   ```
   SQL> truncate table demo.od_salary;
   ```

Configure the row partition

Perform the following steps to configure the row partition:

1. On the **source**, run **sp_ctrl**.
2. On the source, create a row partition that specifies the rows to replicate from the source **od_employee** table, in this case only the rows where the employee department is 1.

   NOTE: The partition only specifies rows. The names of the source and target tables are specified in the configuration file.

   `sp_ctrl>` **add partition to scheme1 set condition = 'EMP_DEPT_NO=1' and route =** *target_sys*@o.*ora_SID*

   where:

   - **scheme1** is the name of the partition scheme.
   - **condition = 'EMP_DEPT_NO=1'** is the column condition
   - **route =** *target_sys*@o.*ora_SID* is the routing that consists of the name of your target system and the ORACLE_SID of the target database.

Specify the partition scheme in the configuration file

Perform these steps on the source system.

1. In **sp_ctrl**, open the **od_config**configuration file for editing.

   `sp_ctrl(source)>`**edit config od_config**

2. In the text editor, edit the configuration file to change the entry for the **od_employee** table so that it uses the partition scheme that you created.

   ```
   # od_config configuration file
   datasource:o.source_SID
   demo.od_department      demo.od_department      target_system@o.target_SID
   demo.od_salary          demo.od_salary          target_system@o.target_SID
   demo.od_timesheet       demo.od_timesheet       target_system@o.target_SID
   demo.od_employee        demo.od_employee        !scheme1
   ```

   Where:

   - *source_SID* is the ORACLE_SID of the source database.
   - *target_system* is the name or IP address of the target system.
   - *target_SID* is the ORACLE_SID of the target database.
   - **!scheme1** is the syntax for listing the partition scheme.

3. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory.

Activate the configuration

Perform the following steps to activate the configuration:

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

1. Activate the configuration.

   ```
   sp_ctrl(source)>activate config od_config
   ```

   NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **config od_config** should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

Replicate data

Perform the following steps to replicate data:

1. On the **source**, log in as the demo schema owner and execute the **od_add_emps** procedure to populate the **od_employee** and **od_salary** tables. This procedure has one IN parameter that specifies the number of employees to insert per department:

   - The default number of departments is 5.
   - Use an IN parameter of 100 to create 500 new employees in the **od_employee** table and 500 entries in the **od_salary** table.

   ```
   SQL>exec od_add_emps(100);
   ```

2. On the **source**, select all rows from the source **od_employee** table.

   ```
   SQL> select * from od_employee;
   ```

   The value for the EMP_DEPT_NO column should range from '1' to '5' across the rows.

3. On the **target**, select all rows from the target **od_employee** table.

   ```
   SQL> select * from od_employee;
   ```

   The value for the `EMP_DEPT_NO` column should be '1' for all rows. Rows where the value for this column is a value other than “1” were not replicated.

Demonstration of vertically partitioned replication

NOTE: Before proceeding, make certain the SharePlex demonstration objects are installed. See Prework for the demonstrations on page 109.

In this demonstration you will :

- Specify a *column partition* in the configuration file. A column partition replicates only the data changes that are made to the specified columns.
- Activate the configuration.
- Load data into the source table and verify replication of the specified columns to the target.

Prepare the tables

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

Perform the following steps to prepare the tables:

1. **If you ran previous demonstrations, do the following:**
   a. On the **source** and **target**, run **sp_ctrl** and issue the following command to shut down **sp_cop**.

      ```
      sp_ctrl (source) shutdown
      ```

      ```
      sp_ctrl (target) shutdown
      ```

   b. On the **source** and **target**, run the **ora_cleansp** utility according to the instructions in "ora_cleansp" in the *SharePlex Reference Guide*. This removes the queues from the previous demonstrations and deactivates the previous configuration.
2. On the **source** and **target**, TRUNCATE the **od_employee** and **od_salary** tables.

   ```
   SQL> truncate table demo.od_employee;
   ```

   ```
   SQL> truncate table demo.od_salary;
   ```

Configure the column partition

Perform these steps on the source system:

1. In **sp_ctrl**, deactivate the **od_config** configuration.

   ```
   sp_ctrl(source)>deactivate config od_config
   ```

2. Confirm that the configuration deactivated successfully. The name **od_config** should appear under **File Name**, and the **State** column should show that it is inactive.

   ```
   sp_ctrl(source)>list config
   ```

3. In **sp_ctrl**, open the **od_config** configuration file for editing.

```
sp_ctrl(source)>edit config od_config
```

4. In the text editor, edit the configuration file to change the entry for the **od_employee** table so that it uses a column partition.

# od_config configuration file		
datasource:o.*source_SID*		
demo.od_department	**demo.od_department**	*target_system***@o.***target_SID*
demo.od_salary !(SAL_VALUE)	**demo.od_salary**	*target_system***@o.***target_SID*
demo.od_timesheet	**demo.od_timesheet**	*target_system***@o.***target_SID*
demo.od_employee (EMP_NO, EMP_FIRST_NAME, EMP_ LAST_NAME	**demo.od_employee**	*target_system***@o.***target_SID*

Where:

- *source_SID* is the ORACLE_SID of the source database.
- *target_system* is the name or IP address of the target system.
- *target_SID* is the ORACLE_SID of the target database.
- **!(SAL_VALUE)** is the syntax for an excluded column partition. All columns except the one listed are replicated.
- **(EMP_NO, EMP_FIRST_NAME, EMP_LAST_NAME)** is the syntax for a column partition. Only the listed columns are replicated.

NOTES:

- This configuration file template is set up in table form to show the source, target, and routing elements clearly. In a real configuration file, the **source** (including the column partition), **target**, and **routing map** should be in that order, all on one line.
- Any columns that are defined as NOT NULL must be included in the column partition, because SharePlex replicates a NULL into columns that are not in the column partition.

5. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory.

Activate the configuration

Perform the following steps to activate the configuration:

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

1. Activate the configuration.

   ```
   sp_ctrl(source)>activate config od_config
   ```

   NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **config od_config** should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

Replicate data

Perform the following steps to replicate data:

1. On the **source**, log in as the demo schema owner and execute the **od_add_emps** procedure to populate the **od_employee** and **od_salary** tables. This procedure has one IN parameter that specifies the number of employees to insert per department:

   - The default number of departments is 5.
   - Use an IN parameter of 100 to create 500 new employees in the **od_employee** table and 500 entries in the **od_salary** table.

   ```
   SQL>exec od_add_emps(100);
   ```

2. On the **source**, select all rows from the source **od_employee** table.

   ```
   SQL> select * from od_employee;
   ```

   There should be values in all of the columns of the table.

3. On the **target**, select all rows from the target **od_employee** table.

   ```
   SQL> select * from od_employee;
   ```

   There should only be values in the EMP_NO, EMP_FIRST_NAME, and EMP_LAST_NAME columns. The other columns should contain null values.

4. On the **target**, select all rows from the target **od_salary** table.

   ```
   SQL> select * from od_salary;
   ```

   There should only be values in the `SALE_EMP_NO` and `SAL_CHANGED` columns. The `SAL_VALUE` column should contain only nulls.

Demonstration of transformation

NOTE: Before proceeding, make certain the SharePlex demonstration objects are installed. See Prework for the demonstrations on page 109.

In this demonstration, you will use a supplied transformation procedure to have SharePlex replicate data from two separate source tables and apply it to one target table.

Prepare the objects

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

Perform the following steps to prepare the objects:

1. **If you ran previous demonstrations, do the following:**
   a. On the **source** and **target**, run **sp_ctrl** and issue the following command to shut down **sp_cop**.

      ```
      sp_ctrl (source) shutdown
      sp_ctrl (target) shutdown
      ```

   b. On the **source** and **target**, run the **ora_cleansp** utility according to the instructions provided in SharePlex Reference Guide. This removes the queues from the previous demonstrations and deactivates the previous configuration.
2. On the **source** and **target**, TRUNCATE the **od_employee** and **od_salary** tables.

   ```
   SQL> truncate table demo.od_employee;
   SQL> truncate table demo.od_salary;
   ```

3. On the **target**, grant the user who owns the demonstration objects the system privilege to execute the **sp_cr** package, which was installed in the SharePlex schema when SharePlex was first installed.

   ```
   SQL> grant execute on sp_cr to user_name
   ```

4. On the **target**, log into SQL*Plus as the user who owns the SharePlex demonstration objects, then run the **transform.sql** script from the **util** sub-directory of the SharePlex product directory. This installs transformation routines named **od_transform_employee_insert** and **od_transform_employee_update**. You are prompted for:
   - a schema and tablespace
   - the name of the SharePlex database user

Configure SharePlex

Perform the following steps to configure SharePlex:

1. On the **target**, open the **transformation**.*SID* file (where *SID* is the ORACLE_SID of the target database) in a text editor. This file is located in the **data** sub-directory in the SharePlex variable-data directory.

   NOTE: Post checks this file to determine if there is a transformation procedure that it must call instead of posting the operation to the database.

2. Create the following entries in the **transformation**.*SID* file.

   Separate each column with at least a few spaces or a tab character.

demo.od_employee	**I**	**demo.od_transform_employee_insert**
demo.od_employee	**U**	**demo.od_transform_employee_update**
demo.od_salary	**I**	**demo.od_transform_employee_insert**
demo.od_salary	**U**	**demo.od_transform_employee_update**

   NOTE: The components of each entry are as follows, in the order they must appear:

   - The target table to which a transformation procedure is assigned.
   - The operation type for which the specified transformation procedure will be called.
   - The name of the assigned transformation procedure to use. Multiple entries can be used to assign different procedures to different operation types for the same table.

3. On the **target**, enable the following parameter.

   ```
   sp_ctrl(target)> set param SP_OPO_XFORM_EXCLUDE_ROWID 1
   ```

4. On the **source**, create a configuration file named **od.transform** that replicates the **od_salary** and **od_employee** tables.

   ```
   sp_ctrl(source)> create config od_transform
   ```

5. In the text editor, build your configuration file based on the following template.

datasource:o.*source_SID*		
demo.od_salary	**demo.od_salary**	*target_system*@**o.***target_SID*
demo.od_employee	**demo.od_employee**	*target_system*@**o.***target_SID*

6. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory.

Activate and start replication

Perform the following steps to activate and start replication:

1. On the **source**, activate the configuration.

   ```
   sp_ctrl(source)> activate config od_transform
   ```

2. Confirm that the configuration activated successfully. The name **od_transform** should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

3. On the **source**, log in as the demo schema owner, then execute the **od_add_emps** procedure to populate the **od_employee** and **od_salary** tables. Use an IN parameter of 10 to create 50 new employees in the **od_sales_emp_data** table.

   ```
   SQL> exec od_add_emps(10);
   ```

View the transformed data

Perform the following steps to view the transformed data:

1. **On the target**, run SQL*Plus.
2. In SQL*Plus, select all rows from **od_sales_emp_data**.
3. View the transformed data. You should see the following results:
   - The `EMPLOYEE_NAME` column contains the first and last name of the employee. Compare this to the source **od_employee** table, where the first and last names are in separate columns.
   - The `DEPARTMENT` column contains the department name. Compare this to the source **od_employee** table, where the EMP_DEPT_NO column contains a number. The transformation procedure transformed the replicated department number into the department name by referencing the **od_department** table.
   - The `SALARY` column contains the salary from the **od_salary** table.
4. [OPTIONAL] To see how transformation works for UPDATEs, you can update the **od_employee** table manually. The **od_transform_employee_update** procedure will make the transformation. To further this demonstration, you may construct a transformation procedure for DELETEs.

Demonstration of conflict resolution for Oracle

NOTE: Before proceeding, make certain the SharePlex demonstration objects are installed. See Prework for the demonstrations on page 109.

In this demonstration, you will configure SharePlex to use generic conflict resolution procedures to resolve a replication conflict. Generic conflict resolution allows you to use one PL/SQL procedure to resolve conflicts for multiple tables.

The following conflict-resolution strategies are demonstrated:

- Timestamp priority – This demonstration is based on UPDATEs. Whichever row was updated LAST takes priority when there is a conflict.
- Trusted-source priority – In the following steps, you will define one system as the "trusted" source that takes priority in the event of a conflict. This demonstration is based on INSERTs. All INSERTs that originate on the trusted source will override INSERTs from the other system. In this demonstration, the trusted source is **peer1** and the other system is **peer2**.

IMPORTANT! Peer-to-peer replication is not compatible with all business applications. When suitable for an environment, it requires careful analysis and execution, including the creation of custom conflict resolution procedures that are typically more complex than those in this demonstration. Do not use this demonstration as the foundation of a production peer-to-peer deployment. For more information about peer-to-peer replication, see SharePlex Administration Guide.

Prepare the objects

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

Perform these steps on both systems (for both databases).

1. Run **sp_ctrl**.
2. Shut down SharePlex.

   ```
   sp_ctrl(peer1)> shutdown
   ```

3. Run the **ora_cleansp** utility according to the instructions in SharePlex Reference Guide. This removes the queues from the previous demonstrations and deactivates the previous configuration.
4. TRUNCATE the **od_employee** tables.
5. Grant the user who owns the demonstration objects the system privilege to execute the **sp_cr** package, which was installed in the SharePlex schema when SharePlex was first installed.

   ```
   SQL>grant execute on sp_cr to user_name
   ```

6. Log into SQL*Plus as the user who owns the SharePlex demonstration objects.

7. Run the **p2p.sql** script from the **util** sub-directory of the SharePlex product directory. This installs the **od_employee_gen** demonstration conflict resolution routine.

   You are prompted for the following:

   - A schema and tablespace for the procedure
   - The name of the SharePlex database user.
   - The name of the system that will be the *trusted source* of accurate data. As in a production deployment, operations the trusted source take priority during a conflict. This system is known as ***peer1*** in this demonstration. The other system is known as ***peer2*** in this demonstration.

Configure SharePlex

Perform the following steps to configure SharePlex:

1. On each system, open the **conflict_resolution**.*SID* file (where *SID* is the ORACLE_SID of the local database) in a text editor. This file is located in the **data** sub-directory of the SharePlex variable-data directory.

   NOTE: Post checks this file when there is a replication conflict to determine if there is a resolution procedure to call.

2. On each system, create the following entries in the **conflict_resolution**.*SID* file. Separate each column with at least a few spaces or a tab character.

demo.od_employee	**IUD**	**demo.od_employee_gen**

   NOTE: The first component is a table, the second specifies the operation types for which a resolution routine will be called if there is a conflict on that table, and the third is the name of the resolution routine that will be used.

3. On each system, start **sp_cop**.
4. On each system, start **sp_ctrl**.
5. On **peer1** (the trusted source), create a configuration file named **od_peer1** that replicates the **od_employee** table to the **od_employee** table on **peer2**.

   `sp_ctrl(peer1)>` **create config od_peer1**

demo.od_employee	**demo.od_employee**	**peer2@o**.*SID*

6. On **peer2** (the secondary source), create a configuration file named **od.peer2** that replicates the **od_employee** table to the **od_employee** table on **peer1**.

   `sp_ctrl(peer2)>` **create config od_peer2**

demo.od_employee	**demo.od_employee**	**peer1@o**.*SID*

NOTE: In order for post to detect out-of-sync inserts where all columns are identical, set `SP_OPO_SUPPRESSED_OOS` to `0`. Issue this command from **sp_ctrl**: set param `SP_OPO_SUPPRESSED_OOS 0` and verify the parameter is set by using the **list param modified** command.

Activate and start replication

Perform the following steps to activate and start replication:

1. On **peer1**, activate the **od_peer1** configuration.

   ```
   sp_ctrl(peer1)> activate od_peer1
   ```

2. On **peer2**, activate the **od_peer2** configuration.

   ```
   sp_ctrl(peer2)> activate od_peer2
   ```

3. On each system, confirm that the configuration activated successfully. The name **od_peer1** or **od_peer2** (depending on the system) should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

Demonstrate trusted-source priority

In this demonstration, an INSERT that originates on **peer1** will override a conflicting INSERT that is replicated from **peer2**.

1. On **both systems**, stop the Post process.
2. On **both systems**, log in to SQL*Plus as **demo** (the owner of the demonstration objects).
3. On **peer2**, insert a row into **od_employee** but do not issue a COMMIT.

   SQL (peer2) > INSERT INTO OD_EMPLOYEE VALUES (1,'John','Doe',to_date ('04/01/1949','MM/DD/RRRR'),1,to_date('01/01/2017','MM/DD/RRRR'));

4. On **peer1**, insert the same row (same values) but do not issue a COMMIT.

   SQL (peer1) > INSERT INTO OD_EMPLOYEE VALUES (1,'John','Doe',to_date ('04/01/1949','MM/DD/RRRR'),1,to_date('01/01/2017','MM/DD/RRRR'));

5. On **both systems**, restart the Post processes.
6. On **peer2**, issue the COMMIT.
7. On **peer1**, issue the COMMIT. This operation should generate a conflict, which Post resolves automatically based on instructions in the **conflict_resolution**.*SID* file.
8. On **both systems**, view the **demo.exc_table** to verify that the conflict was resolved. See View the conflict resolution results.

Demonstrate timestamp priority

In this demonstration, whichever row was updated LAST takes priority when there is a conflict.

NOTE: The `EMP_TIMESTAMP` field, along with the other columns, must be modified to a non-null value in order for timestamp priority to function. The stored procedure won't be able to resolve the conflict if the timestamp column in the **UPDATE** statement is not being updated or is being modified to a null value. Out-of-sync problems will then be seen.

1. On **both systems**, stop the Post process.
2. On **both systems**, log in to `SQL*Plus` as **demo** (the owner of the demonstration objects).
3. On **peer1**, UPDATE the `EMP_FIRST_NAME` and `EMP_TIMESTAMP` columns of the **od_employee** table as follows.

```
SQL (peer1) > UPDATE OD_EMPLOYEE SET EMP_FIRST_NAME = 'James', EMP_
TIMESTAMP = to_date('01/01/2017','MM/DD/RRRR') WHERE EMP_NO = 1;
```

4. On **peer2**, UPDATE the same column using a different update values. but the same key value.

```
SQL (peer2) > UPDATE OD_EMPLOYEE SET EMP_FIRST_NAME = 'Harry', EMP_
TIMESTAMP = to_date('02/02/2017','MM/DD/RRRR') WHERE EMP_NO = 1;
```

5. On **both systems**, issue COMMITs at the same time.
6. On **both systems**, view the post queue to make sure the update operation is in the queue. You will see a message in each queue.

```
sp_ctrl(peer1)>qstatus

sp_ctrl(peer2)>qstatus
```

7. On **both systems**, start the Post process.
8. On **both systems**, select the row that you updated to verify that it contains the more recent EMP_TIMESTAMP value.

View the conflict resolution results

A table named **exc_table** was installed in the schema that you specified when you installed the demonstration objects. You can view it through SQL*Plus to view information about each conflict. The following is the table description.

Column	Description
EXC_NO	The exception number of the conflict.
EXC_TYPE	The type of SQL statement, whether INSERT, UPDATE or DELETE.
EXC_TARGET_TABLE	The table on which the conflict occurred.
EXC_FIXED	The results of the conflict resolution routine. YES means that the routine was successful. NO means that the routine failed and the row needs to be manually changed to the correct value.
EXC_INFO	The cause of the conflict.
EXC_TIMESTAMP	The time that the conflict occurred on this machine.

Demonstration of Oracle DDL replication

This demonstration shows the default DDL replication support that is enabled when SharePlex is installed. You can enable other DDL replication with parameters, as needed.

NOTE: This demonstration supports Oracle source and targets only.

Verify that DDL replication is enabled

On the **source**, verify that the SP_OCT_REPLICATE_DDL parameter is set to the default of 3. This parameter controls basic DDL replication.

```
sp_ctrl(source)>list param modified capture
```

The SP_OCT_REPLICATE_DDL parameter should not be listed in the output. If it is, issue the following command:

```
sp_ctrl(source)>reset param SP_OCT_REPLICATE_DDL
```

Test DDL replication

Perform the following steps to test DDL replication:

1. On the **source**, TRUNCATE **splex.demo_src** to make certain it is empty.

   ```
   SQL> truncate table splex.demo_src;
   ```

   SharePlex replicates the TRUNCATE command to the target.

2. On the **source**, add a column to **splex.demo_src**.

   ```
   SQL> alter table splex.demo_src add (department varchar2(30) not null default 'unknown');
   ```

3. On the **target**, describe the **splex.demo_dest** table.

   ```
   SQL> describe splex.demo_src;
   ```

   The table should now contain four columns, including the new **department** column.

4. On the **source**, drop the **department** column.

   ```
   SQL> alter table splex.demo_src drop column department;
   ```

5. On the **target**, describe the **splex.demo_dest** table.

   ```
   SQL> describe splex.demo_src;
   ```

   The table should now contain only the original three columns.

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.

Contents

- Database setup for HANA
- Database setup for MySQL
- Database setup for Oracle
- Database setup for SQL Server
- Database setup for Snowflake

Database setup for HANA

Overview

Run the Database Setup utility for HANA (**hana_setup**) on a target HANA system to establish a user account, schema, and tables for use by SharePlex.

Supported databases

HANA on supported platforms

Guidelines for use

- Run the setup utility on all target HANA instances in the SharePlex replication configuration.
- Within a server cluster, run the setup utility on the node that has the mount point to the shared disk that contains the SharePlex variable-data directory.
- For consolidated replication, run the setup utility for each variable-data directory.
- The only supported connection method to HANA is by connect string. Connection through a DSN is not supported.
- Make certain that you assign the required permissions that are shown in this setup process.

Required privileges

The setup utility must be run as a HANA Administrator in order to grant SharePlex the required privileges to operate on the database and to create the SharePlex database account.

Run database setup for HANA

1. Shut down any running SharePlex processes and **sp_cop** on the target system.
2. Run the **hana_setup** program from the **bin** subdirectory of the SharePlex product directory.

IMPORTANT! If you installed the SharePlex instance 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.

$ /users/splex/bin> hana_setup -p9400

Table 5: Setup Prompts and Responses

Prompt	Response
`Enter the HANA connection string [] :`	Enter a connection string that connects to the HANA database system. The required components of a connection string for SharePlex to connect to HANA are: • SERVERNODE: The name of the target HANA server, then a colon, then the HANA port number. • DRIVER: The path to the HANA ODBC driver.

Prompt	Response
	• CHAR_AS_UTF8: This must be passed as **CHAR_AS_UTF8=1**. Example: **SERVERNODE=server1.dept.abc.corp:30015;DRIVER=/usr/sap/hdbclient/libodbcHDB.so;CHAR_AS_UTF8=1** You do not have to supply a user, password, and default database, because the setup utility prompts for those.
`Enter the HANA Administrator name :`	Enter the name of the HANA Administrator. This user will perform the work on the SharePlex account.
`Enter the password for the Administrator account :`	Enter the password of the Administrator.
`Enter the name of the database :`	Enter the name of the database that you want to contain the tables and other objects for use by SharePlex. You can enter the name of a new or existing database.
`Database name` *`database`* `does not exist. Would you like to create it? [y] :`	If this prompt is displayed, the specified database does not exist. Press **Enter** to have the setup utility create it for you.
`Would you like to create a new SharePlex user [y]:`	Press **Enter** to accept the default to create a new SharePlex database user account and schema of the same name in the specified database, or enter **n** to use an existing SharePlex account.

`Enter the name of the new SharePlex user:`
 	
`Enter the name of the existing SharePlex`
	One of these prompts is displayed depending on whether you elected to create a new user or use an existing user. Enter the name of the SharePlex user.

Prompt	Response
`user:`	
`Enter the password of the SharePlex user :`	Enter the password of the SharePlex user account.
`Re-enter the password for the SharePlex user :`	This prompt is only shown if you created a new user. Enter the SharePlex password again.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for HANA database configuration

SharePlex User name: splex

Database name: ndb5

Target specification in SharePlex configuration: r.ndb5
```

Grant privileges to SharePlex

Before you start SharePlex replication to a HANA target, grant the following privileges to the SharePlex database user in that target (where *SP_USER* is the name of the SharePlex database user):

- GRANT USER ADMIN TO *sp_user*;
- GRANT TABLE ADMIN TO *sp_user*;
- GRANT CATALOG READ TO *sp_user*;
- GRANT DATA ADMIN TO *sp_user* WITH ADMIN OPTION;
- GRANT ROLE ADMIN TO *sp_user* WITH ADMIN OPTION;

Additionally, log in as the owner of each schema that contains objects that you want to replicate, then make the following grants on the schema:

- GRANT CREATE ANY ON SCHEMA *schema_name* TO *sp_user*;
- GRANT DEBUG ON SCHEMA *schema_name* TO *sp_user*;
- GRANT DELETE, DROP, EXECUTE, INDEX, INSERT, SELECT, UPDATE ON SCHEMA *schema_name* TO *sp_user*;

Database setup for MySQL

Overview

Run the Database Setup utility for MySQL (**mysql_setup**) on a MySQL system to establish SharePlex as a MySQL database user.

This utility creates the following:

- A SharePlex user account with full DBA privileges
- Tables and indexes for use by SharePlex and owned by the SharePlex user in a database of your choosing
- A default database connection.

Supported databases

MySQL on Linux. For supported Linux platforms and versions, see the SharePlex Release Notes.

Guidelines for use

- Run the setup utility on all MySQL instances in the SharePlex replication configuration.
- Within a cluster, run the setup utility on the node to which the shared disk that contains the SharePlex variable-data directory is mounted.
- For consolidated replication, run the setup utility for each variable-data directory.

Required privileges

Review the following requirements to ensure that the setup succeeds:

- The setup utility must be run as a MySQL Administrator that retains all of that user's default privileges. Both local and cloud MySQL Administrators should have the required privileges by default. The Administrator user is able to grant SharePlex the required privileges to operate on the database and to create the SharePlex database account and objects.
- (Cloud installations) Common restrictions on privileges in cloud-hosted database services make it difficult for the setup utility to succeed in every possible type of scenario. To ensure that the database setup succeeds, *only* use the setup utility for the following purposes: To do a *first-time* database setup with a *new* SharePlex user, or, to *modify* an existing SharePlex user that either owns the database or has access to it.

Run database setup for MySQL

Perform the following steps to to run database setup for MySQL:

1. Shut down any running SharePlex processes and **sp_cop** on the MySQL system.
2. Run the **mysql_setup** program from the **bin** subdirectory of the SharePlex product directory.

   IMPORTANT! If you installed the SharePlex instance 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> mysql_setup -p9400
   ```

Table 6: Setup prompts and responses

Prompt	Response
`Enter the MySQL connection string [] :`	Enter a connection string that connects to the MySQL database. Do not use a DSN. If you are replicating data larger than 500 MB to MySQL Aurora on Amazon RDS, include the MySQL parameter **max_allowed_packet** in the connection string and set its value to the maximum size of the data. See the example. **Example connection string** DRIVER=/usr/lib64/libmyodbc5.so;socket=/var/lib/mysql/mysql.sock;character-set-server=utf8;collation-server=utf8_general_ci;max_allowed_packet=2G;wait_timeout= 6000;Server=*servername*.amazonaws.com
`Enter the MySQL Administrator name :`	Enter the name of the MySQL Administrator. This user will perform the setup work on the SharePlex account and schema.
`Enter the password for the Administrator account :`	Enter the password of the Administrator.
`Enter the replication target database name :`	Enter the name of the MySQL database where you want to install the SharePlex objects.
`Database name` *`database`* `does not exist. Would you like to create it? [y] :`	If this prompt is displayed, the specified database does not exist. Press **Enter** to have the setup utility create it for you.
`Would you like to create a new SharePlex user [y]:`	Press **Enter** to accept the default to create a new SharePlex database user account, or enter **n** to use an existing account as the SharePlex database user.

Prompt	Response

`Enter the name of the new SharePlex user:`
 	
`Enter the name of the existing SharePlex user:`
	One of these prompts is displayed depending on whether you elected to create a new user or use an existing user. Enter the name of the SharePlex user.

`Enter the password for the SharePlex user :`	Enter the password of the SharePlex user account.
`Re-enter the password for the SharePlex user :`	Enter the SharePlex password again.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for MySQL database configuration

SharePlex User name: mysql29

Database name: mysql29

Target specification in SharePlex configuration: r.mysql29
```

Configure SharePlex to support MySQL database on Azure platform

You can configure SharePlex to support the MySQL database on the Azure platform using the below steps:

1. Add the DSN entry along with Socket path (SOCKET=) in the Odbc.ini file at /etc/odbc.ini and the …/var/odbc directory.
2. Copy the /etc/odbcinst.ini file to the …/var/odbc directory.
3. Run the MySQL setup utility.

   This process creates the SharePlex tables, login and user accounts needed to run SharePlex replication.

Table 7: Setup Prompts and Responses

Prompt	Response
`Enter the MySQL DSN name or connection string [] :`	Enter the MySQL DSN name or a connection string that connects to the MySQL database system. Example: **pslazpgdb02**
`Is DB hosted over Azure ? :`	Y
`Enter the Database hostname :`	Enter the Azure database host name. **NOTE:** MySQL Administrator name should be in the <username> format (Do not mention the hostname) Example: **pslazpgdb02.mysql.database.azure.com** If a user already exists for SharePlex then it should be entered in the <username@hostname> format. If a new user needs to be created for SharePlex then is should be entered in the <username> format.
`Enter the MySQL Administrator name :`	Enter the name of the MySQL Administrator. This user will perform the setup work on the SharePlex account and schema. Example: **mysql**
`Enter the password for the Administrator account:`	Enter the password of the Administrator. **NOTE:** User 'mysql@pslazpgdb02.mysql.database.azure.com' got the superuser role.
`Enter the replication target database name :`	Enter the target database name on which replication needs to be performed.

Prompt	Response
	Example: **sp_mk**

`Would you like to create a new SharePlex user?`
 	
`Enter the name of the existing SharePlex user:`
	N (If you want to create a user, enter “y”)
	
	Enter the name of the SharePlex user.

`Enter the password of the SharePlex user :`	Enter the password of the SharePlex user account.
`Re-enter the password for the SharePlex user :`	This prompt is only shown if you created a new user. Enter the SharePlex password again.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for MySQL database configuration

SharePlex User name: mysql@pslazpgdb02.mysql.database.azure.com

Database name: sp_mk

The data source identifier in the SharePlex configuration is 'r.sp_mk' –
```

Database setup for Oracle

Overview

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 section before running this 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 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. ◦ The setup steps provided for Oracle are also applicable for setting up Oracle remote Post.

System Type	When to run Oracle Setup
	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
Database with or without ASM	Bequeath
Database with or without ASM	TNS alias (A TNS login is specified for both the database and the ASM instance.)
PDB with ASM	TNS alias for the PDB and either TNS or bequeath for the ASM instance.
Amazon RDS	TNS alias

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;**

Privileges required by SharePlex to perform replication

The following is a list of privileges required by SharePlex to perform replication. If you revoke the DBA role for the SharePlex database user, ensure these privileges are granted.

Privilege	Details
CREATE SESSION	
ALTER SESSION	
ALTER SYSTEM	
RESOURCE role	RESOURCE role privileges required by SharePlex: • CREATE TRIGGER • CREATE SEQUENCE • CREATE TYPE • CREATE PROCEDURE • CREATE CLUSTER • CREATE OPERATOR • CREATE INDEXTYPE • CREATE TABLE
SELECT ANY DICTIONARY	

Privilege	Details
SELECT ANY TABLE	
INSERT ANY TABLE	
UPDATE ANY TABLE	
DELETE ANY TABLE	
UNLIMITED TABLESPACE	
CREATE ANY TABLE	
DROP ANY TABLE	
LOCK ANY TABLE	
EXECUTE ON DBMS_FLASHBACK	
SELECT ANY TRANSACTION	
CREATE ANY INDEX	
DROP ANY INDEX	
ALTER ANY TABLE	
ALTER ANY INDEX	
CREATE ANY VIEW	
ALTER ANY SEQUENCE	
EXP_FULL_DATABASE role	
IMP_FULL_DATABASE role	
SELECT ON SYS.ENC$	
SELECT ON SYS.USER$	

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

IMPORTANT! The Oracle instance must be open before this procedure is performed.

Perform the following steps to run database setup for Oracle:

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.
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 8: 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)
`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.

Prompt	Response

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 you changed the schema, copy the objects from the old schema to the new one to preserve the replication

Prompt	Response
	environment.
`Do you want to enable replication of tables with TDE?`	Press **N** since TDE is not supported for RDS. Press **Y** to capture data encrypted by the Oracle TDE wallet.
`Are the TDE keys stored in the Oracle wallet?`	Press **Y** if you have a valid Oracle path. Press N if the Oracle wallet key is stored on external storage that is HSM. Enter the fully qualified pathname of the HSM PKCS11 library, including the library name: []
`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.

Prompt	Response
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 value of 1. Also if you are using a BEQUEATH connection, the user running SharePlex must be a member of the OSDBA group.
NOTE: If you selected to use a BEQUEATH connection to connect to ASM, the database setup is complete. Continue to Database setup for Oracle. 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.

Run database setup for Oracle remote capture

IMPORTANT! The Oracle instance must be open before this procedure is performed.

Perform the following steps to run database setup for Oracle remote capture:

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. Set the SP_OCT_OLOG_USE_OCI parameter to 1 and restart the Capture process.
3. Shut down any SharePlex processes that are running, including **sp_cop**.
4. Run the Database Setup program from the command prompt of the operating system, using the full path from the SharePlex **bin** subdirectory.
5. 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.

6. For connection type, select **Oracle**.
7. Refer to the following table for the prompts and responses to configure SharePlex correctly for the desired connection type.

Table 9: Setup prompts and response

Prompt	Response
`Will SharePlex install be using a BEQUEATH connection? (Entering 'n' implies a SQL*net connection):`	Press **N** to use a TNS alias connection.
`Are you configuring SharePlex for an AWS RDS database?`	Press **N** if you are not configuring SharePlex for an Oracle database on RDS.
One of the following prompts is shown: • If you selected BEQUEATH= Y: `Enter the Oracle SID 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.

Prompt	Response
• If you selected BEQUEATH = N: `Enter the TNS alias for which SharePlex should be installed:`	

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.

`Enter the Local Oracle Client Library directory:`	Enter the Oracle Client Home directory path. **Example:** /u01/app/oracle/product/19.0.0/clienthome_1
`Enter the TNS alias for which SharePlex should be installed (Blank to abort) :`	Enter the name of the TNS alias
`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 you changed the SharePlex schema, copy the SharePlex objects from the old schema to the new one to preserve the replication environment.

Prompt	Response
`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.
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 value of 1. Also if you are using a BEQUEATH connection, the user running SharePlex must be a member of the OSDBA group.
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**.

NOTE: While running the **ora_cleansp** utility, enter the TNS alias provided while performing **ora_setup**.

Run database setup for Oracle in silent mode

IMPORTANT! The Oracle instance must be open before this procedure is performed.

Perform the following steps to run database setup for Oracle in silent mode:

1. (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. Use the following command to run Ora_setup in the background in silent mode:-
   `nohup ./ora_setup -f <Full_file_path of ora_setup.config> > SilentOra.out 2>&1`
4. For more details on how to configure the `Ora_setup.config` file, refer to the `ora_setup.config` file at the below location:
   `/<product directory>/install/ora_setup.config`

NOTE: After the setup is completed successfully, all of the password fields will be automatically removed from the config file for security reasons.

Database setup for SQL Server

Overview

Run the Database Setup utility for SQL Server (**mss_setup**) on a Microsoft SQL Server system to establish SharePlex as a SQL Server database user.

This utility creates the following:

- A SharePlex user account with the db_owner role
- Tables and indexes for use by SharePlex and owned by the SharePlex user in a database of your choosing
- A default database connection.

Supported databases

For supported SQL Server platforms and versions, see the SharePlex Release Notes.

Guidelines for use

- A *system* DSN (data source name) must exist for the SQL Server database. SharePlex Post uses the DSN to connect to the database through ODBC.
- Run the Database Setup utility on all SQL Server instances in the SharePlex replication configuration.
- Within a cluster, run the Database Setup utility on the node to which the shared disk that contains the variable-data directory is mounted.
- For consolidated replication, run the Database Setup utility for each variable-data directory.

Required privileges

Review the following requirements to ensure that the setup succeeds.

- The Database Setup utility must be run as a SQL Server System Administrator in order to grant SharePlex the required privileges to operate on the database and to create the SharePlex database account and objects.
- The sysadmin role is required to run **mss_setup**, but it is not required by the SharePlex database account used to run SharePlex replication (the SharePlex User). The db_owner role is required by the SharePlex User account and it is granted by **mss_setup** (for source and target).
- (Cloud installations) Common restrictions on privileges in cloud-hosted database services make it difficult for the setup utility to succeed in every possible type of scenario. To ensure that the database setup succeeds, *only* use the setup utility for the following purposes: To do a *first-time* database setup with a *new* SharePlex user, or, to *modify* an existing SharePlex user that either owns the database or has access to it. (Cloud installation is not supported while replicating data from PostgreSQL to SQL Server)

Run database setup for SQL server

Perform the following steps to run database setup for SQL server:

1. Shut down any running SharePlex processes and **sp_cop** on the SQL Server system.
2. Run the **mss_setup** program from the **bin** subdirectory of the SharePlex product directory.

   IMPORTANT! If you installed the SharePlex instance 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>mss_setup-p9400
   ```

Table 10: Setup prompts and responses

Prompt	Response
`Enter the Microsoft SQL Server DSN name [] :`	Enter the data source name (DSN) that connects to SQL Server. Make certain the DSN is a system DSN, not a user DSN.
`Enter the Microsoft SQL Server Administrator name :`	Enter the name of the SQL Server Administrator. This user will perform the setup work on the SharePlex account and schema.
`Enter the password for the Administrator account :`	Enter the password of the Administrator.
`Enter the database name:`	Enter the name of the database where you want to install the SharePlex objects.
`Database name` *`database`* `does not exist. Would you like to create it? [y] :`	If this prompt is displayed, the specified database does not exist. Press **Enter** to have the setup utility create it for you.
`Would you like to create a new SharePlex login [y]:`	Press **Enter** to accept the default to create a new SharePlex database user account, or enter **n** to use an existing account as the SharePlex database user.

`Enter the name of the existing SharePlex login:`
 	
`Enter the name of the new SharePlex login:`
	One of these prompts is displayed depending on whether you elected to create a new user or use an existing user. Enter the name of the SharePlex user.

`Enter the password for` *`login`*`:`	Enter the password of the SharePlex user account.
`Re-enter the password for` *`login`* `:`	Enter the SharePlex password again.

Prompt	Response
`Will this database be used as a source?`	Accept the default of **n** if the database will only be a target. Enter **y** if this database will be a source database for SharePlex. A response of **y** prompts the setup to prepare the database for data capture and installs the SharePlex account and objects.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for Microsoft SQL Server database configuration

SharePlex User name: splex

Database name: db1

Target specification in SharePlex configuration: r.db1
```

Configure remote posting from a Linux to a SQL Server database

To replicate data from a Linux server to a SQL server database, follow the steps below:

1. Install the latest **SQL Server ODBC** drivers on Linux.
   By default, the ODBC drivers will be installed at `/opt/microsoft/msodbcsql18/lib64`.
2. Modify the **odbcinst.ini** and **odbc.ini** files in the `<var_dir>/odbc` directory.
3. Run the **mss_setup** program from the **bin** subdirectory of the SharePlex product directory.
4. Provide other details when it prompts you, like the database name, username, password, etc. For more information, see Run Database Setup for SQL Server.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for Microsoft SQL Server database configuration

SharePlex User name: splex

Database name: db1

Target specification in SharePlex configuration: r.db1
```

Database setup for Snowflake

Snowflake configuration prerequisites:

- If any user has used Snowflake setup utility of a version prior to 11.4, they should first perform the following cleanup steps:

  1. Drop the role 'SHAREPLEX_ROLE' from the Snowflake account using the owner role. For example, if 'SHAREPLEX_ROLE' was created with a user having the 'ACCOUNTADMIN' role, execute the following query using the same privileges:

     ```
     drop role "SHAREPLEX_ROLE";
     ```

  2. Drop the table 'SHAREPLEX_OPEN_TRANS' from the target DB.SCHEMA. Refer to the query below:

     ```
     drop table SHAREPLEX_OPEN_TRANS;
     ```

     Re-run the Snowflake setup utility now, following the guidelines below. 'SHAREPLEX_ROLE' will be re-created correctly by the setup using the 'USERADMIN' role."

- Configure DSN in the ODBC directory as per your requirement (for password based authentication or for RSA based authentication).
- Enter the warehouse name that SharePlex is using in the odbc.ini file. For more information, see Guidelines for use section.
- SharePlex setup requires one Admin user to operate, below is the query is reference. Create an Admin user as shown in the following query::
  `create user <user_name> identified by '<user_password>';`
- Grant the **SYSADMIN** and **USERADMIN** roles to the Admin user using the following queries:
  `grant role SYSADMIN to user <user_name>;`
  `grant role USERADMIN to user <user_name>;`
- `Snowflake_setup` internally uses the USERADMIN role to create a new SharePlex role to be used in replication and user-related tasks. The SYSADMIN role is used to grant access to databases and warehouses.
- If the SYSADMIN role does not have access to the intended database and warehouse, provide access using the following queries:

  ```
  grant OWNERSHIP on database <database name> to SYSADMIN with grant option;

  grant USAGE on warehouse <warehouse_name> to role SYSADMIN with grant option;

  grant OPERATE on warehouse <warehouse_name> to role SYSADMIN with grant option;
  ```

- The required privileges will be granted to the SharePlex role (`SHAREPLEX_ROLE`), which will then be assigned to the SharePlex user provided as input.

Overview

Use the Database Setup utility for Snowflake (**snowflake _setup**) to establish SharePlex as a Snowflake user and create the required SharePlex database objects.

This setup utility:

- Create a new role `SHAREPLEX_ROLE` if it does not exist.
- Creates a new user or uses an existing user
- Creates SharePlex internal tables and objects

It is recommended that you review all of the content in this topic before running the setup utility.

Guidelines for use

- Run the `Snowflake_Setup` utility on the target Snowflake instance in the SharePlex replication configuration.
- Within a server cluster, run the Snowflake Setup utility on the node to which the shared disk that contains the SharePlex variable-data directory is mounted.
- For consolidated replication, run the Snowflake Setup utility for each variable-data directory.
- Provide a DSN (Data Source Name) as mentioned below:
    - If you have a DSN defined, and you want to use it for the SharePlex connection, copy or link the ODBC files in which that DSN is defined (odbc.ini and odbcinst.ini) to the odbc subdirectory of the SharePlex variable-data directory. This prevents connection errors when the SharePlex processes connect to the database.
    - If you do not have a DSN defined but want to use one, you can create it in the template files provided in the odbc subdirectory.

For an example of the sample `odbc.ini` and `odbcinst.ini` files, see the example below.

Sample of an ODBC file in a password based authentication:

```
[sp_snowflake]
Description=SnowflakeDB
Driver=SnowflakeDSIIDriver
Locale=en-US
SERVER=GR22806.east-us-2.azure.snowflakecomputing.com
PORT=443
SSL=on
ACCOUNT=GR22806
WAREHOUSE=WH_SHAREPLEX
~
```

Sample of an ODBC file for in an RSA based authentication:

```
[sp_snowflake_key]
Description=SnowflakeDB
Driver=SnowflakeDSIIDriver
Locale=en-US
SERVER=GR22806.east-us-2.azure.snowflakecomputing.com
PORT=443
SSL=on
ACCOUNT=GR22806
AUTHENTICATOR=SNOWFLAKE_JWT
PRIV_KEY_FILE=/splex/aparopka/ssl_key/rsa_key.p8
WAREHOUSE=WH_SHAREPLEX
```

Privileges granted to SharePlex

The database setup utility creates the `SHAREPLEX_ROLE` and assigns it to SharePlex users. SharePlex assigns the following privileges to the `SharePlex_role`:

Privilege types	Privileges
Database privileges	• SELECT,INSERT,UPDATE,DELETE,TRUNCATE ON FUTURE TABLES IN DATABASE • USAGE ON DATABASE • ALL PRIVILEGES ON FUTURE SCHEMAS IN DATABASE
Schema	ALL PRIVILEGES
Warehouse	Usage

Run database setup for Snowflake

IMPORTANT! The Snowflake instance must be open before this procedure is performed.

Perform the following steps to run database setup for Snowflake:

1. (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 Snowflake database Setup command (`./snowflake_setup`) using password or RSA based admin user.
4. Refer to the following table for the prompts and responses to configure SharePlex correctly for the desired connection type.

Table 11: Setup prompts and response for password based authentication

Prompt	Response
Enter the snowflake DSN name [ ] :	Enter a DSN name. **Example:** `sp_snowflake_5`
Is authentication based on RSA public key? [n] :	For password based authentication, enter "n"
Enter the Snowflake administrator name :	Enter an administrator name.
Enter the password for the administrator account:	Enter the administrator account password.
Enter the replication target database name :	Enter the name of the Snowflake database where you want to install the SharePlex objects. **Example:** shareplex_test
Database name database does not exist. Would you like to create it? [y] :	If this prompt is displayed, the specified database does not exist. Press **Enter** to have the setup utility create it for you.
Enter the DB schema name:	Enter the DB schema name. **Example:** shareplex_test_5
DB schema name does not exist. Would you like to create it? [y] :	If this prompt is displayed, the specified DB schema name does not exist. Press **Enter** to have the setup utility create it for you.
Would you like to create a new SharePlex user? [y/n] :	Press **Enter** to accept the default to create a new SharePlex database user account in the specified database, or enter **n** to use an existing SharePlex account.
Enter the name of the new SharePlex user: Enter the name of the existing SharePlex user:	One of these prompts is displayed depending on whether you elected to create a new user or use an existing user. Enter the name of the SharePlex user.
Enter the password for the new SharePlex user:	Enter a password for the new SharePlex user.
Re-enter the password for the new SharePlex user:	This prompt is only shown if you created a new user. Enter the SharePlex password again.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for Snowflake database configuration

SharePlex user name: amore2

Database name: SHAREPLEX_TEST

- - The datasource identifier in the SharePlex configuration is 'r.SHAREPLEX_
TEST' - -
```

5. The current version of SharePlex setup does not grant privileges to existing objects in the database. After successfully completing the Snowflake setup, run the following queries for smooth replication:

```
grant SELECT,INSERT,UPDATE,DELETE,TRUNCATE ON ALL TABLES IN DATABASE <DataBase name> to role SHAREPLEX_ROLE;

grant ALL PRIVILEGES ON ALL SCHEMAS IN DATABASE <DataBase name> to role SHAREPLEX_ROLE;
```

.

Table 12: Setup prompts and response for RSA based authentication

Prompt	Response
Enter the snowflake DSN name [ ] :	Enter a DSN name. **Example:** `sp_snowflake_5`
Is authentication based on RSA public key? [y] :	For password based authentication, enter "y". **NOTE:** Make sure that the RSA key has been configured for Admin user.
Enter the snowflake administrator name :	Enter an administrator name.
Enter the replication target database name :	Enter the name of the Snowflake database where you want to install the SharePlex objects. **Example:** shareplex_test
Database name database does not exist. Would you like to create it? [y] :	If this prompt is displayed, the specified database does not exist. Press **Enter** to have the setup utility create it for you.
Enter the DB schema name:	Enter the DB schema name. **Example:** shareplex_test_5
DB schema name does not exist. Would you like to create it? [y] :	If this prompt is displayed, the specified database schema does not exist. Press **Enter** to have the setup utility create it for you.
Would you like to create a new SharePlex user? [y/n] :	Press Enter to accept the default to create a new RSA user account in the specified database, or enter **n** to use an existing SharePlex account.
Enter the name of the new SharePlex user: Enter the name of the existing SharePlex user:	One of these prompts is displayed depending on whether you elected to create a new user or use an existing user. Enter the name of the SharePlex user.
Enter path for RSA public key for configuration of New User.	Enter the path for the RSA public key. (This field is applicable only for a new user) **NOTE:** For existing user account, make sure that the RSA key has been configured for Admin user.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for Snowflake database configuration

SharePlex user name: amore2

Database name: PROVIDENCE

- - The datasource identifier in the SharePlex configuration is 'r.SHAREPLEX_
TEST' - -
```

Solve Installation Problems for Oracle

Contents

Solve database setup problems for Oracle

Solve database connection problems

Solve SharePlex startup problems for Oracle

How to find the ORACLE_SID and ORACLE_HOME

Overview

This chapter reviews some common problems that you could experience when installing or running SharePlex for the first time after installation.

Sometimes there are special installation instructions that supersede or supplement certain instructions in this manual. In addition, there can be known issues for this version that you should be aware of during or after installation. Please read the Release Notes for the version of SharePlex that you are installing before you begin the installation process.

Solve database setup problems for Oracle

This section helps you diagnose problems that are associated with the SharePlex database account and connection information that was created with the Database Setup utility when SharePlex was installed on the system.

NOTE: For more information about Database Setup, see Database Setup Utilities in the SharePlex Reference Guide.

If the issue you are experiencing is not listed in this documentation, search the SharePlex Knowledge Base at:

https://support.quest.com.

The Knowledge Base provides filtering options and links to other resources that can help you use and troubleshoot SharePlex.

Oracle setup issues

Problem	Description	Solution
Incorrect ORACLE_SID and/or ORACLE_HOME	If SharePlex cannot interact with Oracle, it might be using the wrong ORACLE_SID and/or ORACLE_HOME.	1. See How to find the ORACLE_SID and ORACLE_HOME on page 164 to determine the Oracle values. 2. Rerun the Database Setup utility. For more information, see Database Setup Utilities in the SharePlex Installation and Setup Guide.

Problem	Description	Solution
Insufficient database privileges	If the Database Setup utility fails, the person who runs it may not have the correct privileges	For more information, see Database Setup Utilities on page 130.
Asterisk as the ORACLE_SID entry	Sometimes, the **oratab** file has an * (asterisk) symbol instead of a value for the ORACLE_SID.	Ensure that a valid ORACLE_SID is in the **oratab** file, and then try running the database setup again.
More than one **oratab** file (Sun Solaris)	On Solaris systems, the **oratab** file is typically located in the **/var/opt/oracle** directory, but because other platforms store the **oratab** file in the **/etc** directory, there could be a second **oratab** in the **/etc** directory.	Either move, rename or delete the secondary **oratab** file, and then try running the database setup again.
Oracle not running	Oracle must be running and the instance must be open while you run the Database Setup utility. The utility accesses Oracle to establish SharePlex as a user and install its internal tables.	Start Oracle and open the instance.
sp_cop is running	The SharePlex **sp_cop** process cannot be running while you are running the Database Setup utility.	If it is running, shut it down using the **shutdown** command in **sp_ctrl.** Run **sp_ctrl** from the bin sub-directory in the SharePlex product directory.
Oracle library location not correct	On Unix and Linux systems, SharePlex expects the Oracle library to be in the **$ORACLE_ HOME/lib or $ORACLE_HOME/lib32** directory. In some environments, the Oracle library has a different name than what SharePlex expects it to be, or it is installed in a different location than expected (or both). In that case, you will see an error message when you attempt to run the Database Setup utility.	Install the appropriate library from Oracle and then re-start SharePlex (if it is stopped). SharePlex will link to the correct library from that point forward.
ld.so.1: sqlplus: fatal: libsunmath.so.1: can't open file: errno=2” error	On Unix and Linux systems, this error indicates that SharePlex cannot find the **libsunmath** and **libshareplex** libraries, even though the link exists in the proper place.	**You can use either of these solutions:** • Create a softlink for **$ORACLE_ HOME/lib/libsunmath.so.1** in the **/usr/lib** directory. or... • In the **ECXpert/config/bdg.ini** file in the [**DB_ENV**] section add the following line: LD_LIBRARYPATH=*full oracle home path*/lib
Wrong user-id	To run Database Setup on Unix and Linux systems, the **set-user-id** for the Oracle software	Set the correct values for **set-user-id**.

Problem	Description	Solution
	need to be **-rwsr-s--x.** Those permissions allow non-Oracle users to log into SQL*Plus.	

Solve database connection problems

Did you verify the credentials?

If SharePlex cannot connect to a source or target database, you can view the login credentials that are being used for that database by using the **connection** command with the **show** option. For example:

```
sp_ctrl> connection o.mydb show
```

You can also view connection settings in the **connections.yaml** file, which is stored in the **data** sub-directory of the SharePlex variable-data directory. If there are no settings in this file, it means that the SharePlex database setup procedure was not performed on this database.

To view the correct database setup procedure for the database, see SharePlex Reference Guide.

You can use the **connection** command to update connection properties. For more information, see the SharePlex Reference Guide.

Did you assign a DBA role to the SharePlex Oracle user?

The SharePlex Oracle user requires a DBA role with unlimited privileges. The SharePlex user is created with the default Oracle profile under the assumption that the profile has the unlimited resource privileges assigned by Oracle as the default. If SharePlex is unable to interact with Oracle, check to see if the default was changed. If so, assign SharePlex a DBA role with unlimited privileges for all definitions.

Solve SharePlex startup problems for Oracle

Is this a cluster environment?

In order for the SharePlex processes to issue name lookups and migrate properly in a clustered environment (where a package name supersedes the local system name), the SP_SYS_HOST_NAME parameter must be set to the correct package name. In addition, the host name set by this parameter must be the same on all members of the cluster so that the name can bind to a socket and the **/etc/hosts** file or nameserver can correctly map the parameter's value to the correct IP address.

The **sp_cop** program should only be started through the cluster management software.

How to find the ORACLE_SID and ORACLE_HOME

When setting up SharePlex to work with an Oracle database, you provide the ORACLE_SID and then SharePlex gets the ORACLE_HOME from the **oratab** file on Unix/Linux. Both values are stored in the SharePlex environment. SharePlex uses the Oracle libraries that are in the location specified with ORACLE_HOME.

To determine the ORACLE_SID and ORACLE_HOME being used by SharePlex:

Issue the **orainfo** command in **sp_ctrl**.

```
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
```

To determine the default ORACLE_SID and ORACLE_HOME on UNIX and Linux:

On most Unix and Linux systems the **oratab** file is under **/etc/oratab**. On Oracle Solaris systems, it is under **/var/opt/oracle**, but sometimes there is an **oratab** file in the **/etc** directory as well.

The entry in the file looks like the following example:

```
qa12:/qa/oracle/ora12/app/oracle/product/12.0
```

In the example, `qa12` is the ORACLE_SID and `/qa/oracle/ora12/app/oracle/product/12.0` is the ORACLE_HOME.

2

Installing and Setting up SharePlex on a PostgreSQL Database as Source and Service

The following installation and setup configurations apply to the PostgreSQL database as a source and service.

Contents

- SharePlex Pre-installation Checklist for PostgreSQL
- Download the SharePlex installer for PostgreSQL
- Install SharePlex on Linux for PostgreSQL as a Source
- Set up Replication from PostgreSQL to Supported Target Types
- Installation and Setup for Cloud-Hosted Databases for PostgreSQL
- Installation and Setup for Remote Capture for PostgreSQL
- Install SharePlex on PostgreSQL High Availability Cluster
- Configure SharePlex on PostgreSQL Azure Flexible Server with High Availability Using Logical Replication
- Generic SharePlex Demonstration for PostgreSQL
- Advanced SharePlex Demonstrations for PostgreSQL
- Database Setup for PostgreSQL
- Database Setup for PostgreSQL Database as a Service
- Installation of pg_hint_plan extension
- Solve Installation Problems for PostgreSQL

SharePlex Pre-installation Checklist for PostgreSQL

Contents

- Network checklist for PostgreSQL
- Installer checklist for PostgreSQL
- Linux system checklist for PostgreSQL
- Amazon EC2 checklist

Overview

Review and satisfy all of the requirements in this checklist before installing SharePlex.

NOTE: The requirements in this checklist apply to all source and target systems where SharePlex will be installed unless otherwise noted.

Network checklist for PostgreSQL

Requirement	Completed? (Y/N)

Ensure that SharePlex can resolve host names.
 	
If you find that SharePlex cannot connect to a host, try mapping the host name to an alphanumeric alias in the following locations:
 	
Unix and Linux: Local **/etc/hosts** file
 	
In these files, put each entry on an individual line. The following is an example, where **sysA** and **sysB** are aliases:
 	
`111.22.33.44  sysA.company.com  sysA  # source system`
`55.66.77.88  sysB.company.com  sysB  # target system`	

Verify the SharePlex port number. By default SharePlex uses the port number 2100 (hex equivalent is 834) for both TCP/IP and UDP. If port 2100 is available to SharePlex, no further action is needed. You will need to enter the SharePlex port number during the installation procedure, at which time you can specify a different port number if needed. **IMPORTANT!** The SharePlex port number must be the same one on all machines in the replication configuration so that they can communicate through TCP/IP connections. Make certain the SharePlex port number is open for TCP/IP on the firewall.	
Verify TCP/IP settings for PostgreSQL: SharePlex replicates over any TCP/IP network connection. Typically, if all of the tables in Postgres database are being replicated, SharePlex processes about 54 percent of the WAL file data for replication, the remainder being mostly for internal data used by the Postgres engine for maintenance. So if WAL file is of 16 MB size (default setting), and all user tables in DB are in replication, then SharePlex will consume approximately 8.6 MB (16 x 0.54) of data for replication. The following is a formula for determining bandwidth. (size of a WAL file) x (number of WAL file switches in 1 hour) x 0.54 = volume of data per hour. For example, if WAL file is of 16 MB in size, and they switch 5 times an hour, SharePlex will process approximately 43.2 MB each hour for replication: 16 x 5 x 0.54 = 43.2 MB/hour	

Installer checklist for PostgreSQL

Requirement	Completed? (Y/N)
Assign a directory to store the downloaded SharePlex installation package. This directory requires approximately the following disk space: • Linux: 200 MB It can be removed after SharePlex is installed.	
Plan the SharePlex product directory. You can create a directory for the SharePlex software files or let the SharePlex installer create it. **This directory requires approximately the following disk space:** • Linux: 600 MB **Install this directory on the following:** • Linux: a separate filesystem from the one that contains the database. Do not install SharePlex on a raw device.	
Plan the SharePlex variable-data (working) directory This directory is installed by the SharePlex installer with a name of your choosing. It contains the working data and varies greatly in size in correlation to the volume of data being generated. Install this directory on a separate file system from the one that contains the database, but not on a raw device. **To estimate the required disk space:** 1. Estimate the longest time that a replication outage can be tolerated. 2. Use the following formula to estimate the amount of data SharePlex would replicate during that amount of time. [size of a WAL log] x [number of log switches per hour] x .54 x [number of hoursdowntime] = required disk space For example: [500 MB WAL log] x [5 switches per hour] x [.54] x [8 hours] = 10.54 GB disk space To replicate data from more than one database on a system, use a variable-data directory for each one. Ideally they should be on different file systems. Do not install the variable-data directory within the SharePlex product directory. Both directories contain identically named files, and SharePlex utilities that clean up the environment (if this becomes necessary) could remove the wrong files. You can install both	

Requirement	Completed? (Y/N)
directories under one parent directory if desired. **NOTE:** Always monitor disk usage when there is an active SharePlex configuration, especially when there are peaks in transaction activity.	
Create the SharePlex security groups. SharePlex provides three security groups to enable access control through **sp_ctrl**. On Linux, unless you install SharePlex as a root user, the SharePlex Administrator user and the SharePlexPostgreSQL **admin** group must exist prior to installation. For more information, see Assign SharePlex users to security groups in the SharePlex Admin guide. **NOTE:** If you install as root, you are prompted by the installer to create these groups.	
Get a valid SharePlex license key. You must have a valid permanent, term, or trial license key from Quest to run SharePlex. SharePlex licenses have validity and usage limits according to specific platforms. For example, you must have a PostgreSQL license to use a PostgreSQL database and a Kafka license to use the Kafka platform. Additionally, SharePlex supports multiple keys for situations where customers need two platforms on one server. For example, if a user is replicating data from an Oracle source to a Kafka target, where one SharePlex instance is serving as both the source and target, the SharePlex server would require both Oracle and Kafka licenses. **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.	

Linux system checklist for PostgreSQL

Requirement	Completed? (Y/N)
Confirm that the platform is supported. Review the SharePlex Release Notes to make certain your operating system is supported.	
Allocate at least 4 GB of memory for SharePlex processes. Plan for per-process memory up to 256 MB. This recommendation enables the Post and Read processes to allocate larger sets of memory when necessary.	
Set the number of semaphores per process. Semaphores help ensure the stability of the SharePlex processes. The required SharePlex settings depend on the platform, as follows: **Red Hat Linux:** • semmni*: 70 • semmns*: 255 • semmnu: 255 • semmsl: 128 • semopm: 64 • semume: 255 • shmmax: 60 MB • shmmin: 1MB • shmmni: 100 • shmseg: 26 *These are additive. Add the database minimum values to the SharePlex minimum values to determine the correct setting. An alternative is to set the value to the number of queues you will be using plus 2.	
Set the ulimit (number of system file descriptors) to as close to 1024 as possible. The ulimit can be set either as a system *hard limit* or a session-based *soft limit*, as follows: • Set a hard limit: (Recommended) A root user and system restart are required to change the hard limit, but the value remains fixed at the correct level to support SharePlex. Consult your System Administrator for assistance. • Set a soft limit: A soft limit setting stays in effect only for the duration of the **sp_cop** session for which it was set, and then it reverts back to a default value that may be lower than the hard limit and too low for SharePlex.	

Requirement	Completed? (Y/N)
Set soft and hard limits for the SharePlex user Set a soft limit and a hard limit for **nproc** and **nofile** for the SharePlex O/S user in the **/etc/security/limits.conf** file. • **shareplex O/S user soft nproc 2048** • **shareplex O/S user hard nproc 16384** • **shareplex O/S user soft nofile 1024** • **shareplex O/S user hard nofile 65536** As an alternative, you can simply use the setting for the PostgreSQL O/S user.	
Set core file parameters. • Set the system core dump block size as large as system resources can accommodate, at minimum 1.5 million blocks. The default is usually 0. Core files help Quest support representatives resolve SharePlex support cases. Higher size settings ensure that enough data is captured to be useful. • Set the core file output location to the **dump** sub-directory of the SharePlex variable-data directory. • Set the naming convention of core files to either **core** or **core.***pid*. **NOTE:** SharePlex renames all core files named **core** to **core.***pid*, except for those generated by **sp_cop**. If these requirements are not met, the SharePlex event log might report that a core file was not generated, even though a file exists.	
Install the ksh shell. Install the **ksh** shell before you install SharePlex. The SharePlex monitoring scripts and other features required this shell. A version of **ksh** called **pdksh** is included with the Red Hat Linux builds. Refer to the Red Hat Linux documentation for more information.	
Install Native POSIX Threading Library (NPTL) Quest recommends using the Native POSIX Threading Library (NPTL) on Linux. NPTL is faster and behaves more like other Unix operating systems than LinuxThreads. Although LinuxThreads can be enabled per process by using the LD_ASSUME_KERNEL environment variable, setting that variable adversely affects the performance of SharePlex. If LD_ASSUME_KERNEL is employed, use a setting of 2.4.21.	

Amazon EC2 checklist

Requirement	Completed? (Y/N)
Install on multiple EBS volumes • Install the database and SharePlex on Amazon Elastic Block Storage (EBS). An EBS volume is persistent storage, whereas the default Amazon storage is non-persistent and data is lost when the instance shuts down. EBS volumes also provide better performance. Minimum size for a volume is 1GB. • To optimize disk performance, create multiple EBS volumes and combine them by using software RAID. According to benchmarks, the optimal number of EBS volumes is 8.	
Assign Elastic IP addresses Amazon Elastic IP addresses are static, which satisfy SharePlex requirements. An Elastic IP must be created and assigned to both the source and target machines that will be used with SharePlex.	

Download the SharePlex installer for PostgreSQL

SharePlex Installer

There are different installers for SharePlex based on the type of operating system. This topic helps you understand the differences and the naming conventions used.

Linux

The SharePlex installer on Linux is a self-extracting installation file with the extension **.tpm**.

`SharePlex-`*`release#-build#-platform-version-chipset`*`.tpm`

The installer creates a temporary target directory, within the current directory, for extraction. This temporary target directory is removed upon installation completion. You can extract the files to a file system that is separate from the SharePlex installation location by using the -t option when running the **.tpm** file.

Where to get SharePlex installers

Download the SharePlex installation package that matches the operating system you are using.

Additionally, download any SharePlex patches, so that you can install them after you install the base software.

For more information on how to download the SharePlex patches, see Download the SharePlex patches.

Install SharePlex on Linux for PostgreSQL as a Source

You can run the SharePlex installer for PostgreSQL on Linux in an Interactive or Unattended mode. For more information on the steps to run the installer, see Install SharePlex on Linux/Unix for open target databases.

Set up Replication from PostgreSQL to Supported Target Types

Contents

Set up replication from PostgreSQL to PostgreSQL

Set up replication from PostgreSQL to Oracle

Set up replication from PostgreSQL to Kafka

Set up replication from PostgreSQL to SQL Server

About these instructions

This chapter contains instructions for configuring SharePlex to replicate from PostgreSQL to a different type of target. This is known as heterogeneous replication.

These instructions highlight specific tasks that are pertinent to the flow of data between source and target. Refer to other topics in the SharePlex documentation as needed to complete the configuration, deploy any optional features that apply, and monitor and maintain the environment.

For additional information, see:

- For the SharePlex-supported datastores, data types and operations that are supported by SharePlex, see the "System Requirements " section of SharePlex Release Notes.
- For additional configuration options, activation steps, and monitoring information, see SharePlex Administration Guide.
- For reference documentation on SharePlex commands, parameters and utilities, see SharePlex Reference Guide.

Set up replication from PostgreSQL to PostgreSQL

NOTE: These instructions apply to all implementations of PostgreSQL unless otherwise noted.

Overview

SharePlex can post replicated PostgreSQL data to a PostgreSQL target database through an Open Database Connectivity (ODBC) interface. SharePlex supports all implementations of the PostgreSQL open-source database.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Configure SharePlex on the source and target

Set up SharePlex and the database on the PostgreSQL source and target system as follows.

Configure replication

On the source, create a SharePlex configuration file that specifies capture and routing information.

Datasource:r.*source_DB*

src_schema.table *tgt_schema.table* *host***@r.***database_name*

where:

- *source_DB is the database name of the source database.*
- *src_schema.table is the schema and name of the source table.*
- *tgt_schema.table is the schema and name of the target table.**
- *host* is the name of the target system.
- *database_name* is the name of the target database.

*** IMPORTANT!**
Target schema or table name defined in the database is case sensitive, be certain to:

- Type the name in the correct case.
- Enclose the name in quotation marks, for example "MySchema"."MyTable".

NOTE: This is a basic one-source, one-target configuration using no additional SharePlex configuration features. See "Configure SharePlex to replicate data" in the SharePlex Administration Guide for important information about creating a configuration file and for additional setup instructions for more complex replication scenarios.

Source configuration example

The following configuration file replicates table `HR.Emp` from PostgreSQL instance `dbname112` to target table `region1.emp` in database `mydb` on target system `hostB`. The source table is case-sensitive.

```
Datasource:r.dbname112

HR."Emp"  region1.emp  hostB@r.mydb
```

Set up replication from PostgreSQL to Oracle

Overview

SharePlex can post replicated PostgreSQL data to an Oracle target database through an Open Database Connectivity (ODBC) interface. SharePlex supports all implementations of the PostgreSQL open-source database.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Configure SharePlex on the source and target

Set up SharePlex and the database on the PostgreSQL source and an Oracle target system as follows.

Configure replication

On the source, create a SharePlex configuration file that specifies capture and routing information.

Datasource:r.dbname		
src_schema.table	*tgt_owner.table*	*host***@o.***database_name*

where:

- *source_DB is the database name of the source database.*
- *src_schema.table is the schema and name of the source table.*
- *tgt_owner.table is the schema and name of the target table.**
- *host* is the name of the target system.
- *database_name* is the name of the target database.

*** IMPORTANT!**
Target schema or table name defined in the database is case sensitive, be certain to:

- Type the name in the correct case.
- Enclose the name in quotation marks, for example "MySchema"."MyTable".

NOTE: This is a basic one-source, one-target configuration using no additional SharePlex configuration features. See "Configure SharePlex to replicate data" in the SharePlex Administration Guide for important information about creating a configuration file and for additional setup instructions for more complex replication scenarios.

An example of a config file for PostgreSQL to Oracle replication:

# data2k configuration file		
datasource:r.source_databasename		
"demo"."data2k"	**"DEMO"."DATA2K"**	**target_system@o.target_databasename**

Set up replication from PostgreSQL to Kafka

Overview

The SharePlex Post process can connect and write to a Kafka broker. The data can be written in JSON or XML output as a sequential series of operations as they occurred on the source, which can then be consumed by a Kafka consumer.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Kafka category

Supported targets

Kafka

Command options

Use the below command options on the SP_CTRL for Kafka:

target x.kafka set kafka broker= 10.250.40.42:9092 (IP address of Kafka server:port number)

target x.kafka set kafka topic= kafkaTarget (topic name)

t**arget x.kafka set format record =** xml/JSON

target x.kafka show

Guidelines for posting to Kafka

- A SharePlex Post process acts as a Kafka producer. A SharePlex Post process can write to one or more topics that have one or more partitions.
- The SharePlex Post process does not create a topic itself, but you can configure the Kafka broker to auto-create topics.

Configure SharePlex on the source

When replicating data to Kafka, configure the source database and SharePlex on the source system as follows.

Configure replication

On the source, create a SharePlex configuration file that specifies capture and routing information. The structure that is required in a configuration file varies, depending on your replication strategy, but this shows you the required syntax for routing data to Kafka.

Datasource:r.*dbname*		
src_schema.table	**!kafka**	*host*

where:

- dbname is a PostgreSQL database name of source
- *src_schema.table* is the schema and name of the source table.
- **!kafka** is a required keyword indicating SharePlex is posting to Kafka.
- *host* is the name of the target system.

NOTE: See Configure SharePlex to replicate data in the SharePlex Administration Guide.

Source configuration example

```
Datasource:r.testdb

MY_SCHEMA.MY_TABLE    !kafka    targetHost
```

Configure SharePlex on the target

These instructions configure the SharePlex Post process to connect to Kafka. You must have a running Kafka broker.

Perform these steps to configure post to Kafka:

1. Create a Kafka topic.
2. Start **sp_cop**. (Do not activate the configuration yet.)
3. Run **sp_ctrl**.
4. Issue the **target** command to configure posting to a Kafka broker and topic. The following are example commands.

   ```
   sp_ctrl> target x.kafka set kafka broker=10.250.40.42:9092

   sp_ctrl> target x.kafkaset kafka topic=shareplex
   ```

   See Set up replication from PostgreSQL to Kafka for command explanations and options.

Set the Kafka Record Format

SharePlex can output to either XML or JSON format as input to Kafka. XML is the default. To set the input format and specify format options, use one of the following **target** commands:

target x.kafka set format record=json

or:

target x.kafka set format record=xml

To view samples of these formats, see the **format** category of the **target** command documentation in the SharePlex Reference Guide.

View and change Kafka settings

To view current property settings for output to Kafka, use the following **target** command:

target x.kafka show

Table 13: Kafka target properties

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**}	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. • 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.	**0**
request.required.acks= *value*	Optional. This is a Kafka client parameter. By default it is set	**-1**

Property	Input Value	Default
	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.	
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 schema 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.enable** 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.

Set up replication from PostgreSQL to SQL Server

Overview

SharePlex can post replicated source data to a Microsoft SQL Server target database through an Open Database Connectivity (ODBC) interface.

These instructions contain setup instructions that are specific to this target. Install SharePlex on the source and target according to the appropriate directions in this manual before performing these setup steps.

For the versions, data types and operations that are supported when using SharePlex to replicate to this target, see the SharePlex Release Notes.

Configure SharePlex on the source

When replicating data to SQL Server, configure the source database and SharePlex on the source system as follows.

Configure replication

On the source, create a SharePlex configuration file that specifies capture and routing information. The structure that is required in a configuration file varies, depending on your replication strategy, but this shows you the required syntax for routing data to SQL Server.

Datasource:r.*dbname*		
src_schema.table	*tgt_owner.table*	*host*

where:

- dbname is a PostgreSQL database name of source
- `src_schema.table` is the schema and name of the source table.
- `tgt_owner.table` is the owner and name of the target table.
- *host* is the name of the target system.

NOTE: See Configure SharePlex to replicate data in the SharePlex Admin Guide.

Source configuration example

The following example replicates table SCOTT.EMP from PostgreSQL instance dbnameA to the MS SQL Server on host sysprod

```
Datasource: r.testdb
SCOTT.EMP    SCOTT.EMP   sysprod@r.mydb
```

Configure SharePlex on the target

Perform the following steps to configure SharePlex on the target:

1. Make certain that the database setup meets all of the requirements in Open target checklist on page 29 .
2. Run Database Setup for SQL Server (**mss_setup**) to establish a database account and connection information for SharePlex. For more information, see Database setup for SQL Server on page 152.

Installation and Setup for Cloud-Hosted Databases for PostgreSQL

Contents

- Overview of SharePlex Setup on Cloud for PostgreSQL
- Post to PaaS Cloud from the Source Server for PostgreSQL
  - Source PostgreSQL to cloud PostgreSQL
  - Source PostgreSQL to cloud Open Target
- Post to PaaS Cloud from an Intermediary Server for PostgreSQL
  - Source PostgreSQL to cloud Open Target

Overview of SharePlex Setup on Cloud for PostgreSQL

SharePlex supports databases installed as services of Amazon Web Services (AWS), Google Cloud SQL for PostgreSQL, and Microsoft Azure. To view the cloud databases that SharePlex supports, refer to the **Supported Cloud Platforms** section in the SharePlex Release Notes document for the respective databases.

There are some differences in the way that SharePlex installs in an IaaS cloud environment, a PaaS cloud environment, and a SaaS cloud environment. These differences are only in the installation and configuration of SharePlex. Once installed and configured, SharePlex operates in the cloud the same way that it operates in on-premise installations. For detailed information regarding how SharePlex is installed in various environments, see Overview of SharePlex Setup on Cloud.

Post to PaaS Cloud from the Source Server for PostgreSQL

These instructions help you set up SharePlex to replicate to a target cloud database directly from the source server. All of the SharePlex processes will run on this server. In a high-volume transactional environment, this may generate unacceptable overhead on the production system. In that case, you should use an intermediary server.

See also:

Post to PaaS cloud from an intermediary server for PostgreSQL

Overview of SharePlex setup on cloud databases

Source PostgreSQL to cloud PostgreSQL

All steps are performed on the source server:

1. Complete the SharePlex pre-installation checklist.
2. Install SharePlex. See Install SharePlex on Linux/Unix for PostgreSQL Database.
3. Run **pg_setup** to establish connection information for the source PostgreSQL database. See Database Setup for PostgreSQL.
4. Add an entry in the local **ODBC.ini** or **ODBCINST.ini** files that points to the target cloud database. This enables you to query the target database from the source server.
5. Run **pg_setup** to establish connection information for the target cloud database. Run the utility as the master user that was created when the service was created. This user has the required DBA privileges.
6. Specify the following in the routing map of the SharePlex configuration file:
   - The name of the source host to specify the target system.
   - The PostgreSQL database name of the target cloud database.

   In the following example, source1 is the source system and `postgresqldatabasename` is the alias from ODBC files of the target cloud database.

datasource: r.orcldbprim

```
#source tables     target tables           routing map

splex.demo_src     splex.demo_dest          source1@r.postgresqldatabasename
```

For more information, see Configure SharePlex to replicate data in the SharePlex Administration Guide.

Source PostgreSQL to cloud Open Target

All the below steps are performed on the source server:

1. Complete the SharePlex preinstallation checklist.
2. Install SharePlex. See: SharePlex on Linux/Unix for PostgreSQL Database.
3. Run the appropriate database setup utility for the source database. See:
   - Database Setup for PostgreSQL.
4. Install the appropriate ODBC client of the target cloud database.
5. Run the appropriate database setup utility for the target cloud database. See Database Setup Utilities.
6. Specify the following in the routing map of the SharePlex configuration file:
   a. the name of the source server as the target host.
   b. the name of the cloud database as the target database

For more information about how to specify configuration components, see **Configure SharePlex to replicate data** in the SharePlex Administration Guide.

Post to PaaS Cloud from an Intermediary Server for PostgreSQL

These instructions help you set up SharePlex to replicate to a target cloud database from an intermediary server that runs the SharePlex Import and Post components.

See also:

Post to PaaS cloud from the source server for PostgreSQL

Overview of SharePlex setup on cloud databases.

Source PostgreSQL to cloud Open Target

1. Complete the SharePlex preinstallation checklist.
2. On the source server, install SharePlex for the source database. See: Install SharePlex on Linux/Unix for PostgreSQL Database.
3. On the source server, run the appropriate database setup utility for the source database. See: Database Setup for PostgreSQL.
4. On the intermediary server, install the appropriate ODBC client of the target cloud database.
5. On the intermediary server, install SharePlex for the target cloud database. See: Install SharePlex on Linux/Unix for PostgreSQL Database.
6. On the intermediary server, run the appropriate database setup utility for the target cloud database. See Database Setup Utilities.
7. Specify the following in the routing map of the SharePlex configuration file:
    a. the name of the intermediary server as the target host.
    b. the name of the cloud database as the target database.

In the following example using a MySQL target, intermediary3 is the intermediary system and sptest3 is the target cloud database.

datasource:r.orcldbprim

```
#source tables       target tables      routing map

HR.EMP               "sptest3"."emp"    intermediary3@r.sptest3
```

For more information, see Configure SharePlex to replicate data in the SharePlex Administration Guide.

Installation and Setup for Remote Capture for PostgreSQL

Using PostgreSQL as a source with remote capture requires that the database instances be accessible from the servers on which SharePlex is installed.

To set up SharePlex for remote capture:

1. Run PG Setup on the SharePlex system according to the instructions provided in the Database Setup Utilities for PostgreSQL section of the SharePlex Reference Guide.
    - When prompted with "Is DB hosted over Azure Single Server?", reply with "No" as Azure Single Server is not supported as a source.
2. Complete all the setup tasks as mentioned in the Database Setup for PostgreSQL Database as a Service.
3. After completing the setup, start **sp_cop** and **sp_ctrl**.
4. For configuring and starting replication, refer to the SharePlex Administration Guide.

Install SharePlex on PostgreSQL High Availability Cluster

SharePlex supports CrunchyData High Availability cluster environment setup.

Follow the below configuration steps:

1. Setup the CrunchyData High Availability cluster environment according to the CrunchyData setup documentation.
2. Install or upgrade to SharePlex 11.1.
3. Run the `pg_setup` utility and enter a slot name.
4. Activate the configuration. The user input slot name will be created in the database after a successful activation.
5. Add the slot name to the respective CrunchyData config [YML or YAML] file to monitor in failover or switchover scenario.
6. Run the deactivate configuration or cleanup [`pg_cleansp`] utility to remove the dedicated slot name from the database. Users need to remove SharePlex dedicated slot name from the CrunchyData config file.
7. Remove the SharePlex dedicated slot name from the CrunchyData config file.

Example of CrunchyData config command: `patronictl -c /etc/patroni/crunchy-demo.yml edit-config`

NOTE: User need to add SharePlex dedicated slot name to the respective CrunchyData config.

LIMITATION: SharePlex internally uses PostgreSQL logical replication with a PostgreSQL database over cloud services. In the event of a failover to the standby server, the logical replication slots are not copied over to the standby server on cloud database services; hence, SharePlex will not handle logical slot re-creation and maintenance with cloud database services. This applies to the AWS Multi-AZ cluster setup of the RDS PostgreSQL, and the Aurora PostgreSQL databases.

Configure SharePlex on PostgreSQL Azure Flexible Server with High Availability Using Logical Replication

SharePlex supports HA with logical replication on PostgreSQL Azure Flexible Server.

Follow the below configuration steps:

1. Enable high availability setup on Azure Flexible Server using the steps provided in the link below:

   https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/how-to-manage-high-availability-portal#enable-high-availability-post-server-creation

   IMPORTANT: Users should be able to access the database using the primary server name (host name).

2. Create `pg_failover_slots` extension setup using the steps provided in the link below:

   https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-extensions#pg_failover_slots-preview

   NOTE: The `pg_failover_slots` extension is supported for PostgreSQL version 11 to 15.

3. Use the `servername = hostname` under the DSN in the **odbc.ini** file. User should use this DSN during `pg_setup`.

   Example:

   ```
   [DSN]

   Servername=pslflexihaserver01.postgres.database.azure.com
   ```

   NOTE: Do not use the IP address of the primary database server, as it may change after failover. As the hostname always points to the current primary database server, we should use only the host name.

4. Stop the **Capture** process before the failover and restart it afterward, in the event of a planned failover.

   In the event of an unplanned failover, the **Capture** process will stop due to an error state after the failover and will need to be manually restarted.

LIMITATION: If consecutive failovers occur, before initiating the capture following the initial failover, the `pg_failover_slots` extension will remove logical slots from both the primary and standby servers. The reason is that after the first failover, the slot on the standby is marked as active and the slot on the primary is marked as inactive. An active state of 'true' on the standby indicates that the slot has not yet synchronized and is not safe for use. Hence, when the failover happens again, the slot on the new primary is lost. To avoid the removal of slots on the primary and standby servers, the user must start the capture after each failover. So, the extension should ideally mark the slot on the standby as inactive (as it being inactive means it is safe to replicate). For additional information, see https://github.com/EnterpriseDB/pg_failover_slots/issues/25.

Generic SharePlex Demonstration for PostgreSQL

Contents

Prework for the demonstrations

Start SharePlex

Create and activate a configuration for PostgreSQL

Demonstration of replication for PostgreSQL

Demonstration of named post queues for PostgreSQL

Overview

This chapter demonstrates the basics of SharePlex replication. This demonstration can be run on Unix or Linux from a PostgreSQL source to supported target databases.

NOTES:

- These demonstrations are for use with databases. They do not support replication to a file or a messaging container.
- These are only demonstrations. Do not use them as the basis for deployment in a production environment. To properly implement replication in your environment, follow the instructions in the SharePlex Installation and Setup Guide and the SharePlex Administration Guide.
- For more information about the commands used in the demonstrations, see the SharePlex Reference Guide.
- The demonstrations assume that SharePlex is fully installed on a source system and one target system, and that any pre- and post-installation setup steps were performed.

What you will learn

- How to activate a configuration
- How SharePlex replicates smoothly from source to target systems
- How SharePlex quickly and accurately replicates large transactions
- How SharePlex queues the data if the target system is unavailable
- How SharePlex resumes from its stopping point when the target system is recovered
- How SharePlex recovers after a primary instance interruption
- How to use named queues to spread the processing of different tables across parallel Post processes

Prework for the demonstrations

Before you run the basic demonstrations, have the following items available.

Tables used in the demonstrations

You will replicate **splex.demo_src** from the source system to **splex.demo_dest** on the target system. These tables are installed by default into the SharePlex schema, which in these demonstrations is "**splex**." Your SharePlex schema may be different. Verify that these tables exist.

Description of the demo tables.

Column Name	Data Type	Null?
NAME	varchar2(30)	
ADDRESS	verchar2(60)	
PHONE	varchar2(12)	

INSERT scripts

- Create a SQL script named **insert_demo_src** that inserts and commits 500 rows into the **splex.demo_src** table. You will run this script during some of the demonstrations.
- If you will be using the demonstration of named post queues, create a SQL script named **insert_demo_dest** that inserts and commits 500 rows into the **splex.demo_dest** table. You will run this script during some of the demonstrations.

Start SharePlex

The following are instructions for starting SharePlex and the **sp_ctrl** command-line interface on Unix and Linux. Start SharePlex on the source and target systems.

To start SharePlex on Unix and Linux systems:

1. Log onto the system as a SharePlex Administrator (a member of the ***SharePlex Admin group***).
2. From the **bin** sub-directory of the SharePlex product directory (*productdir* in the following syntax), run **sp_cop** and **sp_ctrl**.

   $**cd** /*productdir*/**bin**

   $**./sp_cop &**

   $**. /sp_ctrl**

Create and activate a configuration for PostgreSQL

SharePlex gets its replication instructions from a configuration file, which defines the objects that are to be replicated. The file specifies following:

- The *datasource* (source database) — the identifier of the source database.
- The *source* objects — the names of the objects that contain the data to be replicated.
- The *target* objects — the names of the target objects that will receive the replicated data.
- The *routing map* — the name of the target system and, if the target is a database, its identifier.

NOTE: This demonstration demonstrates replication from one database to another. It does not cover replication to a file or a messaging container.

Create a configuration file

Perform these steps on the **source** system. The demonstration objects are assumed to be in the schema named **splex**.

Perform the following steps to create a configuration file:

1. Run **sp_ctrl**.
2. Issue the following command to create a configuration file named **sample_config** in the default text editor.

   `sp_ctrl(source)>`**create config sample_config**
3. In the text editor, build your configuration file based on the appropriate template, as shown in the Configuration templates. Allow no spaces between the characters in a component (source specification, target specification, routing map), and place at least one space between each component.
4. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory of the variable-data directory.
5. In **sp_ctrl**, verify that the configuration file will activate successfully.

   `sp_ctrl(source)>`**verify config sample_config**

Configuration templates

PostgreSQL source to PostgreSQL target

```
datasource:r.source_DB

"qarun"."basic_c127"     "splex"."basic_c127"     10.250.14.105@r.sp_ad
```

where:

- *source_DB* is the database name of the source database
- *"qarun"* is the database user of the source database

- *"splex"* is the database user of the target database
- *"sp_ad"* is the database name of the source database

Activate the configuration

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

Perform the following steps to activate a configuration file:

1. Activate the configuration.

   `sp_ctrl(source)>`**activate config sample_config**

   NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **sample_config** should appear under **File Name**, and the word **Active** should appear under **State**.

   `sp_ctrl(source)>`**list config**

Troubleshooting tips

If your configuration activation fails, isue the **view config sample_config** command to view the file. Compare it to the template and make sure all of the information you entered is correct. Make certain you specified the correct database identifier. Check your syntax for extra spaces or missing components.

Because the configuration file is not active, you can edit it directly with the following command:

`sp_ctrl(source)>`**edit config sample_config**

Save the changes, then re-try the activation.

NOTE: To change an active configuration, you must copy it to a new file first, and then edit and activate the copy. For more information, see *Add or Change Table Specifications in an Active Configuration* in the SharePlex Installation and Setup Guide.

Demonstration of replication for PostgreSQL

This section demonstrates the speed and accuracy of SharePlex replication. It can also be used to verify that SharePlex was installed and configured correctly. For detailed information, see Demonstration of replication.

Demonstration of named post queues for PostgreSQL

This demonstration shows you how to use named post queues to process different tables through different, parallel Post processes to improve performance.

Clean up the replication environment

NOTE: The demonstration objects are assumed to be in the schema named **splex**.

Perform the following steps to clean up the replication environment:

1. **If you ran previous demonstrations, do the following:**
   a. On the **source** and **target**, run **sp_ctrl** and issue the following command to shut down **sp_cop**.

      `sp_ctrl(source)` **shutdown**

      `sp_ctrl(target)` **shutdown**

   b. On the **source** and **target**, run the **pg_cleansp** utility according to the instructions in SharePlex utilities. This removes the queues from the previous demonstrations and deactivates the previous configuration.
2. On the **source**, TRUNCATE the **splex.demo_src**. SharePlex replicates the TRUNCATE to the target.

   truncate table splex.demo_src;

Create a configuration file

Perform these steps on the **source** system. The demonstration objects are assumed to be in the schema named **splex**.

You will configure the following:

- **demo_src** on the source replicates through a post queue named **q1** to target **demo_dest**.
- **demo_dest** on the source replicates through a post queue named **q2** to target **demo_src**.

1. Run **sp_ctrl**.
2. Issue the following command to create a configuration file named **postq_config** in the default text editor.

   `sp_ctrl(source)>`**create config postq_config**
3. In the text editor, build your configuration file based on the appropriate template, as shown in the Configuration templates. Allow no spaces between the characters in a component (source specification, target specification, routing map), and place at least one space between each component.
4. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory of the variable-data directory.

5. In **sp_ctrl**, verify that the configuration file will activate successfully.

   ```
   sp_ctrl(source)>verify config postq_config
   ```

Configuration templates

PostgreSQL to PostgreSQL

datasource:r.dbname		
splex.demo_src	**splex.demo_dest**	*target_system*:**q1@r.***database_name*
splex.demo_dest	**splex.demo_src**	*target_system*:**q2@r.***database_name*

where:

- *dbname* is the database name of the source database
- *splex* is the database user of the target database
- *database_name* is the name of the open target database.

Activate the configuration

IMPORTANT!Make certain you created the scripts described in Prework for the demonstrations on page 192**.**

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

Perform the following steps to activate the configuration:

1. Activate the configuration.

   ```
   sp_ctrl(source)>activate config postq_config
   ```

   NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **postq_config** should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

Generate data

On the **source**, run the following scripts to insert data into the two local demonstration tables (in this demonstration, the **demo_dest** table is used as a source):

- Run the **insert_demo_src** script to insert 500 rows into the **splex.demo_src** table.
- Run the **insert_demo_dest** script to insert 500 rows into the **splex.demo_dest** table.

View the post queues

Perform the following steps to view the post queues:

1. On the **target**, run **sp_ctrl**.
2. On the **target**, issue the show post command with the detail option. This command shows statistics about the number of messages processed and the queues that were used.

   ```
   sp_ctrl(target)>show post detail
   ```

   In the output, the **Queue** field shows the name of each post queue, in this case **q1** and **q2**, and the **Operations Posted** field shows the number of operations that were posted. Each queue should show 500 operations posted.

Advanced SharePlex Demonstrations for PostgreSQL

Contents

Demonstration of conflict resolution - PostgreSQL to PostgreSQL

Demonstration of conflict resolution - PostgreSQL to Oracle

Sample configuration for PostgreSQL to PostgreSQL replication

Configure, activate, and verify replication on the PostgreSQL source

Overview

This chapter demonstrates selected features of SharePlex. These exercises can be run on Unix and Linux systems to demonstrate:

- How to build and verify a replication configuration
- How to use generic conflict resolution in peer-to-peer replication

NOTES:

- These are only demonstrations. Do not use them as the basis for deployment in a production environment. To properly implement replication in your environment, follow the instructions in the SharePlex Installation and Setup Guide and the SharePlex Admin Guide.
- For more information about the commands used in the demonstrations, see the SharePlex Reference Guide.
- The demonstrations assume that SharePlex is installed on a source system and one target system, and that any pre- and post-installation setup steps were performed. For more information, see the SharePlex Installation and Setup Guide.

Demonstration of conflict resolution - PostgreSQL to PostgreSQL

PREREQUISITES:

- Before proceeding, make certain the SharePlex demonstration objects are installed. See Prework for the demonstration.
- Set the `SP_OPX_CREATE_ORIGIN_PG` to 1 before activation. Set it on the PostgreSQL peer for PostgreSQL to Oracle replication and on both peers for PostgreSQL to PostgreSQL replication.

In this demonstration, you will configure SharePlex to use generic conflict resolution procedures to resolve a replication conflict. Generic conflict resolution allows you to use one PL/SQL procedure to resolve conflicts for multiple tables.

The following conflict-resolution strategy is demonstrated:

- **Trusted-source priority** – In the following steps, you will define one system as the "trusted" source that takes priority in the event of a conflict. This demonstration is based on INSERTs. All INSERTs that originate on the trusted source will override INSERTs from the other system. In this demonstration, the trusted source is **peer1** and the other system is **peer2**.
- **Timestamp priority** – This demonstration is based on UPDATEs. When there is a conflict, the row that was updated LAST prevails.

IMPORTANT! Peer-to-peer replication is not compatible with all business applications. When suitable for an environment, it requires careful analysis and execution, including the creation of custom conflict resolution procedures that are typically more complex than those in this demonstration. Do not use this demonstration as the foundation of a production peer-to-peer deployment. For more information about peer-to-peer replication, see the SharePlex Administration Guide.

Pre-work for the demonstrations

Install the demonstration objects as described below before you start the demonstrations.

NOTE: To run the scripts, set Replica Identity to Full for tables that are a part of replication.

Install the demonstration objects

The **pg_p2p.sql** and **pg_od.sql** scripts install the demonstration objects that you will use during these demonstrations. These scripts are stored in the **util** sub-directory of the SharePlex product directory.

To run the scripts:

- Run **pg_od.sql** first, then run **pg_p2p.sql**.
- Run the scripts on the source and target systems that you will be using for the demonstrations.
- Run them in PSQL as an existing user with the DBA role.

NOTE: The scripts prompt for the schema where you want the demonstration objects need to be installed. Additionally, **pg_p2p.sql** prompts for the name of the system that will be the trusted source of accurate data.

Description of the demonstration objects

od_employee

Name	Null?	Type
emp_no	not null	int
emp_first_name		varchar(20)
emp_last_name		varchar(20)
emp_dob		date
emp_dept_no		int
emp_timestamp		date

exc_table

Name	Null?	Type
exc_no	not null	int
exc_type		varchar (6)
exc_target_table		varchar (66)
exc_fixed		varchar (3)
exc_info		varchar (500)
exc_timestamp		date

exc_source

Name	Type
pri_system	varchar

NOTE: Apart from these tables, other demonstration objects are also created those are not part of this demonstration.

Prepare the objects

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

Perform these steps on both systems (for both databases).

1. Run **sp_ctrl**.
2. Shut down SharePlex.

   ```
   sp_ctrl(peer1)>shutdown
   ```

3. Run the **pg_cleansp** utility. This removes the queues from the previous demonstrations and deactivates the previous configuration.
4. TRUNCATE the **od_employee** tables.
5. Log into PSQL as the user who owns the SharePlex demonstration objects.
6. Run the **pg_p2p.sql** script from the **util** sub-directory of the SharePlex product directory. This installs the **od_employee_gen** demonstration conflict resolution routine. You are prompted for the following:
   - A schema for the procedure
   - The name of the system that will be the *trusted source* of accurate data (`pri_system`). As in a production deployment, operations the trusted source take priority during a conflict. This system is known as ***peer1*** in this demonstration. The other system is known as ***peer2*** in this demonstration.
7. Set the **Replica Identity** to **Full** for the `od_employee` tables.

Configure SharePlex

Perform these steps to configure SharePlex:

1. On each system, open the **conflict_resolution**.*database* file in a text editor. This file is located in the **data** sub-directory of the SharePlex variable-data directory.

   NOTE: Post checks this file when there is a replication conflict to determine if there is a resolution procedure to call.

2. On each system, create the following entries in the **conflict_resolution**.*database* file. Separate each column with at least a few spaces or a tab character.

demo.od_employee	**IUD**	**demo.od_employee_gen**

   NOTE: The first component is a table, the second specifies the operation types for which a resolution routine will be called if there is a conflict on that table, and the third is the name of the resolution routine that will be used.

3. On each system, start **sp_cop**.
4. On each system, start **sp_ctrl**.

5. On **peer1** (the trusted source), create a configuration file named **od_peer1** that replicates the **od_employee** table to the **od_employee** table on **peer2**.

   ```
   sp_ctrl(peer1)> create config od_peer1
   ```

demo.od_employee	**demo.od_employee**	**peer2@r**.*database*

6. On **peer2** (the secondary source), create a configuration file named **od.peer2** that replicates the **od_employee** table to the **od_employee** table on **peer1**.

   ```
   sp_ctrl(peer2)> create config od_peer2
   ```

demo.od_employee	**demo.od_employee**	**peer1@r**.database

Activate and start replication

Perform these steps to activate and start replication:

1. On **peer1**, activate the **od_peer1** configuration.

   ```
   sp_ctrl(peer1)> activate od_peer1
   ```

2. On **peer2**, activate the **od_peer2** configuration.

   ```
   sp_ctrl(peer2)> activate od_peer2
   ```

3. On each system, confirm that the configuration activated successfully. The name **od_peer1** or **od_peer2** (depending on the system) should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

Demonstrate trusted-source priority

In this demonstration, an INSERT that originates on **peer1** will override a conflicting INSERT that is replicated from **peer2**.

1. On both systems, stop the **Export** process.
2. On both systems, log in to **PSQL** as **demo** (the owner of the demonstration objects).
3. On **peer2**, insert a row into **od_employee**.

   ```
   insert into od_employee values(1, 'John', 'Doe', '1949-04-01', 1, '2022-01-01');
   ```

4. On **peer1**, insert the same row (same values).

   ```
   insert into od_employee values(1, 'John', 'Doe', '1949-04-01', 1, '2022-01-01');
   ```

5. On both systems, restart the **Export** processes.
6. This operation should generate a conflict, which Post resolves automatically based on instructions in the **conflict_resolution**.*database* file.

7. On both systems, view the **demo.exc_table** to verify that the conflict was resolved. See View the conflict resolution results.

Demonstrate timestamp priority

In this demonstration, whenever there is a conflict for an UPDATE statement, the row with the most current value of the **emp timestamp** column gets precedence.

NOTE: The `EMP_TIMESTAMP` field, along with the other columns, must be modified to a non-null value in order for timestamp priority to function. The stored procedure won't be able to resolve the conflict if the timestamp column in the UPDATE statement is not being updated or is being modified to a null value. Out-of-sync problems will then be seen.

1. On both systems, stop the **Export** process.
2. On both systems, log in to **PSQL** as **demo** (the owner of the demonstration objects).
3. On **peer1**, update the `emp_first_name` and `EMP_TIMESTAMP` columns of the **od_employee** table as follows:
```
update od_employee set emp_first_name = 'James', emp_timestamp = '2022-07-01'
where emp_no = 1;
```
4. On **peer2**, update the same columns using a different update value but the same key value.
```
update od_employee set emp_first_name = 'Harry', emp_timestamp = '2022-08-02'
where emp_no = 1;
```
5. On both systems, restart the **Export** processes.
6. On both systems, select the row that you updated to verify that it contains the most recent `emp_timestamp` value.
7. On both systems, view the `demo.exc_table` to verify that the conflict was resolved. See View the conflict resolution results.

Demonstrate no conflict resolution

In this demonstration, if a DELETE statement generates a conflict, it is ignored.

1. On both systems, stop the **Export** process.
2. On both systems, log in to log in to **PSQL** as **demo** (the owner of the demonstration objects).
3. On peer1, delete a row as follows.
```
delete from od_employee where emp_no = 1;
```
4. On peer2, delete the same row.
```
delete from od_employee where emp_no = 1;
```
5. On both systems, restart the **Export** processes.
6. On both systems, view the `demo.exc_table` to verify that the conflict is not resolved. See View the conflict resolution results.

View the conflict resolution results

A table named **exc_table** was installed in the schema that you specified when you installed the demonstration objects. You can view it through PSQL to view information about each conflict. The following is the table description.

Column	Description
EXC_NO	The exception number of the conflict.
EXC_TYPE	The type of SQL statement, whether INSERT, UPDATE or DELETE.
EXC_ TARGET_ TABLE	The table on which the conflict occurred.
EXC_FIXED	The results of the conflict resolution routine. YES means that the routine was successful. NO means that the routine failed and the row needs to be manually changed to the correct value.
EXC_INFO	The cause of the conflict.
EXC_ TIMESTAMP	The time that the conflict occurred on this machine.

Demonstration of conflict resolution - PostgreSQL to Oracle

PREREQUISITE: Before proceeding, make certain the SharePlex demonstration objects are installed. See Prework for the demonstration.

In this demonstration, you will configure SharePlex to use generic conflict resolution procedures to resolve a replication conflict. Generic conflict resolution allows you to use one PL/SQL procedure to resolve conflicts for multiple tables.

The following conflict-resolution strategy is demonstrated:

- **Trusted-source priority** – In the following steps, you will define one system as the "trusted" source that takes priority in the event of a conflict. This demonstration is based on INSERTs. All INSERTs that originate on the trusted source will override INSERTs from the other system. In this demonstration, the trusted source is **peer1**(i.e. PostgreSQL) and the other system is **peer2**(i.e. Oracle).
- **Timestamp priority** – This demonstration is based on UPDATEs. When there is a conflict, the row that was updated LAST prevails.

IMPORTANT! Peer-to-peer replication is not compatible with all business applications. When suitable for an environment, it requires careful analysis and execution, including the creation of custom conflict resolution procedures that are typically more complex than those in this demonstration. Do not use this demonstration as the foundation of a production peer-to-peer deployment. For more information about peer-to-peer replication, see the SharePlex Administration Guide.

Pre-work for the demonstrations

Install the demonstration objects as described below before you start the demonstrations.

NOTE: The below pre-work is applicable only for the Oracle peer. For the information on the pre-work of PostgreSQL peer, see Pre-work for the Demonstrations - PostgreSQL.

Install the demonstration objects

The **p2p.sql** and **od.sql** scripts install the demonstration objects that you will use during these demonstrations. These scripts are stored in the **util** sub-directory of the SharePlex product directory.

To run the scripts:

- Run **od.sql** first, then run **p2p.sql**.
- Run the scripts on the source and target systems that you will be using for the demonstrations.
- Run them in SQL*Plus as an existing user with the DBA role and SELECT ANY TABLE privileges.

The scripts prompt for the following items:

- The schema where you want the demonstration objects to be installed.
- The tablespace for the demonstration objects.
- Whether or not you want old demonstration objects from a previous version of SharePlex to be removed. To remove the old objects, supply the name of their schema.

Description of the demonstration objects

od_employee

Name	Null?	Type
EMP_NO	NOT NULL	NUMBER
EMP_FIRST_NAME		VARCHAR2
EMP_LAST_NAME		VARCHAR2
EMP_DOB		DATE
EMP_DEPT_NO		NUMBER
EMP_TIMESTAMP		DATE

od_timesheet

Name	Null?	Type
TS_EMP_NO		NUMBER
TS_IN_TIME		DATE
TS_OUT_TIME		DATE
TS_MOD_TIME		DATE

od_department

Name	Null?	Type
DEPT_NO	NOT NULL	NUMBER
DEPT_NAME		VARCHAR2
DEPT_CODE		VARCHAR2

od_salary

Name	Null?	Type
SALE_EMP_NO		NUMBER
SAL_VALUE		NUMBER
SAL_CHANGED		DATE

od_sales_emp_data

Name	Null?	Type
EMP_NO_KEY	NOT NULL	NUMBER
EMPLOYEE_NAME		VARCHAR2 (70)
SALARY		NUMBER
DEPARTMENT		VARCHAR2 (50)

oxc_table

Name	Null?	Type
EXC_NO	NOT NULL	NUMBER
EXC_TYPE		VARCHAR2 (6)
EXC_TARGET_TABLE		VARCHAR2 (66)
EXC_FIXED		VARCHAR2 (3)
EXC_INFO		VARCHAR2 (500)
EXC_TIMESTAMP		DATE

Prepare the objects

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

The following steps provide information on how to prepare the demonstration objects for Oracle peer. For information on how to prepare the demonstration objects of PostgreSQL peer, see Prepare the objects.

Perform these steps on Oracle peer:

1. Run **sp_ctrl**.
2. Shut down SharePlex.

   ```
   sp_ctrl(peer1)>shutdown
   ```

3. Run the **ora_cleansp** utility. This removes the queues from the previous demonstrations and deactivates the previous configuration.
4. TRUNCATE the **od_employee** tables.
5. Grant the user who owns the demonstration objects the system privilege to execute the **sp_cr** package, which was installed in the SharePlex schema when SharePlex was first installed.

   ```
   SQL>grant execute on sp_cr to user_name
   ```

6. Log into SQL *Plus as the user who owns the SharePlex demonstration objects.

7. Run the **p2p.sql** script from the **util** sub-directory of the SharePlex product directory. This installs the **od_employee_gen** demonstration conflict resolution routine. You are prompted for the following:
    - A schema and tablespace for the procedure
    - The name of the SharePlex database user.
    - The name of the system that will be the *trusted source* of accurate data. As in a production deployment, operations the trusted source take priority during a conflict. This system is known as ***peer1*** in this demonstration. The other system is known as ***peer2*** in this demonstration.

Configure SharePlex

Perform these steps to configure SharePlex:

NOTE: In this configuration, Peer 1 is Postgresql database and Peer2 is Oracle database. The name of the conflict_resolution file for Oracle is `conflict_resolution.sid`, and for PostgreSQL it is `conflict_resolution.database`.

1. On each system, open the **conflict_resolution** file in a text editor. This file is located in the **data** sub-directory of the SharePlex variable-data directory.

    NOTE: Post checks this file when there is a replication conflict to determine if there is a resolution procedure to call.

2. On each system, create the following entries in the **conflict_resolution** file. Separate each column with at least a few spaces or a tab character.

demo.od_employee	**IUD**	**demo.od_employee_gen**

    NOTE: The first component is a table, the second specifies the operation types for which a resolution routine will be called if there is a conflict on that table, and the third is the name of the resolution routine that will be used.

3. On each system, start **sp_cop**.
4. On each system, start **sp_ctrl**.
5. On **peer1** (the trusted source), create a configuration file named **od_peer1** that replicates the **od_employee** table to the **od_employee** table on **peer2**.

    `sp_ctrl(peer1)>` **create config od_peer1**

datasource:r.dbname		
"demo"."od_employee" ("emp_no", "emp_first_name", "emp_last_name", "emp_dob", "emp_dept_no", "emp_timestamp")	"DEMO"."OD_EMPLOYEE" ("EMP_NO", "EMP_FIRST_NAME", "EMP_LAST_NAME", "EMP_DOB", "EMP_DEPT_NO", "EMP_TIMESTAMP")	**peer2@o**.*sid*

6. On **peer2** (the secondary source), create a configuration file named **od.peer2** that replicates the **od_employee** table to the **od_employee** table on **peer1**.

```
sp_ctrl(peer2)> create config od_peer2
```

datasource:o.sid		
"DEMO"."OD_EMPLOYEE" ("EMP_NO", "EMP_FIRST_NAME", "EMP_LAST_NAME", "EMP_DOB", "EMP_DEPT_NO", "EMP_TIMESTAMP")	"demo"."od_employee" ("emp_no", "emp_first_name", "emp_last_name", "emp_dob", "emp_dept_no", "emp_timestamp")	**peer2@r.** dbname

NOTE: In the above example, column mapping feature is used since both (source and target) tables have different case sensitivity.

7. Set the `SP_OPO_SUPPRESSED_OOS` parameter to 0 on peer 2.

Activate and start replication

Perform these steps to activate and start replication:

1. On **peer1**, activate the **od_peer1** configuration.

```
sp_ctrl(peer1)> activate od_peer1
```

2. On **peer2**, activate the **od_peer2** configuration.

```
sp_ctrl(peer2)> activate od_peer2
```

3. On each system, confirm that the configuration activated successfully. The name **od_peer1** or **od_peer2** (depending on the system) should appear under **File Name**, and the word **Active** should appear under **State**.

```
sp_ctrl(source)>list config
```

Demonstrate trusted-source priority

In this demonstration, an INSERT that originates on **peer1** will override a conflicting INSERT that is replicated from **peer2**.

1. On both systems, stop the **Export** process.
2. On both systems, log in to database as **demo** (the owner of the demonstration objects).
3. On **peer1**, insert a row into **od_employee**.

```
insert into od_employee values(1, 'John', 'Doe', '1949-04-01', 1, '2022-01-01');
```

4. On **peer2**, insert the same row (same values).

```
INSERT INTO OD_EMPLOYEE VALUES (1,'John','Doe',to_date
('04/01/1949','MM/DD/RRRR'),1,to_date('01/01/2022','MM/DD/RRRR'));
```

5. On both systems, restart the **Export** processes.

6. This operation should generate a conflict, which Post resolves automatically based on instructions in the **conflict_resolution** file.
7. On both systems, view the **demo.exc_table** to verify that the conflict was resolved. See View the conflict resolution results.

Demonstrate timestamp priority

In this demonstration, whenever there is a conflict for an UPDATE statement, the row with the most current value of the `emp_timestamp` column gets precedence.

NOTE: The `EMP_TIMESTAMP` field, along with the other columns, must be modified to a non-null value in order for timestamp priority to function. The stored procedure won't be able to resolve the conflict if the timestamp column in the UPDATE statement is not being updated or is being modified to a null value. Out-of-sync problems will then be seen.

1. On both systems, stop the **Export** process.
2. On both systems, log in to **database** as **demo** (the owner of the demonstration objects).
3. On **peer1**, update the `emp_first_name` and `emp_timestamp` column of the **od_employee** table as follows.
```
update od_employee set emp_first_name = 'James', emp_timestamp = '2022-07-01'
where emp_no = 1;
```
4. On **peer2**, update the same column using a different update value but the same key value.
```
UPDATE OD_EMPLOYEE SET EMP_FIRST_NAME = 'Harry', EMP_TIMESTAMP = to_date
('02/02/2017','MM/DD/RRRR') WHERE EMP_NO = 1;
```
5. On both systems, restart the **Export** processes.
6. On both systems, select the row that you updated to verify that it contains the most recent `emp_timestamp` value.
7. On both systems, view the `demo.exc_table` to verify that the conflict was resolved. See View the conflict resolution results.

Demonstrate no conflict resolution

In this demonstration, if a DELETE statement generates a conflict, it is ignored.

1. On both systems, stop the **Export** process.
2. On both systems, log in to log in to **database** as **demo** (the owner of the demonstration objects).
3. On peer1, delete a row as follows.
```
delete from od_employee where emp_no = 1;
```
4. On peer2, delete the same row.
```
DELETE FROM OD_EMPLOYEE WHERE EMP_NO = 1;
```
5. On both systems, restart the **Export** processes.
6. On both systems, view the `demo.exc_table` to verify that the conflict is not resolved. See View the conflict resolution results.

View the conflict resolution results

A table named **exc_table** was installed in the schema that you specified when you installed the demonstration objects. You can view it through PSQL for PostgreSQL and SQLPLUS for Oracle to view information about each conflict. The following is the table description.

Column	Description
EXC_NO	The exception number of the conflict.
EXC_TYPE	The type of SQL statement, whether INSERT, UPDATE or DELETE.
EXC_ TARGET_ TABLE	The table on which the conflict occurred.
EXC_FIXED	The results of the conflict resolution routine. YES means that the routine was successful. NO means that the routine failed and the row needs to be manually changed to the correct value.
EXC_INFO	The cause of the conflict.
EXC_ TIMESTAMP	The time that the conflict occurred on this machine.

Sample configuration for PostgreSQL to PostgreSQL replication

SharePlex gets its replication instructions from “configurations,” which are user-defined specifications that tell SharePlex what to do. For each group of objects that you want to replicate, you create a configuration file.

Configurations reside on the source system and define:

- **The *datasource* (source database)** — the identifier of the source database.
- **The *source* objects** — the names of the objects within the source database that contain the data to be replicated. You can replicate some or all of the tables within a database.
- **The *target* objects** — the names of the objects in the database on the target system that will receive the replicated data.
- **The *routing map*** — the name of the target system and, if the target is a database, its identifier.

To create the demonstration configuration:

1. Create a replication configuration named **sample_config** by issuing the **create config** command in **sp_ctrl** on the **source** system. This opens the default text editor, which is **vi** for Linux systems.

   sp_ctrl(sysA)> create config sample_config

   Refer to Template 1 below as you construct your configuration.

   Template 1: Basic demonstration configuration sample_config

datasource:r.*source_DB*		
"qarun"."basic_c127"	"splex"."basic_c127"	10.250.14.105@r.sp_ad

2. On the ***first*** non-commented line of the file, type the following, leaving *no* space between any of the items.

   datasource:r.*source_databasename*

   (Substitute the PostgreSQL database name of the source instance for source_databasename.) This tells SharePlex where to find the table whose data will be replicated. The **r**. tells Share- Plex that PostgreSQL data is being replicated.

3. On the next line, enter the owner name (**splex**) and table name (**demo_src**) of the **source** table, separating the two items with a dot (.) but no spaces. Using the owner’s name with a table name ensures that SharePlex replicates the correct table, since different tables in different schemas in a database could have the same name.

   splex.demo_src

4. Type at least a few spaces or a tab to create a second column. **Do not press Enter**.
5. In the second column, enter the owner name (**splex**) and table name (**demo_dest**) of the target table, separating the two items with a dot (.) but *no* spaces.

   splex.demo_dest

6. Type a few spaces or a tab to create a third column. **Do not press Enter**.

7. In the third column, type the following items with *no* space between them. This creates the routing map for your configuration, telling SharePlex where to put the replicated data.
    - the name of the target system
    - the @ symbol
    - the letter r
    - a dot (.)
    - the target instance

    Example:

    sysB@r.databasename
8. Save the file and exit the editor. This returns you to the **sp_ctrl** prompt.
9. [OPTIONAL] To view the configuration, issue the **view config** command in **sp_ctrl** on the **source** system:

    ```
    sp_ctrl(sysA)>view config sample_config
    ```
10. Activate the configuration in **sp_ctrl** on the **source** system. Configuration names are case-sensitive.

    ```
    sp_ctrl(sysA)>activate config sample_config
    ```
11. To confirm that your configuration is active, type the following **sp_ctrl** command on the **source** system to display a list of all configurations. The **sample_config** configuration should appear under “File Name,” and the word “Active” should appear under “State.”

    ```
    sp_ctrl(sysA)>list config
    ```

Tip: If your configuration activation fails, use the **view config sample_config** command in sp_ctrl to view the file. Compare it to Template 1 and make sure all of the information you entered is correct. For example, check for extra spaces that are not supposed to be there, or for missing components, such as the r. before the database name.

To correct mistakes in the configuration file:

1. Run the **edit config sample_config** command in **sp_ctrl**before you activate the configuration file (or if the activation failed).
   This command opens the file in the text editor.
2. Make the changes by editing the file.
3. Save the changes, and re-try the activation.

NOTE: To change an active configuration, you must copy it to a new file first with the copy **config** command, and then edit and activate the copy.

Configure, activate, and verify replication on the PostgreSQL source

NOTE: Before proceeding, make certain the SharePlex demonstration objects are installed.

In this demonstration, you will create and activate a replication configuration, load data to the source table, and then confirm that the data was replicated successfully to the target table. You will also repair a table that is out of synchronization.

Create a configuration file

NOTE: In this demonstration, the demonstration objects are assumed to be in the schema named **demo**. Substitute the actual schema, if different.

Perform these steps on the source system:

1. Run **sp_ctrl** .
2. Issue the following command to create a configuration file named **sample_config** that replicates the **sample_department**, **sample_salary**, **sample_timesheet**, and **sample_employee** tables to target tables of the same names on the target system..

   ```
   sp_ctrl(source)>create config sample_config
   ```

3. In the text editor, build your configuration file based on the following template. Allow no spaces between the characters in a component (source specification, target specification, routing map), and place at least one space between each component.

# sample_config configuration file		
datasource:r.source_databasename		
"demo"."sample_department"	**"demo"."sample_department"**	target_system@r.target_ databasename
"demo"."sample_salary"	**"demo"."sample_salary"**	target_system@r.target_ databasename
"demo"."sample_timesheet"	**"demo"."sample_timesheet"**	*target_system@r.target_ databasename*
"demo"."sample_employee"	**"demo"."sample_employee"**	*target_system@r.target_ databasename*

where:

- # denotes a comment.
- *source_databasename* is the PostgreSQL database name of the source database.
- *target_system* is the name or IP address of the target system.
- *target_databasename* is the PostgreSQL database name of the target database.

4. Save the file, then exit the editor. SharePlex automatically saves the file in the **config** sub-directory.
5. In **sp_ctrl**, verify that the configuration file will activate successfully.

   ```
   sp_ctrl(source)>verify config sample_config
   ```

Syntax configuration

An example of a config file for PostgreSQL to PostgreSQL replication:

```
# data2k configuration file

datasource:r.source_databasename

"demo"."data2k"          "demo"."data2k"          target_system@r.target_
                                                  databasename
```

An example of a config file for PostgreSQL to Oracle replication:

```
# data2k configuration file

datasource:r.source_databasename

"demo"."data2k"          "demo"."data2k"          target_system@o.target_
                                                  databasename
```

Activate the configuration

Perform these steps on the **source** system. When you activate a configuration, SharePlex is ready to capture transactional changes that are made to the specified source data.

1. Activate the configuration.

   ```
   sp_ctrl(source)>activate config sample_config
   ```

   NOTE: Configuration names are case-sensitive.

2. Confirm that the configuration activated successfully. The name **config sample_config** should appear under **File Name**, and the word **Active** should appear under **State**.

   ```
   sp_ctrl(source)>list config
   ```

Database Setup for PostgreSQL

Pre-requisites to configure PostgreSQL database setup as a source:

1. In the PostgreSQL configuration file, set the **wal_level** parameter to **logical**.
   For example: `wal_level = logical # minimal, replica, or logical`
2. Add an entry of SharePlex user in the `pg_hba.conf` file so that SharePlex Capture can establish a connection with the `walsender` process of PostgreSQL
   For example: `# TYPE DATABASE USER ADDRESS METHOD`
   `host replication all 10.10.10.10/32 trust`
3. For all the tables which are going to be part of replication, set the 'replica identity' to 'full' before activation.
   For example: `alter table "DEMO_SRC" replica identity full;`
4. Navigate to the `/var/lib/pgsql/13/data` directory and edit the PostgreSQL file.
   Remove the comment and increase the below two parameters' value as below:
   `wal_sender_timeout = 6000s`
   `wal_receiver_timeout = 6000s`
5. Set the value of `logical_decoding_work_mem = 4096MB` (recommended). This specifies the maximum amount of memory to be used by logical decoding.
6. Navigate to the bin directory and restart database: `cd /usr/pgsql-13/bin/./pg_ctl -D /var/lib/pgsql/13/data/ restart`

NOTES:

- The setup steps provided for PostgreSQL are also applicable for setting up SharePlex to replicate data from source/on-prem to a target cloud database.
- The setup steps provided for PostgreSQL are also applicable for setting up PostgreSQL remote Post.
- SharePlex PostgreSQL as a source only supports replication from the WAL files present in the `pg_wal` directory, located within the variable data directory of the PostgreSQL database installation.
- For controlled character set conversions between source and target, ensure that the PostgreSQL Database character set value and the `PGCLIENTENCODING` database environment parameter value are exactly the same as those specified in the source and target database character set settings.

Pre-requisites to configure PostgreSQL database setup as a target:

Fujitsu Enterprise PostgreSQL

Fujitsu PostgreSQL contains a customized ODBC driver designed to run with the shared libraries supplied by Fujitsu PostgreSQL. In order to ensure the proper configuration, you must create a symbolic link to ensure that the correct libraries are found at runtime.

Issue Description:

If you attempt to run the pg_setup program prior to creating the symbolic link, pg_setup will fail with the following error message:

```
Error: [unixODBC]invalid connection option "target_server"
```

To create a symbolic link:

To ensure that the proper library gets loaded, first locate the path to the directory where the Fujitsu PostgreSQL libraries are installed (for example /opt/fsepv95server64/lib). Then create a symbolic link in the SharePlex OPTDIR/lib/hooks directory pointing to the **libpq.so.5** library in the Fujitsu PostgreSQL library directory. For example:

```
cd /opt/splex/lib/hooks

ln -s /opt/fsepv95server64/lib/libpq.so.5
```

Overview

Run the PG Setup utility for PostgreSQL (**pg_setup**) to establish a user account, schema, and tables for use by SharePlex.

Guidelines for use

- Run the **PG Setup** utility on PostgreSQL instances in the SharePlex replication configuration.
- Within a server cluster, run the **PG Setup** utility on the node to which the shared disk that contains the SharePlex variable-data directory is mounted.
- For consolidated replication, run the **PG Setup** utility for each variable-data directory.
- Provide a DSN (data source name).

Connection Type on Linux	What to do
DSN	If you have a DSN defined, and you want to use it for the SharePlex connection, copy or link the ODBC files in which that DSN is defined (odbc.ini and odbcinst.ini) to the **odbc** subdirectory of the SharePlex variable-data directory. This prevents connection errors when the SharePlex processes connect to the database. If you do not have a DSN defined but want to use one, you can create it in the template files provided in the **odbc** subdirectory.

For an example of the sample `odbc.ini` and `odbcinst.ini` files, see the example below:

```
[postgres_rds_1]
Description=PostgreSQL connection RDS test
Driver=PostgreSQL
Database=postgres
Servername=pslpgdbrdsdb01.cihp157rpcvu.us-west-1.rds.amazonaws.com
Port=5432
UserName=postgres
Password=Quest4me
SSLmode=require
```

```
[qarun@pslinuxpgsp11 ~]$ more /splex/pooja/1100B111/var/odbc/odbc.ini
[testdb]
Driver = PostgreSQL
Database = testdb
Servername = 10.250.40.27
Port = 5432
Uid=splex
Pwd=splex
TrustServerCertificate=Yes
Connection Timeout=30
Trusted_Connection=Yes

[qarun@pslinuxpgsp11 ~]$ more /splex/pooja/1100B111/var/odbc/odbcinst.ini
# Example driver definitions

# Driver from the postgresql-odbc package
# Setup from the unixODBC package
[PostgreSQL]
Description     = ODBC for PostgreSQL
Driver          = /usr/lib/psqlodbcw.so
Setup           = /usr/lib/libodbcpsqlS.so
Driver64        = /usr/lib64/psqlodbcw.so
Setup64         = /usr/lib64/libodbcpsqlS.so
FileUsage       = 1


# Driver from the mysql-connector-odbc package
# Setup from the unixODBC package
[MySQL]
Description     = ODBC for MySQL
Driver          = /usr/lib/libmyodbc5.so
Setup           = /usr/lib/libodbcmyS.so
Driver64        = /usr/lib64/libmyodbc5.so
Setup64         = /usr/lib64/libodbcmyS.so
FileUsage       = 1
[qarun@pslinuxpgsp11 ~]$
```

Required Privilege

Review the following requirements to ensure that the setup succeeds.

- The Database Setup utility must be run as a PostgreSQL Administrator in order to grant SharePlex the required privileges to operate on the database and to create the SharePlex database account.
- (Symfoware only) If the person who is running the setup utility is not a Fujitsu Enterprise PostgreSQL owner, set the environment variable LD_LIBRARY_PATH to include the path to the **lib** subdirectory in the Fujitsu Enterprise PostgreSQL installation directory. The LD_LIBRARY_PATH is set in the **.bash_profile** file of the Fujitsu Enterprise PostgreSQL owner.

  Example:

  export LD_LIBRARY_PATH= /opt/symfoserver64/lib:$LD_LIBRARY_PATH

  If you do not set this path, the following error occurs:

  ```
  symbol lookup error: /opt/fsepv95client64/odbc/lib/psqlodbca.so: undefined
  symbol: PQconnectdbParams
  ```

- Cloud installations:
  - Common restrictions on privileges in cloud-hosted database services make it difficult for the setup utility to succeed in every possible type of scenario. To ensure that the database setup succeeds, only use the setup utility for the following purposes: To do a first-time database setup with a new SharePlex user, or, to modify an existing SharePlex user that either owns the database or has access to it.
  - You might need to grant usage/privileges on the target schema and tables to the SharePlex user, as shown in the following example:

    Log in as the schema owner, then issue the following commands:

    grant usage on schema *schema_name* to *user_name*;

    grant all privileges on all tables in schema *schema_name* to *user_name*;
  - For Google Cloud database, grant the following privileges if a user wants to change the owner of a table in replication:

    ```
    alter user splex_user createdb createrole login;

    create role role_name;

    grant role_name to splex_user;
    ```

Run PG setup for PostgreSQL

Follow the below instructions to run the PG setup for PostgreSQL:

1. Shut down any running SharePlex processes and **sp_cop** on the source system.
2. Run the **pg_setup** program from the **bin** subdirectory of the SharePlex product directory.

   IMPORTANT! If you installed the SharePlex instance 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.

   $ /users/splex/bin> pg_setup -p9400

Table 14: Setup prompts and responses

Prompt	Response
(Linux) `Enter the PostgreSQL DSN name:`	Enter a DSN.
`Is DB hosted over "Azure Single Server"?`	Enter "**n**" for a database hosted on the platform other than "Azure single server". **NOTE:** Ensure Admin users have the below mentioned roles before initiating setup-

Prompt	Response
	On-Prem DB: `superuser` **AWS hosted DB:** `rds_superuser` **Azure hosted DB:** `azure_pg_admin`
`Enter the PostgreSQL Administrator name :`	Enter the name of the PostgreSQL Administrator. This user will perform the work on the SharePlex account.
`Enter the password for the Administrator account :`	Enter the password of the Administrator.
`Enter the database name:`	Enter the name of the database that you want to contain the SharePlex tables and other objects for use by SharePlex. You can enter the name of a new or existing database.
`Database name` *`database`* `does not exist. Would you like to create it? [y] :`	If this prompt is displayed, the specified database does not exist. Press **Enter** to have the setup utility create it for you.
`Would you like to create a new SharePlex user [y]:`	Press **Enter** to accept the default to create a new SharePlex database user account and schema of the same name in the specified database, or enter **n** to use an existing SharePlex account.
`Enter the name of the new SharePlex user:` `Enter the name of the existing SharePlex user:`	One of these prompts is displayed depending on whether you elected to create a new user or use an existing user. Enter the name of the SharePlex user. **NOTE:** Enter the user name in lowercase letters only.
`Enter the password of the SharePlex user :`	Enter the password of the SharePlex user account.
`Re-enter the password for the SharePlex user :`	This prompt is only shown if you created a new user. Enter the SharePlex password again.
`Will this database be used as a source? [n] :`	Enter **Y** to use the DB connection for a source. You will see the below prompts only for the DB connection for the source. Enter **N** to use the DB connection for a target.
`Enter the replication slot name? [test_slot] :`	Enter the replication slot name. **NOTE:** Replication slot `test_slot` will be created in database after successful config activation.
`Enter the default tablespace for use by SharePlex [pg_default] :`	Enter the default tablespace by SharePlex

Prompt	Response
`Enter the index tablespace for use by SharePlex [pg_default] :`	Enter the index tablespace by SharePlex
`Do you want Logical replication? [n] :`	This prompt is displayed only for on-premise hosted PostgreSQL database. Enter "y" to perform logical replication. Enter "n" to perform physical replication. Setup will now install SharePlex objects.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for PostgreSQL database configuration

SharePlex User name: splexab

Database name: testdb

-- The datasource identifier in the SharePlex configuration is 'r.testdb' --
```

IMPORTANT!

Applicable only for source configuration:

SharePlex holds the WAL logs when configuration is not active. In SharePlex 11.0, when `pg_setup` is executed, it creates a replication slot. From SharePlex 11.1 onwards, slot is created while activating configuration. This replication slot maintains the latest LSN value of the WAL records that Capture has received and processed. Also, it holds all the WAL files from getting removed or archived, starting from the stored LSN in this slot.

In SharePlex 11.1 onwards, the replication slots are dropped while deactivating the configuration.

If SharePlex is shut down without deactivating the config file, then run the `pg_cleansp` utility to drop the slot. This utility also deletes slot names present in connections.yaml from the installation of version 11.0.

The following steps need to be performed if you do not want to run `pg_cleansp`:

1. Get the replication slot name from the `var/data/connections.yaml` file.
2. Check if the replication slot exists in the database using the below query:

   ```
   select count(*) FROM pg_replication_slots where slot_name = <slot_name>;
   ```

3. If a replication slot exists, then delete that replication slot from the database using the below query:

   ```
   select pg_drop_replication_slot(<slot_name>);
   ```

4. If you want to activate the config file, then first execute the `pg_setup` utility, which will create the replication slot again, and then activate the config file.

Support for hyphen, dot, and plus (-, ., and +) special characters

When the database name contains a hyphen, dot, or plus (-, ., or +) special characters, then remove these characters. Append the `_<count>` value to the identifier, where the count depends on the datasource identifier existing in the **connections.yaml** file.

Example:

- For the database name `aaa.db`, the datasource identifier will be `r.aaadb_1`.
- For the database name `aaa+db`, the datasource identifier will be `r.aaadb_2`, as `r.aaadb_1` already points to the `aaa.db` database.

If the database does not contain any special characters but its entry exists in the **connections.yaml** file, append `_<count>` to its identifier.

Example:

If the database name is `aaadb_1` and its entry exists in the **connections.yaml** file, and the database name is different, the identifier will become `r.aaadb_1_1`.

If the database name does not contain a hyphen, dot, or plus character, then the database name is the same as the datasource identifier.

Configure SharePlex to support PostgreSQL database on Azure platform

You can configure SharePlex to support the PostgreSQL database on the Azure platform using the below steps:

1. Add the DSN entry in the `Odbc.ini file at /etc/odbc.ini` and the `…/var/odbc` directory.
2. Copy the `/etc/odbcinst.ini` file to the `…/var/odbc` directory.
3. Run the PostgreSQL setup utility.

   This process creates the SharePlex tables, login and user accounts needed to run SharePlex replication.

NOTE: While performing DML operations in bulk for PostgreSQL PaaS (Azure), a performance slowness issue is observed.

Table 15: Setup Prompts and Responses

Prompt	Response
`Enter the PostgreSQL DSN name or connection string [] :`	Enter the PostgreSQL DSN name or a connection string that connects to the PostgreSQL database system. Example: **pslazpgdb02**
`Is DB hosted over "Azure Single Server"?`	Enter "Y" only when PostgreSQL DB is hosted over 'Azure Single Server'.
`Enter the PostgreSQL Administrator name :`	Enter the name of the PostgreSQL Administrator. This user will perform the setup work on the SharePlex account and schema. Example: **PostgreSQL**
`Enter the password for the Administrator account:`	Enter the password of the Administrator.
`Enter the replication target database name :`	Enter the target database name on which replication needs to be performed. Example: **sp_mk**
`Would you like to create a new SharePlex user?` `Enter the name of the existing SharePlex user:`	N (If you want to create a user, enter “y”) Enter the name of the SharePlex user.
`Enter the password of the SharePlex user:`	Enter the password of the SharePlex user account.

Prompt	Response
`Re-enter the password for the SharePlex user:`	This prompt is only shown if you created a new user. Enter the SharePlex password again.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for PostgreSQL database configuration

SharePlex User name: postgres@pslazpgdb02.postgres.database.azure.com

Database name: sp_mk

The data source identifier in the SharePlex configuration is 'r.sp_mk' –
```

Database Setup for PostgreSQL Database as a Service

Pre-requisites to configure PostgreSQL database setup as a source:

1. Modify the cloud database parameters :

   AWS : `rds.logical_replication 1`

   Azure : `wal_level LOGICAL`

2. Add database connection details in the ODBC files in SharePlex `var/odbc` directory. See the sample screenshot.

3. For all the tables which are going to be part of replication, set the 'replica identity' to 'full' before activation.
   For example: `alter table "DEMO_SRC" replica identity full;`

NOTES:

- SharePlex PostgreSQL cloud service as a source only supports logical replication.
- For controlled character set conversions between source and target, ensure that the PostgreSQL Database character set value and the `PGCLIENTENCODING` database environment parameter value are exactly the same as those specified in the source and target database character set settings.

Overview

Run the PG Setup utility for PostgreSQL (**pg_setup**) to establish a user account, schema, and tables for use by SharePlex.

Guidelines for Use

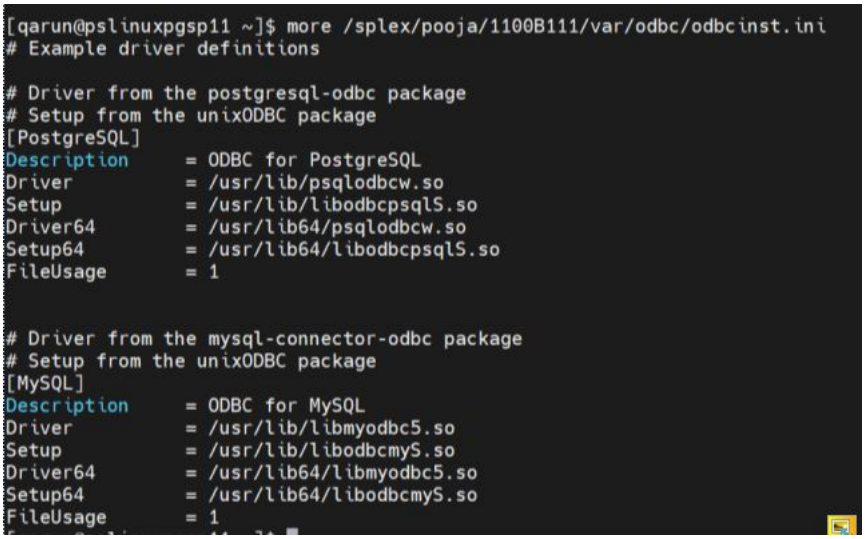
- Run the **PG Setup** utility on an independent Linux VM from where the PostgreSQL instance is accessible.
- For consolidated replication, run the **PG Setup** utility for each variable-data directory.
- For PostgreSQL Database as a Service, follow the connection handling best practices recommended by the respective vendor.

- You can provide a DSN (data source name). Note the following when deciding on the connection method on Linux:

Connection Type on Linux	What to do
DSN	If you have a DSN defined, and you want to use it for the SharePlex connection, copy or link the ODBC files in which that DSN is defined (odbc.ini and odbcinst.ini) to the **odbc** subdirectory of the SharePlex variable-data directory. This prevents connection errors when the SharePlex processes connect to the database. If you do not have a DSN defined but want to use one, you can create it in the template files provided in the **odbc** subdirectory.

For an example of the sample `odbc.ini` and `odbcinst.ini` files, see the example below:

```
[postgres_rds_1]
Description=PostgreSQL connection RDS test
Driver=PostgreSQL
Database=postgres
Servername=pslpgdbrdsdb01.cihp157rpcvu.us-west-1.rds.amazonaws.com
Port=5432
UserName=postgres
Password=Quest4me
SSLmode=require
```



Required Privilege

Review the following requirements to ensure that the setup succeeds.

- The Database Setup utility must be run as a PostgreSQL Administrator in order to grant SharePlex the required privileges to operate on the database and to create the SharePlex database account.
- (Symfoware only) If the person who is running the setup utility is not a Fujitsu Enterprise PostgreSQL owner, set the environment variable LD_LIBRARY_PATH to include the path to the **lib** subdirectory in the Fujitsu Enterprise PostgreSQL installation directory. The LD_LIBRARY_PATH is set in the **.bash_profile** file of the Fujitsu Enterprise PostgreSQL owner.

  Example:

  export LD_LIBRARY_PATH= /opt/symfoserver64/lib:$LD_LIBRARY_PATH

  If you do not set this path, the following error occurs:

  ```
  symbol lookup error: /opt/fsepv95client64/odbc/lib/psqlodbca.so: undefined
  symbol: PQconnectdbParams
  ```

- Cloud installations:
  - Common restrictions on privileges in cloud-hosted database services make it difficult for the setup utility to succeed in every possible type of scenario. To ensure that the database setup succeeds, only use the setup utility for the following purposes: To do a first-time database setup with a new SharePlex user, or, to modify an existing SharePlex user that either owns the database or has access to it.
  - You might need to grant usage/privileges on the target schema and tables to the SharePlex user, as shown in the following example:

    Log in as the schema owner, then issue the following commands:

    grant usage on schema *schema_name* to *user_name*;

    grant all privileges on all tables in schema *schema_name* to *user_name*;

  - For Google Cloud database, grant the following privileges if a user wants to change the owner of a table in replication:

    ```
    alter user splex_user createdb createrole login;

    create role role_name;

    grant role_name to splex_user;
    ```

Run PG setup for PostgreSQL

Follow the below instructions to run the PG setup for PostgreSQL:

1. Shut down any running SharePlex processes and **sp_cop** on the source system.
2. Run the **pg_setup** program from the **bin** subdirectory of the SharePlex product directory.

   IMPORTANT! If you installed the SharePlex instance 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.

   $ /users/splex/bin> pg_setup -p9400

Table 16: Setup prompts and responses

Prompt	Response
`Enter the PostgreSQL DSN name:`	Enter a DSN.
`Is DB hosted over "Azure Single Server"?`	Enter "**n**" for a database hosted on the platform other than "Azure single server". **NOTE:** Ensure Admin users have the below mentioned roles before initiating setup- **On-Prem DB:**`superuser` **AWS hosted DB:**`rds_superuser` **Azure hosted DB:**`azure_pg_admin`
`Enter the PostgreSQL Administrator name :`	Enter the name of the PostgreSQL Administrator. This user will perform the work on the SharePlex account.
`Enter the password for the Administrator account :`	Enter the password of the Administrator.
`Enter the database name:`	Enter the name of the database that you want to contain the SharePlex tables and other objects for use by SharePlex. You can enter the name of a new or existing database.
`Database name` *`database`* `does not exist. Would you like to create it? [y] :`	If this prompt is displayed, the specified database does not exist. Press **Enter** to have the setup utility create it for you.
`Would you like to create a new SharePlex user [y]:`	Press **Enter** to accept the default to create a new SharePlex database user account and schema of the same name in the specified database, or enter **n** to use an existing SharePlex account.
`Enter the name of the new SharePlex user:`	One of these prompts is displayed depending on whether you elected to create a new user or use an existing user.

Prompt	Response
`Enter the name of the existing SharePlex user:`	Enter the name of the SharePlex user. **NOTE:** Enter the user name in lowercase letters only.
`Enter the password of the SharePlex user :`	Enter the password of the SharePlex user account.
`Re-enter the password for the SharePlex user :`	This prompt is only shown if you created a new user. Enter the SharePlex password again.
`Will this database be used as a source? [n] :`	Enter **Y** to use the DB connection for a source. You will see the below prompts only for the DB connection for the source. Enter **N** to use the DB connection for a target.
`Enter the replication slot name? [test_slot] :`	Enter the replication slot name. **NOTE:** Replication slot `test_slot` will be created in database after successful config activation.
`Enter the default tablespace for use by SharePlex [pg_default] :`	Enter the default tablespace by SharePlex
`Enter the index tablespace for use by SharePlex [pg_default] :`	Enter the index tablespace by SharePlex
`Do you want Logical replication? [n] :`	This prompt is displayed only for on-premise hosted PostgreSQL database. Enter "y" to perform logical replication. Enter "n" to perform physical replication. Setup will now install SharePlex objects.

A successful setup terminates with a message similar to the following:

```
Completed SharePlex for PostgreSQL database configuration

SharePlex User name: splexab

Database name: testdb

-- The datasource identifier in the SharePlex configuration is 'r.testdb' --
```

Installation of pg_hint_plan extension

To install pg_hint_plan:

For On-Prem Databases:

1. Install the RPM according to the PostgreSQL version support matrix at https://github.com/ossc-db/pg_hint_plan.
2. Run the query with the user [Shareplex user]: `CREATE EXTENSION IF NOT EXISTS pg_hint_plan;`
3. Reboot the instance.

For Cloud PostgreSQL:

1. Set the static parameter in the parameter group: `shared_preload_libraries=pg_hint_plan`.
2. Reboot the instance.

NOTE: The total number of background workers that can exist at any one time is limited by both **max_worker_processes** and **max_parallel_workers**:

- **max_worker_processes** is a database cluster-level parameter and should be set the same as the available CPU vCore.
- **max_parallel_workers** is a session-level parameter and should be set less than or equal to max_worker_processes.

Solve Installation Problems for PostgreSQL

Contents

Solve database setup problems for PostgreSQL

Solve database connection problems for PostgreSQL

Overview

This chapter reviews some common problems that you could experience when installing or running SharePlex for the first time after installation.

Sometimes there are special installation instructions that supersede or supplement certain instructions in this manual. In addition, there can be known issues for this version that you should be aware of during or after installation. Please read the Release Notes for the version of SharePlex that you are installing before you begin the installation process.

Solve database setup problems for PostgreSQL

This section helps you diagnose problems that are associated with the SharePlex database account and connection information that was created with the Database Setup utility when SharePlex was installed on the system.

NOTE: For more information about Database Setup, see Database Setup Utilities in the SharePlex Reference Guide.

If the issue you are experiencing is not listed in this documentation, search the SharePlex Knowledge Base at:

https://support.quest.com.

The Knowledge Base provides filtering options and links to other resources that can help you use and troubleshoot SharePlex.

PostgreSQL setup issues

Problem	Description	Solution
Insufficient database privileges	If the database setup utility fails, the person who runs it may not have the correct privileges	For more information, see Database Setup Utilities on page 130.
sp_cop is running	The SharePlex**sp_cop** process cannot be running while you are running the Database Setup utility.	If it is running, shut it down using the **shutdown** command in **sp_ctrl.** Run **sp_ctrl** from the bin sub-directory in the SharePlex product directory.
ODBC driver configuration problem	*Error: [unixODBC][Driver Manager]Data source name not found and no default driver specified*. The issue is observed while running pg-setup.	Ensure that the `odbc.ini` and `odbcinst.ini` configurations are verified. For more information, see Database Setup for PostgreSQL.

Solve database connection problems for PostgreSQL

Did you verify the credentials?

If SharePlex cannot connect to a source or target database, you can view the login credentials that are being used for that database by using the **connection** command with the **show** option. For example:

```
sp_ctrl> connection r.mydb show
```

You can also view connection settings in the **connections.yaml** file, which is stored in the **data** sub-directory of the SharePlex variable-data directory. If there are no settings in this file, it means that the SharePlex database setup procedure was not performed on this database.

To view the correct database setup procedure for the database, see SharePlex Reference Guide.

You can use the **connection** command to update connection properties. For more information, see the SharePlex Reference Guide.

Did you assign a DBA role to the SharePlex PostgreSQL user?

The SharePlex PostgreSQL user requires a DBA role with unlimited privileges. The SharePlex user is created with the default PostgreSQL profile under the assumption that the profile has the unlimited resource privileges assigned by PostgreSQL as the default. If SharePlex is unable to interact with PostgreSQL, check to see if the default was changed. If so, assign SharePlex a DBA role with unlimited privileges for all definitions.

3

Installing SharePlex on a Docker container

You can install SharePlex on a Docker container manually or through a Dockerfile.

To install SharePlex on a Docker container through a Dockerfile, follow these steps:

1. Create a docker build file in the following format (This is a sample Docker file format created for the Linux platform):

```
# -------- Pull RHEL UBI ------------------------

ARG UBI_MINIMAL_VERSION="latest"

FROM --platform=linux/amd64 registry.access.redhat.com/ubi8/ubi:${UBI_
MINIMAL_VERSION}

LABEL maintainer="<maintainer email address>"

# -------- Set Environment ------------------------

ENV HOME_DIR=/home/splex \

BASE_DIR=/u01/splex \

SP_SYS_PRODDIR=/u01/splex/prod \

SP_SYS_VARDIR=/u01/splex/var \

SP_COP_TPORT=2100 \

PATH=$SP_SYS_PRODDIR/bin:$PATH \

ODBCINI=$SP_SYS_VARDIR/odbc/odbc.ini \

ODBCSYSINI=$SP_SYS_VARDIR/odbc

# -------- Set ROOT Password ------------------------

RUN echo "root:****" | chpasswd

# -------- Install CPIO, PostgresSQL ODBC ------------------------

RUN yum -y install cpio hostname util-linux-user procps --nogpgcheck

RUN yum install -y --nogpgcheck openssh-server openssh-clients

RUN /usr/bin/ssh-keygen -A

RUN /usr/sbin/sshd
```

```
RUN yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-
7-x86_64/pgdg-redhat-repo-latest.noarch.rpm

RUN yum install -y postgresql14-odbc --nogpgcheck

RUN yum install -y https://yum.oracle.com/repo/OracleLinux/OL8/appstream/x86_
64/getPackage/ksh-20120801-257.0.1.el8.x86_64.rpm --nogpgcheck

RUN yum install -y unixODBC-devel.x86_64 unixODBC.x86_64 postgresql15-
odbc.x86_64 --nogpgcheck

RUN cd /usr/lib64;ln -s /usr/pgsql-15/lib/psqlodbcw.so psqlodbcw.so

RUN cd /usr/lib;ln -s /usr/pgsql-15/lib/psqlodbcw.so psqlodbcw.so

# -------- Add Group and User ------------------------

RUN groupadd -g 54321 spadmin

RUN useradd -g spadmin -md /home/splex splex

RUN echo "splex:*****" | chpasswd

# RUN chown splex:spadmin ${BASE_DIR}

# -------- Copy SharePlex Binaries, Reponse file, Installtion script ---------
---------------

COPY --chown=splex:spadmin --chmod=777 SharePlex.tpm ${BASE_DIR}/

COPY --chown=splex:spadmin --chmod=777 product.rsp ${BASE_DIR}/

COPY --chown=splex:spadmin --chmod=777 installsplex.sh ${BASE_DIR}/

# -------- Change User and Install SharePlex ------------------------

USER splex

# CMD [ "/u01/splex/SharePlex-11.0.0-b1-c048c979-ONEOFF-SPO-24709-rhel-amd64-
m64.tpm -r /u01/splex/product.rsp" ]

RUN /u01/splex/installsplex.sh

RUN rm ${BASE_DIR}/installsplex.sh ${BASE_DIR}/product.rsp ${BASE_
DIR}/SharePlex.tpm

EXPOSE 22

VOLUME [ "${BASE_DIR}" ]
```

2. Create a SharePlex install script file similar to the below sample file: (This is a sample SharePlex install script file format created for the Linux platform)

```
#!/bin/bashset -e

echo 'export SP_HOME=/u01/splex

export SP_SYS_HOST_NAME=$(hostname)

export SP_SYS_PRODDIR=${SP_HOME}/prod

export SP_SYS_VARDIR=${SP_HOME}/var

export SP_SYS_BINDIR=${SP_SYS_PRODDIR}/bin

export SP_COP_TPORT=4545

export SP_COP_UPORT=4545

export PATH=${SP_SYS_BINDIR}:${PATH}

alias spbin="cd ${SP_SYS_BINDIR}/"

alias spvar="cd ${SP_SYS_VARDIR}/"

alias splog="cd ${SP_SYS_VARDIR}/log"

alias spc="${SP_SYS_BINDIR}/sp_ctrl"

alias gocop="${SP_SYS_BINDIR}/sp_cop -u${SP_COP_TPORT} &"

export ODBCINI=${SP_SYS_VARDIR}/odbc/odbc.ini

export ODBCSYSINI=${SP_SYS_VARDIR}/odbc

' >> /home/splex/.bashrc

/u01/splex/SharePlex.tpm -r /u01/splex/product.rsp
```

3. Create SharePlex silent install file (Examples: **product.rsp**) used for silent `ora_setup`. Not applicable for PostgreSQL. For additional information, see Run database setup for Oracle in silent mode section.
4. Install Docker (Example: Docker desktop).
5. Choose a base image (Example: CentOS).
6. Install necessary packages (Example: PostgreSQL ODBC driver / Oracle Instant Client).
7. Add custom files to the project directory (SharePlex files).
8. Specify the user that will run your container (Example: splex:spadmin).
9. Specify the exposed ports (Example: 2100).
10. Define the entry point (Example: bash).
11. Define a configuration method (Example: environment variables / scripts).
12. Externalize your data (define an external volume for persistent data).

Build and run SharePlex container

1. Create the following files and and copy them to the project directory (Example: splex_to_container) :
   - **dockerfile** (Docker build file)
   - **installsplex.sh** (SharePlex install script)
   - **product.rsp** (SharePlex silent install file)
   - **SharePlex.tpm** (SharePlex TPM installer)
2. From within this directory, run the following commands:
   a. `docker build -t shareplex:11`

      This command will build the SharePlex image with the name "shareplex" and tag "11".

   b. `docker run -itd --privileged -d --memory-reservation=100m --expose 22 --ulimit nofile=2000:2000 -p 4545:4545 -p 80:22 --name shareplex -it shareplex:11.4`

      This command will start the container from the image "shareplex:11" named "shareplex."

   c. `docker exec -it shareplex bash`

      This command will start an interactive terminal session into the running SharePlex container. You can now run a database setup utility.

4

Assign SharePlex Users to Security Groups

Contents

About the SharePlex Security Groups

Create and Populate SharePlex Groups on Unix and Linux

Overview

The SharePlex security groups provide access control to the SharePlex command and control system. Without proper configuration of these groups, anyone with permissions on the system can use the commands that view, configure, and control data replication.

About the SharePlex Security Groups

To monitor, control, or change SharePlex replication, a person must be assigned to one of the SharePlex security groups on the systems where he or she will be issuing commands. Each group corresponds to an authorization level, which determines which SharePlex commands a person can issue. To execute a command, a user must have that command's authorization level or higher. Use the **authlevel** command to determine your authorization level for issuing SharePlex commands on a system.

Description of the SharePlex security groups

Refer to the following table to determine the group and authorization level that you want to grant each SharePlex user.

User Authorization Levels and Roles

Auth level	User type	User group	User roles
1	Administration	spadmin*	You need at least one user with Administrator rights on each source and target system. Can issue all SharePlex commands. Commands that can *only* be issued by a SharePlex Administrator are: • **startup, shutdown** • all configuration commands relating to an active configuration • all parameter commands except **list param** • **start capture** • **stop capture** • **abort capture** • **truncate log** The SharePlex Administrator user must be in the Oracle **dba** group. For Oracle RAC and ASM 11*g*R2 and above, the user must also be in the Oracle Inventory group. For example: **$ useradd –g spadmin –G dba,oinstall**. The membership in Oracle Inventory group must be listed explicitly in the **etc/group** file. On Unix and Linux, unless you install SharePlex as a root user, the SharePlex Administrator user and the SharePlex **admin** group must exist prior to installation.
2	Operator	spopr	Can issue all SharePlex commands except those listed above.
3	Viewer	spview	Can view lists, status screens, and logs to monitor replication only.

NOTE: The default name for the SharePlex administrator group is **spadmin**, but you can designate any group or specify any name for that group during installation.

Create and Populate SharePlex Groups on Unix and Linux

Where and when to create the SharePlex groups on Unix and Linux depends on whether you install SharePlex as a root or non-root user.

- If you install as non-root, create the groups in the **/etc/group** file before you run the SharePlex installer. In a cluster, create them on all nodes.*
- If you install SharePlex as a root user, you can direct the installer to create the groups in the **/etc/group** file. If you install in a cluster, the installer creates the groups on the primary node, but you must create them yourself on the other nodes.

  * The groups must exist because the installer adds the SharePlex Administrator user to the **spadmin** group during the installation process. In a cluster, this user is only added to the primary node. You must add the SharePlex Administrator user to the other nodes.

To create the groups in /etc/group :

 # groupadd spadmin

 # groupadd spopr

 # groupadd spview

To assign a user to a group:

1. Open the **/etc/group** file.
2. Add the Unix or Linux user name to the appropriate group. To assign a list of user names to a group, use a comma-separated list (see the following example).

   ```
   spadmin:*:102:spadmin,root,jim,jane,joyce,jerry
   ```

   If the password field is null, no password is associated with the group. In the example, the asterisk (*) represents the password, “102” represents the numerical group ID, and **spadmin** is the group. The group ID must be unique.
3. Save the file.

Users can verify their authorization levels by issuing the **authlevel** command in **sp_ctrl**.

5

Solve Installation Problems for SharePlex

Contents

Solve Installer Problems

Solve SharePlex Startup Problems

Solve sp_ctrl Problems

Solve Host Connection Problems

Overview

This chapter reviews some common problems that you could experience when installing or running SharePlexfor the first time after installation.

Sometimes there are special installation instructions that supersede or supplement certain instructions in this manual. In addition, there can be known issues for this version that you should be aware of during or after installation. Please read the Release Notes for the version of SharePlex that you are installing before you begin the installation process.

Solve Installer Problems

Is sp_cop shut down?

If you installed SharePlex on this system before, and you are re-installing it, the installation will return errors if SharePlex is running on this system. Shut down SharePlex using the **shutdown** command in **sp_ctrl**. If you are unable to run **sp_ctrl**, or if any SharePlex processes will not die, locate the process (using **ps -ef | grep sp_** on Unix and Linux systems) and kill it. When all SharePlex processes have been killed, run the installation program again.

Are all systems connected to the network?

Check to see that all systems on which you are loading SharePlex are connected to the network. The network node name and IP address of each system must be established sufficiently to allow SharePlex to perform TCP operations, even though the target machines themselves are not yet configured.

Did you enter the SharePlex groups in the name service?

If your environment uses a name service such as NIS or NISPLUS, you need to add the SharePlex groups and services to the nameserver before you run the SharePlex installation program, and the SharePlex Administrator must be named in the ***SharePlex Admin group*** on the nameserver before you install SharePlex. Instructions are

on page 31. If these procedures are not performed, the installation will generate an error at the point in which it attempts to verify that the groups exist.

Is the database open?

The database must be open while you are installing SharePlex.

Solve SharePlex Startup Problems

Was the user an authorized SharePlex user?

Only a member of the SharePlex administrator group (default name is **spadmin**) can start **sp_cop**. A root user that is not a member of this group can start **sp_cop**, but no users (including root) will be able to connect through **sp_ctrl** to issue commands. For more information, see Assign SharePlex Users to Security Groups on page 237.

Was the filesystem mounted as nosuid?

On Unix and Linux systems, if the filesystem is mounted as nosuid, SharePlex must be started by the installation owner. In this case, members of the SharePlex administrator group (**spadmin** by default), other than the installation owner, will not be able to run SharePlex.

Solve sp_ctrl Problems

Did you assign the users to the SharePlex groups?

Only one SharePlex user, the Administrator who owns the SharePlex binaries and files, is created during SharePlex installation. Other users must be assigned to the appropriate SharePlex user groups. These groups control the authorization levels for various SharePlex functions.

To issue a specific command (such as **activate config** or **stop export**), a user must have that command's authorization level or higher. For example, a SharePlex Administrator (authorization level 1) can issue any command, but a member of the **spview** group can only issue status commands and a few other commands that do not directly affect the replication processes.

For more information, see Assign SharePlex Users to Security Groups on page 237.

Solve Host Connection Problems

If SharePlex cannot resolve a host name, try creating an alias for it using a simple alphanumeric name.

Map the name to the alias in the following locations:

- Network: The NIS and DNS servers
- UNIX: Local **/etc/hosts** file

In the hosts files, put each entry on an individual line. The following is an example, where **sysA** and **sysB** are aliases:

```
111.22.33.44    sysA.company.com    sysA    # source system
55.66.77.88     sysB.company.com    sysB    # target system
```

6

Remove SharePlex from a System

Contents

Remove SharePlex from Unix/Linux

Overview

This section contains instructions for using the SharePlex uninstallation program to remove SharePlex from a system. The uninstaller permanently removes the replication environment from the system.

To preserve the replication environment, including the queues that store the data, you can install a SharePlex upgrade or reinstall the current version, rather than uninstall SharePlex. Before you upgrade or reinstall SharePlex, see the Release Notes for the version you are installing to determine if there are any special upgrade or installation requirements.

Remove SharePlex from Unix/Linux

Perform the following steps to remove SharePlex from Unix or Linux platforms:

1. Log on as a user who has privileges to remove the SharePlex directories.
2. Run **sp_ctrl**.
3. Issue the **shutdown** command to shut down SharePlex.

   ```
   sp_ctrl>  shutdown
   ```

4. Exit **sp_ctrl**.
5. Use the following command to make certain that no SharePlex processes are running.

   # ps -ef | grep sp_

6. Kill any processes that are still running.
7. Remove the SharePlex product directory, variable-data directory, and the hidden .shareplex directory from the system.

NOTE: If there are multiple installations, do not remove the hidden directory.

8. Drop the SharePlex database user from the database.

NOTE: If desired, remove any objects that SharePlex installs in the database. These objects begin with `SHAREPLEX_` or `shareplex_`. You can remove any such objects using standard SQL commands.

IMPORTANT!

Points to remember when removing the SharePlex objects from the PostgreSQL database:

- Remove the replication slot that was created during `pg_setup` using the following query:

  ```
  select pg_drop_replication_slot(<slot_name>);
  ```

- Remove replication origins created by the Post-process:
  - Get replication origin names using the following query:
    ```
    Select a roname from pg_replication_origin that is similar to "sp_
    post_%";
    ```
  - Remove these origins with the following query:
    ```
    select pg_replication_origin_drop(<origin_name>);
    ```
- If the **SP_CAP_DENIED_SESSION_PG** parameter was used, remove the replication origin created:
  - Get replication origin names using the following query:
    ```
    Select a roname from pg_replication_origin that is similar to "sp_
    deny_%";
    ```
  - Remove these origins with the following query:
    ```
    select pg_replication_origin_drop(<origin_name>);
    ```

A

Advanced SharePlex Installer Options

The use of additional command line options when installing SharePlex is usually not necessary. These options are typically employed when working with Support to resolve specific issues.

The installer command line options and their descriptions follow:

```
USAGE

     tpm [<options>] [ [<package> | <location>] ... ]


OPTIONS

          -v, --verbose Turns verbose mode on

     -h, -?, --help Prints out this message

               --debug Starts the interactive debugger

               --info Print information about installed


packages

               --install Perform product installation

               --remove Perform product deinstallation

               --commit Commit last installation

               --revert Revert last installation

          -t, --tmp <directory> Temporary directory location

     -d, --directory <directory> Working directory

     -f, --force Unconditionally update existing files

          --no-cleanup Do not perform cleanup on failure

          --nocleanup Same as --no-cleanup, for compatibility

          --list List the content of the archive

          --extract Extract the archive into the current directory

     -r, --responses <yaml file> Use the responses from a specified file

     -D, --defaults Accept default answers

     -l, --log Leave the installation log file
```

```
DESCRIPTION

Provides package management facilities. Packages can be installed, removed,
reverted or committed. The utility also figures out its role based on the
command name of its invocation path. For example, "tpm-install" is treated as
"tpm --install", "tpm-remove" as "tpm --remove", etc.

It can also be invoked as part of a self extracting package invocation, in
which case it is treated as "tpm --install".
```

NOTE: All command line options for the .tpm file are preceded by two dashes.

B

Install SharePlex as Root

You can install SharePlex as a root user. When you install as a root user, the installer prompts you to select whether or not to create the SharePlex user groups. When the installer creates the groups, it adds the SharePlex Administrator user to the **spadmin** group. For more information about these groups, see Assign SharePlex Users to Security Groups on page 237.

In a cluster, the installer adds the SharePlex groups to the primary node, but you must add them to the other nodes yourself.

Additionally, see Network checklist on page 17 for instructions on adding the groups to a nameserver.

To install as root:

1. Log in to the system as a root user.
2. Copy the SharePlex installer file to a temporary directory where you have write permissions. The installer file has a naming format of:

   `SharePlex-release#-platform.tpm`

3. Change the permissions of the file as follows:

   `# chmod 555 SharePlex-release#-platform.tpm`

4. Run the installer as directed in Install SharePlex on page 33.

C

SharePlex Installed Items

The following describes the items that are installed during the SharePlex setup process.

SharePlex directories

SharePlex uses two main directories:

The product directory: This is the SharePlex installation directory, where the SharePlex programs and libraries are stored.

The variable-data directory: This is the SharePlex working directory, where the queue files, log files and other components that comprise the current replication environment are stored.

NOTE: These directories are often referred to as *productdir* and *vardir*, respectively.

Do not remove, rename or edit any files or directories installed by SharePlex. Some directories contain hidden files that are essential for replication. Some files appear empty but must exist under their original names because they are referenced by one or more SharePlex processes. Some items in the directories are for use only under the supervision of Quest Technical Support.

Programs meant for general use in a production environment are documented in the published SharePlex documentation. If you do not find documentation for a program in a SharePlex directory, do not attempt to run it. Contact Quest Technical Support first.

Files and directories can vary from version to version of SharePlex, but the basic structure appears as follows.

SharePlex product directory

Sub-directory	Contents
BACKUP	Uninstall information
bin	SharePlex executable files
config	Internally used content.
data	Default parameter settings
doc	Catalog of exception messages
install	(Unix and Linux only) Scripts related to installation, licensing and upgrades
lib	SharePlex shared libraries
log	SharePlex log files
mks_oe	Runtime installation files for third-party software used by SharePlex.
util	SharePlex utilities

Sub-directory	Contents
.app-modules	(Unix and Linux only) Hidden internal directory that contains raw executables. Do not use the contents of this directory to launch processes.
.meta-inf	(Unix and Linux only) Hidden internal directory that contains meta information used during the installation process.

SharePlex variable-data directory

Sub-directory	Contents
config	Configuration files for this installation of SharePlex.
data	Status Database, configuration activation information, user-defined parameter settings, and other user-defined files that direct replication activities.
db	Configuration internal database for each activation of a configuration file.
downgrd	Information about SharePlex targets that are a lower version than the source.
dump	Core files (if a process fails)
log	SharePlex log files
rim	Queue files (working data files)
save	Information about active and inactive configurations.
state	Information about the current state of SharePlex when a configuration is active, such as the object and sequence caches.
temp	Used by the copy and append features and other SharePlex sync-related processes.
oos	Stores the transactions that contain out-of-sync operations when the SP_OPO_SAVE_OOS_TRANSACTION parameter is enabled.

SharePlex installed objects

Much of the replication process is controlled and tracked through a series of internal objects that are installed into the source or target database during the installation of SharePlex. They are essential for SharePlex to operate, so do not alter them in any way.

NOTE: Not all objects are used for all databases. Most are used for Oracle databases. If you do not see an object in your database, it is not relevant to the database, or the information is stored internally within the SharePlex configuration. If you see an object that is in your database but not in this list, it is not being used in the current release.

Table	Object type	Description
DEMO_SRC	Table	Used as the source table for the SharePlex demonstrations.
DEMO_DEST	Table	Used as the target table for the SharePlex demonstrations.
SHAREPLEX_ACTID	Table	Used by Capture to checkpoint its state.
SHAREPLEX_ANALYZE	Table	Used by the **analyze** command.
SHAREPLEX_CHANGE_OBJECT	Table	Used by users to stop and resume replication for an object.
SHAREPLEX_COMMAND	Table	Used for the **flush**, **abort** and **purge** commands.
SHAREPLEX_CONFIG	Table	Used by the activation and Capture processes to mark the start of a new activation.
SHAREPLEX_DATA	Table	Used by the SharePlex wallet for Oracle TDE replication.
SHAREPLEX_DATAEQUATOR	Table	Used by the **compare** and **repair** commands and the Post process to synchronize their operations.
SHAREPLEX_DATAEQUATOR_INSERT_TEMP	Table	Used as a temporary table by the **compare** and **repair** commands.
SHAREPLEX_DATAEQUATOR_UPDATE_TEMP	Table	Used as a temporary table by the **compare** and **repair** commands.
SHAREPLEX_DATAEQUATOR_DELETE_TEMP	Table	Used as a temporary table by the **compare** and **repair** commands.
SHAREPLEX_DDL_CONTROL	Table	Used to refine control of DDL that is enabled for replication by the SP_OCT_REPLICATE_ALL_DDL parameter.
SHAREPLEX_JOBID	Sequence	Used by the **sp_cnc** process and the **compare**, **repair**, and **copy** commands to provide a unique job ID.
SHAREPLEX_JOBS	Table	Used by the **sp_cnc** process and the **compare**, **repair**, and **copy** commands to store information about a job.
SHAREPLEX_JOB_STATS	Table	Used by the **sp_cnc** process and the **compare**, **repair**, and **copy** commands to store information about a job.

Table	Object type	Description
SHAREPLEX_JOBS_ CONFIG	Table	Used by the **disable jobs** and **enable jobs** commands.
SHAREPLEX_LOB_ CACHE	Table	Used by the Capture process when processing VARRAYs stored as LOB.
SHAREPLEX_LOBMAP	Table	Used by the Capture process to map LOBIDs and rows when a table with LOB columns does not have PK/UK logging enabled.
SHAREPLEX_LOGLIST	Table	Used by the Capture process to track inactive RAC instances.
SHAREPLEX_MARKER	Table	Used by the Read process when PK/UK logging is not enabled.
SHAREPLEX_OBJMAP	Table	Used by the activation and Capture processes to define the objects in replication.
SHAREPLEX_ PARTITION_CACHE	Table	Used by the Capture process to map Oracle partition IDs to tables in replication.
SHAREPLEX_SYNC_ MARKER	Table	Used by the **copy** command and the Read and Post processes to sync their operations.
SHAREPLEX_TRANS or SHAREPLEX_OPEN_ TRANS	Table	Used by the Post process to store checkpoints and to mark transactions that were applied in a primary-to-primary configuration.

D

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