



# Oracle Staging Push with TDE

## Staging Push

Staging Push introduces the concept of a staging instance for Oracle dSource ingestion. Storage for the staging instance is provided by the Delphix Engine via NFS. Existing backups are then used to restore the source database onto the Delphix provided storage, at which point a dSource Snapshot can be taken. Staging database enhances the data ingestion mechanism by providing the following:

- » Any backup vendor and backup location can be used to restore the backups.
- » The staging database can also be configured as a physical standby using Oracle Data Guard / Oracle Active Data Guard.
- » No access to the production environment or database is required at any point, providing true “zero production touch” capability.
- » Unlike with the Delphix OBI plugin, Staging Push snapshots can be provisioned in the exact same manner as regular Oracle snapshots.

An [existing whitepaper](#) explains the process of ingesting data from an Oracle database using Staging Push with sample scripts.

## Configure TDE for Staging Databases

Refer Oracle [documentation](#) to setup TDE for a staging database in different configurations. As an example, following steps can be used to configure TDE for a staging database associated with a Staging Push dSource.

Please note that in case of multitenant databases, a Staging Push CDB cannot have both encrypted and non-encrypted PDBs.

```
# Statement of Support
# This software is provided as-is, without warranty of any kind or
# commercial support
# through Delphix. See the associated license for additional details.
# Questions,
# issues, feature requests, and contributions should be directed to
# the community as
# outlined in the Delphix Community Guidelines.
```

1. Link a non-multitenant/CDB dSource. Refer the Delphix documentation [here](#) to link the dSource. On successful linking, a new staging Oracle instance will be created on the remote Oracle host..
2. In the staging Oracle installation, configure TDE keystore location in “sqlnet.ora”:

```
ENCRYPTION_WALLET_LOCATION=
  (SOURCE=
    (METHOD=FILE)
    (METHOD_DATA=
      (DIRECTORY=<staging keystore location>)))
```

For Oracle 18+ versions, the keystore location can also be configured by setting the "wallet\_root" system parameter:

```
ALTER SYSTEM SET WALLET_ROOT='<staging keystore location>'
SCOPE=BOTH SID='*';
ALTER SYSTEM SET TDE_CONFIGURATION="KESTORE_
CONFIGURATION=<keystore_type>" SCOPE=BOTH SID='*';
```

3. Create the keystore directory:

```
mkdir -p <staging keystore location>
```

4. Create an empty keystore at the keystore location:

```
SQL> administer key management create keystore '<staging keystore
location>' identified by "<staging keystore password>";
keystore altered.
```

5. Copy the source database keystore on the host where staging Oracle instance resides. Merge the source keystore into the newly created empty keystore for staging database.

```
SQL> administer key management merge keystore '<source keystore location>'
identified by "<source keystore password>" into existing keystore '<staging
keystore location>' identified by "<staging keystore password>" with backup;
keystore altered.
```

Note that this step must be executed for every PDB coming from a different source CDB.

6. Verify the status of the staging keystore:

```
SQL> select * from v$encryption_wallet;
WRL_TYPE      WRL_PARAMETER      STATUS
FILE          /u01/tde/CDOMLOSRE17C/      CLOSED
```

7. Open the staging keystore:

```
SQL> administer key management set keystore open force keystore identified by
"<staging keystore password>";
keystore altered.
```

8. Verify the status of the staging keystore:

```
SQL> select * from v$encryption_wallet;
WRL_TYPE      WRL_PARAMETER      STATUS
FILE          /u01/tde/CDOMLOSRE17C/      OPEN
```

After this setup, an encrypted source database backup can be restored and recovered on the staging instance before taking a dSource snapshot. The staging database can also be setup as a physical standby database to automatically receive incremental data from the source database.

For provisioning, standard Delphix procedures to provision a [non-multitenant database](#) or a [virtual pluggable database](#) with TDE can be used as is.

# Oracle Staging Push with TDE



Delphix is the industry leader for DevOps test data management.

Businesses need to transform application delivery but struggle to balance speed with data security and compliance. Our DevOps Data Platform automates data security, while rapidly deploying test data to accelerate application releases. With Delphix, customers modernize applications, adopt multicloud, achieve CI/CD, and recover from downtime events such as ransomware up to 2x faster.

Leading companies, including Choice Hotels, Banco Carrefour, and Fannie Mae, use Delphix to accelerate digital transformation and enable zero trust data management. Visit us at [www.delphix.com](https://www.delphix.com).

©2023, Delphix. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document and all scripts contained within and/or described therein are provided as-is without warranty of any kind. Delphix and its licensors disclaim any and all warranties, express or implied, including, but not limited to, any implied warranties and conditions of merchantability, fitness for a particular purpose and noninfringement. Delphix does not warrant that the scripts will meet your requirements or that the use thereof will be bug-free, uninterrupted or error-free. Scripts contained within this document and/or described therein are not regression tested and are not covered by Delphix Support Entitlements. Delphix does not warrant the quality, accuracy, completeness, or reliability of any results obtained from your use of the scripts. Delphix and its licensors specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic, mechanical or otherwise, for any purpose, without our prior written permission.