Delphix Engine 4.2 Release Notes
You can find the most up-to-date technical documentation at:
dacls.delphix.com The Delphix Web site also provides the latest product updates.
If you have comments about this documentation, submit your feedback to: infodev@delphix.com

© 2015 Delphix Corp. All rights reserved.
Delphix and the Delphix logo and design are registered trademarks or trademarks of Delphix Corp. in the United States and/or other jurisdictions.
All other marks and names mentioned herein may be trademarks of their respective companies.

Delphix Corp.
275 Middlefield Road, Suite 50
Menlo Park, CA 94025
1. Introduction ........................................................................................................... 12
  1.1 Database Virtualization with Delphix ............................................................... 13
    1.1.1 The Delphix Approach to Agile Data Management ................................. 14
    1.1.2 Delphix Engine Overview ........................................................................... 17
    1.1.3 Database Linking Overview ........................................................................ 19
    1.1.4 Database Provisioning Overview ................................................................. 21
  2. QuickStart ............................................................................................................ 23
    2.1 Quick Start Guide for the Delphix Engine ...................................................... 24
      2.1.1 Create a Group ........................................................................................... 25
      2.1.2 SQL Server Quick Start Topics ................................................................. 26
        2.1.2.1 Set up a SQL Server Target Environment .............................................. 27
        2.1.2.2 Set up a SQL Server Source Environment ............................................. 29
        2.1.2.3 Link a SQL Server Data Source ............................................................. 31
        2.1.2.4 Provision a SQL Server VDB ................................................................. 34
      2.1.3 Oracle Quick Start Topics ......................................................................... 37
        2.1.3.1 Set up an Oracle Single Instance or RAC Environment ..................... 38
        2.1.3.2 Link an Oracle Data Source ................................................................. 40
        2.1.3.3 Provision an Oracle VDB ...................................................................... 42
      2.1.4 PostgreSQL Quick Start Topics .................................................................. 45
        2.1.4.1 Add a PostgreSQL Environment ......................................................... 46
        2.1.4.2 Link a PostgreSQL Data Source ......................................................... 48
        2.1.4.3 Provision a PostgreSQL VDB ............................................................... 50
      2.1.5 SAP ASE Quick Start Topics ...................................................................... 52
        2.1.5.1 Add an SAP ASE Environment ............................................................ 53
        2.1.5.2 Link a SAP ASE Data Source ............................................................... 55
        2.1.5.3 Provision an SAP ASE VDB ................................................................. 57
      2.1.6 Delete a VDB ............................................................................................... 59
      2.1.7 Delete a dSource ......................................................................................... 60
      2.1.8 Disable a dSource ....................................................................................... 61
  3. SysAdmin ............................................................................................................. 62
    3.1 System Installation, Configuration, and Management ..................................... 63
      3.1.1 Installation and Initial Configuration Requirements .................................. 64
        3.1.1.1 Supported Web Browsers and Operating Systems ............................... 65
        3.1.1.2 Virtual Machine Requirements for VMware ESX ................................ 66
        3.1.1.3 Virtual Machine Requirements for AWS/EC2 Platform ...................... 73
        3.1.1.4 General Network and Connectivity Requirements ............................. 77
      3.1.2 Installation and Initial System Configuration ............................................. 85
        3.1.2.1 The delphix_admin and sysadmin User Roles ..................................... 86
        3.1.2.2 Using HostChecker to Confirm Source and Target Environment Configuration .................................................. 87
        3.1.2.3 Installing the Delphix Engine ................................................................. 88
        3.1.2.4 Setting Up Network Access to the Delphix Engine .............................. 90
        3.1.2.5 Customizing the Delphix Engine System Settings ................................. 93
        3.1.2.6 Configuring SNMP ............................................................................... 95
        3.1.2.7 Setting Up the Delphix Engine .............................................................. 96
        3.1.2.8 Retrieving the Delphix Engine Registration Code ............................... 100
      3.1.3 Upgrading the Delphix Engine ................................................................... 101
        3.1.3.1 Upgrading or Patching a New Version of the Delphix Engine .......... 102
        3.1.3.2 Upgrading to a New Version of the Delphix Engine ......................... 105
        3.1.3.3 Upgrading VM Tools and Hardware .................................................... 108
      3.1.4 Shutting Down the Delphix Engine and Using Factory Reset .................... 109
    3.2 Managing System Administrators ..................................................................... 110
      3.2.1 System Administrators and Delphix Users .............................................. 111
      3.2.2 Adding New System Administrators ......................................................... 112
      3.2.3 Deleting and Suspending System Administrators ...................................... 113
      3.2.4 Reinstating System Administrators ............................................................ 114
    3.3 Capacity and Resource Management ............................................................... 116
      3.3.1 An Overview of Capacity and Performance Information ......................... 117
      3.3.2 Adding and Expanding Storage Devices ................................................... 120
      3.3.3 Working with the Capacity Management Graphs in the Graphical User Interface ................................................................................................................. 121
      3.3.4 Setting Quotas ........................................................................................... 122
      3.3.5 Deleting Objects to Increase Capacity ....................................................... 123
      3.3.6 Changing Snapshot Retention to Increase Capacity .................................. 124
    3.4 Monitoring ......................................................................................................... 125
      3.4.1 Viewing Action Status ............................................................................... 125
      3.4.2 Monitoring Faults ...................................................................................... 126
      3.4.3 Viewing System Events ............................................................................. 132
      3.4.4 Accessing Audit Logs ............................................................................... 133
      3.4.5 Creating Support Logs ............................................................................... 134
3.4.6 Setting Support Access Control .......................................................... 135
3.4.7 Setting SysLog Preferences ................................................................. 136
3.4.8 Diagnosing Connectivity Errors .......................................................... 137

3.5 Performance Tuning ............................................................................. 139
3.5.1 Configuration Options for Improved Performance ..................... 140
3.5.1.1 Network Performance Configuration Options .......................... 141
3.5.1.1.1 Optimal Network Configuration Parameters for the Delphix Engine 142
3.5.1.1.2 Network Operations Using the Delphix Session Protocol ........ 145
3.5.1.1.3 Using the Network Performance Tool ................................. 149
3.5.1.1.4 Interpreting Results from the Network Performance Tool ....... 152
3.5.1.2 Storage Performance Configuration Options ............................. 153
3.5.1.2.1 Optimal Storage Configuration Parameters for the Delphix Engine 154
3.5.1.2.2 Storage Performance Tool ............................................. 156
3.5.1.3 Host Performance Configuration Options ................................. 164
3.5.1.3.1 Target Host Configuration Options for Improved Performance 165

3.5.2 Performance Analytics ................................................................. 172
3.5.2.1 Performance Analytics Tool Overview ....................................... 173
3.5.2.2 Working with Performance Analytics Graphs in the Graphical User Interface 176
3.5.2.3 Performance Analytics Statistics Reference ............................. 180
3.5.2.4 Performance Analytics Tool API Reference ............................. 181
3.5.2.5 Performance Analytics Case Study: Using a Single Statistic ...... 184
3.5.2.6 Performance Analytics Case Study: Using Multiple Statistics .... 188

3.6 Security ............................................................................................... 194
3.6.1 Configuring a Boot Password ......................................................... 195
3.6.2 Configuring a Security Banner ....................................................... 196

4. DelphixAdmin ....................................................................................... 197

4.1 Users, Permissions, and Policies ....................................................... 198
4.1.1 Users and Groups .......................................................................... 199
4.1.1.1 Users, Groups, and Permissions: An Overview ....................... 200
4.1.1.2 User Privileges for Delphix Objects ....................................... 201
4.1.1.3 Adding Delphix Users and Privileges ..................................... 202
4.1.1.4 Editing, Deleting, and Suspending Delphix Users .................... 203
4.1.1.5 Assigning Group and Object Ownership ................................. 204
4.1.1.6 Editing and Deleting Groups .................................................. 205
4.1.1.7 Adding Delphix Admin Users ............................................... 206
4.1.1.8 Managing Individual Profile Information ............................... 207

4.1.2 Managing Policies ...................................................................... 208
4.1.2.1 Managing Policies: An Overview ......................................... 209
4.1.2.2 Creating Custom Policies ..................................................... 211
4.1.2.3 Creating Policy Templates ................................................... 212
4.1.2.4 Policies and Time Zones ...................................................... 213
4.1.2.5 Configuring Retention on Individual Snapshots ...................... 215

4.2 Oracle Environments and Data Sources ...................................... 216
4.2.1 Oracle Support and Requirements .............................................. 217
4.2.1.1 Supported Operating Systems and DBMS Versions for Oracle Environments 218
4.2.1.2 Requirements for Oracle Source Hosts and Databases ........... 220
4.2.1.3 Requirements for Oracle Target Hosts and Databases .......... 224
4.2.1.4 Network and Connectivity Requirements for Oracle Environments 227
4.2.1.5 Sudo Privilege Requirements .............................................. 231
4.2.1.6 Sudo File Configurations ..................................................... 233

4.2.2 Managing Oracle Environments ............................................... 237
4.2.2.1 Using HostChecker to Validate Oracle Source and Target Environments 238
4.2.2.2 Adding an Oracle Single Instance or RAC Environment ........ 243
4.2.2.3 Adding a Database Installation Home to an Oracle Environment 245
4.2.2.4 Adding a Database Group ..................................................... 246
4.2.2.5 Discovering Oracle Pluggable Databases in an Oracle Environment 247
4.2.2.6 Adding a Listener to an Oracle Environment ............................ 248
4.2.2.7 Changing the Host Name or IP Address for Oracle Source and Target Environments 249
4.2.2.8 Editing Oracle Environment Attributes ................................. 251
4.2.2.9 Managing Oracle Environment Users .................................. 253
4.2.2.10 Enabling Linking and Provisioning for Oracle Databases ....... 254
4.2.2.11 Deleting an Oracle Environment ......................................... 255
4.2.2.12 Refreshing an Oracle Environment ...................................... 256

4.2.3 Managing Oracle, Oracle RAC, and Oracle PDB Data Sources .............. 257
4.2.3.1 Linking an Oracle Data Source ............................................. 258
4.2.3.2 Linking an Oracle Pluggable Database .................................. 260
4.2.3.3 Advanced Data Management Settings for Oracle DSources .... 262
4.2.3.4 Using Pre- and Post-Scripts with Oracle dSources .................. 264
4.2.3.5 Enabling Validated Sync for Oracle ....................................... 266
4.2.3.6 Linking dSources from an Encrypted Oracle Database .......... 268
4.2.3.7 Linking Oracle Physical Standby Databases ......................... 269
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.4.4 Using Pre- and Post-Scripts with SQL Server VDBs</td>
<td>412</td>
</tr>
<tr>
<td>4.3.4.5 Extended Properties for SQL Server VDBs</td>
<td>414</td>
</tr>
<tr>
<td>4.3.4.6 Upgrading SQL Server VDBs</td>
<td>415</td>
</tr>
<tr>
<td>4.3.4.7 Migrating a SQL Server VDB</td>
<td>416</td>
</tr>
<tr>
<td>4.3.4.8 Renaming a SQL Server VDB</td>
<td>418</td>
</tr>
<tr>
<td>4.3.4.9 Rewinding a SQL Server VDB</td>
<td>419</td>
</tr>
<tr>
<td>4.3.5 Customizing SQL Server Management</td>
<td>420</td>
</tr>
<tr>
<td>4.3.5.1 Customizing SQL Server dSource Management with Pre- and Post-</td>
<td>421</td>
</tr>
<tr>
<td>Scripts</td>
<td></td>
</tr>
<tr>
<td>4.3.5.2 Customizing SQL Server VDB Management with Hook Operations</td>
<td>423</td>
</tr>
<tr>
<td>4.3.5.2.1 SQL Server VDB Hook Operation Types</td>
<td>427</td>
</tr>
<tr>
<td>4.4 PostgreSQL Environments and Data Sources</td>
<td>429</td>
</tr>
<tr>
<td>4.4.1 PostgreSQL Support and Requirements</td>
<td>430</td>
</tr>
<tr>
<td>4.4.1.1 Requirements for PostgreSQL Source Hosts and Databases</td>
<td>431</td>
</tr>
<tr>
<td>4.4.1.2 Requirements for PostgreSQL Target Hosts and Databases</td>
<td>433</td>
</tr>
<tr>
<td>4.4.1.3 Supporting Operating Systems and Database Versions for PostgreSQL Environments</td>
<td>435</td>
</tr>
<tr>
<td>4.4.1.4 Network and Connectivity Requirements for PostgreSQL Environments</td>
<td>436</td>
</tr>
<tr>
<td>4.4.2 Managing PostgreSQL Environments</td>
<td>440</td>
</tr>
<tr>
<td>4.4.2.1 Setting Up PostgreSQL Environments: An Overview</td>
<td>441</td>
</tr>
<tr>
<td>4.4.2.2 Using HostChecker to Validate PostgreSQL Source and Target Environments</td>
<td>442</td>
</tr>
<tr>
<td>4.4.2.3 Adding a PostgreSQL Environment</td>
<td>445</td>
</tr>
<tr>
<td>4.4.2.4 Adding an Installation to a PostgreSQL Environment</td>
<td>447</td>
</tr>
<tr>
<td>4.4.2.5 Adding a Database Cluster to a PostgreSQL Environment</td>
<td>448</td>
</tr>
<tr>
<td>4.4.2.6 Editing PostgreSQL Environment Attributes</td>
<td>449</td>
</tr>
<tr>
<td>4.4.2.7 Managing PostgreSQL Environment Users</td>
<td>450</td>
</tr>
<tr>
<td>4.4.2.8 Deleting a PostgreSQL Environment</td>
<td>451</td>
</tr>
<tr>
<td>4.4.2.9 Refreshing a PostgreSQL Environment</td>
<td>452</td>
</tr>
<tr>
<td>4.4.2.10 Enabling Staging, Linking and Provisioning for PostgreSQL Environments</td>
<td>453</td>
</tr>
<tr>
<td>4.4.2.11 Changing the Host Name or IP Address for PostgreSQL Source and Target Environments</td>
<td>454</td>
</tr>
<tr>
<td>4.4.3 Managing PostgreSQL Data Sources</td>
<td>455</td>
</tr>
<tr>
<td>4.4.3.1 Linking PostgreSQL Data Sources: Overview</td>
<td>456</td>
</tr>
<tr>
<td>4.4.3.2 Linking a PostgreSQL dSource</td>
<td>457</td>
</tr>
<tr>
<td>4.4.3.3 Advanced Data Management Settings for PostgreSQL Data Sources</td>
<td>459</td>
</tr>
<tr>
<td>4.4.3.4 Using Pre- and Post-Scripts with PostgreSQL dSources</td>
<td>460</td>
</tr>
<tr>
<td>4.4.3.5 Enabling and Disabling PostgreSQL dSources</td>
<td>461</td>
</tr>
<tr>
<td>4.4.3.6 Detaching and Re-Attaching PostgreSQL dSources</td>
<td>462</td>
</tr>
<tr>
<td>4.4.3.7 Deleting a PostgreSQL dSource</td>
<td>464</td>
</tr>
<tr>
<td>4.4.3.8 PostgreSQL dSource Icon Reference</td>
<td>465</td>
</tr>
<tr>
<td>4.4.4 Provisioning VDBs from PostgreSQL dSources</td>
<td>466</td>
</tr>
<tr>
<td>4.4.4.1 Provisioning PostgreSQL VDBs: Overview</td>
<td>467</td>
</tr>
<tr>
<td>4.4.4.2 Provisioning a PostgreSQL VDB</td>
<td>469</td>
</tr>
<tr>
<td>4.4.4.3 Enabling and Disabling PostgreSQL VDBs</td>
<td>471</td>
</tr>
<tr>
<td>4.4.4.4 Refreshing a PostgreSQL VDB</td>
<td>472</td>
</tr>
<tr>
<td>4.4.4.5 Deleting a PostgreSQL VDB</td>
<td>473</td>
</tr>
<tr>
<td>4.4.4.6 Migrating a PostgreSQL VDB</td>
<td>474</td>
</tr>
<tr>
<td>4.4.4.7 Provisioning a SQL Server VDB from a Replicated dSource VDB</td>
<td>475</td>
</tr>
<tr>
<td>4.4.4.8 SQL Server VDB Icon Reference</td>
<td>476</td>
</tr>
<tr>
<td>4.4.4.9 Customizing PostgreSQL VDB Configuration Settings</td>
<td>477</td>
</tr>
<tr>
<td>4.4.5 Customizing PostgreSQL Management with Hook Operations</td>
<td>478</td>
</tr>
<tr>
<td>4.4.5.1 PostgreSQL Hook Operation Types</td>
<td>482</td>
</tr>
<tr>
<td>4.5 SAP ASE Environments and Data Sources</td>
<td>485</td>
</tr>
<tr>
<td>4.5.1 SAP ASE Support and Requirements</td>
<td>486</td>
</tr>
<tr>
<td>4.5.1.1 Requirements for SAP ASE Source Hosts and Databases</td>
<td>487</td>
</tr>
<tr>
<td>4.5.1.2 Requirements for SAP ASE Target Hosts and Databases</td>
<td>489</td>
</tr>
<tr>
<td>4.5.1.3 Network and Connectivity Requirements for SAP ASE Environments</td>
<td>490</td>
</tr>
<tr>
<td>4.5.1.4 Supporting Operating Systems and Database Versions for SAP ASE</td>
<td>494</td>
</tr>
<tr>
<td>4.5.2 Managing SAP ASE Environments</td>
<td>495</td>
</tr>
<tr>
<td>4.5.2.1 Managing SAP ASE Environments: An Overview</td>
<td>496</td>
</tr>
<tr>
<td>4.5.2.2 Using HostChecker to Validate SAP ASE Source and Target Environments</td>
<td>498</td>
</tr>
<tr>
<td>4.5.2.3 Adding an SAP ASE Environment</td>
<td>499</td>
</tr>
<tr>
<td>4.5.2.4 Editing SAP ASE Environment Attributes</td>
<td>501</td>
</tr>
<tr>
<td>4.5.2.5 Changing the Host Name or IP Address of an SAP ASE Environment</td>
<td>502</td>
</tr>
<tr>
<td>4.5.2.6 Deleting an SAP ASE Environment</td>
<td>503</td>
</tr>
<tr>
<td>4.5.2.7 Managing SAP ASE Environment Users</td>
<td>504</td>
</tr>
<tr>
<td>4.5.2.8 Refreshing an SAP ASE Environment</td>
<td>505</td>
</tr>
<tr>
<td>4.5.2.9 Enabling Linking and Provisioning for SAP ASE Environments</td>
<td>506</td>
</tr>
<tr>
<td>4.5.3 Managing SAP ASE Data Sources</td>
<td>507</td>
</tr>
<tr>
<td>4.5.3.1 Linking SAP ASE Data Sources: An Overview</td>
<td>508</td>
</tr>
<tr>
<td>4.5.3.2 Linking a SAP ASE Data Source</td>
<td>509</td>
</tr>
<tr>
<td>4.5.3.3 Advanced Data Management Settings for SAP ASE dSources</td>
<td>511</td>
</tr>
<tr>
<td>4.5.3.4 Using Pre- and Post-Scripts with SAP ASE Data Sources</td>
<td>512</td>
</tr>
<tr>
<td>4.5.3.5 Deleting an SAP ASE dSource</td>
<td>514</td>
</tr>
</tbody>
</table>
### 8.2.1 About the Delphix Agile Data Masking Solution and This Guide

8.2.2 About the Major Delphix Agile Data Masking Components

#### 8.2.2.1 About Certifying Data

- 8.2.2.1.1 A Practical Certification Example
- 8.2.2.1.2 Certification and Delta Masking

#### 8.2.2.2 About Masking Data

- 8.2.2.2.1 About Masking In-Place
- 8.2.2.2.2 About On-The-Fly Masking
- 8.2.2.2.3 Masking a Primary Key Column

#### 8.2.2.3 About Profiling Data

#### 8.2.2.4 About Provisioning (Subsetting Data)

- 8.2.2.4.1 A Practical Subset Example

#### 8.2.3 Getting Started with Agile Data Masking

- 8.2.3.1 Getting Started with Delphix Agile Data Masking
- 8.2.3.2 Create or Import an Environment
- 8.2.3.3 Create and Define Connections
- 8.2.3.4 Create and Define a Rule Set
- 8.2.3.5 Profile Data and Define Inventory
- 8.2.3.6 Mask Data
- 8.2.3.7 Basic Tasks for a Database File
- 8.2.3.8 Certify Data
- 8.2.3.9 Provision Data

#### 8.2.4 Using Delphix Agile Data Masking

- 8.2.4.1 Admin Tab
- 8.2.4.1.1 About
- 8.2.4.1.2 Users
- 8.2.4.2 Managing Connectors
  - 8.2.4.2.1 Creating or Editing a Connector
  - 8.2.4.2.2 Database Connectors
  - 8.2.4.2.3 Deleting Connectors
  - 8.2.4.2.4 File Connectors
  - 8.2.4.2.5 Mainframe Connectors
  - 8.2.4.2.6 The Connector List
- 8.2.4.3 Managing Environments with Agile Data Masking
  - 8.2.4.3.1 Copying and Deleting Environments
  - 8.2.4.3.2 Creating or Importing an Environment
  - 8.2.4.3.3 Exporting an Environment
  - 8.2.4.3.4 The Environment List_Summary Screen
  - 8.2.4.3.5 The Environment Overview Screen
- 8.2.4.4 Managing Jobs
  - 8.2.4.4.1 Creating a New Certify Job
  - 8.2.4.4.2 Creating a New Masking Job
  - 8.2.4.4.3 Creating a New Profiling Job
  - 8.2.4.4.4 Creating a New Provisioning Job
  - 8.2.4.4.5 Creating New Jobs
  - 8.2.4.4.6 Jobs on the Environment Overview Screen
  - 8.2.4.4.7 Running and Stopping Jobs from the Environment Overview Screen
- 8.2.4.5 Managing Masking Inventory
  - 8.2.4.5.1 Importing and Exporting Inventory
  - 8.2.4.5.2 Inventory Settings
  - 8.2.4.5.3 Managing a Database Inventory
  - 8.2.4.5.4 Managing a File Inventory
  - 8.2.4.5.5 The Inventory Screen
- 8.2.4.6 Managing Rule Sets
  - 8.2.4.6.1 Creating and Copying a Rule Set
  - 8.2.4.6.2 Deleting a Rule Set
  - 8.2.4.6.3 Editing a Rule Set
  - 8.2.4.6.4 Modifying Tables in a Rule Set (For Distributed Environment)
  - 8.2.4.6.5 The Rule Set Screen
- 8.2.4.7 Monitor Jobs
- 8.2.4.8 Scheduler Tab
  - 8.2.4.8.1 Creating SQL Statements to Run Before and After Jobs (For Distributed Environment)
  - 8.2.4.8.2 Enabling and Disabling Database Constraints
  - 8.2.4.8.3 Job Completion E-mail Message
  - 8.2.4.8.4 Scheduling Job(s) to Run
- 8.2.4.9 Settings Tab
  - 8.2.4.9.1 Algo...
Introduction
Database Virtualization with Delphix

These topics cover the basic concepts of the Delphix Engine and data virtualization.

- The Delphix Approach to Agile Data Management
- Delphix Engine Overview
- Database Linking Overview
- Database Provisioning Overview
The Delphix Approach to Agile Data Management

This topic describes the three tiers of the Delphix Engine’s agile data architecture, and explains the benefits of the features in each tier.

The Delphix approach to agile data management for the enterprise is to focus on data management in three tiers of activity, and to provide features in each tier that result in an integrated application stack for enterprise data management.

Data Virtualization

Features in the data virtualization tier of the Delphix application stack focus on delivering improved performance for data storage and retrieval through data compression and consolidation.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Mapping</td>
<td>• Patented, flat metadata design scales to unlimited virtual copies</td>
</tr>
<tr>
<td></td>
<td>• 10 - 50x storage reduction and the ability to add parallel environments at no cost</td>
</tr>
<tr>
<td>Compression</td>
<td>• Block aware compression adds 2-4x data reduction</td>
</tr>
<tr>
<td></td>
<td>• 2-4x reduction across virtual copies and backups</td>
</tr>
<tr>
<td>Filtering</td>
<td>• Intelligent filtering eliminates temporary or empty blocks</td>
</tr>
<tr>
<td></td>
<td>• DB awareness drives 10-20% greater efficiency</td>
</tr>
<tr>
<td>Caching</td>
<td>• Drives performance, consolidation: 60%+ I/O load serviced by cache</td>
</tr>
<tr>
<td></td>
<td>• Run 20 Virtual Databases (VDBs) in space of one with good performance</td>
</tr>
</tbody>
</table>

Related Links

- [Database Provisioning Overview](#)
- [Database Linking Overview](#)
- [Capacity and Resource Management](#)
- [Managing Data Sources](#)
- [Provisioning Virtual Databases](#)

Data Orchestration

Features in the data orchestration tier focus on delivering the right data, at the right time, to the right team.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
</table>

### Sync
- Efficiently sync heterogeneous sources in near real time
- Deliver right data to right team at right time

### Record
- Synthesize, record all changes into a continuous TimeFlow
- Database continuity, superior Recover Point Objective (RPO)

### Play
- Fast database provisioning, refresh, rollback, data integration
- Reduce time from 10 days to 10 minutes, from 4 teams to 1 team

### Move
- Promote, demote, consolidate, and recover databases
- Quickly move data through application, development lifecycle stages

### Replicate
- Efficient replication to secondary Delphix virtual appliance
- High availability, disaster recovery, backup

### Related Links
- [Database Linking Overview](#)
- [Managing Data Sources: An Overview](#)
- [Linking an Oracle Data Source](#)
- [Database Provisioning Overview](#)
- [Provisioning Virtual Databases](#)
- [Provisioning an Oracle VDB](#)
- [Refreshing a VDB](#)
- [Replication](#)
- [V2P: Virtual to Physical](#)

### Self-Service Enterprise Data Management
Features in the self-service enterprise data management tier focus on combining easy access to data with fine-grained access control, with the aim of improving team productivity and automating many data management tasks.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
</table>
| Policy Engine| • Granular, role-based control over user and group rights management  
|              | • Easy alignment with enterprise data management policies |
| Automation Engine | • Generation of VDBs from configuration templates and scheduled refreshes  
|                  | • Time and labor savings along with independent data access  
| Self-Service Interfaces | • Web-based Graphical User Interface, Command Line Interface, and Web Services API  
|                     | • Integration with branded portals and other enterprise applications such as ticketing systems  
| Auditability and Security | • Comprehensive event logging and reporting with preservation of source security  
|                         | • Ensure adherence to internal policies and regulations  

**Related Links**

- [User Privileges for Delphix Objects](#)
- [Managing Policies: An Overview](#)
- [Customizing Oracle VDB Configuration Settings](#)
- [Command Line Interface Guide](#)
- [Web Service API Guide](#)
- [System Monitoring](#)
Delphix Engine Overview

The Delphix Engine is a virtual database appliance that provides superior performance over physical databases for application development, disaster recovery, and data warehousing. This topic describes the basic functions of the Delphix Engine and its underlying technology.

What is the Delphix Engine?

- The Delphix Engine virtualizes database infrastructure to provide complete, fully functional databases that operate in a fraction of the space, with improved agility, manageability, and performance.
- The Delphix Engine is a self-contained operating environment and application that is provided as a Virtual Appliance.

What Does the Delphix Engine Do?

The Delphix Engine links to source physical databases via standard APIs and asks the source databases to send copies of their entire file and log blocks to it. The Delphix Engine uses intelligent filtering and compression to reduce the copy of the source database down to as little as 25% of the original size. The copy of the source database stored in the Delphix Engine, along with all incremental updates, is referred to as the dSource in Delphix terminology.

After the initial loading, the Delphix Engine maintains synchronization with source databases based on policy - for example, once daily, or within seconds of the last transaction. Once linked, Delphix maintains a Timeflow of the source database - a rolling record of file and log changes retained by a policy (for example, "keep for two weeks.") From any time within that retention window, a virtual database (referred to in Delphix terminology as a VDB) can be instantly provisioned from the Delphix Engine. VDBs are served from the shared storage footprint of the dSource database Timeflow, so no additional storage is required.

Multiple VDBs can be provisioned from any point in time in a Timeflow, down to the second. Once provisioned, a VDB is an independent, read-write database, and changes made to the VDB by users or applications are written to new, compressed blocks in Delphix storage. VDBs can be provisioned from other VDBs, and the data within VDBs
refreshed from its parent VDB or dSource.
Database Linking Overview

This topic provides an overview of linking to a database within Delphix.

Linking to a Database

The Delphix Engine provides the ability to "link" to an external database by creating a dSource within the Delphix system. Linking to a database is:

- **Non-disruptive** - A linked database continues to operate as a standalone database in the absence of Delphix. No changes to the production workflow are required.
- **Minimally invasive** - Delphix uses standard protocols and APIs to pull changes from the source database, and can be configured to run according to policies that minimize impact to the source system.
- **Continuous** - Synchronization will resume from the previous point, maintaining a continuous history of changes even if the operation is interrupted or servers are temporarily unavailable.
- **Efficient** - Only the changed data is sent once the initial load is complete, and all data is compressed to fit in the fraction of the space.

Once linked, Delphix will maintain a complete history of the database as part of a **Timeflow**, limited by the retention policies configured by the administrator. This timeflow is maintained through the use of **SnapSync** and **LogSync**.

Delphix will automatically discover databases within an environment (host or Oracle cluster), though source databases can also be added manually if they are configured in a non-standard manner that prevents discovery.

**SnapSync**

The SnapSync operation will pull over the complete data set during initial load using standard database protocols. For more information how this works for specific database types, see the following topics:

- Managing Oracle, Oracle RAC, and Oracle PDB Data Sources
- Managing SQL Server Data Sources
- Managing PostgreSQL Data Sources

Subsequent SnapSync operations will pull only the incremental changes and store them in an efficient fashion. At the end of each SnapSync operation, a snapshot is created that serves as the base point for provisioning operations.

When provisioning, the closer the origin point is to a snapshot created via SnapSync, the faster the provisioning operation will occur. The time to provision from a snapshot is directly proportional to the time it took to run the SnapSync operation. It is recommended that users run an incremental SnapSync after the initial load is complete for this reason, as provisioning from the initial snapshot can take a significant amount of time depending on the size of the source database and the rate of change.

SnapSync can be run manually or periodically as part of a policy. See Managing Policies: An Overview for more information.

**LogSync**

In addition to SnapSync, LogSync will periodically connect to the host(s) running the source database via standard protocols and pull over any log files associated with the database. These log files are stored separately from the SnapSync data, and are used to provision from points in between SnapSync snapshots.

When provisioning from a point between snapshots, the additional time it takes to provision is directly proportional to
the time difference between the provision point and the last snapshot. The rate of change on the source database dictates the amount of data that must be replayed to bring a virtual database to the correct point in time.

LogSync data is maintained separately from snapshots according to the configured retention policy. LogSync can be disabled on a dSource, but logs will still need to be fetched while running SnapSync to account for the time taken running the SnapSync operation.
Database Provisioning Overview

This topic describes how provisioning virtual databases (VDBs) works, and the benefits they provide.

Virtual Databases

A virtual database is created within Delphix from the timeflow of another database (dSource or VDB). Virtual databases are provisioned to a particular installation contained within a host or cluster, and Delphix handles the configuration and management of the externally visible database. Virtual databases have the following attributes:

- **Lightweight** - Virtual databases take up very little storage space. The amount of space consumed is directly proportional to the amount of data changed, either through executing DML queries against the database, or through replaying the logs required to reach the requested provision point.

- **Flexible** - VDBs support a variety of operations that are cumbersome with physical databases, such as taking snapshots, provisioning copies of VDBs, refreshing the data within a VDB, or migration to a physical repository

- **Simple** - Management of VDBs is controlled via Delphix. While the database can be managed through the use of native database tools, Delphix will handle provisioning, teardown, start and stop, migration to a different host, and more, through the Delphix administrative interface.

Provisioning

When a VDB is provisioned, the user selects a point along the timeflow of the parent database and a target environment in which to create the database instance. Delphix will:

- Create a writeable copy of the physical storage in a time and space efficient manner
- Export the physical storage to the target server over the appropriate protocol (e.g. NFS)
- Configure a database instance in the target environment according to the settings provided by the administrator
- Use the instance to take the database through recovery and replay (if necessary)
- Take a consistent snapshot of the resulting storage
- Make the database available to end users

The time it takes to provision is dependent on the point chosen in the parent timeflow and the type of database. As a general rule, provisioning from a snapshot will be faster than provisioning from a point between snapshots. The time to provision from a snapshot is roughly proportional to the time it took to run the SnapSync operation (if a dSource), or constant time if the snapshot is from a VDB. When provisioning between snapshots, the time to provision is roughly proportional to the amount of change incurred within the source database between the last snapshot and the provision point.

See the topics under [Provisioning Virtual Databases](#) for more information.

Refresh

The contents of a VDB can be refreshed without affecting the external identity of the database instance. During refresh, Delphix will:

- Shut down the external database instance
- Create a writeable copy of the physical storage in a time and space efficient manner
- Export the physical storage to the target server over the appropriate protocol (e.g. NFS)
• Use the instance to take the database through recovery and replay (if necessary)
• Make the database available to end users

During this process, the logical configuration of the database instance does not change, so end users see only a brief downtime, followed by the contents of the database being updated to the latest version.

See the topics Managing Policies: An Overview and Refreshing a VDB for more information.

Virtual to Physical

While VDBs offer a great deal of flexibility in terms of management, there are times when creating a physical copy, where the storage is not within Delphix and the database instance is not managed by Delphix, is required. Examples might be physical copies for rigorous performance testing, or for restoring a physical copy in the event of a disaster or logical data corruption. During the V2P operation, Delphix will:

• Configure a database instance in the target environment according to the settings provided by the administrator
• Copy the physical data from Delphix to the target environment
• Use the instance to take the database through recovery and replay (if necessary)
• Make the database available to end users

Unlike provisioning, the resulting database instance is not managed by Delphix, and Delphix will treat it as any other external database once the V2P operation is complete. Because the full data set must be copied to the remote server, the V2P time is roughly proportional to the logical size of the VDB.

See the topics under V2P: Virtual to Physical for more information.
QuickStart
Quick Start Guide for the Delphix Engine

These topics describe the basic processes for setting up environments, setting up dSources, provisioning VDBs, and then deleting dSources and VDBs.

These topics are excerpted from the larger user guide, and are intended to provide you with a quick overview of basic procedures for working with database objects in the Delphix Engine. This guide does not cover setting up and configuring the Delphix Engine, and the requirements of your installation and database platform may require more detailed instructions. We highly recommend that you read the topics linked from the topics in this guide, as well as the conceptual overview topics included at the beginning of each chapter of the user guide, before undertaking complex operations with the Delphix Engine.

- Create a Group
- SQL Server Quick Start Topics
  - Set Up a SQL Server Target Environment
  - Set Up a SQL Server Source Environment
  - Link a SQL Server Data Source
  - Provision a SQL Server VDB
- Oracle Quick Start Topics
  - Set Up an Oracle Single Instance or RAC Environment
  - Link an Oracle Data Source
  - Provision an Oracle VDB
- PostgreSQL Quick Start Topics
  - Add a PostgreSQL Environment
  - Link a PostgreSQL Data Source
  - Provision a PostgreSQL VDB
- SAP ASE Quick Start Topics
  - Add an SAP ASE Environment
  - Link a SAP ASE Data Source
  - Provision an SAP ASE VDB
- Delete a VDB
- Delete a dSource
- Disable a dSource
Create a Group

Before you can link to a dSource or provision a VDB, you will need to create a group that will contain your database objects. Permissions and policies for database objects are also determined within the group, as described in Users, Groups, and Permissions: An Overview.

When you first start up the Delphix Engine, a default group, <New Group>, is already defined. You can edit the name of this group, as well as the policies and permissions associated with it, to use as your first group, or you can create a group as described in the following steps.

Groups for dSources and VDBs

Since policies and permissions for database objects are set by the group they belong too, you may want to create two groups, one for dSources, one for VDBs, so you can set policies and permissions by object types.

Adding a Group

1. Log into the Delphix Admin application as a user with Delphix Admin privileges.
2. In the Databases menu, select Add New Group.
3. Enter a Group Name and an optional description.
4. Click OK.

Deleting a Group

1. Log into the Delphix Admin application as a user with Delphix Admin privileges or group OWNER privileges for the target group.
2. Open the group card in the Databases panel by selecting the target group
3. Click the Trash Can icon.
4. Click OK.

Deleting Groups Containing Objects

A group cannot be deleted if it contains VDBs or dSources. All databases within a group must be deleted prior to deleting the group.

At Least One Group Must Exist

At least one group must always exist on the Delphix Engine in order to link a dSource. If you delete the last group, you will need to create a new group in order to create a dSource.
SQL Server Quick Start Topics

These topics, which are excerpted from the larger User Guide, are intended to provide you with quick overview of how to work with SQL Server database objects in the Delphix Engine. Before undertaking any of these procedures we strongly recommend that you read the topics in the SQL Server Support and Requirements section.

- Set Up a SQL Server Target Environment
- Set Up a SQL Server Source Environment
- Link a SQL Server Data Source
- Provision a SQL Server VDB
Set Up a SQL Server Target Environment

As explained in **Setting Up SQL Server Environments: An Overview** SQL Server targets can be used for three purposes in a Delphix Engine deployment:

- They can host a target environment for the provisioning of Virtual Databases (VDBs)
- They can host a staging database for a linked dSource and run the validated sync process
- They can serve as a proxy host for database discovery on source hosts

Regardless of the specific purpose, all Windows targets must have the Delphix Connector installed to enable communication between the host and the Delphix Engine. The instructions in this topic cover initiating the Add Target process in the Delphix Engine interface, running the Delphix Connector installer on the target machine, and then verifying that the target has been added in the Delphix Engine interface.

**Prerequisites**

- Make sure your target environment meets the requirements described in Requirements for SQL Server Target Hosts and Databases.
- On the Windows machine that you want to use as a target, you will need to download the Delphix Connector software through the Delphix Engine interface, install it and then register that machine with the Delphix Engine.

**Procedure**

1. From the machine that you want to use as a target, log into the Delphix Admin application.
2. Select **Manage > Environments**.
3. Next to **Environments**, click the green **Plus** icon.
4. In the **Add Environment** dialog, select **Windows** in the operating system menu.
5. Select **Target**.
6. Click the download link for the **Delphix Connector Installer**. The Delphix Connector will download to your local machine.
7. On the Windows machine that you want to use as a target, run the Delphix Connector installer. Click **Next** to advance through each of the installation wizard screens.

   - **The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.**

   a. For **Connector Configuration**, make sure there is no firewall in your environment blocking traffic to the port on the target environment that the Delphix Connector service will listen to.
   b. For **Select Installation Folder**, either accept the default folder, or click **Browse** to select another.
   c. Click **Close** to complete the installation process.
   d. Run this batch script as Administrator: `<Delphix Connector installation folder>\Delphix\DelphixConnector\connector\addhostgui.cmd`. When the Add Windows Target Environment Wizard launches, provide the **Target Host IP Address**, **Delphix Engine IP Address**, your login credentials, and the environment user on the Windows target host.
   e. After providing this information, click **Submit**, and then click **Yes** to confirm the target environment addition request.

8. In the Delphix Engine interface, you will see a new icon for the Target environment, and two jobs running in the Delphix Admin Job History, one to Create and Discover an environment, and another to Create an environment. When the jobs are complete, click on the icon for the new environment, and you will see the
details for the environment.

Post-Requisites

- On the target machine, in the Windows Start Menu, go to Services > Extended Services, and make sure that the Delphix Connector service has a Status of Started, and that the Startup Type is Automatic.

Video

Related Links

- Setting Up SQL Server Environments: An Overview
- Requirements for SQL Server Target Hosts and Databases
Set Up a SQL Server Source Environment

As explained in Setting Up SQL Server Environments: An Overview SQL Server targets can be used for three purposes in a Delphix Engine deployment:

- They can host a target environment for the provisioning of Virtual Databases (VDBs)
- They can host a staging database for a linked dSource and run the validated sync process
- They can serve as a proxy host for database discovery on source hosts

Regardless of the specific purpose, all Windows targets must have the Delphix Connector installed to enable communication between the host and the Delphix Engine. The instructions in this topic cover initiating the Add Target process in the Delphix Engine interface, running the Delphix Connector installer on the target machine, and then verifying that the target has been added in the Delphix Engine interface.

Prerequisites

- Make sure your target environment meets the requirements described in Requirements for SQL Server Target Hosts and Databases.
- On the Windows machine that you want to use as a target, you will need to download the Delphix Connector software through the Delphix Engine interface, install it and then register that machine with the Delphix Engine.

Procedure

1. From the machine that you want to use as a target, log into the Delphix Admin application.
2. Select Manage > Environments.
3. Next to Environments, click the green Plus icon.
4. In the Add Environment dialog, select Windows in the operating system menu.
5. Select Target.
6. Click the download link for the Delphix Connector Installer. The Delphix Connector will download to your local machine.
7. On the Windows machine that you want to want to use as a target, run the Delphix Connector installer. Click Next to advance through each of the installation wizard screens.

⚠️ The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.

a. For Connector Configuration, make sure there is no firewall in your environment blocking traffic to the port on the target environment that the Delphix Connector service will listen to.
b. For Select Installation Folder, either accept the default folder, or click Browse to select another.
c. Click Close to complete the installation process.
d. Run this batch script as Administrator: `<Delphix Connector installation folder>\Delphix\DelphixConnector\connector\addhostgui.cmd`. When the Add Windows Target Environment Wizard launches, provide the Target Host IP Address, Delphix Engine IP Address, your login credentials, and the environment user on the Windows target host.
e. After providing this information, click Submit, and then click Yes to confirm the target environment addition request.

8. In the Delphix Engine interface, you will see a new icon for the Target environment, and two jobs running in the Delphix Admin Job History, one to Create and Discover an environment, and another to Create an environment. When the jobs are complete, click on the icon for the new environment, and you will see the
details for the environment.

Post-Requisites

- On the target machine, in the Windows Start Menu, go to Services > Extended Services, and make sure that the Delphix Connector service has a Status of Started, and that the Startup Type is Automatic.

Video

Related Links

- Setting Up SQL Server Environments: An Overview
- Requirements for SQL Server Target Hosts and Databases
Link a SQL Server Data Source

- **Prerequisites**
- **Procedure**
- **Related Links**

**Prerequisites**

- Be sure that the source database meets the requirements described in [Requirements for SQL Server Target Hosts and Databases](#).
- You should already have set up a staging target environment as described in [Setting Up SQL Server Environments: An Overview](#) and [Adding a Windows Target Environment](#).

**Maximum Size of a Database that Can Be Linked**

- If the staging environment uses the Windows 2003 operating system, the largest size of database that you can link to the Delphix Engine is 2TB. This is also the largest size to which a virtual database (VDB) can grow.
- For all other Windows versions, the maximum size for databases and VDBs is 32TB.

In both cases, the maximum size of the database and resulting VDBs is determined by the operating system on the staging target host.

**Failover cluster environments cannot be used for staging**

When linking a dSource, you cannot use SQL Server failover cluster instances as staging instances. When linking, select a standalone SQL Server instance to use.

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials or as the owner of the database from which you want to provision the dSource.
2. Select Manage.
3. Select Databases.
4. Select Add dSource. Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
5. In the Add dSource wizard, select the source database.
6. Enter your login credentials for the source database.
7. Click Verify Credentials.
8. Click Next.
9. Select a Database Group for the dSource.
10. Click Next. Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and

**Changing the Environment User**

If you need to change or add an environment user for the source database, see [Managing SQL Server Environment Users](#).
its objects, such as snapshots. For more information, see the topics under Users, Permissions, and Policies.

⚠️ If your data source name contains non-ASCII characters, you will need to change the default dSource name to something that uses only ASCII characters.

11. Select the method for the Initial Load. For details on initial load options, see Linking a dSource from a SQL Server Database: An Overview.

12. Enter a backup path from which the source database backups will be available for the Delphix Engine to restore. Alternatively, select Autodiscover to have the Delphix Engine automatically locate the backups by querying MSDB.

13. Select the target environment for creating the staging database for validated sync.

14. Select a standalone SQL Server instance on the target environment for hosting the staging database.

15. Select whether the data in the database is Masked.

16. Select whether you want LogSync enabled for the dSource. For more information, see Advanced Data Management Settings for SQL Server dSources.

⚠️ LogSync Disabled by Default

LogSync is disabled by default for SQL Server data sources. For more information about how LogSync functions with SQL Server data sources, see Managing SQL Server Data Sources.

17. Click Advanced to edit retention policies and specify pre- and post-scripts. For details on pre- and post-scripts, refer to Using Pre- and Post-Scripts with SQL Server dSources.

18. Click Next.

19. Review the dSource Configuration and Data Management information.

20. Click Finish.

The Delphix Engine will initiate two jobs to create the dSource, DB_Link and DB_Sync. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have completed successfully, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will appear in the list of My Databases under its assigned group.

You can view the current state of Validated Sync for the dSource on the dSource card itself.

İ The dSource Card

After you have created a dSource, the dSource card allows you to view information about it and make modifications to its policies and permissions. In the Databases panel, click the Open icon to view the front of the dSource card. You can then flip the card to see information such as the Source Database and Data Management configuration. For more information, see the topic Advanced Data Management Settings for SQL Server dSources.

Related Links

- Users, Permissions, and Policies
- Setting Up SQL Server Environments: An Overview
- Linking a dSource from a SQL Server Database: An Overview
- Advanced Data Management Settings for SQL Server dSources
• Adding a SQL Server Standalone Target Environment
• Requirements for SQL Server Target Hosts and Databases
• Using Pre- and Post-Scripts with SQL Server dSources
Provision a SQL Server VDB

Prerequisites

- You will need to have linked a dSource from a source database, as described in Linking a SQL Server dSource, or have already created a VDB from which you want to provision another VDB.
- You should already have set up Windows target environments and installed the Delphix Connector on them, as described in Adding a SQL Server Standalone Target Environment.
- Make sure you have the required privileges on the target environment as described in Requirements for SQL Server Target Hosts and Databases.
- If you are provisioning to a different target environment than the one where the staging database has been set up, you need to make sure that the two environments have compatible operating systems, as described in Requirements for SQL Server Target Hosts and Databases. For more information on the staging database and the validated sync process, see Setting Up SQL Server Environments: An Overview.

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a means of provisioning. See Provisioning by Snapshot and LogSync in this topic for more information.
5. Click Provision. The Provision VDB panel will open, and the Database Name and Recovery Model will auto-populate with information from the dSource.
6. Select a target environment from the left pane.
7. Select an Instance to use.
8. If the selected target environment is a Windows Failover Cluster environment, select a drive letter from Available Drives. This drive will contain volume mount points to Delphix storage.
9. Specify any Pre or Post Scripts that should be used during the provisioning process. See Using Pre- and Post-Scripts with dSources and SQL Server VDBs for more information.
10. Click Next.
11. Select a Target Group for the VDB. Click the green Plus icon to add a new group, if necessary.
12. Select a Snapshot Policy for the VDB. Click the green Plus icon to create a new policy, if necessary.
13. Click Next.
14. If your Delphix Engine system administrator has configured the Delphix Engine to communicate with an SMTP server, you will be able to specify one or more people to notify when the provisioning is done. You can choose other Delphix Engine users, or enter email addresses.
15. Click Finish. When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the VDB will be included in the group you designated, and listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.
You can select a SQL Server instance that has a higher version than the source database and the VDB will be automatically upgraded. See the topic SQL Server Operating System Compatibility Matrices for more information about compatibility between different versions of SQL Server.

Provisioning by Snapshot or LogSync

When provisioning by snapshot, you can provision to the start of any particular snapshot, either by time or LSN.

You can take a new snapshot of the dSource and provision from it by clicking the Camera icon on the dSource card.

<table>
<thead>
<tr>
<th>Provisioning By Snapshot</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by Time</td>
<td>You can provision to the start of any snapshot by selecting that snapshot card from the TimeFlow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.</td>
</tr>
<tr>
<td>Provision by LSN</td>
<td>You can use the Slide to Provision by LSN control to open the LSN entry field. Here, you can type or paste in the LSN you want to provision to. After entering a value, it will &quot;snap&quot; to the start of the closest appropriate snapshot.</td>
</tr>
</tbody>
</table>

If LogSync is enabled on the dSource, you can provision by LogSync information. When provisioning by LogSync information, you can provision to any point in time, or to any LSN, within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the Slide to Open LogSync control at the top of an individual snapshot card.

<table>
<thead>
<tr>
<th>Provisioning By LogSync</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by Time</td>
<td>Use the Slide to Open LogSync control to view the time range within that snapshot. Drag the red triangle to the point in time that you want to provision from. You can also enter a date and time directly.</td>
</tr>
<tr>
<td>Provision by LSN</td>
<td>Use the Slide to Open LogSync and Slide to Provision by LSN controls to view the range of LSNs within that snapshot. You must type or paste in the specific LSN you want to provision to. Note that if the LSN doesn't exist, you will see an error when you provision.</td>
</tr>
</tbody>
</table>

Related Links
- Linking a SQL Server dSource
- Adding a SQL Server Standalone Target Environment
- Adding a SQL Server Failover Cluster Target Environment
• Requirements for SQL Server Target Hosts and Databases
• Setting Up SQL Server Environments: An Overview
• Using Pre- and Post-Scripts with dSources and SQL Server VDBs
Oracle Quick Start Topics

These topics, which are excerpted from the larger User Guide, are intended to provide you with a quick overview of working with Oracle database objects in the Delphix Engine. Before undertaking any of these procedures we strongly recommend that you read the topics in the Oracle Support and Requirements section.

- Set Up an Oracle Single Instance or RAC Environment
- Link an Oracle Data Source
- Provision an Oracle VDB
Set Up an Oracle Single Instance or RAC Environment

Prerequisites
- See the topics Requirements for Oracle Target Hosts and Databases and Supported Operating Systems and DBMS Versions for Oracle Environments

Procedure
1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click the Plus icon next to Environments.
4. In the Add Environment dialog, select Unix/Linux.
5. Select Standalone Host or Oracle Cluster, depending on the type of environment you are adding.
6. For standalone Oracle environments enter the Host IP address.
7. For Oracle RAC environments, enter the Node Address and Cluster Home.
8. Enter an optional Name for the environment.
9. Enter the SSH port.
   The default value is 22.
10. Enter a Username for the environment.
    See Requirements for Oracle Target Hosts and Databases for more information on the required privileges for the environment user.
11. Select a Login Type.
    For Password, enter the password associated with the user in Step 10.

   Using Public Key Authentication
   If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      ii. Run chmod 755 ~ to make your home directory writable only by your user.

   The public key needs to be added only once per user and per environment.
   You can also add public key authentication to an environment user's profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

12. For Password Login, click Verify Credentials to test the username and password.
13. Enter a Toolkit Path.
    The toolkit directory stores scripts used for Delphix Engine operations, and should have a persistent working directory rather than a temporary one. The toolkit directory will have a separate sub-directory for each database instance. The toolkit path must have 0770 permissions and at least 345MB of free space.
14. Click OK.
Post-Requisites

After you create the environment, you can view information about it by selecting Manage > Environments, and then select the environment name.

Related Links

- Requirements for Oracle Target Hosts and Databases
Link an Oracle Data Source

Prerequisites

- Make sure you have the correct user credentials for the source environment, as described in Requirements for Oracle Target Hosts and Databases.
- If you are linking a dSource to an Oracle or Oracle RAC physical standby database, you should read the topic Linking Oracle Physical Standby Databases.
- If you are using Oracle Enterprise Edition, you must have Block Change Tracking (BCT) enabled as described in Requirements for Oracle Source Hosts and Databases.
- The source database should be in ARCHIVELOG mode and the NOLOGGING option should be disabled as described in Requirements for Oracle Source Hosts and Databases.
- You may also want to read the topic Advanced Data Management Settings for Oracle dSources.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > Add dSource. Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
3. In the Add dSource wizard, select the source database.
4. Enter your login credentials for the source database and click Verify Credentials. If you are linking a mounted standby, click Advanced and enter non-SYS login credentials as well. Click Next. See the topics under Linking Oracle Physical Standby Databases for more information about how the Delphix Engine uses non-SYS login credentials.
5. Select a Database Group for the dSource, and then click Next. Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users, Permissions, and Policies for more information.
6. Select an Initial Load option. By default, the initial load takes place upon completion of the linking process. Alternatively, you can set the initial load to take place according to the SnapSync policy, for example if you want the initial load to take place when the source database is not in use, or after a set of operations have taken place.
7. Select whether the data in the database is Masked. This setting is a flag to the Delphix Engine that the database data is in a masked state. Selecting this option will not mask the data.
8. Select a SnapSync policy. See Advanced Data Management Settings for Oracle dSources for more information.
9. Click Advanced to edit LogSync, Validated Sync, and Retention policies. See Advanced Data Management Settings for Oracle dSources for more information.
10. Click Next.
11. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs, DB_Link and DB_Sync, to create the dSource. You can monitor...
these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have successfully completed, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of My Databases under its assigned group.

### The dSource Card

After you have created a dSource, you can view information about it on the dSource card, and also make modifications to its policies and permissions. In the Databases panel, click on the Open icon to view the front of the dSource card. The card will then flip, showing you information such as the Source Database and Data Management configuration. See the topic Advanced Data Management Settings for Oracle dSources for more information.

### Related Links

- [Advanced Data Management Settings for Oracle dSources](#)
- [Requirements for Oracle Target Hosts and Databases](#)
- [Linking dSources from an Encrypted Oracle Database](#)
- [Linking Oracle Physical Standby Databases](#)
- [Users, Permissions, and Policies](#)
Provision an Oracle VDB

Prerequisites

- You will need to have linked a dSource from a source database, as described in Linking an Oracle Data Source, or have already created a VDB from which you want to provision another VDB.
- You will need to have the correct OS User privileges on the target environment, as described in Requirements for Oracle Target Hosts and Databases.
- If you want to use customized database configuration settings, first create a VDB Config Template as described in Customizing Oracle VDB Configuration Settings.
- If you are creating a VDB from a dSource linked to an encrypted database, make sure you have copied the wallet file to the target environment as described in Provisioning a VDB from an Encrypted Oracle Database.

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a dSource snapshot.
   See Provisioning by Snapshot and LogSync in this topic for more information on provisioning options.
   You can take a snapshot of the dSource to provision from by clicking the Camera icon on the dSource card.
5. Optional: Slide the LogSync slider to open the snapshot timeline, and then move the arrow along the timeline to provision from a point of time within a snapshot.
   You can provision from the most recent log entry by opening the snapshot timeline, and then click the red Arrow icon next to the LogSync Slider.
6. Click Provision.
   The Provision VDB panel will open, and the fields Installation Home, Database Unique Name, SID, Database Name, Mount Base, and Environment User will auto-populate with information from the dSource.
   If provisioning Oracle RAC VDBs please note:
   Delphix, by default, provisions 'n' nodes (instances) RAC VDB, where 'n' is the number of nodes in the target cluster environment, this can be overridden by deseleting the individual nodes from the GUI while provisioning, however this will not effect functionality of the VDB just which number of 'n' nodes the VDB will connect to.
   If you need to add a new target environment for the VDB, click the green Plus icon next to the Filter Target field, and follow the instructions in Adding an Oracle Single Instance or RAC Environment.
   Review the information for Installation Home, Database Unique Name, SID, and Database Name and edit as necessary.
   Review the Mount Base and Environment User and edit as necessary.
   The Environment User must have permissions to write to the specified Mount Base, as described in Requirements for Oracle Target Hosts and Databases. You may also want to create a new writeable directory in the target environment with the correct permissions, and use that as the Mount Base for the VDB.
10. Select **Provide Privileged Credentials** if you want to use login credentials on the target environment other than those associated with the **Environment User**.

11. Click **Advanced** to select Oracle Node Listeners or enter any VDB configuration settings or file mappings. See [Customizing Oracle VDB Configuration Settings](#) and [Customizing VDB File Mappings](#) for more information.

   ![](image)

   If you are provisioning to a target environment that is running a Linux OS, you will need to compare the `SGA_TARGET` configuration parameter with the shared memory size in `/dev/shm`. The shared memory configured on the target host should match the SGA memory target. You can check this by opening the **Advanced** settings, and then finding the value for `SGA_TARGET` under **DB Configuration**.

12. Click **Next**.

13. Select a **Target Group** for the VDB.

   Click the green **Plus** icon to add a new group, if necessary.

14. Select a **Snapshot Policy** for the VDB.

   Click the green **Plus** icon to create a new policy, if necessary.

15. Click **Next**.

16. Enter any operations that should be run at **Hooks** during the provisioning process.

   See [Customizing Oracle Management with Hook Operations](#) for more information.

17. Click **Next**.

18. Click **Finish**.

   When provisioning starts, you can review progress of the job in the Databases panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the VDB will be included in the group you designated, and listed in the **Databases** panel. If you select the VDB in the Databases panel and click the **Open** icon, you can view its card, which contains information about the database and its Data Management settings.

### Provisioning by Snapshot or LogSync

When provisioning by snapshot, you can provision to the start of any particular snapshot, either by time or SCN.

<table>
<thead>
<tr>
<th>Provisioning By Snapshot</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by Time</td>
<td>You can provision to the start of any snapshot by selecting that snapshot card from the Timeflow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.</td>
</tr>
<tr>
<td>Provision by SCN</td>
<td>You can use the <strong>Slide to Provision by SCN</strong> control to open the SCN entry field. Here, you can type or paste in the SCN you want to provision to. After entering a value, it will &quot;snap&quot; to the start of the closest appropriate snapshot.</td>
</tr>
</tbody>
</table>

When provisioning by LogSync information, you can provision to any point in time, or to any SCN, within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the **Slide to Open LogSync** control at the top of an individual snapshot card.
### Provisioning By LogSync

<table>
<thead>
<tr>
<th>Provision by</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by SCN</td>
<td>Use the <strong>Slide to Open LogSync</strong> and <strong>Slide to Provision by</strong> SCN controls to view the range of SCNs within that snapshot. Drag the red triangle to the LSN that you want to provision from. You can also type or paste in the specific SCN you want to provision to. Note that if the SCN doesn't exist, you will see an error when you provision.</td>
</tr>
<tr>
<td>Provision by Time</td>
<td>Use the <strong>Slide to Open LogSync</strong> control to view the time range within that snapshot. Drag the red triangle to the point in time that you want to provision from. You can also enter a date and time directly.</td>
</tr>
</tbody>
</table>

### Video

[Get Adobe Flash Player]

### Related Links

- [Linking an Oracle Data Source](#)
- [Requirements for Oracle Target Hosts and Databases](#)
- [Customizing Oracle VDB Configuration Settings](#)
- [Customizing VDB File Mappings](#)
- [Customizing Oracle Management with Hook Operations](#)
PostgreSQL Quick Start Topics

These topics, which are excerpted from the larger User Guide, are intended to provide you with a quick overview of working with PostgreSQL data sources in the Delphix Engine. Before undertaking any of these procedures we strongly recommend that you read the topics in the PostgreSQL Support and Requirements section.

- Add a PostgreSQL Environment
- Link a PostgreSQL Data Source
- Provision a PostgreSQL VDB
Add a PostgreSQL Environment

This topic describes how to add a PostgreSQL source environment to the Delphix Engine.

Prerequisites

Make sure your environment meets the requirements described in the following topics:

- Requirements for PostgreSQL Source Hosts and Databases
- Requirements for PostgreSQL Target Hosts and Databases
- Supported Operating Systems and Database Versions for PostgreSQL Environments

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Environments.
3. Next to Environments, click the green Plus icon.
4. In the Add Environment dialog, select Unix/Linux in the operating system menu.
5. Select Standalone Host.
6. Enter the Host IP address.
7. Enter an optional Name for the environment.
8. Enter the SSH port.
   The default value is 22.
9. Enter a Username for the environment.
   See Requirements for PostgreSQL Target Hosts and Databases and Requirements for PostgreSQL Source Hosts and Databases for more information about the environment user requirements.
10. Select a Login Type.
    For Password, enter the password associated with the user in Step 9.

   ![Using Public Key Authentication]
   If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      ii. Run chmod 755 ~ to make your home directory writable only by your user.
   The public key needs to be added only once per user and per environment.
   You can also add public key authentication to an environment user’s profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

11. For Password Login, click Verify Credentials to test the username and password.
12. Enter a Toolkit Path.
    See Requirements for PostgreSQL Target Hosts and Databases and Requirements for PostgreSQL Source Hosts and Databases for more information about the toolkit directory requirements.
13. Click OK. As the new environment is added, you will see two jobs running in the Delphix Admin Job History, one to Create and Discover an environment, and another to Create an environment. When the jobs are complete, you will see the new environment added to the list in the Environments panel. If you don’t see it, click the Refresh icon in your browser.

Post-Requisites

- After you create the environment, you can view information about it by selecting Manage > Environments, and then select the environment name.

Related Links

- Setting Up PostgreSQL Environments: An Overview
- Requirements for PostgreSQL Source Hosts and Databases
- Requirements for PostgreSQL Target Hosts and Databases
- Supported Operating Systems and Database Versions for PostgreSQL Environments
- Adding an Installation to a PostgreSQL Environment
**Link a PostgreSQL Data Source**

This topic describes the basic procedure for linking a dSource from a PostgreSQL database to the Delphix Engine.

**Prerequisites**
- Make sure you have the correct user credentials for the source environment, as described in Requirements for PostgreSQL Source Hosts and Databases.
- You may also want to read the topic Advanced Data Management Settings for PostgreSQL Data Sources.

**Procedure**

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > Add dSource. Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
3. In the Add dSource wizard, select the source database.

   ![Changing the Environment User]

   If you need to change or add an environment user for the source database, see Managing PostgreSQL Environment Users.

4. Enter your login credentials for DB Cluster User and DB Cluster Password.
5. Click Advanced to enter a Connection Database. The Connection Database will be used when issuing SQL queries from the Delphix Engine to the linked database. It can be any existing database that the DB Cluster User has permission to access.
6. Click Next.
7. Select a Database Group for the dSource, and then click Next. Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users, Permissions, and Policies for more information.
8. Select a SnapSync Policy, and, if necessary, a Staging Installation for the dSource. The Staging installation represents the PostgreSQL binaries that will be used on the staging target to backup and restore the linked database to a warm standby.
9. Click Advanced to select whether the data in the data sources is Masked, to select a Retention Policy, and to indicate whether any pre or post scripts should be executed during the dSource creation. See Advanced Data Management Settings for PostgreSQL Data Sources and Using Pre- and Post-Scripts with PostgreSQL dSources for more information.
10. Click Next.
11. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs, DB_Link and DB_Sync, to create the dSource. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have successfully completed, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of My Databases under its assigned group.
The dSource Card

After you have created a dSource, you can view information about it on the dSource card, and also make modifications to its policies and permissions. In the Databases panel, click on the Open icon to view the front of the dSource card. The card will then flip, showing you information such as the Source Database and Data Management configuration. See the topic Advanced Data Management Settings for PostgreSQL Data Sources for more information.

Related Links

- Advanced Data Management Settings for PostgreSQL Data Sources
- Requirements for PostgreSQL Target Hosts and Databases
- Using Pre- and Post-Scripts with PostgreSQL dSources
- Users, Permissions, and Policies
Provision a PostgreSQL VDB

This topic describes how to provision a virtual database (VDB) from a PostgreSQL dSource.

Prerequisites

- You will need to have linked a dSource from a source database, as described in Linking a PostgreSQL dSource, or have already created a VDB from which you want to provision another VDB.

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a dSource snapshot.
   See Provisioning by Snapshot and LogSync in this topic for more information on provisioning options.

5. Optional: Slide the LogSync slider to the open the snapshot timeline, and then move the arrow along the timeline to provision from a point in time within a snapshot.
6. Click Provision.
   The VDB Provisioning Wizard will open, and the fields Installation, Mount Base, and Environment User will auto-populate with information from the environment configuration.
7. Enter a Port Number.
   The TCP port upon which the VDB will listen.
8. Click Advanced to enter any VDB configuration settings.
   See Customizing PostgreSQL VDB Configuration Settings for more information.
9. Click Next to continue to the VDB Configuration tab.
10. Modify the VDB Name if necessary.
11. Select a Target Group for the VDB.
12. Click the green Plus icon to add a new group, if necessary.
13. Select a Snapshot Policy for the VDB.
14. Click the green Plus icon to create a new policy, if necessary.
15. Click Next to continue to the Hooks tab.
16. Specify any Hooks to be used during the provisioning process.
   See Customizing PostgreSQL Management with Hook Operations for more information.
17. Click Next to continue to the Summary tab.
18. Click Finish.
   When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the VDB will be included in the group you designated, and listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.

Provisioning by Snapshot or LogSync
When provisioning by snapshot, you can provision to the start of any snapshot by selecting that snapshot card from the Timeflow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.

When provisioning by LogSync information, you can provision to any point in time within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the Slide to Open LogSync control at the top of an individual snapshot card to view the time range within that snapshot. Drag the red triangle to the point in time that you want to provision from. You can also enter a date and time directly.

Related Links

- Linking a PostgreSQL dSource
- Requirements for PostgreSQL Target Hosts and Databases
- Using Pre- and Post-Scripts with dSources and VDBs
- Customizing PostgreSQL VDB Configuration Settings
SAP ASE Quick Start Topics

These topics, which are excerpted from the larger User Guide, are intended to provide you with a quick overview of working with PostgreSQL data sources in the Delphix Engine. Before undertaking any of these procedures we strongly recommend that you read the topics in the SAP ASE Support and Requirements section.

Add an ASE Environment

Link a SAP ASE Data Source

Provision an SAP ASE VDB
Add an SAP ASE Environment

Prerequisites

See Requirements for SAP ASE Source Hosts and Databases.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click the Plus icon next to Environments.
4. In the Add Environment dialog, select Unix/Linux.
5. Select Standalone Host.
6. Enter the Host IP address.
7. Enter an optional Name for the environment.
8. Enter the SSH port.
   The default value is 22.
9. Enter a Username for the environment.
10. Select a Login Type.
11. For Password, enter the password associated with the user in Step 10.

   Using Public Key Authentication
   If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
      c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
         i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
         ii. Run chmod 755 ~ to make your home directory writable only by your user.

   The public key needs to be added only once per user and per environment.

   You can also add public key authentication to an environment user’s profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

12. For Password Login, click Verify Credentials to test the username and password.
13. Enter a Toolkit Path.
    The toolkit directory stores scripts used for Delphix Engine operations. It must have a persistent working directory rather than a temporary one. The toolkit directory will have a separate sub-directory for each database instance. The toolkit path must have 0770 permissions.
14. Click the Discover SAP ASE checkbox.
15. Enter a Username for an instance on the environment.
16. Enter the Password associated with the user in Step 15.
17. Click OK.
Post-Requisites

After you create the environment, you can view information about it by selecting Manage > Environments and then selecting the environment name.
Link a SAP ASE Data Source

This topic describes the process of linking to a source database and creating a dSource.

Prerequisites

- Make sure you have correctly set up the source and target environment, as described in Managing SAP ASE Environments

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > Add dSource.
   Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
3. In the Add dSource wizard, select the source database.

   - Changing the Environment User
     If you need to change or add an environment user for the source database, see Managing SAP ASE Environment Users.

4. Enter your login credentials for the source database.
5. Click Verify Credentials.
6. Click Next.
7. Select a Database Group for the dSource.
   Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. For more information, see the topics under Users, Permissions, and Policies.
8. Click Next.
9. Select an Initial Load option and enter any additional settings needed. There are three different options for the initial load of the dSource:
   - New Full Backup - Let Delphix create a new full backup file and load it. Note that when Delphix creates the backup, it is dumped to Delphix storage, not the Backup Location specified in the next step.
   - Most Recent Existing Full Backup - Find the most recent existing full backup file in the Backup Location and load it.
   - Specific Existing Full Backup - Specify which backup files in the Backup Location that you want to load.
10. Enter the Backup Location. This is the directory where the database backups are stored.
11. Select whether the data in the database is Masked.
   This setting is a flag to the Delphix Engine that the database data is in a masked state. Selecting this option will not mask the data.
12. Enable or disable LogSync.
13. Select Backup Location Type.
14. Click Advanced to edit Retention policies, Pre and Post Scripts and External Data Directory.
15. Click Next.
16. Review the dSource Configuration and Data Management information, and then click Finish.
   The Delphix Engine will initiate two jobs, DB_Link and DB_Sync, to create the dSource. You can monitor
these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System > Event Viewer**. When the jobs have successfully completed, the database icon will change to a dSource icon on the **Environments > Databases** screen, and the dSource will be added to the list of **My Databases** under its assigned group.

---

### The dSource Card

After you have created a dSource, you can view information about it on the dSource card. You can also make modifications to its policies and permissions. To view the front of the dSource card, click the **Open** icon in the **Databases** panel. The card will then flip, showing you information such as the **Source Database** and **Data Management** configuration.

---

**Related Links**

- [Requirements for SAP ASE Source Environments](#)
- [Requirements for SAP ASE Target Environments](#)
- [Users, Permissions, and Policies](#)
Provision an SAP ASE VDB

This topic describes how to provision a virtual database (VDB) from a SQL Server dSource.

Prerequisites

- You must have linked a dSource from a source database, as described in Linking a SAP ASE Data Source, or have already created a VDB from which you want to provision another VDB.
- You must have set up target environments as described in Adding an SAP ASE Environment.
- Make sure you have the required privileges on the target environment as described in Requirements for SAP ASE Target Hosts and Databases.
- If you are provisioning to a different target environment from the one where you set up the staging database, you must make sure that the two environments have compatible operating systems, as described in Requirements for SAP ASE Target Hosts and Databases. For more information on the staging database and the validated sync process, see Managing SAP ASE Environments: An Overview.

Procedure

1. Login to the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a means of provisioning.
   See Provisioning by Snapshot and LogSync in this topic for more information.
5. Click Provision.
   The Provision VDB panel will open, and the Instance and Database Name fields will auto-populate with information from the dSource.
6. Select whether to enable Truncate Log on Checkpoint database option for the VDB.
7. Click Next.
8. Select a Target Group for the VDB.
   Click the green Plus icon to add a new group, if necessary.
9. Select a Snapshot Policy for the VDB.
   Click the green Plus icon to create a new policy, if necessary.
10. Click Next.
11. Specify any Hooks to be used during the provisioning process.
    See Customizing SAP ASE Management with Hook Operations for more information.
12. If your Delphix Engine system administrator has configured the Delphix Engine to communicate with an SMTP server, you will be able to specify one or more people to notify when the provisioning is done. You can choose other Delphix Engine users or enter email addresses.
13. Click Finish.
    When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the VDB will be included in the group you designated, and it will be listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.

Provisioning by Snapshot

You can provision to the start of any snapshot by selecting that snapshot card from the TimeFlow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.
Provisioning by LogSync

If LogSync is enabled on the dSource, you can provision by LogSync information. When provisioning by LogSync information, you can provision to any point in time within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the Slide to Open LogSync control at the top of an individual snapshot card. Drag the red triangle to the point in time from which you want to provision. You can also enter a date and time directly.

Related Links

- Linking a SAP ASE Data Source
- Adding an SAP ASE Environment
- Requirements for SAP ASE Target Hosts and Databases
- Managing SAP ASE Environments: An Overview
- Customizing SAP ASE Management with Hook Operations
Delete a VDB

Excerpt Page not found

The page: Deleting a VDB was not found. Please check/update the page name used in the 'multiexcerpt-include macro.
Delete a dSource

Excerpt Page not found

The page: **Deleting a dSource** was not found. Please check/update the page name used in the 'multiexcerpt-include macro.
Disable a dSource

Excerpt Page not found

The page: **Enabling and Disabling dSources** was not found. Please check/update the page name used in the `multiexcerpt-include` macro.
System Installation, Configuration, and Management

These topics describe installation and initial system configuration of the Delphix Engine, as well as topics related to system administration, such as user management and system monitoring.

- Installation and Initial Configuration Requirements
- Installation and Initial System Configuration
- Upgrading the Delphix Engine
- Shutting Down the Delphix Engine and Using Factory Reset
Installation and Initial Configuration Requirements

These topics cover requirements for installing and configuring the Delphix Engine, including network and connectivity requirements, memory and storage requirements, and supported web browsers and operating systems. It also includes topics on requirements for specific database platforms, such as Oracle and Oracle RAC clusters, and Microsoft SQL Server.

- Supported Web Browsers and Operating Systems
- Virtual Machine Requirements for VMWare Platform
- Virtual Machine Requirements for AWS/EC2 Platform
- General Network and Connectivity Requirements
- Checklist of Information Required for Installation and Configuration
- Virtual Machine Requirements for OpenStack with the KVM Hypervisor
Supported Web Browsers and Operating Systems

This topic describes the Web browsers and operating systems that have been tested for use with the Delphix Engine management console.

The Delphix Engine management console requires http access to the server through a Web browser with a Flash plug-in. Supported browsers and operating systems are listed in this table:

Tested Browser Configuration Matrix

<table>
<thead>
<tr>
<th>OS Supported</th>
<th>Browsers Supported</th>
<th>Adobe Flash/Flex</th>
<th>Minimum Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows XP Professional SP3</td>
<td>Internet Explorer 8.x*</td>
<td>10.x</td>
<td>4GB</td>
</tr>
<tr>
<td>Windows XP Professional SP3</td>
<td>Firefox, Chrome</td>
<td>10.x</td>
<td>4GB</td>
</tr>
<tr>
<td>Windows Vista SP2</td>
<td>Internet Explorer 8.x*, 9.x</td>
<td>10.x</td>
<td>4GB</td>
</tr>
<tr>
<td>Windows 7</td>
<td>Internet Explorer 8.x*, 9.x, 10.x, 11.x</td>
<td>10.x</td>
<td>4GB</td>
</tr>
<tr>
<td>Windows 7</td>
<td>Firefox, Chrome</td>
<td>10.x</td>
<td>4GB</td>
</tr>
<tr>
<td>Windows 7 x64</td>
<td>Internet Explorer 8.x*, 9.x, 10.x, 11.x</td>
<td>10.x</td>
<td>4GB</td>
</tr>
<tr>
<td>Windows 7 x64</td>
<td>Firefox, Chrome</td>
<td>10.x</td>
<td>4GB</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>Firefox, Chrome</td>
<td>9.0.3 (6531.9)</td>
<td>4GB</td>
</tr>
</tbody>
</table>

* IE8.x support is deprecated, and will be removed in a future Delphix Engine version.
Virtual Machine Requirements for VMWare Platform

This topic covers the virtual machine requirements, including memory and data storage, for installation of the Delphix Engine on a VMware virtualization platform.

Ideally, the Delphix Engine Virtual Machine should be placed on a server where it will not contend with other VMs for network, storage or compute resources. The Delphix Engine is an I/O intensive application, and deploying it in an environment where it must share resources with other virtual machines, especially in configurations that involve sharing I/O channels, disk spindles, and network connections, can significantly reduce virtual database performance.

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirements</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtualization Platform</td>
<td>• VMware ESX/ESXi 5.x (recommended), VMware ESX/ESXi 4.x (supported)</td>
<td>• VMware ESX/ESXi 5.0 or higher is recommended for better performance. Delphix Engine is tested and supported on ESX/ESXi versions up to 5.5 Update 2.</td>
</tr>
<tr>
<td>Virtual CPUs</td>
<td>• 8 vCPUs</td>
<td>• CPU resource shortfalls can occur under high I/O throughput conditions. CPU reservation is strongly recommended for the Delphix VM, so that Delphix is guaranteed the full complement of vCPUs even when resources are overcommitted.</td>
</tr>
<tr>
<td>Memory</td>
<td>• 128 GB vRAM (recommended)</td>
<td>• The Delphix Engine uses its memory to cache database blocks. More memory will provide better read performance.</td>
</tr>
<tr>
<td></td>
<td>• 64GB vRAM (minimum)</td>
<td>• Memory reservation is required for the Delphix VM. Performance of the Delphix Engine will be significantly impacted by over-commitment of memory resources in the ESX Server. Reservation ensures that the Delphix Engine will not stall while waiting for its memory to be paged in by the ESX Server.</td>
</tr>
</tbody>
</table>

⚠️ Do Not Allocate All Memory to the Delphix Engine
Never allocate all available physical memory to the Delphix VM. Memory for the ESX Server to perform hypervisor activities must be set aside before assigning memory to Delphix and other VMs. The default ESX minimum free memory requirement is 6% of total RAM. When free memory falls below 6%, ESX starts swapping out the Delphix guest OS. We recommend leaving about 8-10% free to avoid swapping.

For example, when running on an ESX Host with 512GB of physical memory, no more than 470GB (92%) should be allocated to the Delphix VM (and all other VMs on that host).
### Network

1. The .ova is pre-configured to use one virtual Ethernet adapter of type **VMXNET 3**. If additional virtual network adapters are desired, they should also be of type **VMXNET 3**.
2. A 10GbE NIC in the ESX Server is recommended.
3. If the network load in the ESX Server hosting the Delphix Engine VM is high, dedicate one or more physical NICs to the Delphix Engine.

- Jumbo frames are supported
- For environments having only Gigabit networks, it is possible to aggregate several physical 1GbE NICs together to increase network bandwidth (but not necessarily to reduce latency). Refer to the VMware Knowledge Base article [NIC Teaming in ESXi and ESX](https://kb.vmware.com/s/article/1007375).
- Do not aggregate NICs in the Delphix Engine VM.
- See [General Network and Connectivity Requirements](#) for information about specific port configurations, and [Network Performance Configuration Options](#) for information about network performance tuning.

### SCSI Controller

- **LSI Logic Parallel**

When adding virtual disks make sure that they are evenly distributed across the 4 virtual SCSI controllers. Spreading the disks across all available SCSI controllers will ensure optimal IO performance from the disks. For example, a VM with 4 SCSI controllers and 6 virtual disks should distribute the disks across the controllers as follows:

<table>
<thead>
<tr>
<th>Disk</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk1</td>
<td>SCSI(0:0)</td>
</tr>
<tr>
<td>disk2</td>
<td>SCSI(0:1)</td>
</tr>
<tr>
<td>disk3</td>
<td>SCSI(1:0)</td>
</tr>
<tr>
<td>disk4</td>
<td>SCSI(1:1)</td>
</tr>
<tr>
<td>disk5</td>
<td>SCSI(2:0)</td>
</tr>
<tr>
<td>disk6</td>
<td>SCSI(3:0)</td>
</tr>
</tbody>
</table>

### General Storage Configuration

See [Optimal Storage Configuration Parameters for the Delphix Engine](#)
| Delphix VM Configuration Storage | 1. The Delphix VM configuration should be stored in a VMFS volume (often called a "datastore").  
2. The VMFS volume should have enough available space to hold all ESX configuration and log files associated with the Delphix Engine. | • If a memory reservation is not enabled for the Delphix Engine (in violation of memory requirements stated above), then space for a paging area equal to the Delphix Engine's VM memory must be added to the VMFS volume containing the Delphix VM configuration data. |
| Delphix Engine System Disk Storage | 1. The Delphix Engine system disk should be stored in a VMDK.  
2. The Delphix .ova file is configured for a 150GB system drive. The VMFS volume where the .ova is deployed should therefore have at least 150GB of free space prior to deploying the .ova.  
3. The VMFS volume must be located on shared storage in order to use vMotion and HA features. | • The VMDK for the Delphix Engine System Disk Storage is often created in the same VMFS volume as the Delphix VM definition. In that case, the datastore must have sufficient space to hold the Delphix VM Configuration, the VDMK for the system disk, and a paging area if a memory reservation was not enabled for the Delphix Engine. |
| Database Storage | 1. VMDKs or RDMs operating in virtual compatibility mode can be used for database storage.  
2. A minimum of 3 VMDKs or RDMs should be allocated for database storage. | • Allocating a minimum of 3 VMDKs or RDMs for database storage enables the Delphix File System (DxFS) to make sure that its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage. |
3. If using VMDKs:
   - Each VMDK should be in a different VMFS volume
   - Each VMDK should be the only VMDK in its VMFS volume
   - The VMFS volumes should be assigned to dedicated physical LUNs on redundant storage. The VMFS volumes should not be shared with the ESX Server Console or any other Virtual Machines.

   - On vSphere 5.x, the VMDKs should be created with the Thick Provision Eager Zeroed option.
   - On vSphere 4.x, it is necessary to use a two-step process to create VMDKs that are thick provisioned and eager-zeroed. The VMDKs should be first provisioned with the Thick option (Thin provisioning not selected). After creating the VMDK, use the ESX vmkfs command to write zeroes to the VMDK before importing the VMDK into the Delphix Engine.

4. All VDMKs or RDMs should be of the same size.

   - Provisioning VMDKs from isolated VMFS volumes on dedicated physical LUNs:
     - Reduces contention for the underlying physical LUNs
     - Eliminates contention for locks on the VMFS volumes from other VMs and/or the ESX Server Console
     - Enables higher availability of the Delphix VM by allowing vSphere to vMotion the VM to a different ESX host in the event of a failure of the Delphix ESX host

   - If the underlying storage array allocates physical LUNs by carving them from RAID groups, the LUNs should be allocated from different RAID groups. This eliminates contention for the underlying disks in the RAID groups as the Delphix Engine distributes IO across its storage devices.

   - If the storage array allocates physical LUNs from storage pools comprising dozens of disk drives, the LUNs should be distributed evenly across the available pools.

   - Using thin-provisioned LUNs in the storage array for VMFS volumes can be useful if you anticipate adding storage to the Delphix Engine in the future. In this case, the LUNs should be thin-provisioned with a size larger than the amount of storage that will be initially allocated to the Delphix Engine. When you want to add more storage to the Delphix Engine, use vSphere to expand the size of the VMDKs. Be sure to specify that the additional storage is also thick-provisioned and eager-zeroed.
5. The physical LUNs used for VMFS volumes and RDMs should be of the same type in terms of performance characteristics such as latency, RPMs, and RAID level. In addition, the total number of disk drives that comprise the set of physical LUNs should be capable of providing the desired aggregate I/O throughput (MB/sec) and IOPS (Input/Output Operations per Second) for all virtual databases that will be hosted by the Delphix Engine.

6. The physical LUNs used for VMFS volumes can be thin-provisioned in the storage array.

7. For best performance, the LUNs used for RDMs should **not** be thin-provisioned in the storage array, but should be thick-provisioned with a size equal to the amount of storage that will be initially allocated to the Delphix Engine. The RDM can be expanded in the future when more storage is needed.

8. Shared storage is required in order to use vMotion and HA features.

In addition to making sure the latest VMware patches have been applied, check with your hardware vendor for updates specific to your hardware configuration.

**Known Issues**

According to the following HP advisory, "On HP ProLiant servers configured with any of the HP Smart Array Controllers listed in the Scope section (below) and running VMware ESXi 5.0, 5.1, or 5.5, or Red Enterprise Hat Linux 6 or 7, an out-of-memory condition may lead to a server halt and purple screen after upgrading to HP Smart Array Controller Driver (hpsa) Version 5.x.0.58-1 (ESXi 5.0 and ESXi 5.1), Version 5.5.0.58-1 (ESXi 5.5), or Version 3.4.4-125 (Red Hat Enterprise Linux).":

Virtual Machine Requirements for AWS/EC2 Platform

This topic covers the virtual machine requirements, including memory and data storage, for installation of the Delphix Engine on Amazon's Elastic Cloud Compute platform.

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirements</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Types</td>
<td>Storage Optimized Instances</td>
<td>• The Delphix Engine most closely resembles a storage appliance and performs best when provisioned using a storage optimized instance type</td>
</tr>
<tr>
<td></td>
<td>• i2.2xlarge</td>
<td>• Larger instance types provide more CPU, which can prevent resource shortfalls under high I/O throughput conditions</td>
</tr>
<tr>
<td></td>
<td>• i2.4xlarge</td>
<td>• Larger instances also provide more memory, which the Delphix Engine uses to cache database blocks. More memory will provide better read performance.</td>
</tr>
<tr>
<td></td>
<td>• i2.8xlarge</td>
<td></td>
</tr>
</tbody>
</table>
| Network Configuration | • Virtual Private Cloud  
• Static Public IP  
• Security Group Configuration | • EC2 instances can be launched in either an EC2-Classic network or a Virtual Private Cloud (VPC) network. You must deploy the Delphix Engine and all of the source and target environments in a VPC network to ensure that private IP addresses are static and do not change when you restart instances.  
• When adding environments to the Delphix Engine, you must use the host's VPC IP address  
• The EC2 instance must be launched with a static public IP address. The default behavior for VPC instances is to launch with a dynamic public IP address which can change whenever you restart the instance. This can be done using AWS Elastic IP Addresses.  
• The default security group will only open port 22 for secure shell (SSH) access. You must modify the security group to allow access to all of the networking ports used by the Delphix Engine and the various source and target platforms. See General Network and Connectivity Requirements for information about specific port configurations.  
• See Network Performance Configuration Options for information about network performance tuning |
| EBS Configuration | • EBS Provisioned IOPS Volumes  
• EBS Optimized Instance (except for i2.xlarge instance type) | • All attached storage devices must be EBS volumes. Delphix does not support the use of instance store volumes.  
• Because EBS volumes are connected to EC2 instances via the network, other network activity on the instance can affect throughput to EBS volumes. EBS optimized instances provide guaranteed throughput to EBS volumes and are required (for instance types that support it) in order to provide consistent and predictable storage performance. The i2.8xlarge instance type does not support EBS optimized instances; however, this instance type supports 10 Gigabit networking that often provides suitable performance.  
• Use EBS volumes with provisioned IOPs in order to provide consistent and predictable performance. The number of provisioned IOPs depends on the estimated IO workload on the Delphix Engine. Provisioned IOPs volumes must be configured with a volume size at least 30 GiB times the number of provisioned IOPs. For example, a volume with 3,000 IOPS must be configured with at least 100 GiB.  
• I/O requests of up to 256 kilobytes (KB) are counted as a single I/O operation (IOP) for provisioned IOPs volumes. Each volume can be configured for up to 4,000 IOPs. |
| General Storage Configuration | • Allocate initial storage equal to the size of the physical source databases. For high redo rates and/or high DB change rates, allocate an additional 10-20% storage.  
• Add storage when storage capacity approaches 30% free  
• Keep all EBS volumes the same size. Add new storage by provisioning new volumes of the same size.  
• Maximize Delphix Engine RAM for a larger system cache to service reads  
• Use at least 3 EBS volumes to maximize performance. This enables the Delphix File System (DxFS) to make sure that its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.  
• See [Optimal Storage Configuration Parameters for the Delphix Engine](#) |

---

See [Optimal Storage Configuration Parameters for the Delphix Engine](#)
General Network and Connectivity Requirements

This topic covers the general network and connectivity requirements for the Delphix Engine, including connection requirements, port allocation, and firewall and Intrusion Detection System (IDS) considerations. For platform-specific network and connectivity requirements, see the relevant topics under the Requirements section for each platform.

General Port Allocation

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

### General Outbound from the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>25</td>
<td>Connection to a local SMTP server for sending email</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>53</td>
<td>Connections to local DNS servers</td>
</tr>
<tr>
<td>UDP</td>
<td>123</td>
<td>Connection to an NTP server</td>
</tr>
<tr>
<td>UDP</td>
<td>162</td>
<td>Sending SNMP TRAP messages to an SNMP Manager</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
<td>SSL connections from the Delphix Engine to the Delphix Support upload server</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>636</td>
<td>Secure connections to an LDAP server</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Connections to a Delphix replication target (see Configuring Replication)</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections to source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>

### General Inbound to the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>80</td>
<td>HTTP connections to the Delphix GUI</td>
</tr>
<tr>
<td>Protocol</td>
<td>Port</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>UDP</td>
<td>161</td>
<td>Messages from an SNMP Manager to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>443</td>
<td>HTTPS connections to the Delphix GUI</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections from source and target environments for network performance tests via the Delphix CLI (see <a href="#">Network Performance Tool</a>)</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>32768-65535</td>
<td>Required for NFS mountd and status services from target environment only if the firewall between Delphix and the target environment does not dynamically open ports. <strong>Note:</strong> If no firewall exists between Delphix and the target environment, or the target environment dynamically opens ports, this port range is not explicitly required.</td>
</tr>
</tbody>
</table>

**Firewalls and Intrusion Detection Systems (IDS)**

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

**SSHD Configuration**

Both source and target Unix environments are required to have `sshd` running and configured such that the Delphix Engine can connect over `ssh`.

The Delphix Engine expects to maintain long-running, highly performant `ssh` connections with remote Unix environments. The following `sshd` configuration entries can interfere with these `ssh` connections and are thus disallowed:

**Disallowed sshd Configuration Entries**
ClientAliveInterval

ClientAliveCountMax

Related Links

- Network and Connectivity Requirements for SQL Server Environments
- Network and Connectivity Requirements for PostgreSQL Environments
- Network and Connectivity Requirements for Windows Environments
- Network and Connectivity Requirements for Unix Environments
- Network and Connectivity Requirements for Oracle Environments
- Network and Connectivity Requirements for SAP ASE Environments
Checklist of Information Required for Installation and Configuration

This topic describes the information that is required for initial installation and configuration of the Delphix Engine.

Information Required for Installation

- Name of the Delphix Engine
- Network configuration with static DHCP lease or static IP address
  - Static IP address in CDIR notation (for example, 10.80.142.82/24) (not required for static DHCP configuration)
  - Subnet mask
  - Gateway IP address
  - DNS server IP address(es)
  - Domain of the Delphix Engine installation, such as mydelphix.com
- Selection of storage devices for the operating system disks. See Virtual Machine Requirements for VMWare Platform for more information on memory and storage requirements.

Information Required for Initial Configuration

- NTP server used for setting system time (optional)
- DNS server for name resolution (not required for static IP configuration or if DHCP supplies the DNS servers)
- Web proxy server (if used)
- SMTP server to support email notification
  - IP address or name (for example, smtp.gmail.com) and port number
  - Email address from which you want the Delphix Engine to send email
- If using LDAP for authentication, server information, including name or IP address and port number
- If using SNMP for notifications, server name and IP address, port number, and community string
- Desired password for sysadmin user (this default user can be replaced) - For the ServerSetup application
  - IP address or name and port number
- Email address for messages sent to the sysadmin user
- Desired password for delphix_admin user (this default user can be replaced) - For the Delphix Admin application
  - See The delphix_admin and sysadmin User Roles for more information.
- Email address for messages sent to the delphix_admin user

Related Links

- Virtual Machine Requirements for VMWare Platform
- The delphix_admin and sysadmin User Roles
### Virtual Machine Requirements for OpenStack with the KVM Hypervisor

This topic covers the virtual machine requirements, including memory and data storage, for installation of the Delphix Engine on an OpenStack compute node running on the KVM virtualization platform.

The Delphix Engine Virtual Machine (VM) should be placed on a compute node where it will not contend with other VMs for network, storage, or compute resources. The Delphix Engine is an I/O intensive application; deploying it in an environment where it must share resources with other virtual machines can significantly reduce virtual database performance, especially in configurations that involve sharing I/O channels, disk spindles, and network connections.

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirements</th>
<th>Notes</th>
</tr>
</thead>
</table>
| Virtualization Platform    | • Ubuntu versions >= 12.04 and RHEL versions >= 6 are the supported Linux versions for the OpenStack compute node.  
• Required OpenStack services for administering Delphix include Compute (Nova), Image (Glance), Block Storage (Cinder), Networking (Neutron) or legacy networking (Nova Network). | To set the vCPU model for your compute node, add the following lines to the [libvirt] section of `nova.conf` (see list to the left for acceptable `cpu_model` values):
  ```
  cpu_mode = custom
  cpu_model = Westmere
  virt_type = kvm
  ``` |
| Virtual CPUs               | • 8 vCPUs                                                                   |                                                                      |
|                            | • vCPUs must be model Westmere (preferred if supported by physical CPU), Nehalem, Penryn, Conroe, or kvm64.       |                                                                      |
### Memory
- 128 GB vRAM (recommended)
- 64 GB vRAM (minimum)

- The Delphix Engine uses its memory to cache database blocks. More memory will provide better read performance.

- Memory overcommit should be disabled on the compute node by setting where the Delphix VM is running, if possible. Overcommit causes the Delphix Engine to stall while waiting for its memory to be paged in by the compute node. You can disable Overcommit by adding the following line to the `[DEFAULT]` section of `nova.conf`:

  ```bash
  ram_allocation_ratio = 1.0
  ```

  Alternatively, you can simply run the Delphix Engine as the sole VM on the OpenStack Compute node where it is located.

### Network
- If the network load is high in the compute node hosting the Delphix Engine VM, dedicate one or more physical NICs to the Delphix Engine.

- A 10GbE NIC in the compute node is recommended.

- The Delphix Engine installation image is pre-configured to use one virtual Ethernet adapter of type virtio. If you want additional virtual network adapters, they should also be of type virtio.

- You can launch instances in either a Networking (Neutron) or legacy (Nova Networking) network.

- Jumbo frames are supported.

- See [General Network and Connectivity Requirements](#) for information about specific port configurations, and [Network Performance Configuration Options](#) for information about network performance tuning.
| Delphix Engine System Disk Storage | • The Delphix Engine system disk must be created from the installation image with format qcow2.  
• The Delphix Engine installation image is configured for a 150GB system drive. The physical location where the volume is deployed should therefore have at least 150GB of free space prior to deploying the Delphix Engine VM. |
| --- | --- |
| Database Storage Configuration | • Use the OpenStack Block Storage service (Cinder) for provisioning volumes for database storage.  
• Allocate at least 3 volumes for database storage.  
• All volumes should have the same capacity.  
• If the Block Storage driver for provisioning storage allows it, volumes should be thick provisioned and eager zeroed.  
• Volumes should be assigned to dedicated physical LUNs on redundant storage.  
• The physical resources used for volume storage should be of the same type in terms of performance characteristics such as latency, RPMs, and RAID level.  
• Allocate initial storage equal to the size of the physical source databases. For high redo rates and/or high DB change rates, allocate an additional 10-20% storage.  
• Add storage when storage capacity approaches 70% used.  
• Use at least 3 volumes to maximize performance. This enables the Delphix File System (DxFS) to make sure that its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.  
• The set of disk drives that comprise the set of physical LUNs must be capable of providing the desired aggregate I/O throughput (MB/sec) and IOPS (Input/Output Operations per Second) for all virtual databases that will be hosted by the Delphix Engine.  
• Different Block Storage drivers will have different configuration options. For generalized advice on setting up high-performance Delphix Engine database storage, see [Optimal Storage Configuration Parameters for the Delphix Engine](#). |
Installation and Initial System Configuration

These topics describe the initial installation and configuration of the Delphix Engine, the delphix_admin and sysadmin roles, and using the system console.

- The delphix_admin and sysadmin User Roles
- Using HostChecker to Confirm Source and Target Environment Configuration
- Installing the Delphix Engine
- Setting Up Network Access to the Delphix Engine
- Customizing the Delphix Engine System Settings
- Configuring SNMP
- Setting Up the Delphix Engine
- Retrieving the Delphix Engine Registration Code
The delphix_admin and sysadmin User Roles

This topic describes the function of the delphix_admin and sysadmin roles.

After installation, the Delphix Engine creates a sysadmin user with the initial password sysadmin. The sysadmin launches the initial ServerSetup configuration application and has access to a command-line system administration console. Through the command line console or the ServerSetup application the sysadmin can also undertake typical system administration duties such as managing memory, storage, and support logs for the Delphix Engine, and performing upgrades and patches.

After initial configuration, the delphix_admin user manages the Delphix Engine’s user data objects: dSources, virtual databases (VDBs), users, groups, and related policies and resources, all collectively referred to as the Delphix Engine Domain. The delphix_admin user manages the Delphix Engine domain using either the Command Line Interface (CLI) or the browser-based Delphix Admin application.

Email addresses are required inputs for both the sysadmin and delphix_admin accounts, and you can create additional sysadmin and delphix_admin users as described in the topics under Managing System Administrators.

Related Links

- Managing System Administrators
Using HostChecker to Confirm Source and Target Environment Configuration

This topic describes the HostChecker script that is used to check the configuration of source and target environments.

The HostChecker is a standalone program which validates that host machines are configured correctly before the Delphix Engine uses them as data sources and provision targets.

Please note that HostChecker does not communicate changes made to hosts back to the Delphix Engine. If you reconfigure a host, you must refresh the host in the Delphix Engine in order for it to detect your changes.

You can run the tests contained in the HostChecker individually, or all at once. You must run these tests on both the source and target hosts to verify their configurations. As the tests run, you will either see validation messages that the test has completed successfully, or error messages directing you to make changes to the host configuration.

See these topics for more specific information about the tests that are run for each data platform supported by the Delphix Engine:

- [Using HostChecker to Validate SQL Server Source and Target Environments](#)
- [Using HostChecker to Validate PostgreSQL Source and Target Environments](#)
- [Using HostChecker to Validate Oracle Source and Target Environments](#)
- [Using HostChecker to Validate SAP ASE Source and Target Environments](#)
Installing the Delphix Engine

This topic describes how to install the Delphix Engine from the OVA file.

Prerequisites

Make sure you have read the requirement and support information in the Installation and Initial Configuration Requirements topics.

Procedure to install an OVA

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements and deploys to one 150 GB hard disk with 8 vCPUs and 64GB RAM. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from the Delphix download site or obtain it from your Delphix solutions architect.

2. Log in using the vSphere client to the vSphere server (or vCenter Server) where you want to install the Delphix Engine.

3. In the vSphere Client, select File > Deploy OVA Template.

4. Browse to the OVA file, and then click Next.

5. Select a hostname for the Delphix Engine.

6. Select the data center where the Delphix Engine will be located.

7. Select the Cluster and the ESX host.

8. Select a data store that has at least 166 GB free space. (This will need to be larger if you plan to increase the size of memory on the VM from 16GB)

9. Select a disk format.

10. Select the virtual network you want to use.

11. Click Finish.

   The installation will begin and the Delphix Engine will be created in the location you specified.

Procedure to install an AMI

Use the Delphix-supplied AMI file to install the Delphix Engine.

1. On the Delphix download site, click the AMI you would like to share, and accept the Delphix License agreement, or follow a link given by your Delphix solutions architect.

2. On the Amazon Web Services Account Details form presented:
   a. Enter your AWS Account Identifier, which can be found here: https://console.aws.amazon.com/billing/home#/account
   b. Select which AWS Region you would like the AMI to be shared in. If you would like the AMI shared in a different region, please contact your Delphix account representative to make the proper arrangements.

3. Click Share.

4. The Delphix Engine will appear in your list of AMIs in AWS momentarily.
5. Reference the Installation and Configuration Requirements for AWS/EC2 when deploying the AMI.

Post-Requisites

After installing the server, follow the procedures in these topics to specify and customize the Delphix Engine network, and to make modifications to the memory size, number of CPUs, and number of disks used for storage.

- Setting Up Network Access to the Delphix Engine
- Customizing the Delphix Engine System Settings
Setting Up Network Access to the Delphix Engine

This topic describes how to configure the Delphix Engine network during initial installation.

Prerequisites

Follow the initial installation instructions in Installing the Delphix Engine.

NAT Configuration

Delphix communicates its IP address in application layer data and this cannot be translated by NAT.

Procedure

1. Power on the Delphix Engine and open the Console.
2. Wait for the Delphix Management Service and Delphix Boot Service to come online.
3. Press F2 to access the sysadmin console.
4. Enter sysadmin@SYSTEM for the username and sysadmin for the password.
5. You will be presented with a description of available network settings and instructions for editing.
Delphix Engine Network Setup

To access the system setup through the browser, the system must first be configured for networking in your environment. From here you can configure the primary interface, DNS, hostname, and default route. When DHCP is configured, all other properties are derived from DHCP settings.

To see the current settings, run 'get'. To change a property, run 'set <property>=<value>'. To commit your changes, run 'commit'. To exit this setup and return to the standard CLI, run 'discard'.

- **defaultRoute**: IP address of the gateway for the default route (e.g. '1.2.3.4').
- **dhcp**: Boolean value indicating whether DHCP should be used for the primary interface. Setting this value to 'true' will cause all other properties (address, hostname, and DNS) to be derived from the DHCP response.
- **dnsDomain**: DNS Domain (e.g. 'delphix.com').
- **dnsServers**: DNS server(s) as a list of IP addresses (e.g. '1.2.3.4,5.6.7.8').
- **hostname**: Canonical system hostname, used in alert and other logs (e.g. 'myserver').
- **primaryAddress**: Static address for the primary interface in CIDR notation (e.g. '1.2.3.4/22').

Current settings:

- defaultRoute: 192.168.1.1
- dhcp: false
- dnsDomain: example.com
- dnsServers: 192.168.1.1
- hostname: Delphix
- primaryAddress: 192.168.1.100/24

6. **Configure the hostname**. If you are using DHCP, this step can be skipped.

   delphix network setup update *> set hostname=<hostname>

   **⚠️ The hostname should be the same name you entered during the server installation.**

7. **Configure DNS**. If you are using DHCP, this step can be skipped.

   delphix network setup update *> set dnsDomain=<domain>
   delphix network setup update *> set dnsServers=<server1-ip>[,<server2-ip>,...]

8. **Configure either a static or DHCP address**: 

DHCP Configuration

delphix network setup update *> set dhcp=true

Static Configuration

delphix network setup update *> set dhcp=false

delphix network setup update *> set primaryAddress=<address>/<prefix-len>

⚠️ The static IP address must be specified in CIDR notation (for example, 192.168.1.2/24)

9. Configure a default gateway. If you are using DHCP, this step can be skipped.

delphix network setup update *> set defaultRoute=<gateway-ip>

10. Commit your changes. Note that you can use the get command prior to committing to verify your desired configuration.

   delphix network setup update *> commit
   Successfully committed network settings. Further setup can be done through the browser at:

       http://<address>

   Type 'exit' to disconnect, or any other commands to continue using the CLI.

11. Check that the Delphix Engine can now be accessed through a Web browser by navigating to the displayed IP address, or hostname if using DNS.

12. Exit setup:

   delphix> exit
Customizing the Delphix Engine System Settings

This topic describes how to customize the initial system set up requirements for memory, number of CPUs, storage disks, and network configuration.

The OVA file that you use to install the Delphix Engine is configured for the minimum system requirements. You can customize these to match the capabilities of your specific system.

Prerequisites

- Follow the initial installation instructions in Installing the Delphix Engine.

Procedure

1. Shut down the guest operating system and power off the Delphix Engine.
3. You can now customize the system settings.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Size</td>
<td>Set to 64GB or larger based on sizing analysis. In the “Resource Allocation” panel, ensure that “Reserve all guest memory” is checked.</td>
</tr>
<tr>
<td>Number of CPUs</td>
<td>Allocate 8 vCPUs or more based on licensing. vCPUs should be fully reserved to ensure that the Delphix Engine does not compete for CPU cycles on an over-committed host.</td>
</tr>
<tr>
<td>Disks for Data Storage</td>
<td>Add virtual disks to provide storage for user data such as dSources and VDBs. The underlying storage must be redundant. Add a minimum of 150GB per storage disk. All virtual disks should be the same size and have the same performance characteristics. If using VMFS, use thick provisioned, eager zeroed disks.</td>
</tr>
<tr>
<td>Data Storage Multipathing Policy</td>
<td>For EMC storage, the multipathing policy should always be set to roundrobin (default for 5.X). Additionally, the I O Operation Limit should be changed from the default of 1000 to 1. This should be strongly considered for other storage platforms as well. See VMware KB article EMC VMAX and DMX Symmetrix Storage Array Recommendations for Optimal Performance on VMware ESXi/ESX</td>
</tr>
<tr>
<td>Network</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>The network configuration is set to have a VMXNET3 network adapter. VMXNET3 is a tuned network interface that is included with the VMtools provided in the OVA file. It will be assigned to VM Network.</td>
<td></td>
</tr>
</tbody>
</table>

### JUMBO Frames
VMXNET3 supports Ethernet jumbo frames, and this can be used to maximize throughput and minimize CPU utilization.

### Adding Link Aggregation via VMware NIC Teaming
To increase throughput or for failover, add multiple physical NICs to the vSwitch that is connected to the Delphix Engine. To increase throughput, NIC Teaming must use the Route Based on IP Hash protocol for load balancing. See VMware KB article [Troubleshooting IP-Hash outbound NIC selection](#).

### Dedicate Physical NICs to the Delphix Engine
For best performance assign the Delphix Engine to network adapters that are used exclusively by Delphix.

---

**Post-Requisites**

- After making any changes to the system settings, power on the Delphix Engine again and proceed with the initial system configuration as described in [Setting Up the Delphix Engine](#).
Configuring SNMP

This topic describes how to configure SNMP.

SNMP is a standard protocol for managing devices on IP networks. The Delphix Engine can be configured to send alerts to an external SNMP manager.

Prerequisites

1. At least one SNMP manager must be available, and must be configured to accept SNMPv2 InformRequest notifications.
2. Delphix's MIB (Management Information Base) files must be installed on the SNMP manager or managers. These MIB files describe the information that the Delphix Engine will send out. They are attached to this topic.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Attachment Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELPHIX-ALERT-MIB.txt</td>
<td>/download/attachments/44801170/DELPHIX-ALERT-MIB.txt?version=1&amp;modificationDate=1395063745239</td>
</tr>
<tr>
<td>DELPHIX-MIB.txt</td>
<td>/download/attachments/44801170/DELPHIX-MIB.txt?version=1&amp;modificationDate=1363114824507</td>
</tr>
</tbody>
</table>

Procedure

1. Log into the Server Setup application using sysadmin credentials.
2. Select Server Preferences > SNMP Configuration.
3. Select Use SNMP.
4. Set the severity level of the messages you want to be sent to the SNMP manager(s).
5. Click the + icon.
6. Enter an SNMP Manager hostname / IP address.
   Provide a community string and adjust the port number if necessary.
7. Click Save.
   The newly-entered manager will appear in the list.
8. An attempt will be made to connect with the SNMP manager by transmitting an informational level message. If a response is received from the manager within 20 seconds, a checkmark will appear along with the manager entry. If not, a red X will appear – check your settings and try again.
Setting Up the Delphix Engine

This topic describes how to set up the initial system configuration for the Delphix Engine, including system time, storage, web proxy, SMTP server, email to Delphix Support, and LDAP authentication.

Prerequisites

You must have sysadmin privileges to perform this procedure. See The delphix_admin and sysadmin User Roles for more information.

Procedure

The setup procedure uses a wizard process to take you through five configuration screens:

- System Time
- Network
- Storage
- Serviceability
- Authentication Service

   The ServerSetup application will launch when you connect to the server.

2. Enter your login credentials and then click Next.

System Time

1. Select an option for maintaining the system time.

<table>
<thead>
<tr>
<th>Option</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set NTP Server</td>
<td>After selecting this option, select an NTP server from the list, or click Add NTP Server to manually enter a server.</td>
</tr>
<tr>
<td></td>
<td>When configuring Delphix Engine on VMware, be sure to configure the NTP client on the host to use the same servers that you enter here. On a vSphere client, the NTP client is set in the Security Profile section of the configuration process.</td>
</tr>
</tbody>
</table>

| Manually Select Time and Date | Click Use Browser Time and Date to set the system time, or select the date and time by using the calendar and clock displays. Then select the Time Zone. |
|                              | If you select Use Browser Time and Date, the date and time will persist as your local time, even if you change the time zone.                |
|                              | Snapshots from dSources and VDBs reflect the time zone of the source or target environment, not that of the Delphix Engine.             |

2. Click Next.
Network Configuration

The initial out-of-the-box network configuration in the OVA file is set to use a VMXNET3 network adapter.

1. Under Network Interfaces, click Settings.
2. The Network Interface is Enabled by default.
3. Select DHCP or Static network addressing. For Static addressing, enter an IP Address and Subnet Mask.
   - The static IP address must be specified in CIDR notation (for example, 192.168.1.2/24)
4. Select whether to use Jumbo Frames. VMXNET3 supports Ethernet jumbo frames, which can be used to maximize throughput and minimize CPU utilization.
5. Click Save.
6. Repeat for any other interfaces you have added to the virtual machine (VM) that you wish to configure. They will not be enabled by default.
7. Under Routing, enter a Default Gateway.
8. Under DNS Services, enter a DNS Domain Name and DNS Server.
9. Click Next.

Storage

The Delphix Engine automatically discovers and displays storage devices. For each device, set the Usage Assignment to Data and set the Storage Profile to Striped.

You can associate additional storage devices with the Delphix Engine after initial configuration, as described in Adding and Expanding Storage Devices.

Storage Disk Usage Assignment Options

- Data
  - Once you set the storage unit assignment for a disk to Data and save the configuration, you cannot change it again.
- Unassigned
  - These are disks being held for later use.
- Unused
  - These disks can be configured later to add capacity for existing data disks.

The Minimum Number of Storage Disks

- You should configure at least 3 disks for storage of user data. This makes the Delphix Engine storage manager function more efficiently, since duplicated metadata can be distributed across multiple disks.
Serviceability

1. If a Web Proxy Server is necessary for your environment, select **Use a Web Proxy** and enter the required information.

2. If you want the Delphix Engine to send information to the Delphix Support site periodically over https, select **Phone Home Service enabled**. This feature requires a connection to the internet and will use the **Web Proxy Server** configuration.

3. Select **Use an SMTP Server** and enter the required information to enable email notifications for events and alerts.

   - **Setting Up an SMTP Server for Alert Emails**
     
     When a critical fault occurs with the Delphix Engine, it will automatically send an email alert to the `delphix_admin` user. Make sure that you configure the SMTP server so that alert emails can be sent to this user. See **System Faults** for more information.

Authentication Service

- **To avoid configuration issues, consult with your light weight directory access protocol (LDAP) administrator before attempting to set up LDAP authentication of users for the Dephix Engine.**

1. Select **Use LDAP** to enable LDAP authentication of users.

2. Enter the **LDAP Server** IP address or hostname, and **Port** number.

3. Select the **Authentication** method.

4. Select whether you want to **Protect LDAP traffic with SSL/TLS**.
   
   If you select this option, you must import the server certificate.

5. When you are done with the LDAP configuration, click **Test Connection**.

6. Click **Next**.

   - **LDAP Authentication When Adding Users**
     
     If LDAP has been set up as the authentication service for the Delphix Engine, you must add new users with LDAP as their authentication mechanism. See **Adding Delphix Users and Privileges** for more information. Note that you can only add individual LDAP users, not groups.

Registration

If your local machine is connected to the external Internet, you can auto-register the Delphix Engine:

1. Enter your **Support Username** and **Support Password**.

2. Click **Register**.

If external connectivity is not immediately available, you must perform manual registration.

1. Copy the **Delphix Engine registration code** in one of two ways:
   
   a. Manually highlight the registration code and copy it to clipboard. Or,
   
   b. Click **Copy Registration Code to Clipboard**.
2. Transfer the Delphix Engine's registration code to a location with an external network connection. For example, this could be accomplished by e-mailing the registration code to an externally accessible e-mail account.


4. Log in with your support credentials and paste the Registration Code.

5. Click Register.

⚠️ While your Delphix Engine will work without registration, we strongly recommend that you register each Delphix Engine as part of setup. Failing to register the Delphix Engine will impact its supportability and security in future versions.

**Summary**

1. Review your configurations for System Time, Network, Storage, Serviceability, and Authentication. Click Modify to change the configuration for any of these server settings.

2. Click Finish.

3. Click Yes to confirm that you want to save the configuration.

4. Click OK to acknowledge the successful configuration.

**Post-Requisites**

- After configuration is complete, the Delphix Engine will restart and launch the browser-based Delphix Admin interface. The URL for this will be http://<Delphix Engine>/Server.html.
- After the Delphix Admin interface launches, the delphix_admin can log in using the username delphix_admin and password delphix.
- You can access the ServerSetup application at any time by navigating to http://<Delphix Engine>/ServerSetup.html and entering the sysadmin credential.

**Related Links**

- [The delphix_admin and sysadmin User Roles](#)
- [System Faults](#)
- [Adding Delphix Users and Privileges](#)
- [Adding and Expanding Storage Devices](#)
Retrieving the Delphix Engine Registration Code

This topic describes how to retrieve the registration code for a Delphix Engine. It is strongly recommended that registration be performed as a part of Delphix Engine setup. However, you may also retrieve the registration code for a Delphix Engine following setup.

Procedure

1. Retrieving the Delphix Engine Registration Code can be done through the ServerSetup application after logging in with the sysadmin credentials.
2. In the Registration panel, click the View button.
3. The Registration Code is displayed in the bottom half of the Registration window.
4. If your local machine is connected to the external Internet, you can auto-register the Delphix Engine by entering your Support Username and Support Password, and then clicking Register.
5. If external connectivity is not immediately available, manual registration must be performed.
   a. Copy the Delphix Engine registration code by either manually highlighting and copying to clipboard or clicking Copy Registration Code to Clipboard.
   b. Transfer the Delphix Engine's registration code to a location with an external network connection. For example, this could be accomplished by e-mailing the registration code to an externally accessible e-mail account.
   c. On a machine with external network access, use your browser to navigate to the Delphix Registration Portal at https://register.delphix.com.
   d. Log in with your support credentials and paste the Registration Code.
   e. Click Register.

⚠️ While your Delphix Engine will work without registration, it is strongly recommended that you register each Delphix Engine as part of setup. Failing to register the Delphix Engine will impact its supportability and security in future versions.

Post-Requisites

- Following registration, you will receive an e-mail confirming the registration of your Delphix Engine.
Upgrading the Delphix Engine

These topics describe processes for upgrading the Delphix Engine.

- Upgrading or Patching a New Version of the Delphix Engine
- Upgrading to a New Version of the Delphix Engine
- Upgrading VM Tools and Hardware
Upgrading or Patching a New Version of the Delphix Engine

This topic describes how to upgrade to a new version of the Delphix Engine.

Prerequisites

⚠️ Upgrades of the Delphix Engine are performed with the assistance of Delphix support. A support engineer will be responsible for performing the steps described below. In preparation for any upgrade, a support case must be opened and the upgrade planned with the assistance of your support engineer.

Because the Delphix Engine may experience brief downtime while it is being upgraded to the new version, you should schedule appropriate downtime for your VDB applications. The VDBs themselves will be automatically disabled during upgrade by the Delphix Engine. The amount of downtime will be proportional to the number of VDBs.

Aside from VDBs being inaccessible during the upgrade, only system administrator users will be able to login to the Delphix Engine. The upgrade will automatically logout any logged in users, and will prevent new users from logging in.

Procedure

1. Upload an upgrade image through the Delphix Setup GUI.
   a. Log in to the Server Setup application.
   b. In the System Upgrade Management panel, click View.
   c. Click the up arrow to upload a new version

2. Using the command line interface, login as system administrator.

3. Navigate to system version and list existing versions by entering list.
   The version you uploaded in step 1 will be listed there. Select it by name as follows.

   ```
   delphix> system version
   delphix system version> list
   NAME     STATUS             OSRUNNING  BUILDDATE
   4.0.6.0  UPLOADED           false      2014-06-17T03:12:48.000Z
   4.0.5.0  CURRENTLY_RUNNING  true       2014-06-10T14:41:28.000Z
   delphix system version> select 4.0.6.0
   delphix system version "4.0.6.0">
   ```

4. Start the upgrade by entering apply, then commit.
   Following commit, progress will be displayed as the upgrade executes. When the Delphix Engine is rebooted to the new version, your CLI session will automatically be terminated. You will be able to log in again when the Delphix Engine has successfully started running the new version.
Deferred OS Upgrade

Each Delphix Engine upgrade image contains software for DelphixOS (the operating system that runs Delphix) in addition to Delphix management software. DelphixOS is versioned, and the OS version that is delivered with any given Delphix Engine version is displayed as "osVersion" in the version properties. By default, when applying a new version, if that version delivers a new OS (newer than what is currently running), the system will reboot to the new OS as part of the upgrade process. This requires scheduling downtime for VDBs, as VDBs are disabled during upgrade.

In some cases, it may be possible to defer upgrading DelphixOS even when a new version is included in the upgrade image. You can determine if this is a possibility by comparing the "minOsVersion" property of the new version with the "osVersion" that is currently running. For example:

```
  delphix system version> list
  NAME     STATUS             OSRUNNING  BUILDDATE
  4.0.6.0  UPLOADED           false      2014-06-17T03:12:48.000Z
  4.0.5.0  CURRENTLY_RUNNING  true       2014-06-10T14:41:28.000Z
```

Here, the running OS comes from version 4.0.5.0, and we’d like to see if the OS version in 4.0.5.0 meets the minimum requirements for 4.0.6.0, the version we’re upgrading to:

```
  delphix system version> select 4.0.5.0 get osVersion
  4.0.2014.07.01
  delphix system version> select 4.0.6.0 get osVersion
  4.0.2014.06.07
  delphix system version> select 4.0.6.0 get minOsVersion
  4.0.2014.04.24
```

Here, although 4.0.6.0 includes a newer version of DelphixOS than what is currently running, its minimum OS version requirement is met by the currently running OS. We may then choose to defer upgrading DelphixOS when upgrading from 4.0.5.0 to 4.0.6.0 by setting the "defer" property in the "apply" context. In general, one can defer upgrading DelphixOS when the currently running OS version is greater than or equal to the minimum OS version requirements of the version being upgraded to. When you do such a deferred OS upgrade, the OS version will still be installed, but the system will simply not reboot to that new version. The Delphix Engine will still restart to the new
version, but this restart will not result in downtime for VDBs. After that point, the "STATUS" column of the running version will show "DEFERRED" instead of "CURRENTLY_RUNNING" indicating that while this version is running, the OS upgrade was deferred.

The OS can later be updated to the current version by applying the running version again and not setting the defer property. When you do this, the system will simply reboot to the current version of DelphixOS (this will result in downtime for your VDBs).
Upgrading to a New Version of the Delphix Engine

This topic describes how to upgrade to a new version of the Delphix Engine.

**Prerequisites**

> Upgrades of the Delphix Engine are performed with the assistance of Delphix support. In preparation for such upgrades, you must open a support case and plan the upgrade with the assistance of a support engineer. The support engineer will be responsible for performing the upgrade procedure.

During the upgrade, VDBs will be inaccessible, and only system administrator users will be able to login to the Delphix Engine. The upgrade will automatically logout any users who are logged in at the time and will prevent new users from logging in.

**Uploading an Upgrade Version**

Before you can upgrade a Delphix Engine, you must upload to the engine an upgrade file for the version to which you want to upgrade. Upgrade files are available on the [Delphix download site](#). The procedure for uploading an upgrade version to the Delphix Engine is:

1. Download an upgrade version from the Delphix download site to a directory that is visible from the host running the web browser.
2. Login to the Server Setup application.
3. In the System Upgrade Management panel, click View.
4. Click the up arrow to upload a new version.
5. A file dialog will popup. Select the upgrade version you downloaded from the download site.

Once the file has been uploaded to the Delphix Engine, it will be unpacked in the background and ultimately displayed in the list of versions on the left-hand side of the System Upgrade Management screen.

**Scheduling VDB Downtime**

If a new version of the operating system is included in the new Delphix version, then your Delphix Engine will automatically disable all VDBs and dSources during the upgrade process in order to safely reboot to the new version. This will only happen if a new version of the OS is being installed. To determine if an upgrade will result in a reboot and VDB downtime, compare the OS version in the currently-running Delphix version with the OS version in the newly-uploaded Delphix version to which you will be upgrading. The OS version is included in the version details displayed in the System Setup application's System Upgrade Management screen.

If the OS will not be updated as part of the upgrade, then the upgrade process will have no impact on the availability of VDBs, and you do not need to schedule any downtime for your VDB applications.

If the OS will be updated as part of the upgrade, then you should schedule appropriate downtime for your VDB applications. The Delphix Engine will automatically disable VDBs and dSources during upgrade. The length of downtime will be proportional to the number of VDBs.

> Long running jobs including replication and snapsync will fail during any upgrade.

**Upgrade Verification**
The Delphix Engine provides a feature that allows you to verify, or validate, an upgrade before applying it. The verification does a dry run of some of the upgrade procedures in order to alert the administrator of potential problems before continuing with the upgrade. It is strongly recommended that you perform this verification a day or two in advance before your upgrade downtime begins in order to give yourself time to address any problems flagged by the verification.

The procedure for verifying an upgrade is:

1. Login to the Server Setup application.
2. In the System Upgrade Management panel, click View.
3. On the left-hand side, select the version to which you will be upgrading. Details on the version will be displayed on the right.
4. Below the version details, click Verify Upgrade.

Verification will be run in the background. You can view the progress of the verification in the Action sidebar.

**Upgrade Procedure**

Once you have uploaded an upgrade version, optionally verified the upgrade, and optionally scheduled downtime for VDBs, you can apply the upgrade.

1. Login to the Server Setup application.
2. In the System Upgrade Management panel, click View.
3. On the left-hand side, select the version to which you will be upgrading.
4. Click Apply Upgrade to initiate the upgrade process.

The upgrade will run in the background. You can view the progress of the upgrade in the Action sidebar.

**Deferred OS Upgrade**

Each Delphix Engine upgrade image contains both Delphix management software and software for DelphixOS, the operating system that runs Delphix. DelphixOS is versioned, and the OS version that is delivered with any given Delphix Engine version is displayed as `osVersion` in the version properties. By default, when you apply a new version, if that version delivers a newer OS than what is currently running, the system will reboot to the new OS as part of the upgrade process. This requires scheduling downtime for VDBs, because VDBs are disabled during upgrade.

In some cases, it may be possible to defer upgrading DelphixOS even when a new version is included in the upgrade image. You can determine if this is possible by comparing the `minOsVersion` property of the new version with the `osVersion` that is currently running. For example:

```
delphix system version> list
NAME     STATUS             OSRUNNING  BUILDDATE
4.0.6.0  UPLOADED           false      2014-06-17T03:12:48.000Z
4.0.5.0  CURRENTLY_RUNNING  true       2014-06-10T14:41:28.000Z
```

Here, the running OS comes from version 4.0.5.0. You want to see if the OS version in 4.0.5.0 meets the minimum requirements for version 4.0.6.0, to which you are upgrading:
In this example, although 4.0.6.0 includes a newer version of DelphixOS than what is currently running, the currently-running OS meets its minimum OS version requirement. Consequently, you can choose to defer upgrading DelphixOS when upgrading from 4.0.5.0 to 4.0.6.0 by setting the `defer` property in the `apply` context.

In general, you can defer upgrading DelphixOS when the currently-running OS version is greater than or equal to the minimum OS version requirements of the version to which you are upgrading. When you perform a deferred OS upgrade, the OS version will still be installed, but the system will not reboot to that new version. The Delphix Engine will still restart to the new version, but this restart will not result in downtime for VDBs. After that point, the `STATUS` column of the running version will show `DEFERRED` instead of `CURRENTLY_RUNNING`. This indicates that although this version is running, the OS upgrade was deferred.

Later, you can update the OS to the current version by applying the running version again and not setting the `defer` property. When you do this, the system will reboot to the current version of DelphixOS. This will result in downtime for your VDBs.

⚠️ Contact Delphix support to determine whether a deferred OS upgrade is appropriate for your Delphix Engine. You should be aware of what changes are included in the new OS version before making this determination.
Upgrading VM Tools and Hardware

This topic describes considerations for upgrading VM Tools and Hardware.

Every release of Delphix Engine includes updates to VM Tools that are current with the latest version of vSphere. You should not attempt a manual update of VM Tools after upgrading to a new version of vSphere.

If you need to update to a new version of virtual hardware after a vSphere upgrade, you can do so without impact to the Delphix Engine. However, unless you need a specific virtual hardware feature, it is best avoid this update. This is a permanent change that will prevent you from being able to run the Delphix Engine on older vSphere versions.

If you are considering an upgrade to vSphere 8, there are two features relevant to the Delphix virtual machine. vSphere 8 supports 256GB of RAM and 8 CPUs for the virtual machine. If you don’t need this much memory and processor support, or don’t have the appropriate vSphere and Delphix licenses to utilize this amount of memory and processor support, you should avoid an upgrade to vSphere 8.
Shutting Down the Delphix Engine and Using Factory Reset

This topic describes the process for shutting down the Delphix Engine and returning it to "factory default" settings.

Prerequisites

Shut down all VDBs before shutting down the Delphix Engine. Failure to do so can lead to stale NFS mounts on the target environment (for unix environments) or iSCSI I/O errors (SQL Server environments). For the same reason, disable all dSources that use pre-provisioning (all SQL Server dSources, and any Oracle dSources with validated sync enabled).

Procedure

1. Launch the ServerSetup application and log in using the sysadmin or other system administrator credentials.
2. Click Shut Down Server.
3. To return the Delphix Engine to the state of initial configuration, click Factory Reset.

⚠️ Use Factory Reset only when a complete reset and reconfiguration of the Delphix Engine is necessary, as all Delphix Engine objects will be de-allocated.
Managing System Administrators

These topics describe setting up and managing system administrators for the Delphix Engine.

- System Administrators and Delphix Users
- Adding New System Administrators
- Changing System Administrator Passwords
- Deleting and Suspending System Administrators
- Reinstating System Administrators
System Administrators and Delphix Users

This topic describes the different kinds of users in Delphix, including sysadmin and delphix_admin.

System Administrators

Delphix system administrator users are responsible for managing the Delphix Engine itself, but not the objects (Environments, dSources, VDB's) within the server. For example, a system administrator is responsible for setting the time on the Delphix Engine and its network address, restarting it, creating new system administrator users (but not Delphix users), and other similar tasks.

The sysadmin user is the default system administrator user. While this user can be suspended, it may not be deleted.

System administrators administer the Delphix Engine through the ServerSetup interface, which is accessed through a Web browser at http:<Delphix Engine>/ServerSetup.html, as well as through the command line interface accessible via ssh.

Delphix Users

Delphix users are responsible for managing the objects within the Delphix Engine. These include:

- dSources
- VDBs
- Groups
- Policies
- Space and Bandwidth
- Replication Services
- Backup and Restore

A Delphix user can be marked as a Delphix Admin. Delphix Admins have three special privileges:

- They can manage other Delphix users
- They implicitly have Owner privileges for all Delphix objects
- They can create new groups and new environments

The delphix_admin is the default Delphix user provided with a Delphix Engine and is a Delphix Admin. Like the sysadmin user, delphix_admin can not be deleted.

A Delphix Admin user accesses objects with the Delphix Engine Admin Interface, which is accessed through a Web browser at http:<Delphix Engine>/Server.html.

Updating Credentials

System administrator users can change the password of any other system administrator user. Delphix Admin users can change the password of any other Delphix user (including other Delphix Admins). Regular Delphix users can change their own passwords but must provide their old password to do this.
Adding New System Administrators

This topic describes how to add system administrators to the Delphix Engine.

Procedure

1. Launch the ServerSetup application and log in using sysadmin level credentials.
2. In the System User Management panel, click +.
3. Enter the required information.
4. Click Save.
Changing System Administrator Passwords

This topic describes how to change system administrator passwords on the Delphix Engine.

Procedure

1. Launch the ServerSetup application and log in using sysadmin level credentials.
2. In the System User Management panel, click the user whose password you want to change.
3. Select Change Password?
4. Enter the new password in the New Password and Verify New Password fields.
5. Click OK.
Deleting and Suspending System Administrators

This topic describes how delete and suspend system administrators on the Delphix Engine.

Procedure

1. Launch the ServerSetup application and log in using the sysadmin (or other system administrator) credentials.
2. In the System User Management panel, click the user you want to suspend or delete.
3. Suspend the user by clicking the red, crossed circle icon in the lower left corner of the System User Management panel.
4. Delete the user by clicking the trash can icon in the lower left corner of the panel.

Suspending the sysadmin User

The sysadmin user is a required user for the Delphix Engine. This user cannot be deleted, but can be suspended. Suspending the sysadmin user prevents that user from being able to log into ServerSetup or to the console via ssh. The sysadmin user can still log into the console on the install machine, even if the sysadmin account has been suspended.
Reinstating System Administrators

This topic describes how to reinstate system administrators whose accounts have been suspended.

Procedure

1. Launch the **ServerSetup** application and log in using system administrator credentials.
2. In the **System User Management** panel, click on the name of the user you want to reinstate.
3. Reinstate the user by clicking the yellow checkmark icon in the lower left corner of the **System User Management** panel.
Capacity and Resource Management

These topics describe procedures and concepts for capacity and resource management.

- An Overview of Capacity and Performance Information
- Adding and Expanding Storage Devices
- Working with the Capacity Management Graphs in the Graphical User Interface
- Setting Quotas
- Deleting Objects to Increase Capacity
- Changing Snapshot Retention to Increase Capacity
An Overview of Capacity and Performance Information

This topic describes the Delphix Engine performance reservoir and capacity threshold warnings, and various ways to obtain information about capacity and resource usage for the Delphix Server.

The Performance Reservoir and Capacity Threshold Warnings

In order to obtain best performance, the Delphix Engine requires a performance reservoir of 15% of the total quota for storage space. As storage capacity approaches this threshold, the following system faults occur:

- **When 78% of the total storage quota is reached**, a **Warning** fault is triggered. You can resolve this fault by deleting objects in the Delphix Engine, adding storage, or changing policies, as described in the topics [Adding and Expanding Storage Devices](#), [Deleting Objects to Increase Capacity](#), and [Changing Snapshot Retention to Increase Capacity](#).

- **When 85% of the total storage quota is reached**, a **Critical** fault is triggered, and the Delphix Engine will enter into **Maintenance Mode**. When this occurs:
  - All pending link, sync, refresh, and provisioning processes will be cancelled, and no new operations can be initiated
  - Policy operations such as SnapSync, Snapshot, and Refresh are suspended for all platforms
  - dSources stop pulling in new changes. LogSync is suspended for all Oracle and PostgreSQL dSources. Validated sync is disabled for SQL Server dSources.
  - No Virtual Database (VDB) snapshots can be taken

- **When 95% of the total storage quota is reached**, a second **Critical** fault is triggered for SQL Server. All SQL Server VDBs stop in order to maintain data integrity.

To take the system out of Maintenance Mode, increase the storage capacity. This can be done by adding storage devices, deleting objects, or changing policy settings. When you have increased the storage capacity, the system will automatically exit **Maintenance Mode**.

- **When the system falls below 95% of the total storage quota**, you can manually start SQL Server VDBs that had stopped
- **When the system falls below 90% of the total storage quota**, SQL Server VDBs that had stopped will automatically start
- **When the system falls below 82% of the total storage quota**:
  - New link, sync, refresh and provisioning operations are allowed
  - Policy operations such as SnapSync, Snapshot, and Refresh resume for all platforms
  - dSources start pulling in new changes from their corresponding data sources. LogSync is resumed for Oracle and PostgreSQL dSources. Validated sync is enabled for SQL Server dSources.

See [Setting Quotas](#) for more information.

Ways to View Capacity Usage

You can access capacity and performance information for the Delphix Engine through several different means, including the **TimeFlow** view, the **Dashboard** view, and the **Capacity** screen.

The TimeFlow View

You can access the **TimeFlow** view by clicking the **Delphix Logo** in the **Delphix Admin** application, or selecting **Databases > My Databases**.
The TimeFlow view provides three summary performance metrics:

- **VDBS** - all VDBs on the Delphix Engine. Inactive VDBs are greyed out.
- **TimeFlow Ratio** - the total of Delphix Engine storage against the projected usage in a traditional physical database
- **Consolidation Ratio** - the amount of space that dSources and VDBs occupy compared to the amount that would be occupied by a traditional physical database

The Dashboard View

You can access the **Dashboard** view in the Delphix Admin application by clicking **Dashboard** in the top menu bar. Note that the Dashboard view provides only summary information about capacity and performance. You must access the **Capacity** and **Performance** screens in the Resources menu to manage storage space and database objects.

The Dashboard view provides more detailed information about the overall performance of the Delphix Engine and its network in five panels:

- **Job History** - all jobs that have been initiated in the system and their outcome
- **Capacity Management** - the amount of physical storage available and what is currently used
- **TimeFlow Ratio** - see above
- **VDB Ratio** - a measure of the amount of physical space that would be occupied by the database content against the amount of storage occupied by that same database content as VDBs.
- **Performance Management** - the amount of network bandwidth available and the amount that VDBs are currently utilizing, as well as information about specific VDB network usage

The Capacity Screen

You can access the **Capacity** screen through the Resources menu in the Delphix Admin application.

The **Capacity** screen provides a view of storage allocation for dSources, VDBs, and Snapshots by group and objects within the group, as well as a summary metric. You can also access a graph view of the Capacity screen by clicking **Graph View** in the upper right corner of the screen. This view shows **Available Space** as green, **Used Capacity** as blue, and **Reserved Space** as yellow.

<table>
<thead>
<tr>
<th>Grid Column</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the group or database object. Click the expand icon next to a group name to see the objects in that group.</td>
</tr>
<tr>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>Quota</td>
<td>The maximum amount of storage space allocated to the group or object, also known as the ceiling. See Setting Quotas for more information. You can see quota allocations for groups and objects in the Graph view of the Capacity screen.</td>
</tr>
<tr>
<td>Used</td>
<td>Amount of storage space used by the group or object.</td>
</tr>
<tr>
<td>Unvirtualized</td>
<td>Estimated amount of space that the group or object would occupy in an unvirtualized state.</td>
</tr>
<tr>
<td>Ratio</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dSource Ratio</td>
<td>The total amount of storage space occupied by the sources of all dSources as opposed to the amount of storage space occupied by the dSources themselves.</td>
</tr>
<tr>
<td>VDB Ratio</td>
<td>The total amount of storage space occupied by the databases that are the sources for the VDBs as opposed to the amount of storage space occupied by the VDBs.</td>
</tr>
<tr>
<td>TimeFlow Ratio</td>
<td>The total amount of storage space occupied by all snapshots multiplied by their unvirtualized size as opposed to the amount of storage space occupied by the virtualized snapshots, archive logs, and temp files.</td>
</tr>
</tbody>
</table>

### Summary Metric

<table>
<thead>
<tr>
<th></th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep Until</td>
<td>For Snapshots, the number of days it is retained as set by the Snapshot Retention Policy. See the topics under Managing Policies for more information.</td>
</tr>
</tbody>
</table>

### Related Links

- Adding and Expanding Storage Devices
- Changing Snapshot Retention to Increase Capacity
- Deleting Objects to Increase Capacity
- Managing Policies
- Setting Quotas
Adding and Expanding Storage Devices

This topic describes adding and expanding storage devices after initial configuration.

Prerequisites

If you are expanding a storage device after initial configuration, first make sure to add capacity to it using the storage management tools available through the device's operating system. In vSphere, for example, you can add capacity using Edit System Settings.

Procedure

1. Launch the ServerSetup application and log in using the sysadmin credentials.
2. In the Storage section of the Server Setup Summary screen, click Modify.
3. The Delphix Engine should automatically detect any new storage devices. If a newly added storage device does not appear in the Storage section of the Server Setup Summary screen, click Rediscover.
4. Select Expand for each device that you want to expand. The Expand checkbox appears next to the name of devices that have added capacity (in other words, the underlying LUN has been expanded), and the Unused column indicates how much capacity is available for each device. Newly-added devices will have a drop-down in the Usage Assignment column. Set the Usage Assignment to DATA for newly-added devices that you wish to add to the storage pool.
5. Click OK.

⚠️ WARNING: DO NOT REMOVE A CONFIGURED STORAGE DEVICE

Do not remove a configured storage device or reduce its capacity. Removing or reducing a configured storage device will cause a fault with the Delphix Engine, and will require the assistance of Delphix Support for recovery.

Related Links

- Setting Up the Delphix Engine
Working with the Capacity Management Graphs in the Graphical User Interface

This topic describes $description

Section Title

(Optional. Use only if there are two sections. Section headers will be shown in table of contents at the top of topic)

Section Title

Related Links

(Optional. Type "[" and letters from other topic titles to view an auto-complete list of topics)
**Setting Quotas**

This topic describes how to set quotas for database objects.

**Procedure**

1. Log into the Delphix Admin application with `delphix_admin` credentials.
2. Select **Resources > Capacity**.
3. In the **Quotas** column, click next to the **group** or **object** for which you want to set a quota.
4. Enter the amount of storage space you want to allocate for a quota.
5. Click outside the column again to set the amount.

---

**Quotas and Low Space Errors**

Be very careful setting quotas. As a group or virtual database (VDB) approaches the quota level, snapshots may fail and logs may not be captured, causing LogSync to fail. If quotas prevent logs from being written to snapshots, or if a low space condition arises, the Delphix Engine will halt.

When a low space condition arises, the Delphix Engine will generate an **Out of Space** error message. If you do not correct the low space condition, the server will enter **Maintenance Mode**. This will disable all SnapSyncs, LogSyncs, and growth of VDBs, including VDB archive logs. If the situation is severe enough, it may require the assistance of Delphix Support to recover the system.
Deleting Objects to Increase Capacity

This topic describes how to delete database objects to create additional capacity.

Deleting unused or outdated objects should be a regular part of Delphix Engine administration. This is especially important to prevent low space errors, which can cause the Delphix Engine to stop.

Procedure

1. Log into the Delphix Admin application using delphix_admin credentials.
2. Select Resources > Capacity.
3. Select the groups or objects you want to delete.
   As you select items, you will see them added to the Total Capacity of Objects Selected for Deletion.
4. Click Delete.

- Dependencies

If there are dependencies on the SnapShot you will not be able to delete the SnapShot free space; the dependencies rely on the data associated with the SnapShot. If there are dependencies on the SnapShot you will not be able to delete the SnapShot free space; the dependencies rely on the data associated with the SnapShot.
Changing Snapshot Retention to Increase Capacity

This topic describes how to manage capacity by changing the Snapshot discard date on the Capacity screen.

The accumulation of snapshots can have a substantial effect on capacity. For this reason, it is important that you set the snapshot frequency policy to accurately reflect the rate of change in your database, and set snapshot and log retention policies that are appropriate for your desired ability to recover Delphix Engine objects. See the topics under Managing Policies for more information. If you want to discard snapshots outside of a set policy, you can change the number of days they are retained on the Capacity screen.

Procedure

1. Log into the Delphix Admin application using delphix_admin credentials.
2. Select Resources > Capacity.
3. Click in the Keep Until column for the snapshot you want to edit.
4. Select the number of days you want to preserve the snapshot.
5. Click outside the column to set the change.

⚠️ Deleting Snapshots Associated with VDBs

Snapshots that have been used to provision Virtual Databases cannot be deleted.
System Monitoring

These topics describe system monitoring features.

- Viewing Action Status
- System Faults
- Viewing System Events
- Accessing Audit Logs
- Creating Support Logs
- Setting Support Access Control
- Setting SysLog Preferences
- Diagnosing Connectivity Errors
Viewing Action Status

This topic describes how to view the status of actions for the Delphix Engine.

Action Sidebar Overview

The Action sidebar consists of two sections. The top section lists actions that are currently running on the Delphix Engine. The bottom section, labeled Recently completed, contains actions which have recently completed. Each action is initially collapsed and only presents the title of the action. Click an action to expand it and see more details such as progress, elapsed time, and a description of the operation in progress. The following is an example of the Action sidebar when a Link action is running.

If you are a Delphix Admin or a System Admin, you will be able to see all actions of your respective application. If you are not an admin user, you will only see actions you have permissions to see. To view the status of actions that are currently running on the Delphix Engine, open the Action sidebar. To view details of currently-running and completed jobs, open the Dashboard.

Procedure

1. Login to the Delphix Admin or ServerSetup application.
   Depending on the width of the window, the Action sidebar may be automatically displayed on the right of the screen.
2. To see the Action sidebar, click Action on the top navigation bar.

Sub-Action

Each action may contain one or several sub-actions which represent the execution of a subset of the action itself. Click an action to see its sub-actions and their respective details. Note that the list of sub-actions is created dynamically during the execution of the action.

The following is an example of an Environment Refresh action and its sub-actions.
Action Errors

When an error condition occurs during the execution of an action, the background color of the action's box becomes red, and the action remains in the top section until you dismiss it.

1. Click the action title to expand it.
   The action will expand to display a description of the error, suggestions to resolve it, and sometimes the raw output of command execution.

To dismiss the action:

1. Click the X next to the action displaying an error.

The following is an example of an action failure displayed in the Action sidebar.
Procedure to Open the Dashboard

1. Login to the Delphix Admin application using delphix_admin credentials.

2. Click Dashboard in the top menu bar.

   The Jobs panel displays all jobs that have been initiated by the Delphix Engine, and their status.

3. Click Settings to set date range and filter criteria for jobs.

4. Click Search.

5. To view details for a job, click the information icon.
System Faults

This topic describes the purpose and function of system faults.

System Faults Overview

System faults describe states and configurations that may negatively impact the functionality of the Delphix Engine which can only be resolved through active user intervention. When you login to the Delphix Admin application as a delphix_admin, the number of outstanding system faults appears on the right-hand side of the navigation bar at the top of the screen. Faults serve as a record of all issues impacting the Delphix Engine and can never be deleted. However, ignored and resolved faults are not displayed in the faults list.

Viewing Faults

To view the list of active system faults:

1. Click Faults on the right-hand side of the navigation bar.
2. Click any fault in the list to expand it and see its details.

Each fault is comprised of six parts:

- **Severity** – How much of an impact the fault will have on the system. A fault may have a severity of either Warning or Critical. A Warning Fault implies that the system can continue despite the fault but may not perform optimally in all scenarios. A Critical Fault describes an issue that breaks certain functionality and must be resolved before some or all functions of the Delphix Engine can be performed.
- **Date** – The date the fault was diagnosed by the Delphix Engine
- **Target Object** – The object that the fault was posted against
- **Title** – A short descriptive summary of the fault
- **Details** – A detailed summary of the cause of the fault
- **User Action** – The action you can take to resolve the fault

Click on the screenshot below to view the six parts.
Addressing Faults

After viewing a fault and deciding on the appropriate course of action, you can address the fault through the user interface (UI). You can mark a fault as **Ignored** or **Resolved**. If you have fixed the underlying cause of the fault, mark it as **Resolved**. Note that if the fault condition persists, it will be detected in the future and re-diagnosed. You can mark the fault as **Ignored** if it meets the following criteria:

- The fault is caused by a well-understood issue that cannot be changed
- Its impact to the Delphix Engine is well understood and acceptable

In this case, the fault will not be re-diagnosed even if the fault condition persists. You will receive no further notifications.

To address a fault, follow the steps below.

1. In the top menu bar, click **Faults**.
2. In the list of faults, click a **fault date/name** to view the fault details.
3. If the fault condition has been resolved, click **Mark Resolved**.
   Note that if the fault condition persists it will be detected in the future and re-diagnosed.
4. If the fault condition describes a configuration with well-understood impact to the Delphix Engine that cannot be changed, you can ignore the fault by clicking **Ignore**.
   Note that an ignored fault will not be diagnosed again even if the underlying condition persists.

⚠️ When a **critical** fault occurs, the Delphix Engine immediately sends an email to the **delphix_admin**. Make sure you have configured an SMTP server so that this email can be sent. See **Setting Up the Delphix Engine** for more information.

Fault Lifecycle Example

Below is an image of the fault card for the fault "TCP slot table entries below recommended minimum."
The Details section of the fault explains that the sunrpc.tcp_slot_table_entries property on frodo.dcenter.delphix.com is set to a value that is below the recommended minimum of 128. The User Action section instructs you to adjust the value of the sunrpc.tcp_slot_table_entries property upward to the recommended minimum. The process for adjusting this property differs between operating systems. To resolve the underlying issue, search “how to adjust sunrpc.tcp_slot_table_entries” using a search engine and find that the second result is a link to the Delphix community forum describing how to resolve this issue. After following the instructions applicable to your operating system, return to the Delphix UI and mark the fault Resolved.

Related Links

- Setting Up the Delphix Engine
Viewing System Events

This topic describes how to view system event information.

Procedure

1. Launch the Delphix Admin application and log in with delphix_admin credentials.
2. Select System > Event Viewer.
3. Select a time range.
4. Click Search.
Accessing Audit Logs

This topic describes how to access audit logs. The audit log provides a record of all actions that were initiated by a policy or user, regardless of whether that action was successful.

Procedure

1. Log into the Delphix Admin application using delphix_admin credentials.
2. Select System > Audit Logs.
3. Select an audit log time range.
4. Click Search.
Creating Support Logs

This topic describes how to create support bundles and manage server access control for Delphix Support.

Support bundles are used by Delphix Support as diagnostic tools for resolving Delphix Engine issues and routine maintenance. Support bundles can be transferred directly to Delphix Support or downloaded. No customer-specific data is included in the support bundle information, all passwords and personal data are either encrypted or omitted.

Procedure

1. Log into the Delphix Admin application using delphix_admin credentials.
2. Select System > Support Logs
3. Select Transfer or Download.
   If you select Download, then the support bundle will be downloaded as a .tar file. If you select Transfer, then the support bundle will be uploaded over HTTPS to Delphix Support. If you have configured an HTTP proxy, it will be used to send the support bundle.
4. Click OK.

You can also access support log functionality in the ServerSetup application using sysadmin credentials. Click Support Bundles in the top menu bar.
Setting Support Access Control

This topic describes how to set the Support Access Control for Delphix Support.

Support access control enables Delphix Support to access your instance of the Delphix Engine for a defined period of time using an access token.

Procedure

1. Log into the ServerSetup application using sysadmin credentials.
2. Select Server Preferences > Support Access.
3. Click Enable.
4. Set the time period during which you want to allow Delphix Support to have access to your instance of the Delphix Engine.
5. Click Generate Token.
   Provide the token to Delphix Support to enable access to your server.
Setting SysLog Preferences

Syslog is a widely used standard for message logging. It permits the separation of the software that generates messages, the system that stores them, and the software that reports and analyzes them. Delphix makes use of syslog as one of the standard mechanisms, along with SNMP and email, to distribute important user and system events, such as alerts, faults, and audits. In the case of Delphix, each Delphix Engine acts as a syslog client which propagates the events to a centralized syslog server.

The network protocol over which Delphix Engine communicates with the syslog server is standardized in RFC 5424. As a protocol, it supports using either UDP (RFC 5426) or TCP (RFC 6587) as the underlying transport and optional TLS mapping has been introduced to encrypt the messages over the wire for security purposes (RFC 5425). However, as of this release, we only support syslog over UDP with no encryption, which implies that syslog messages are always sent in the clear and may be lost during transmission and delivered out of order due to the limitations of UDP.

To configure for syslog support, you must specify the communication end point to which the syslog server listens, which includes the hostname or IP address of the syslog server and an optional port number. The latter defaults to 514 according to the syslog standard but it can be changed if necessary.

System and user events generated by Delphix are always forwarded immediately to the syslog server, which ensures timely delivery of important events that may require immediate action.

A couple of different output formats are supported for messages delivered over syslog, namely, TEXT and JSON. The TEXT format is the default. To change the message format, as of this release, you must do so via the CLI.

Procedure

1. Log into the ServerSetup application using sysadmin credentials.
2. Select Server Preferences > Syslog Configuration.
3. Select the severity level of the messages you want sent to the SysLog server.
4. Click Add Server.
5. Enter the SysLog server hostname/IP address and port number.
6. Click Add.
7. Click Enable.
Diagnosing Connectivity Errors

Prior to the current release, when the Delphix Engine ran into an error operating on an external database or environment, it reported the immediate error that it had encountered; there was no mechanism for automatic analysis of the root causes of failures. The current release includes infrastructure for automatic diagnosis of errors. When one of these errors occurs, the Delphix Engine now launches a set of tests to locate the root cause of the problem and present the result of the diagnosis. This will help you easily identify the true sources of errors such as closed ports or misconfigured routers.

**Failed Actions**

The Delphix Engine communicates failures in two different manners: actions that fail to complete, and faults. To view failed actions:

1. In the top right-hand corner of the Delphix user interface (UI), click **Actions**.
2. For more information about why the action failed, click the () icon to show the error dialog as seen in the image below.

This shows a popup message with more information about the problem and what actions to take to resolve it. For some errors, the Delphix Engine will be able to diagnose the problem further and display this extra information under **Diagnosing Information**. In the screenshot above, the job failed because the Delphix Engine was unable to lookup the host address.

**Faults**

A fault symbolizes a condition that can affect the performance or functionality of the Delphix Engine and must be addressed. Faults can be either warnings or critical failures that prevent the Delphix Engine from functioning normally. For example, a problem with a source or target environment can cause SnapSync or LogSync policy jobs to fail. Faults will show up as active as long as:

- The error is still occurring, or
- You have chosen to manually resolve it or ignore it

For example, if a background job fails, it will create a fault that describes the problem. To view any active faults:

1. In the top right-hand corner of the Delphix UI, click **Faults**.

This brings up a popup box listing all active faults.
The screenshot above illustrates a fault with regard to database network connectivity. The Delphix Engine will mark an object with a warning triangle to indicate that it is affected by an external problem. You can view more details of the fault by looking at the active faults and their fault effects.
Performance Tuning

These topics describe how to use the performance analytics tool to improve performance of the Delphix Engineer, and topics that describe specific configuration recommendations for hosts, networks, and storage to improve performance.

- **Configuration Options for Improved Performance**
  - **Network Performance Configuration Options**
    - Optimal Network Configuration Parameters for the Delphix Engine
    - Network Operations Using the Delphix Session Protocol
    - Using the Network Performance Tool
    - Interpreting Results from the Network Performance Tool
  - **Storage Performance Configuration Options**
    - Optimal Storage Configuration Parameters for the Delphix Engine
    - Storage Performance Tool
  - **Host Performance Configuration Options**
    - Target Host Configuration Options for Improved Performance

- **Performance Analytics**
  - Performance Analytics Tool Overview
  - Working with Performance Analytics Graphs in the Graphical User Interface
  - Performance Analytics Statistics Reference
  - Performance Analytics Tool API Reference
  - Performance Analytics Case Study: Using a Single Statistic
  - Performance Analytics Case Study: Using Multiple Statistics
Configuration Options for Improved Performance

These topics describe configuration options to maximize the network, storage, database, and host performance for a Delphix Engine installation.

- **Network Performance Configuration Options**
  - Optimal Network Configuration Parameters for the Delphix Engine
  - Network Operations Using the Delphix Session Protocol
  - Using the Network Performance Tool
  - Interpreting Results from the Network Performance Tool

- **Storage Performance Configuration Options**
  - Optimal Storage Configuration Parameters for the Delphix Engine
  - Storage Performance Tool

- **Host Performance Configuration Options**
  - Target Host Configuration Options for Improved Performance
Network Performance Configuration Options

These topics describe configuration options to maximize the network performance of a Delphix Engine deployment.

- Optimal Network Configuration Parameters for the Delphix Engine
- Network Operations Using the Delphix Session Protocol
- Using the Network Performance Tool
- Interpreting Results from the Network Performance Tool
Optimal Network Configuration Parameters for the Delphix Engine

This topic describes basic network performance considerations for the Delphix Engine.

Network Architecture and Latency

All VDB I/O operations are serviced over the network. Delphix uses NFS as the primary transport for Oracle VDBs, and iSCSI+NTFS for MS SQL VDBs. The network architecture, latency, and capacity between the Delphix Engine and the target environment are the key network components for improving performance of a Delphix deployment. Latency between the Delphix Engine and the source environment is not as critical for best performance of VDBs.

For optimal performance of VDBs, round-trip latency between the Delphix Engine and the target environment should be kept under 1 millisecond, and preferably in the range of 300 microseconds. If network latency exceeds 500 microseconds, the VDBs will not perform as well as a database connected to physical storage.

Latency can be introduced by having to route the network packets across multiple networks, or by the presence of routers, switches, and firewalls between the Delphix Engine and the target environment. Best practices to reduce network latency include:

- Keep the Delphix Engine on the same subnet as the target environment
- Reduce the number of hops between the Delphix Engine and the target environment
  - Reduce the number of switches in the network. Each switch can add 50 - 100 microseconds of latency to the network.
  - Reduce the number of routers in the network. Each router can add 500 - 1000 microseconds of latency in a network, and the round trip for an I/O operation could increase by as much as 1 - 2 milliseconds.
- There should be no firewalls between the Delphix Engine and the target environment.
- When linking the Delphix Engine to a source database across a WAN, consider the time needed for the initial link and load. It may be necessary to schedule the load operation as multiple steps across multiple days.

A Common WAN Deployment Architecture
Deployment of the Delphix Engine on Separate Sub-Nets

**Network Throughput and Bandwidth**

Network throughput measures the rate at which data can be sent continuously between two servers on a network. Network throughput is affected by network latency, but the dominant factor affecting throughput is the bandwidth of the network. As a point of comparison, consider the bandwidth available for three types of Ethernet networks:

<table>
<thead>
<tr>
<th>Ethernet Type</th>
<th>Network Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>100Mb Ethernet (100Base-T)</td>
<td>~10MB/sec</td>
</tr>
<tr>
<td>Gigabit Ethernet (GbE)</td>
<td>~100MB/sec</td>
</tr>
<tr>
<td>10 Gigabit Ethernet (10GbE)</td>
<td>~1GB/sec</td>
</tr>
</tbody>
</table>

Low network throughput can impact the Delphix Engine in a number of ways:

- Increasing the amount of time it takes to perform a SnapSync operation, both for initial load and subsequent regular snapshots
- Managing LogSync operations in a high change environment
- Poor VDB performance when an application is performing large sequential I/O operations, such as sequential table scans for reporting or business intelligence, or RMAN backups of the VDB.

Delphix Engine throughput must exceed the sum of the peak I/O loads of all VDBs. Delphix incorporates an I/O-Collector toolkit to collect I/O data from each production source database and pre-production server.

Best practices to improve network throughput include:

- Use 10 Gigabit Ethernet (10GbE)
- Use a dedicated storage network
If you are concerned about your network throughput, you can test it with the open source tool netio. You will need to work with Delphix Support to install netio on the Delphix Engine and run the tests.
Network Operations Using the Delphix Session Protocol

This topic describes how the Delphix Engine uses the Delphix Service Protocol (DSP) for network operations, and how this affects features such as replication, V2P, and SnapSync.

Overview

Delphix Session Protocol, or DSP, is a communication protocol that operates at the session and presentation layer in the Open Systems Interconnection (OSI) model.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer</td>
<td>application specific logic</td>
</tr>
<tr>
<td>Presentation Layer</td>
<td>data encoding, digest, compression, encryption</td>
</tr>
<tr>
<td>Session Layer</td>
<td>connection management, error recovery, security, remote operation</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>end-to-end connection, message segmentation, sequencing, reliability, flow control</td>
</tr>
<tr>
<td>Network Layer</td>
<td>packet fragmentation, routing, logical addressing</td>
</tr>
<tr>
<td>Data Link Layer</td>
<td>physical addressing</td>
</tr>
<tr>
<td>Physical Layer</td>
<td>media, signal, binary transmission</td>
</tr>
</tbody>
</table>

DSP supports the request-reply pattern for communication between two networked peers. In contrast to the traditional remote procedure call (RPC) models, which focus exclusively on low level details such as data encoding and wire framing, DSP implements a generic session layer that supports a number of advanced functionalities desired for network communication, including:

- Full duplex remote operation execution and end-to-end cancellation support
- Advanced connectivity model with connection trunking and ordered delivery
- Fault resilience with automatic connection and session recovery, exactly-once semantics, and optional data digest
- High performance with concurrent execution, session flow control, optional data compression and bandwidth throttling
- Built-in security support with pluggable SASL authentication mechanisms and optional TLS encryption
- Asynchronous model for session management and remote operation

Most of the features above are essential to the proper operation of a distributed application and yet non-trivial to implement. By offering them in the framework, we can significantly simplify the development of enterprise quality distributed applications.

DSP is officially registered with the Internet Assigned Numbers Authority under the service name of dlpx-sp and port number 8415.
Currently, DSP supports the Java language binding and provides a java based service framework for distributed applications.

**Key Concepts**

The foundation of DSP is built on top of a few key abstractions, namely, **exchange**, **task**, **nexus**, and **service**. For an overview of how DSP works and the features it provides, let’s start with these abstractions.

An **exchange** refers to an application defined protocol data unit which may be a request or a response. DSP supports the request-response pattern for communication. For each request sent, there is a corresponding response which describes the result of the execution. An application protocol is made up of a set of exchanges.

A **nexus** (a.k.a., session) refers to a logical conduit between the client and server application. In contrast, a transport connection (a.k.a., connection) refers to a “physical” link. A nexus has a separate naming scheme from the connection, which allows it to be uniquely and persistently identified independent of the physical infrastructure. A nexus has a different lifecycle than the connection. It is first established over a leading connection. After it comes into existence, new connections may be added and existing ones removed. It must have at least one connection to remain operational but may live on even after all connections are lost. Nexus lifecycle management actions, such as create, recover, and destroy, are always initiated by the client with the server remaining passive.

A nexus has dual channels, namely, the fore channel and the back channel. The fore channel is used for requests initiated from the client to the server; and the back channel from the server to the client. From a request execution perspective, the nexus is full duplex and the channels are functionally identical, modulo the operational parameters that may be negotiated independently for each channel. A channel supports a number of features for request processing, such as ordered delivery, concurrent execution, remote cancellation, exactly-once semantics, and throughput throttling.

A **service** refers to a contract that consists of all exchanges (both the requests and the corresponding responses) defined in an application protocol. Given the full duplex nature of request execution in DSP, part of the service is fulfilled by the server and the remaining by the client, where the client and server are from the nexus management perspective.

A **task** implements a workflow that typically involves multiple requests executed in either or both directions over the nexus. A task is a self-contained building block, available in the form of a sharable module including both the protocol exchanges and implementation, that can be easily integrated into other application protocols. A library of tasks may significantly simplify distributed application development by making it more of an assembly experience.

The following is a diagram that illustrates the key abstractions and how they are related to each other.
Security

As a network protocol, DSP is designed with security in mind from the onset. It supports strong authentication as well as data encryption. It follows a session based authentication model which requires each connection to authenticate before it is allowed to join the session. Authentication is performed using the Simple Authentication and Security Layer (SASL) framework, a standard based pluggable security framework. The currently supported SASL mechanisms include DIGEST-MD5, PLAIN with TLS, CRAM, and ANONYMOUS. Optionally, TLS encryption may be negotiated between the client and the server for data privacy.

Performance

DSP offers a number of features to enable the support for high performance network applications. For example, it allows multiple requests to be exchanged in both directions simultaneously, which provides effective pipelining of data transfer to minimize the impact of network latency while ensuring the total ordering at the same time. It supports trunking that can effectively aggregate the throughput across multiple connections, which is crucial for long fat network (LFN) and 10GigE. It also provides optional compression support which boosts performance over bandwidth limited network. We have observed, through both internal benchmarking and in customer environment, DSP based applications delivering multi GigE in an ideal environment and getting a performance boost of as much as x10 in bandwidth limited settings.

Resiliency

DSP automatically recovers from transient connection loss without any application involvement. It may also detect random data corruption on the wire and automatically recovers from it. In both cases, outstanding requests are retried once the fault condition is resolved.

DSP offers control over a remotely executing request. Once a request is initiated, the application may cancel it at any time before completion. In the rare event of a session loss, a new session creation request will be held until the old session has been reinstated. It ensures that we never leave any unknown or unwanted activities on the remote side and provides better predictability and consistency guarantees over an otherwise unreliable network.

Diagnosability

Application exceptions encountered during remote execution of a request are communicated back to the initiator through DSP. A standard Java API is used to facilitate the handling of remote exceptions that is in many ways
identical to local ones.

DSP provides detailed information and statistics at the session level. The information may be used to examine the state of the session as well as diagnose performance problems. It is currently exposed via an internal support tool called jmxtool.

**Supported Applications**

Replication is the first feature to take advantage of DSP. It has been rebuilt on top of DSP and shipping in the field since 3.1. In the latest release, a number of host based applications, such as SnapSync, V2P, and Delphix connector, use DSP as well.
Using the Network Performance Tool

This topic describes how to use the network performance tool to measure network performance between the Delphix Engine and environment hosts.

**Prerequisites**

The network performance tool measures network performance between a Delphix Engine and an environment host. You must have added an environment in order to use this tool. At this time, this tool only supports Unix environments.

⚠️ The transmission control protocol (TCP) throughput test uses TCP port 50001 by default. The port can also be configured on a per-test-run basis. For the duration of a given throughput test, a server on the receiver will be listening on this port. For a transmit test, the receiver is the remote host; for a receive test, the receiver is the Delphix Engine.

### Measuring Round-Trip Latency

The network latency test measures network round-trip latency by transmitting ICMP echo requests (like the ping utility) and measuring the time to receive replies from the remote host. To execute a test:

1. Login as a domain user to the Delphix Engine CLI using ssh.
2. Create a test.

   ```
   delphix> network test latency
   delphix network test latency> create
   delphix network test latency create *> get
   type: NetworkLatencyTestExecuteParameters
   remoteHost: (required)
   requestCount: 20
   requestSize: 8B
   ```

3. Set test parameters and commit to execute the test. You must set the `remoteHost` property to the name of an environment host. The `requestCount` parameter sets the number of requests transmitted, at a rate of one per second. The `requestSize` sets the size, in bytes, of each request.

   ```
   delphix network test latency create *> set remoteHost=oracletarget
   delphix network test latency create *> commit
   Dispatched job JOB-20
   NETWORK_LATENCY_TEST_EXECUTE job started for "oracletarget-2014-06-20T18:57:28.6592".
   Executing network latency test.
   ```

4. View results. All times are in microseconds.
Measuring Throughput

The network throughput test measures sustained throughput using a synthetic workload to or from a remote host. To execute a test:

1. Login as a domain user to the Delphix Engine CLI using ssh.
2. Create a test.

```bash
delphix> network test throughput
delphix network test throughput> create
delphix network test throughput create *> get
type: NetworkThroughputTestParameters
blockSize: 128KB
direction: TRANSMIT
duration: 30
numConnections: 0
port: 50001
receiveSocketBuffer: 4MB
remoteHost: (required)
sendSocketBuffer: 4MB
```

3. Set test parameters and commit to execute the test. You must set the remoteHost property to the name of an environment host. The remaining arguments are optional, and are:
   - **direction**: Determines whether the test is a transmit or receive test from the perspective of the Delphix Engine. The default is transmit.
   - **duration**: Determines the duration of the test in seconds. The default is 30 seconds.
   - **numConnections**: Determines the number of TCP connections to use concurrently during the test. By default, the test automatically calculates the optimal number of connections by running mock 5-second tests with varying numbers of connections.
3. delphix network test throughput create *> set remoteHost=oraclesource
   delphix network test throughput create *> commit
   Dispatched job JOB-21
   NETWORK_THROUGHPUT_TEST_EXECUTE job started for
   "oraclesource-2014-06-20T19:30:12.566Z".
   Executing network throughput transmit test.
   Measuring throughput with variable number of connections: 1.
   Measuring throughput with variable number of connections: 2.
   Measuring throughput with variable number of connections: 4.
   Measuring throughput with variable number of connections: 6.
   Measuring throughput with variable number of connections: 8.
   Measuring maximum sustained throughput for 30 seconds with 8 connections.
   NETWORK_THROUGHPUT_TEST_EXECUTE job for

4. View results.

   delphix network test throughput> list
   NAME                            DIRECTION  THROUGHPUT
   oraclesource-2014-06-20T19:30:12.566Z  TRANSMIT   695.6Mbps
   delphix network test throughput> select oraclesource-2014-06-20T19:30:12.566Z
   delphix network test throughput "oraclesource-2014-06-20T19:30:12.566Z"> get
   type: NetworkThroughputTest
   name: oraclesource-2014-06-20T19:30:12.566Z
   numConnections: 8
   parameters:
   type: NetworkThroughputTestParameters
   blockSize: 128KB
   direction: TRANSMIT
   duration: 30
   numConnections: 0
   port: 50001
   receiveSocketBuffer: 4MB
   remoteHost: oraclesource
   sendSocketBuffer: 4MB
   reference: NETWORK_THROUGHPUT_TEST-2
   remoteAddress: 172.16.203.184
   state: COMPLETED
   throughput: 695.6Mbps
Interpreting Results from the Network Performance Tool

This topic describes how to interpret results from the network performance tool for the Delphix Engine.

Section Title

(Optional. Use only if there are two sections. Section headers will be shown in table of contents at the top of topic)

Section Title

Related Links

(Optional. Type"[" and letters from other topic titles to view an auto-complete list of topics)
Storage Performance Configuration Options

These topics describe options for maximizing the storage performance of a Delphix Engine deployment.

- [Optimal Storage Configuration Parameters for the Delphix Engine](#)
- [Storage Performance Tool](#)
Optimal Storage Configuration Parameters for the Delphix Engine

This topic describes minimum capacity and throughput requirements for storage devices used with the Delphix Engine.

Storage for the Delphix Engine must be able to sustain the aggregated Input/Output Operations Per Second (IOPS) and throughput (MBPS) requirements of all its Virtual Databases. Throughput required for data source synchronization (SnapSync and LogSync) must also be supported.

The Delphix Engine requires storage for:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A copy of each Source Database</td>
<td>The copies are compressed.</td>
</tr>
<tr>
<td>Unique Block Changes in VDBs</td>
<td>When changes are made to a VDB, the Delphix Engine stores the changes in new blocks associated with only that VDB. The new blocks are compressed.</td>
</tr>
<tr>
<td>Timeflow for dSources and VDBs</td>
<td>The TimeFlow kept for each dSource and VDB comprises snapshots of the database (blocks changed since the previous snapshot) and archive logs. The retention period for this history of changes is determined by polices established for each dSource and VDB. The TimeFlow is compressed.</td>
</tr>
</tbody>
</table>

In addition to the storage for these items, the Delphix Engine requires 30% free space in its storage for best performance. See An Overview of Capacity and Performance Information and related topics for more details on managing capacity for the Delphix Engine.

Best practices for storage performance include:

- Initial storage equal to the size of the physical source databases. For high redo rates and/or high DB change rates, allocate an additional 10-20% storage.
- Add storage when storage capacity approaches 30% free
- Use physical LUNS allocated from storage pools or RAID groups that are configured for availability
- Never share physical LUNs between the Delphix Engine and other storage clients.
- Keep all physical LUNs the same size. Add new storage by provisioning new LUNs of the same size.
- Provision storage using VMDKs or RDMs operating in virtual compatibility mode.
- VMDKs should be Thick Provisioned, Eager Zeroed. The underlying physical LUNs can be thin provisioned.
- Physical LUNs used for RDMs should be thick provisioned.
- Measure or estimate the required IOPS and manage the storage disks to provide this capacity. It is common to use larger numbers of spindles to provide the IOPS required.
- Physical LUNs carved from RAID 1+0 groups or pools with dedicated spindles provide higher IOPS performance than other configurations
- Maximize Delphix Engine vRAM for a larger system cache to service reads

Example

There are two production dSources, totalling 5 TB in size. 5 VDBs will be created for each. Sum of read and write rates on the production source database is moderate (1000 iops), sum of VDB read rate is moderate (950 iops), and VDB update rate is low (50 iops).
• Initial storage equal to 5TB, provisioned as 5 x 1 TB physical LUNs, Thin Provisioned. Allow for expansion of the LUNs to 2TB.
• Provision as 5 x 950 GB Virtual Disks. VMDKs must be Thick Provisioned, Eager Zeroed. Using 1 TB LUNs allows expansion to 2 TB (ESX 5.1 limit).
• The storage provisioned to the Delphix Engine storage must be able to sustain 1000 IOPs (950 + 50). For this reason, each physical LUN provisioned to the Delphix Engine must be capable of sustaining 200 IOPs. IOPs on the source databases are not relevant to the Delphix Engine.
• 64GB Delphix Engine vRAM for a large system cache

Related Topics
• Optimal Network Configuration Parameters for the Delphix Engine
• An Overview of Capacity and Performance Information
Storage Performance Tool

Overview

The Storage Performance Tool executes a synthetic workload on the data storage to evaluate the performance characteristics of the storage devices that are available to the Delphix Engine. The Storage Performance Tool is a new feature that is only available from the command line interface (CLI).

Understanding the Storage Performance Tool

The Storage Performance Tool is typically run prior to setting up the Delphix Engine, for evaluating the performance characteristics of the storage devices. It is also run after setup when you add new storage devices to the Delphix Engine. You can only run the Storage Performance Tool on storage devices that have not been configured. For example, disks are configured by adding them to ZFS pools. Once the disks are handed over to ZFS, the Storage Performance Tool will refuse to touch them because of the data that now exists on the devices.

Getting Started

Prior to setting up the Delphix Engine, the admin can login to the Delphix CLI using the `sysadmin` account to launch the Storage Performance Tool.

Prerequisites

Storage devices that have been configured and that are in use by the Delphix Engine cannot be used for performance evaluation.

Storage Test Parameters

The Storage Performance Tool takes the following parameters:

- **Devices** – List of storage devices to evaluate. An empty list (unset) results in all of the unused storage devices being used for performance evaluation. The default is an empty list.
- **Duration** – The duration, in seconds, that each workload is run. The default value is 120 seconds.
- **TestRegion** – The data region, spanning all the devices that have been specified, to be used for performance evaluation. The default value is 128 GB.
- **InitializeDevices** – Setting this parameter initializes the devices by writing a pattern on to them prior to running the test workloads. This is on by default.
- **InitializeEntireDevice** – Setting this parameter results in the entire device being initialized, rather than just the TestRegion. This is off by default.
- **Tests** – This parameter specifies a subset of the workloads to be run. It can take the following values:
  - **ALL** – Run all of the workloads. This is the default.
  - **MINIMAL** – Run a minimal set of workloads needed to generate the storage grades
  - **RANDREAD** – Run only the random read workloads
  - **READ** – Run only the sequential read workloads
  - **WRITE** – Run only the write workloads

Storage Test Procedure in CLI

1. Login as a domain user to the Delphix Engine CLI using ssh. Prior to setup, login as `sysadmin`.

   ```
   bash-4.2$ ssh sysadmin@delphixvm
   Password:
   Last login: Mon Jan 5 19:36:17 2015 from 172.16.102.82
   Delphix Engine Network Setup
   ```
To access the system setup through the browser, the system must first be configured for networking in your environment. After this, you can configure the primary interface, DNS, hostname, and default route. When DHCP is configured, all other properties are derived from DHCP settings.

To see the current settings, run `get`. To change a property, run `set <property>=<value>`. To commit your changes, run `commit`. To exit this setup and return to the standard CLI, run `discard`.

For detailed help on each property, run `help properties`.

Current settings:

- defaultRoute: 10.43.0.1
- dhcp: true
- dnsDomain: delphix.com
- dnsServers: 172.16.101.11
- hostname: delphix
- primaryAddress: 10.43.10.233/16

2. If the Delphix Engine has not been setup yet, the network setup prompt appears. Discard the command.

```
delphix network setup update *> discard
```

delphix>

3. Create a storage test.

```
delphix> storage test
```
```
delphix storage test> create
```
```
delphix storage test create *>
```

4. Modify the test parameters as needed and commit the test to start the storage test job.

```
delphix storage test create *> get
```
```
  type: StorageTestParameters
  devices: (unset)
  duration: 120
  initializeDevices: true
  initializeEntireDevice: false
  testRegion: 128GB
  tests: ALL
```
```
delphix storage test create *> commit
```
STORAGE_TEST-1

Dispatched job JOB-1

STORAGE_TEST_EXECUTE job started for "SYSTEM".

Initializing storage test.

Starting storage device initialization.

ETA: 1:28:44.

Storage device initialization complete.

Starting storage benchmarking.

Starting random read workload with 4 KB block size and 8 jobs.
Starting random read workload with 4 KB block size and 16 jobs.
Starting random read workload with 4 KB block size and 32 jobs.
Starting random read workload with 4 KB block size and 64 jobs.
Starting random read workload with 8 KB block size and 8 jobs.
Starting random read workload with 8 KB block size and 16 jobs.
Starting random read workload with 8 KB block size and 32 jobs.
Starting random read workload with 8 KB block size and 64 jobs.
Starting sequential write workload with 1 KB block size and 4 jobs.
Starting sequential write workload with 4 KB block size and 4 jobs.
Starting sequential write workload with 8 KB block size and 4 jobs.
Starting sequential write workload with 16 KB block size and 4 jobs.
Starting sequential write workload with 32 KB block size and 4 jobs.
Starting sequential write workload with 64 KB block size and 4 jobs.
Starting sequential write workload with 128 KB block size and 4 jobs.
Starting sequential write workload with 1024 KB block size and 4 jobs.
Starting sequential write workload with 1 KB block size and 16 jobs.
Starting sequential write workload with 4 KB block size and 16 jobs.
Starting sequential write workload with 8 KB block size and 16 jobs.
Starting sequential write workload with 16 KB block size and 16 jobs.
Starting sequential write workload with 32 KB block size and 16 jobs.
Starting sequential write workload with 64 KB block size and 16 jobs.
Starting sequential write workload with 128 KB block size and 16 jobs.
Starting sequential write workload with 1024 KB block size and 16 jobs.
Starting sequential read workload with 64 KB block size and 4 jobs.
Starting sequential read workload with 64 KB block size and 8 jobs.
Starting sequential read workload with 64 KB block size and 16 jobs.
Starting sequential read workload with 64 KB block size and 32 jobs.
Starting sequential read workload with 64 KB block size and 64 jobs.
Starting sequential read workload with 128 KB block size and 4 jobs.
Starting sequential read workload with 128 KB block size and 8 jobs.
Starting sequential read workload with 128 KB block size and 16 jobs.
Starting sequential read workload with 128 KB block size and 32 jobs.
Starting sequential read workload with 128 KB block size and 64 jobs.
Starting sequential read workload with 1024 KB block size and 4 jobs.
Starting sequential read workload with 1024 KB block size and 8 jobs.
Starting sequential read workload with 1024 KB block size and 16 jobs.
Starting sequential read workload with 1024 KB block size and 32 jobs.
Starting sequential read workload with 1024 KB block size and 64 jobs.

Storage benchmarking complete.
Generating results.
Storage test completed successfully.

STORAGE_TEST_EXECUTE job for "SYSTEM" completed successfully.

delphix storage test>

5. Retrieve the result of the storage performance test.

delphix storage test> select STORAGE_TEST-1

delphix storage test 'STORAGE_TEST-1'> result
Test Results

----------

Test ID: 1

Test System UUID: 564dc710-7bb1-c064-12c2-2659032acf1b

Start Time: 03-Feb-2015 10:52:31 -0800
End Time: 03-Feb-2015 12:20:25 -0800

Test Grades:

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Latency</th>
<th>Load Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>95th %ile</td>
</tr>
<tr>
<td>Random 8K Reads w/ 16 jobs</td>
<td>2.16</td>
<td>4.77</td>
</tr>
<tr>
<td>Random 4K Reads w/ 16 jobs</td>
<td>1.62</td>
<td>3.73</td>
</tr>
<tr>
<td>Sequential 1M Reads w/ 4 jobs</td>
<td>62.60</td>
<td>182.00</td>
</tr>
<tr>
<td>Sequential 1K Writes w/ 4 jobs</td>
<td>1.30</td>
<td>2.61</td>
</tr>
<tr>
<td>Sequential 128K Writes w/ 4 jobs</td>
<td>10.19</td>
<td>26.00</td>
</tr>
</tbody>
</table>

IO Histogram:

<table>
<thead>
<tr>
<th>Test Name</th>
<th>us50</th>
<th>us100</th>
<th>us250</th>
<th>us500</th>
<th>ms1</th>
<th>ms2</th>
<th>ms4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>us10</td>
<td>ms20</td>
<td>ms50</td>
<td>ms100</td>
<td>ms250</td>
<td>ms500</td>
<td>s1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s5</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>Rand 4K Reads w/ 8 Jobs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>46</td>
<td>41</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Rand 4K Reads w/ 16 Jobs</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rand 4K Reads w/ 32 Jobs</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rand 4K Reads w/ 64 Jobs</td>
<td>16</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rand 8K Reads w/ 8 Jobs</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rand 8K Reads w/ 16 Jobs</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rand 8K Reads w/ 32 Jobs</td>
<td>18</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rand 8K Reads w/ 64 Jobs</td>
<td>85</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 1K Writes w/ 4 Jobs</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 4K Writes w/ 4 Jobs</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 8K Writes w/ 4 Jobs</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 16K Writes w/ 4 Jobs</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 32K Writes w/ 4 Jobs</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 64K Writes w/ 4 Jobs</td>
<td>33</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 128K Writes w/ 4 Jobs</td>
<td>76</td>
<td>16</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 1M Writes w/ 4 Jobs</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>57</td>
<td>14</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Seq 1K Writes w/ 16 Jobs</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 4K Writes w/ 16 Jobs</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 8K Writes w/ 16 Jobs</td>
<td>12</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sequence Size</td>
<td>Jobs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>16K Writes</td>
<td>16</td>
<td>23</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>32K Writes</td>
<td>16</td>
<td>74</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64K Writes</td>
<td>16</td>
<td>2</td>
<td>81</td>
<td>12</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>128K Writes</td>
<td>16</td>
<td>1</td>
<td>2</td>
<td>85</td>
<td>7</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>1M Writes</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>45</td>
<td>38</td>
</tr>
<tr>
<td>64K Reads</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64K Reads</td>
<td>8</td>
<td>74</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64K Reads</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64K Reads</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64K Reads</td>
<td>64</td>
<td>14</td>
<td>53</td>
<td>27</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>128K Reads</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>75</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>128K Reads</td>
<td>8</td>
<td>64</td>
<td>29</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>128K Reads</td>
<td>16</td>
<td>1</td>
<td>45</td>
<td>45</td>
<td>6</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>128K Reads</td>
<td>32</td>
<td>0</td>
<td>1</td>
<td>65</td>
<td>24</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>128K Reads</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>54</td>
<td>29</td>
</tr>
<tr>
<td>1M Reads</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1M Reads</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>33</td>
<td>52</td>
<td>11</td>
</tr>
<tr>
<td>1M Reads</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Seq 1M Reads w/ 32 Jobs  0  0  0  0  0  0  0  0  0  0  1  4  19  58  11  6  1
Seq 1M Reads w/ 64 Jobs  0  0  0  0  0  0  0  0  0  0  1  2  10  40  32  11  2

Grading Key:

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Grade:</th>
<th>A+</th>
<th>A</th>
<th>A-</th>
<th>B</th>
<th>B-</th>
<th>C</th>
<th>C-</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Random Reads</td>
<td>2.0</td>
<td>4.0</td>
<td>6.0</td>
<td>8.0</td>
<td>10.0</td>
<td>12.0</td>
<td>14.0</td>
<td>&gt; 14.0</td>
<td></td>
</tr>
<tr>
<td>Large Seq Reads</td>
<td>12.0</td>
<td>14.0</td>
<td>16.0</td>
<td>18.0</td>
<td>20.0</td>
<td>22.0</td>
<td>24.0</td>
<td>&gt; 24.0</td>
<td></td>
</tr>
<tr>
<td>Small Seq Writes</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>2.0</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>&gt; 3.5</td>
<td></td>
</tr>
<tr>
<td>Large Seq Writes</td>
<td>2.0</td>
<td>4.0</td>
<td>6.0</td>
<td>8.0</td>
<td>10.0</td>
<td>12.0</td>
<td>14.0</td>
<td>&gt; 14.0</td>
<td></td>
</tr>
</tbody>
</table>

delphix storage test 'STORAGE_TEST-1'
Host Performance Configuration Options

These topics describe configuration options to maximize host performance in a Delphix Engine deployment.

- **Target Host Configuration Options for Improved Performance**
Target Host Configuration Options for Improved Performance

This topic describes configuration options to maximize the performance of a target host in a Delphix Engine deployment.

**OS-Specific Tuning Recommendations for Oracle Databases**

**Solaris**

- When exclusively using Oracle's Direct NFS Feature (dNFS), it is unnecessary to tune the native NFS client. However, tuning network parameters is still relevant and may improve performance.

**Tuning the Kernel NFS Client**

- On systems using Oracle Solaris Zones, the kernel NFS client can only be tuned from the global zone.

On Solaris, by default the maximum I/O size used for NFS read or write requests is 32K. When Oracle does I/O larger than 32K, the I/O is broken down into smaller requests that are serialized. This may result in poor I/O performance. To increase the maximum I/O size:

1. As superuser, add to the `/etc/system` file:

   ```
   * For Delphix: change the maximum NFS block size to 1M
   set nfs:nfs3_bsize=0x100000
   ```

2. Run this command:

   ```
   # echo "nfs3_bsize/W 100000" | mdb -kw
   ```

   It is critical that the above command be executed exactly as shown, with quotations and space. Errors in the command may cause a system panic and reboot.

**Tuning TCP Buffer Sizes**

- On systems using Oracle Solaris Zones, TCP parameters, including buffer sizes, can only be tuned from the global zone or in exclusive-IP non-global zones. Shared-IP non-global zones always inherit TCP parameters from the global zone.

**Solaris 10**

It is necessary to install a new Service Management Facility (SMF) service that will tune TCP parameters after every boot. These are samples of the files needed to create the service:

<table>
<thead>
<tr>
<th>File</th>
<th>Installation location</th>
</tr>
</thead>
<tbody>
<tr>
<td>dlpx-tcptune</td>
<td>/lib/svc/method/dlpx-tcptune</td>
</tr>
<tr>
<td>dlpx-tune.xml</td>
<td>/var/svc/manifest/site/dlpx-tune.xml</td>
</tr>
</tbody>
</table>
1. As superuser, download the files and install in the path listed in the **Installation location** in the table.

2. Run the commands:

   ```
   # chmod 755 /lib/svc/method/dlpx-tcptune
   # /usr/sbin/svccfg validate /var/svc/manifest/site/dlpx-tune.xml
   # /usr/sbin/svccfg import /var/svc/manifest/site/dlpx-tune.xml
   # /usr/sbin/svcadm enable site/tcptune
   ```

Verify that the SMF service ran after being enabled by running the command:

```
# cat `svcprop -p restarter/logfile tcptune`
```

You should see output similar to this:

```
[ May 14 20:02:02 Executing start method ("/lib/svc/method/dlpx-tcptune start"). ]
Tuning TCP Network Parameters
tcp_max_buf adjusted from 1048576 to 4194304
tcp_cwnd_max adjusted from 1048576 to 4194304
tcp_xmit_hiwat adjusted from 49152 to 4194304
tcp_recv_hiwat adjusted from 128000 to 4194304
[ May 14 20:02:02 Method "start" exited with status 0. ]
```

**Solaris 11**

**As superuser**

Run the following commands:

```
# ipadm set-prop -p max_buf=4194304 tcp
# ipadm set-prop -p _cwnd_max=4194304 tcp
# ipadm set-prop -p send_buf=4194304 tcp
# ipadm set-prop -p recv_buf=4194304 tcp
```

**Linux/Redhat/CentOs**

*Tuning the Kernel NFS Client*

In Linux, the number of simultaneous NFS requests is limited by the Remote Procedure Call (RPC) subsystem. The maximum number of simultaneous requests defaults to 16. Maximize the number of simultaneous requests by changing the kernel tunable `sunrpc.tcp_slot_table_entries` value to 128.

**RHEL4 through RHEL5.6**

1. As superuser, run the following command to change the instantaneous value of simultaneous RPC
commands:

```
# sysctl -w sunrpc.tcp_slot_table_entries=128
```

2. Edit the file `/etc/modprobe.d/modprobe.conf.dist` and change the line:

```
install sunrpc /sbin/modprobe --first-time --ignore-install sunrpc && {
/bin/mount -t rpc_pipefs sunrpc /var/lib/nfs/rpc_pipefs > /dev/null 2>&1 || ;;
```

to

```
install sunrpc /sbin/modprobe --first-time --ignore-install sunrpc && {
/bin/mount -t rpc_pipefs sunrpc /var/lib/nfs/rpc_pipefs > /dev/null 2>&1;
/sbin/sysctl -w sunrpc.tcp_slot_table_entries=128;
```

- Improper changes to the modprobe.conf.dist file may disrupt use of NFS on the system. Check with your system administrator or operating system vendor for assistance. Save a copy of the modprobe.conf.dist in a directory other than `/etc/modprobe.d` before starting.

**RHEL 5.7 through RHEL 6.2**

1. As superuser, run the following command to change the instantaneous value of simultaneous RPC commands:

```
# sysctl -w sunrpc.tcp_slot_table_entries=128
```

2. If it doesn't already exist, create the file `/etc/modprobe.d/rpcinfo` with the following contents:

```
options sunrpc tcp_slot_table_entries=128
```

**RHEL 6.3 onwards**

Beginning with RHEL 6.3, the number of RPC slots is dynamically managed by the system and does not need to be tuned. Although the `sunrpc.tcp_slot_table_entries` tuneable still exists, it has a default value of 2, instead of 16 as in prior releases. The maximum number of simultaneous requests is determined by the new tuneable, `sunrpc.tcp_max_slot_table_entries`, which has a default value of 65535.

**Tuning TCP Buffer Sizes**

1. As superuser, add or replace the following entries in `/etc/sysctl.conf`.
   
   Note: the `*rmem`, `*wmem` parameter values are minimum recommendations, so no change is needed if already set to higher values.
On AIX, by default the maximum I/O size used for NFS read or write requests is 64K. When Oracle does I/O larger than 64K, the I/O is broken down into smaller requests that are serialized. This may result in poor I/O performance. IBM can provide an Authorized Program Analysis Report (APAR) that allows the I/O size to be configured to a larger value.

1. Determine the appropriate APAR for the version of AIX you are using:

<table>
<thead>
<tr>
<th>AIX Version</th>
<th>APAR Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>IV24594</td>
</tr>
<tr>
<td>7.1</td>
<td>IV24688</td>
</tr>
</tbody>
</table>

2. Check if the required APAR is already installed by running this command:

```
# /usr/sbin/instfix -ik IV24594
```

If the APAR is installed, you will see a message similar to this:

```
All filesets for IV24594 were found.
```

If the APAR is not yet installed, you will see a message similar to this:

```
There was no data for IV24594 in the fix database.
```

3. Download and install the APAR, as necessary. To find the APARs, use the main search function at [http://www.ibm.com/us/en/](http://www.ibm.com/us/en/), specifying the name of the APAR you are looking for from step 1.
4. Configure the maximum read and write sizes using the commands below:

```
# nfso -p -o nfs_max_read_size=524288
# nfso -p -o nfs_max_write_size=524288
```

5. Confirm the correct settings using the command:

```
# nfso -L nfs_max_read_size -L nfs_max_write_size
```

You should see an output similar to this:

```
NAME CUR DEF BOOT MIN MAX UNIT TYPE
DEPENDENCIES
-----------------------------------------------
nfs_max_read_size 512K 64K 512K 512 512K Bytes D
--------------------------------------------------------------------------------
nfs_max_write_size 512K 64K 512K 512 512K Bytes D
--------------------------------------------------------------------------------
```

**HP-UX**

**Tuning the Kernel NFS Client**

On HP-UX, by default the maximum I/O size used for NFS read or write requests is 32K. When Oracle does I/O larger than 32K, the I/O is broken down into smaller requests that are serialized. This may result in poor I/O performance.

1. As superuser, run the following command:

```
# /usr/sbin/kctune nfs3_bsize=1048576
```

2. Confirm the changes have occurred and are persistent by running the following command and checking the output:

```
# grep nfs3 /stand/system
  tunable nfs3_bsize 1048576
```

**Tuning TCP Buffer Sizes**

1. As superuser, edit the `/etc/rc.config.d/nddconf` file, adding or replacing the following entries:
TRANSPORT_NAME[0]=tcp
NDD_NAME[0]=tcp_recv_hiwater_def
NDD_VALUE[0]=4194304
#
TRANSPORT_NAME[1]=tcp
NDD_NAME[1]=tcp_xmit_hiwater_def
NDD_VALUE[1]=4194304

⚠️ In this example, the array indices are shown as 0 and 1. In the actual configuration file, each index used must be strictly increasing, with no missing entries. See the comments at the beginning of /etc/rc.config.d/nddconf for more information.

2. Run the command:

    /usr/bin/ndd -c

3. Confirm the settings:

    # ndd -get /dev/tcp tcp_recv_hiwater_def
    4194304
    # ndd -get /dev/tcp tcp_xmit_hiwater_def
    4194304

---

**OS-Specific Tuning Recommendations for SQL Server Databases**

These are our recommendations for Windows iSCSI initiator configuration. Please note that the parameters below will affect all applications running on the Windows target host, so you should make sure that the following recommendations do not contradict best practices for other applications running on the host.

For targets running Windows Server, the iSCSI initiator driver timers can be found at: HKLM\SYSTEM\CurrentControlSet\Control\Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\<Instance Number>\<Parameters>. Please see **How to Modify the Windows Registry** on the Microsoft Support site for details about configuring registry settings.

<table>
<thead>
<tr>
<th>Registry Value</th>
<th>Type</th>
<th>Default</th>
<th>Recommended</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxTransferLength</td>
<td>REG_DWORD</td>
<td>262144</td>
<td>131072</td>
<td>This controls the maximum data size of an I/O request. A value of 128K is optimal for the Delphix Engine as it reduces segmentation of the packets as they go through the stack.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------</td>
<td>----------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MaxBurstLength</td>
<td>REG_DWORD</td>
<td>262144</td>
<td>131072</td>
<td>This is the negotiated maximum burst length. 128K is the optimal size for the Delphix Engine.</td>
</tr>
<tr>
<td>MaxPendingRequests</td>
<td>REG_DWORD</td>
<td>255</td>
<td>512</td>
<td>This setting controls the maximum number of outstanding requests allowed by the initiator. At most this many requests will be sent to the target before receiving response for any of the requests.</td>
</tr>
<tr>
<td>MaxRecvDataSegmentLength</td>
<td>REG_DWORD</td>
<td>65536</td>
<td>131072</td>
<td>This is the negotiated MaxRecvDataSegmentLength.</td>
</tr>
</tbody>
</table>

**Related Links**
- [SQL Server Target Host iSCSI Configuration Parameter Recommendations](#)
- [Set Up a SQL Server Target Environment](#)
Performance Analytics

These topics describe how to use the Performance Analytics tool to optimize the performance of a Delphix Engine deployment.

- Performance Analytics Tool Overview
- Working with Performance Analytics Graphs in the Graphical User Interface
- Performance Analytics Statistics Reference
- Performance Analytics Tool API Reference
- Performance Analytics Case Study: Using a Single Statistic
- Performance Analytics Case Study: Using Multiple Statistics
Performance Analytics Tool Overview

This topic describes the Performance Analytics tool and illustrates some basic uses of it.

Introduction

The performance analytics tool allows introspection into how the Delphix Engine is performing. The introspection techniques it provides are tuned to allow an iterative investigation process, helping to narrow down the cause associated with the performance being measured. Performance analytics information can be accessed through the Delphix Admin application, as described in Working with Performance Analytics Graphs in the Graphical User Interface, as well as the CLI and the web services API, as described in other topics in this section. The default statistics that are being collected on the Delphix Engine include CPU utilization, network utilization, and disk, NFS, and iSCSI IO operations (see Performance Analytics Statistics Reference for details).

The performance tool operates with two central concepts: statistics and statistic slices.

Statistics

Each statistic describes some data that can be collected from the Delphix Engine. The first piece of information a statistic provides is its type, which you will use as a handle when creating a statistic slice. It also gives the minimum collection interval, which puts an upper bound on the frequency of data collection. The actual data a statistic can collect is described through a set of axes, each of which describe one “dimension” of that statistic. For example, the statistic associated with Network File System (NFS) operations has a latency axis, as well as an operation type axis (among many others), which allows users to see NFS latencies split by whether they were reads or writes.

Each axis has some important information embedded in it.

- The name of the axis provides a short description of what the axis collects and is used when creating a statistic slice
- A value type, which tells you what kind of data will be collected for this axis. The different value types are integer, boolean, string, and histogram. The first three are straightforward, but statistic axes with a histogram type can collect a distribution of all the values encountered during each collection interval. This means that instead of seeing an average NFS operation latency every collection interval, you can see a full distribution of operation latencies during that interval. This allows you to see outliers as well as the average, and observe the effects of caching on the performance of your system more easily.
- A constraint type, which is only relevant while creating a statistic slice, and will be described in more detail below

One last bit of information that an axis provides makes the most sense after seeing how datapoints are queried. In the most basic situation, you would only collect one axis of a statistic, such as the latency axis from the NFS operations statistic. When you ask for data, you would get back a datapoint for every collection interval in the time range you requested. These datapoints would be grouped into a single stream.
However, if you had collected the operation type axis as well as the latency axis, you would get two streams of datapoints: one for read operations, and one for write operations.

Because the operation axis applies to many datapoints, the datapoints returned are split into two streams, and the operation axis is stored with the top-level stream instead of with each datapoint in the streams. However, the latency axis will be different for each datapoint in a stream, so it is not an attribute of the stream, but instead an attribute of the datapoint.

**Statistic Slices**

Statistics describe what data can be collected and are auto-populated by the system, but statistic slices are responsible for actually collecting the data, and you must create them manually when you want to collect some performance data. Each slice is an instantiation of exactly one statistic, and can only gather data which is described by that statistic. “Slices” are so named because each one provides a subset of the information available from the parent statistic it is associated with. A statistic can be thought of as describing the axes of a multidimensional space, whereas you typically will only want to collect a simpler slice of that space due to the large number of axes available.
When you specify a slice, there are several fields which you must supply:

- The statistic type this slice is associated with. This must be the same type as the statistic this is an instantiation of.
- The collection interval, which must be greater than the minimum collection interval the parent statistic gives
- The axes of the parent statistic this slice will collect

Finally, a slice can place constraints on axes of its parent statistic, allowing you to limit the data you get back. For instance, if you're trying to narrow down the cause of some high NFS latency outliers, it may be useful to filter out any NFS latencies which are shorter than one second. To do this, you would place a constraint on the latency axis of an NFS operation slice that states that the values must be higher than one second. You can constrain any axis in the same fashion, and each axis' description in the parent statistic gives a constraint type which can be applied to it. This allows you to place different types of constraints on the latency axis (which is a number measured in nanoseconds) than the operation type axis (which is an enum that can take the values "read" or "write").

**Persisting Analytics Data**

Data collected by slices is persisted temporarily on the Delphix Engine. Performance data is guaranteed to be available at the finest resolution for 6 hours, then is compressed to per-minute data and held for 7 days, and finally compressed to per-hour data and held for 30 days. If data of a certain resolution will be needed longer than these limits, you should instruct the slice to remember the data permanently until you are done using it. The commands to manage this are listed in the [Performance Analytics Tool API Reference](#).

**Related Links**

- The [Performance Analytics Tool API Reference](#) provides a detailed list of all statistics which can be collected, what their axes represent, and how those axes can be constrained, and outlines all management operations which are available.
- [Performance Analytics Case Study: Using a Single Statistic](#) and [Performance Analytics Case Study: Using Multiple Statistics](#) show two investigations and the commands used to conduct each one.
- [Working with Performance Analytics Graphs in the Graphical User Interface](#)
Working with Performance Analytics Graphs in the Graphical User Interface

This topic describes the performance analytics graphs that are available in the Delphix Engine graphical user interface, and the controls for changing the views of those graphs.

Accessing the Performance Analytics Graphs

1. Log into the Delphix Admin application using delphix_admin credentials.
2. In the Resources menu, select Performance Analytics.
3. Use the controls described below to view statistics and their related graphs.

General Graph Display and Controls

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Control Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Selector</td>
<td>Specifies which graphs are displayed.</td>
<td>View a graph by selecting the checkbox next to its name. To hide a graph, clear the checkbox.</td>
</tr>
<tr>
<td>Zoom Level</td>
<td>Controls the time range of data displayed in the graph. Available values are 1 minute, 1 hour, and 1 day. By default 1 minute is selected.</td>
<td>Select Minute, Hour, or Day to change the Zoom Level.</td>
</tr>
<tr>
<td>Shown Data Timeline</td>
<td>Displays timestamps of data points in the graph.</td>
<td></td>
</tr>
<tr>
<td>Available Data Timeline</td>
<td>Displays navigable time ranges for historical data.</td>
<td></td>
</tr>
</tbody>
</table>
| **Timeline Selector** | Specifies the start and end time for the currently displayed data. The range displayed is controlled by the **Zoom Level**. | Drag the **Timeline Selector** to view statistics for a specific time in the past, or click the scroll bar arrows to view the desired time period. You can also use the slider controls within the Timeline Selector to change the length of time for which data is displayed.

When the Timeline Selector is aligned to the right of the timeline, it represents live data that is updated every second. If the Timeline Selector is moved from right alignment with the timeline, the data displayed is historical and no live updates are displayed. To resume live data updates, move the Timeline Selector back to the right-aligned position representing the current time. The data will be refreshed to the latest data, and live updates will resume every second. |
| **Graph Legend** | If more than one set of information is presented on the graph, the Graph Legend displays a description and color for each set and allows a user to toggle that set off and on.

For example, in the network graph there can be multiple network interfaces, and for each network interface the graph displays four statistics (bytes sent, bytes received, packets sent, packets received). When a user toggles off a specific network interface, all four statistics corresponding to that interface are hidden from the screen. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tip</strong></td>
<td>The color for lines representing bytes sent and packets sent is the same. Similarly, the color for lines representing bytes received and packets received is the same. This makes it easier to correlate the number of bytes and number of packets sent/received for a given network interface.</td>
</tr>
<tr>
<td><strong>To hide a set of information, click on the set name within the Graph Legend. Data representing that set is removed from the graph, and the set's name is greyed out. To show a set that has been hidden, click on the set name.</strong></td>
<td></td>
</tr>
<tr>
<td>Control Name</td>
<td>Control Description</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Timeline Page Left/Right Button</strong></td>
<td>Scrolls Available Data Timeline by a specified time range depending on the current Zoom Level.</td>
</tr>
<tr>
<td><strong>Graph Value Tooltip</strong></td>
<td>Shows a value, along with the time stamp, for a specific data point.</td>
</tr>
<tr>
<td><strong>Latency Range Selector</strong> (shown on latency heatmaps only)</td>
<td>Controls the lower and upper limits for displayed latency buckets.</td>
</tr>
<tr>
<td><strong>Latency Outlier Selector</strong> (shown on latency heatmaps only)</td>
<td>Hides infrequent latencies (outliers) based on a percentage threshold. Its range is 0%-10%, with the default of 0%. The percentage establishes a threshold below which buckets are considered &quot;outliers&quot; and are hidden from the graph. Each bucket is assigned a percentage based on the ratio of its count vs the maximum count of any bucket in the graph.</td>
</tr>
</tbody>
</table>

Related Links

Performance Analytics Statistics Reference
Performance Analytics Statistics Reference

This topic describes the various performance statistics that are available for the Delphix Engine and how they can be used to analyze and improve performance.

The Delphix Engine is shipped with a default set of statistics that are collected on Delphix Engine virtual appliance, as listed below. The statistics are stored for up to 30 days for historical analysis.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Utilization</td>
<td>Total CPU utilization for all CPUs. This statistic includes both kernel and user time.</td>
</tr>
<tr>
<td>Network Throughput</td>
<td>Measures throughput in bytes and packets, broken down by sent vs. received data and by network interface. Each network interface shows four graphed lines: bytes sent, bytes received, packets sent, and packets received. To help easily correlate bytes and packets, the same color is used for both bytes and packet values.</td>
</tr>
<tr>
<td>Disk IO</td>
<td>Measures a number of IO operations, and the latencies and throughput of the underlying storage layer. The statistic is represented by the graphs - a column chart for IO operations, a heat map for latency distribution, and a line chart for throughput. IO operations are grouped by reads and writes. A shaded rectangle on a latency heat map represents an IO operation (read or write) which falls within a particular time range (bucket). The shading of rectangles depends on the number of IO operations that fall within a particular bucket - the higher the count the darker the shading.</td>
</tr>
<tr>
<td>NFS</td>
<td>Measures a number of IO operations and the latencies and throughput of the NFS server layer in the Delphix Engine. Its graphical representation is similar to the Disk IO graph. It is useful to diagnose performance of dSources and VDBs that use NFS mounts (Oracle, PostgreSQL).</td>
</tr>
<tr>
<td>iSCSI</td>
<td>Measures the number of IO operations, and the latencies and throughput, of the iSCSI server layer in the Delphix Engine. Its graphical representation is similar to the Disk IO graph. It is useful to diagnose performance of Microsoft SQL Server dSources and VDBs.</td>
</tr>
</tbody>
</table>

Related Links
- Working with Performance Analytics Graphs in the Graphical User Interface
- Performance Analytics Tool Overview
Performance Analytics Tool API Reference

This topic describes basic commands and command syntax for using the Performance Analytics tool.

Statistic Types

More documentation can be found about each statistic type through the CLI and webservices API, but the following table provides more information about how similar I/O stack statistic types relate to each other.

<table>
<thead>
<tr>
<th>Statistic Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS_OPS</td>
<td>Provides information about Network File System operations. This is the entrypoint to the Delphix Engine for all Oracle database file accesses.</td>
</tr>
<tr>
<td>iSCSI_OPS</td>
<td>Provides information about iSCSI operations. This is the entrypoint to the Delphix Engine for all SQL Server file accesses.</td>
</tr>
<tr>
<td>VFS_OPS</td>
<td>This layer sits immediately below NFS_OPS and iSCSI_OPS, and should give almost exactly the same latencies, assuming no unexpected behavior is occurring.</td>
</tr>
<tr>
<td>DxFS_OPS</td>
<td>This layer sits immediately below VFS_OPS, and the two of them should give almost exactly the same latencies.</td>
</tr>
<tr>
<td>DxFS_IO_QUEUE_OPS</td>
<td>This layer sits below DxFS_OPS, but the latencies will differ from that layer because this layer batches together operations to increase throughput.</td>
</tr>
<tr>
<td>DISK_OPS</td>
<td>This layer sits below DxFS_IO_QUEUE_OPS at the bottom of the I/O stack, and measures interactions the Delphix Engine has with disks.</td>
</tr>
<tr>
<td>CPU_UTIL</td>
<td>This is unrelated to the layers of the I/O stack - it measures CPU utilization on the Delphix Engine.</td>
</tr>
</tbody>
</table>

Statistic Axis Information

Values are returned when a slice's data is queried. Each axis has a value type, which specifies how the data will be returned.

<table>
<thead>
<tr>
<th>Value Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>The value is returned as an integer. For information about what units the integer is measured in, read the documentation for the related datapoint or datapoint stream type.</td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>The value is returned as a boolean.</td>
</tr>
</tbody>
</table>
The value is returned as a string. This is used for enum values as well, although the set of strings which can be returned is limited.

The value is returned as a log-scale histogram. The histogram has size buckets whose minimum and maximum value get doubled. Histograms are returned as JSON maps, where the keys are the minimum value in a bucket and the values are the height of each bucket.

Here is an example histogram. Notice that buckets with a height of zero are not included in the JSON object, and that keys and values are represented as strings.

```json
{
    "32768": "10",
    "65536": "102",
    "262144": "15",
    "524288": "2"
}
```

Axis constraints are used to limit the data which a slice can collect. Each axis specifies a constraint type which can be used to limit that axis' values.

<table>
<thead>
<tr>
<th>Constraint Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BooleanConstraint</td>
<td>A superclass which constraints on boolean values must extend. Currently, the only subclass is BooleanEqualConstraint, which requires that a boolean axis equal either true or false (depending on user input).</td>
</tr>
<tr>
<td>EnumConstraint</td>
<td>A superclass which constraints on enum values must extend. Currently, the only subclass is EnumEqualConstraint, which requires that an enum axis be equal to a user-specified value.</td>
</tr>
<tr>
<td>IntegerConstraint</td>
<td>A superclass which constraints on integer values must extend. Subclasses include IntegerLessThanConstraint, IntegerGreaterThanConstraint, and IntegerEqualConstraint, which map to the obvious comparators for integers.</td>
</tr>
<tr>
<td>NullConstraint</td>
<td>This class signifies that an axis cannot be constrained. This makes the most sense for axes which provide an average value - placing a constraint on an average doesn’t make sense because you are not able to include or discard a particular operation based on what its effects would be on the average of all operations.</td>
</tr>
</tbody>
</table>
PathConstraint

A superclass which constraints on file path values must extend. Currently, the only subclass is `PathDescendantConstraint`, which requires that a path value must be a descendant of the specified path (it must be contained within it). This only applies to paths on the Delphix Engine itself, and all paths used must be canonical Unix paths starting from the root of the filesystem.

StringConstraint

A superclass which constraints on string values must extend. Currently, the only subclass is `StringEqualsConstraint`, which requires that a string value must equal a user-specified string.

Statistic Slice Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description and Usage Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>getData</td>
<td>This is used to fetch data from a statistic slice which has been collecting data for a while. It returns a datapoint set, which is composed of datapoint streams, which contain datapoints. For a full description, see the Performance Analytics Tool Overview.</td>
</tr>
<tr>
<td>rememberRange</td>
<td>This is used to ensure that data collected during an ongoing investigation doesn't get deleted unexpectedly. If this is not used, data is only guaranteed to be persisted for 24 hours. If it is used, data will be remembered until a corresponding call to <code>stopRememberingRange</code> is made.</td>
</tr>
<tr>
<td>stopRememberingRange</td>
<td>This is used to allow previously-remembered data to be forgotten. The data will be forgotten on the same schedule as brand new data, so you will have at least 24 hours before data which you have stopped remembering is deleted. This undoes the rememberRange operation.</td>
</tr>
<tr>
<td>pause</td>
<td>This command pauses the collection of a statistic slice, causing no data to be collected until <code>resume</code> is called.</td>
</tr>
<tr>
<td>resume</td>
<td>This command resumes the collection of a statistic slice, undoing a <code>pause</code> operation.</td>
</tr>
</tbody>
</table>

Related Links

- The Performance Analytics Tool Overview gives an overview of how all the pieces on this page interact.
- The case studies (Performance Analytics Case Study: Using a Single Statistic, Performance Analytics Case Study: Using Multiple Statistics) give command-by-command examples with extensive explanation.
Performance Analytics Case Study: Using a Single Statistic

This topic describes how to perform a sample performance investigation with one statistic from the Performance Analytics tool.

Introduction

The Delphix Engine uses Network File System (NFS) as the transport for Oracle installations. An increase in the NFS latency could be causing sluggishness in your applications running on top of Virtual Databases. This case study illustrates how this pathology can be root caused using the analytics infrastructure. This performance investigation uses one statistic to debug the issue, and utilizes the many axes of that statistic to filter down the probably cause of the issue. This technique uses an approach of iteratively drilling down by inspecting new axes of a single statistic, and filtering the data to only include information about the operations that appear slow. This technique is valuable for determining which use patterns of a resource might be causing the system to be sluggish. If you isolate a performance issue using this approach, but aren’t sure what is causing it or how to fix it, Delphix Support can provide assistance for your investigation.

The following example inspects the statistic which provides information about NFS I/O operations on the Delphix Engine. This statistic can be collected a maximum of once every second, and the axes it can collect, among others, are:

- **latency**, a histogram of wait times between NFS requests and NFS responses
- **size**, a histogram of the NFS I/O sizes requested
- **op**, whether the NFS requests were reads or writes
- **client**, the network address of the NFS client which was making requests

Roughly the same performance information can be obtained from the iSCSI interface as well.

Investigation

1. Begin the performance investigation by examining some high-level statistic such as **latency**.

   a. Create a slice with statistic type **NFS_OPS**.
   b. Set the slice to collect the **latency** axis.
   c. Do not add any constraints.
   d. Set the collection interval.

   Anything over one second will work, but ten seconds gives good data resolution and will not use a lot of storage to persist the data that is collected. The rest of this example will assume a collection period of ten seconds for all other slices, but any value could be used.

   ```
   /analytics
   create
   set name=step1
   set statisticType=NFS_OPS
   set collectionInterval=10
   set collectionAxes=latency
   commit
   ```

   This will collect a time-series of histograms describing NFS latencies as measured from inside the Delphix Engine, where each histogram shows how many NFS I/O operations fell into each latency bucket during every ten-second interval. After a short period of time, read the data from the statistic
slice:

```sql
select step1
getData
setopt format=json
commit
setopt format=text
```

The `setopt` steps are optional but allow you to see the output better via the CLI. The output looks like this:

```json
{
  "type": "DatapointSet",
  "collectionEvents": [],
  "datapointStreams": [{
    "type": "NfsOpsDatapointStream",
    "datapoints": [{
      "type": "IoOpsDatapoint",
      "latency": {
        "32768": "16",
        "65536": "10"
      },
      "timestamp": "2013-05-14T15:51:40.000Z"
    }, ...]
  }, ...
  "resolution": 10
}
```

The data is returned as a set of datapoint streams. Streams hold the fields which are shared by all the datapoints they contain. Later on in this example, the `opt` and `client` fields will be added to the streams, and multiple streams will be returned. Streams are described in more detail in Performance Analytics Tool Overview. The `resolution` field indicates the number of seconds that corresponds to each datapoint, which in our case matches the requested `collectionInterval`. The `collectionEvents` field is not used in this example, but lists when the slice was paused and resumed, to distinguish between moments when no data was collected because the slice was paused, and moments when there was no data to collect.

2. If the latency distributions show some slow NFS operations, the next step would be to determine whether the slow operations are reads or writes.
   a. Specify a new `NFS_OPS` slice to collect this by collecting the `op` and `latency` axes.
   b. To limit output to the long-running operations, create a constraint on the `latency` axis that prohibits the collection of data on operations with latency less than 100ms.
The `greaterThan` field is 100ms converted into nanoseconds.

Reading the data proceeds in the same way as the first step, but there will be two streams of datapoints, one where `op=write`, and one where `op=read`.

3. After inspecting the two data streams, you might find that almost all slow operations are writes, so it could be valuable to determine which clients are requesting the slow writes, and how large each of the writes is.

   a. To collect this data, create a new `NFS_OPS` slice which collects the `size` and `client` axes.

   b. Add constraints ensuring that the `op` axis should be constrained to only collect data for `write` operations, and the `latency` axis should be constrained to filter operations taking less than 100ms.

   Because the constraint on the `op` axis dictates that it will always have the value `write`, it is not necessary to collect the `op` axis anymore.
/analytics
create
set name=step3
set statisticType=NFS_OPS
set collectionInterval=10
set collectionAxes=size,client

edit axisConstraints.0
set type=IntegerGreaterThanConstraint
set axisName=latency
set greaterThan=10000000
back

edit axisConstraints.1
set type=StringEqualConstraint
set axisName=op
set equals=write
back

commit

Reading the data proceeds in the same way as the first two steps, but there will be one stream for every NFS client. The dataset collected by this will consist of a set of streams, one corresponding to each NFS client, and each stream will be a time-series of histograms showing write sizes that occurred during each ten second interval.

Continuing to use this approach will allow you to narrow down the slow writes to a particular NFS client, and you may be able to tune that client in some way to speed it up.

Related Links
- The Performance Analytics Tool API Reference gives a full list of the commands, axes, and data types used by the analytics tool.
Performance Analytics Case Study: Using Multiple Statistics

This topic describes how to perform a sample performance investigation with multiple statistics from the Performance Analytics tool.

- Introduction
- Investigation
  - Setup
  - Analysis
- Related Links

Introduction

This case study illustrates an investigation involving more than one metric. In typical performance investigations you will need to peel out multiple layers of the stack in order to observe the component causing the actual performance pathology. This case study specifically examines sluggish application performance caused due to slow IO responses from the disk sub-system. This example will demonstrate a technique of looking at the performance of each layer in the I/O stack to find which layer is responsible for the most latency, then looking for constrained resources that the layer might need to access. This technique is valuable for finding the most-constrained resource in the system, potentially giving actionable information about resources that can be expanded to increase performance.

For the following example, we will inspect latency at two layers: the Network File System (NFS) layer on the Delphix Engine, and the disk layer below it. Both of these layers provide the latency axis, which gives a histogram of wait times for the clients of each layer.

Investigation

The analytics infrastructure enables users to observe the latency of multiple layers of the software stack. This investigation will examine the latency of both layers, and then draw conclusions about the differences between the two.

Setup

To measure this data, create two slices. When attempting to correlate data between two different statistics, it can be easier to determine causation when collecting data at a relatively high frequency. The fastest that each of these statistics will collect data is once per second, so that is value used.

1. A slice collecting the latency axis for the statistic type NFS_OPS.

    ```bash
    /analytics
    create
    set name=slice1
    set statisticType=NFS_OPS
    set collectionInterval=1
    set collectionAxes=latency
    commit
    ```

2. A slice collecting the latency axis for the statistic type DISK_OPS.
After a short period of time, read the data from the first statistic slice.

```bash
select slice2
getData
setopt format=json
commit
setopt format=text
```

The same process works for the second slice. The `setopt` steps are optional but allow you to see the output better via the CLI. The output for the first slice might look like this:
For the second slice, it might look like this:

```json
{
   "type": "DatapointSet",
   "collectionEvents": [],
   "datapointStreams": [{
      "type": "DiskOpsDatapointStream",
      "datapoints": [{
         "type": "IoOpsDatapoint",
         "latency": {
            "262144": "1",
            "524288": "11",
            "1048576": "13",
            "2097152": "34",
            "4194304": "7",
            "8388608": "2",
            "16777216": "3",
            "33554432": "1"
         },
         "timestamp": "2013-05-14T15:51:40.000Z"
      },
      "type": "IoOpsDatapoint",
      "latency": {
            "262144": "5",
            "524288": "10",
            "1048576": "14",
            "2097152": "26",
            "4194304": "7",
            "8388608": "4",
            "16777216": "2"
         },
         "timestamp": "2013-05-14T15:51:41.000Z"
      }]}
}
```

The data is returned as a set of datapoint streams. Streams hold the fields that would otherwise be shared by all the datapoints they contain, but only one is used in this example because there are no such fields. Streams are discussed in more detail in the Performance Analytics Tool Overview. The resolution field indicates how many seconds each datapoint corresponds to, which in our case matches the requested collectionInterval. The collectionEvents field is not used in this example, but lists when the slice was paused and resumed to distinguish between moments when no data was collected because the slice was paused, and moments when there was no data to collect.

Graphically, these four histograms across two seconds look like this:
Analysis

Because the NFS layer sits above the disk layer, all NFS operations that use the disk synchronously (synchronous writes and uncached reads) will have latencies which are slightly higher than those of their corresponding disk operations. Usually, because disks have very high seek times compared to the time the NFS server spends on CPU, disk operations are responsible for almost all of the latency of these NFS operations. In the graphical representation, you can see this by looking at how the slower cluster of NFS latencies (around 2ms-8ms) have similar latencies to the median of the disk I/O (around 2ms-4ms). Another discrepancy between the two plots is that the number of disk operations is much lower than the corresponding number of NFS operations. This is because the Delphix filesystem batches together write operations to improve performance.

If database performance is not satisfactory and almost all of the NFS operation time is spent waiting for the disks, it suggests that the disk is the slowest piece of the I/O stack. In this case, disk resources (the number of IOPS to the disks, the free space on the disks, and the disk throughput) should be investigated more thoroughly to determine if adding more capacity or a faster disk would improve performance. However, care must be taken when arriving at these conclusions, as a shortage of memory or a recently-rebooted machine can also cause the disk to be used more heavily due to fewer cache hits.

Sometimes, disk operations will not make up all of the latency, which suggests that something between the NFS
server and the disk (namely, something in the Delphix Engine) is taking a long time to complete its work. If this is the case, it is valuable to check whether the Delphix Engine is resource-constrained, and the most common areas of constraint internal to the Delphix Engine are CPU and memory. If either of those is too limited, you should investigate whether expanding the resource would improve performance. If no resources appear to be constrained or more investigation is necessary to convince you that adding resources would help the issue, Delphix Support is available to help debug these issues.

While using this technique, you should take care to recognize the limitations that caching places on how performance data can be interpreted. In this example, the Delphix Engine uses a caching layer for the data it stores, so asynchronous NFS writes will not go to disk quickly because they are being queued into larger batches, and cached NFS reads won't use the disk at all. This causes these types of NFS operations to return much more quickly than any disk operations are able to, resulting in a very large number of low-latency NFS operations in the graph above. For this reason, caching typically creates a bimodal distribution in the NFS latency histograms, where the first cluster of latencies is associated with operations that only hit the cache, and the second cluster of latencies is associated with fully or partially uncached operations. In this case, cached NFS operations should not be compared to the disk latencies because they are unrelated. It is possible to use techniques described in the first example to filter out some of the unrelated operations to allow a more accurate mapping between disk and NFS latencies.

**Related Links**

- The [Performance Analytics Tool API Reference](#) gives a full list of the commands, axes, and data types used by the analytics tool.
Security

These topics describe an overview of security features, and their configuration, for the Delphix Engine.

- Configuring a Boot Password
- Configuring a Security Banner
Configuring a Boot Password

This topic describes how to configure a boot password on the Delphix Engine. The Delphix Engine uses a boot loader to select the underlying system image to run, along with associated options. Access to the console is controlled through the virtualization framework, which should be sufficient to secure the Delphix Engine. System users can configure a boot password if additional compliance regulations require a boot loader password on top of the access control provided by the virtualization framework.

Procedure

1. Log into the CLI as a system user.

   The boot password is not currently configurable through the browser UI.

2. Switch to the service security context and execute the update command.

   delphix> service security
   delphix service security> update

3. Set the bootPassword property. The password can be entered on a separate line to avoid echoing the contents.

   delphix service security update *> set bootPassword
   Enter bootPassword: *****

   To clear an existing boot password, run unset bootPassword.

4. Commit the change.

   delphix service security update *> commit
Configuring a Security Banner

This topic describes how the system administrator can configure a security banner on the Delphix Engine. All users will see the security banner prior to login, over Secure Shell (SSH) and in the browser.

Procedure

1. Log into the command line interface (CLI) as a system user.

   Currently, you cannot configure the security banner through the browser user interface (UI).

2. Switch to the `service security` context and execute the `update` command.

   delphix> service security
   delphix service security> update

3. Set the `banner` property, using quotation marks if you require spaces.

   delphix service security update *> set banner="Use is subject to license terms."

   The banner is in plain text. HTML or other markup is not supported.

   To clear an existing banner, run `unset banner`.

4. Commit the change.

   delphix service security update *> commit
Users, Permissions, and Policies

These topics describe concepts and tasks related to the Delphix Domain and working with Groups.

- **Users and Groups**
  - [Users, Groups, and Permissions: An Overview](#)
  - [User Privileges for Delphix Objects](#)
  - [Adding Delphix Users and Privileges](#)
  - [Editing, Deleting, and Suspending Delphix Users](#)
  - [Assigning Group and Object Ownership](#)
  - [Adding and Deleting Groups](#)
  - [Adding Delphix Admin Users](#)
  - [Managing Individual Profile Information](#)

- **Managing Policies**
  - [Managing Policies: An Overview](#)
  - [Creating Custom Policies](#)
  - [Creating Policy Templates](#)
  - [Policies and Time Zones](#)
  - [Configuring Retention on Individual Snapshots](#)
Users and Groups

These topics describe how to work with users and groups to assign permissions for objects in the Delphix Engine.

- Users, Groups, and Permissions: An Overview
- User Privileges for Delphix Objects
- Adding Delphix Users and Privileges
- Editing, Deleting, and Suspending Delphix Users
- Assigning Group and Object Ownership
- Adding and Deleting Groups
- Adding Delphix Admin Users
- Managing Individual Profile Information
Users, Groups, and Permissions: An Overview

This topic describes the basic concepts behind Groups and the Delphix Domain, and information that can be found on the Group card.

Groups and the Delphix Domain

As described in the topics under Managing System Administrators, a Delphix Engine installation is based on two levels of object ownership. The top level is the Delphix Domain. The Delphix Domain is the top level container of all objects in the Delphix Engine installation, and is administered by users with Delphix Admin credentials.

Beneath the Delphix Domain are Groups. A Delphix Admin user creates groups, but can assign owner or auditor privileges over groups and objects within them. All dSources and VDBs are created within a group. The topic User Privileges for Delphix Objects describes the exact object privileges associated with the Owner and Auditor roles within a group.

When you first install the Delphix Engine, it automatically creates a first group, <New Group>. You can edit the name of this group to begin creating your own groups.

The Use of Groups

Creating groups helps you manage policies and privileges over objects within that group. Policies can be created as policy templates, and when applied at the group level, they extend over all objects within that group. See Creating Policy Templates for more information. Group privileges function in the same way. When object privileges are created for users at the group level, those privileges apply to all objects of that type within the group. When new objects are created or added to the group, the policies and privileges you have created at the group level will be applied to them.

Databases are added to groups as part of the dSource creation process. See Linking an Oracle Data Source for more information.

The Group Card

You can find information about a group, such as its allocation, total storage used, and total storage free, on the group card. When you log into the Delphix Admin application, all groups are listed in the Databases panel. Click the Expand icon next to a group to view its information. You can also access the list of databases by selecting My Databases from the Databases menu.

Video

This short video illustrates the Delphix Object Model.

Related Links

- User Roles in the Delphix Domain
- Linking an Oracle Data Source
- Creating Policy Templates
User Privileges for Delphix Objects

This topic describes the user privileges for Delphix objects.

Owner and Auditor Privileges

The user privileges on Delphix objects consist of two types, Auditor and Owner, which the Delphix Admin user assigns. These privileges apply both to objects, such as dSources and Virtual Databases (VDBs), and to groups, which are containers that hold those objects.

The Delphix Admin user can assign privileges to groups, dSources and VDBs. Privileges are inherited, meaning that privileges assigned to a group are effective for the dSources and VDBs contained in that group.

If a user does not have a privilege in relation to an object or group, then he or she has no visibility into that object or group.

All commands are limited by the privilege level of the user who is executing them. For example, a user with Auditor privileges cannot execute the Virtual to Physical command.

⚠️ Owner Rights for Target Group

In order to provision a VDB to a target group, you must also have owner privileges for that group.

Roles and Privileges for Delphix Objects

<table>
<thead>
<tr>
<th>Role</th>
<th>Object Privileges</th>
<th>Group Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auditor</td>
<td>- Can access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption</td>
<td>- Can access statistics on all dSources, VDBs, or snapshots in the group such as usage, history, and space consumption</td>
</tr>
<tr>
<td>Owner</td>
<td>- Can provision VDBs from owned dSources and VDBs</td>
<td>- Can provision VDBs from all dSources and VDBs in the group</td>
</tr>
<tr>
<td></td>
<td>- Can perform Virtual to Physical (V2P) from owned dSources</td>
<td>- Can create new dSources from linked environments</td>
</tr>
<tr>
<td></td>
<td>- Can access the same statistics as an Auditor</td>
<td>- Can perform Virtual to Physical (V2P) from owned dSources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Can apply Custom policies to dSources and VDBs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Can create Template policies for the group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Can assign Owner privileges for dSources and VDBs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Can access the same statistics as an Auditor</td>
</tr>
</tbody>
</table>

Related Links

- Adding Delphix Users
Adding Delphix Users and Privileges

This topic describes how to add Delphix users and assign them privileges on objects.

Prerequisites

If you intend to validate user logins using LDAP authentication, make sure a system administrator has configured LDAP as described in Setting Up the Delphix Engine.

Procedure

1. Launch the Delphix Admin application and log in as a Delphix Admin user.
2. Select Manage > Users.
3. Click Add User.
   A new user profile will open on the right side.
4. Enter user name, email, and password information for the new user.
5. Clear the Delphix Admin selection, if necessary, and click Save.
   A Privileges tab will be added to the user profile. See User Privileges for Delphix Objects for more information about privileges.
6. Assign the user Owner or Auditor privileges for appropriate Delphix objects.

Assigning Owner and Auditor Privileges

Assigning Owner privileges at the Group level conveys ownership privileges over all objects in that group. Click the expand icon next to each group name to see all objects in that group.

You can also assign ownership privileges only for specific objects in a group. You do not have to assign owner or auditor privileges for all Delphix objects, only those for which you want to grant the user specific access.

Related Links

- Setting Up the Delphix Engine
- Adding Delphix Admin Users
- User Privileges for Delphix Objects
Editing, Deleting, and Suspending Delphix Users

This topic describes how to suspend or delete Delphix users, and how to edit user information.

The delphix_admin user

The user named delphix_admin cannot be deleted since this is a user created by the Delphix Engine. However, you can suspend it.

Procedure

1. Launch the the Delphix Admin application and log in as a Delphix Admin user.
2. Select Manage > Users.
3. Click the user's name to open the user's profile panel.
4. Edit the user's profile information or object privileges as necessary.
5. Click the suspend icon to suspend that user.
6. Click the trash can icon to delete the user.

Deleting a user cannot be undone.
Assigning Group and Object Ownership

This topic describes how to assign group and object ownership to users in the Delphix Domain.

Procedure

1. Log into the Delphix Admin application as a user with Delphix Admin privileges.
2. Select Manage > Users.
3. For an existing user, click the user name to open the User Profile manager.
4. Click the Privileges tab.
5. Assign Owner or Auditor rights for groups or objects within groups.
   You do not have to assign a specific owner or auditor right for each object.
6. Click Commit when finished.
7. For new users, follow the instructions in Adding Delphix Users and Privileges. When you click Save, the User Profile manager will reload, and then you can follow steps 4 - 6 to assign privileges.

Related Links

- Adding Delphix Users and Privileges
- User Privileges for Delphix Objects
Adding and Deleting Groups
This topic describes how to add and delete groups within the Delphix Domain.

Adding a Group
1. Log into the Delphix Admin application as a user with Delphix Admin privileges.
2. In the Databases menu, select Add New Group.
3. Enter a Group Name and an optional description.
4. Click OK.

Deleting a Group
1. Log into the Delphix Admin application as a user with Delphix Admin privileges or group OWNER privileges for the target group.
2. Open the group card in the Databases panel by selecting the target group.
3. Click the Trash Can icon.
4. Click OK.

⚠️ Deleting Groups Containing Objects
A group cannot be deleted if it contains VDBs or dSources. All databases within a group must be deleted prior to deleting the group.

✅ At Least One Group Must Exist
At least one group must always exist on the Delphix Engine in order to link a dSource. If you delete the last group, you will need to create a new group in order to create a dSource.
Adding Delphix Admin Users

This topic describes how to add Delphix Admin users.

Prerequisites

You must be a Delphix Admin user to create another Delphix Admin user.

Procedure

1. Launch the the Delphix Admin application and log in.
2. Select Manage > Users.
3. Click Add User.
   A new user profile panel will open on the right side.
4. Enter user name, email, and password information for the new user.
5. Select Delphix Admin.
   Unlike ordinary Delphix users, Delphix Admin users are not shown a Privileges tab. This is because they have full privileges over all objects.
6. Click Save.
Managing Individual Profile Information

This topic describes how individual users can manage personal settings such as personal information, passwords, event notifications, and session timeouts. It also describes how users can view their privileges for Delphix objects.

Procedure

1. After logging in, click your name in the menu bar.
2. Click Profile.
3. Edit profile information as necessary.
4. Select options for the event level that will trigger a notification email.
5. Select a time period for Session Timeout.
6. Click Password to edit your password.
7. Click OK when finished.
8. Click Privileges to see your privileges (Auditor or Owner) for Delphix objects.
Managing Policies

These topics describe creating and managing SnapSync, LogSync, Retention, and VDB Refresh policies.

- Managing Policies: An Overview
- Creating Custom Policies
- Creating Policy Templates
- Policies and Time Zones
- Configuring Retention on Individual Snapshots
Managing Policies: An Overview

This topic describes the types of policies that you can use to manage database objects in the Delphix Engine.

There are four categories of policies that the Delphix Engine uses in conjunction with database objects:

1. **SnapSync** - how often snapshots of a source database are taken for a dSource.

   SnapSync policies only apply to Oracle databases and dSources. For information on how Microsoft SQL Server dSources stay in sync with the source database, see [Setting Up SQL Server Environments: An Overview](#).

2. **VDB Snapshot** - how often snapshots are taken of the VDB.

3. **Retention** - how long snapshots and log files are retained for dSources and VDBs.

4. **VDB Refresh** - a destructive process that is used only if you need to re-provision VDBs from their sources at regular intervals. The default setting for this policy is **None**.

   **Setting the VDB Refresh Policy Interval**

   Since VDB Refresh is a re-provisioning process, it is important to set the policy interval for an amount of time that will allow the VDB to fully re-provision before another refresh takes place. For example, if you set the VDB Refresh policy to initiate a refresh every 15 minutes, it is possible that the VDB will not fully re-provision in that amount of time, and your refresh process will fail.

There can additionally be *default*, *custom*, or *template* policies for each of these categories.

<table>
<thead>
<tr>
<th>Policy Type</th>
<th>Description</th>
<th>Who Can Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Default policies exist at the domain level and are applied across all objects in a category. You can modify the settings for a default policy in a category, but you cannot change the name <strong>default</strong>.</td>
<td>• Users with Delphix Admin credentials</td>
</tr>
<tr>
<td>Custom</td>
<td>Custom policies can only be applied to a specific database object. These cannot be saved to be used with other objects. You can create custom policies for dSources during the dSource linking process, as described in the <strong>Linking and Advanced Data Management Settings</strong> topics for each database platform type. See also <a href="#">Creating Custom Policies</a>.</td>
<td>• Users with Delphix Admin credentials • Group and object owners</td>
</tr>
</tbody>
</table>
| Template | Template policies are **named** policies that can be saved and applied to other database objects and to groups. These are created on the **Policy Management** screen. See [Creating Policy Templates](#) for more information. | • Users with Delphix Admin credentials  
• Group and object owners |

**Setting Different Policies for Objects in a Group**

Policies applied at the group level will affect all objects in that group. If you want to set different policies for objects in a group, apply the policies at the group level first, then apply policies at the object level.
Creating Custom Policies

This topic describes creating custom policies based on cron expressions for specific database objects or groups.

Custom policies are created by editing a policy associated with a database object, either during its creation or through the Policy Management screen after it has been created. For information about creating custom policies for dSources and VDBs during the linking and provisioning processes, see the Linking and Provisioning topics listed for each data platform.

Procedure

1. Log into the Delphix Admin application as a user with Delphix Admin privileges.
2. Select Manage > Policies.
3. Select the policy for the object or group you want to modify.
4. Click Apply New Policy.
5. Select Customized.
6. Enter the cron expressions you want to use for the policy. The expected format is compatible with the Quartz CronTrigger scheduler.
7. Click OK.
Creating Policy Templates

This topic describes how to create policy templates that can be applied to groups and database objects.

Unlike custom policies, template policies can be saved and applied to other groups and database objects. See the topics under Users, Permissions, and Policies for more information on using policies with groups.

Procedure

1. Log into the Delphix Admin application as a user with Delphix Admin privileges.
2. Select Manage > Policies.
3. Click Modify Policy Templates.
4. Under the category where you want to create the template, click Add New Policy.
5. Enter a Name for the template.
6. Enter the new policy settings.
7. Click OK.

Post-Requisites

- You can apply the new policy by selecting the appropriate policy category for an existing object or group, and then select the template policy

Related Links

- Users, Permissions, and Policies
Policies and Time Zones

You can configure the SnapSync, VDBSnapshot, and VDBRefresh policies with the time zone in which the policy should be scheduled.

To edit the time zone of a policy:

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
4. Select the policy.
5. In the policy card, click the pencil icon.
6. Select the appropriate time zone from the dropdown list.

Note: Retention and Quota policies are not schedulable and do not need a time zone.

Upgrading to Version 4.1.2 or higher:

Prior to version 4.1.2, a policy operated under the time zone of the policy’s target. For example, a SnapSync Policy scheduled for 4:00am every day that targeted a dSource in Eastern Standard Time (EST) and a dSource in Pacific Standard Time (PST) fired twice a day: once at 4:00am EST and once at 4:00am PST.

To maintain the same behavior of the Delphix Engine after upgrade, the upgrade process clones existing policies with these clones differing only in their time zone. After upgrading, you may notice that the names of policies change to include the time zones in which they operate.

Note: Default policies are not cloned and always operate under the time zone of the Delphix Engine.

Example Upgrade of an Engine

Pre-Upgrade

Sample Policy Dashboard, Version 4.0.0

Post-Upgrade

Sample Policy Dashboard, Version 4.1.2

In this example, the dSources and VDBs originally operated under either EST (America/New_York) or CST (America/Mexico_City), and new policies were created to reflect this.
<table>
<thead>
<tr>
<th>Original Policy</th>
<th>New Policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserSnapSync</td>
<td>UserSnapSync (America/Mexico_City)</td>
</tr>
<tr>
<td></td>
<td>UserSnapSync (America/New_York)</td>
</tr>
<tr>
<td>SnapshotTest</td>
<td>SnapshotTest (America/Mexico_City)</td>
</tr>
<tr>
<td></td>
<td>SnapshotTest (America/New_York)</td>
</tr>
<tr>
<td>UserRefresh</td>
<td>UserRefresh (America/Mexico_City)</td>
</tr>
<tr>
<td></td>
<td>UserRefresh (America/New_York)</td>
</tr>
</tbody>
</table>

After an upgrade, ensure that the policies are configured as expected; it may have been unclear prior to this upgrade when policies were actually firing.
Configuring Retention on Individual Snapshots

This topic describes adding a custom retention definition for individual snapshots. This value will override that of the policy currently assigned to the container, for example if ‘forever’ is selected then the snapshot will no longer be deleted via the retention policy.

Procedure
1. Log into the Delphix Admin application as a user with Delphix Admin privileges.
2. Select Resources > Capacity.
3. Expand the object (dsource or vdb) to modify.
4. Expand the snapshots. (it may take a few minutes for the individual snapshots to appear)
5. Configure the desired value in the ‘keep until’ column, either the number of days or tick forever.
Oracle Environments and Data Sources
Oracle Support and Requirements

These topics describe specific requirements for Oracle environments, such as user privileges, and the supported versions of Oracle DBMS and compatible operating systems.

- [Network and Connectivity Requirements for Oracle Environments](#)
- [Requirements for Oracle Source Hosts and Databases](#)
- [Requirements for Oracle Target Hosts and Databases](#)
- [Sudo File Configurations](#)
- [Sudo Privilege Requirements](#) — This topic describes the rationale behind specific sudo privilege requirements for working with the Delphix Engine.
- [Supported Operating Systems and DBMS Versions for Oracle Environments](#)
Supported Operating Systems and DBMS Versions for Oracle Environments

This topic describes the Oracle Database Management System (DBMS) versions that are supported by Delphix, as well as the compatible operating systems (OS), for use on target and source environments.

Source and Target OS and DBMS Compatibility

The source and target must be running the same DBMS/Operating System combination (for example, Oracle 10.2.0.4 on RHEL 5.2) in order to successfully provision a VDB to the target. If the DBMS versions are compatible, the OS version on a target host can be different from the OS version on the source host.

Supported DBMS Versions

- Oracle 9.2.0.8
- Oracle 10.2
- Oracle 11.1
- Oracle 11.2
- Oracle 12.1

Oracle 9.2.0.8

The Delphix Engine has limited support for Oracle 9.2.0.8 and cannot link to a database that has a compatibility setting lower than 9.2.0.8.

Oracle 10.2.0.4

The Delphix Engine does not support Oracle 10.2.0.4 databases using Automatic Storage Management (ASM) that do not have the patch set for Oracle Bug 7207932. This bug is fixed in patch set 10.2.0.4.2 onward.

Supported Operating Systems

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Version</th>
<th>Processor Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris</td>
<td>9, 10U5 - 10U11, 11U1, 11U2</td>
<td>SPARC</td>
</tr>
<tr>
<td>Solaris</td>
<td>10U5 - 10U11, 11U1, 11U2</td>
<td>x86_64</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux</td>
<td>4.7, 4.8, 4.9, 5.3 - 5.11, 6.0 - 6.6</td>
<td>x86_64</td>
</tr>
<tr>
<td>Oracle Enterprise Linux</td>
<td>5.3 - 5.11, 6.0 - 6.6</td>
<td>x86_64</td>
</tr>
<tr>
<td>Novell SUSE Linux Enterprise Server</td>
<td>10, 10SP1, 10SP2, 10SP3 11, 11SP1, 11SP2, 11SP3</td>
<td>x86_64</td>
</tr>
<tr>
<td>AIX</td>
<td>5.3, 6.1, 7.1</td>
<td>Power</td>
</tr>
</tbody>
</table>
Delphix supports all 64-bit OS environments for source and target, though 64-bit Linux environments also require that a 32-bit version of glibc installed.

⚠️ **Required HP-UX patch for Target Servers**

PHNE_37851 - resolves a known bug in HP-UX NFS client prior to HP-UX 11.31.
Requirements for Oracle Source Hosts and Databases

This topic describes the configuration and settings requirements for Oracle source hosts and databases, collectively referred to as source environments. Virtual database copies (VDBs) are created from these source environments.

- **Source Host Requirements**
- **Source Database Requirements**
- **Additional requirements for RAC sources**
- **Related Links**

Source Host Requirements

1. The ORATAB file must exist (typically in `/etc/oratab` or `/var/opt/oracle/oratab`).
2. On 64-bit Linux environments, there must be a 32-bit version of `glibc`.
   
   **How to Check for 32-bit glibc on 64-bit Linux**
   
   ```
   $ rpm -qa|grep glibc
   glibc-devel-2.12-1.107.el6_4.5.x86_64 == 64-bit
   glibc-devel-2.12-1.107.el6_4.5.i686 == 32-bit
   glibc-2.12-1.107.el6_4.5.x86_64
   glibc-common-2.12-1.107.el6_4.5.x86_64
   glibc-headers-2.12-1.107.el6_4.5.x86_64
   glibc-2.12-1.107.el6_4.5.i686 =========== 32-bit
   ```

3. There must be an operating system user (**delphix_os**) with these privileges:
   
   a. Profile and privileges should be the same as the Oracle user (**oracle**) on the host. For example, **delphix_os** should have the same environment variable settings (`$PATH`, `$ORACLE_HOME`, etc.) and `ulimit` settings, as **oracle**.
   
   b. Ability to log into the source host via SSH.
   
   c. Group memberships:
      
      i. The user’s primary group must be the UNIX group that is mapped to OSDBA by the Oracle installation. This is typically the **dba** group on the host.
      
      **Oracle 12c**
      
      For Oracle 12c and later versions of Oracle databases, the **delphix_os** user can also use OSBACKUPDBA as its primary group. This is typically the **bacdb** group on the host.
      
      ii. If the Oracle install group (typically **oinstall**), exists on the host, it should be set as a secondary group for the user.
      
      iii. If the Oracle ASM groups (typically **asmadmin** and **asmdba**) exist on the host, they should be assigned to the user as secondary groups.
d. Read access to either /etc/orainst.loc or /var/opt/oracle/orainst.loc.

e. Read access to the Oracle inventory file (inventory.xml) identified by the contents of oraist.loc (for example, $INVENTORY_HOME/ContentsXML/inventory.xml).

f. For each Oracle Home listed in the inventory file, the delphix_os user should have:

i. Read access to $ORACLE_HOME and all underlying files and directories.

ii. Write permission to the $ORACLE_HOME/sqlplus directory. This is required for execution of certain RMAN scripts.

iii. Execute permission for the programs in $ORACLE_HOME/bin.

iv. The $ORACLE_HOME/bin/oracle executable must have the SETUID and SETGID flags set. Permissions on the oracle binary must be -rwxr-s-x (06751) but you can also use more permissive settings.

g. Permission to run pargs on Solaris hosts and ps on AIX, HP-UX, Linux hosts, as super-user. This permission is usually granted via sudo authorization of the commands. See the topic Sudo Privilege Requirements for further explanation of this requirement, and Sudo File Configurations for examples of file configurations.

h. If the source host is a node in a RAC cluster, the delphix_os user must have these privileges:

i. Must exist as a user on all nodes in the cluster.

ii. Must have the same configuration on all nodes in the cluster, including profile, ulimits, user id, group membership, etc.

iii. Must have permission to run crsctl and srvctl on each node in the cluster.

Example: This shows that the group dba has read/write/execute permission on the database resources

```
$ crsctl get hostname
node2
```

4. There must be a directory on the source host where the Delphix Engine Toolkit can be installed, for example: /var/opt/delphix/Toolkit.

a. The delphix_os user must own the directory.

b. The directory must have permissions -rwxrwx--- (0770), but you can also use more permissive settings.

c. The delphix_os user must have read and execute permissions on each directory in the path leading to the toolkit directory. For example, when the toolkit is stored in /var/opt/delphix/Toolkit, the permissions on /var, /var/opt, and /var/opt/delphix should allow read and execute for ‘others’ (for example, -rwxr-xr-x).

d. The directory should have 1.5GB of available storage: 400MB for the toolkit and 400MB for the set of
logs generated by each client that runs out of the toolkit.

5. `alter database enable block change tracking;

On a Solaris host, `gtar` must be installed. Delphix uses `gtar` to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The `gtar` binary should be installed in one of the following directories:

- `/bin:/usr`
- `/bin:/sbin:/usr`
- `/sbin:/usr/contrib`
- `/bin:/usr/sfw`
- `/bin:/opt/sfw`
- `/bin:/opt/csw/bin`

6. The Delphix Engine must be able to make an SSH connection to the source host.

**Source Database Requirements**

1. Source databases must be in **ARCHIVELOG** mode to ensure that redo logs are archived.

2. If the source database is running Oracle Enterprise Edition or is an 11g Standby, Delphix recommends enabling **Block Change Tracking (BCT)**.

   Enter this command to enable BCT:

   ```
   alter database enable block change tracking;
   ```

   You can also use this alternative version of the command:

   ```
   alter database enable block change tracking using file '<user specified file>;
   ```

3. Do not use the **NOLOGGING** option for operations on databases that are linked by the Delphix Engine. A best practice is to use **FORCE LOGGING** at the database level to force logging of changes to the redo. Oracle also requires use of FORCE LOGGING for proper management of standby databases. If you must use NOLOGGING to meet specific performance criteria, take a new snapshot of the source database after doing the NOLOGGING operations to bring the dSource up-to-date before provisioning VDBs.

   Enter this command to enable FORCE LOGGING:

   ```
   SQL> ALTER DATABASE force logging;
   ```

4. If the online redo log files are located on RAW or ASM devices, then the Delphix Engine LogSync feature can operate in **Archive Only** mode only. See the topics [Advanced Data Management Settings for Oracle dSources](#) and [Linking Oracle Physical Standby Databases](#) for more information.

5. There must be a database user (`delphix_db`) created by the `createDelphixDBUser.sh` script. This script is part of the HostChecker bundle, and grants **SELECT** privileges on specific system tables for the user. See the topics [Using HostChecker to Validate Oracle Source and Target Environments](#) for more about using
the HostChecker bundle.

<table>
<thead>
<tr>
<th><strong>Oracle pluggable databases</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>For an Oracle pluggable database, there must be one database user (<em>delphix_db</em>) for the pluggable database and one common database user (<em>c##delphix_db</em>) for its container database. The createDelphixDBUser.sh script can create both users.</td>
</tr>
</tbody>
</table>

### Additional requirements for RAC sources
- The Delphix Toolkit must be installed in the same directory on each of the nodes in the source cluster.
- All datafiles and archive logs must be located on storage shared by all of the cluster nodes. Each node in the cluster must be able to access archive logs from all other nodes.

### Related Links
- [Requirements for Oracle Target Environments and Databases](#)
- [Using HostChecker to Configure Oracle Environments](#)
- [Using HostChecker to Confirm Source and Target Environment Configuration](#)
- [Sudo File Configurations](#)
- [Sudo Privilege Requirements](#)
Requirements for Oracle Target Hosts and Databases

This topic describes the user privileges, and environment discovery requirements, that are required for Oracle and Oracle RAC target hosts and databases, collectively referred to as target environments.

Target Host Requirements

1. The operating system on the target host must be the same as, or binary compatible, with the operating system on the source host.
2. There must be a version of the Oracle binaries on the target environment that is identical (same version and patch set) to the binaries for the source database on the source host. For example, if the source database is running on Oracle 11.2.0.3.4, then the target environment must have the binaries for 11.2.0.3.4 installed.
3. On 64-bit Linux environments, there must be a 32-bit version of glibc.

```
How to Check for 32-bit glibc on 64-bit Linux

$ rpm -qa|grep glibc
  glibc-devel-2.12-1.107.el6_4.5.x86_64 <= 64-bit
  glibc-devel-2.12-1.107.el6_4.5.i686 <= 32-bit
  glibc-2.12-1.107.el6_4.5.x86_64
  glibc-common-2.12-1.107.el6_4.5.x86_64
  glibc-headers-2.12-1.107.el6_4.5.x86_64
  glibc-2.12-1.107.el6_4.5.i686 <= 32-bit
```

4. There must be an operating system user (delphix_os) with these privileges:
   a. Profile and privileges should be the same as the Oracle user (oracle) on the target host.
      For example, delphix_os should have the same environment variable settings ($PATH, $ORACLE_HOME, etc.) and ulimit settings, as oracle.
   b. Ability to log into the source host via SSH.
   c. Group memberships:
      i. The user's primary group must be the UNIX group that is mapped to OSDBA by the Oracle installation. This is typically the dba group on the host.
      ii. If the Oracle install group (typically oinstall), exists on the host, it should be set as a secondary group for the user.
      iii. If the Oracle ASM groups (typically asmadmin and asmdba) exist on the host, they should be assigned to the user as secondary groups.

```
LDAP/NIS User

If the delphix_os user is a LDAP/NIS user, this user must be a member of the dba and oinstall groups in /etc/groups locally in order for Oracle commands to run properly.
```

d. Read access to either /etc/orainst.loc or /var/opt/oracle/orainst.loc.
e. Read access to the Oracle inventory file (inventory.xml) identified by the contents of orainst.loc (for example, $INVENTORY_HOME/ContentsXML/inventory.xml).
f. For each Oracle Home listed in the inventory file, the delphix_os user should have:
i. Read access to $ORACLE_HOME and all underlying files and directories
ii. Write permission to the $ORACLE_HOME/dbs directory
iii. The $ORACLE_HOME/bin/oracle executable must have the SETUID and SETGID flags set. Permissions on the oracle binary must be -rwxr-s-x (06751) but more permissive settings can also be used.

4. Write permission to the the selected Delphix mount-point directory.

5. The following permissions are usually granted via sudo authorization of the commands. See Sudo Privilege Requirements for further explanation of the commands, and Sudo File Configurations for examples of the /etc/lsudoers file on different operating systems.

   i. Permission to run mount, umount, mkdir, rmdir, ps as super-user.
   ii. Permission to run pargs on Solaris hosts and ps on AIX, HP-UX, Linux hosts, as super-user.
   iii. If the target host is an AIX system, permission to run the nfsd command as super-user.

   i. If the target host is a node in a RAC cluster, the delphix_os must have these privileges:
      i. Must exist as a user on all nodes in the cluster
      ii. Must have the same configuration (profile, privileges, user id, group membership, etc.) on all nodes in the cluster
      iii. Must have permission to run crsctl and srvctl on each node in the cluster.

Example: This shows that the group dba has read/write/execute permission on the database resources

```
$ crsctl getperm resource ora.trois.db
Name: ora.trois.db
owner:orall2:rwx,pgrp:dba:rwx,other::r--
```

5. There must be a directory on the target environment where the Delphix Engine toolkit can be installed, for example /home/delphix/Toolkit.

   a. The directory must be owned by the delphix_os user.
   b. The directory must have permissions 0770 (for example, -rwxrwx--), but more permissive settings can also be used.
   c. The delphix_os user must have read and execute permissions on each directory in the path leading to the toolkit directory. For example, when the toolkit is stored in /home/delphix/Toolkit, the permissions on /home and /home/delphix should allow read and execute for ‘others’ (for example, -rwxr-xr-x).
   d. The directory should have a total of at least 800MB of storage plus 1MB of storage per VDB that will be provisioned to the target.

6. On a Solaris host, gtar must be installed. Delphix uses gtar to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The gtar binary should be installed in one of the following directories:

   * /bin:/usr
   * /bin:/sbin:/usr
   * /sbin:/usr/contrib
7. There must be an empty directory (/delphix) that will be used as a container for the mount points that are created when provisioning a VDB to the target host. The group associated with the directory must be the primary group of the delphix_os user (typically dba). Group permissions for the directory should allow read, write, and execute by members of the group.

8. The Delphix Engine must be able to make an ssh connection to the target host.

9. An Oracle listener process should be running on the target host. The listener’s version should be equal to or greater than the highest Oracle version that will be used to provision a VDB.

10. NFS client services must be running on the target host.

Oracle Target Container Databases Requirements

To provision an Oracle pluggable database, there must be a running Oracle multitenant container database in the target environment. In the multitenant container database, there must be a common database user (c##delphix_db) created by the createDelphixDBUser.sh script. This script is part of the HostChecker bundle, and grants SELECT privileges on specific system tables for the user. See the topics Using HostChecker to Validate Oracle Source and Target Environments for more about using the HostChecker bundle.

Related Links

- Requirements for Oracle Source Hosts and Databases
- Using HostChecker to Validate Oracle Source and Target Environments
- Network and Connectivity Requirements for Oracle Environments
- Sudo Privilege Requirements
- Sudo File Configurations
Network and Connectivity Requirements for Oracle Environments

- General Port Allocation
  - General Outbound from the Delphix Engine Port Allocation
  - General Inbound to the Delphix Engine Port Allocation
- Firewalls and Intrusion Detection Systems (IDS)
- SSHD Configuration

Network and Connectivity Requirements for Oracle Port Allocation for Oracle Environments
- Outbound from the Delphix Engine Port Allocation
- Inbound to the Delphix Engine Port Allocation

General Port Allocation

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

General Outbound from the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>25</td>
<td>Connection to a local SMTP server for sending email</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>53</td>
<td>Connections to local DNS servers</td>
</tr>
<tr>
<td>UDP</td>
<td>123</td>
<td>Connection to an NTP server</td>
</tr>
<tr>
<td>UDP</td>
<td>162</td>
<td>Sending SNMP TRAP messages to an SNMP Manager</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
<td>SSL connections from the Delphix Engine to the Delphix Support upload server</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>636</td>
<td>Secure connections to an LDAP server</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Connections to a Delphix replication target (see Configuring Replication)</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections to source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>

General Inbound to the Delphix Engine Port Allocation
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>80</td>
<td>HTTP connections to the Delphix GUI</td>
</tr>
<tr>
<td>UDP</td>
<td>161</td>
<td>Messages from an SNMP Manager to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>443</td>
<td>HTTPS connections to the Delphix GUI</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections from source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>32768 - 65535</td>
<td>Required for NFS mountd and status services from target environment only if the firewall between Delphix and the target environment does not dynamically open ports. <strong>Note</strong>: If no firewall exists between Delphix and the target environment, or the target environment dynamically opens ports, this port range is not explicitly required.</td>
</tr>
</tbody>
</table>

**Firewalls and Intrusion Detection Systems (IDS)**

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

**SSHD Configuration**

Both source and target Unix environments are required to have sshd running and configured such that the Delphix...
Engine can connect over \texttt{ssh}.

The Delphix Engine expects to maintain long-running, highly performant \texttt{ssh} connections with remote Unix environments. The following \texttt{sshd} configuration entries can interfere with these \texttt{ssh} connections and are thus disallowed:

<table>
<thead>
<tr>
<th>Disallowed \texttt{sshd} Configuration Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientAliveInterval</td>
</tr>
<tr>
<td>ClientAliveCountMax</td>
</tr>
</tbody>
</table>

### Network and Connectivity Requirements for Oracle

- IP connections must exist between the Delphix Engine and source and target environments.

- For source environments, Delphix Engine uses an \texttt{SSH} connection to each source host, an \texttt{HTTP} connection from each source environment to Delphix Engine, and a DSP connection to the Delphix Engine. The Delphix Engine uses \texttt{SQL*Net} connections to the DBMS on the source environment.

- For target environments, Delphix uses an \texttt{SSH} connection to each target environment, and an \texttt{NFS} connection to Delphix Engine. Delphix Engine uses \texttt{SQL*Net} connections to the virtual databases on the target environment.

⚠️ **scp Availability**

The \texttt{scp} program must be available in the environment in order to add an environment.

### Port Allocation for Oracle Environments

The Delphix Engine makes use of the following network ports for Oracle dSources and VDBs:

#### Outbound from the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to source and target environments</td>
</tr>
<tr>
<td>TCP</td>
<td>xxx</td>
<td>Connections to the Oracle SQL*Net Listener on the source and target environments (typically port 1521)</td>
</tr>
</tbody>
</table>

#### Inbound to the Delphix Engine Port Allocation
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/UDP</td>
<td>111</td>
<td>Remote Procedure Call (RPC) port mapper used for NFS mounts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: RPC calls in NFS are used to establish additional ports, in the high range 32768-65535, for supporting services. Some firewalls interpret RPC traffic and open these ports automatically. Some do not; see below.</td>
</tr>
<tr>
<td>TCP</td>
<td>1110</td>
<td>NFS Server daemon status and NFS server daemon keep-alive (client info)</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>2049</td>
<td>NFS Server daemon from VDB to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>4045</td>
<td>NFS lock daemon/manager</td>
</tr>
<tr>
<td>TCP</td>
<td>8341</td>
<td>Sending data from source to the Delphix Engine (for LogSync)</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>SnapSync control and data from source to the Delphix Engine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V2P control and data from the target environment to the Delphix Engine.</td>
</tr>
<tr>
<td>UDP</td>
<td>33434 - 33464</td>
<td>Traceroute from source and target database servers to the Delphix Engine (optional)</td>
</tr>
<tr>
<td>UDP/TCP</td>
<td>32768 - 65535</td>
<td>NFS mountd and status services, which run on a random high port. Necessary when a firewall does not dynamically open ports.</td>
</tr>
</tbody>
</table>
# Sudo Privilege Requirements

This topic describes the rationale behind specific `sudo` privilege requirements for working with the Delphix Engine.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Sources</th>
<th>Targets</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>`ps</td>
<td>pargs`</td>
<td>Optional, Strongly Recommended</td>
<td>Optional, Strongly Recommended</td>
</tr>
<tr>
<td><code>mkdir/rmdir</code></td>
<td>Not Required</td>
<td>Optional</td>
<td>Delphix dynamically makes and removes directories under the provisioning directory during VDB operations. This privilege is optional, provided the provisioning directory permissions allow the <code>delphix_os</code> user to make and remove directories.</td>
</tr>
<tr>
<td><code>mount/umount</code></td>
<td>Not Required</td>
<td>Required</td>
<td>Delphix dynamically mounts and unmounts directories under the provisioning directory during VDB operations. This privilege is required because <code>mount</code> and <code>umount</code> are typically reserved for superuser.</td>
</tr>
<tr>
<td>nfso (AIX only)</td>
<td>Not Required</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>Delphix monitors NFS read and write sizes on an AIX target host. It uses the <code>nfs o</code> command to query the sizes in order to optimize NFS performance for VDBs running on the target host. Only a superuser can issue the <code>nfs o</code> command.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Links

- [Requirements for Oracle Source Hosts and Databases](#)
- [Requirements for Oracle Target Hosts and Databases](#)
- [Sudo File Configurations](#)
Sudo File Configurations

This topic describes sudo file privilege configurations for using the Delphix Engine with various operating systems and for specific security requirements.

Configuring sudo Access on Solaris SPARC for Source and Target Environments

Sudo access to \texttt{pargs} on the Solaris operating system is required for the detection of listeners with non-standard configurations on both source and target environments. Super-user access level is needed to determine the \texttt{TNS_ADMIN} environment variable of the user running the listener (typically \texttt{oracle}, the installation owner). From \texttt{TNS_ADMIN}, the Delphix OS user \texttt{delphix_os} can derive connection parameters.

Here are examples of sudo entries for a Delphix Source and Target environment:

### Example: Solaris /etc/sudoers entries for a Delphix Source

```bash
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/usr/bin/pargs
```

On a Solaris target, sudo access to \texttt{mount}, \texttt{umount}, \texttt{mkdir} and \texttt{rmdir} is also required. In this customer example, super-user privilege is restricted to the virtual database mount directory \texttt{/data/oracle/Delphix}.

### Example: Solaris /etc/sudoers entries for a Delphix Target

```bash
User_Alias DELPHIX_USER=delphix_os

Cmd_Alias DELPHIX_CMDS= \n/usr/sbin/mount /data/oracle/Delphix/*, \n/usr/sbin/mount 100.245.235.12:* /data/oracle/Delphix/*, \n/usr/sbin/mount -o* 100.245.235.12:* /data/oracle/Delphix/*, \n/usr/sbin/umount /data/oracle/Delphix/*, \n/usr/sbin/umount -f /data/oracle/Delphix/*, \n/usr/bin/mkdir /data/oracle/Delphix/*, \n/usr/bin/mkdir -p /data/oracle/Delphix/*, \n/usr/bin/rmdir /data/oracle/Delphix/*, \n/usr/bin/pargs

DELPHIX_USER ALL=(ALL) NOPASSWD: DELPHIX_CMDS
```

Configuring sudo Access on Linux for Source and Target Environments

Sudo access to \texttt{ps} on the Linux operating system is required for the detection of listeners with non-standard configurations on both source and target environments. Super-user access level is needed to determine the \texttt{TNS_ADMIN} environment variable of the user running the listener (typically \texttt{oracle}, the installation owner). From \texttt{TNS_ADMIN}, the Delphix OS user \texttt{delphix_os} can derive connection parameters.

Here are examples of sudo entries for a Delphix Source and Target environment:

### Example: Linux /etc/sudoers entries for a Delphix Source

```bash
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/bin/ps
```

On a Linux target, sudo access to \texttt{mount}, \texttt{umount}, \texttt{mkdir} and \texttt{rmdir} is also required. In this customer example,
super-user privilege is restricted to the virtual database mount directory `/oracle/.

### Example: Linux `/etc/sudoers file for a Delphix Target`

```plaintext
Defaults:delphix_os !requiretty

Cmd_Alias DELPHIX_ADMIN_CMDS= \
/bin/mount /oracle/*, \ 
/bin/mount * /oracle/*, \ 
/bin/umount /oracle/*, \ 
/bin/umount * /oracle/*, \ 
/bin/mkdir -p -m 755 /oracle/*, \ 
/bin/mkdir -p /oracle/*, \ 
/bin/mkdir /oracle/*, \ 
/bin/rmdir /oracle/*, \ 
/bin/ps

Host_Alias DELPHIX_HOSTS=delphix001, delphix002

delphix_os DELPHIX_HOSTS=NOPASSWD:DELPHIX_ADMIN_CMDS
```

### Configuring `sudo` Access on AIX for Source and Target Environments

Sudo access to `ps` on the AIX operating system is required for the detection of listeners with non-standard configurations on both source and target environments. Super-user access level is needed to determine the `TNS_ADMIN` environment variable of the user running the listener (typically `oracle`, the installation owner). From `TNS_ADMIN`, the Delphix OS user `delphix_os` can derive connection parameters.

#### Example: AIX `/etc/sudoers entries for a Delphix Source`

```plaintext
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/bin/ps
```

In addition to sudo access to the `mount`, `umount`, `mkdir`, `rmdir`, and `ps` commands on AIX target hosts, Delphix also requires `sudo` access to `nfso`. This is required on target hosts for Delphix to monitor the NFS read write sizes configured on the AIX system. Super-user access level is needed to run the `nfso` command.

#### Example: AIX `/etc/sudoers File for a Delphix Target`

```plaintext
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/bin/mount, \ 
/bin/umount, \ 
/bin/mkdir, \ 
/bin/rmdir, \ 
/bin/ps, \ 
/usr/sbin/nfso
```

### Configuring `sudo` Access on HP-UX for Source and Target Environments

No `sudo` privileges are required on source environments running HP-UX. The HP-UX OS does not allow the `delphi x_os` user to determine `TNS_ADMIN` environment variable setting for the `oracle` user. This means that non-standard listener configurations, with non-default `TNS_ADMIN` settings, cannot be auto-discovered.
On the HP-UX target, `sudo` access to `mount`, `umount`, `mkdir` and `rmdir` is required as with other operating systems.

**Example: HP-UX /etc/sudoers file for a Delphix Target**

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /sbin/mount, /sbin/umount, /bin/mkdir, /bin/rmdir
```

**Other Examples of Limiting `sudo` Access for the Delphix OS User to the VDB Mount Directory Only**

In situations where security requirements prohibit giving the Delphix user root privileges to mount, unmount, make directory, and remove directory on the global level, it is also possible to configure the `sudoers` file to provide these privileges on the virtual database mount directory only, as shown in these two examples.

**Example 1**

This example restricts the `delphix_os` user's use of `sudo` privileges to the directory `/oracle`.

Note that wildcards are allowed for the options on `mount` and `umount` because those commands expect a fixed number of arguments after the options. The option wildcard on the `mount` command also makes it possible to specify the file-system being mounted from the Delphix Engine.

But wildcards are not acceptable on `mkdir` and `rmdir` because they can have any number of arguments after the options. For those commands you are required to specify the exact options (`-p`, `-p -m 755`) used by the Delphix Engine.

```
Example /etc/sudoers File Configuration on the Target Environment for sudo Privileges on the VDB Mount Directory Only

Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: 
/bin/mount * /oracle/*, 
/bin/umount * /oracle/*, 
/bin/umount /oracle/*, 
/bin/mkdir -p /oracle/*, 
/bin/mkdir -p -m 755 /oracle/*, 
/bin/mkdir /oracle/*, 
/bin/rmdir /oracle/*, 
/bin/ps
```

```
Example /etc/sudoers File Configuration on the Source Environment to grant Super-User privileges when running PS

Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: /bin/ps
```

**Example 2**

This example restricts the `delphix_os` user's use of `sudo` privileges to the directory `/mnt/delphix`.

This example demonstrates a very restrictive syntax for the `mount` and `umount` commands. The `umount` command
allows no user-specified options. The `mount` command specifies the Delphix Engine’s server name (or IP address) on the mount command so as to limit which file systems can be mounted.

### A Second Example of Configuring the `/etc/sudoers` File on the Target Environment for Privileges on the VDB Mount Directory Only

| Defaults | delphix_os !requiretty
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>delphix_os ALL=(root) NOPASSWD:</td>
<td></td>
</tr>
<tr>
<td>/usr/sbin/mount &lt;delphix-server-name&gt;* /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/sbin/mount * &lt;delphix-server-name&gt;* /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/sbin/mount &lt;delphix-server-ip&gt;* /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/sbin/mount * &lt;delphix-server-ip&gt;* /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/sbin/umount /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/sbin/umount -f /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/bin/mkdir [<em>] /mnt/delphix/</em>,</td>
<td></td>
</tr>
<tr>
<td>/usr/bin/mkdir /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/bin/mkdir -p /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/bin/mkdir -p -m 755 /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/bin/rmdir /mnt/delphix/*,</td>
<td></td>
</tr>
<tr>
<td>/usr/bin/ps,</td>
<td></td>
</tr>
<tr>
<td>/bin/ps</td>
<td></td>
</tr>
</tbody>
</table>

### Considerations for `sudo` access and account locking

The Delphix Engine tests its ability to run the `mount` command using `sudo` on the target environment by issuing the `sudo mount` command with no arguments. Many of the examples shown in this topic do not allow that, and in those cases the attempt will be blocked. In most situations, this does not cause a problem.

Similarly, the `ps` or `pargs` command is used for target environment operations such as initial discovery and refresh. The most restrictive `sudo` setups might not allow the commands `ps` (pargs), `mkdir`, and `rmdir`; strictly speaking, Delphix can still function without these privileges (see Sudo Privilege Requirements for a full explanation).

However, some users configure the security on the target environments to monitor `sudo` failures and lock out the offending account after some threshold. In those situations, the `delphix_os` account can become locked. One work-around for this situation is to increase the threshold for locking out the user account. Another option is to modify `/etc/sudoers` to permit the `delphix_os` user to run `ps` (pargs), `mkdir`, `rmdir`, and `mount` command without parameters.

### Related Links
- Sudo Privilege Requirements
- Requirements for Oracle Source Hosts and Databases
- Requirements for Oracle Target Hosts and Databases
Managing Oracle Environments

These topics describe special tasks and concepts for Oracle environments.

- Using HostChecker to Validate Oracle Source and Target Environments
- Adding an Oracle Single Instance or RAC Environment
- Adding a Database Installation Home to an Oracle Environment
- Adding a Database to an Oracle Environment
- Discovering Oracle Pluggable Databases in an Oracle Environment
- Adding a Listener to an Oracle Environment
- Changing the Host Name or IP Address for Oracle Source and Target Environments
- Editing Oracle Environment Attributes
- Managing Oracle Environment Users
- Enabling Linking and Provisioning for Oracle Databases
- Deleting an Oracle Environment
- Refreshing an Oracle Environment
Using HostChecker to Validate Oracle Source and Target Environments

- What is HostChecker?
- Prerequisites
- Procedure
- Non-Interactive Mode
- Tests Run
- Related Links

What is HostChecker?

The HostChecker is a standalone program which validates that host machines are configured correctly before the Delphix Engine uses them as data sources and provision targets.

Please note that HostChecker does not communicate changes made to hosts back to the Delphix Engine. If you reconfigure a host, you must refresh the host in the Delphix Engine in order for it to detect your changes.

You can run the tests contained in the HostChecker individually, or all at once. You must run these tests on both the source and target hosts to verify their configurations. As the tests run, you will either see validation messages that the test has completed successfully, or error messages directing you to make changes to the host configuration.

The Oracle HostChecker is distributed as a set of Java files and executables. You can find these files and executables in 5 distinct tarballs, each containing a different jdk corresponding to a particular platform (OS + processor). Together, these tarballs comprise the set of platforms supported by Delphix. When validating Oracle hosts during a new deployment, it is important to download the appropriate tarball for the host you are validating. Tarballs follow the naming convention "hostchecker_<OS>_<processor>.tar." For example, if you are validating a linux x86 host, you should download the tarball named hostchecker_linux_x86.tar.

The Oracle HostChecker is also included in the Delphix Toolkit which is pushed to every environment managed by the Delphix Engine. It can be found in /<toolkit-path>/<Delphix_COMMON>/client/hostchecker.

Prerequisites

- Make sure your Oracle environment meets the requirements described in Requirements for Oracle Source Hosts and Databases and Requirements for Oracle Target Hosts and Databases
- At minimum, the hostchecker requires Java 6 to run. However, the Java 6 binaries are included in each of the platform specific hostchecker tarballs and will be extracted if necessary.

Procedure

1. Download the appropriate HostChecker tarball for your platform. Tarballs follow the naming convention "hostchecker_<OS>_<processor>.tar". For example, if you are validating a linux x86 host you should download the hostchecker_linux_x86.tar tarball.

2. Create a working directory and extract the HostChecker files from the HostChecker tarball.

   ```bash
   mkdir dlpx-host-checker
   cd dlpx-host-checker/
   tar -xf hostchecker_linux_x86.tar
   ```

3. Run the `sh` script contained within:
This will extract the JDK included in the tarball (if necessary) and invoke the hostchecker.

```
oral0205@bbdhcp:/home/ora10205/hostchecker-> sh hostchecker.sh
Installed version of Java (version: 1.4.2) is not compatible with the hostchecker.
Java version 1.6 or greater required.
Using the JDK from the included tarball (already extracted).
```

⚠️ Don’t Run as Root

Do not run the HostChecker as root; this will cause misleading or incorrect results from many of the checks.

4. Select which **checks** you want to run.

✔️ **Run Tests without the Interface**

You can also run checks without spawning the interface. Enter `--help` to get a list of arguments you can pass to the HostChecker.

5. Enter the requested **arguments** as the checks are made.

6. Read the output of the check.
   The general format is that severity increases as you scroll down the output. First comes informational output, then warnings, then errors.

⚠️ **Internal Errors**

If you see a message that starts with Internal Error, please forward it to Delphix Support immediately. This represents a potential bug in the HostChecker, and not necessarily a problem with your environment.

7. Error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Don’t be surprised or undo your work if more errors appear the next time you run HostChecker, because the error you just fixed may have been masking other problems.

8. Repeat steps 3 - 7 until all the checks return no errors or warnings.

**Non-Interactive Mode**

The Java hostchecker can also be invoked in non-interactive mode. Each check is associated with a numeric flag; the association can be displayed using the `-help` input flag. To run a particular check pass in the associated flag.
java -jar hostchecker.jar -help
usage: java -jar hostchecker.jar [OPTIONS]
-0             Check homedir permissions
-1             Check network port access
-10            Check toolkit path
-2             Check the Oracle CRS home
-3             Check the Oracle installation
-4             Check Oracle DB Instance
-5             Check the oratab file
-6             Check for ssh connectivity
-7             Check sshd_config for timeout configuration
-8             Check user sudo privileges
-9             Check sunrpc.tcp_slot_table_entries
-all           Execute all checks
-help          Print this message
-input <arg>   Input to hostchecker as a JSON string

In non-interactive mode, the input parameters necessary to run the checks must be passed to the hostchecker as a JSON string using the -input flag.

java -jar hostchecker.jar -0 -1 -10 -2 -3 -5 -6 -7 -8 -9 -input
'{"toolkitPath":="/work","applianceIP":"kfc-trunk.dcenter.delphix.com","sudoCredentials
":{"username":"ora10205","password":"ora10205"},"sshCredentials":{"username":"ora10205
 ","password":"ora10205"},"port":"4000","source":true,"oracleCRSHome":="/work"}"

Tests Run

<table>
<thead>
<tr>
<th>Test</th>
<th>Applicable Platforms</th>
<th>Oracle Source</th>
<th>Oracle Target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Host Secure Shell (SSH) Connectivity</td>
<td>All</td>
<td>X</td>
<td>X</td>
<td>Verifies that the environment is accessible via SSH</td>
</tr>
<tr>
<td>Check Tool Kit Path</td>
<td>All</td>
<td>X</td>
<td>X</td>
<td>Verifies that the toolkit installation location is suitable – for example, that it has the proper ownership, permissions, and enough free space</td>
</tr>
<tr>
<td>Check Home Directory Permissions</td>
<td>All</td>
<td>X</td>
<td>X</td>
<td>Verifies that the host can be accessed via SSH using public key authentication. If you do not need this feature, you can ignore the results of this check, or you can choose not to run it.</td>
</tr>
<tr>
<td>Check Inventory Access</td>
<td>All</td>
<td>X</td>
<td>X</td>
<td>Verifies that the current user has access to the Oracle inventory file</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----</td>
<td>---</td>
<td>---</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>Check Oracle Installation</td>
<td>All</td>
<td>X</td>
<td>X</td>
<td>Verifies basic information about the Oracle installation on the system, including that various files are in expected locations, that they are formatted properly, and that they have the correct permissions</td>
</tr>
<tr>
<td>Check ORATAB File</td>
<td>All</td>
<td>X</td>
<td></td>
<td>Verifies that the oratab file is in an expected location and is formatted appropriately. You only need to run this on source machines.</td>
</tr>
<tr>
<td>Check Oracle DB Instance</td>
<td>All</td>
<td>X</td>
<td></td>
<td>Verifies more specific information both about the installation of oracle on the system and about the various databases. Information includes not only file locations, formatting, and permissions, but also the presence of DB listeners, database settings, oracle versions, oracle user permissions, and more. You only need to run this on source machines.</td>
</tr>
<tr>
<td>Check Oracle CRS Installation</td>
<td>All</td>
<td>X</td>
<td>X</td>
<td>Verifies settings related to Oracle CRS. You only need to run this on machines that have CRS set up.</td>
</tr>
<tr>
<td>Feature</td>
<td>Extent</td>
<td>Requirement</td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>Check OS User Privileges</td>
<td>All</td>
<td>X</td>
<td>Verifies that the operating system user can execute certain commands with necessary privileges via <code>sudo</code>. You only need to run this on target environments. See the topic <a href="#">Sudo Privilege Requirements</a> for more information.</td>
<td></td>
</tr>
<tr>
<td>Check SnapSync Connectivity</td>
<td>All</td>
<td>X</td>
<td>Verifies that the source host is able to connect to the Delphix Engine at port 8415 for SnapSync check transmission control protocol (TCP) slot table entries.</td>
<td></td>
</tr>
<tr>
<td>Check transmission control protocol (TCP) slot table entries</td>
<td>Linux RHEL 4.0-5.6</td>
<td>X</td>
<td>Check that the maximum number of (TCP) RPC requests that can be in flight is at least 128.</td>
<td></td>
</tr>
</tbody>
</table>

**Related Links**
- [Requirements for Oracle Source Hosts and Databases](#)
- [Requirements for Oracle Target Hosts and Databases](#)
Adding an Oracle Single Instance or RAC Environment

This topic describes how to add a new Oracle or Oracle RAC environment.

Prerequisites

- See the topics Requirements for Oracle Target Hosts and Databases and Supported Operating Systems and DBMS Versions for Oracle Environments

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click the Plus icon next to Environments.
4. In the Add Environment dialog, select Unix/Linux.
5. Select Standalone Host or Oracle Cluster, depending on the type of environment you are adding.
6. For standalone Oracle environments enter the Host IP address.
7. For Oracle RAC environments, enter the Node Address and Cluster Home.
8. Enter an optional Name for the environment.
9. Enter the SSH port.
   The default value is 22.
10. Enter a Username for the environment.
    See Requirements for Oracle Target Hosts and Databases for more information on the required privileges for the environment user.
11. Select a Login Type.
    For Password, enter the password associated with the user in Step 10.

   **Using Public Key Authentication**

   If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      ii. Run chmod 755 ~ to make your home directory writable only by your user.

   The public key needs to be added only once per user and per environment.

   You can also add public key authentication to an environment user’s profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

12. For Password Login, click Verify Credentials to test the username and password.
13. Enter a Toolkit Path.
    The toolkit directory stores scripts used for Delphix Engine operations, and should have a persistent working directory rather than a temporary one. The toolkit directory will have a separate sub-directory for each database instance. The toolkit path must have 0770 permissions and at least 345MB of free space.
14. Click OK.

Post-Requisites

After you create the environment, you can view information about it by selecting Manage > Environments, and then select the environment name.

Related Links

- Requirements for Oracle Target Hosts and Databases
Adding a Database Installation Home to an Oracle Environment

This topic describes how to add a database installation home.

When you add an environment with the Delphix Admin application, all database installation homes on it are automatically discovered. However, if a database installation home is not automatically discovered, you can add it manually to the environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Click the green Plus icon next to Add Installation Home.
5. Enter the Installation Home.
6. Click the Check icon when finished.

Related Links

- Adding a Database to an Oracle Environment
Adding a Database to an Oracle Environment

This topic describes adding a source database to an environment.

When you add an environment with the Delphix Admin application, all database instances that are running on it are automatically discovered. However, if a database is not automatically discovered, you can add it manually to the environment.

Prerequisites

- Make sure your source database meets the requirements described in Requirements for Oracle Source Hosts and Databases, as well as general database user requirements as described in Requirements for Oracle Target Hosts and Databases.
- Before adding a database, the installation home of the database must exist in the environment. If the installation home does not exist in the environment, follow the steps in Adding a Database Installation Home to an Oracle Environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Choose the installation home where the database is installed. Click the Up icon next to the installation home path to show details if needed.
5. Click the green Plus icon next to Add Databases.
6. Enter the Database Unique Name, Database Name, and Instance Name.
7. Click the Check icon when finished.

Related Links

- Requirements for Oracle Source Hosts and Databases
- Requirements for Oracle Target Hosts and Databases
- Adding a Database Installation Home to an Oracle Environment
Discovering Oracle Pluggable Databases in an Oracle Environment

When you add an environment with the Delphix Admin application, all Oracle database instances that are running on it are automatically discovered. These include multitenant container database instances. However, pluggable databases are not discovered. This topic describes how to discover Oracle pluggable database in the Oracle environment.

Prerequisites

- Make sure that the multitenant container database and its pluggable databases meet the requirements described in Requirements for Oracle Source Environments and Databases.
- Make sure that the multitenant container database is already discovered by the Delphix Engine. If the container database does not exist in the environment, follow the steps in Adding a Database to an Oracle Environment.

Procedure

1. Login into the Delphix Admin application using Delphix Admin credentials
2. Select Manage > Environments.
3. Click Databases.
4. Choose the installation which has the multitenant container database and click the Up icon next to the the installation path to show details.
5. Click “Discover CDB” next to the multitenant container database.
6. Enter the credentials for the multitenant container database and click the Check icon.
7. After pluggable databases are discovered, an Up button appears next the the container database. Click on it to see all discovered pluggable databases.

Related Links

- Requirements for Oracle Source Environments and Databases
- Adding a Database to an Oracle Environment
- Adding a Database Installation Home to an Oracle Environment
Adding a Listener to an Oracle Environment

This topic describes how to add listeners for an Oracle environment.

When you add an environment with the Delphix Admin application, all listeners that are running on it are automatically discovered. However, if a listener is not automatically discovered, you can add it manually to the environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of an environment to view its basic information.
4. Next to Listeners, click the green Plus icon to add a Listener Service.
5. Enter a Name for the new Listener Service, and an IP address for its Endpoint.
6. Click the green Plus icon next to Add Endpoints to enter additional endpoints.
7. Click the Check icon to save your changes.
Changing the Host Name or IP Address for Oracle Source and Target Environments

This topic describes how to change the host name or IP address for source and target environments, and for the Delphix Engine.

- **Procedure**
  - **For Source Environments**
    1. Disable the dSource as described in [Enabling and Disabling dSources](#).
    2. If the **Host Address** field contains an IP address, edit the IP address.
    3. If the **Host Address** field contains a host name, update your Domain Name Server to associate the new IP address to the host name. The Delphix Engine will automatically detect the change within a few minutes.
    4. In the **Environments** screen of the Delphix Engine, refresh the host.
    5. Enable the dSource.

- **For VDB Target Environments**
  1. Disable the VDB as described in [Enabling and Disabling Virtual Databases](#).
  2. If the **Host Address** field contains an IP address, edit the IP address.
  3. If the **Host Address** field contains a host name, update your Domain Name Server to associate the new IP address to the host name. The Delphix Engine will automatically detect the change within a few minutes.
  4. In the **Environments** screen of the Delphix Engine, refresh the host.
  5. Enable the VDB.

- **For the Delphix Engine**
  1. Stop all running VDBs by clicking the red **Stop** button on the VDB card.
  2. Disable all dSources as described in [Enabling and Disabling dSources](#).
  3. You can use either the command line interface or the Server Setup application to change the IP address of the Delphix Engine.
    a. To use the command line interface, press F2 and follow the instructions described in [Setting Up Network Access to the Delphix Engine](#).
    b. To use the Server Setup application, go to **System > Server Setup** in the Delphix Admin interface, or click **Server Setup** in the Delphix Engine login screen.
      i. In the **Network** panel, click **Modify**.
      ii. Under **DNS Services**, enter the new IP address.
      iii. Click **OK**.
  4. Refresh all Environments by clicking the Blue/Green Refresh Symbol on the Environments screen.
  5. Enable all dSources as described in [Enabling and Disabling dSources](#).
  6. Start all VDBs by clicking the **Start** button on the VDB card.
⚠️ **Using Custom init.ora or spfile.ora Files**

If you are using custom *init.ora* or *spfile.ora* files with your Oracle VDBs, you should use the Oracle command line interface (*sqlplus/srvctl*) to shut down any active VDBs and copy the parameter files to a backup location. Complete the steps above, then replace the files and re-start the VDB from the Oracle command line to restore your custom settings. See [Customizing VDB File Mappings](#) for more information about customizing *init.ora* and other configuration files.
Editing Oracle Environment Attributes

This topic describes how to edit attributes of an environment such as name, host address, ssh port, or toolkit path, as well as descriptions of more advanced attributes for specific data platforms.

Procedure
1. Log in to the Delphix Admin application with Delphix Admin credentials or as the owner of an environment.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of an environment to view its attributes.
4. Under Attributes, click the Pencil icon to edit an attribute.
5. Click the Check icon to save your edits.

Common Editable Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment Users</td>
<td>The users for that environment. These are the users who have permission to ssh into an environment, or access the environment through the Delphix Connector. See the Requirements topics for specific data platforms for more information on the environment user requirements.</td>
</tr>
<tr>
<td>Host Address</td>
<td>The IP address for the environment host.</td>
</tr>
<tr>
<td>Notes</td>
<td>Any other information you want to add about the environment.</td>
</tr>
</tbody>
</table>

Oracle Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment Name (RAC)</td>
<td>The Environment Name field under Attributes is used to provide the name of the environment host in the case of cluster environments. This field defaults to the IP address of the host unless you specify another name.</td>
</tr>
<tr>
<td>Cluster User (RAC)</td>
<td>The user who has permission to access the cluster home.</td>
</tr>
<tr>
<td>Virtual IP (RAC)</td>
<td>The IP address that will failover to another node in the cluster when a failure is detected. Click the green + to add another virtual IP domain and IP address.</td>
</tr>
<tr>
<td>Listeners</td>
<td>The listener used to connect incoming client requests to the database. See Adding a Listener to an Oracle Environment for more information.</td>
</tr>
<tr>
<td>SSH Port</td>
<td>The port used for secure shell connection to the host.</td>
</tr>
<tr>
<td>Toolkit Path</td>
<td>The directory used for storing Delphix toolkit files.</td>
</tr>
</tbody>
</table>
| Under the **Show Details** link | • **Remote Listener**: a network name that resolves to an address or address list of Oracle Net remote listeners. Click the green + to add a remote listener.  
• **SCAN**: Single Client Access Name that is used to allow clients to access cluster databases. Click the green + to add a SCAN.  
• **SCAN Listener**: Listener used with SCAN to establish client connections to the database. Click the green + to add a SCAN listener name and endpoints. |


Managing Oracle Environment Users

This topic describes how to manage the users associated with an environment. For information on providing Delphix users with privileges for groups and database objects, see the topics under Managing Delphix Users and Managing Policies.

- **Prerequisites**
- **Procedure**

**Prerequisites**

Users that you add to an environment must meet the requirements for that environment as described in the platform-specific Requirements topics.

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Environments.
4. Click the name of an environment to open the environment information screen.
5. Under Basic Information, click the green Plus icon to add a user.
6. Enter the Username and Password for the OS user in that environment.

### Using Public Key Encryption

If you want to use public key encryption for logging into your environment:

- a. Select Public Key for the Login Type.
- b. Click View Public Key.
- c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
  
  i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
  
  ii. Run chmod 755 ~ to make your home directory writable only by your user.

The public key needs to be added only once per user and per environment.

You can also add public key authentication to an environment user’s profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

7. Click the Check icon to save the new user.
8. To change the primary user for this environment, click the Pencil icon next to Environment Users.
9. To delete a user, click the Trash icon next to their username.
Enabling Linking and Provisioning for Oracle Databases

This topic describes how to enable and disable staging, provisioning and linking for databases.

Before a database can be used as a dSource, you must first make sure that you have defined a staging environment and enabled linking to it. Similarly, before you can provision a VDB to a target database, you must make sure that you have enabled provisioning to it.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Slide the button next to Use as Staging to Yes or No to enable or disable staging.
5. Slide the button next to Allow Provisioning to On or Off to enable or disable provisioning.
6. Slide the button next to Allow Linking to On or Off to enable or disable linking.
Deleting an Oracle Environment

This topic describes how to delete an environment.

Prerequisites

You cannot delete an environment that has any dependencies, such as dSources or VDBs. These must be deleted before you can delete the environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, select the environment you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.
Refresh an Oracle Environment

This topic describes how to refresh an environment.

After you make changes to an environment that you have already set up in the Delphix Admin application, such as installing a new database home, creating a new database, or adding a new listener, you may need to refresh the environment to reflect these changes.

Procedure

1. Log in to the Delphix Admin application with Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of the environment to you want to refresh.
4. Click the Refresh icon.

To refresh all environments, click the Refresh icon next to Environments.
Managing Oracle, Oracle RAC, and Oracle PDB Data Sources

These topics describe specific options and tasks for linking dSources from Oracle and Oracle RAC databases.

- Linking an Oracle Data Source
- Linking an Oracle Pluggable Database
- Advanced Data Management Settings for Oracle dSources
- Using Pre- and Post-Scripts with Oracle dSources
- Enabling Validated Sync for Oracle
- Linking dSources from an Encrypted Oracle Database
- Linking Oracle Physical Standby Databases
- Specifying External Data Directories for Oracle dSources and VDBs
- Linking to Oracle dSources with RMAN Compression or Encryption Enabled
- Upgrading dSources after an Oracle Upgrade
- Enabling and Disabling Oracle dSources
- Detaching and Re-Attaching Oracle dSources
- Deleting an Oracle dSource
- Provisioning from a Replicated Oracle dSource
- Oracle dSource Icon Reference
- Oracle Source Continuity
- Oracle LiveSources
  - Oracle LiveSource User Workflows
Linking an Oracle Data Source

This topic describes the process of linking to a source database and creating a dSource.

Prerequisites

- Make sure you have the correct user credentials for the source environment, as described in Requirements for Oracle Target Hosts and Databases.
- If you are linking a dSource to an Oracle or Oracle RAC physical standby database, you should read the topic Linking Oracle Physical Standby Databases.
- If you are using Oracle Enterprise Edition, you must have Block Change Tracking (BCT) enabled as described in Requirements for Oracle Source Hosts and Databases.
- The source database should be in ARCHIVELOG mode and the NOLOGGING option should be disabled as described in Requirements for Oracle Source Hosts and Databases.
- You may also want to read the topic Advanced Data Management Settings for Oracle dSources.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > Add dSource. Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
3. In the Add dSource wizard, select the source database.
4. Enter your login credentials for the source database and click Verify Credentials. If you are linking a mounted standby, click Advanced and enter non-SYS login credentials as well. Click Next. See the topics under Linking Oracle Physical Standby Databases for more information about how the Delphix Engine uses non-SYS login credentials.
5. Select a Database Group for the dSource, and then click Next. Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users, Permissions, and Policies for more information.
6. Select an Initial Load option. By default, the initial load takes place upon completion of the linking process. Alternatively, you can set the initial load to take place according to the SnapSync policy, for example if you want the initial load to take place when the source database is not in use, or after a set of operations have taken place.
7. Select whether the data in the database is Masked. This setting is a flag to the Delphix Engine that the database data is in a masked state. Selecting this option will not mask the data.
8. Select a SnapSync policy. See Advanced Data Management Settings for Oracle dSources for more information.
9. Click Advanced to edit LogSync, Validated Sync, and Retention policies. See Advanced Data Management Settings for Oracle dSources for more information.
10. Click Next.
11. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs, DB_Link and DB_Sync, to create the dSource. You can monitor.
these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System > Event Viewer**. When the jobs have successfully completed, the database icon will change to a dSource icon on the **Environments > Databases** screen, and the dSource will be added to the list of **My Databases** under its assigned group.

### The dSource Card

After you have created a dSource, you can view information about it on the dSource card, and also make modifications to its policies and permissions. In the **Databases** panel, click on the **Open** icon to view the front of the dSource card. The card will then flip, showing you information such as the **Source Database** and **Data Management** configuration. See the topic [Advanced Data Management Settings for Oracle dSources](#) for more information.

### Related Links

- [Advanced Data Management Settings for Oracle dSources](#)
- [Requirements for Oracle Target Hosts and Databases](#)
- [Linking dSources from an Encrypted Oracle Database](#)
- [Linking Oracle Physical Standby Databases](#)
- [Users, Permissions, and Policies](#)
Linking an Oracle Pluggable Database

This topic describes how to link an Oracle 12c pluggable database to the Delphix Engine to create a dSource.

Prerequisites

- Make sure the Delphix Engine has already discovered the multitenant container database and its pluggable databases. If the container database does not exist in the environment, follow the steps in Adding a Database to an Oracle Environment. If the pluggable database you want to link does not exist in the environment, follow the steps in Discovering Oracle Pluggable Databases in an Oracle Environment.
- You should have Block Change Tracking (BCT) enabled for the container database, as described in Requirements for Oracle Source Hosts and Databases.
- The container database should be in ARCHIVELOG mode and the NOLOGGING option should be disabled, as described in Requirements for Oracle Source Hosts and Databases.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > Add dSource. Alternatively, on the Environment Management screen, you can click Add dSource next to the pluggable database name to start the dSource creation process.
3. In the Add dSource wizard, select the source pluggable database.

   - If the container database is shown but the pluggable database is not, select the container database, enter its database credentials, and click Verify Credentials. The Delphix Engine will discover and list all pluggable databases in the container database. Select the pluggable database from the list.

4. Enter your login credentials for the source database and click Verify Credentials.
5. Click Next.
6. Select a Database Group for the dSource.
7. Click Next.
8. Select an Initial Load option. By default, the initial load takes place upon completion of the linking process. Alternatively, you can set the initial load to take place according to the SnapSync policy. For example, you can set the initial load to take place when the source database is not in use, or after a set of operations have taken place.
9. Select a SnapSync policy. See Advanced Data Management Settings for Oracle dSources for more information.
10. Click Advanced to edit Oracle Sync Options Settings and Retention policies. See Advanced Data Management Settings for Oracle dSources for more information.
11. Click Next.
12. Review the dSource Configuration and Data Management information.
13. Click Finish. The Delphix Engine will initiate two jobs, DB_Link and DB_Sync, to create the dSource. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have completed successfully, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of My Databases under its assigned group.
**Link/Sync of the Multitenant Container Database**

The **DB_Link** job will also link the pluggable database's multitenant container database if it has not been linked yet.

You can also initiate a **DB_Sync** job for the container database.

**Related Links**

- [Adding a Database to an Oracle Environment](#)
- [Discovering Oracle Pluggable Databases in an Oracle Environment](#)
- [Requirements for Oracle Source Hosts and Databases](#)
- [Advanced Data Management Settings for Oracle dSources](#)
Advanced Data Management Settings for Oracle dSources

- **Accessing Data Management Settings**
- **Retention Policies**
- **Oracle Initial Load Options**
- **Oracle SnapSync Policy Settings**
- **Oracle LogSync Policy Settings**
- **Oracle Sync Options Settings**

This topic describes advanced data management settings for dSources.

When linking a dSource, you can use default data management settings for Retention, sync validation, and LogSync policies. However, you can also use custom settings to improve overall performance and match the needs of your specific server and data environment.

**Accessing Data Management Settings**

There are three ways you can set or modify data management settings for dSources:

1. During the dSource linking process, click **Advanced** in the **Data Management** panel of the **Add dSource** wizard.
2. Click the **Edit** icon next to a policy or data management setting on the back of the dSource card. For **SnapSync** and **Retention** policies, click the policy name. This will open the **Policy Management** screen.
3. Select **Manage > Policies** in the top menu bar.

   This will open the **Policy Management** screen. Select the policy for the dSource you want to modify, and click **Modify**. See **Creating Custom Policies** and **Creating Policy Templates**, for more information.

**Retention Policies**

Retention policies determine the length of time that snapshots and log files are retained. The retention time for logs must be equal to or longer than the retention time for snapshots. This policy, in combination with the SnapSync policy, can have a significant impact on the performance of the Delphix Engine.

**Oracle Initial Load Options**

Select whether you want the initial load to take place immediately, or according to the SnapSync schedule. For **Masked**, select **Yes** or **No** to indicate whether the source data is masked.

**Oracle SnapSync Policy Settings**

The SnapSync policy determines how often snapshots are taken of the source database. In the **Default SnapSync** policy, a snapshot is taken daily at 3:30 AM local time, and times out after four hours. If SnapSync does not complete within this four hour period, it will resume at the next scheduled daily time, until the process is complete. Click the **Edit** icon to change the **Default SnapSync** policy, or click the **Add** icon to create a new SnapSync policy. See **Creating Policy Templates** for more information.

**Oracle LogSync Policy Settings**

- **Enabled** - LogSync adds log files from the source database to the dSource, enabling the ability to provision a VDB from a specific point in time or, a database change number (SCN in the case of Oracle databases). LogSync must be enabled for this provisioning functionality to work. For Oracle
dSources, LogSync is disabled if the source database is detected to be mounted but not open.

- **Archive Only, Archive and Online Redo** - these settings determine whether LogSync fetches logs from archive storage in the source database file system, or both the file system and online redo logs. Setting Archive Only permits the use of LogSync with raw devices or Oracle Automatic Storage Management devices. If LogSync detects that the online logs reside on a raw device or ASM storage during the dSource creation process, and is in Archive and Online Redo mode, it will automatically enter into Archive Only mode.

**LogSync for Oracle Standby Databases**

LogSync **must** be enabled for Oracle physical standby databases in **Level Backup** mode and using Real Time Apply mode. See Linking Oracle Physical Standby Databases.

**LogSync for Oracle Pluggable Databases**

LogSync policy settings for Oracle pluggable databases **must** be set at their corresponding container databases.

**Oracle Validated Sync Settings**

Oracle validated sync is disabled by default. When enabled, validated sync is performed immediately after every subsequent SnapSync. See Enabling Validated Sync for Oracle for more information.

**Oracle Sync Options Settings**

- **Compression** - enable compression of backup data sent over the network. Default is **Enabled**.
- **Bandwidth Limit** - select the network bandwidth limit in unit of MB/s between the dSource and the Delphix Engine. Default is **0**, or no bandwidth limit enforced.
- **Number of Connections** - select the number of TCP connections to use between the dSource and the Delphix Engine. Multiple connections may improve network throughput especially over long latency and highly congested networks. Default is **1**.
- **Encrypted Linking** - turn on encryption between the dSource and the Delphix Engine. Default is Disabled.
- **Data Load Channels**
  The channels settings determine the number of channels and data files per backup set. While these settings can be increased, you should consider the potential adverse effects this could have on the throughput of database operations for that server.
  - **Number of Channels** - set the number of RMAN channels used during SnapSync. Default is **2**.
  - **Files per Channel** - maximum number of data files in a backup set. Together with number of channels, it sets the maximum data files concurrently backed up by RMAN. Default is **5**.
- **Block Checking** - select to enable logical block validations during RMAN backup. Default is **Disabled**.
- **Level Backup** - Level Backup mode can be used to improve SnapSync performance with Oracle 11g physical standby databases. See Linking Oracle Physical Standby Databases for more information. By default, SnapSync uses SCN Backup mode.
Using Pre- and Post-Scripts with Oracle dSources

- Contexts for Oracle Pre- and Post-Scripts
- Specifying Arguments for Oracle and Scripts
  - An Example with Three Arguments
  - An Example with an Apostrophe

This topic describes the use of pre- and post-scripts with dSources.

For each script, you can add the script path and its arguments when linking a dSource in one of two ways. Either:

- During the Add dSource wizard process, in the Data Management screen, click Advanced.

or

- On the back of the dSource card, click the Pencil icon next to the Pre Script and Post Script fields.

To update pre- and post-script information:

1. Flip over the dSource card.
2. Click on the Pencil icon next to the Pre Script and Post Script fields.
3. When finished, click the check mark icon.

Note that pre-scripts are executed before the SnapSync policy, and if the script fails, SnapSync will fail as well. In the case of a post-script, the script will execute after SnapSync is complete. If a post-script fails, you will see an error message.

Contexts for Oracle Pre- and Post-Scripts

- For Oracle dSources, each time that the SnapSync policy for a dSource executes, a pre- or post-script can be run as part of the process.
- For Oracle single instance environments, pre- and post-scripts must exist and be readable on the source environment for SnapSync.
- For clustered environments (for example, Oracle RAC), scripts must exist and be readable on all nodes. This allows the Delphix Engine to read the scripts even if cluster nodes are down.

Specifying Arguments for Oracle and Scripts

You can specify multiple arguments for a script. In the Pre or Post Script field, enter the path to the script, and then list the arguments. If the argument contains spaces, enclose it in single or double quotes. You can escape single quotes within the argument with a backslash.

An Example with Three Arguments

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh one "second argument in double quotes" 'third argument in single quotes'
```

An Example with an Apostrophe

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh 'I'd rather be in Hawaii.'
Enabling Validated Sync for Oracle

This topic describes the validated sync process for Oracle databases using both the Delphix Engine Graphic User Interface (GUI) and Command Line Interface (CLI)

Traditional Oracle dSource snapshots require some recovery during provisioning. By configuring validated sync for Oracle, the Delphix Engine will select a compatible Oracle installation and apply the recovery necessary to provision a snapshot immediately after each SnapSync. Snapshots that have been through this validated sync process step do not require recovery during provisioning.

⚠️ The Delphix Engine may be unable to perform validated sync on a physical standby database in Real Time Apply mode. This is because the standby may apply changes before copying the logs that contain those changes. Without the logs necessary to perform recovery, validated sync cannot be executed. However, you can still provision the snapshot when the archive logs become available on the standby.

Prerequisite - Designating a Staging Host

In order to validate an Oracle dSource snapshot after a sync, the Delphix Engine requires a host with an Oracle installation that is compatible with the dSource. This machine is known as the staging host. You must explicitly designate which machines you want the Delphix Engine to use as staging hosts. All machines that have been marked as staging hosts are added to a pool. During sync validation, the Delphix Engine will select a compatible host from the pool, export the requisite archived redo logs and datafiles, and execute Oracle media recovery on the host. Follow these steps to designate a staging host.

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, select the environment you want to designate as staging.
4. Next to Environment Details, select Databases.
5. Scroll down to the installations you want to designate as staging and slide the Use as Staging control to Yes.

To configure validated sync for multiple dSources with different Oracle versions, you must designate a compatible staging source for each. If multiple compatible staging sites exist, the Delphix Engine will select one at random.

⚠️ The validated sync process will consume some resources on the staging host when snapshots are taken. Designating a performance critical host as a staging host is not recommended.

⚠️ The default OS user for the staging host must have access to the Oracle installation that will be used to perform recovery during validated sync.

ℹ️ Oracle Pluggable Database

Validated sync for Oracle pluggable databases is not supported in this release.

Procedure - Enabling Validated Sync

Oracle validated sync can be enabled at link time or on any existing dSource.

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. In the **Databases** panel, select the dSource for which you want to enable sync validation.

4. Move the slider control for **Sync Validation** from **Disabled** to **Enabled**.

**Related Links**

- [CLI Cookbook: Enabling Oracle Validated Sync](#)
Linking dSources from an Encrypted Oracle Database

This topic describes the behavior of the Delphix Engine when linking to a dSource based on an encrypted Oracle database.

Beginning with version 10gR2, Oracle supports the encryption of permanent tablespaces using Transparent Data Encryption (TDE). You can link dSources from databases using TDE by following the basic procedure described in Linking an Oracle Data Source. However, in order to provision a VDB from a dSource that is linked to an encrypted database, you must copy wallet files from the physical database in the source environment to the target environment. See Provisioning a VDB from an Encrypted Oracle Database for more information.

Related Links

- Linking an Oracle Data Source
- Provisioning a VDB from an Encrypted Oracle Database
Linking Oracle Physical Standby Databases

This topic describes special considerations for linking Oracle physical standby databases.

The Delphix Engine supports linking both physical and logical standby databases. In previous versions of the Delphix Engine, limitations were placed upon support for Oracle RAC physical standby databases in Real Time Apply mode. In version 3.0 of the Delphix Engine, these restrictions have been lifted.

Using Block Change Tracking (BCT) on a Physical Standby Database

In general, Delphix recommends enabling Block Change Tracking (BCT) on a primary or standby source database. See Physical Standby Database Support Matrix in this topic for restrictions on enabling BCT on a standby database.

BCT is available from Oracle release 11.1.0.6 onward for physical standby databases only if they are licensed for the Active Data Guard option.

- Release 11.1.0.6 is unstable for the BCT on physical standby feature
- Release 11.1.0.7 requires a patch for Oracle bug 7613481
- Release 11.2.0.2 requires patches for Oracle bugs 10170431, 12312133
- Release 11.2.0.3 requires patches for Oracle bugs 12312133, 16052165

Patches Required

Enabling BCT on a physical standby database without these patches is not recommended because of serious performance and stability issues.

BCT on a primary database has been stable since Oracle version 10.2.0.5.

Physical Standby Database Support Matrix

<table>
<thead>
<tr>
<th>Oracle Version</th>
<th>Apply Mode</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2 or 11.x in Level Backup mode</td>
<td>Archive Apply mode</td>
<td>No special restrictions.</td>
</tr>
<tr>
<td></td>
<td>Real Time Apply mode</td>
<td>LogSync must be enabled.</td>
</tr>
<tr>
<td>11.x in SCN Backup mode</td>
<td>Archive Apply mode</td>
<td>Due to Oracle bug 10146187, Redo Apply must be stopped and the database opened in read-only mode during SnapSync. See the section Stopping and Restarting Redo Apply for more information.</td>
</tr>
</tbody>
</table>
Real Time Apply mode

LogSync must be enabled.

Due to Oracle bug 10146187, Redo Apply must be stopped and the database opened in read-only mode during SnapSync. See the section Stopping and Restarting Redo Apply for more information.

In addition, to avoid Oracle Bug 13075226, which results in a hang during the restart of Redo Apply, Delphix requires disable using BCT on the standby database. The hang occurs when BCT is enabled on a standby database that uses SCN backup mode.

Both Oracle bug 10146187 and 13075226 are fixed starting from Oracle 11.2.0.4. There is no need to configure stop and restart of Redo Apply or disable BCT if the Physical Standby Database is at version 11.2.0.4 or above.

Level Backup Mode for SnapSync

By default, the Delphix Engine’s SnapSync feature uses SCN Backup mode and is designed to not interfere with other backups that may already be in use. However, in cases where RMAN is not being used outside of the Delphix Engine, the Delphix Engine can use the Level Backup mode that improves SnapSync behavior on Oracle 11g physical standby databases. In this mode, redo apply does not have to be stopped during SnapSync. See Advanced Data Management Settings for Oracle dSources for more information about SnapSync settings.

Requirements for Using Level Backup Mode

- No use of RMAN outside of Delphix

OR all of the following:

- Physical standby database running Oracle 11.2.0.2 or later version
- All RMAN backups must use tags
- RMAN CROSSCHECK commands must specify tags
- RMAN DELETE commands must specify tags
- RMAN DUPLICATE commands must specify tags

Failure to meet all of these requirements may cause external RMAN backups to be incomplete or result in corrupt SnapSync snapshots.

Stopping and Restarting Redo Apply
Oracle bug 10146187 requires stopping of redo apply before an SCN-based incremental backup can be issued. These scripts can be used as pre- and post-scripts during the dSource linking process to stop and restart **Redo Apply**. See Using Pre- and Post-Scripts with Oracle dSources for more information about scripts.

- SnapSync pre-script: `stopStandbyApply.sh.template`
- SnapSync post-script: `startStandbyApply.sh.template`

These scripts must be modified for local use, particularly in regard to whether the physical standby database operates in MOUNTED or OPEN mode.

---

Failure to properly customize these scripts could violate your Oracle license terms by running redo apply on an open database, which requires an Oracle Active Data Guard license.

---

**Linking and Provisioning a Mounted Standby**

For databases that are in the **mounted** state, the Delphix database user account must be **SYS** (having the **SYSDBA** role), **SYSBACKUP** (having the **SYSBACKUP** role) or **SYSDG** (having the **SYSDG** role). However, for an **open** standby (Active Data Guard) database, only a regular database user account is required.

Connecting to a **mounted** standby with a **SYS** user account requires that the mounted standby be configured with a password file. Delphix does not capture the password file during SnapSync, and for this reason cannot provision or sync validate a database with a **SYS** user. A secondary, regular database user account can be specified through either the Delphix Admin application or CLI. This database user will then be used to connect to the database during provisioning and validated sync. Note that the **SYS** user is still required to perform snapshots of the source database.

In the Delphix Admin application, the **non-SYS user** can be specified from within the **Add dSource** wizard, or on the back of the Oracle dSource Card after linking.

**Setting the Non-Sys User on the Oracle dSource Card**

1. Log into the Delphix Admin application using **delphix_admin** credentials.
2. In the **Manage** menu, select **Databases > My Databases**.
3. Select the Oracle dSource for which you want to add a **non-SYS user**.
4. Click the dSource's **Expand** icon to open the dSource card, then click the **Flip** icon on the card to view the back.
5. Click the **Edit** button next to **Non-SYS User**.
6. Enter a non-SYS user and credentials that exist on the standby.
7. Click the **Accept** button to save this user and associated credentials.

The non-SYS user will be used to connect to all VDBs provisioned from snapshots of this dSource that are created after the non-SYS user has been set.

**Setting the Non-Sys User with the Command Line Interface**

**SYSBACKUP** and **SYSDG** roles are only available in Oracle 12.1 and later releases.

1. Select the **source config** of the mounted standby.
delphix> sourceconfig select pomme

2. Execute the update command.

```
delphix sourceconfig "pomme" update
```

3. Set the nonSysUser and nonSysCredentials to a non-SYS user that exists on the standby.

```
delphix sourceconfig "pomme" update *> set nonSysUser=<non-sys-username>
delphix sourceconfig "pomme" update *> set nonSysCredentials.type=PasswordCredential
```

```
delphix sourceconfig "pomme" update *> set nonSysCredentials.password=<non-sys-password>
```

4. Commit the operation.

```
delphix sourceconfig "pomme" update *> commit
```

**Video**

**Related Links**

- [Linking an Oracle Data Source](#)
- [Advanced Data Management Settings for Oracle dSources](#)
- [Using Pre- and Post-Scripts with Oracle dSources](#)
Specifying External Data Directories for Oracle dSources and VDBs

This topic describes the process for including external data files with dSource snapshots and VDBs.

In the Advanced section of the Data Management screen during the dSource linking process, and on the back of the dSource card after its created, you can specify the directory for any external data files that should be included with the dSource snapshots.

External File Import for the Delphix Engine and VDBs

The Delphix Engine will not fetch external tables or external data types such as BFILE. Instead, in order to link external data files to the source database and make it available to the Delphix Engine, you must create a directory in the file system and the database. Any data files in the directory you specify will be applied, recursively, to the dSource.

External data will be provisioned to each VDB that is created from this dSource. You will need to update the external file/data type definition to point to the new location after creating VDBs. Provisioning a VDB with external data creates a directory named external in the VDB mount point location.

Configuring the rsync Command Location for an Environment

Files from the external data directory are fetched using the rsync command installed in the source environment. In order to SnapSync a dSource with an external data directory, rsync must be installed in the source environment. If rsync is installed in a non-standard location, the path to the rsync command can be configured in the Environment Details for the source environment on the Environment Management screen.

Example of Attaching and Redirecting External Data Files for Oracle Databases

This example uses two environments:

1. 172.16.200.446 as the source environment
dinosaur as the source database
2. 172.16.200.447 as the target environment
vdino as the target database

Linking a dSource

1. Create an external data directory and an external data file, and attach the directory to the source database.
   a. Log into 172.16.200.446 as the environment user.
   b. Create a physical directory on the source environment.

   $ mkdir /work/extdata
   c. Create a directory in Oracle.

   $ sqlplus / as sysdba
   SQL> create or replace directory extdata as '/work/extdata';

   d. Create a text file /work/extdata/exttab.dat.
CREATE an external table exttab.

```
$ cat > /work/extdata/exttab.dat
1, aaa
2, bbb
3, ccc
^C
```

Query the table.

```
SQL> create table exttab (id number, text varchar2(10))
    2  organization external (default directory extdata
location('exttab.dat'));
```

During the process of linking the dSource to the Dinosaur database, or on the back of the dSource card after creating the link, enter /work/extdata in the External Data Directory field.

**Provisioning a VDB**

1. Provision vdino from Dinosaur.
2. Modify the directory extdata in vdino

   a. Log into the target environment 172.16.200.447
   b. Set SID to vdino

```
$ export ORACLE_SID=vdino
```

c. A query to exttab will fail.

```
$ sqlplus / as sysdba
SQL> select * from exttab
select * from exttab *
ERROR at line 1:
ORA-29913: error in executing ODCIEXTTABLEOPEN callout
ORA-29400: data cartridge error
KUP-04063: unable to open log file EXTTAB_23394.log
OS error No such file or directory
ORA-06512: at "SYS.ORACLE_LOADER", line 19
```

3. Modify directory to the new location.
3. SQL> create or replace directory extdata as '/mnt/provision/vdino/external';

4. Query exttab again.

SQL> select * from exttab;
   ID TEXT
  --- ----
   1 aaa
   2 bbb
   3 ccc

Related Links

- Linking an Oracle Data Source
- Provisioning an Oracle VDB
Linking to Oracle dSources with RMAN Compression or Encryption Enabled

This topic describes the behavior of the Delphix Engine when linking to a dSource with RMAN compression or encryption enabled.

In earlier versions of the Delphix Engine, the dSource linking process would fail if RMAN compression or encryption was enabled. In order for the linking process to complete, the administrator was required to ensure that compression was not enabled for device type SBT_TAPE, and that encryption was also not enabled.

Beginning with Delphix Engine version 2.7, linking a dSource will succeed if compression or encryption is enabled, but the RMAN backup that creates the dSource will not be compressed or encrypted. This is true in the case where the administrator has enabled compression for tape, and in the case where the administrator is using OSB and has enabled encryption for tape.

You can check the RMAN compression and encryption settings with the commands `show device type` and `show encryption for database`, respectively.

Related Links
- [Linking an Oracle Data Source](#)
Upgrading dSources after an Oracle Upgrade

This topic describes how to upgrade dSources and VDBs after an Oracle database upgrade.

Prerequisites

Do not suspend LogSync on the Delphix Engine during an Oracle upgrade of the source environment. LogSync will detect the Oracle version change, and automatically update this information on the Delphix Engine for all the associated dSources and VDBs.

Procedure

1. Refresh all environments.
2. Log into the Delphix Admin application using delphix_admin credentials.
3. Select Manage > Databases > My Databases.
4. For each dSource and VDB, click the Open icon to open its card.
5. On the back of the dSource card, click the blue upward arrow to open the Database Upgrade panel.
6. Select the desired Installation and Environment User.
   If the desired Installation is not in the options menu, click Refresh.
7. Click the check mark to finish the upgrade.
8. Repeat step 3 - 7 for each dSource and VDB that needs to be upgraded.

Updating the Oracle User after an Upgrade

There may be cases when you upgrade the Oracle home and the Oracle User (who owns the binary) is a different user than the previous Oracle User. You will then need to update the Oracle User for each environment, and then re-connect each dSource and VDB to the upgraded Oracle home using the new Oracle User.

The new Oracle User must be in the same OS group (for example, dba or oinstall) as the previous one.

1. Log into the Delphix Admin application using delphix_admin credentials.
2. Select Manage > Environments.
3. Select the environment where you want to add the user.
4. Next to Environment Users, click the Pencil icon to add the new user.
5. Set the new user as the default user.
6. Follow the procedure to upgrade the dSources and VDBs described in this topic.
Enabling and Disabling Oracle dSources

This topic describes how to enable and disable dSources for operations such as backup and restore.

For certain processes, such as backing up and restoring the source database, you may want to temporarily disable your dSource. Disabling a dSource turns off communication between the dSource and the source database, but does not tear down the configuration that enables communication and data updating to take place. When a disabled dSource is later enabled, it will resume communication and incremental data updates from the source database according to the original policies and data management configurations that you set.

Disabling a dSource is also a pre-requisite for several other operations, like database migration and upgrading the dSource after upgrade of the associated data source. See the topics in the Related Links section for more information.

Procedure

1. Go to Manage > Databases > My Databases.
2. Select the dSource to disable.
3. On the back of the dSource card, move the slider control from Enabled to Disabled.
4. Click Yes to acknowledge the warning.

When you are ready to enable the dSource again, move the slider control from Disabled to Enabled, and the dSource will continue to function as it did previously.
Detaching and Re-Attaching Oracle dSources

- Detaching a dSource
- Attaching a dSource

This topic describes how to detach dSources and re-attach them to a different source database.

Each dSource contains an association with the source database, as well as the data it has pulled from the source database up to that point. It is possible to detach, or unlink, a dSource from its source database. This breaks the association with the source database without affecting the data within Delphix. Detached dSources and their source databases have these properties:

- Detached dSources can be used as the source of VDB provisioning operations
- The source database can be re-linked as a different dSource
- The dSource can be attached to a different source database and resume incremental SnapSync, provided the source database is an image of the same logical database (for example, a standby replica). Currently this is only supported through the CLI, and is most often used when performing an initial load from a standby replica and then later attaching the dSource to the production database for subsequent incremental syncs.

Detaching a dSource

1. Log into the Delphix Admin application as a user with OWNER privileges on the dSource, group, or domain.
2. Select Manage > Databases > My Databases.
3. Select the database you want to unlink or delete.
4. Click the Unlink icon.
   A warning message will appear. Click Yes

Attaching a dSource
Rebuilding Source Databases and Using VDBs

In situations where you want to rebuild a source database, you will need to detach the original dSource and create a new one from the rebuilt data source. However, you can still provision VDBs from the detached dSource.

1. Detach the dSource as described in this topic.
2. Re-name the detached dSource by clicking the Edit icon next to its name. This is necessary only if you intend to give the new dSource the same name as the original one. If you attempt to give the new dSource the same name as the detached one, you will see an error message.
3. Create the new dSource from the rebuilt database.

You will now be able to provision VDBs from both the detached dSource and the newly created one, but the detached dSource will only represent the state of the source database prior to being detached.

The attach operation is currently only supported in the CLI. Full GUI support will be added in a future release. Only databases that represent the same physical database can be re-attached

1. Log into the Delphix CLI as a user with OWNER privileges on the dSource, group, or domain.
2. Select the dSource by name using `database select <dSource>`
3. Run the `attachSource` command.
4. Set the source config you want to attach to using `set source.config=<newSource>`. Source configs are named by their database unique name.
5. Set any other source configuration operations as you would for a normal link operation.
6. Run the `commit` command.

Backup Mode for Attaching Oracle dSources

For Oracle dSources, the SnapSync backup option should be set to SCN Backup mode. Level Backup mode is based on information stored in the database control file. If the control file of the newly attached database does not contain information about the previous backups, an initial backup will be created. In addition, Block Change Tracking will not be in sync, and the next SnapSync will need to read the entire database to determine which blocks have changed. See A dvanced Data Management Settings for Oracle dSources for more information about Backup Mode.
Deleting an Oracle dSource

- **Prerequisites**
- **Procedure**

This topic describes how to delete a dSource.

**Prerequisites**
- You cannot delete a dSource that has dependent VDBs. Before deleting a dSource, make sure all dependent VDBs have been deleted as described in [Deleting a VDB](#).

**Procedure**
1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. In the Databases panel, select the dSource you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.

⚠️ Deleting a dSource will also delete all snapshots, logs, and descendant VDB Refresh policies for that database, and the deletion cannot be undone.
**Provisioning from a Replicated Oracle dSource**

This topic describes how to provision from a replicated dSource or virtual database (VDB). The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the **replica** containing the replicated object.

- **Prerequisites**
- **Procedure**
- **Post-Requisites**

**Prerequisites**

- You must have replicated a dSource or a VDB to the target host, as described in [Replication Overview](#).
- You must have added a compatible target environment on the target host

**Procedure**

1. Log into the Delphix Admin application for the target host.
2. Select **Manage > Databases > My Databases**.
3. In the list of replicas, select the **replica** that contains the dSource or VDB you want to provision.
4. The provisioning process is now identical to the process for provisioning standard objects.

**Post-Requisites**

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.
## Oracle dSource Icon Reference

This topic illustrates the icons that appear on dSources and Virtual Databases (VDBs) in the Delphix Engine Graphic User Interface, and describes the meaning of each, along with tips for clearing those that represent errors.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Alert" /></td>
<td>There is a critical fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Warning" /></td>
<td>There is a warning fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Hourglass" /></td>
<td>The Delphix Engine is checking the VDB status.</td>
</tr>
<tr>
<td><img src="image" alt="X" /></td>
<td>The dSource has been deleted or the Source status is UNKNOWN.</td>
</tr>
<tr>
<td><img src="image" alt="Question" /></td>
<td>The state of the VDB is unknown. This is often associated with a connection error.</td>
</tr>
<tr>
<td><img src="image" alt="Stop" /></td>
<td>The VDB is inactive.</td>
</tr>
<tr>
<td><img src="image" alt="Unlock" /></td>
<td>The dSource has been unlinked from the source database.</td>
</tr>
<tr>
<td><img src="image" alt="Shield" /></td>
<td>The VDB is disabled, is in the process of being created, or the creation process has been canceled or failed. See <a href="#">Enabling and Disabling VDBs</a> for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Shield" /></td>
<td>The VDB is running normally</td>
</tr>
<tr>
<td><img src="image" alt="Unlock" /></td>
<td>The dSource is disabled. See <a href="#">Enabling and Disabling dSources</a> for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Shield" /></td>
<td>The dSource or VDB is ready for Linux Transformation.</td>
</tr>
</tbody>
</table>
Oracle Source Continuity

**Oracle Source Continuity Overview**

In earlier versions of the Delphix Engine, when an Oracle database underwent a resetlogs operation, the user was required to re-link the Oracle source. This meant that the Oracle database had to be completely backed up and stored again on the Delphix Engine. If any VDBs were provisioned from the dSource and needed to be saved, the old dSource had to be renamed and saved, resulting in a possible doubling of storage space consumed on the Delphix Engine. The old virtual databases (VDBs) could not be refreshed to the relinked dSource. Beginning with Delphix Engine version 4.1.1.0/4.0.6.0, the Oracle database no longer requires you to re-link sources after a resetlogs operation. The Delphix Engine will detect this condition, automatically take a new full backup, and create a new timeflow for the next SnapSync of the source. Benefits of the Oracle source continuity feature include:

- Lower storage costs and easier administration.
  - Only the changed blocks of the new SnapSync backup will be stored on the Delphix Engine. Because of the way the Delphix Engine handles duplicate blocks, the full backup is likely have a storage requirement similar to an incremental backup.

- Existing VDBs provisioned from previous snapshots for the source will remain.
  - You can use and refresh those VDBs to the new snapshot.

The improved user workflow replaces the old user workflow, which directed users to troubleshoot when SnapSync would fail. Begin Oracle Source Continuity in the following way:

1. The database undergoes a resetlogs operation.
2. If LogSync is enabled, it generates a fault and stops.
3. Start SnapSync. The SnapSync does a full restore of the database to a new timeflow, clears the fault, and restarts LogSync. If you created VDBs prior to the resetlogs operation, they will still exist after the SnapSync; you can refresh them from the new snapshot.

**Creating a New Timeflow**

When LogSync detects the resetlogs operation, it will still stop and generate a fault. LogSync must stop, because a new timeline has been created on the database. This usually happens because the database has been rewound to a past point. The transaction logs being generated on the new timeline are out of sync and conflict with logs from the old timeline. The data files are also out of sync with the data files on the Delphix Engine. You must create a corresponding new timeflow on the Delphix Engine to store the new logs and new versions of the data files. This requires taking a new backup of the database. The following screenshot shows an example of a fault from a resetlogs operation being detected. Note the fault and that LogSync is inactive.
Once LogSync detects the resetlogs operation and throws the fault, no more changes will be retrieved from the database until you start a new SnapSync. This SnapSync will take a full backup, clear the fault, and restart LogSync. Only the new snapshot and timeflow will be visible in the dSource TimeFlow view in the graphical user interface (GUI). Previous snapshots and timeflow will still exist and be visible through the command line interface (CLI) and the Capacity Timeflow view of the GUI. The following screenshot shows the same Delphix Engine after a SnapSync has been performed. Note that the fault has been cleared, LogSync is now active, and only the new snapshot is visible in the GUI.

The following CLI output shows that the old and new timeflow and snapshots are still available. The name of the original timeflow for the database is "default." The name of the new timeflow that was created during the SnapSync is "CLONE@2015-01-15T17:07:20."

delphix> /timeflow list display=name,container

<table>
<thead>
<tr>
<th>NAME</th>
<th>CONTAINER</th>
</tr>
</thead>
</table>

Version 4.2.0
The old snapshots and timeflow will still be subject to logfile and snapshot retention policies. You can also delete the snapshots manually. In addition, you can use the CLI to provision from the old timeflow.
Oracle LiveSources

- **Oracle LiveSources Overview**
  - Understanding Oracle LiveSources
  - Understanding How to Use Oracle LiveSources
  - Oracle LiveSources Quickly Sync with Consistent Snapshots
  - Oracle LiveSources Use Resync and Apply
    - LiveSource Resync is a two-step operation consisting of:

- **Pre-requisites: Configuration and Installation of Staging Environments To Host a Standby Database**

- **Related Links**

Oracle LiveSources Overview

Prior to Delphix Engine version 4.2, users ran reports against virtual databases (VDBs) that they created with the Delphix Engine. Although this workflow helped them offload the reporting load from production, the data in the VDBs was not updated asynchronously. If users wanted newer data, they had to stop their reporting applications, refresh their VDBs, and resume. In the current release, you can run reports against data that is constantly being updated. There is one live data feed per source database that is linked as a dSource on the Delphix Engine. You can point your reporting applications to this live feed. Additionally, you will continue to have all existing Delphix functionality from the dSource, such as creating read/write VDBs.

**Understanding Oracle LiveSources**

Oracle LiveSources leverage native Oracle Active Data Guard technology to keep a standby database up-to-date with changes happening on the source. The standby database is kept open for reads while it applies changes from the source. You can now connect to this standby database for real-time reporting needs. Using Delphix in conjunction with Active Data Guard gives you the ability to get both live up-to-date data and historical points in time from which you can provision virtual databases.

**Understanding How to Use Oracle LiveSources**

**Oracle LiveSources Provide a Read-Only Live Data Stream from Delphix**

You can convert an Oracle dSource to a LiveSource, which is a real-time read-only feed of the linked source. You can access the LiveSource using a JDBC string. Internally, a LiveSource is a standby database instance tracking the Linked Source in real time managed mode and opened in read-only mode.

**Understanding Oracle LiveSources with Data Age and Threshold**

One of the important utilities of a LiveSource is that it provides a real-time feed of the linked Source. In some instances, due to slow networks or other reasons, the LiveSource might fall behind the Linked Source it is tracking. When adding a LiveSource the user can specify a data age threshold. If the LiveSource falls behind the linked source by more than the data age threshold, a fault will be generated and you will be informed.

The Data Age of the LiveSource is displayed on the LiveSource timeflow. A spinning gear, as seen below, indicates that the LiveSource standby database instance is actively receiving data from the source database. Delphix continuously monitors the standby instance and notifies users of any abnormalities.
Users can change the Data Age Threshold at any time by flipping the LiveSource card and updating the threshold value in the card as seen below.

**Oracle LiveSources Quickly Sync with Consistent Snapshots**

Taking snapshots of a LiveSource is instantaneous since the standby database for the LiveSource is constantly receiving data from the source database and recovering it. Taking snapshots occurs instantaneously by taking a filesystem level snapshot of the data on the Delphix Engine without requiring a RMAN backup of the source database. All LiveSource snapshots are consistent; as a result, provisioning from LiveSource snapshots is fast, because no database recovery needs to happen.

**Oracle LiveSources Use Resync and Apply**
Resync is a way to refresh the LiveSource to the current point in the linked source. The following situations require a Resync to be performed:

- There are unresolvable gaps in the log sequence – for example, logs from the source database deleted before the primary database could ship them over to the LiveSource standby.

- The source database was taken through a point in time recovery / flashback, resulting in a changed incarnation.

- The source database contains non-logged changes. In this case, a Resync is needed only if you are interested in moving the non-logged data over to the LiveSource.

- The LiveSource is significantly behind the source database due to network communication issues or large amount of writes.

**LiveSource Resync is a two-step operation consisting of:**

- **Start Resync** – Start Resync performs an incremental backup of the source database to transfer the latest changes to the Delphix Engine. This operation does not affect the availability of the LiveSource.

- **Apply Resync Data** – Applying the Resync data will perform one more incremental backups from the source database to ensure up to date data, and recreate the LiveSource instance while preserving all the configurations. This operation requires downtime for the LiveSource.

If the prepared resync data is no longer needed or resync data has become obsolete (for example, another controlled change has been done on the source database), you can discard the current resync data with **Discard Resync Data**. The next Resync will refetch data from the source database.

**Pre-requisites: Configuration and Installation of Staging Environments To Host a Standby Database**

**Oracle Active Data Guard Required**

The LiveSource feature requires an Active Data Guard license. Delphix uses Active Data Guard to replicate changes from the source database to a standby database that it creates on the staging environment.

**Network Requirements**

LiveSource requires a Data Guard connection between the source and the standby database which utilizes TNS listeners associated with the databases.

**Database Requirements**

LiveSource requires Enterprise Edition of Oracle Database.

<table>
<thead>
<tr>
<th><strong>Oracle Support Matrix</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LiveSource Supports:</strong></td>
</tr>
<tr>
<td>Oracle 11g and non-consolidated Oracle 12c</td>
</tr>
<tr>
<td>SI and RAC source databases</td>
</tr>
<tr>
<td>Physical and standby source databases</td>
</tr>
</tbody>
</table>

**Related Links**

[Oracle LiveSource User Workflows](#)
Oracle LiveSource User Workflows

Please use the following documentation as a guide to identify and act on common Oracle LiveSource User Workflows. The following table of contents includes steps for how to convert a dSource into a LiveSource, provision from a LiveSource, sync a LiveSource, convert a LiveSource back to a dSource, and many other data procedures.

- Converting to LiveSource from a dSource
  - Convert to LiveSource, Section 1 of 6 in the LiveSource Wizard
  - Convert to LiveSource, Section 2 of 6 in the LiveSource Wizard
  - Convert to LiveSource, Section 3 of 6 in the LiveSource Wizard
  - Convert to LiveSource, Section 4 of 6 in the LiveSource Wizard
  - Convert to LiveSource, Section 5 of 6 in the LiveSource Wizard
  - Convert to LiveSource, Section 6 of 6 in the LiveSource Wizard
- Setting up Log Transport between a dSource or Primary Database and a LiveSource or Standby Database
- Removing a LiveSource
- Performing a Snapshot on a LiveSource
- Provisioning from a LiveSource Timeflow
- Enabling, Disabling, and Detaching a LiveSource
- Resyncing a LiveSource + Applying the Resync
- Discarding Resync Data
  - Prerequisites
- Applying Resync Data
  - Prerequisites
  - Procedure
- Migrating a LiveSource
- Upgrading a LiveSource
- Related Links

Converting to LiveSource from a dSource

To get a live feed to the source database data through the Delphix Engine, you must first link the database to the Delphix Engine to create a dSource. You can then convert the dSource into a LiveSource by following the steps outlined below:

1. In the left-hand panel, click the **dSource**.
2. Flip the **dSource card**.
3. Click **Convert to LiveSource**, as highlighted above. This launches the **Convert to LiveSource wizard**.

**Convert to LiveSource, Section 1 of 6 in the LiveSource Wizard**

1. Enter a **DB Unique Name** for the LiveSource.
2. Enter a **Database SID** for the LiveSource.
Convert to LiveSource, Section 2 of 6 in the LiveSource Wizard

Select the environment on which the LiveSource will be created:

1. Select an **Environment User** for the LiveSource instance.
2. Enter the **Mount Point** for the LiveSource instance.
3. Select **Listeners** as needed. If you enable **Auto Select Listeners**, the Delphix Engine will pick the first available listener from the environment.

**Note:** The LiveSource database name must be same as the database name of the primary database; therefore, this value is read-only.
4. Click **Next**.

*Convert to LiveSource, Section 3 of 6 in the LiveSource Wizard*

The image below illustrates where a user is to configure virtual database (VDB) templates and DB configuration parameters.

1. Select **VDB configuration templates** for the LiveSource.
2. Enter additional **DB configuration parameters** for the LiveSource.
3. Click **Next**.

**Convert to LiveSource, Section 4 of 6 in the LiveSource Wizard**

1. The image below illustrates where you will enter the **data age warning threshold** for the LiveSource. If the data in LiveSource lags behind the source database by more than this threshold, the Delphix Engine will raise a fault and notify you.

![Image of Convert To LiveSource window](image)

2. Click **Next**.

**Convert to LiveSource, Section 5 of 6 in the LiveSource Wizard**

1. As seen in the image below, you can enter the **operations** to be performed on initial conversion. These operations are performed after the Delphix Engine has created the standby database for the LiveSource.
2. Click **Next**.

**Note:** These operations will also be performed when resyncing a LiveSource.

*Convert to LiveSource, Section 6 of 6 in the LiveSource Wizard*

1. As seen in the image below, review the configuration summary.

2. Click **Convert** to begin the conversion.
Setting up Log Transport between a dSource or Primary Database and a LiveSource or Standby Database

After adding a LiveSource instance, you must configure the log transport between the dSource or primary database and the LiveSource or standby database. For details on configuring a standby database, refer to the Oracle Data Guard Concepts and Administration guide.

At source/primary database:

1. Configure the LOG_ARCHIVE_CONFIG parameter to enable the sending of redo logs to remote destinations and the receipt of remote redo logs (the LiveSource instance). For example:
   ```sql
   alter system set log_archive_config='DG_CONFIG=(sourcedb,livesource)
   scope=both;
   
   2. Configure the LOG_ARCHIVE_DEST_n parameter to point the redo logs to the LiveSource instance. For example:
   ```sql
   alter system set log_archive_dest_2='SERVICE=livesource ASYNC
   VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE) DB_UNIQUE_NAME=livesource scope=both;
   
   3. Set up `tnsnames.ora` in both source and target sites.

   4. Create a `passwd` file for the LiveSource into the target site.

   5. Configure the corresponding LOG_ARCHIVE_DEST_STATE_n parameter to identify whether the log transport is enabled. For example:
   ```sql
   alter system set log_archive_dest_state_2='ENABLE' scope=both;
   
   6. Configure the STANDBY_FILE_MANAGEMENT parameter to enable automatic standby file management. For example:
   ```sql
   alter system set standby_file_management='AUTO' scope=both;
   
   At the Staging Environment where the LiveSource standby database environment is running:

   1. Configure the FAL_SERVER parameter to point to the primary database for proper fetch archive log function. For example:
   ```sql
   ALTER system SET
   fal_server='service="(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sourcedb.dcenter.delphix.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=sourcedb)(SERVER=DEDICATED))")"';
   
   2. If not already created, configure a password file for Data Guard.

Removing a LiveSource

1. In the left-hand panel, click the LiveSource.

2. Flip the LiveSource card.

3. Click Convert to dSource, as highlighted in the lower right-hand corner of the LiveSource card below:
Performing a Snapshot on a LiveSource

As seen in the image below, you can take a snapshot of a LiveSource by clicking the camera icon on the front of the LiveSource card. LiveSource snapshots are instantaneous, Quick Provision snapshots and don't require an RMAN backup of the source database.

Provisioning from a LiveSource Timeflow
Provisioning from a LiveSource timeflow is the same process as *provisioning from a snapshot for dSource timeflow*. The only difference is that you will select a LiveSource and a LiveSource snapshot.

**Enabling, Disabling, and Detaching a LiveSource**

A LiveSource is **enabled** the same way as a regular dSource.

1. Click **Manage**.
2. Select **Databases**.
3. Select **My Databases**.
4. Select the **LiveSource** to enable.
5. On the back of the **LiveSource card**, move the slider control from **Disabled** to **Enabled**.
6. Click **Yes** to acknowledge the warning.

```
Note: When you enable the LiveSource, the Delphix Engine will recreate the standby database on the staging environment.
```

A LiveSource is **disabled** the same way as a regular dSource.

1. Click **Manage**.
2. Select **Databases**.
3. Select **My Databases**.
4. Select the **LiveSource** to disable.
5. On the back of the **LiveSource card**, move the slider control from **Enabled** to **Disabled**.
6. Click **Yes** to acknowledge the warning.

```
Note: Disabling a LiveSource shuts down the standby database that Delphix manages on the staging environment.
```

You can detach a LiveSource in the same way as **detaching a regular dSource**. Detaching a LiveSource will implicitly convert the LiveSource into a regular dSource. After a dSource is re-attached, you can convert it back into a LiveSource. (Detaching a LiveSource and re-attaching as regular dSource is currently only supported from the CLI)

**Resyncing a LiveSource + Applying the Resync**

Resync is a way to refresh the LiveSource to the current point in the linked source. Resync is a multi-phase operation comprised of the following:

**Perform Resync**

1. Click **Manage**.
2. Select **Databases**.
3. Select **My Databases**.
4. Flip the **LiveSource card**.
5. Click the **Start Resync Data** icon, as highlighted in the image below.
The LiveSource can stay up while the Resync is in progress.

**Discarding Resync Data**

**Prerequisites**
- Resync is started and ready to apply

After Resync has finished, you can choose to not apply but rather discard the data that was brought over from the source database as part of Resync. To discard the data:

1. Click **Manage**.
2. Select **Databases**.
3. Select **My Databases**.
4. Flip the **LiveSource card**.
5. Click the **Discard Resync Data** icon, as highlighted in the image below.

**Applying Resync Data**

**Prerequisites**
- Resync started and ready to apply

**Procedure**
1. Click **Manage**.
2. Select Databases.


4. Flip the LiveSource card.

5. Click the Apply Resync Data, as highlighted in the image below.

If the apply resync data process failed, first investigate and resolve the cause of failure, such as a full disk. Then follow the procedure to start resync.
Migrating a LiveSource

1. Click **Manage**.
2. Select **Databases**.
3. Select **My Databases**.
4. Flip the **LiveSource card**.
5. Disable the **dSource**.
6. Click the **Migrate** icon on the lower right-hand side of the **LiveSource card**, as seen below:
7. Update the environment, user, and repository, as illustrated in the image below:
8. Enable the **dSource**.

**Note:** After the LiveSource is migrated to a different staging environment, you must ensure that the log transport between the source database and the LiveSource instance on the new staging environment is set up correctly.

### Upgrading a LiveSource

If the source database for the LiveSource has been upgraded, users would have to inform Delphix of the updated Oracle installation and the associated environment user for both the source database and the LiveSource. This can be done by following the steps below:

1. Click **Manage**.
2. Select **Databases**.
3. Select **My Databases**.
4. Flip the **Live Source card**.
5. Disable the **LiveSource**.
6. On the back of the **LiveSource card**, click the **upgrade** icon in the lower right-hand corner, as highlighted in the image below.

![Image of LiveSource card and upgrade icon](image)

7. Specify the **new installation** and **environment user** for the Linked Source and the LiveSource, as illustrated in the image below.
8. Enable the **LiveSource**.

**Related Links**

[Oracle LiveSources Overview](#)
Provisioning VDBs from Oracle, Oracle RAC, and Oracle PDB Sources

These topics describe concepts and tasks for provisioning a VDB from an Oracle, Oracle RAC or Oracle PDB Source.

- **Provisioning Oracle VDBs: An Overview**
- **Provisioning an Oracle VDB**
- **Provisioning an Oracle Virtual Pluggable Database**
- **Customizing Oracle VDB Configuration Settings** — This topic describes how to customize Oracle and Oracle RAC virtual database (VDB) configuration settings, including settings that the Delphix Engine reserves and those that it removes from the database configuration file during the provisioning process.
- **Customizing Oracle VDB Management with Hook Operations**
- **Customizing VDB File Mappings**
- **Provisioning a VDB from an Encrypted Oracle Database**
- **Time Flows for RAC Provisioning of VDBs**
  - **TimeFlow Patching**
- **Provisioning from a Replicated Oracle VDB**
- **Enabling and Disabling an Oracle VDB**
- **Rewinding an Oracle VDB**
- **Refreshing an Oracle VDB**
- **Deleting an Oracle VDB**
- **Migrating an Oracle VDB**
- **Upgrading an Oracle VDB**
- **Oracle VDB Icon Reference**
- **Migrate a vPDB**
Provisioning Oracle VDBs: An Overview

This topic describes basic concepts behind the Oracle virtual database (VDB) provisioning process.

A dSource is a virtualized representation of a physical or logical source database. As a virtual representation, it cannot be accessed or manipulated using database tools. Instead, you must create a Virtual Database (VDB) from a dSource snapshot. A VDB is an independent, writeable copy of a dSource snapshot. You can also create VDBs from other VDBs. Once a VDB has been provisioned to a target environment, you can also implement a snapshot policy for that VDB, to capture changes within it as if it were any other logical or physical database.

Working with Snapshots and LogSync

When you first create a dSource from a physical or logical database, SnapSync takes an initial snapshot of the database; it then continues to take snapshots based on your SnapSync policy settings. If LogSync is also enabled, you can use log files in conjunction with the snapshot to provision a VDB that represents the exact state of the source database at a point in time.

Snapshots accumulate over time. To view them:

1. In the Delphix Admin application, click My Databases.
2. Select a dSource.

Each snapshot is represented as a card that includes information about the source database, operating system, end stamp, and snapshot SCN. You can scroll through these cards to select the one you want, or you can enter a date and time to search for a specific snapshot.

If LogSync is enabled, you will also see a LogSync slider control at the top of the snapshot card. If you slide this control to the right to open LogSync, you will see a pointer that you can move along a timeline to select the exact time from which you want to provision a VDB.

Once a VDB has been provisioned, you can also take snapshots of it, either manually or through a VDB Snapshot policy. You can then provision additional VDBs from these VDB snapshots. To view VDB snapshots:

1. In the Delphix Admin application, click My Databases.
2. Select the VDB.

Target Environments for VDBs

It is possible to provision a VDB to the same source environment that contains the dSource, but performance and efficiency improve if the dSource and VDB are located in different environments. For example, if you attempt to provision an encrypted database to its source environment, the provisioning will fail because of two instances attempting to access the same wallet file.

The source and target environments must have the same operating system and database types. For example, if the source environment is using Oracle Enterprise Linux 6.0 as the operating system and is running Oracle 10gR2 as its DBMS, then the target environment must have this same configuration. There are also specific requirements for the target environment user. The supported operating systems for source and target environments are described in Supported Operating Systems and DBMS Versions for Oracle Environments. User requirements are described in Requirements for Oracle Target Hosts and Databases.

Customizing VDB Configuration Settings and File Paths

When you provision a VDB, you have the option of customizing its configuration settings and the file paths that it will use on the target environment. To view the default configuration settings and file paths during the provisioning
process, click Advanced in the Target Environment screen of the Provision VDB wizard. For more information, see Customizing Oracle VDB Configuration Settings and Customizing VDB File Mappings.

Repository Templates

Repository templates are a new feature introduced in the Fhloston release. The primary use case and motivation for this new capability is to provide the Delphix administrator with control over the Oracle database parameters used during the staging phase of the VDB provisioning process. It is useful to be able to control these configuration parameters when the physical capabilities of the staging machine, such as CPU count and memory, are inferior to the physical capabilities of the machines hosting the source database repository.

The repository template is a three-way relationship between three entities:

- A database repository – The entity that contains database instances on host environments
- A database container – An entity that represents all of the physical data associated with the database
- A VDB configuration template – A list of database configuration parameter names and values that you can and save on the Delphix Engine to be used at a later time

During the staging process, if a repository template is not specified, then by default the Delphix Engine will use the configuration parameters taken from the source database to configure the staged database. These parameters may not be appropriate, because the machine used for staging may be physically inferior to the machine hosting the source database.

Instead, the Delphix admin can create a VDB configuration template, which would be appropriate for the physical machine hosting staging repository. (See Create VDB Config Template.) Then the admin can create a repository template entry which will bind together the VDB configuration template, database repository, and database container. This instructs the Delphix Engine to use configuration parameters from the VDB configuration template whenever the database container is staged on the database repository specified, instead of the parameters on the source database.

Currently, repository template relations can only be created via the command line interface (CLI) in repository->template.

1. Switch to the repository->template context and create a new template entry.

```bash
    delphix> repository template
    delphix> create
    delphix repository template create *> set name=RepositoryTemplate1
    delphix repository template create *> set container=DBContainer1
    delphix repository template create *> set repository=DBRepository1
    delphix repository template create *> set template=DBTemplate1
    delphix repository template create *> commit
```

Related Links

- Supported Operating Systems and DBMS Versions for Oracle Environments
• Requirements for Oracle Target Hosts and Databases
• Customizing Oracle VDB Configuration Settings
• Customizing VDB File Mappings
Provisioning an Oracle VDB

This topic describes how to provision a virtual database (VDB) from a dSource or another VDB.

Prerequisites

- You will need to have linked a dSource from a source database, as described in Linking an Oracle Data Source, or have already created a VDB from which you want to provision another VDB.
- You will need to have the correct OS User privileges on the target environment, as described in Requirements for Oracle Target Hosts and Databases.
- If you want to use customized database configuration settings, first create a VDB Config Template as described in Customizing Oracle VDB Configuration Settings.
- If you are creating a VDB from a dSource linked to an encrypted database, make sure you have copied the wallet file to the target environment as described in Provisioning a VDB from an Encrypted Oracle Database.

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a dSource snapshot.
   See Provisioning by Snapshot and LogSync in this topic for more information on provisioning options.

5. Optional: Slide the LogSync slider to the open the snapshot timeline, and then move the arrow along the timeline to provision from a point of time within a snapshot.

6. Click Provision.
   The Provision VDB panel will open, and the fields Installation Home, Database Unique Name, SID, Database Name, Mount Base, and Environment User will auto-populate with information from the dSource.

7. If you need to add a new target environment for the VDB, click the green Plus icon next to the Filter Target field, and follow the instructions in Adding an Oracle Single Instance or RAC Environment.
8. Review the information for Installation Home, Database Unique Name, SID, and Database Name and edit as necessary.
9. Review the Mount Base and Environment User and edit as necessary.
   The Environment User must have permissions to write to the specified Mount Base, as described in Requirements for Oracle Target Hosts and Databases. You may also want to create a new writeable directory in
the target environment with the correct permissions, and use that as the Mount Base for the VDB.

10. Select **Provide Privileged Credentials** if you want to use login credentials on the target environment other than those associated with the **Environment User**.

11. Click **Advanced** to select Oracle Node Listeners or enter any VDB configuration settings or file mappings. See [Customizing Oracle VDB Configuration Settings](#) and [Customizing VDB File Mappings](#) for more information.

   ![If you are provisioning to a target environment that is running a Linux OS, you will need to compare the SGA_TARGET configuration parameter with the shared memory size in /dev/shm. The shared memory configured on the target host should match the SGA memory target. You can check this by opening the Advanced settings, and then finding the value for SGA_TARGET under DB Configuration.](#)

12. Click **Next**.

13. Select a **Target Group** for the VDB.

   Click the green **Plus** icon to add a new group, if necessary.

14. Select a **Snapshot Policy** for the VDB.

   Click the green **Plus** icon to create a new policy, if necessary.

15. Click **Next**.

16. Enter any operations that should be run at **Hooks** during the provisioning process.

   See [Customizing Oracle Management with Hook Operations](#) for more information.

17. Click **Next**.

18. Click **Finish**.

   When provisioning starts, you can review progress of the job in the Databases panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the VDB will be included in the group you designated, and listed in the **Databases** panel. If you select the VDB in the Databases panel and click the **Open** icon, you can view its card, which contains information about the database and its Data Management settings.

### Provisioning by Snapshot or LogSync

When provisioning by snapshot, you can provision to the start of any particular snapshot, either by time or SCN.

<table>
<thead>
<tr>
<th>Provisioning By Snapshot</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by Time</td>
<td>You can provision to the start of any snapshot by selecting that snapshot card from the Timeflow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.</td>
</tr>
<tr>
<td>Provision by SCN</td>
<td>You can use the <strong>Slide to Provision by SCN</strong> control to open the SCN entry field. Here, you can type or paste in the SCN you want to provision to. After entering a value, it will “snap” to the start of the closest appropriate snapshot.</td>
</tr>
</tbody>
</table>

When provisioning by LogSync information, you can provision to any point in time, or to any SCN, within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an
individual snapshot, use the **Slide to Open LogSync** control at the top of an individual snapshot card.

<table>
<thead>
<tr>
<th>Provisioning By LogSync</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by SCN</td>
<td>Use the <strong>Slide to Open LogSync</strong> and <strong>Slide to Provision by SCN</strong> controls to view the range of SCNs within that snapshot. Drag the red triangle to the LSN that you want to provision from. You can also type or paste in the specific SCN you want to provision to. Note that if the SCN doesn't exist, you will see an error when you provision.</td>
</tr>
<tr>
<td>Provision by Time</td>
<td>Use the <strong>Slide to Open LogSync</strong> control to view the time range within that snapshot. Drag the red triangle to the point in time that you want to provision from. You can also enter a date and time directly.</td>
</tr>
</tbody>
</table>

**Video**

![Get Adobe Flash Player](GetAdobeFlashPlayer.png)

**Related Links**

- [Linking an Oracle Data Source](Linking an Oracle Data Source)
- [Requirements for Oracle Target Hosts and Databases](Requirements for Oracle Target Hosts and Databases)
- [Customizing Oracle VDB Configuration Settings](Customizing Oracle VDB Configuration Settings)
- [Customizing VDB File Mappings](Customizing VDB File Mappings)
- [Customizing Oracle Management with Hook Operations](Customizing Oracle Management with Hook Operations)
Provisioning an Oracle Virtual Pluggable Database

This topic describes how to provision a virtual pluggable database (VPDB) from an Oracle pluggable database source (dSource).

Prerequisites

- You must have done one of the following:
  - linked a PDB dSource from a multitenant container database, as described in Linking an Oracle Pluggable Database
  - already created a VPDB from which you want to provision another VPDB
- There must be a target environment that has a compatible multitenant container database to host the VPDB you are about to create
  - You will need to have the correct operating system (OS) user privileges on this target environment. For more information, refer to Requirements for Oracle Target Hosts and Databases.
- The multitenant container databases (CDBs) of the source PDB and the target that will host the VPDB must meet the following requirements:
  - They must have the same endian format
  - They must be in ARCHIVELOG mode
  - They must have compatible character sets and national character sets, which means:
    - Every character in the source CDB character set is available in the target CDB character set
    - Every character in the source CDB character set has the same code point value in the target CDB character set
  - They must have the same set of database options installed. For example, if the source CDB is a real application cluster (RAC) database, the target CDB must be a RAC database.

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a PDB dSource or a previously provisioned VPDB.
4. Select a snapshot.
   For more information on provisioning options, see the Provisioning by Snapshot or LogSync section in Provisioning an Oracle VDB.

You can take a snapshot of the source database to provision from by clicking the Camera icon on the source card.

5. Optional: Slide the LogSync slider to open the snapshot timeline, and then move the arrow along the timeline to provision from a point of time within a snapshot.
6. Click **Provision**.
   The **Provision VDB** panel will open, and the provision target fields **Installation Home**, **Container Database**, **Database Name**, **Mount Base**, and **Environment User** will auto-populate. Information from the selected target environment will be highlighted on the left hand pane.

7. For each selected **Installation Home**, there can be more than one **Container Database**. Use the drop down box to further select which **Container Database** you are about to provision to host your VPDB.

8. Review the information for **Installation Home**, **Container Database**, and **Database Name**. Change or edit as necessary.

9. Review the **Mount Base** and **Environment User** and edit as necessary.
   The **Environment User** must have permissions to write to the specified Mount Base, as described in [Requirements for Oracle Target Environments and Databases](#). You may also want to create a new writeable directory in the target environment with the correct permissions, and use that as the Mount Base for the VDB.

10. Select **Provide Privileged Credentials** if you want to use login credentials on the target environment other than those associated with the **Environment User**.

11. Click **Advanced** to enter any **file mappings** setting for your VPDB.
   See [Customizing VDB File Mappings](#) for more information.

12. Click **Next**.

13. Enter the **VDB Name** for the VDB you are about to provision.

14. Select a **Target Group** for the VDB.
    Click the green **Plus** icon to add a new group, if necessary.

15. Select a **Snapshot Policy** for the VDB.
    Click the green **Plus** icon to create a new policy, if necessary.

16. Click **Next**.

17. Enter any operations that should be run at **Hooks** during the provisioning process.
    See [Customizing Oracle Management with Hook Operations](#) for more information.

18. Click **Next**.

19. Review the provision summary. Click **Finish** to proceed with provisioning the VPDB. When provisioning starts, you can review progress of the job in the **Databases** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the VPDB will be included in the group you designated, and listed in the **Databases** panel. If you select the VPDB in the **Databases** panel and click the **Open** icon, you can view its card, which contains information about the database and its Data Management settings.

   ![Tip]
   The container database of the VPDB will be automatically linked if it has not been linked already.

   ![Tip]
   **Temporary CDB Instance**
   During VPDB provisioning, a temporary CDB instance is created in the target environment to recover the VPDB to a consistent state. This temporary CDB will be automatically deleted after the VPDB is provisioned successfully.
Related Links

- Linking an Oracle Pluggable Database
- Provision an Oracle VDB
- Discovering Oracle Pluggable Databases in an Oracle Environment
- Requirements for Oracle Target Hosts and Databases
- Customizing VDB File Mappings
- Migrate a vPDB
- Customizing Oracle Management with Hook Operations
Customizing Oracle VDB Configuration Settings

This topic describes how to customize Oracle and Oracle RAC virtual database (VDB) configuration settings, including settings that the Delphix Engine reserves and those that it removes from the database configuration file during the provisioning process.

When you create a VDB, the Delphix Engine copies configuration settings from the dSource and uses them to create the VDB. Most settings are copied directly, and you can see these settings by clicking the Advanced link in the Target Environment screen in the Provision VDB wizard.

When a VDB is provisioned in the browser, you can specify configuration parameters directly or copy them from a VDB Configuration Template. Once set, the Delphix Engine will use these parameters whenever the VDB is refreshed, even if you change the original template. With the command line interface (CLI), you can store a reference to the template with the VDB instead of copying the values. When using a template reference, subsequent refreshes will use the configuration settings from the template. In this case, changes to a template will appear in any VDBs when they are refreshed. If you delete a template while VDBs still refer to it, the template property will be unset for those VDBs and the current settings will be used when refreshed in the future.

It is important to know, however, that some configuration parameters cannot be customized. In addition, some configuration parameters are stripped out during the provisioning process but are customizable. The list of restricted and customizable parameters can be found below.

Creating a VDB Config Template

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > VDB Config Templates to open the VDB Configuration Template Manager.
3. Click the green Plus icon to add a parameter.
4. Enter the name and value for the parameter.
5. Repeat steps 3 and 4 for each parameter you want to add to the template.
6. When you are done adding parameters, click Save Template.

Building Templates from Other Templates

If you have already created a VDB Config Template, you can use it as the basis for creating other templates.

1. Open the VDB Configuration Template Manager, and then select a template.
2. Click Create New Template.
3. Enter a name for the template.
4. Click OK.
   A new template is created with the parameters from the selected template. You can now edit or add parameters as necessary.
5. Click Save Template.

Applying a VDB Config Template
You can apply a VDB Config Template to a VDB during the provisioning process. As described in the introduction, this copies the values from the template. Any subsequent changes to the template will not appear in the VDB, even when that VDB is refreshed.

1. In the **Target Environment** screen of the **Provision VDB** wizard, click **Advanced**.
2. Select **Use Template**.
3. Select a **template**.
4. Click **Next**.

**Dynamically Template Parameters on Refresh (CLI Only)**

In the CLI, you can set a reference to the template instead of copying the parameters. Subsequent refreshes will use the values from the template, even if those values have changed. Setting a template reference causes the Delphix Engine to ignore any manually-set parameters.

1. Select the **target source**.

   ```
   delphix> source select vexample
   ```

2. Update the source.

   ```
   delphix source "vexample"> update
   ```

3. Set the **configTemplate** property using the name of your desired template.

   ```
   delphix source "vexample" update *> set configTemplate=ExampleTemplate
   ```

4. Commit the result.

   ```
   delphix source "vexample" update *> commit
   ```

You can also set the template reference during provisioning. See the [CLI Cookbook: Provisioning a Single Instance Oracle VDB](#) topic for more information.

**Video**

![Get ADOBE FLASH PLAYER](#)

**Restricted Parameters**

These parameters are restricted for use by the Delphix Engine. Attempting to customize these parameters through the use of a VDB Config Template will cause an error during the provisioning process.

- `active_instance_count`
- `cluster_database`
- cluster_database_instances
- cluster_interconnects
- control_files
- db_block_size
- db_create_file_dest
- db_create_online_log_dest_1
- db_create_online_log_dest_2
- db_create_online_log_dest_3
- db_create_online_log_dest_4
- db_create_online_log_dest_5
- db_file_name_convert
- db_name
- db_recovery_file_dest
- db_recovery_file_dest_size
- db_unique_name
- dg_broker_config_file1
- dg_broker_config_file2
- dg_broker_start
- fal_client
- fal_server
- instance_name
- instance_number
- local_listener
- log_archive_config
- log_archive_dest
- log_archive_duplex_dest
- log_file_name_convert
- spfile
- standby_archive_dest
- standby_file_management
- thread
- undo_tablespace
- __db_cache_size
- __java_pool_size
- __large_pool_size
- __oracle_base
- __pga_aggregate
- __sga_target
- __shared_io_pool_size
- __shared_pool_size
- __streams_pool_size

Customizable Parameters

The default value for these parameters is cleared during the provisioning process. They are removed from the VDB configuration file unless you set values for them through a VDB Config Template.
• audit_file_dest
• audit_sys_operations
• audit_trail
• background_dump_dest
• core_dump_dest
• db_domain
• diagnostic_dest
• dispatchers
• fast_start_mttr_target
• log_archive_dest_1
• log_archive_dest_2
• log_archive_dest_3
• log_archive_dest_4
• log_archive_dest_5
• log_archive_dest_6
• log_archive_dest_7
• log_archive_dest_8
• log_archive_dest_9
• log_archive_dest_10
• log_archive_dest_11
• log_archive_dest_12
• log_archive_dest_13
• log_archive_dest_14
• log_archive_dest_15
• log_archive_dest_16
• log_archive_dest_17
• log_archive_dest_18
• log_archive_dest_19
• log_archive_dest_20
• log_archive_dest_21
• log_archive_dest_22
• log_archive_dest_23
• log_archive_dest_24
• log_archive_dest_25
• log_archive_dest_26
• log_archive_dest_27
• log_archive_dest_28
• log_archive_dest_29
• log_archive_dest_30
• log_archive_dest_31
• log_archive_dest_state_1
• log_archive_dest_state_2
• log_archive_dest_state_3
• log_archive_dest_state_4
• log_archive_dest_state_5
- log_archive_dest_state_6
- log_archive_dest_state_7
- log_archive_dest_state_8
- log_archive_dest_state_9
- log_archive_dest_state_10
- log_archive_dest_state_11
- log_archive_dest_state_12
- log_archive_dest_state_13
- log_archive_dest_state_14
- log_archive_dest_state_15
- log_archive_dest_state_16
- log_archive_dest_state_17
- log_archive_dest_state_18
- log_archive_dest_state_19
- log_archive_dest_state_20
- log_archive_dest_state_21
- log_archive_dest_state_22
- log_archive_dest_state_23
- log_archive_dest_state_24
- log_archive_dest_state_25
- log_archive_dest_state_26
- log_archive_dest_state_27
- log_archive_dest_state_28
- log_archive_dest_state_29
- log_archive_dest_state_30
- log_archive_dest_state_31
- remote_listener
- user_dump_dest
Customizing Oracle VDB Management with Hook Operations

Hook operations allow you to execute an ordered list of custom operations as part of the dSource syncing, provisioning, and refresh process. For details on the types of operations that are available, see children of this page.

**dSource Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Sync</td>
<td>Operations performed before a sync. These operations can quiesce data to capture more information into the sync or stop applications that may interfere with the sync.</td>
</tr>
<tr>
<td>Post-Sync</td>
<td>Operations performed after a sync. These operations can undo any quiescing of data or undo any changes made by the Pre-Sync hook.</td>
</tr>
</tbody>
</table>

**Virtual Dataset Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Clone</td>
<td>Operations performed after initial provision or refresh of a virtual dataset. This hook runs when the virtual dataset has been started. During a refresh, the Delphix Engine reaches this hook before the Post-Refresh hook.</td>
</tr>
<tr>
<td>Pre-Refresh</td>
<td>Operations performed before a refresh. These operations can back up any data or configurations from the running source before doing the refresh.</td>
</tr>
<tr>
<td>Post-Refresh</td>
<td>Operations performed after a refresh. You can use these operations to restore any data or configurations backed up by the Pre-Refresh hook. During a refresh, the Delphix Engine runs this after the Configure Clone hook.</td>
</tr>
</tbody>
</table>

**Operation Failure**

If a hook operation fails, it will fail the entire sync, provision, or refresh job. The job will halt immediately and no further hook operations will be run.

**Hook Operations on Cluster Environments**

When linking from, or provisioning to, cluster environments such as Oracle RAC environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine picks a node in the cluster at random and guarantees all operation within any single hook will execute serially on this node. Note that the Delphix Engine does not guarantee the same node is chosen for the execution of every hook.

**Setting Hook Operations**

You can construct hook operation lists through the Delphix Admin application or the command line interface (CLI). You can either define the operation lists as part of the linking or provisioning process or edit them on dSources or
virtual datasets that already exist.

Setting Hook Operations through the Delphix Admin Application

To specify hook operations during linking or provisioning, navigate to the **Hooks** tab of the **Linking Wizard** or **Provision Wizard**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Select the **type of operation** or click **Import** to load a hook operation template.
4. Click the **text area** and edit the contents of the operation.
5. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
6. To remove an operation from the list, click the **Trash** icon on the operation.
7. When you have set all hook operations, click **Next** to continue with the provisioning process.

To edit hook operations on an already-existing dSource or virtual dataset, navigate to the **Hooks** tab on the back of the dSource card or virtual dataset card.

1. Select the **hook** to edit.
2. The current operations at this hook will be displayed. To edit this list of operations, click the **Pencil** icon in the top right-hand corner of the card.
3. Click the **Plus** icon to add a new operation.
4. Select the **type of operation** or click **Import** to load a hook operation template.
5. Click the **text area** and edit the contents of the operation.
6. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
7. To remove an operation from the list, click the **Trash** icon on the operation.
8. When you have set all hook operations, click **Check** to save the changes.

Setting Hook Operations through the CLI

To specify hook operations during linking, edit the relevant hook's array of operations defined on the **LinkingParameters > Source > Operations** object.

To specify hook operations during provisioning, edit the relevant hook's array of operations defined on the **ProvisionParameters > Source > Operations** object.

To edit hook operations on an already-created dSource, edit the relevant hook's array of operations defined on the **Source > Operations** object.

To edit hook operations on an already-created virtual dataset, edit the relevant hook's array of operations defined on the **Source > Operations** object.

For more information about these CLI objects, see the LinkedSourceOperations, VirtualSourceOperations, RunCommandOnSourceOperation, and RunExpectOnSourceOperation API documentation in the **Help** menu of the Delphix Admin application.

Example of Editing Hook Operations through the CLI

1. Navigate to relevant source’s **VirtualSourceOperations** object.
2. Select a **hook** to edit.
3. Add an operation at index 0.

```bash
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 0 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 0 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 0 *> ls
```

Properties
- type: RunCommandOperation (*)
- command: echo Refresh completed. (*)

```bash
delphix source "pomme" update operations postRefresh 0 *> commit
```

4. Add another operation at index 1 and then delete it.

```bash
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 1 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 1 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 1 *> back
delphix source "pomme" update operations postRefresh *> unset 1
delphix source "pomme" update operations postRefresh *> commit
```

**Hook Operation Templates**

You can use templates to store commonly-used operations, which allows you to avoid repeated work when an operation is applicable to more than a single dSource or virtual dataset. You manage templates through the Delphix Admin application.

**Creating a Hook Operation Template**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Operation Templates.
4. Click the Plus icon to add a new operation template.
5. Enter a Name for the template.
6. Select an operation Type.
7. Enter a Description detailing what the operation does or how to use it.
8. Enter operation Contents to implement the operation partially or fully.
9. Click Create.

**Importing a Hook Operation Template**
To import a hook operation template, navigate to the **Hooks** tab on the back of the **dSource** or **virtual dataset card**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Click **Import**.
4. Select the **template** to import.
5. Click **Import**.
6. When you have set all hook operations, click **Check** to save the changes.

**Exporting a Hook Operation Template**

To export a hook operation template, navigate to the **Hooks** tab on the back of the **dSource** or **virtual dataset card**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Select the **type of operation**.
4. Click the **text area** and edit the contents of the operation.
5. Click **Export**.
6. Enter a **Name** for the template.
7. Enter a **Description** detailing what the operation does or how to use it.
8. Click **Create**.
Customizing VDB File Mappings

This topic describes how to customize file path mappings when provisioning a virtual database (VDB).

In the VDB provisioning process, it may be necessary to create mappings between files and directories that exist on the source, and files or file directories that exist on the target. An example of this is creating a copy in the target environment of a wallet file for an encrypted database that exists in the source environment.

Pattern Matching Example

You can use pattern matching rules to create full path names for data files and control files.

Pattern matching rules have the form `source-regex-expression-KEY : target-replacement-VALUE`. You can use multiple rules, which are applied successively. Multiple rules with the same source key are allowed.

In this example, several rules are applied to the source file path `/app/oracle/oradata/system01.dbf`.

1. Applying the rule `ora:foo` results in: `/app/foocle/foodata/system01.dbf`
2. Applying the rule `foo:bar` results in: `/app/barcle/bardata/system01.dbf`
3. Applying the rule `ora:no` results in an error, because `ora` is no longer found in the pathname.
4. Applying the rule `bar:oranew` results in: `/app/oranewcle/oranewdata/system01.dbf`
5. Applying the rule `ora:yes` results in `/app/yesnewcle/yesnewdata/system01.dbf`

During the pattern matching process, two errors can be generated.

1. **No match for specified mapping rules**
   This is the result when no rules match a source file

2. **Invalid regex pattern specified for path mapping**
   This is the result of an invalid regex rule mapping

*This topic on the java.regex.util class*, hosted on docs.oracle.com, shows the regular expression syntax and constructs recognized by the Delphix Engine pattern-matching operations.

Applying VDB File Mappings During the Provisioning Process

1. In the Target Environment screen of the Provision VDB wizard, click Advanced.
2. Click the green Plus icon to add a mapping rule.
3. Enter the mapping rule.
4. Click Validate to see the results of applying the rule. If not matches are found, you will see an error message.
5. Click Next to continue with the provisioning process.
Provisioning a VDB from an Encrypted Oracle Database

This topic describes how to provision a VDB from an encrypted database.

The Delphix Engine supports provisioning from a dSource linked to a physical database that has been encrypted with Oracle’s Transparent Database Encryption (TDE), which can be used to encrypt columns or tablespaces.

Provisioning a VDB from an encrypted dSource requires that you have an auto-open wallet set up in the target environment, because the provisioning process requires the master key stored in the wallet file. You can either set up an auto-open wallet for the source database and copy the auto-open wallet files (both ewallet.p12 and cwallet.sso) to the target environment, or first copy the encryption wallet file (ewallet.p12) to the target environment, and then generate an auto-open wallet from the encryption wallet file.

When provisioning a VDB from an encrypted dSource, if the target environment has other databases that also use TDE, each database should use a different wallet. Please check Oracle documentation on how to set up different wallet locations for different databases. For example, you can use $ORACLE_SID in the DIRECTORY clause of the ENCRYPTION_WALLET_LOCATION parameter in sqlnet.ora:

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/opt/oracle/oradata/nf/wallet/$ORACLE_SID)))
```

Procedure

1. Check for any encrypted columns or tablespaces on the source database by using these commands:

   ```
   SELECT t.name name, e.encryptionalg algorithm FROM v$tablespace t, v$encrypted_tablespaces e 
   WHERE t.ts# = e.ts# and upper(e.encryptedts) = 'YES';
   ```

2. Copy wallet files from the source database to the target environment, and then configure the sqlnet.ora file on the target to point to the directory where the wallet is located.

   ```
   $ more sqlnet.ora
   ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=file)
   (METHOD_DATA=(DIRECTORY=/opt/oracle/oradata/ora/data/nf/wallet)))
   ```

3. If the source database does not use auto-open wallet, create the auto-open wallet at the target environment.

   ```
   $ orapki wallet create -wallet /opt/oracle/oradata/nf/wallet -auto_login
   ```

4. Proceed with provisioning the VDB as described in Provisioning an Oracle VDB.
Time Flows for RAC Provisioning of VDBs

This topic describes special considerations when provisioning by time stamp from a RAC time flow.

Time stamps in Oracle RAC time flows can be imprecise because of time skew among the hosts in a RAC configuration. The time stamps will generally track the host with the fastest clock. For this reason, provisioning by a time stamp may not leave the VDB provisioned at the exact time desired. Provision by SCN should be used if more fine-grained control is required when provisioning.
TimeFlow Patching

Introduction

The Delphix Engine provides the ability to link to an external database by creating a dSource within the Delphix system. Once linked, the Delphix Engine maintains a complete history of the database as part of a timeflow, limited by the retention policies configured by the administrator. From any time within that timeflow, you can provision a virtual database (VDB) from the Delphix Engine. This timeflow is maintained through the use of SnapSync and LogSync.

The SnapSync operation pulls over the complete data set of the external database during initial load. Subsequent SnapSync operations pull and store only incremental changes. At the end of each SnapSync operation, a snapshot is created that serves as the base point for provisioning operations. In addition, LogSync periodically connects to the host(s) running the source database and pulls over any log files associated with the database. These log files are stored separately from the SnapSync data and are used to provision from points in between SnapSync snapshots. Usually SnapSync operates against a live database with changes actively being made to it. Hence the data that it pulls over is “fuzzy” and logs must be applied to the data to make it consistent and provisionable. If LogSync is enabled, SnapSync relies on it to copy the logs over. If LogSync is not enabled, SnapSync copies the logs itself. Occasionally, LogSync or SnapSync is not able to retrieve one or more log files from the database. This creates a break in the timeflow or can prevent a snapshot from being provisioned. To remedy this situation, the Delphix Engine has tools to repair, or patch, a snapshot and the timeflow.

Snapshot Repair

When missing log files prevent a snapshot from being provisioned, you can use the graphical user interface (GUI) to determine the missing logs and repair the snapshot. The Delphix Engine will generate a fault whenever missing logs prevent a snapshot from being provisionable. The fault will likely have the title “Cannot provision database from snapshot” and will contain a description of the cause. The most common causes are:

- Logs were deleted/moved/archived from the database before the Delphix Engine could retrieve them. In this case, the archive log retention policy on the source database may be too aggressive. Use the GUI snapshot repair tool to fetch the logs.

- LogSync is still fetching the logs. SnapSync is relying on LogSync to fetch the logs needed to make the snapshot consistent. SnapSync normally will wait up to 15 minutes for LogSync to fetch the logs. If LogSync has not fetched the logs by then, SnapSync will generate a fault and finish. The best course of action in this case may be to wait for LogSync to fetch the logs.

- The source database is a physical standby in real-time apply mode. The changes described in the current online log of the database are needed to make the snapshot consistent. LogSync cannot retrieve the log until it is archived, and SnapSync cannot force the log to be archived because the source database is a physical standby. Force a log switch on the primary database or wait until the log is naturally archived.

Below is a screenshot of a snapshot with missing logs. Hovering the cursor over the (i) symbol on the snapshot card will cause the list of missing log(s) to be shown. In this example, log sequences 18 and 19 are missing.
If the snapshot can be repaired by fetching the logs from the source database, then you can use the GUI snapshot repair tool to fetch the logs. Clicking on the **tools** symbol on the snapshot card causes the tool to be started.

To use the snapshot repair tool, as seen in the above illustration:

1. Enter a **Hostname**. This should be the host from which to retrieve the log(s).

2. Enter a **Username** and **Password**. These should be the credentials for a user who can read the archived log file(s).

3. Enter a **File Path**. This should be the name of the directory containing the missing log(s).

Note: If more than one file is missing, they should all exist in the directory specified by **File Path**. The tool will read every file in the **File Path** directory so it is best that it only contains the files that are to be retrieved.
**Timeflow Patching**

When missing log files cause a break in the timeflow, you can use the command line interface (CLI) to determine the missing logs and patch the timeflow. The Delphix Engine will generate a fault whenever there are missing logs on a portion of the timeflow. The fault will likely have the title “Cannot provision a database from a portion of TimeFlow” and will contain a description of the cause. The most common cause is an overly aggressive archive log retention policy on the source database causing a log to be deleted before LogSync can fetch it. Other faults can also be generated describing the specific errors encountered when fetching the log(s).

You can use the CLI to list the missing logs and patch the timeflow. The following CLI Cookbook entry demonstrates how to do this: [CLI Cookbook: Repairing a Timeflow](#).
Provisioning from a Replicated Oracle VDB

This topic describes how to provision from a replicated dSource or virtual database (VDB). The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the replica containing the replicated object.

- Prerequisites
- Procedure
- Post-Requisites

Prerequisites
- You must have replicated a dSource or a VDB to the target host, as described in Replication Overview
- You must have added a compatible target environment on the target host

Procedure
1. Log into the Delphix Admin application for the target host.
2. Select Manage > Databases > My Databases.
3. In the list of replications, select the replica that contains the dSource or VDB you want to provision.
4. The provisioning process is now identical to the process for provisioning standard objects.

Post-Requisites

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.
Enabling and Disabling an Oracle VDB

This topic describes how to enable and disable a virtual database (VDB).

Disabling a VDB is a pre-requisite for procedures such as VDB migration or upgrade. Disabling a VDB removes all traces of it, including any configuration files, from the target environment to which it was provisioned. When the VDB is later enabled again, these configuration files are restored on the target environment.

Procedure

1. Go to Manage > Databases > My Databases.
2. Select the VDB to disable.
3. On the back of the dSource card, move the slider control from Enabled to Disabled.
4. Click Yes to acknowledge the warning.

When you are ready to enable the VDB again, move the slider control form Disabled to Enabled, and the VDB will continue to function as it did previously.
Rewinding an Oracle VDB

This topic describes the procedure for rewinding a VDB.

Rewinding a VDB rolls it back to a previous point in its TimeFlow and re-provisions the VDB. The VDB will no longer contain changes after the rewind point.

Although the VDB no longer contains changes after the rewind point, the rolled over Snapshots and TimeFlow still remain in Delphix and are accessible through the Command Line Interface (CLI). See the topic CLI Cookbook: Rolling Forward a VDB for instructions on how to use these snapshots to refresh a VDB to one of its later states after it has been rewound.

Prerequisites

To rewind a VDB, you must have the following permissions:

1. Auditor permissions on the dSource associated with the VDB.
2. Owner permissions on the VDB itself.

You do not need owner permissions for the group that contains the VDB. A user with Delphix Admin credentials can perform a VDB Rewind on any VDB in the system.

Procedure

1. Log into the Delphix Admin application.
2. Under Databases, select the VDB you want to rewind.
3. Select the rewind point as a snapshot or a point in time.
4. Click Rewind.
5. Click Provide Privileged Credentials if you want to use login credentials on the target environment other than those associated with the Environment User.
6. Click Yes to confirm.

TimeFlow bookmarks can be used as the rewind point when using the CLI. Bookmarks can be useful to:

- Mark where to rewind to - before starting a batch job on a VDB for example.
- Provide a semantic point to revert back to in case the chosen rewind point turns out to be incorrect.

See CLI Cookbook: Provisioning a VDB from a TimeFlow Bookmark for a CLI example using a TimeFlow bookmark.

Video
Refreshing an Oracle VDB

- **Prerequisites**
- **Procedure**

This topic describes how to manually refresh a VDB.

Refreshing a VDB will re-provision the VDB from the dSource. As with the normal provisioning process, you can choose to refresh the VDB from a snapshot or a specific point in time. However, you should be aware that refreshing a VDB will delete any changes that have been made to it over time. When you refresh a VDB, you are essentially re-setting it to the state you select during the refresh process. You can refresh a VDB manually, as described in this topic, or you can set a VDB refresh policy, as described in the topics Managing Policies: An Overview, Creating Custom Policies, and Creating Policy Templates.

> Although the VDB no longer contains the previous contents, the previous Snapshots and TimeFlow still remain in Delphix and are accessible through the Command Line Interface (CLI).

**Prerequisites**

To refresh a VDB, you must have the following permissions:

1. **Auditor** permissions on the dSource associated with the VDB.
2. **Auditor** permissions on the group that contains the VDB.
3. **Owner** permissions on the VDB itself.

A user with Delphix Admin credentials can perform a VDB Refresh on any VDB in the system.

**Procedure**

1. Log into the Delphix Admin application.
2. Under Databases, select the VDB you want to refresh, and then click the Open icon to open its card.
3. On the back of the VDB card, click the Refresh VDB icon in the lower right corner. This will open the screen to re-provision the VDB.
4. Select the refresh point as a snapshot or a point in time.
5. Click Refresh VDB.
6. Click Provide Privileged Credentials if you want to use login credentials on the target environment other than those associated with the Environment User.
7. Click Yes to confirm.
Deleting an Oracle VDB

This topic describes how to delete a VDB.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. Select the VDB you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.
Migrating an Oracle VDB

This topic describes how to migrate a Virtual Database (VDB) from one target environment to another.

There may be situations in which you want to migrate a virtual database to a new target environment, for example when upgrading the host on which the VDB resides, or as part of a general data center migration. This is easily accomplished by first disabling the database, then using the Migrate VDB feature to select a new target environment.

- Prerequisites
- Procedure
- Video

Prerequisites

- You should have already set up a new target environment that is compatible with the VDB that you want to migrate.

Procedure

1. Log into your Delphix Engine using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. Select the VDB you want to migrate, and then click the Open icon.
4. Slide the Enable/Disable control to Disabled, and click Yes to confirm.
   When the VDB is disabled, its icon will turn grey.
5. On the bottom-right corner of the VDB card, click the VDB Migrate icon.
6. Select the new target environment for the VDB, the user for that environment, and the database installation where the VDB will reside.
7. Click the Check icon to confirm your selections.
8. Slide the Enable/Disable control to Enabled, and click Yes to confirm.
   Within a few minutes your VDB will re-start in the new environment, and you can continue to work with it as you would any other VDB.

Video
Upgrading an Oracle VDB

This topic describes how to upgrade dSources and VDBs after an Oracle database upgrade.

Prerequisites

Do not suspend LogSync on the Delphix Engine during an Oracle upgrade of the source environment. LogSync will detect the Oracle version change, and automatically update this information on the Delphix Engine for all the associated dSources and VDBs.

Procedure

1. Refresh all environments.
2. Log into the Delphix Admin application using delphix_admin credentials.
3. Select Manage > Databases > My Databases.
4. For each dSource and VDB, click the Open icon to open its card.
5. On the back of the dSource card, click the blue upward arrow to open the Database Upgrade panel.
6. Select the desired Installation and Environment User.
   If the desired Installation is not in the options menu, click Refresh.
7. Click the check mark to finish the upgrade.
8. Repeat step 3 - 7 for each dSource and VDB that needs to be upgraded.

✔️ Updating the Oracle User after an Upgrade

There may be cases when you upgrade the Oracle home and the Oracle User (who owns the binary) is a different user than the previous Oracle User. You will then need to update the Oracle User for each environment, and then re-connect each dSource and VDB to the upgraded Oracle home using the new Oracle User.

The new Oracle User must be in the same OS group (for example, dba or oinstall) as the previous one.

1. Log into the Delphix Admin application using delphix_admin credentials.
2. Select Manage > Environments.
3. Select the environment where you want to add the user.
4. Next to Environment Users, click the Pencil icon to add the new user.
5. Set the new user as the default user.
6. Follow the procedure to upgrade the dSources and VDBs described in this topic.
**Oracle VDB Icon Reference**

This topic illustrates the icons that appear on dSources and Virtual Databases (VDBs) in the Delphix Engine Graphic User Interface, and describes the meaning of each, along with tips for clearing those that represent errors.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>There is a critical fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>There is a warning fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The Delphix Engine is checking the VDB status.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource has been deleted or the Source status is UNKNOWN.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The state of the VDB is unknown. This is often associated with a connection error.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The VDB is inactive.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource has been unlinked from the source database.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The VDB is disabled, is in the process of being created, or the creation process has been canceled or failed. See <a href="#">Enabling and Disabling VDBs</a> for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The VDB is running normally</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource is disabled. See <a href="#">Enabling and Disabling dSources</a> for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource or VDB is ready for Linux Transformation.</td>
</tr>
</tbody>
</table>
Migrate a vPDB

There may be situations in which you want to migrate a virtual pluggable database (vPDB) to a new container database on the same or a different target environment, for example when upgrading the host on which the vPDB resides, or as part of a general data center migration. This is easily accomplished by first disabling the vPDB, then using the Migrate vPDB feature to select a new container database.

Pre-requisites

You should already set up and have Delphix discover a container database in the same environment as the vPDB currently is or from a different environment to which the vPDB will be migrated to.

Procedure

Login to your Delphix Engine using Delphix Admin credentials.

1. Click Manage.
2. Select Databases.
4. Select the vPDB you want to migrate.
5. Flip the vPDB card.
6. Move the slider control to Disabled.
7. Click Yes to confirm. When the vPDB is disabled, its icon will turn gray.
8. In the bottom right-hand corner of the vPDB card, click the vPDB Migrate icon.
9. Select the new container database for the vPDB, the user for that environment, and the database installation where the container database of the vPDB will reside.
10. Click the Check icon to confirm your selections.
11. Move the slider control to Enabled.
12. Click Yes to confirm.

Within a few minutes, your vPDB will re-start in the new environment, and you can continue to work with it as you would any other vPDB.

Related Links

- [Linking an Oracle Pluggable Database](#)
- [Provisioning an Oracle Virtual Pluggable Database](#)
- [Provision an Oracle VDB](#)
- [Discovering Oracle Pluggable Databases in an Oracle Environment](#)
- [Requirements for Oracle Target Hosts and Databases](#)
- [Customizing VDB File Mappings](#)
- [Customizing Oracle Management with Hook Operations](#)
Customizing Oracle Management with Hook Operations

Hook operations allow you to execute an ordered list of custom operations as part of the dSource syncing, provisioning, and refresh process. For details on the types of operations that are available, see children of this page.

**dSource Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Sync</td>
<td>Operations performed before a sync. These operations can quiesce data to capture more information into the sync or stop applications that may interfere with the sync.</td>
</tr>
<tr>
<td>Post-Sync</td>
<td>Operations performed after a sync. These operations can undo any quiescing of data or undo any changes made by the Pre-Sync hook.</td>
</tr>
</tbody>
</table>

**Virtual Dataset Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Clone</td>
<td>Operations performed after initial provision or refresh of a virtual dataset. This hook runs when the virtual dataset has been started. During a refresh, the Delphix Engine reaches this hook before the Post-Refresh hook.</td>
</tr>
<tr>
<td>Pre-Refresh</td>
<td>Operations performed before a refresh. These operations can back up any data or configurations from the running source before doing the refresh.</td>
</tr>
<tr>
<td>Post-Refresh</td>
<td>Operations performed after a refresh. You can use these operations to restore any data or configurations backed up by the Pre-Refresh hook. During a refresh, the Delphix Engine runs this after the Configure Clone hook.</td>
</tr>
</tbody>
</table>

**Operation Failure**

If a hook operation fails, it will fail the entire sync, provision, or refresh job. The job will halt immediately and no further hook operations will be run.

**Hook Operations on Cluster Environments**

When linking from, or provisioning to, cluster environments such as Oracle RAC environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine picks a node in the cluster at random and guarantees all operations within any single hook will execute serially on this node. Note that the Delphix Engine does not guarantee the same node is chosen for the execution of every hook.

**Setting Hook Operations**

You can construct hook operation lists through the Delphix Admin application or the command line interface (CLI).
You can either define the operation lists as part of the linking or provisioning process or edit them on dSources or virtual datasets that already exist.

 Setting Hook Operations through the Delphix Admin Application

To specify hook operations during linking or provisioning, navigate to the Hooks tab of the Linking Wizard or Provision Wizard.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Select the type of operation or click Import to load a hook operation template.
4. Click the text area and edit the contents of the operation.
5. You can reorder operations either through drag-and-drop or by clicking the arrow icons.
6. To remove an operation from the list, click the Trash icon on the operation.
7. When you have set all hook operations, click Next to continue with the provisioning process.

To edit hook operations on an already-existing dSource or virtual dataset, navigate to the Hooks tab on the back of the dSource card or virtual dataset card.

1. Select the hook to edit.
2. The current operations at this hook will be displayed. To edit this list of operations, click the Pencil icon in the top right-hand corner of the card.
3. Click the Plus icon to add a new operation.
4. Select the type of operation or click Import to load a hook operation template.
5. Click the text area and edit the contents of the operation.
6. You can reorder operations either through drag-and-drop or by clicking the arrow icons.
7. To remove an operation from the list, click the Trash icon on the operation.
8. When you have set all hook operations, click Check to save the changes.

 Setting Hook Operations through the CLI

To specify hook operations during linking, edit the relevant hook's array of operations defined on the LinkingParameters > Source > Operations object.

To specify hook operations during provisioning, edit the relevant hook's array of operations defined on the ProvisionParameters > Source > Operations object.

To edit hook operations on an already-created dSource, edit the relevant hook's array of operations defined on the Source > Operations object.

To edit hook operations on an already-created virtual dataset, edit the relevant hook's array of operations defined on the Source > Operations object.

For more information about these CLI objects, see the LinkedSourceOperations, VirtualSourceOperations, RunCommandOnSourceOperation, and RunExpectOnSourceOperation API documentation in the Help menu of the Delphix Admin application.

 Example of Editing Hook Operations through the CLI

1. Navigate to relevant source's VirtualSourceOperations object.
2. Select a hook to edit.

```delphix
2. Select a hook to edit.

delphix> source
delphix source> select "pomme"
delphix source "pomme"> update
delphix source "pomme" update *> edit operations
delphix source "pomme" update operations *> edit postRefresh
```

3. Add an operation at index 0.

```delphix
3. Add an operation at index 0.

delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 0 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 0 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 0 *> ls
Properties
  type: RunCommandOperation (*)
  command: echo Refresh completed. (*)
delphix source "pomme" update operations postRefresh 0 *> commit
```

4. Add another operation at index 1 and then delete it.

```delphix
4. Add another operation at index 1 and then delete it.

delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 1 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 1 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 1 *> back
delphix source "pomme" update operations postRefresh *> unset 1
delphix source "pomme" update operations postRefresh *> commit
```

**Hook Operation Templates**

You can use templates to store commonly-used operations, which allows you to avoid repeated work when an operation is applicable to more than a single dSource or virtual dataset. You manage templates through the Delphix Admin application.

**Creating a Hook Operation Template**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Operation Templates.
4. Click the Plus icon to add a new operation template.
5. Enter a Name for the template.
6. Select an operation Type.
7. Enter a Description detailing what the operation does or how to use it.
8. Enter operation Contents to implement the operation partially or fully.
9. Click Create.

**Importing a Hook Operation Template**
To import a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Click Import.
4. Select the template to import.
5. Click Import.
6. When you have set all hook operations, click Check to save the changes.

Exporting a Hook Operation Template

To export a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Select the type of operation.
4. Click the text area and edit the contents of the operation.
5. Click Export.
6. Enter a Name for the template.
7. Enter a Description detailing what the operation does or how to use it.
8. Click Create.
Oracle Hook Operation Types

- **RunCommand Operation**
  - Examples of RunCommand Operations
- **RunExpect Operation**
  - Example of a RunExpect Operation
- **RunPowershell Operation**
  - Example of a RunPowershell Operation
- **Oracle Environment Variables**
  - dSource Environment Variables
  - VDB Environment Variables

**RunCommand Operation**

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at `bin/sh`. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

**Examples of RunCommand Operations**

You can input the full command contents into the RunCommand operation.

```bash
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
if test -d "$remove_dir"; then
  rm -rf "$remove_dir" || exit 1
fi
exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```bash
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

**RunExpect Operation**

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as `ssh`. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Admin application and CLI to aid in debugging.
If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunExpect Operation

Start an `ssh` session while interactively providing the user's password.

```bash
spawn ssh user@delphix.com
expect {
    -re {Password: } { send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {
        puts "Timed out waiting for password prompt."
        exit 1
    }
} exit 0
```

RunPowershell Operation

The RunPowershell operation executes a Powershell script on a Windows environment. The Delphix Engine captures and logs all output of the script. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunPowershell Operation

You can input the full command contents into the RunPowershell operation.

```powershell
$removedir = $Env:DIRECTORY_TO_REMOVE
if ((Test-Path $removedir) -And (Get-Item $removedir) -is [System.IO.DirectoryInfo]) {
    Remove-Item -Recurse -Force $removedir
} else {
    exit 1
}
exit 0
```

Oracle Environment Variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set environment variables so that the user-provided operations can use them to access the dSource or VDB.

**dSource Environment Variables**

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE_SID</td>
<td>The SID of the dSource</td>
</tr>
<tr>
<td>ORACLE_BASE</td>
<td>The home directory of the Oracle software hosting the dSource</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>The Oracle home for the dSource</td>
</tr>
</tbody>
</table>
VDB Environment Variables

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRS_HOME (only set for RAC dSources)</td>
<td>The home directory for cluster services hosting the dSource</td>
</tr>
<tr>
<td>ORAENV_ASK</td>
<td>Always set to NO</td>
</tr>
<tr>
<td>ORACLE_SID</td>
<td>The SID for the VDB</td>
</tr>
<tr>
<td>ORACLE_BASE</td>
<td>The home directory for the Oracle software hosting the VDB</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>The Oracle home for the VDB</td>
</tr>
<tr>
<td>CRS_HOME (only set for RAC VDBs)</td>
<td>The home directory for cluster services hosting the RAC VDB</td>
</tr>
<tr>
<td>DELPHIX_MOUNT_PATH</td>
<td>The root of the NFS mount hosting the VDB data</td>
</tr>
<tr>
<td>ORAENV_ASK</td>
<td>Always set to NO</td>
</tr>
</tbody>
</table>

**PATH and LD_LIBRARY_PATH configuration**

- **PATH** is configured by appending the **bin** directory in the Oracle home for the dSource or VDB.
- **LD_LIBRARY_PATH** is configured by appending the **lib** directory in the Oracle home for the dSource or VDB.
SQL Server Environments and Data Sources
SQL Server Support and Requirements

These topics describe specific requirements for Windows environments, such as user privileges, as well as supported Microsoft SQL Server versions and compatible operating systems.

- Delphix as a Backup Solution to SQL Server
- Network and Connectivity Requirements for SQL Server Environments
- Requirements for SQL Server Source Hosts and Databases
- Requirements for SQL Server Target Hosts and Databases
- Requirements for SQL Server Validated Sync Targets
- SQL Server Operating System Compatibility Matrices
- SQL Server Target Host iSCSI Configuration Parameter Recommendations
- Supported Operating Systems, Server Versions, and Backup Software for SQL Server
Requirements for SQL Server Source Hosts and Databases

Source hosts are the servers that contain data sources to which the Delphix Engine links and from which it provisions virtual databases. Collectively, the source host and its databases are referred to as the source environment. This topic describes the requirements for creating connections between the Delphix Engine and SQL Server source hosts and databases.

- **Requirements for SQL Server Source Hosts and Databases**
  - **Windows Domain User Requirements**
  - **Database User Requirements**
  - **Related Links**

**Requirements for SQL Server Source Hosts and Databases**

Each SQL Server source host must meet these requirements:

- Either the source host must belong to the same Windows domain as the target environments, or the domain that the source environment uses must have appropriate cross-domain trust relationships established with the target environments' domains.
- Source hosts can be running any supported Windows operating system version. For more information, see the topic [Supported Operating Systems, Server Versions, and Backup Software for SQL Server](#).
- The SQL Server instance on the source host should run as either domain users or local service accounts. Delphix does not support running SQL Server instances as local user accounts.
- The validated sync environment that the Delphix Engine uses must have access to an existing full backup of the source database in order to create the first full copy. Alternatively, the Windows Database User described below must have permissions to initiate a copy-only full backup of the source database.
- Backup images of the source database, including full, differential, and/or transaction logs, must be available over an SMB share to a staging environment. For more information, see the topic [Setting Up SQL Server Environments: An Overview](#).
- You must enable TCP/IP access for each SQL Server instance that contains a database to which the Delphix Engine will link.
  - To enable TCP/IP access, access the SQL Server Config Manager and select Network Configuration > Protocols > TCP/IP.
- If the source database is backed up with third-party backup software like LiteSpeed or Red Gate SQL Backup Pro, you must also install the backup software on the validated sync environment. For backup software compatibility requirements, see the topic [Supported Operating Systems, Server Versions, and Backup Software for SQL Server](#).

**Linking to Databases on Windows Server Failover Clusters**

You can use databases on Windows Server Failover Clusters (WSFC) as data sources. Add the environment as described below, based on which WSFC feature the source databases use:

- **Failover Cluster Instances**
  - Add the environment as a **standalone** source using the cluster name or address.
- **AlwaysOn Availability Groups**
  - Add the environment as a **cluster** source using the cluster name or address.

**Windows Domain User Requirements**

The source environment must have a Windows Domain user (for example, `delphix_src`) that the Delphix Engine
can use. This is the user that you provide when adding the source environment to the Delphix Engine. The user provides remote read-only access to the Windows Registry for discovering SQL Server instances and databases. This user must meet these requirements:

- Be a member of the **Backup Operators** or **Administrators** group on the source host
- Be a member of the **Backup Operators** or **Administrators** group on the staging host that will be used to create staging copies of the source databases on the source host
- Be a login with Windows Authentication to each SQL Server instance with which the Delphix Engine will communicate
  - To create a new login, access the **SQL Server Management Studio** and select **Security > Logins**
- Have the database role **db_datareader** for the **master** database on each SQL Server instance with which the Delphix Engine will communicate
  - To edit the user properties and set this role, access the **SQL Server Management Studio**, select **Security > Logins > delphix_src > User Mapping**
- If the source host belongs to a cluster, the user must have these privileges on all hosts that comprise the cluster

**Database User Requirements**

There must be a database user (for example, **delphix_db**) for each source database that will link to the Delphix Engine. This user must meet these requirements:

- Be able to login with SQL Authentication over Java database connectivity (JDBC) to the database
- For the **master** database, have the database role **db_datareader**
  - To set this role, access the **SQL Server Management Studio** and select **Security > Logins > delphix_db > User Mappings**
- For the **msdb** database, have the database role **db_datareader** for reading backup history
  - To set this role, access the **SQL Server Management Studio** and select **Security > Logins > delphix_db > User Mappings**
- If the Delphix Engine will initiate copy-only full backups of the database, the database user must have the database role **db_backupoperator** for the database
  - To set this role, access **SQL Server Management Studio** and select **Security > Logins > delphix_db > User Mappings**

**Related Links**

- [Supported Operating Systems, Server Versions, and Backup Software for SQL Server](#)
- [Setting Up SQL Server Environments: An Overview](#)
Requirements for SQL Server Target Hosts and Databases

This topic describes requirements for SQL Server target hosts, and Windows Domain or local users, for connecting with the Delphix Engine. The target host server, and the databases it contains, are collectively referred to as target environments. Windows target environments can be used for staging and/or provisioning of virtual databases (VDBs), as well as proxies for discovery of database instances on source environments. See the topic Setting Up SQL Server Environments: An Overview for more information. The requirements described in this topic apply equally to target environments used as staging environments and for provisioning VDBs.

Requirements for SQL Server Target Hosts

Each SQL Server target host must meet these requirements:

1. It must either belong to the same Windows domain as the source hosts, or the domain used by the target host must have appropriate cross-domain trust relationships established with the source hosts' domains.
2. The SQL Server instance on the target host should run as either domain users or local service accounts. Delphix does not support running SQL Server instances as local user accounts.
3. The SQL Server instance on the target host should be the same version or higher than the instance hosting the source database, with one exception. If a source database comes from a SQL Server 2005 instance, then the target hosts that will host VDBs from that source must be running either a SQL Server 2005 instance or a SQL Server 2012 instance or higher.
4. The target host must have 64-bit Windows as the operating system. Delphix does not support 32-bit target systems.
5. To add a Windows cluster as a target environment see the topic Adding a SQL Server Failover Cluster Target Environment.
6. If the target host is a VMware virtual machine, then the Windows Server operating system must be configured to use the VMXNET3 network driver.
7. The operating system version on a target host that will be used for the provisioning of VDBs should be equal to or higher than the operating system on the target that is hosting the staging databases for the dSource from which the VDB is being provisioned. There is no OS compatibility requirement between source and target hosts. See the topic SQL Server Operating System Compatibility Matrices for more information.
8. Windows PowerShell 2.0 or higher must be installed.
9. Execution of Windows PowerShell scripts must be enabled on the target host. While running Windows PowerShell as an Administrator, enter this command to enable script execution: `Set-ExecutionPolicy Unrestricted`.
10. For Windows 2003 target hosts, the following should be installed:

   a. Windows Server iSCSI initiator (available for download).
   b. Hotfix documented in Microsoft Knowledge Base article KB 943043.
11. The Windows iSCSI Initiator Service should have its Startup Type set to Automatic, and the service should be running. See the topic SQL Server Target Host iSCSI Configuration Parameter Recommendations for configuring the Windows iSCSI Initiator Service.

☑️ Upgrading VDBs from SQL Server 2005

You can first provision a VDB to SQL Server 2005 and then upgrade it to a higher version by following the steps described in the topic Upgrading SQL Server VDBs. See the topic SQL Server Operating System Compatibility Matrices for more information about compatibility between different versions of SQL Server.
12. The Delphix Connector must be installed, as described in the topics Setting Up SQL Server Environments: An Overview and Adding a SQL Server Standalone Target Environment.

**Flash Player Required for Connector Download**

A Flash player must be available on the target host to download the Delphix Connector when using the Delphix GUI. If the target host does not have a Flash player installed, you can download the connector directly from the Delphix Engine by navigating to this URL: http://<name of your Delphix Engine>/connector/DelphixConnectorInstaller.msi

13. **Shared Memory** must be enabled as a Network Protocol for the SQL instances on the target.

   In SQL Server Config Manager, navigate to Client Protocols > Shared Memory to enable this.

14. TCP/IP access must be enabled for each SQL Server instance on the target host to allow remote connections to instances.

   In SQL Server Config Manager, navigate to Network Configuration > Protocols > TCP/IP to enable TCP/IP access.

**Windows User Requirements**

There must be a Windows user (for example, delphix_trgt) for the target host that can be used by the Delphix Engine. This user can be a Windows domain user or a local user. However, using a local user account will prevent the target host from being used as a staging target. This user must meet these requirements:

1. Must be a member of the local Administrators group for access to discovery operations on source hosts, and for mounting iSCSI LUNs presented by the Delphix Engine to the target host.

2. Must have the server role sysadmin for each SQL Server instance that the Delphix Engine will communicate with.

   In SQL Server Management Studio, navigate to Security > Logins >delphix_trgt > Server Roles to set this role for the user.

3. Must have Log on as a batch job rights so the Delphix Engine can remotely execute commands via Powershell.

   Using the secpol.msc security policy editor, navigate to Local Policies > User Rights Assignment > Log on as a batch job to set this right.

**Related Links**

- Setting Up SQL Server Environments: An Overview
- SQL Server Operating System Compatibility Matrice
- SQL Server Target Host iSCSI Configuration Parameter Recommendations
Requirements for SQL Server Validated Sync Targets

This topic describes additional requirements for SQL Server environments that will be used as targets for validated sync. For more information, see Setting Up SQL Server Environments: An Overview.

- Requirements for SQL Server Validated Sync Target Environments
  - Windows Domain User Requirements
- Related Links

Requirements for SQL Server Validated Sync Target Environments

Each SQL Server target environment used for validated sync must meet these requirements:

- Only standalone target environments can be used as validated sync target environments. Windows Failover Cluster target environments and SQL Server Failover Cluster instances cannot be used.
- The SQL Server instance must be the same version as the instance hosting the source database. For more information about compatibility between different versions of SQL Server, see SQL Server Operating System Compatibility Matrices.
- The owner of the SQL Server instances on the target environment that are used for the staging databases must have SMB read access to the location containing the backup images of the source databases.
- If the source database is backed up with third-party backup software like LiteSpeed or Red Gate SQL Backup Pro, you must install the backup software on both the source and the validated sync environment. For backup software compatibility requirements, see Supported Operating Systems, Server Versions, and Backup Software for SQL Server.

Windows Domain User Requirements

The Windows Domain user (for example, delphix_src) that the Delphix Engine uses on a source environment must also be enabled on the target that hosts the staging databases for the source databases. This user must be a member of the Backup Operators or Administrators group on the staging environment.

The Windows Domain user (for example, delphix_trgt) that the Delphix Engine uses on a validated sync environment must:

- Meet all the requirements for the Windows user on a target host as outlined in Requirements for SQL Server Target Hosts and Databases
- Have SMB read access to the location holding the backup files of the source database

Related Links

- Setting Up SQL Server Environments: An Overview
- SQL Server Operating System Compatibility Matrices
- Supported Operating Systems, Server Versions, and Backup Software for SQL Server
- Requirements for SQL Server Target Hosts and Databases
Supported Operating Systems, Server Versions, and Backup Software for SQL Server

This topic describes the versions of the Windows operating system and Microsoft SQL Server that Delphix supports.

- **Supported Versions of Windows OS**
- **Supported Versions of SQL Server**
- **Supported SQL Server Backup Software**

**Supported Versions of Windows OS**

- Windows Server 2003 SP2, 2003 R2
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012, 2012 R2

⚠️ Delphix supports only 64-bit versions of Windows on target hosts and validated-sync-target hosts.

Target hosts and validated-sync-target hosts running Windows Server 2003 SP2 or 2003 R2 must install the hotfix documented in **KB 943043**.

ℹ️ There are further restrictions on supported Windows and SQL Server versions for SQL Server Failover Cluster target environments. See [Adding a SQL Server Failover Cluster Target Environment](#) for details.

**Supported Versions of SQL Server**

<table>
<thead>
<tr>
<th>SQL Server Version</th>
<th>Delphix Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server 2005 (9.0)</td>
<td>Delphix 3.x</td>
</tr>
<tr>
<td>SQL Server 2008 (10.0)</td>
<td>Delphix 3.x</td>
</tr>
<tr>
<td>SQL Server 2008 R2 (10.5)</td>
<td>Delphix 3.x</td>
</tr>
<tr>
<td>SQL Server 2012 (11.0)</td>
<td>Delphix 3.1.2 and beyond</td>
</tr>
<tr>
<td>SQL Server 2014 (12.0)</td>
<td>Delphix 4.1.3 and beyond</td>
</tr>
</tbody>
</table>

**Supported SQL Server Backup Software**

The Delphix Engine interacts with source database backups in the following ways:

- When linking a new source database, the Delphix Engine can use an existing full backup to load the source database data
- When performing a sync of an existing dSource, the Delphix Engine can use an existing full backup
- After the dSource is created, the Delphix Engine picks up any new backups that are taken on the source database and applies them to the copy of the source database on the Delphix Engine. This includes:
  - Transaction log backups for databases in Full or Bulk-Logged recovery models
  - Differential and full backups for databases in Simple recovery model
Delphix currently supports the following backup software for source database backups:

- SQL Server native backups
- Quest/NetVault LiteSpeed
  - If the source database is backed up with LiteSpeed, the source and the staging environments must have LiteSpeed installed on them. The version of LiteSpeed on the staging environment must be the same or higher than that on the source. Delphix currently supports LiteSpeed v5.0.0.0 and onwards.
- Red Gate SQL Backup Pro
  - If the source database is backed up with SQL Backup Pro, the source and the staging environments must have SQL Backup Pro installed on them. The version of SQL Backup Pro on the staging environment must be the same as that on the source. Delphix currently supports SQL Backup Pro v7.3 and onwards.

⚠️ Delphix does not support encrypted backups.
SQL Server Operating System Compatibility Matrices

The matrices in this topic illustrate the compatibility between various versions of SQL Server software used for source, staging target, and provisioning target environments.

Source Environment OS Compatibility

Source environments can be running any supported Windows operating system version. There are no compatibility requirements between the source environment’s operating system and that on the target environments.

Staging and Provisioning Target Environment OS Compatibility Matrix

The operating system version on the target environment that will contain the virtual databases should be equal to or higher than that on the staging target. See Setting Up SQL Server Environments: An Overview for more information.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Staging Target Environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows 2003 SP2/R2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 2008</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 2008 R2</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 2012</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 2012 R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Source and Staging Environment SQL Server Compatibility Matrix

The SQL Server version on the staging environment should be equal to that on the source environment.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Server 2005</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SQL Server 2008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
### Source and Provisioning Environment SQL Server Compatibility Matrix

When provisioning a VDB, the SQL Server version on the target can be equal to or higher than that on the source.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server 2005</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SQL Server 2008</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SQL Server 2008 R2</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SQL Server 2012</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SQL Server 2014</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

⚠️ **Provisioning to Higher SQL Versions When the Source is SQL Server 2005**

For SQL Server 2005, direct provisioning to higher SQL Server versions is only supported for provisioning to SQL Server 2012 or higher. You can first provision a VDB to SQL Server 2005 and then upgrade it to a higher version by following the steps outlined in the topic [Upgrading SQL Server VDBs](#).
SQL Server Target Host iSCSI Configuration Parameter Recommendations

This topic describes Microsoft iSCSI Configuration Parameter Recommendations.

We recommend the following Microsoft iSCSI Initiator configuration parameters for target and staging hosts. See How to Modify the Windows Registry on the Microsoft Support site for details about configuring registry settings.

A Windows Server reboot is required after changing iSCSI configuration parameters.

<table>
<thead>
<tr>
<th>Registry Key</th>
<th>Registry Value</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\iSCSI\Discovery</td>
<td>MaxRequestHoldTime</td>
<td>REG_DWORD</td>
<td>0x384 (900)</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Disk</td>
<td>TimeoutValue</td>
<td>REG_DWORD</td>
<td>0x384 (900)</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Tcpip\Parameters\Interfaces&lt;Interface GUID&gt;</td>
<td>TcpAckFrequency</td>
<td>REG_DWORD</td>
<td>0x1 (1)</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class{4D36E97B-E325-11CE-BFC1-08002BE10318}&lt;Instance Number&gt;\Parameters</td>
<td>iSCSIDisableNagle</td>
<td>REG_DWORD</td>
<td>0x1 (1)</td>
</tr>
</tbody>
</table>

For systems running Windows 2003 see Microsoft Knowledge base article 815230 for hotfix information regarding changing TcpAckFrequency.
Network and Connectivity Requirements for SQL Server Environments

- **General Port Allocation**
  - General Outbound from the Delphix Engine Port Allocation
  - General Inbound to the Delphix Engine Port Allocation
- **Firewalls and Intrusion Detection Systems (IDS)**
- **SSHD Configuration**

Connectivity Requirements for SQL Server Environments

Port Allocation for SQL Server Environments

- **Outbound from the Delphix Engine Port Allocation**
- **Inbound to the Delphix Engine Port Allocation**
- **Port Allocation Between Source and Staging Target Environments**

**General Port Allocation**

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

**General Outbound from the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>25</td>
<td>Connection to a local SMTP server for sending email</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>53</td>
<td>Connections to local DNS servers</td>
</tr>
<tr>
<td>UDP</td>
<td>123</td>
<td>Connection to an NTP server</td>
</tr>
<tr>
<td>UDP</td>
<td>162</td>
<td>Sending SNMP TRAP messages to an SNMP Manager</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
<td>SSL connections from the Delphix Engine to the Delphix Support upload server</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>636</td>
<td>Secure connections to an LDAP server</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Connections to a Delphix replication target (see Configuring Replication)</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections to source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>

**General Inbound to the Delphix Engine Port Allocation**
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>80</td>
<td>HTTP connections to the Delphix GUI</td>
</tr>
<tr>
<td>UDP</td>
<td>161</td>
<td>Messages from an SNMP Manager to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>443</td>
<td>HTTPS connections to the Delphix GUI</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections from source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>32768 - 65535</td>
<td>Required for NFS mountd and status services from target environment only if the firewall between Delphix and the target environment does not dynamically open ports.</td>
</tr>
</tbody>
</table>

**Note:** If no firewall exists between Delphix and the target environment, or the target environment dynamically opens ports, this port range is not explicitly required.

**Firewalls and Intrusion Detection Systems (IDS)**

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

**SSHD Configuration**

Both source and target Unix environments are required to have sshd running and configured such that the Delphix
Engine can connect over ssh.

The Delphix Engine expects to maintain long-running, highly performant ssh connections with remote Unix environments. The following sshd configuration entries can interfere with these ssh connections and are thus disallowed:

<table>
<thead>
<tr>
<th>Disallowed sshd Configuration Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientAliveInterval</td>
</tr>
<tr>
<td>ClientAliveCountMax</td>
</tr>
</tbody>
</table>

Connectivity Requirements for SQL Server Environments

- For source environments, the Delphix Engine uses JDBC connections to the SQL Server instances on the source environment.
- For target environments, Delphix uses a Delphix Connector connection to each target host from the Delphix Engine, and an iSCSI connection from the target environment to Delphix Engine.
- Between the source and target environments:
  - Delphix runs commands on a source environment through the Delphix Connector running on a target environment using SMB, and SQL Server commands remotely over TCP from the target to the source environment.
  - During initial load, Delphix will take a full backup of the source database and direct the backup to go to a SMB share hosted on the staging host from the source environment.
  - Delphix will attempt to access the source database's backups in the backup location from the staging target over SMB.

Port Allocation for SQL Server Environments

Refer to Setting Up SQL Server Environments: An Overview for information on SQL Server environments. The Delphix Engine makes use of the following network ports for SQL Server dSources and VDBs:

### Outbound from the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>9100</td>
<td>Delphix Connector connections to target environments</td>
</tr>
<tr>
<td>TCP</td>
<td>xxxx</td>
<td>JDBC Connections to the SQL Server instances on the source environments (typically port 1433)</td>
</tr>
</tbody>
</table>

### Inbound to the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>3260</td>
<td>iSCSI target daemon for connections from iSCSI initiators on the target environments to the Delphix Engine</td>
</tr>
</tbody>
</table>

Port Allocation Between Source and Staging Target Environments
<table>
<thead>
<tr>
<th>Outgoing</th>
<th>Incoming</th>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Environment</td>
<td>Staging Environment</td>
<td>SMB</td>
<td>445</td>
<td>Full backup of the source database during sync directed to the staging environment</td>
</tr>
</tbody>
</table>
| Staging Environment    | Source Environment     | SMB      | 445         | 1. Remote command execution on the source through Delphix Connector on the target environment  
|                        |                         |          |             | 2. Access to source database backups for restoring on the staging environment |
Delphix as a Backup Solution to SQL Server

Using Delphix as a Backup Solution to SQL Server

Delphix provides you the option to automatically manage backups of your SQL Server source databases. Prior to this release, users could not link source databases that were backed up by an unsupported backup software. In this release, we introduce a new feature that allows you to have Delphix manage the backups for your databases. This is the first step in Delphix becoming a full-featured backup solution for MSSQL databases. When the Delphix Engine manages the backups for a dSource, it takes regular, copy-only full backups of the source database, so activating the feature will not interfere with existing backup management solutions. You can configure the schedule of when the Delphix Engine takes these copy-only full backups by specifying a SnapSync policy for the dSource. You can change the SnapSync policy for a dSource at any time by visiting the policy screen; there, you can either select a new SnapSync policy or modify the current one.

If you use a backup solution that is not supported by Delphix, you cannot use your existing backups to keep your dSources in sync. However, enabling Delphix-managed backups will overcome this issue by using automatic copy-only full backups to keep dSources in sync. In the current release, dSources linked when this feature is enabled will not support LogSync functionality, which means that you can only provision VDBs from snapshots and not from any time between snapshots. Additionally, in the current release, the Delphix Engine cannot take differential or transaction log backups of the source database.

Linking SQL Server dSources with Delphix Managed Backups

The Data Management tab of the link wizard for SQL Server dSources provides the option to enable Delphix Managed Backups, as shown below:

![Delphix Managed Backups](image)

It is only possible to enable this feature here at link time. Once a dSource has been linked, you cannot modify the use of this feature. If you enable this feature, the dSource can only use Delphix-taken copy-only full backups to stay in sync with its source; the Delphix Engine will prohibit syncs using existing backups. Checking the Enabled box results in the following changes to the Data Management page:

- The initial load option is set to a Delphix-taken copy-only full backup
- The ability to provide a backup path is removed
- A SnapSync selection screen is added

You can select from the list of existing SnapSync policies, or click the green plus to create a new one. Proceeding through the remainder of the link wizard will create a dSource with Delphix-managed backups enabled. You can confirm that a dSource has the feature by expanding its dSource card and checking the Delphix Managed
Backups section, as displayed below:

**Data Management**

- Delphix Managed Backups: Yes
- Masked: No
- Pre Script: None
- Post Script: None
- SnapSync: Default SnapSync
- Retention: Default Retention
- Staging Environment: 10-43-15-66.ad.d...
- Staging Database: SQL2014STD
- Timezone: America/Los_Angeles/PST-0800

### Understanding SnapSync Policies

SnapSync policies provide you the ability to specify the frequency at which the Delphix Engine should take a copy-only full backup of a source database. As shown in the section above, selecting an initial SnapSync policy is mandatory at dSource link time. However, you can change the SnapSync policy applied on a dSource at any time by visiting the policy management screen:

1. Click **Manage**.
2. Click **Policies**.

For dSources that have Delphix-managed backups enabled, the current SnapSync policy will be displayed under the SnapSync column. The rows corresponding to dSources that do not use Delphix-managed backups will be greyed out. Clicking the **current SnapSync policy** for a dSource will display a drop-down menu of existing SnapSync policies along with the option to create a new SnapSync policy. Selecting a SnapSync policy from this list will change the current SnapSync policy for the dSource. When creating a new policy, you will see the following screen:
Here, you can configure the frequency at which the Delphix Engine takes backups of your source database. You can modify these schedules at any time by clicking the Modify Policy Templates button in the top right-hand corner of the policy management screen.

The **Timeout** field above specifies how long a SnapSync job is allowed to run before it is terminated. If a SnapSync job exceeds its timeout window, the Delphix Engine discards the new backup and rolls back the dSource to the most recent snapshot.
Managing SQL Server Environments

These topics describe special tasks and concepts for working with Windows environments containing SQL Server databases.

- Setting Up SQL Server Environments: An Overview
- Using HostChecker to Validate SQL Server Source and Target Environments
- Adding a SQL Server Standalone Target Environment
- Adding a SQL Server Source Environment
- Changing the Host Name or IQN of a SQL Server Target or Staging Host
- Editing SQL Server Environment Attributes
- Managing SQL Server Environment Users
- Deleting a SQL Server Environment
- Refreshing a SQL Server Environment
- Enabling Linking and Provisioning for SQL Server Environments
- Adding a SQL Server Failover Cluster Target Environment
Setting Up SQL Server Environments: An Overview

This topic describes the high-level process for adding SQL Server environments, linking SQL Server databases to the Delphix Engine, and provisioning virtual databases.

Block Diagram of Linking Architecture between SQL Server Environments and the Delphix Engine

Block Diagram of SQL Server Provisioning Architecture
The Delphix Connector and Environment Set Up

The Delphix Connector is a Windows service that enables communication between the Delphix Engine and the Windows target environment where it is installed.

This target machine can serve three purposes in a Delphix Engine deployment. It can:

- serve as a proxy for database discovery on source hosts
- host a staging database for a linked dSource and run the validated sync process
- host a target environment for the provisioning of Virtual Databases (VDBs)

Database discovery is initiated during the environment set up process. When you specify a production source environment that contains the databases you want to manage with the Delphix Engine, you must also specify a target environment where you have installed the Delphix Connector to act as a proxy for communication with the source environment. This is necessary because Delphix does not require that you install the Delphix Connector software on the production source environment. When you register the source environment with the Delphix Engine, the Delphix Engine uses the Delphix Connector on the proxy environment to discover SQL Server instances and databases on the source. You can then create dSources from the discovered databases. If you later refresh the source environment, the Delphix Engine will execute instance and database re-discovery through the proxy host.

SQL Server dSources are backed by a staging database that runs on a target host, as shown in the diagram. There is no requirement for additional local storage on this target host, as the storage is mounted over iSCSI from the Delphix Engine. At Delphix, we refer to the creation and maintenance of this staging database on the staging host as "validated sync," because it prepares the dSource data on the Delphix Engine for provisioning VDBs later on. After the Delphix Engine creates the staging database, it continuously monitors the source database for new transaction log backups. When it detects a new transaction log backup, it restores that backup to the staging database. The result is a TimeFlow with consistent points from which you can provision a VDB, and a faster provisioning process, because there is no need for any database recovery during provisioning.
When you later provision a VDB, you can specify any environment as a target, including the environment that contains the staging database. However, for best performance, Delphix recommends that you choose a different target environment. The only requirements for the target are:

- it must have the Delphix Connector installed
- it must have an operating system that is compatible with the one running on the validated host, as described in Requirements for SQL Server Target Hosts and Databases

**Workflow and Tasks for SQL Server Environments**

1. Install the Delphix Connector on the standalone hosts that you will use as provisioning targets for VDBs and for the discovery and validated sync processes, as described in Adding a SQL Server Standalone Target Environment.
2. Install the Delphix Connector on all cluster nodes of Windows Failover Clusters which you will use for provisioning targets for VDBs as described in Adding a SQL Server Failover Cluster Target Environment.
3. Set up the Windows source environments using standalone target environments where you have installed the Delphix Connector as proxies, as described in Adding a SQL Server Source Environment.
4. Link to the source database as described in Linking a SQL Server dSource.
5. Provision VDBs as described in Provisioning a SQL Server VDB.

**Related Links**

- SQL Server Support and Requirements
Using HostChecker to Validate SQL Server Source and Target Environments

- What is HostChecker?
- Prerequisites
- Procedure to Validate Target Environments
- Procedure to Validate Source Environments
- Tests Run
- Additional options
- Related Links

What is HostChecker?

The HostChecker is a standalone program which validates that host machines are configured correctly before the Delphix Engine uses them as data sources and provision targets.

Please note that HostChecker does not communicate changes made to hosts back to the Delphix Engine. If you reconfigure a host, you must refresh the host in the Delphix Engine in order for it to detect your changes.

You can run the tests contained in the HostChecker individually, or all at once. You must run these tests on both the source and target hosts to verify their configurations. As the tests run, you will either see validation messages that the test has completed successfully, or error messages directing you to make changes to the host configuration.

Prerequisites

- Make sure that your source and target environments meet the requirements specified in SQL Server Support and Requirements.

Procedure to Validate Target Environments

1. Download and install the Delphix Connector as described in Adding a SQL Server Standalone Target Environment.
2. Login to the Windows target host using the Windows user account that will later be used to add the target.
3. Open Windows Powershell using the Run as Administrator option.
4. Execute the host checker script by running:
   
   \<Delphix Connector installation folder>\etc\dlpx-host-checker.ps1
5. Select a path where a report file will be saved, such as C:\temp\delphix-host-checker-report.txt.
6. Select the default option of Target Host.
7. Read the output of the checks.
8. The error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Do not be surprised or undo your work if more errors appear the next time you run HostChecker; the error you just fixed may have been masking other problems.
9. Repeat steps 4–7 until all the checks return no errors or warnings.

Procedure to Validate Source Environments

1. Select the Windows target host which will be used as a proxy for discovering the source environment as described in Setting Up SQL Server Environments: An Overview.
2. Login to the Windows target host using the Windows user account that will later be used to add the source environment.
3. Open Windows Powershell using the Run as Administrator option.
4. Execute the host checker script by running:
   `<Delphix Connector installation folder>\etc\dlpx-host-checker.ps1`
5. Select a path where a report file will be saved, such as `C:\temp\delphix-host-checker-report.txt`.
6. Select the Source Host option.
7. Specify the host name or IP address of the source environment to verify.
8. Select ALL to discover all databases, or specify a single database to verify.
9. Specify the SQL Server database login and password that will later be used to add the source environment.
10. Read the output of the checks.
11. The error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Do not be surprised or undo your work if more errors appear the next time you run HostChecker; the error you just fixed may have been masking other problems.
12. Repeat steps 4–9 until all the checks return no errors or warnings.

### Tests Run

<table>
<thead>
<tr>
<th>Test</th>
<th>MS SQL Server Source</th>
<th>MS SQL Server Target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Powershell Version</td>
<td>X</td>
<td>X</td>
<td>Verifies that Powershell 2.0 or greater is installed</td>
</tr>
<tr>
<td>Check OS User Privileges</td>
<td>X</td>
<td>X</td>
<td>For target hosts, verifies that the operating system (OS) user has administrative rights. For source hosts, verifies that the OS user can successfully perform remote registry access from the target host to the source host.</td>
</tr>
<tr>
<td>Check host settings</td>
<td>X</td>
<td>X</td>
<td>Verifies that the Delphix Engine can discover host environment details from the Windows registry</td>
</tr>
<tr>
<td>Check SQL Server instance discovery</td>
<td>X</td>
<td>X</td>
<td>Verifies that the Delphix Engine can discover SQL Server instances</td>
</tr>
</tbody>
</table>
### Additional options

Run the following to view additional host checker options:

```
dlpx-host-checker.ps1 -?
```

### Related Links
- [SQL Server Support and Requirements](#)
Adding a SQL Server Standalone Target Environment

This topic describes how to add a SQL Server standalone target environment to the Delphix Engine.

As explained in Setting Up SQL Server Environments: An Overview SQL Server targets can be used for three purposes in a Delphix Engine deployment:

- They can host a target environment for the provisioning of Virtual Databases (VDBs)
- They can host a staging database for a linked dSource and run the validated sync process
- They can serve as a proxy host for database discovery on source hosts

Regardless of the specific purpose, all Windows targets must have the Delphix Connector installed to enable communication between the host and the Delphix Engine. The instructions in this topic cover initiating the Add Target process in the Delphix Engine interface, running the Delphix Connector installer on the target machine, and then verifying that the target has been added in the Delphix Engine interface.

Prerequisites

- Make sure your target environment meets the requirements described in Requirements for SQL Server Target Hosts and Databases.
- On the Windows machine that you want to use as a target, you will need to download the Delphix Connector software through the Delphix Engine interface, install it and then register that machine with the Delphix Engine.

Procedure

1. From the machine that you want to use as a target, log into the Delphix Admin application.
2. Select Manage > Environments.
3. Next to Environments, click the green Plus icon.
4. In the Add Environment dialog, select Windows in the operating system menu.
5. Select Target.
7. Click the download link for the Delphix Connector Installer. The Delphix Connector will download to your local machine.
8. On the Windows machine that you want to want to use as a target, run the Delphix Connector installer. Click Next to advance through each of the installation wizard screens.

   ![The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.]

   a. For Connector Configuration, make sure there is no firewall in your environment blocking traffic to the port on the target environment that the Delphix Connector service will listen to.
   b. For Select Installation Folder, either accept the default folder, or click Browse to select another.
   c. Click Close to complete the installation process.
   d. Run this batch script as Administrator: `<Delphix Connector installation folder>\Delphix\DelphixConnector\connector\addhostgui.cmd`.
      When the Add Windows Target Environment Wizard launches, provide the Target Host IP Address, Delphix Engine IP Address, your login credentials, and the environment user on the Windows target host.
   e. After providing this information, click Submit, and then click Yes to confirm the target environment addition request.
9. In the Delphix Engine interface, you will see a new icon for the Target environment, and two jobs running in the Delphix Admin Job History, one to Create and Discover an environment, and another to Create an environment. When the jobs are complete, click on the icon for the new environment, and you will see the details for the environment.

Post-Requisites

- On the target machine, in the Windows Start Menu, go to Services > Extended Services, and make sure that the Delphix Connector service has a Status of Started, and that the Startup Type is Automatic.

Video

Related Links

- Setting Up SQL Server Environments: An Overview
- Requirements for SQL Server Target Hosts and Databases
Adding a SQL Server Source Environment

This topic describes how to add a SQL Server source environment.

Prerequisites

- You must have already set up SQL Server target environments, as described in *Adding a SQL Server Standalone Target Environment*
  - You will need to specify a target environment that will act as a proxy for running SQL Server instance and database discovery on the source, as explained in *Setting Up SQL Server Environments: An Overview*
- Make sure your source environment meets the requirements described in *Requirements for SQL Server Target Hosts and Databases*

Procedure

1. Login to the Delphix Admin application.
2. Select Manage > Environments.
3. Next to Environments, click the green Plus icon.
4. In the Add Environment dialog, select Windows in the operating system menu.
5. Select Source.
   a. If you are adding a Windows Server Failover Cluster (WSFC), add the environment based on which WSFC feature the source databases use:
      i. Failover Cluster Instances
         Add the environment as a standalone source using the cluster name or address.
      ii. AlwaysOn Availability Groups
         Add the environment as a cluster source using the cluster name or address.
   b. Otherwise, add the environment as a standalone source.
   Connector environments are used as proxy for running discovery on the source. If no connector environments are available for selection, you will need to set them up as described in *Adding a SQL Server Standalone Target Environment*. Connector environments must:
   - have the Delphix Connector installed
   - be registered with the Delphix Engine from the host machine where they are located.
7. Enter the Host Address, Username, and Password for the source environment.
8. Click Validate Credentials.
9. Click OK, and then click Yes to confirm the source environment addition request.
   As the new environment is added, you will see multiple jobs running in the Delphix Admin Job History to Create and Discover an environment. In addition, if you are adding a cluster environment, you will see jobs to Create and Discover each node in the cluster and their corresponding hosts. When the jobs are complete, you will see the new environment added to the list in the Environments panel. If you don’t see it, click the Refresh icon.

Related Links

- *Setting Up SQL Server Environments: An Overview*
- *Adding a SQL Server Standalone Target Environment*
- *Adding a SQL Server Failover Cluster Target Environment*
- *Requirements for SQL Server Target Hosts and Databases*
Changing the Host Name or IQN of a SQL Server Target or Staging Host

This topic describes how to change the host name or iSCSI Qualified Name (IQN) of a Windows target or staging host.

By default, Windows servers generate an IQN based on the host name assigned to the host. Changing the host name will change the host IQN as well. Because the Delphix Engine exports storage for dSources and VDBs to Windows hosts using iSCSI, changes to the Windows host name must be made according to the following procedure. If you have set a non-default IQN on a Windows target or staging host, and want to change that IQN, you must follow these procedures when changing the IQN.

Changing the host name or IQN of a Windows target or staging server requires that you modify the iSCSI Initiator configuration on the Windows host. Doing so incorrectly can cause failures in dSources, VDBs, or non-Delphix users of iSCSI on the Windows host.

The instructions in this topic describe how to change the IQN using the iscsicli command line utility. Because many people are less familiar with the iscsicli utility, the instructions also include information for using the iSCSI Initiator graphical user interface.

Failing to carefully follow the steps below in order can cause availability issues for your dSources and VDBs. If you have questions about the following instructions, please contact Delphix Support for help.

Procedure

1. Disable the dSources as described in Enabling and Disabling dSources.
2. Disable the VDBs as described in Enabling and Disabling Virtual Databases.

   If your Windows server has dSources or VDBs from more than one Delphix Engine, you will need to disable the dSources and VDBs on each Delphix Engine.

3. Remove any remaining persistent volumes from the Windows server.
   In the iSCSI Initiator configuration tool, use the options available in the Volumes and Devices tab.

   Follow these steps to use the iscsicli command line utility:
   a. List the persistent volumes

   ```
   PS C:\> iscsicli reportpersistentdevices
   Microsoft iSCSI Initiator Version 6.1 Build 7601
   Persistent Volumes
   "\\\storage#volume#(bb38add1-d03f-11e1-8767-005056b37fe6)#000000008010000
   #{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}"
   "C:\Program
   Files\Delphix\DelphixConnector\564d6fbb-df9d-e90b-00f1-da37b17011d3-staging
   -15\ARCHIVE"
   [...]
   The operation completed successfully.
   ```
   b. Volumes with a "normal" path correspond to mounted volumes. For example, "C:\Program
   Files\Delphix is a normal path. If you see any normal paths in the output, be sure you have disabled all of the VDBs and dSources.
c. Volumes with a path beginning "\\?\" correspond to unmounted persistent volumes. Remove each of them:

```
PS C:\> iscsicli RemovePersistentDevice "\\?\storage\volume\{(bb38add1-d03f-11e1-8767-005056b37fe6)#000000008010000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}""
```

d. Alternately, if all of the persistent devices are for unmounted volumes, you can remove them all at once with this command:

```
PS C:\> iscsicli clearpersistentdevices
```

4. Remove all of the persistent targets.
In the iSCSI Initiator configuration tool, use the options available in the Favorite Targets tab.

Follow these steps to use iscsicli command line utility:

a. List persistent targets:

```
PS C:\> iscsicli ListPersistentTargets
```

b. Remove the appropriate persistent targets. Below is sample output listing the persistent targets:

```
PS C:\> iscsicli ListPersistentTargets
[...]  
  Target Name : iqn.2008-07.com.delphix:02:02843619-12c4-e4d2-8041-f5c56a647bc2  
  Address and Socket : 10.43.5.45 3260  
  Session Type : Data  
  Initiator Name : Root\ISCSIPRT\0000_0  
  Port Number : <Any Port>  
  Security Flags : 0x0  
  Version : 0  
  Information Specified : 0x20  
  Login Flags : 0x0  
  Username :  
[...]
```
Misleading Help for RemovePersistentTarget Command

The help for iscsicli RemovePersistentTarget is misleading:

```
iscsicli RemovePersistentTarget <Initiator Name>
    <TargetName>
    <Port Number>
    <Target Portal Address>
    <Target Portal Socket>
```

<Initiator Name> and <Target Name> show up in the listing and should be taken directly from there. <Port Number> can be taken from the listing output, but a * should be used if Any Port is listed. <Target Portal Address> and <Target Portal Socket> are shortened to Address and Socket in the ListPersistentTargets output. The term Socket in both places is what is more typically referred to as a port.

Use the RemovePersistentTarget command to remove the target, as shown in this example:

```
PS C:\> iscsicli RemovePersistentTarget Root\ISCSIPRT\0000_0
    iqn.2008-07.com.delphix:02:02843619-12c4-e4d2-8041-f5c56a647bc2 *
    10.43.5.45 3260
```

5. Log out of any sessions.
In the iSCSI Initiator configuration tool, use the options available in the Targets tab to log out. Selected a connected session under Discovered Targets, and then click Disconnect.

Follow these steps to use the iscsicli command line utility:

a. List the sessions:

```
PS C:\> iscsicli sessionlist
Session Id : ffffa8003fb0018-4000013700000001
Target Node Name : (null)
Target Name :
iqn.2008-07.com.delphix:02:02843619-12c4-e4d2-8041-f5c56a647bc2
[...]
```

b. Log out from the target:

```
PS C:\> iscsicli logouttarget ffffa8003fb0018-40000137000000001
```

6. Change the host name or IQN

a. If you are changing the host name, follow the instructions in the Microsoft TechNet article "Rename
the Computer."

\[\text{Note that if the computer is on a domain, you will need a domain administrator to perform the rename or re-add the computer to the domain depending on the version of Windows it is running.}\]

b. If you are changing the IQN only, change it through the Microsoft iSCSI Initiator GUI following the instructions in the \textit{Microsoft iSCSI User Guide}.

7. Wait for the computer to finish rebooting.
8. Verify the new IQN in the iSCSI initiator.

\[\text{If you are using the default IQN and have changed the host name, the IQN should include the new host name.}\]

10. Re-enable the dSources as described in \textit{Enabling and Disabling dSources}.
11. Re-enable the VDBs as described in \textit{Enabling and Disabling Virtual Databases}.
12. Using the iscsicli command line utility, verify that the sessions on the Windows server are using the new IQN:

\begin{verbatim}
PS C:\> iscsicli sessionlist
Microsoft iSCSI Initiator Version 6.1 Build 7601
Total of 1 sessions
Session Id : fffffa8003f77018-4000013700000004
Initiator Node Name : <NEW IQN>
[...]
\end{verbatim}

Related Links
- \textit{Enabling and Disabling dSources}
- \textit{Enabling and Disabling Virtual Databases}
- \textit{Microsoft TechNet} article “Renaming the Computer”
- \textit{Microsoft iSCSI User Guide} (download)
Editing SQL Server Environment Attributes

- **Procedure**
  - **Common Editable Attributes**
  - **SQL Server Attributes**

This topic describes how to edit attributes of an environment such as name, host address, ssh port, or toolkit path, as well as descriptions of more advanced attributes for specific data platforms.

**Procedure**

1. Log in to the Delphix Admin application with Delphix Admin credentials or as the owner of an environment.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of an environment to view its attributes.
4. Under Attributes, click the Pencil icon to edit an attribute.
5. Click the Check icon to save your edits.

**Common Editable Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment Users</td>
<td>The users for that environment. These are the users who have permission to ssh into an environment, or access the environment through the Delphix Connector. See the Requirements topics for specific data platforms for more information on the environment user requirements.</td>
</tr>
<tr>
<td>Host Address</td>
<td>The IP address for the environment host.</td>
</tr>
<tr>
<td>Notes</td>
<td>Any other information you want to add about the environment.</td>
</tr>
</tbody>
</table>

**SQL Server Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delphix Connector Port</td>
<td>For target environments, the port used for communication with the Delphix Connector. See Setting Up SQL Server Environments: An Overview for more information.</td>
</tr>
<tr>
<td>Connector Host</td>
<td>The host where the Delphix Connector is installed. See Setting Up SQL Server Environments: An Overview and Adding a SQL Server Target Environment for more information.</td>
</tr>
</tbody>
</table>
Managing SQL Server Environment Users

This topic describes how to manage the users associated with an environment. For information on providing Delphix users with privileges for groups and database objects, see the topics under Managing Delphix Users and Managing Policies.

- **Prerequisites**
- **Procedure**

**Prerequisites**

Users that you add to an environment must meet the requirements for that environment as described in the platform-specific Requirements topics.

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Environments.
4. Click the name of an environment to open the environment information screen.
5. Under Basic Information, click the green Plus icon to add a user.
6. Enter the Username and Password for the OS user in that environment.

### Using Public Key Encryption

If you want to use public key encryption for logging into your environment:

a. Select Public Key for the Login Type.

b. Click View Public Key.

c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.

   i. Run chmod 600 authorized_keys to enable read and write privileges for your user.

   ii. Run chmod 755 ~ to make your home directory writable only by your user.

The public key needs to be added only once per user and per environment.

You can also add public key authentication to an environment user's profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

7. Click the Check icon to save the new user.
8. To change the primary user for this environment, click the Pencil icon next to Environment Users.
9. To delete a user, click the Trash icon next to their username.
Deleting a SQL Server Environment

This topic describes how to delete an environment.

Prerequisites

You cannot delete an environment that has any dependencies, such as dSources or VDBs. These must be deleted before you can delete the environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, select the environment you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.
Refreshing a SQL Server Environment

This topic describes how to refresh an environment.

After you make changes to an environment that you have already set up in the Delphix Admin application, such as installing a new database home, creating a new database, or adding a new listener, you may need to refresh the environment to reflect these changes.

Procedure

1. Log in to the Delphix Admin application with Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of the environment to you want to refresh.
4. Click the Refresh icon.

To refresh all environments, click the Refresh icon next to Environments.
Enabling Linking and Provisioning for SQL Server Environments

This topic describes how to enable and disable staging, provisioning and linking for databases.

Before a database can be used as a dSource, you must first make sure that you have defined a staging environment and enabled linking to it. Similarly, before you can provision a VDB to a target database, you must make sure that you have enabled provisioning to it.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Slide the button next to Use as Staging to Yes or No to enable or disable staging.
5. Slide the button next to Allow Provisioning to On or Off to enable or disable provisioning.
6. Slide the button next to Allow Linking to On or Off to enable or disable linking.
Adding a SQL Server Failover Cluster Target Environment

This topic describes how to add a SQL Server Failover Cluster target environment to the Delphix Engine.

Adding a Failover Cluster target environment will discover SQL Server Failover Cluster instances that are running. You can then provision VDBs to these Failover Cluster instances.

Prerequisites

- You must add each node in the Window Failover Cluster individually as a standalone target environment using a non-cluster address. See Adding a SQL Server Standalone Target Environment.
  - A cluster node added as a standalone environment will only have non-clustered SQL Server instances discovered.
  - A cluster target environment will only have SQL Server Failover Cluster instances discovered.
- Each clustered SQL Server instance must have at least one clustered disk added to the clustered instance resource group which can be used for creating mount points to Delphix storage.
  - The clustered drive must have a drive letter assigned to it.
  - The clustered drive must be formatted using the "GUID Partition Table (GPT)" partition style.
- An additional target environment that can be used as a Connector Environment must exist. This environment must NOT be a node in the cluster. See Adding a SQL Server Standalone Target Environment.

Hotfix required for Windows 2008 R2 hosts

The following hotfix is required for Windows 2008 R2 Cluster nodes:

"0x80070490 Element Not found" error when you enumerate a cluster disk resource by using the WMI MSCLuster_Disk class query in a Windows Server 2008 R2-based failover cluster

http://support.microsoft.com/kb/2720218

Cluster environment restrictions

Failover Cluster target environments cannot be used as staging environments.

Supported Operating System and SQL Server Versions for Cluster Target Environments

<table>
<thead>
<tr>
<th>Supported Operating System Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 2008 R2</td>
</tr>
<tr>
<td>Windows 2012</td>
</tr>
<tr>
<td>Windows 2012 R2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supported SQL Server Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server 2008 (10.0)</td>
</tr>
<tr>
<td>SQL Server 2008 R2 (10.5)</td>
</tr>
</tbody>
</table>
SQL Server 2012 (11.0)  
SQL Server 2014 (12.0)

**Procedure**

1. Click **Manage**.
2. Select **Environments**.
3. Next to **Environments**, click the green **Plus** icon.
4. In the **Add Environment** dialog, select **Windows** in the operating system menu.
5. Select **Target**.
6. Select **Cluster**.
7. Specify the cluster address of the Windows Failover Cluster.
8. Select a host which is NOT a node in the cluster as the **Connector Environment**.
9. Enter the **Username** and **Password** for the target environment.
10. Click **Validate Credentials**.
11. Click **OK**.
12. Click **Yes** to confirm the target environment addition request.

   In the Delphix Engine interface, you will see a new icon for the Target environment, and two jobs running in the **Delphix Admin Job History**, one to **Create and Discover** an environment, and another to **Create** an environment. When the jobs are complete, click the icon for the new environment, and you will see the details for the environment.

**Example Environment**
In this example environment, the Delphix Connector was installed on **Connector Environment, Cluster Node 1**, and **Cluster Node 2**. Each host was added to Delphix as standalone target environments. Next, the **Windows Failover Cluster** was added as a Windows Target Cluster environment using the cluster address. **Cluster Node 1** is currently the active node for the SQL Server Failover Cluster resource group. Delphix has exported iSCSI LUs and has created the corresponding Cluster Disk resources for each VDB.

**Related Links**
- Setting Up SQL Server Environments: An Overview
- Adding a SQL Server Standalone Target Environment
- Adding a SQL Server Source Environment
- Requirements for SQL Server Target Hosts and Databases
Managing SQL Server Data Sources

These topics describe special tasks and concepts for linking SQL Server dSources.

- Linking a dSource from a SQL Server Database: An Overview
- Linking a SQL Server dSource
- Upgrading a dSource after a SQL Server Upgrade
- Migrating a SQL Server Staging Database
- Changing the Staging Target Environment for a SQL Server dSource
- Advanced Data Management Settings for SQL Server dSources
- Using Pre- and Post-Scripts with SQL Server dSources
- Enabling and Disabling SQL Server dSources
- Detaching and Re-Attaching SQL Server dSources
- Deleting a SQL Server dSource
- Provisioning from a Replicated SQL Server dSource
- SQL Server dSource Icon Reference
Linking a dSource from a SQL Server Database: An Overview

This topic describes basic concepts behind the creation of dSources from SQL Server databases.

When you create a dSource from a SQL Server database, the initial snapshot is derived from a full or differential database backup of the source database. The database backup can be a new full database backup initiated by the Delphix Engine, the most recent existing database backup, or a specific existing database backup as identified by its backup_set_uuid. When loading from an existing backup, the backup should be in a location that is accessible over SMB by both the staging target's environment user and the Windows user running the SQL Server instance on the staging host, as shown in the diagram in Setting Up SQL Server Environments: An Overview.

After obtaining the initial snapshot and linking the dSource, the Delphix Engine keeps the dSource and the source database in sync by monitoring the source database for new backups, and then using those backups to perform a restore on the staging database, as described in Setting Up SQL Server Environments: An Overview. If the source database is in full or bulk-logged recovery model, the Delphix Engine monitors and applies new transaction log backups. If the source database is in simple recovery model, the Delphix Engine monitors and applies new full and differential database backups.

After you have linked a database into the Delphix Engine, you can re-initialize it by performing a sync on the dSource. Performing a sync restores a database backup, which can be a new full database backup taken by the Delphix Engine, the most recent full or differential database backup, or a specific full or differential backup as identified by its backup_set_uuid.

⚠️ Simple Recovery Model

If the source database is using simple recovery model, using a new full database backup taken through the Delphix Engine is not supported for initial load and sync of a dSource.

The Delphix Engine supports source database backups that SQL Server creates natively, as well as backups created by Quest/Netvault LiteSpeed and Red Gate SQL Backup Pro. For more information, see the topic Supported Operating Systems, Server Versions, and Backup Software for SQL Server.

Related Topics

- Setting Up SQL Server Environments: An Overview
- Supported Operating Systems, Server Versions, and Backup Software for SQL Server
Linking a SQL Server dSource

This topic describes how to link a dSource from a Microsoft SQL Server database.

- **Prerequisites**
- **Procedure**
- **Related Links**

**Prerequisites**

- Be sure that the source database meets the requirements described in [Requirements for SQL Server Target Hosts and Databases](#).
- You should already have set up a staging target environment as described in [Setting Up SQL Server Environments: An Overview](#) and [Adding a Windows Target Environment](#).

⚠️ **Maximum Size of a Database that Can Be Linked**

- If the staging environment uses the Windows 2003 operating system, the largest size of database that you can link to the Delphix Engine is 2TB. This is also the largest size to which a virtual database (VDB) can grow.
- For all other Windows versions, the maximum size for databases and VDBs is 32TB

In both cases, the maximum size of the database and resulting VDBs is determined by the operating system on the staging target host.

⚠️ **Failover cluster environments cannot be used for staging**

When linking a dSource, you cannot use SQL Server failover cluster instances as staging instances. When linking, select a standalone SQL Server instance to use.

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials or as the owner of the database from which you want to provision the dSource.
2. Select Manage.
3. Select Databases.
4. Select Add dSource.
   Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
5. In the Add dSource wizard, select the source database.

   ✔️ **Changing the Environment User**

   If you need to change or add an environment user for the source database, see [Managing SQL Server Environment Users](#).

6. Enter your login credentials for the source database.
7. Click Verify Credentials.
8. Click Next.
9. Select a Database Group for the dSource.
10. Click Next.
Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. For more information, see the topics under **Users, Permissions, and Policies**.

If your data source name contains non-ASCII characters, you will need to change the default dSource name to something that uses only ASCII characters.

11. Select the **method** for the Initial Load. For details on initial load options, see [Linking a dSource from a SQL Server Database: An Overview](#).

12. Enter a **backup path** from which the source database backups will be available for the Delphix Engine to restore. Alternatively, select **Autodiscover** to have the Delphix Engine automatically locate the backups by querying MSDB.

13. Select the **target environment** for creating the staging database for validated sync.

14. Select a **standalone SQL Server instance** on the target environment for hosting the staging database.

15. Select whether the data in the database is **Masked**.

16. Select whether you want **LogSync** enabled for the dSource. For more information, see [Advanced Data Management Settings for SQL Server dSources](#).

**LogSync Disabled by Default**

LogSync is disabled by default for SQL Server data sources. For more information about how LogSync functions with SQL Server data sources, see [Managing SQL Server Data Sources](#).

17. Click **Advanced** to edit retention policies and specify pre- and post-scripts. For details on pre- and post-scripts, refer to [Using Pre- and Post-Scripts with SQL Server dSources](#).

18. Click **Next**.

19. Review the **dSource Configuration** and **Data Management** information.

20. Click **Finish**.

The Delphix Engine will initiate two jobs to create the dSource, **DB_Link** and **DB_Sync**. You can monitor these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System > Event Viewer**. When the jobs have completed successfully, the **database** icon will change to a **dSource** icon on the **Environments > Databases** screen, and the dSource will appear in the list of **My Databases** under its assigned group.

You can view the current state of **Validated Sync** for the dSource on the **dSource card** itself.

**The dSource Card**

After you have created a dSource, the **dSource card** allows you to view information about it and make modifications to its policies and permissions. In the **Databases** panel, click the **Open** icon to view the front of the dSource card. You can then flip the card to see information such as the **Source Database** and **Data Management** configuration. For more information, see the topic [Advanced Data Management Settings for SQL Server dSources](#).

**Related Links**

- **Users, Permissions, and Policies**
- [Setting Up SQL Server Environments: An Overview](#)
- [Linking a dSource from a SQL Server Database: An Overview](#)
• Advanced Data Management Settings for SQL Server dSources
• Adding a SQL Server Standalone Target Environment
• Requirements for SQL Server Target Hosts and Databases
• Using Pre- and Post-Scripts with SQL Server dSources
Upgrading a dSource after a SQL Server Upgrade

This topic describes how to upgrade dSources after an SQL Server database upgrade.

Prerequisites

- The source SQL Server database has been upgraded by attaching to a higher version of SQL Server instance.

Procedure

1. Refresh all environments.
2. Log into the Delphix Admin application using delphix_admin credentials.
3. Select Manage > Databases > My Databases.
4. Disable the dSource to be upgraded.
5. Click the Expand icon to open its card.
6. Click the crown icon on the bottom of the dSource card.
   The Upgrade Database screen will open. The new instance should appear in the dropdown list. If it does not, go to Manage->Environments, select a card with the environment containing the new instance, and click Refresh Environment on that card.
7. Select the new SQL Server instance that the source database is attached to.
8. Select the appropriate staging environment and instance. The staging instance must be the same version as the new SQL Server instance.
9. Click OK.
10. Enable the dSource.
11. Click Snapshot on the dSource card to run SnapSync for the dSource.

Related Links

- Refreshing an Environment
- Linking a SQL Server dSource
- Enabling and Disabling dSources
Migrating a SQL Server Staging Database

This topic describes how to migrate a SQL Server staging database to a different environment. For an overview of what a staging database is used for, see Setting Up SQL Server Environments: An Overview.

Prerequisites

- The dSource for the staging database has to be disabled first before the migration. Follow the steps in Enabling and Disabling dSources to disable the dSource.
- The target environment for the migrated staging database should already have been added to the Delphix Engine. Follow the steps in Adding a SQL Server Standalone Target Environment to add the environment as a target environment. The environment should also meet the requirements for hosting a staging database as described in Requirements for SQL Server Target Hosts and Databases.

Procedure

1. Go to Manage > Database > My Databases
2. Select the dSource for the staging source.
3. Modify the Staging Environment for the dSource by clicking the Pencil icon next to it.
4. Select the new target environment for the staging source.
5. Select the SQL Server instance on the new target environment.
6. Accept the change.

Post-Requisites

- Enable the dSource following the steps outlined in Enabling and Disabling dSources.

Related Links

- Setting Up SQL Server Environments: An Overview
- Adding a SQL Server Standalone Target Environment
- Enabling and Disabling dSources
- Requirements for SQL Server Target Hosts and Databases
Changing the Staging Target Environment for a SQL Server dSource

This topic describes how to change the staging target environment for a SQL Server dSource.

Prerequisites

The dSource for the staging database must be disabled before the staging target environment can be changed. Follow the steps in Enabling and Disabling dSources to disable the dSource.

Procedure

1. In the Databases pane, select the dSource for which you want to change the staging target environment.
2. Click the Open icon for the dSource to view its information card.
3. On the front of the information card, click the Flip icon to view the Staging Environment on the back of the dSource card.
4. Click the Pencil icon next to Staging Environment to edit the target server and the SQL Server instance on the server to use for staging.
5. Click the Check icon to save your changes.
Advanced Data Management Settings for SQL Server dSources

- **Accessing Data Management Settings**
- **Retention Policies**
- **LogSync for SQL Server dSources**

This topic describes advanced data management settings for dSources

When linking a dSource, you can use default data management settings for Retention, sync validation, and LogSync policies. However, you can also use custom settings to improve overall performance and match the needs of your specific server and data environment.

**Accessing Data Management Settings**

There are three ways you can set or modify data management settings for dSources:

1. During the dSource linking process, click **Advanced** in the **Data Management** panel of the **Add dSource** wizard.
2. Click the **Edit** icon next to a policy or data management setting on the back of the dSource card. For **SnapSync** and **Retention** policies, click the policy name. This will open the **Policy Management** screen.
3. Select **Manage > Policies** in the top menu bar. This will open the **Policy Management** screen. Select the policy for the dSource you want to modify, and click **Modify**.

See [Creating Custom Policies](#) and [Creating Policy Templates](#), for more information.

**Retention Policies**

Retention policies determine the length of time that snapshots and log files are retained. The retention time for logs must be equal to or longer than the retention time for snapshots. This policy, in combination with the SnapSync policy, can have a significant impact on the performance of the Delphix Engine.

**LogSync for SQL Server dSources**

LogSync is disabled by default for SQL Server dSources because snapshots are triggered by transaction log backups of the source database taken by the user, rather than SnapSync policies. On average, transaction log backups are taken every 30 minutes to an hour, resulting in frequent snapshots. LogSync should be enabled if there is a need to be able to provision to a finer granularity than is possible using transaction log backups. Enabling LogSync will increase the storage required for the dSource on the Delphix Engine because the Delphix Engine will start storing the transaction log backups according to the log retention policy.

LogSync settings are accessible during the **Add dSource** process and on the back of the dSource card.

- **Enabled** - LogSync adds log files from the source database to the dSource, enabling the ability to provision a VDB from a specific point in time or LSN for SQL Server databases. LogSync must be enabled for this provisioning functionality to work.
Using Pre- and Post-Scripts with SQL Server dSources

- **Using Scripts with SQL Server dSources**
- **Execution Context for SQL Server Scripts**
- **Available Variables for SQL Server dSource Scripts**
- **Error handling for SQL Server PowerShell Scripts**

This topic describes the use of pre- and post-scripts with dSources.

For each script, you can add the script path and its arguments when linking a dSource in one of two ways. Either:

- During the **Add dSource** wizard process, in the **Data Management** screen, click **Advanced**.

or

- On the back of the **dSource card**, click the **Pencil** icon next to the **Pre Script** and **Post Script** fields.

To update pre- and post-script information:

1. Flip over the **dSource card**.
2. Click on the **Pencil** icon next to the **Pre Script** and **Post Script** fields.
3. When finished, click the **check mark** icon.

Note that pre-scripts are executed before the SnapSync policy, and if the script fails, SnapSync will fail as well. In the case of a post-script, the script will execute after SnapSync is complete. If a post-script fails, you will see an error message.

**Using Scripts with SQL Server dSources**

- For SQL Server dSources, pre- and post-scripts are incorporated into the validated sync process.
- For SQL Server single instance environments, scripts must exist and be readable on the staging environment.
- For SQL Server, dSource scripts can be either text or binary executables.

**Execution Context for SQL Server Scripts**

Pre- and Post-Scripts for dSources are executed in the context of the primary Windows user account of the staging environment for the dSource.

**Available Variables for SQL Server dSource Scripts**

These environment variables are set by Delphix Engine for scripts running on a SQL Server dSource:

<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURCE_INSTANCE_HOST</td>
<td>Hostname of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_INSTANCE_PORT</td>
<td>Port of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_INSTANCE_NAME</td>
<td>Name of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_DATABASE_NAME</td>
<td>Name of database linked for the dSource</td>
</tr>
</tbody>
</table>

**Error handling for SQL Server PowerShell Scripts**

If the pre- or post-script execution results in an error, the Delphix Engine expects the script to return with a non-zero
exit code. Otherwise, the error will not be detected.

PowerShell gives you a few ways to handle errors in your scripts.

- **Set $ErrorActionPreference.** This only applies to PowerShell Cmdlets. For scripts or other executables such as `sqlcmd`, PowerShell will return with exit code 0 even if there is an error, regardless of the value of `$ErrorActionPreference`. The allowable values for `$ErrorActionPreference` are:
  - **Continue** (default): Continue even if there is an error.
  - **SilentlyContinue**: SilentlyContinue acts like Continue with the exception that errors are not displayed.
  - **Inquire**: Prompts the user in case of error.
  - **Stop**: Stops execution after the first error.

- Use exception handling by using traps and try/catch blocks to detect errors and return with non-zero exit codes
- Custom error handling that can be invoked after each command execution to correctly detect errors:

```
function die {
    Write-Error "Error: [args[0]]"
    exit 1
}

function verifySuccess {
    if (!?) {
        die "[args[0]]"
    }
}

Write-Output "I'd rather be in Hawaii"
verifySuccess "WRITE_OUTPUT_FAILED"

& C:\Program Files\Delphix\scripts\myscript.ps1
verifySuccess "MY_SCRIPT_FAILED"
```
Enabling and Disabling SQL Server dSources

This topic describes how to enable and disable dSources for operations such as backup and restore.

For certain processes, such as backing up and restoring the source database, you may want to temporarily disable your dSource. Disabling a dSource turns off communication between the dSource and the source database, but does not tear down the configuration that enables communication and data updating to take place. When a disabled dSource is later enabled, it will resume communication and incremental data updates from the source database according to the original policies and data management configurations that you set.

Disabling a dSource is also a pre-requisite for several other operations, like database migration and upgrading the dSource after upgrade of the associated data source. See the topics in the Related Links section for more information.

Procedure

1. Go to Manage > Databases > My Databases.
2. Select the dSource to disable.
3. On the back of the dSource card, move the slider control from Enabled to Disabled.
4. Click Yes to acknowledge the warning.

When you are ready to enable the dSource again, move the slider control from Disabled to Enabled, and the dSource will continue to function as it did previously.
Detaching and Re-Attaching SQL Server dSources

This topic describes how to detach dSources and re-attach them to a different source database.

Each dSource contains an association with the source database, as well as the data it has pulled from the source database up to that point. It is possible to detach, or unlink, a dSource from its source database. This breaks the association with the source database without affecting the data within Delphix. Detached dSources and their source databases have these properties:

- Detached dSources can be used as the source of VDB provisioning operations
- The source database can be re-linked as a different dSource
- The dSource can be attached to a different source database and resume incremental SnapSync, provided the source database is an image of the same logical database (for example, a standby replica). Currently this is only supported through the CLI, and is most often used when performing an initial load from a standby replica and then later attaching the dSource to the production database for subsequent incremental syncs.

Detaching a dSource

1. Log into the Delphix Admin application as a user with OWNER privileges on the dSource, group, or domain.
2. Select Manage > Databases > My Databases.
3. Select the database you want to unlink or delete.
4. Click the Unlink icon.
   A warning message will appear. Click Yes

Attaching a dSource

<table>
<thead>
<tr>
<th>Rebuilding Source Databases and Using VDBs</th>
</tr>
</thead>
<tbody>
<tr>
<td>In situations where you want to rebuild a source database, you will need to detach the original dSource and create a new one from the rebuilt data source. However, you can still provision VDBs from the detached dSource.</td>
</tr>
<tr>
<td>1. Detach the dSource as described in this topic.</td>
</tr>
<tr>
<td>2. Re-name the detached dSource by clicking the Edit icon next to its name. This is necessary only if you intend give the new dSource the same name as the original one. If you attempt to give the new dSource the same name as the detached one, you will see an error message.</td>
</tr>
<tr>
<td>3. Create the new dSource from the rebuilt database.</td>
</tr>
<tr>
<td>You will now be able to provision VDBs from both the detached dSource and the newly created one, but the detached dSource will only represent the state of the source database prior to being detached.</td>
</tr>
</tbody>
</table>

The attach operation is currently only supported in the CLI. Full GUI support will be added in a future release. Only databases that represent the same physical database can be re-attached.
1. Log into the **Delphix CLI** as a user with **OWNER** privileges on the dSource, group, or domain.
2. Select the dSource by name using `database select <dSource>`
3. Run the `attachSource` command.
4. Set the source config you want to attach to using `set source.config=<newSource>`. Source configs are named by their database unique name.
5. Set any other source configuration operations as you would for a normal link operation.
6. Run the `commit` command.
Deleting a SQL Server dSource

This topic describes how to delete a dSource.

**Prerequisites**

- You cannot delete a dSource that has dependent VDBs. Before deleting a dSource, make sure all dependent VDBs have been deleted as described in [Deleting a VDB](#).

**Procedure**

1. Log into the Delphix Admin application using **Delphix Admin** credentials.
2. Select **Manage > Databases > My Databases**.
3. In the **Databases** panel, select the dSource you want to delete.
4. Click the **Trash Can** icon.
5. Click **Yes** to confirm.

⚠️ Deleting a dSource will also delete all snapshots, logs, and descendant VDB Refresh policies for that database, and the deletion cannot be undone
Provisioning from a Replicated SQL Server dSource

This topic describes how to provision from a replicated dSource or virtual database (VDB). The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the replica containing the replicated object.

- **Prerequisites**
- **Procedure**
- **Post-Requisites**

**Prerequisites**
- You must have replicated a dSource or a VDB to the target host, as described in [Replication Overview](#).
- You must have added a compatible target environment on the target host.

**Procedure**
1. Log into the Delphix Admin application for the target host.
2. Select Manage > Databases > My Databases.
3. In the list of replicas, select the replica that contains the dSource or VDB you want to provision.
4. The provisioning process is now identical to the process for provisioning standard objects.

**Post-Requisites**

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.
## SQL Server dSource Icon Reference

This topic illustrates the icons that appear on dSources and Virtual Databases (VDBs) in the Delphix Engine Graphic User Interface, and describes the meaning of each, along with tips for clearing those that represent errors.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Alert Icon" /></td>
<td>There is a critical fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Warning Icon" /></td>
<td>There is a warning fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Hourglass Icon" /></td>
<td>The Delphix Engine is checking the VDB status.</td>
</tr>
<tr>
<td><img src="image" alt="Trash Bin Icon" /></td>
<td>The dSource has been deleted or the Source status is UNKNOWN.</td>
</tr>
<tr>
<td><img src="image" alt="Question Mark Icon" /></td>
<td>The state of the VDB is unknown. This is often associated with a connection error.</td>
</tr>
<tr>
<td><img src="image" alt="Stop Icon" /></td>
<td>The VDB is inactive.</td>
</tr>
<tr>
<td><img src="image" alt="Linkage Icon" /></td>
<td>The dSource has been unlinked from the source database.</td>
</tr>
<tr>
<td><img src="image" alt="VDB Icon" /></td>
<td>The VDB is disabled, is in the process of being created, or the creation process has been canceled or failed. See <a href="#">Enabling and Disabling VDBs</a> for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Running Icon" /></td>
<td>The VDB is running normally.</td>
</tr>
<tr>
<td><img src="image" alt="Disabled Icon" /></td>
<td>The dSource is disabled. See <a href="#">Enabling and Disabling dSources</a> for more information.</td>
</tr>
<tr>
<td><img src="image" alt="Ready Icon" /></td>
<td>The dSource or VDB is ready for Linux Transformation.</td>
</tr>
</tbody>
</table>
Provisioning VDBs from SQL Server dSources

These topics describe special tasks and concepts for provisioning VDBs from SQL Server dSources.

- Provisioning SQL Server VDBs: An Overview
- Provisioning a SQL Server VDB
- File Permissions for SQL Server VDBs
- Using Pre- and Post-Scripts with SQL Server VDBs
- Extended Properties for SQL Server VDBs
- Upgrading SQL Server VDBs
- Migrating a SQL Server VDB
- Renaming a SQL Server VDB
- Rewinding a SQL Server VDB
Provisioning SQL Server VDBs: An Overview

This topic describes the basic concepts involved with provisioning VDBs from SQL Server dSources.

A dSource is a virtualized representation of a physical or logical source database. As a virtual representation, it cannot be accessed or manipulated using database tools. Instead, you must create a Virtual Database (VDB) from a dSource snapshot. A VDB is an independent, writeable copy of a dSource snapshot. You can also create VDBs from other VDBs. Once a VDB has been provisioned to a target environment, you can also implement a snapshot policy for that VDB, to capture changes within it as if it were any other logical or physical database.

For an overview of the high-level components involved in provisioning a SQL Server VDB refer to Setting Up SQL Server Environments: An Overview.

Validated Sync and LogSync

When you link a source database into Delphix, you must also specify a target environment that will host a staging database for the validated sync process, as described in Setting Up SQL Server Environments: An Overview. In this process, the Delphix Engine continuously monitors the source database for new transaction log backups. When it detects one, it restores that backup to the staging database. The result is a TimeFlow with consistent points from which you can provision a VDB, also known as snapshots.

Snapshots accumulate over time, and are available when you select the dSource in the My Databases panel of the Delphix Admin application. Each snapshot is represented as a card that includes information about the source database, operating system, end stamp, and snapshot database change number (SCN for Oracle and LSN for SQL Server). You can scroll through these cards to select the one you want, or enter a date and time to search for a specific snapshot.

If LogSync is enabled, you will also see a LogSync slider control at the top of the snapshot card. If you slide this control to the right to open LogSync, you will see a pointer that you can move along a timeline to select the exact time from which you want to provision a VDB.

Once a VDB has been provisioned, you can also take snapshots of it. As with the dSource snapshots, you can find these when you select the VDB in the My Databases panel. You can then provision additional VDBs from these VDB snapshots.

SQL Server and SAP ASE VDBs do not have LogSync support. You can only provision from VDB snapshots.

Dependencies

If there are dependencies on the SnapShot you will not be able to delete the SnapShot free space; the dependencies rely on the data associated with the SnapShot.

Related Links

- Setting Up SQL Server Environments: An Overview
- Provisioning a SQL Server VDB
Provisioning a SQL Server VDB

This topic describes how to provision a virtual database (VDB) from a SQL Server dSource.

Prerequisites

- You will need to have linked a dSource from a source database, as described in [Linking a SQL Server dSource](#), or have already created a VDB from which you want to provision another VDB.
- You should already have set up Windows target environments and installed the Delphix Connector on them, as described in [Adding a SQL Server Standalone Target Environment](#).
- Make sure you have the required privileges on the target environment as described in [Requirements for SQL Server Target Hosts and Databases](#).
- If you are provisioning to a different target environment than the one where the staging database has been set up, you need to make sure that the two environments have compatible operating systems, as described in [Requirements for SQL Server Target Hosts and Databases](#). For more information on the staging database and the validated sync process, see [Setting Up SQL Server Environments: An Overview](#).

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a means of provisioning. See [Provisioning by Snapshot and LogSync](#) in this topic for more information.
5. Click Provision. The Provision VDB panel will open, and the Database Name and Recovery Model will auto-populate with information from the dSource.
6. Select a target environment from the left pane.
7. Select an Instance to use.
8. If the selected target environment is a Windows Failover Cluster environment, select a drive letter from Available Drives. This drive will contain volume mount points to Delphix storage.
9. Specify any Pre or Post Scripts that should be used during the provisioning process. See [Using Pre- and Post-Scripts with dSources and SQL Server VDBs](#) for more information.
10. Click Next.
11. Select a Target Group for the VDB. Click the green Plus icon to add a new group, if necessary.
12. Select a Snapshot Policy for the VDB. Click the green Plus icon to create a new policy, if necessary.
13. Click Next.
14. If your Delphix Engine system administrator has configured the Delphix Engine to communicate with an SMTP server, you will be able to specify one or more people to notify when the provisioning is done. You can choose other Delphix Engine users, or enter email addresses.
15. Click Finish. When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the VDB will be included in the group you designated, and listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.
You can select a SQL Server instance that has a higher version than the source database and the VDB will be automatically upgraded. See the topic SQL Server Operating System Compatibility Matrices for more information about compatibility between different versions of SQL Server.

**Provisioning by Snapshot or LogSync**

When provisioning by snapshot, you can provision to the start of any particular snapshot, either by time or LSN.

You can take a new snapshot of the dSource and provision from it by clicking the Camera icon on the dSource card.

<table>
<thead>
<tr>
<th>Provisioning By Snapshot</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by Time</td>
<td>You can provision to the start of any snapshot by selecting that snapshot card from the TimeFlow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.</td>
</tr>
<tr>
<td>Provision by LSN</td>
<td>You can use the Slide to Provision by LSN control to open the LSN entry field. Here, you can type or paste in the LSN you want to provision to. After entering a value, it will &quot;snap&quot; to the start of the closest appropriate snapshot.</td>
</tr>
</tbody>
</table>

If LogSync is enabled on the dSource, you can provision by LogSync information. When provisioning by LogSync information, you can provision to any point in time, or to any LSN, within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the Slide to Open LogSync control at the top of an individual snapshot card.

<table>
<thead>
<tr>
<th>Provisioning By LogSync</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision by Time</td>
<td>Use the Slide to Open LogSync control to view the time range within that snapshot. Drag the red triangle to the point in time that you want to provision from. You can also enter a date and time directly.</td>
</tr>
<tr>
<td>Provision by LSN</td>
<td>Use the Slide to Open LogSync and Slide to Provision by LSN controls to view the range of LSNs within that snapshot. You must type or paste in the specific LSN you want to provision to. Note that if the LSN doesn't exist, you will see an error when you provision.</td>
</tr>
</tbody>
</table>

**Related Links**
- Linking a SQL Server dSource
- Adding a SQL Server Standalone Target Environment
- Adding a SQL Server Failover Cluster Target Environment
• **Requirements for SQL Server Target Hosts and Databases**
• **Setting Up SQL Server Environments: An Overview**
• **Using Pre- and Post-Scripts with dSources and SQL Server VDBs**
File Permissions for SQL Server VDBs

When provisioning a VDB the "access control lists" (ACLs) of database and log files are modified to help prevent unintentional data loss through file deletion. This could happen if there is an attempt to DROP a VDB directly through SQL Server management studio or other native SQL Server tools.

Each database and log file ACL is updated to include a deny delete "access control entry" (ACE) for the user account running the SQL Server instance.

VDBs can still be dropped directly through SQL Server tools, however a warning message will be displayed and the files will remain on the volume exported by Delphix. This file deletion prevention also applies to attempts to remove files from a database using the ALTER DATABASE .. REMOVE FILE command.

If a VDB is inadvertently dropped, the database can be reattached using SQL Server tools.

If you attempt to delete a database or log file and then try to add a file of the same name this may fail because the original file was prevented from being deleted by the deny delete ACE.

If it is your intention to delete the files from the volume provided by Delphix, the ACLs on the files can be changed using the icacls command.

\[ icacls \ <file> \ /remove \ <SQL Server instance account> :deny(D) \]

Accounts other than the SQL Server instance account will not be prevented from deleting the VDB database and log files.
Using Pre- and Post-Scripts with SQL Server VDBs

- Using Scripts with SQL Server dSources
- Execution Context for SQL Server Scripts
- Available Variables for SQL Server dSource Scripts
- Error handling for SQL Server PowerShell Scripts

This topic describes the use of pre- and post-scripts with dSources.

For each script, you can add the script path and its arguments when linking a dSource in one of two ways. Either:

- During the Add dSource wizard process, in the Data Management screen, click Advanced.

or

- On the back of the dSource card, click the Pencil icon next to the Pre Script and Post Script fields.

To update pre- and post-script information:

1. Flip over the dSource card.
2. Click on the Pencil icon next to the Pre Script and Post Script fields.
3. When finished, click the check mark icon.

Note that pre-scripts are executed before the SnapSync policy, and if the script fails, SnapSync will fail as well. In the case of a post-script, the script will execute after SnapSync is complete. If a post-script fails, you will see an error message.

Using Scripts with SQL Server dSources

- For SQL Server dSources, pre- and post-scripts are incorporated into the validated sync process.
- For SQL Server single instance environments, scripts must exist and be readable on the staging environment.
- For SQL Server, dSource scripts can be either text or binary executables.

Execution Context for SQL Server Scripts

Pre- and Post-Scripts for dSources are executed in the context of the primary Windows user account of the staging environment for the dSource.

Available Variables for SQL Server dSource Scripts

These environment variables are set by Delphix Engine for scripts running on a SQL Server dSource:

<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURCE_INSTANCE_HOST</td>
<td>Hostname of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_INSTANCE_PORT</td>
<td>Port of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_INSTANCE_NAME</td>
<td>Name of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_DATABASE_NAME</td>
<td>Name of database linked for the dSource</td>
</tr>
</tbody>
</table>

Error handling for SQL Server PowerShell Scripts

If the pre- or post-script execution results in an error, the Delphix Engine expects the script to return with a non-zero
exit code. Otherwise, the error will not be detected.

PowerShell gives you a few ways to handle errors in your scripts.

- Set $ErrorActionPreference. This only applies to PowerShell Cmdlets. For scripts or other executables such as sqlcmd, PowerShell will return with exit code 0 even if there is an error, regardless of the value of $ErrorActionPreference. The allowable values for $ErrorActionPreference are:
  - **Continue** *(default)*: Continue even if there is an error.
  - **SilentlyContinue**: SilentlyContinue acts like Continue with the exception that errors are not displayed.
  - **Inquire**: Prompts the user in case of error.
  - **Stop**: Stops execution after the first error.

- Use exception handling by using traps and try/catch blocks to detect errors and return with non-zero exit codes
- Custom error handling that can be invoked after each command execution to correctly detect errors:

```powershell
function die {
    Write-Error "Error:  ($args[0])"
    exit 1
}

function verifySuccess {
    if (!$?) {
        die "$($args[0])"
    }
}

Write-Output "I'd rather be in Hawaii"
verifySuccess "WRITE_OUTPUT_FAILED"
& C:\Program Files\Delphix\scripts\myscript.ps1
verifySuccess "MY_SCRIPT_FAILED"
```
Extended Properties for SQL Server VDBs

This topic describes extended properties on VDBs that can be used to track the origin of VDBs through SQL Server tools on target servers.

These are the extended properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dlpx_server_name</td>
<td>Address of the Delphix Engine hosting the VDB</td>
</tr>
<tr>
<td>dlpx_server_uuid</td>
<td>UUID of the Delphix Engine hosting the VDB</td>
</tr>
<tr>
<td>dlpx_source_id</td>
<td>Internal reference of the VDB</td>
</tr>
</tbody>
</table>

These properties can be found under the Extended Properties page of the Properties window for a VDB using the SQL Server Management Studio tool. They can also be displayed by using the sp_dlpx_vdbinfo stored procedure. This stored procedure can be installed by running the SQL code contained in <Delphix Connector install path>\etc\sp_dlpx_vdbinfo.sql.
Upgrading SQL Server VDBs

This topic describes how to upgrade a SQL Server VDB to a higher version of SQL Server instance.

**Procedure for VDB In-Place Upgrade**
1. Remove any VDB Refresh Policy assigned to the VDB.
2. Upgrade the target SQL Server instance.
3. Refresh the target environment.

**Procedure to Upgrade a VDB to a New SQL Instance**
1. Refresh all environments.
2. Log into the Delphix Admin application using delphix_admin credentials pr as the owner of the VDB.
3. Select Manage > Databases > My Databases.
4. Disable the VDB to be upgraded.
5. Click the Expand icon to open the VDB card.
6. Click the crown icon on the bottom of the VDB card. The Upgrade Database screen will open.
7. Select the new SQL Server instance you want the VDB to upgrade to.
8. Click OK.
9. Enable the VDB.
10. Repeat step 2 to 9 for each VDB you want to upgrade.

**Notes**
- Upgrading a SQL Server 2005 VDB to SQL Server 2008 or 2008 R2 is not supported.

**Related Links**
- Refreshing an Environment
- Enabling and Disabling dSources
Migrating a SQL Server VDB

This topic describes how to migrate a SQL Server VDB to a different environment.

Prerequisites
- The VDB has to be disabled first before migrating it by following the steps outlined in Enabling and Disabling Virtual Databases.
- The target environment where the VDB is to be migrated should already have been added to the Delphix Engine. Follow the steps outlined in Adding a SQL Server Standalone Target Environment.

Procedure via GUI
1. Select the VDB you would like to migrate
2. Disable VDB
3. Select the Migrate VDB icon
4. Select the New Environment and Installation
5. Hit the Check Mark
6. After job finishes Enable VDB

Procedure via CLI
1. Select the source associated with the VDB.
   ```
   delphix> source
   delphix source > select "vexample"
   ```
2. Select the source config associated with the source.
   ```
   delphix source "vexample"> get config
   vexample
   delphix source "vexample"> /sourceconfig
   delphix sourceconfig > select "vexample"
   delphix sourceconfig "vexample">
   ```
3. Update the repository to the repository on the target environment for the migration, and the environment user associated with the source config.
   ```
   delphix sourceconfig "vexample"> update
   delphix sourceconfig "vexample" update *> set repository="new target environment"/"MSSQL instance"
   delphix sourceconfig "vexample" update *> set environmentUser="new target environment"/"new target's user"
   delphix sourceconfig "vexample" update *> commit
   delphix sourceconfig "vexample">
   ```

Post-Requisites
- Enable the VDB following the steps outlined in Enabling and Disabling Virtual Databases.
Related Links

- Adding a SQL Server Standalone Target Environment
- Enabling and Disabling Virtual Databases
Renaming a SQL Server VDB

This topic describes how to rename the database name on the SQL Server instance for a SQL Server VDB through Delphix.

Database name on SQL Server vs VDB name on Delphix

The database name that is changed in this procedure is what you would see under the SQL Server instance on the target environment and is on the back of the VDB card. The name of the VDB object itself is a name internal to Delphix and is what you see on front of the VDB card.

Prerequisites

- The VDB should be running on the target environment.
- The SQL Server instance on the target environment where the VDB is should be up and reachable.

Procedure

1. Select the source associated with the VDB.

   delphix> source "vexample"

2. Select the source config associated with the source.

   delphix source "vexample" > get config
   vexample
   delphix source "vexample" > /sourceconfig "vexample"
   delphix sourceconfig "vexample" >

3. Update the databaseName to the new name.

   delphix sourceconfig "vexample" > update
   delphix sourceconfig "vexample" update *> set databaseName=newDBName
   delphix sourceconfig "vexample" update *> commit
   delphix sourceconfig "vexample" >
Rewinding a SQL Server VDB

This topic describes the procedure for rewinding a VDB.

Rewinding a VDB rolls it back to a previous point in its TimeFlow and re-provisions the VDB. The VDB will no longer contain changes after the rewind point.

Although the VDB no longer contains changes after the rewind point, the rolled over Snapshots and TimeFlow still remain in Delphix and are accessible through the Command Line Interface (CLI). See the topic [CLI Cookbook: Rolling Forward a VDB](#) for instructions on how to use these snapshots to refresh a VDB to one of its later states after it has been rewound.

Prerequisites

To rewind a VDB, you must have the following permissions:

1. **Auditor** permissions on the dSource associated with the VDB.
2. **Owner** permissions on the VDB itself.

You **do not** need owner permissions for the group that contains the VDB. A user with Delphix Admin credentials can perform a VDB Rewind on any VDB in the system.

Procedure

1. Log into the [Delphix Admin](#) application.
2. Under **Databases**, select the VDB you want to rewind.
3. Select the rewind point as a snapshot or a point in time.
4. Click **Rewind**.
5. Click **Provide Privileged Credentials** if you want to use login credentials on the target environment other than those associated with the **Environment User**.
6. Click **Yes** to confirm.

TimeFlow bookmarks can be used as the rewind point when using the CLI. Bookmarks can be useful to:

- Mark where to rewind to - before starting a batch job on a VDB for example.
- Provide a semantic point to revert back to in case the chosen rewind point turns out to be incorrect.

See [CLI Cookbook: Provisioning a VDB from a TimeFlow Bookmark](#) for a CLI example using a TimeFlow bookmark.

Video

[Get Adobe Flash Player]
Customizing SQL Server Management

This topic describes how to customize SQL Server dSources and VDB's. SQL Server dSources are customized using pre- and post-scripts. SQL Server VDBS are customized using Hook Operations.
Customizing SQL Server dSource Management with Pre- and Post-Scripts

- **Using Scripts with SQL Server dSources**
- **Execution Context for SQL Server Scripts**
- **Available Variables for SQL Server dSource Scripts**
- **Error handling for SQL Server PowerShell Scripts**

This topic describes the use of pre- and post-scripts with dSources.

For each script, you can add the script path and its arguments when linking a dSource in one of two ways. Either:

- During the Add dSource wizard process, in the Data Management screen, click Advanced.

or

- On the back of the dSource card, click the Pencil icon next to the Pre Script and Post Script fields.

To update pre- and post-script information:

1. Flip over the dSource card.
2. Click on the Pencil icon next to the Pre Script and Post Script fields.
3. When finished, click the check mark icon.

Note that pre-scripts are executed before the SnapSync policy, and if the script fails, SnapSync will fail as well. In the case of a post-script, the script will execute after SnapSync is complete. If a post-script fails, you will see an error message.

**Using Scripts with SQL Server dSources**

- For SQL Server dSources, pre- and post-scripts are incorporated into the validated sync process.
- For SQL Server single instance environments, scripts must exist and be readable on the staging environment.
- For SQL Server, dSource scripts can be either text or binary executables.

**Execution Context for SQL Server Scripts**

Pre- and Post-Scripts for dSources are executed in the context of the primary Windows user account of the staging environment for the dSource.

**Available Variables for SQL Server dSource Scripts**

These environment variables are set by Delphix Engine for scripts running on a SQL Server dSource:

<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURCE_INSTANCE_HOST</td>
<td>Hostname of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_INSTANCE_PORT</td>
<td>Port of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_INSTANCE_NAME</td>
<td>Name of linked instance for the dSource</td>
</tr>
<tr>
<td>SOURCE_DATABASE_NAME</td>
<td>Name of database linked for the dSource</td>
</tr>
</tbody>
</table>

**Error handling for SQL Server PowerShell Scripts**

If the pre- or post-script execution results in an error, the Delphix Engine expects the script to return with a non-zero
exit code. Otherwise, the error will not be detected.

PowerShell gives you a few ways to handle errors in your scripts.

- Set \$ErrorActionPreference. This only applies to PowerShell Cmdlets. For scripts or other executables such as sqlcmd, PowerShell will return with exit code 0 even if there is an error, regardless of the value of \$ErrorActionPreference. The allowable values for \$ErrorActionPreference are:
  
  - **Continue (default)**: Continue even if there is an error.
  - **ContinuelyContinue**: SilentlyContinue acts like Continue with the exception that errors are not displayed.
  - **Inquire**: Prompts the user in case of error.
  - **Stop**: Stops execution after the first error.

- Use exception handling by using traps and try/catch blocks to detect errors and return with non-zero exit codes.
- Custom error handling that can be invoked after each command execution to correctly detect errors:

```powershell
function die {
    Write-Error "Error: @{$args[0]}"
    exit 1
}

function verifySuccess {
    if (!?) {
        die " @{$args[0]}"
    }
}

Write-Output "I'd rather be in Hawaii"
verifySuccess "WRITE_OUTPUT_FAILED"

& C:\Program Files\Delphix\scripts\myscript.ps1
verifySuccess "MY_SCRIPT_FAILED"
```
Customizing SQL Server VDB Management with Hook Operations

⚠️ Hook Operations for dSources Are NotSupported
Please see Customizing SQL Server dSource Management with Pre- and Post-Scripts.

Hook operations allow you to execute an ordered list of custom operations as part of the dSource syncing, provisioning, and refresh process. For details on the types of operations that are available, see children of this page.

### dSource Hooks

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Sync</td>
<td>Operations performed before a sync. These operations can quiesce data to capture more information into the sync or stop applications that may interfere with the sync.</td>
</tr>
<tr>
<td>Post-Sync</td>
<td>Operations performed after a sync. These operations can undo any quiescing of data or undo any changes made by the Pre-Sync hook.</td>
</tr>
</tbody>
</table>

### Virtual Dataset Hooks

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Clone</td>
<td>Operations performed after initial provision or refresh of a virtual dataset. This hook runs when the virtual dataset has been started. During a refresh, the Delphix Engine reaches this hook before the Post-Refresh hook.</td>
</tr>
<tr>
<td>Pre-Refresh</td>
<td>Operations performed before a refresh. These operations can back up any data or configurations from the running source before doing the refresh.</td>
</tr>
<tr>
<td>Post-Refresh</td>
<td>Operations performed after a refresh. You can use these operations to restore any data or configurations backed up by the Pre-Refresh hook. During a refresh, the Delphix Engine runs this after the Configure Clone hook.</td>
</tr>
</tbody>
</table>

⚠️ Operation Failure

If a hook operation fails, it will fail the entire sync, provision, or refresh job. The job will halt immediately and no further hook operations will be run.

⚠️ Hook Operations on Cluster Environments

When linking from, or provisioning to, cluster environments such Oracle RAC environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine picks a node in the cluster at random and guarantees all operation within any single hook will execute serially on this node. Note that the Delphix Engine does not guarantee the same node is chosen for the execution of every hook.
Setting Hook Operations

You can construct hook operation lists through the Delphix Admin application or the command line interface (CLI). You can either define the operation lists as part of the linking or provisioning process or edit them on dSources or virtual datasets that already exist.

Setting Hook Operations through the Delphix Admin Application

To specify hook operations during linking or provisioning, navigate to the Hooks tab of the Linking Wizard or Provision Wizard.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Select the type of operation or click Import to load a hook operation template.
4. Click the text area and edit the contents of the operation.
5. You can reorder operations either through drag-and-drop or by clicking the arrow icons.
6. To remove an operation from the list, click the Trash icon on the operation.
7. When you have set all hook operations, click Next to continue with the provisioning process.

To edit hook operations on an already-existing dSource or virtual dataset, navigate to the Hooks tab on the back of the dSource card or virtual dataset card.

1. Select the hook to edit.
2. The current operations at this hook will be displayed. To edit this list of operations, click the Pencil icon in the top right-hand corner of the card.
3. Click the Plus icon to add a new operation.
4. Select the type of operation or click Import to load a hook operation template.
5. Click the text area and edit the contents of the operation.
6. You can reorder operations either through drag-and-drop or by clicking the arrow icons.
7. To remove an operation from the list, click the Trash icon on the operation.
8. When you have set all hook operations, click Check to save the changes.

Setting Hook Operations through the CLI

To specify hook operations during linking, edit the relevant hook's array of operations defined on the LinkingParameters > Source > Operations object.

To specify hook operations during provisioning, edit the relevant hook's array of operations defined on the ProvisionParameters > Source > Operations object.

To edit hook operations on an already-created dSource, edit the relevant hook's array of operations defined on the Source > Operations object.

To edit hook operations on an already-created virtual dataset, edit the relevant hook's array of operations defined on the Source > Operations object.

For more information about these CLI objects, see the LinkedSourceOperations, VirtualSourceOperations, RunCommandOnSourceOperation, and RunExpectOnSourceOperation API documentation in the Help menu of the Delphix Admin application.
Example of Editing Hook Operations through the CLI

1. Navigate to relevant source’s `VirtualSourceOperations` object.

2. Select a hook to edit.

   ```
   delphix> source
   delphix source> select "pomme"
   delphix source "pomme"> update
   delphix source "pomme" update *> edit operations
   delphix source "pomme" update operations *> edit postRefresh
   ```

3. Add an operation at index 0.

   ```
   delphix source "pomme" update operations postRefresh *> add
   delphix source "pomme" update operations postRefresh 0 *> set
type=RunCommandOnSourceOperation
   delphix source "pomme" update operations postRefresh 0 *> set command="echo Refresh completed."
   delphix source "pomme" update operations postRefresh 0 *> ls
   Properties
   type: RunCommandOperation (*)
   command: echo Refresh completed. (*)
   delphix source "pomme" update operations postRefresh 0 *> commit
   ```

4. Add another operation at index 1 and then delete it.

   ```
   delphix source "pomme" update operations postRefresh *> add
   delphix source "pomme" update operations postRefresh 1 *> set
type=RunCommandOnSourceOperation
   delphix source "pomme" update operations postRefresh 1 *> set command="echo Refresh completed."
   delphix source "pomme" update operations postRefresh 1 *> back
   delphix source "pomme" update operations postRefresh *> unset 1
   delphix source "pomme" update operations postRefresh *> commit
   ```

**Hook Operation Templates**

You can use templates to store commonly-used operations, which allows you to avoid repeated work when an operation is applicable to more than a single dSource or virtual dataset. You manage templates through the Delphix Admin application.

**Creating a Hook Operation Template**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Operation Templates.
4. Click the Plus icon to add a new operation template.
5. Enter a Name for the template.
6. Select an operation Type.
7. Enter a Description detailing what the operation does or how to use it.
8. Enter operation Contents to implement the operation partially or fully.
9. Click Create.

Importing a Hook Operation Template

To import a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Click Import.
4. Select the template to import.
5. Click Import.
6. When you have set all hook operations, click Check to save the changes.

Exporting a Hook Operation Template

To export a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Select the type of operation.
4. Click the text area and edit the contents of the operation.
5. Click Export.
6. Enter a Name for the template.
7. Enter a Description detailing what the operation does or how to use it.
8. Click Create.
SQL Server VDB Hook Operation Types

RunCommand Operation

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at /bin/sh. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Examples of RunCommand Operations

You can input the full command contents into the RunCommand operation.

```bash
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
  fi

exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

RunExpect Operation

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as ssh. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Admin application and CLI to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunExpect Operation

Start an ssh session while interactively providing the user's password.
spawn ssh user@delphix.com
expect {
  -re {Password: } {
    send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
  }
  timeout {
    puts "Timed out waiting for password prompt."
    exit 1
  }
}
exit 0

RunPowershell Operation

The RunPowershell operation executes a PowerShell script on a Windows environment. The Delphix Engine captures and logs all output of the script. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunPowershell Operation

You can input the full command contents into the RunPowershell operation.

```powershell
$removedir = $Env:DIRECTORY_TO_REMOVE

if ((Test-Path $removedir) -And (Get-Item $removedir) -is [System.IO.DirectoryInfo]) {
    Remove-Item -Recurse -Force $removedir
} else {
    exit 1
}
exit 0
```

SQL Server VDB Environment Variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific virtual databases (VDBs), the Delphix Engine will always set environment variables so that the user-provided operations can use them to access the VDB.

<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDB_INSTANCE_HOST</td>
<td>Hostname of linked instance for the VDB</td>
</tr>
<tr>
<td>VDB_INSTANCE_PORT</td>
<td>Port of linked instance for the VDB</td>
</tr>
<tr>
<td>VDB_INSTANCE_NAME</td>
<td>Name of linked instance for the VDB</td>
</tr>
<tr>
<td>VDB_DATABASE_NAME</td>
<td>Name of database linked for the VDB</td>
</tr>
</tbody>
</table>
PostgreSQL Environments and Data Sources
PostgreSQL Support and Requirements

These topics describe specific requirements for PostgreSQL environments, such as user privileges, as well as the supported operating systems and database versions.

- Requirements for PostgreSQL Source Hosts and Databases
- Requirements for PostgreSQL Target Hosts and Databases
- Supported Operating Systems and Database Versions for PostgreSQL Environments
- Network and Connectivity Requirements for PostgreSQL Environments
Requirements for PostgreSQL Source Hosts and Databases

Source hosts are servers which contain the source databases from which virtual database copies are made. Collectively, the source host and database are referred to as the source environment. This topic describes the requirements for creating connections between PostgreSQL source environments and the Delphix Engine.

Source Host Requirements

1. On 64-bit Linux environments, a 32-bit version of glibc must be installed.
2. There must be an operating system user with the following privileges:
   a. The Delphix Engine must be able to make an SSH connection to the source environment using the operating system user.
   b. The operating system user should have read and execute privileges on the PostgreSQL binaries installed on the source environment.
   c. The operating system user should have read access to the PostgreSQL data directories on the source environment.
3. There must be a directory on the source host where the Delphix Engine toolkit can be installed (e.g., /var/tmp) with the following properties:
   a. The toolkit directory must be writable by the operating system user mentioned above.
   b. The toolkit directory must have at least 256 MB of available storage.
4. TCP/IP connectivity to and from the source environment must be configured as described in General Network and Connectivity Requirements.

Source Database Requirements

1. The database must accept read/write connections (in other words, it must not be in standby mode).
2. The Delphix Engine must have access to a PostgreSQL role that has superuser, replication and login privileges. This can be the built-in postgres role or a newly-created role (for example, delphix).

Creating a Role for Use with the Delphix Engine

To create a new role for use with the Delphix Engine, use the following command:

```
SQL> CREATE ROLE delphix SUPERUSER LOGIN REPLICATION [ PASSWORD 'password']
```

3. The following changes must be made to postgresql.conf (for more information, see the Server Configuration chapter in the PostgreSQL documentation):
   a. TCP/IP connectivity must be configured to allow the role mentioned above to connect to the source database from the Delphix Engine and from the standby DBMS instance set up by the Delphix Engine on the staging environment. This can be done by modifying the listen_addresses parameter, which specifies the TCP/IP addresses on which the DBMS is to listen for connections from client applications.

listen_addresses Configuration

The simplest way to configure Postgres is so that it listens on all available IP interfaces:

```
listen_addresses = '*'  # Default is 'localhost'
```
b. The value of `max_wal_senders`, which specifies the maximum number of concurrent connections from standby servers or streaming base backup clients, must be increased from its desired value by four. That is, in addition to the allowance of connections for consumers other than the Delphix Engine, there must be an allowance for four additional connections from consumers set up by the Delphix Engine.

```plaintext
max_wal_senders Configuration

The default value of `max_wal_senders` is zero, meaning replication is disabled. In this configuration, the value of `max_wal_senders` must be increased to two for the Delphix Engine:

```
max_wal_senders = 4  # Default is 0
```
```
wal_level Configuration

The default value of `wal_level` is `minimal`, which writes only the information needed to recover from a crash or immediate shutdown to the WAL archives. In this configuration, one must add the logging required for WAL archiving as follows:

```
wal_level = archive  # Default is minimal
```
```
4. PostgreSQL must be configured to allow PostgreSQL client connections from the Delphix Engine and from the staging target environment, as well as PostgreSQL replication client connections from the staging target environment by adding the following entries to `pg_hba.conf`:

```plaintext
pg_hba.conf Configuration

<table>
<thead>
<tr>
<th>Role</th>
<th>Host</th>
<th>IP Address</th>
<th>Auth Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>&lt;role&gt;</td>
<td>&lt;ip-address_of_delphix_engine&gt;/32</td>
<td>&lt;auth-method&gt;</td>
</tr>
<tr>
<td>all</td>
<td>&lt;role&gt;</td>
<td>&lt;ip-address_of_staging_target&gt;/32</td>
<td>&lt;auth-method&gt;</td>
</tr>
<tr>
<td>replication</td>
<td>&lt;role&gt;</td>
<td>&lt;ip-address_of_staging_target&gt;/32</td>
<td>&lt;auth-method&gt;</td>
</tr>
</tbody>
</table>
```

`<auth-method>` must be `md5` or `trust` to indicate if a password is required (`md5`) or not (`trust`). For more information on how to configure `pg_hba.conf`, see the [Client Authentication] chapter in the PostgreSQL documentation.

**Related Links**
- [General Network and Connectivity Requirements](#)
- [Server Configuration](#) in the PostgreSQL documentation
- [Client Authentication](#) in the PostgreSQL documentation
Requirements for PostgreSQL Target Hosts and Databases

This topic describes user privileges and other requirements for PostgreSQL target hosts and databases, collectively referred to as the target environment.

Target Environment Requirements

1. The operating system and architecture of the target environment must match those of the source environment.

2. There must be an installation of PostgreSQL on the target environment that is compatible with an installation of PostgreSQL on the source environment. Two installations of PostgreSQL are compatible if and only if:
   a. They share the same vendor (for example, PostgreSQL is incompatible with EnterpriseDB Postgres Plus Advanced Server).
   b. They share the same major version number (for example, 8.4.2 is compatible with 8.4, 8.4.1 and 8.4.6; however, it is incompatible with 8.3, 8.3.1, or 9.2).
   c. They were compiled against the same architecture (in other words, 32-bit and 64-bit installations of Postgres are incompatible).
   d. They were compiled with the same WAL segment size. The default WAL segment size of 16 MB is rarely changed in practice, so almost all installations of PostgreSQL are compatible with each other in terms of WAL segment size.

3. On 64-bit Linux environments, a 32-bit version of glibc must be installed.

4. There must be an operating system user with the following privileges:
   a. The Delphix Engine must be able to make an SSH connection to the target environment using the operating system user.
   b. The operating system user must have read and execute privileges on the PostgreSQL binaries installed on the target environment.
   c. The operating system user must have permission to run `mount` and `umount` as the superuser via `sudo` with neither a password nor a TTY via the following entries in `/etc/sudoers.conf`:

   ```
   Defaults:<username> !requiretty
   <username> ALL=NOPASSWD:/bin/mount, /bin/umount
   ```

5. There must be a directory on the target environment where the Delphix Engine toolkit can be installed (for example, `/var/tmp`) with the following properties:
   a. The toolkit directory must be writable by the operating system user mentioned above.
   b. The toolkit directory must have at least 256 MB of available storage.

6. There must be a mount point directory (for example, `/mnt/provision`) that will be used as the base for mount points that are created when provisioning a VDB with the following properties:
   a. The mount point directory must be writable by the operating system user mentioned above.
b. The mount point directory should be empty.

7. TCP/IP connectivity to and from the source environment must be configured as described in General Network and Connectivity Requirements.

Related Links

- Using HostChecker to Confirm Source and Target Environment Configuration
- sudoers Manual Page
Supported Operating Systems and Database Versions for PostgreSQL Environments

This topic describes supported operating systems and database versions for PostgreSQL.

**Source and Target OS and DBMS Compatibility**

The source and target environments must be running the same DBMS/Operating System combination (for example, PostgreSQL 9.2 on RHEL 6.3) in order to successfully perform linking and provisioning.

**Supported DBMS Versions**

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Version</th>
<th>Processor Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostgreSQL</td>
<td>9.2</td>
<td>x86_64</td>
</tr>
<tr>
<td>EnterpriseDB Postgres Plus Advanced Server</td>
<td>9.2</td>
<td>x86_64</td>
</tr>
</tbody>
</table>

**Supported Operating Systems**

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Version</th>
<th>Processor Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux 5</td>
<td>RHEL 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11</td>
<td>x86_64</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 6</td>
<td>RHEL 6.0, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6</td>
<td>x86_64</td>
</tr>
</tbody>
</table>
Network and Connectivity Requirements for PostgreSQL Environments

- **General Port Allocation**
  - General Outbound from the Delphix Engine Port Allocation
  - General Inbound to the Delphix Engine Port Allocation
- **Firewalls and Intrusion Detection Systems (IDS)**
- **SSHD Configuration**

Connection Requirements for PostgreSQL Environments

Port Allocation for PostgreSQL Environments

- **Outbound from the Delphix Engine Port Allocation**
- **Inbound to the Delphix Engine Port Allocation**
- **Port Allocation Between Source and Staging Target Environments**

**General Port Allocation**

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

**General Outbound from the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>25</td>
<td>Connection to a local SMTP server for sending email</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>53</td>
<td>Connections to local DNS servers</td>
</tr>
<tr>
<td>UDP</td>
<td>123</td>
<td>Connection to an NTP server</td>
</tr>
<tr>
<td>UDP</td>
<td>162</td>
<td>Sending SNMP TRAP messages to an SNMP Manager</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
<td>SSL connections from the Delphix Engine to the Delphix Support upload server</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>636</td>
<td>Secure connections to an LDAP server</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Connections to a Delphix replication target (see Configuring Replication)</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections to source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>

**General Inbound to the Delphix Engine Port Allocation**
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>80</td>
<td>HTTP connections to the Delphix GUI</td>
</tr>
<tr>
<td>UDP</td>
<td>161</td>
<td>Messages from an SNMP Manager to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>443</td>
<td>HTTPS connections to the Delphix GUI</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections from source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>32768 - 65535</td>
<td>Required for NFS mountd and status services from target environment only if the firewall between Delphix and the target environment does not dynamically open ports. <strong>Note:</strong> If no firewall exists between Delphix and the target environment, or the target environment dynamically opens ports, this port range is not explicitly required.</td>
</tr>
</tbody>
</table>

**Firewalls and Intrusion Detection Systems (IDS)**

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

**SSHD Configuration**

Both source and target Unix environments are required to have sshd running and configured such that the Delphix
Engine can connect over ssh.

The Delphix Engine expects to maintain long-running, highly performant ssh connections with remote Unix environments. The following sshd configuration entries can interfere with these ssh connections and are thus disallowed:

<table>
<thead>
<tr>
<th>Disallowed sshd Configuration Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientAliveInterval</td>
</tr>
<tr>
<td>ClientAliveCountMax</td>
</tr>
</tbody>
</table>

Connection Requirements for PostgreSQL Environments

- The Delphix Engine uses an SSH connection to each source environment and PostgreSQL client connections to the PostgreSQL instances on the source environment.
- The Delphix Engine uses an SSH connection to each target environment, NFS connections from each target environment to the Delphix Engine, and PostgreSQL client connections to the virtual databases on the target environment.
- Once connected to a staging target environment through SSH, the Delphix Engine initiates a PostgreSQL replication client connection from the target environment to the source environment.

Port Allocation for PostgreSQL Environments

Refer to Setting Up PostgreSQL Environments: An Overview for information on PostgreSQL environments. The Delphix Engine makes use of the following network ports for PostgreSQL dSources and VDBs:

**Outbound from the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to source and target database environments</td>
</tr>
<tr>
<td>TCP</td>
<td>xxx</td>
<td>PostgreSQL client connections to the PostgreSQL instances on the source and target environments (port 5432 by default)</td>
</tr>
</tbody>
</table>

**Inbound to the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/UDP</td>
<td>111</td>
<td>Remote Procedure Call (RPC) port mapper used for NFS mounts</td>
</tr>
<tr>
<td>TCP</td>
<td>1110</td>
<td>Network Status Monitor (NSM) client from target hosts to the Delphix Engine</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>2049</td>
<td>NFS client from target hosts to the Delphix Engine</td>
</tr>
</tbody>
</table>
TCP | 4045 | Network Lock Manager (NLM) client from target hosts to the Delphix Engine

UDP | 33434 - 33464 | Traceroute from source and target database servers to the Delphix Engine (optional)

### Port Allocation Between Source and Staging Target Environments

<table>
<thead>
<tr>
<th>Outgoing</th>
<th>Incoming</th>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Environment</td>
<td>Source Environment</td>
<td>PostgreSQL replication client</td>
<td>xxx</td>
<td>PostgreSQL replication client connection to the PostgreSQL instances on the source environment (port 5432 by default)</td>
</tr>
</tbody>
</table>
Managing PostgreSQL Environments

These topics describe special tasks and concepts for working with PostgreSQL environments.

- Setting Up PostgreSQL Environments: An Overview
- Using HostChecker to Validate PostgreSQL Source and Target Environments
- Adding a PostgreSQL Environment
- Adding an Installation to a PostgreSQL Environment
- Adding a Database Cluster to a PostgreSQL Environment
- Editing PostgreSQL Environment Attributes
- Managing PostgreSQL Environment Users
- Deleting a PostgreSQL Environment
- Refreshing a PostgreSQL Environment
- Enabling Staging, Linking and Provisioning for PostgreSQL Environments
- Changing the Host Name or IP Address for PostgreSQL Source and Target Environments
Setting Up PostgreSQL Environments: An Overview

This topic describes the high-level process for adding PostgreSQL environments, linking PostgreSQL data sources to the Delphix Engine, and provisioning virtual databases from PostgreSQL data sources.

Types of PostgreSQL Environments

At a high level, the Delphix Engine maintains an internal representation of a data source, from which one can provision virtual databases (VDBs). In order to link a data source and provision a VDB, the following types of environments are required:

A **source environment** is where the unvirtualized source database runs. The Delphix Engine uses the backup, restore, and replication features of the PostgreSQL DBMS to maintain its internal representation of the source database, to be used for provisioning VDBs. The Delphix Engine must be able to connect to the source environment in order to discover running source databases and to orchestrate the backup, restore, and replication functionality necessary to keep its representation synchronized with the source database. The Delphix Engine is designed to have a minimal impact on the performance of the source database and the source environment.

A **target environment** is where virtualized databases run. PostgreSQL target environments serve two purposes:

1. Since PostgreSQL does not provide a native incremental backup API, a warm standby server (in other words, one in log-shipping mode) must be created with all database files stored on the Delphix Engine for each source database. We refer to the creation and maintenance of this **staging database** as validated sync. During validated sync, we retrieve data from the source and ensure that all the components necessary for provisioning a VDB have been validated. The result of validated sync is both a TimeFlow with consistent points from which you can provision a VDB, and a faster provisioning process, because there is no need for any database recovery when provisioning a VDB. In order to create a staging database, you must designate a target environment for this task when linking a dSource. During the linking process, database files are exported over the network to the target environment, where the staging database instance runs as a warm standby server. A target environment that hosts one or more staging databases is referred to as a **staging target** for validated sync.

2. Once a staging database has been set up, you can provision **virtual databases** from any point in time along the TimeFlow mentioned above to any compatible target environment (for more information, see [Requirements for PostgreSQL Target Hosts and Databases](#)). Database files are exported over the network to the target environment, where the virtual database instance runs.

Workflow for PostgreSQL Environments

Prior to linking a data source, you must add both a source environment and a compatible target environment (to be used for the staging database mentioned above) to the Delphix Engine. Prior to provisioning a virtual database, you must add a compatible target environment to the Delphix Engine. This may be the same target environment as that used for the staging instance, or it may be a different target environment.

Once an environment is added to the Delphix Engine, environment discovery takes place. Environment discovery is the process of enumerating PostgreSQL installations and configurations when a source or target environment is added to the Delphix Engine. We also repeat the discovery process during environment refresh in order to detect new PostgreSQL installations and clusters. Environment objects can be added manually if discovery is not possible due to non-standard setup.
Using HostChecker to Validate PostgreSQL Source and Target Environments

This topic describes how to use HostChecker to configure PostgreSQL environments.

- **What is HostChecker?**
- **Prerequisites**
- **Procedure**
- **Tests Run**
- **Related Links**

What is HostChecker?

The HostChecker is a standalone program which validates that host machines are configured correctly before the Delphix Engine uses them as data sources and provision targets.

Please note that HostChecker does not communicate changes made to hosts back to the Delphix Engine. If you reconfigure a host, you must refresh the host in the Delphix Engine in order for it to detect your changes.

You can run the tests contained in the HostChecker individually, or all at once. You must run these tests on both the source and target hosts to verify their configurations. As the tests run, you will either see validation messages that the test has completed successfully, or error messages directing you to make changes to the host configuration.

Prerequisites

- Make sure that your source and target environments meet the requirements specified in PostgreSQL Support and Requirements.

Procedure

1. Download the HostChecker tarball: delphix_4.0.2.0_2014-04-29-08-38.hostchecker.tar
2. Create a working directory and extract the HostChecker files from the HostChecker tarball.
   ```
   mkdir dlpx-host-checker
   cd dlpx-host-checker/
   tar -xf delphix_4.0.2.0_2014-04-29-08-38.hostchecker.tar
   ```
3. Change to the working directory and enter this command. Note that for the target environments, you would change source to target.
   ```
   $ ./chkHost.pl source postgresql
   ```
   - Don’t Run as Root

   Don’t run the HostChecker as root; this will cause misleading or incorrect results from many of the checks.

4. Select which checks you want to run. We recommend you run all checks if you are running Hostchecker for the first time.
5. Pass in the arguments the checks ask for.
6. Read the output of the check.
7. The error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Don't be surprised or undo your work if more errors appear the next time you run HostChecker, because the error you just fixed may have been masking other problems.

8. Repeat steps 3–7 until all the checks return no errors or warnings.

### Tests Run

<table>
<thead>
<tr>
<th>Test</th>
<th>PostgreSQL Source</th>
<th>PostgreSQL Target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Host SSH Connectivity</td>
<td>X</td>
<td>X</td>
<td>Verifies that the environment is accessible via SSH</td>
</tr>
<tr>
<td>Check Tool Kit Path</td>
<td>X</td>
<td>X</td>
<td>Verifies that the toolkit installation location is has the proper ownership, proper permissions, and enough free space.</td>
</tr>
<tr>
<td>Check Home Directory Permissions</td>
<td>X</td>
<td>X</td>
<td>Verifies that the environment can be accessed via SSH using public key authentication. If you don't need this feature, you can ignore the results of this check.</td>
</tr>
<tr>
<td>Check OS User Privileges</td>
<td></td>
<td>X</td>
<td>Verifies that the operating system user can execute certain commands with necessary privileges via sudo. This only needs to be run on target environments. See the topic Requirements for PostgreSQL Target Hosts and Databases for more information.</td>
</tr>
<tr>
<td>Check PostgreSQL OS compatibility</td>
<td>X</td>
<td>X</td>
<td>Verifies that the environment is running a compatible operating system. See the topic Supported Operating Systems and Database Versions for PostgreSQL Environments for more information.</td>
</tr>
<tr>
<td>Check PostgreSQL installations</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Attempts to discover existing PostgreSQL installations and validate that they are of a compatible version and that each instance meets the requirements for PostgreSQL source databases. See the topics [Requirements for PostgreSQL Source Hosts and Databases](#) and [Supported Operating Systems and Database Versions for PostgreSQL Environments](#) for more information.

**Related Links**
- [PostgreSQL Support and Requirements](#)
Adding a PostgreSQL Environment

This topic describes how to add a PostgreSQL source environment to the Delphix Engine.

Prerequisites

Make sure your environment meets the requirements described in the following topics:

- Requirements for PostgreSQL Source Hosts and Databases
- Requirements for PostgreSQL Target Hosts and Databases
- Supported Operating Systems and Database Versions for PostgreSQL Environments

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Environments.
3. Next to Environments, click the green Plus icon.
4. In the Add Environment dialog, select Unix/Linux in the operating system menu.
5. Select Standalone Host.
6. Enter the Host IP address.
7. Enter an optional Name for the environment.
8. Enter the SSH port.
   The default value is 22.
9. Enter a Username for the environment.
   See Requirements for PostgreSQL Target Hosts and Databases and Requirements for PostgreSQL Source Hosts and Databases for more information about the environment user requirements.
10. Select a Login Type.
    For Password, enter the password associated with the user in Step 9.

   ![Using Public Key Authentication](image)
   If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      ii. Run chown 755 ~ to make your home directory writable only by your user.
   The public key needs to be added only once per user and per environment.
   You can also add public key authentication to an environment user’s profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

11. For Password Login, click Verify Credentials to test the username and password.
12. Enter a Toolkit Path.
    See Requirements for PostgreSQL Target Hosts and Databases and Requirements for PostgreSQL Source Hosts and Databases for more information about the toolkit directory requirements.
13. Click **OK**.
   As the new environment is added, you will see two jobs running in the Delphix Admin Job History, one to **Create and Discover** an environment, and another to **Create** an environment. When the jobs are complete, you will see the new environment added to the list in the **Environments** panel. If you don't see it, click the **Refresh** icon in your browser.

**Post-Requisites**
- After you create the environment, you can view information about it by selecting **Manage > Environments**, and then select the environment name.

**Related Links**
- [Setting Up PostgreSQL Environments: An Overview](#)
- [Requirements for PostgreSQL Source Hosts and Databases](#)
- [Requirements for PostgreSQL Target Hosts and Databases](#)
- [Supported Operating Systems and Database Versions for PostgreSQL Environments](#)
- [Adding an Installation to a PostgreSQL Environment](#)
Adding an Installation to a PostgreSQL Environment

This topic describes how to add an installation to a PostgreSQL environment.

When you add an environment with the Delphix Admin application, all PostgreSQL installations on it are automatically discovered. However, if an installation is not automatically discovered, you can add it manually to the environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Click the green Plus icon next to Add Dataset Home.
5. Under Dataset Home Type, select PostgreSQL.
6. Enter the path to the Installation.
7. Click the Check icon when finished.

Related Links

- Adding a Database Cluster to a PostgreSQL Environment
Adding a Database Cluster to a PostgreSQL Environment

This topic describes how to add a database cluster to a PostgreSQL environment.

When you add an environment with the Delphix Admin application, all database clusters on it are automatically discovered. However, if a database cluster is not automatically discovered, you can add it manually to the environment.

Prerequisites

- Make sure your source database meets the requirements described in Requirements for PostgreSQL Source Hosts and Databases and Requirements for PostgreSQL Target Hosts and Databases.
- Before adding a database, the installation of the database must exist in the environment. If the installation does not exist in the environment, follow the steps in Adding an Installation to a PostgreSQL Environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Choose the installation which has been used to start the database cluster.
   - Click the Up icon next to the the installation path to show details if needed.
5. Click the green Plus icon next to Add DB Cluster.
6. Enter the Path of the data cluster directory.
7. Click the Check icon when finished.

Related Links

- Requirements for PostgreSQL Source Hosts and Databases
- Requirements for PostgreSQL Target Hosts and Databases
- Adding an Installation to a PostgreSQL Environment
Editing PostgreSQL Environment Attributes

- **Procedure**
  - **Common Editable Attributes**
  - **PostgreSQL Attributes**

This topic describes how to edit attributes of an environment such as name, host address, ssh port, or toolkit path, as well as descriptions of more advanced attributes for specific data platforms.

**Procedure**

1. Log in to the Delphix Admin application with Delphix Admin credentials or as the owner of an environment.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of an environment to view its attributes.
4. Under Attributes, click the Pencil icon to edit an attribute.
5. Click the Check icon to save your edits.

**Common Editable Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment Users</td>
<td>The users for that environment. These are the users who have permission to ssh into an environment, or access the environment through the Delphix Connector. See the Requirements topics for specific data platforms for more information on the environment user requirements.</td>
</tr>
<tr>
<td>Host Address</td>
<td>The IP address for the environment host.</td>
</tr>
<tr>
<td>Notes</td>
<td>Any other information you want to add about the environment.</td>
</tr>
</tbody>
</table>

**PostgreSQL Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH Port</td>
<td>The port used for secure shell connection to the host.</td>
</tr>
<tr>
<td>Toolkit Path</td>
<td>The directory used for storing Delphix toolkit files.</td>
</tr>
</tbody>
</table>
Managing PostgreSQL Environment Users

This topic describes how to manage the users associated with an environment. For information on providing Delphix users with privileges for groups and database objects, see the topics under Managing Delphix Users and Managing Policies.

- **Prerequisites**
- **Procedure**

**Prerequisites**

Users that you add to an environment must meet the requirements for that environment as described in the platform-specific Requirements topics.

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Environments.
4. Click the name of an environment to open the environment information screen.
5. Under Basic Information, click the green Plus icon to add a user.
6. Enter the Username and Password for the OS user in that environment.

#### Using Public Key Encryption

If you want to use public key encryption for logging into your environment:

a. Select Public Key for the Login Type.

b. Click View Public Key.

c. Copy the public key that is displayed, and append it to the end of your 
   
   ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
   
   i. Run `chmod 600 authorized_keys` to enable read and write privileges for your user.
   
   ii. Run `chmod 755 ~` to make your home directory writable only by your user.

The public key needs to be added only once per user and per environment.

You can also add public key authentication to an environment user's profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

7. Click the Check icon to save the new user.
8. To change the primary user for this environment, click the Pencil icon next to Environment Users.
9. To delete a user, click the Trash icon next to their username.
Deleting a PostgreSQL Environment

This topic describes how to delete an environment.

Prerequisites

You cannot delete an environment that has any dependencies, such as dSources or VDBs. These must be deleted before you can delete the environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, select the environment you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.
Refreshing a PostgreSQL Environment

This topic describes how to refresh an environment.

After you make changes to an environment that you have already set up in the Delphix Admin application, such as installing a new database home, creating a new database, or adding a new listener, you may need to refresh the environment to reflect these changes.

Procedure

1. Log in to the Delphix Admin application with Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of the environment to you want to refresh.
4. Click the Refresh icon.

To refresh all environments, click the Refresh icon next to Environments.
Enabling Staging, Linking and Provisioning for PostgreSQL Environments

This topic describes how to enable and disable staging, provisioning and linking for databases.

Before a database can be used as a dSource, you must first make sure that you have defined a staging environment and enabled linking to it. Similarly, before you can provision a VDB to a target database, you must make sure that you have enabled provisioning to it.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Slide the button next to Use as Staging to Yes or No to enable or disable staging.
5. Slide the button next to Allow Provisioning to On or Off to enable or disable provisioning.
6. Slide the button next to Allow Linking to On or Off to enable or disable linking.
Changing the Host Name or IP Address for PostgreSQL Source and Target Environments

This topic describes how to change the host name or IP address for source and target environments, and for the Delphix Engine.

- **Procedure**
  - **For Source Environments**
  - **For VDB Target Environments**
  - **For the Delphix Engine**

**Procedure**

**For Source Environments**

1. Disable the dSource as described in [Enabling and Disabling dSources](#).
2. If the **Host Address** field contains an IP address, edit the IP address.
3. If the **Host Address** field contains a host name, update your Domain Name Server to associate the new IP address to the host name. The Delphix Engine will automatically detect the change within a few minutes.
4. In the **Environments** screen of the Delphix Engine, refresh the host.
5. Enable the dSource.

**For VDB Target Environments**

1. Disable the VDB as described in [Enabling and Disabling Virtual Databases](#).
2. If the **Host Address** field contains an IP address, edit the IP address.
3. If the **Host Address** field contains a host name, update your Domain Name Server to associate the new IP address to the host name. The Delphix Engine will automatically detect the change within a few minutes.
4. In the **Environments** screen of the Delphix Engine, refresh the host.
5. Enable the VDB.

**For the Delphix Engine**

1. Stop all running VDBs by clicking the red **Stop** button on the VDB card.
2. Disable all dSources as described in [Enabling and Disabling dSources](#).
3. You can use either the command line interface or the Server Setup application to change the IP address of the Delphix Engine.
   a. To use the command line interface, press F2 and follow the instructions described in [Setting Up Network Access to the Delphix Engine](#).
   b. To use the Server Setup application, go to to **System > Server Setup** in the Delphix Admin interface, or click **Server Setup** in the Delphix Engine login screen.
      i. In the **Network** panel, click **Modify**.
      ii. Under **DNS Services**, enter the new IP address.
      iii. Click **OK**.
4. Refresh all Environments by clicking the Blue/Green Refresh Symbol on the Environments screen.
5. Enable all dSources as described in [Enabling and Disabling dSources](#).
6. Start all VDBs by clicking the **Start** button on the VDB card.
Managing PostgreSQL Data Sources

These topics describe concepts and tasks for linking PostgreSQL data sources to the Delphix Engine.

- Linking PostgreSQL Data Sources: Overview
- Linking a PostgreSQL dSource
- Advanced Data Management Settings for PostgreSQL Data Sources
- Using Pre- and Post-Scripts with PostgreSQL dSources
- Enabling and Disabling PostgreSQL dSources
- Detaching and Re-Attaching PostgreSQL dSources
- Deleting a PostgreSQL dSource
- PostgreSQL dSource Icon Reference
Linking PostgreSQL Data Sources: Overview

This topic describes basic concepts behind the creation of dSources from PostgreSQL data sources.

Initial Linking and Staging Databases

A dSource is the copy of a physical database that is created when the Delphix Engine links to and loads the database. The Delphix Engine keeps the dSource in sync with the source database in order to facilitate the provisioning of Virtual Databases (VDBs) from the dSource's TimeFlow. Since PostgreSQL does not provide a native incremental backup API, a warm standby server (in other words, one in log-shipping mode) must be created with all database files stored on the Delphix Engine for each source database, as described in Setting Up PostgreSQL Environments: An Overview. We refer to the creation and maintenance of this staging database as validated sync, and a target environment that hosts one or more staging databases is referred to as a staging target for validated sync.

When you create a dSource from a PostgreSQL database, the Delphix Engine initiates a full database backup of the source database by running `pg_basebackup(1)` on the staging target. The initial snapshot of the dSource is derived from this backup.

After obtaining the initial snapshot and linking the dSource, the Delphix Engine keeps the dSource and the source database in sync by monitoring the source database for new transaction logs on the staging target, and then applying those transaction logs on the staging database. Transaction logs are fetched by running `pg_receivexlog(1)` on the staging target.

Target Environments for Staging and VDB Provisioning

It is possible to provision a VDB to the same source environment that contains the dSource, but performance and efficiency are improved if the dSource is located in one environment, and the VDB in another. A target environment can be used both for staging databases and for VDBs. The source and target environments must be running the same DBMS/Operating System combination (for example, PostgreSQL 9.2 on RHEL 6.3) in order to successfully link a dSource, as described in Supported Operating Systems and Database Versions for PostgreSQL Environments.

Related Links

- Setting Up PostgreSQL Environments: An Overview
- PostgreSQL Support and Requirements
Linking a PostgreSQL dSource

This topic describes the basic procedure for linking a dSource from a PostgreSQL database to the Delphix Engine.

Prerequisites
- Make sure you have the correct user credentials for the source environment, as described in Requirements for PostgreSQL Source Hosts and Databases
- You may also want to read the topic Advanced Data Management Settings for PostgreSQL Data Sources.

Procedure
1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > Add dSource. Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
3. In the Add dSource wizard, select the source database.
   - Changing the Environment User
     If you need to change or add an environment user for the source database, see Managing PostgreSQL Environment Users.
4. Enter your login credentials for DB Cluster User and DB Cluster Password.
5. Click Advanced to enter a Connection Database. The Connection Database will be used when issuing SQL queries from the Delphix Engine to the linked database. It can be any existing database that the DB Cluster User has permission to access.
6. Click Next.
7. Select a Database Group for the dSource, and then click Next. Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users, Permissions, and Policies for more information.
8. Select a SnapSync Policy, and, if necessary, a Staging Installation for the dSource. The Staging installation represents the PostgreSQL binaries that will be used on the staging target to backup and restore the linked database to a warm standby.
9. Click Advanced to select whether the data in the data sources is Masked, to select a Retention Policy, and to indicate whether any pre or post scripts should be executed during the dSource creation. See Advanced Data Management Settings for PostgreSQL Data Sources and Using Pre- and Post-Scripts with PostgreSQL dSources for more information.
10. Click Next.
11. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs, DB_Link and DB_Sync, to create the dSource. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have successfully completed, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of My Databases under its assigned group.
The dSource Card

After you have created a dSource, you can view information about it on the dSource card, and also make modifications to its policies and permissions. In the Databases panel, click on the Open icon to view the front of the dSource card. The card will then flip, showing you information such as the Source Database and Data Management configuration. See the topic Advanced Data Management Settings for PostgreSQL Data Sources for more information.

Related Links

- Advanced Data Management Settings for PostgreSQL Data Sources
- Requirements for PostgreSQL Target Hosts and Databases
- Using Pre- and Post-Scripts with PostgreSQL dSources
- Users, Permissions, and Policies
Advanced Data Management Settings for PostgreSQL Data Sources

- **Accessing Data Management Settings**
- **Retention Policies**
- **PostgreSQL LogSync Settings**
- **PostgreSQL SnapSync Policy Settings**
  - **Schedule By Settings**

This topic describes advanced data management settings for dSources

When linking a dSource, you can use default data management settings for Retention, sync validation, and LogSync policies. However, you can also use custom settings to improve overall performance and match the needs of your specific server and data environment.

**Accessing Data Management Settings**

There are three ways you can set or modify data management settings for dSources:

1. During the dSource linking process, click **Advanced** in the **Data Management** panel of the **Add dSource** wizard.
2. Click the **Edit** icon next to a policy or data management setting on the back of the dSource card. For **SnapSync** and **Retention** policies, click the policy name. This will open the **Policy Management** screen.
3. Select **Manage > Policies** in the top menu bar. This will open the **Policy Management** screen. Select the policy for the dSource you want to modify, and click **Modify**. See [Creating Custom Policies](#) and [Creating Policy Templates](#), for more information.

**Retention Policies**

Retention policies determine the length of time that snapshots and log files are retained. The retention time for logs must be equal to or longer than the retention time for snapshots. This policy, in combination with the SnapSync policy, can have a significant impact on the performance of the Delphix Engine.

**PostgreSQL LogSync Settings**

LogSync is always enabled by default for PostgreSQL dSources.

**PostgreSQL SnapSync Policy Settings**

* **Schedule By Settings**

In the default SnapSync policy setting, snapshots are taken daily at a set time, with a four hour period. You can modify the snapshot schedule and frequency by changing the **Schedule By** setting.
Using Pre- and Post-Scripts with PostgreSQL dSources

- Using Pre- and Post-Scripts with PostgreSQL dSources
- Specifying Arguments for PostgreSQL Scripts
  - An Example with Three Arguments
  - An Example with an Apostrophe

This topic describes the use of pre- and post-scripts with dSources.

For each script, you can add the script path and its arguments when linking a dSource in one of two ways. Either:

- During the Add dSource wizard process, in the Data Management screen, click Advanced.

or

- On the back of the dSource card, click the Pencil icon next to the Pre Script and Post Script fields.

To update pre- and post-script information:

1. Flip over the dSource card.
2. Click on the Pencil icon next to the Pre Script and Post Script fields.
3. When finished, click the check mark icon.

Note that pre-scripts are executed before the SnapSync policy, and if the script fails, SnapSync will fail as well. In the case of a post-script, the script will execute after SnapSync is complete. If a post-script fails, you will see an error message.

Using Pre- and Post-Scripts with PostgreSQL dSources

- For PostgreSQL dSources, pre- and post-scripts are run during initial sync and during redo-basebackup
- For PostgreSQL environments, scripts must exist and be readable on the source environment for initial sync and redo-basebackup
- PostgreSQL dSource scripts must be text scripts only

Specifying Arguments for PostgreSQL Scripts

You can specify multiple arguments for a script. In the Pre or Post Script field, enter the path to the script, and then list the arguments. If the argument contains spaces, enclose it in single or double quotes. You can escape single quotes within the argument with a backslash.

An Example with Three Arguments

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh one "second argument in double quotes" 'third argument in single quotes'
```

An Example with an Apostrophe

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh 'I\'d rather be in Hawaii.'
```
Enabling and Disabling PostgreSQL dSources

This topic describes how to enable and disable dSources for operations such as backup and restore.

For certain processes, such as backing up and restoring the source database, you may want to temporarily disable your dSource. Disabling a dSource turns off communication between the dSource and the source database, but does not tear down the configuration that enables communication and data updating to take place. When a disabled dSource is later enabled, it will resume communication and incremental data updates from the source database according to the original policies and data management configurations that you set.

Disabling a dSource is also a pre-requisite for several other operations, like database migration and upgrading the dSource after upgrade of the associated data source. See the topics in the Related Links section for more information.

Procedure

1. Go to Manage > Databases > My Databases.
2. Select the dSource to disable.
3. On the back of the dSource card, move the slider control from Enabled to Disabled.
4. Click Yes to acknowledge the warning.

When you are ready to enable the dSource again, move the slider control from Disabled to Enabled, and the dSource will continue to function as it did previously.
Detaching and Re-Attaching PostgreSQL dSources

- **Detaching a dSource**
- **Attaching a dSource**

This topic describes how to detach dSources and re-attach them to a different source database.

Each dSource contains an association with the source database, as well as the data it has pulled from the source database up to that point. It is possible to detach, or unlink, a dSource from its source database. This breaks the association with the source database without affecting the data within Delphix. Detached dSources and their source databases have these properties:

- Detached dSources can be used as the source of VDB provisioning operations
- The source database can be re-linked as a different dSource
- The dSource can be attached to a different source database and resume incremental SnapSync, provided the source database is an image of the same logical database (for example, a standby replica). Currently this is only supported through the CLI, and is most often used when performing an initial load from a standby replica and then later attaching the dSource to the production database for subsequent incremental syncs.

**Detaching a dSource**

1. Log into the Delphix Admin application as a user with OWNER privileges on the dSource, group, or domain.
2. Select **Manage > Databases > My Databases**.
3. Select the database you want to unlink or delete.
4. Click the **Unlink** icon.
   A warning message will appear. Click **Yes**

**Attaching a dSource**
Rebuilding Source Databases and Using VDBs

In situations where you want to rebuild a source database, you will need to detach the original dSource and create a new one from the rebuilt data source. However, you can still provision VDBs from the detached dSource.

1. Detach the dSource as described in this topic.
2. Re-name the detached dSource by clicking the Edit icon next to its name. This is necessary only if you intend to give the new dSource the same name as the original one. If you attempt to give the new dSource the same name as the detached one, you will see an error message.
3. Create the new dSource from the rebuilt database.

You will now be able to provision VDBs from both the detached dSource and the newly created one, but the detached dSource will only represent the state of the source database prior to being detached.

The attach operation is currently only supported in the CLI. Full GUI support will be added in a future release. Only databases that represent the same physical database can be re-attached.

1. Log into the Delphix CLI as a user with OWNER privileges on the dSource, group, or domain.
2. Select the dSource by name using `database select <dSource>`
3. Run the `attachSource` command.
4. Set the source config you want to attach to using `set source.config=<newSource>`. Source configs are named by their database unique name.
5. Set any other source configuration operations as you would for a normal link operation.
6. Run the `commit` command.

Attaching PostgreSQL dSource

Attaching PostgreSQL dSource requires a staging instance. This is specified by the `pptRepository` parameter under the `attachSource` command.
Deleting a PostgreSQL dSource

This topic describes how to delete a dSource.

Prerequisites

- You cannot delete a dSource that has dependent VDBs. Before deleting a dSource, make sure all dependent VDBs have been deleted as described in Deleting a VDB.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. In the Databases panel, select the dSource you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.

⚠️ Deleting a dSource will also delete all snapshots, logs, and descendant VDB Refresh policies for that database, and the deletion cannot be undone
**PostgreSQL dSource Icon Reference**

This topic illustrates the icons that appear on dSources and Virtual Databases (VDBs) in the Delphix Engine Graphic User Interface, and describes the meaning of each, along with tips for clearing those that represent errors.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Exclamation Mark" /></td>
<td>There is a critical fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image2.png" alt="Warning Triangle" /></td>
<td>There is a warning fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Hourglass" /></td>
<td>The Delphix Engine is checking the VDB status.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Error" /></td>
<td>The dSource has been deleted or the Source status is UNKNOWN.</td>
</tr>
<tr>
<td><img src="image5.png" alt="Question Mark" /></td>
<td>The state of the VDB is unknown. This is often associated with a connection error.</td>
</tr>
<tr>
<td><img src="image6.png" alt="Circle" /></td>
<td>The VDB is inactive.</td>
</tr>
<tr>
<td><img src="image7.png" alt="Unlock" /></td>
<td>The dSource has been unlinked from the source database.</td>
</tr>
<tr>
<td><img src="image8.png" alt="Disassembled" /></td>
<td>The VDB is disabled, is in the process of being created, or the creation process has been canceled or failed. See <a href="#">Enabling and Disabling VDBs</a> for more information.</td>
</tr>
<tr>
<td><img src="image9.png" alt="Normal" /></td>
<td>The VDB is running normally</td>
</tr>
<tr>
<td><img src="image10.png" alt="Suspend" /></td>
<td>The dSource is disabled. See <a href="#">Enabling and Disabling dSources</a> for more information.</td>
</tr>
<tr>
<td><img src="image11.png" alt="Ready" /></td>
<td>The dSource or VDB is ready for Linux Transformation.</td>
</tr>
</tbody>
</table>
Provisioning VDBs from PostgreSQL dSources

These topics describe concepts and tasks for provisioning virtual databases (VDBs) from PostgreSQL dSources.

- Provisioning PostgreSQL VDBs: Overview
- Provisioning a PostgreSQL VDB
- Enabling and Disabling PostgreSQL VDBs
- Refreshing a PostgreSQL VDB
- Deleting a PostgreSQL VDB
- Migrating a PostgreSQL VDB
- Provisioning a PostgreSQL VDB from a Replicated dSource or VDB
- PostgreSQL VDB Icon Reference
- Customizing PostgreSQL VDB Configuration Settings
Provisioning PostgreSQL VDBs: Overview

This topic describes the basic concepts involved with provisioning a virtual database (VDB) from a PostgreSQL dSource.

A dSource is a virtualized representation of a physical or logical source database. As a virtual representation, it cannot be accessed or manipulated using database tools. Instead, you must create a virtual database (VDB) from a dSource snapshot. A VDB is an independent, writeable copy of a dSource snapshot. You can also create VDBs from other VDBs. Once a VDB has been provisioned to a target environment, you can also implement a snapshot policy for that VDB, to capture changes within it as if it were any other logical or physical database.

For an overview of the high-level components involved in provisioning a PostgreSQL VDB, see to Setting Up PostgreSQL Environments: An Overview.

Validated Sync and LogSync

When you link a source database into the Delphix Engine, you must also specify a target environment that will host a staging database for the validated sync process, as described in Setting Up PostgreSQL Environments: An Overview. In this process, the Delphix Engine continuously monitors the source database for new transaction logs. When it detects one, it applies that transaction log to the staging database. The result is a TimeFlow.

Snapshots accumulate over time, and are available when you select the dSource in the My Databases panel of the Delphix Admin application. Each snapshot is represented as a card that includes information about the source database, operating system, end stamp, and snapshot database change number (LSN). You can scroll through these cards to select the one you want, or enter a date and time to search for a specific snapshot.

If LogSync is enabled, you will also see a LogSync slider control at the top of the snapshot card. If you slide this control to the right to open LogSync, you will see a pointer that you can move along a timeline to select the exact time from which you want to provision a VDB.

Once a VDB has been provisioned, you can also take snapshots of it. As with the dSource snapshots, you can find these when you select the VDB in the My Databases panel. You can then provision additional VDBs from these VDB snapshots.

Target Environments for VDBs

It is possible to provision a VDB to the same source environment that contains the dSource, but performance and efficiency are improved if the dSource is located in one environment, and the VDB in another. A target environment may be used both for staging databases and for VDBs. The source and target environments must be running the same DBMS/Operating System combination (for example, PostgreSQL 9.2 on RHEL 6.3) in order to successfully provision a VDB, as described in Supported Operating Systems and Database Versions for PostgreSQL Environments.

Customizing VDB Configuration Settings and File Paths

When you provision a VDB, you have the option of customizing its configuration settings, and the file paths, that it will use on the target environment. During the provisioning process, you can see the default configuration settings and file paths by clicking the Advanced link in the Target Environment screen of the VDB Provisioning Wizard.

Related Links

- Setting Up PostgreSQL Environments: An Overview
- Requirements for PostgreSQL Target Hosts and Databases
• **Supported Operating Systems and Database Versions for PostgreSQL Environments**

• **Provisioning a PostgreSQL VDB**
Provisioning a PostgreSQL VDB

This topic describes how to provision a virtual database (VDB) from a PostgreSQL dSource.

Prerequisites

- You will need to have linked a dSource from a source database, as described in Linking a PostgreSQL dSource, or have already created a VDB from which you want to provision another VDB

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a dSource snapshot.
   - See Provisioning by Snapshot and LogSync in this topic for more information on provisioning options.
   - You can take a snapshot of the dSource to provision from by clicking the Camera icon on the dSource card.
5. Optional: Slide the LogSync slider to the open the snapshot timeline, and then move the arrow along the timeline to provision from a point in time within a snapshot.
6. Click Provision.
   - The VDB Provisioning Wizard will open, and the fields Installation, Mount Base, and Environment User will auto-populate with information from the environment configuration.
7. Enter a Port Number.
   - The TCP port upon which the VDB will listen.
8. Click Advanced to enter any VDB configuration settings.
   - See Customizing PostgreSQL VDB Configuration Settings for more information.
9. Click Next to continue to the VDB Configuration tab.
10. Modify the VDB Name if necessary.
11. Select a Target Group for the VDB.
12. Click the green Plus icon to add a new group, if necessary.
13. Select a Snapshot Policy for the VDB.
14. Click the green Plus icon to create a new policy, if necessary.
15. Click Next to continue to the Hooks tab.
16. Specify any Hooks to be used during the provisioning process.
   - See Customizing PostgreSQL Management with Hook Operations for more information.
17. Click Next to continue to the Summary tab.
18. Click Finish.
   - When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the VDB will be included in the group you designated, and listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.

Provisioning by Snapshot or LogSync
When provisioning by snapshot, you can provision to the start of any snapshot by selecting that snapshot card from the Timeflow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.

When provisioning by LogSync information, you can provision to any point in time within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the Slide to Open LogSync control at the top of an individual snapshot card to view the time range within that snapshot. Drag the red triangle to the point in time that you want to provision from. You can also enter a date and time directly.

Related Links

- Linking a PostgreSQL dSource
- Requirements for PostgreSQL Target Hosts and Databases
- Using Pre- and Post-Scripts with dSources and VDBs
- Customizing PostgreSQL VDB Configuration Settings
Enabling and Disabling PostgreSQL VDBs

This topic describes how to enable and disable staging, provisioning and linking for databases.

Before a database can be used as a dSource, you must first make sure that you have defined a staging environment and enabled linking to it. Similarly, before you can provision a VDB to a target database, you must make sure that you have enabled provisioning to it.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Slide the button next to Use as Staging to Yes or No to enable or disable staging.
5. Slide the button next to Allow Provisioning to On or Off to enable or disable provisioning.
6. Slide the button next to Allow Linking to On or Off to enable or disable linking.
Refreshing a PostgreSQL VDB

- **Prerequisites**
- **Procedure**

This topic describes how to manually refresh a VDB.

Refreshing a VDB will re-provision the VDB from the dSource. As with the normal provisioning process, you can choose to refresh the VDB from a snapshot or a specific point in time. However, you should be aware that refreshing a VDB will delete any changes that have been made to it over time. When you refresh a VDB, you are essentially re-setting it to the state you select during the refresh process. You can refresh a VDB manually, as described in this topic, or you can set a VDB refresh policy, as described in the topics Managing Policies: An Overview, Creating Custom Policies, and Creating Policy Templates.

Although the VDB no longer contains the previous contents, the previous Snapshots and TimeFlow still remain in Delphix and are accessible through the Command Line Interface (CLI).

### Prerequisites

To refresh a VDB, you must have the following permissions:

1. **Auditor** permissions on the dSource associated with the VDB.
2. **Auditor** permissions on the group that contains the VDB.
3. **Owner** permissions on the VDB itself.

A user with Delphix Admin credentials can perform a VDB Refresh on any VDB in the system.

### Procedure

1. Log into the Delphix Admin application.
2. Under Databases, select the VDB you want to refresh, and then click the Open icon to open its card.
3. On the back of the VDB card, click the Refresh VDB icon in the lower right corner. This will open the screen to re-provision the VDB.
4. Select the refresh point as a snapshot or a point in time.
5. Click Refresh VDB.
6. Click Provide Privileged Credentials if you want to use login credentials on the target environment other than those associated with the Environment User.
7. Click Yes to confirm.
Deleting a PostgreSQL VDB

This topic describes how to delete a VDB.

Procedure
1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. Select the VDB you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.
Migrating a PostgreSQL VDB

This topic describes how to migrate a Virtual Database (VDB) from one target environment to another.

There may be situations in which you want to migrate a virtual database to a new target environment, for example when upgrading the host on which the VDB resides, or as part of a general data center migration. This is easily accomplished by first disabling the database, then using the Migrate VDB feature to select a new target environment.

- **Prerequisites**
- **Procedure**
- **Video**

**Prerequisites**

- You should have already set up a new target environment that is compatible with the VDB that you want to migrate.

**Procedure**

1. Log into your Delphix Engine using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. Select the VDB you want to migrate, and then click the Open icon.
4. Slide the Enable/Disable control to Disabled, and click Yes to confirm. When the VDB is disabled, its icon will turn grey.
5. On the bottom-right corner of the VDB card, click the VDB Migrate icon.
6. Select the new target environment for the VDB, the user for that environment, and the database installation where the VDB will reside.
7. Click the Check icon to confirm your selections.
8. Slide the Enable/Disable control to Enabled, and click Yes to confirm. Within a few minutes your VDB will re-start in the new environment, and you can continue to work with it as you would any other VDB.

**Video**
Provisioning a PostgreSQL VDB from a Replicated dSource or VDB

This topic describes how to provision from a replicated dSource or virtual database (VDB). The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the replica containing the replicated object.

- **Prerequisites**
- **Procedure**
- **Post-Requisites**

**Prerequisites**

- You must have replicated a dSource or a VDB to the target host, as described in [Replication Overview](#).
- You must have added a compatible target environment on the target host.

**Procedure**

1. Log into the Delphix Admin application for the target host.
2. Select Manage > Databases > My Databases.
3. In the list of replicas, select the replica that contains the dSource or VDB you want to provision.
4. The provisioning process is now identical to the process for provisioning standard objects.

**Post-Requisites**

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.
## PostgreSQL VDB Icon Reference

This topic illustrates the icons that appear on dSources and Virtual Databases (VDBs) in the Delphix Engine Graphic User Interface, and describes the meaning of each, along with tips for clearing those that represent errors.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="Icon" /></td>
<td>There is a critical fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>There is a warning fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The Delphix Engine is checking the VDB status.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The dSource has been deleted or the Source status is UNKNOWN.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The state of the VDB is unknown. This is often associated with a connection error.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The VDB is inactive.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The dSource has been unlinked from the source database.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The VDB is disabled, is in the process of being created, or the creation process has been canceled or failed. See <a href="#">Enabling and Disabling VDBs</a> for more information.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The VDB is running normally</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The dSource is disabled. See <a href="#">Enabling and Disabling dSources</a> for more information.</td>
</tr>
<tr>
<td><img src="" alt="Icon" /></td>
<td>The dSource or VDB is ready for Linux Transformation.</td>
</tr>
</tbody>
</table>
Customizing PostgreSQL VDB Configuration Settings

This topic describes how to customize VDB configuration settings, including settings that are reserved by the Delphix Engine, those that are removed from the database configuration file during the provisioning process, and those that can be customized.

VDB Configuration

When you create a VDB, configuration settings are copied from the dSource and used to create the VDB. Most settings are copied directly, and you can see these settings by clicking the Advanced link in the Target Environment screen in the VDB Provisioning Wizard. When a VDB is provisioned, you can specify configuration parameters directly. It is important to know, however, that some configuration parameters are not customizable, and some are stripped out during the provisioning process but are customizable. The list of restricted and customizable parameters can be found below.

VDB Access Control

By default, VDBs are provisioning with a pg_hba.conf file that only permits local connections to the VDB. To enable remote connections to VDBs, customize the pg_hba.conf settings in the provisioning wizard.

Restricted Parameters

These parameters are restricted for use by the Delphix Engine. Attempting to customize these parameters will cause an error during the provisioning process.

- archive_command
- archive_mode
- wal_level
- port
- data_directory
- config_file
- hba_file
- ident_file
- max_stack_depth
- wal_segment_size
- block_size
- lc_ctype
- segment_size
- wal_block_size
- lc_collate
- server_version
- integer_datetimes
- server_encoding
- server_version_num
- max_identifier_length
- max_index_keys
- max_function_args
- include
- include_if_exists
Customizing PostgreSQL Management with Hook Operations

Hook operations allow you to execute an ordered list of custom operations as part of the dSource syncing, provisioning, and refresh process. For details on the types of operations that are available, see children of this page.

**dSource Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Sync</td>
<td>Operations performed before a sync. These operations can quiesce data to capture more information into the sync or stop applications that may interfere with the sync.</td>
</tr>
<tr>
<td>Post-Sync</td>
<td>Operations performed after a sync. These operations can undo any quiescing of data or undo any changes made by the Pre-Sync hook.</td>
</tr>
</tbody>
</table>

**Virtual Dataset Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Clone</td>
<td>Operations performed after initial provision or refresh of a virtual dataset. This hook runs when the virtual dataset has been started. During a refresh, the Delphix Engine reaches this hook before the Post-Refresh hook.</td>
</tr>
<tr>
<td>Pre-Refresh</td>
<td>Operations performed before a refresh. These operations can back up any data or configurations from the running source before doing the refresh.</td>
</tr>
<tr>
<td>Post-Refresh</td>
<td>Operations performed after a refresh. You can use these operations to restore any data or configurations backed up by the Pre-Refresh hook. During a refresh, the Delphix Engine runs this after the Configure Clone hook.</td>
</tr>
</tbody>
</table>

**Operation Failure**

If a hook operation fails, it will fail the entire sync, provision, or refresh job. The job will halt immediately and no further hook operations will be run.

**Hook Operations on Cluster Environments**

When linking from, or provisioning to, cluster environments such as Oracle RAC environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine picks a node in the cluster at random and guarantees all operation within any single hook will execute serially on this node. Note that the Delphix Engine does not guarantee the same node is chosen for the execution of every hook.

**Setting Hook Operations**

You can construct hook operation lists through the Delphix Admin application or the command line interface (CLI).
You can either define the operation lists as part of the linking or provisioning process or edit them on dSources or virtual datasets that already exist.

**Setting Hook Operations through the Delphix Admin Application**

To specify hook operations during linking or provisioning, navigate to the **Hooks** tab of the **Linking Wizard** or **Provision Wizard**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Select the **type of operation** or click **Import** to load a hook operation template.
4. Click the **text area** and edit the contents of the operation.
5. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
6. To remove an operation from the list, click the **Trash** icon on the operation.
7. When you have set all hook operations, click **Next** to continue with the provisioning process.

To edit hook operations on an already-existing dSource or virtual dataset, navigate to the **Hooks** tab on the back of the **dSource card** or **virtual dataset card**.

1. Select the **hook** to edit.
2. The current operations at this hook will be displayed. To edit this list of operations, click the **Pencil** icon in the top right-hand corner of the card.
3. Click the **Plus** icon to add a new operation.
4. Select the **type of operation** or click **Import** to load a hook operation template.
5. Click the **text area** and edit the contents of the operation.
6. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
7. To remove an operation from the list, click the **Trash** icon on the operation.
8. When you have set all hook operations, click **Check** to save the changes.

**Setting Hook Operations through the CLI**

To specify hook operations during linking, edit the relevant hook's array of operations defined on the **LinkingParameters > Source > Operations** object.

To specify hook operations during provisioning, edit the relevant hook's array of operations defined on the **ProvisionParameters > Source > Operations** object.

To edit hook operations on an already-created dSource, edit the relevant hook's array of operations defined on the **Source > Operations** object.

To edit hook operations on an already-created virtual dataset, edit the relevant hook's array of operations defined on the **Source > Operations** object.

For more information about these CLI objects, see the **LinkedSourceOperations, VirtualSourceOperations, RunCommandOnSourceOperation, and RunExpectOnSourceOperation** API documentation in the **Help** menu of the **Delphix Admin** application.

**Example of Editing Hook Operations through the CLI**

1. Navigate to relevant source's **VirtualSourceOperations** object.
2. Select a hook to edit.

```bash
delphix> source
delphix source> select "pomme"
delphix source "pomme"> update
delphix source "pomme" update *> edit operations
delphix source "pomme" update operations *> edit postRefresh
```

3. Add an operation at index 0.

```bash
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 0 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 0 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 0 *> ls
Properties
    type: RunCommandOperation (*)
    command: echo Refresh completed. (*)
delphix source "pomme" update operations postRefresh 0 *> commit
```

4. Add another operation at index 1 and then delete it.

```bash
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 1 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 1 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 1 *> back
delphix source "pomme" update operations postRefresh *> unset 1
delphix source "pomme" update operations postRefresh *> commit
```

**Hook Operation Templates**

You can use templates to store commonly-used operations, which allows you to avoid repeated work when an operation is applicable to more than a single dSource or virtual dataset. You manage templates through the Delphix Admin application.

**Creating a Hook Operation Template**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Operation Templates.
4. Click the Plus icon to add a new operation template.
5. Enter a Name for the template.
6. Select an operation Type.
7. Enter a Description detailing what the operation does or how to use it.
8. Enter operation Contents to implement the operation partially or fully.
9. Click Create.

**Importing a Hook Operation Template**
To import a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Click Import.
4. Select the template to import.
5. Click Import.
6. When you have set all hook operations, click Check to save the changes.

Exporting a Hook Operation Template

To export a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Select the type of operation.
4. Click the text area and edit the contents of the operation.
5. Click Export.
6. Enter a Name for the template.
7. Enter a Description detailing what the operation does or how to use it.
8. Click Create.
PostgreSQL Hook Operation Types

- **RunCommand Operation**
  - [Examples of RunCommand Operations](#)
- **RunExpect Operation**
  - [Example of a RunExpect Operation](#)
- **RunPowershell Operation**
  - [Example of a RunPowershell Operation](#)
- **PostgreSQL Environment Variables**
  - [dSource Environment Variables](#)
  - [VDB Environment Variables](#)

**RunCommand Operation**

The **RunCommand** operation runs a shell command on a Unix environment using whatever binary is available at `/bin/sh`. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

**Examples of RunCommand Operations**

You can input the full command contents into the RunCommand operation.

```bash
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

if test -d "$remove_dir"; then
  rm -rf "$remove_dir" || exit 1
fi

exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```bash
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

**RunExpect Operation**

The **RunExpect** operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as `ssh`. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Admin application and CLI to aid in debugging.
If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunExpect Operation

Start an `ssh` session while interactively providing the user's password.

```bash
spawn ssh user@delphix.com
expect {
  -re {Password: } { send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
  }
  timeout {
    puts "Timed out waiting for password prompt."
    exit 1
  }
}
exit 0
```

RunPowershell Operation

The `RunPowershell` operation executes a Powershell script on a Windows environment. The Delphix Engine captures and logs all output of the script. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunPowershell Operation

You can input the full command contents into the `RunPowershell` operation.

```powershell
$removedir = $Env:DIRECTORY_TO_REMOVE
if ((Test-Path $removedir) -And (Get-Item $removedir) -is [System.IO.DirectoryInfo]) {
  Remove-Item -Recurse -Force $removedir
} else {
  exit 1
}
exit 0
```

PostgreSQL Environment Variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set certain environment variables so that the user-provided script can use them to access the dSource or VDB.

dSource Environment Variables

No environment variables are set.

VDB Environment Variables

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGDATA</td>
<td>The path to the VDB data files mounted from the Delphix Engine</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>PGPORT</td>
<td>The port on which the VDB is listening</td>
</tr>
<tr>
<td>PGUSER</td>
<td>The database user used to connect to the VDB</td>
</tr>
<tr>
<td>PGDATABASE</td>
<td>The VDB name within the PostgreSQL database cluster</td>
</tr>
</tbody>
</table>
SAP ASE Environments and Data Sources
SAP ASE Support and Requirements

These topics describe specific requirements for SAP ASE environments, such as user privileges, as well as the supported operating systems and database versions.

- **Requirements for SAP ASE Source Hosts and Databases**
- **Requirements for SAP ASE Target Hosts and Databases**
- **Network and Connectivity Requirements for SAP ASE Environments**
- **Supported Operating Systems and Database Versions for SAP ASE** — The source and target must be running the same DBMS version. For example, if the source is running SAP ASE 15.7, then so must the target. Operating System platform must be the same between the source and target, although the operating system version may differ. For example, if the source is running Red Hat Enterprise Linux 6.2 x86_64 then the target could be running Red Hat Enterprise Linux 6.4 x86_64, but not Solaris 10 SPARC.
Requirements for SAP ASE Source Hosts and Databases

This topic describes the configuration and settings requirements for Sybase ASE source environments and databases.

Source environments are servers which contain the source databases from which virtual database copies are made.

Source Host Requirements

- There must be an operating system user, such as `delphix_os`, that meets the following requirements:
  - The $PATH environment variable includes the location for the isql binary
  - The $SYBASE environment variable is set (via the .bashrc configuration file as we're using a non-interactive shell)
  - Can log into the source host via SSH
  - Has read access for the Sybase ASE Backup Server log files
- There must be a directory on the source host where you can install the Delphix Engine toolkit, for example: /
  var/opt/delphix/Toolkit
  - The `delphix_os` user must own the directory
  - The directory must have permissions 0770, for example, `-rwxrwx---`. However, you can also use more permissive settings.
  - The directory should have 256MB of available storage.
- The Delphix Engine must be able to make an ssh connection (for example, TCP port 22) to the source host.

Source Database Requirements

- There must be a database user, such as `delphix_disc`, that has `SELECT` privileges on `sysdatabases`, `sysservers` and `syslisteners` tables
- The discovery database user must have these privileges for each instance on the source host
- There must be a database user, such as `delphix_link`, that has `SELECT` privileges on the above tables.
  - If you will select “New Full Backup” option when linking, this user must also have privileges to take a new full database dump of the source database. For more information about linking options, see Linking a SAP ASE Data Source.
  - The link database user can be different for each instance and database on the source host.

You can also use the `delphix_link` user for discovery, in which case only one user login is needed.

Delphix creates a minimum of 8 data devices and 8 log devices. As a result, a minimum of 16 devices per dSource is created and the same is true for each VDB when provisioned.

Related Links

- For more information about using the HostChecker bundle, see Using HostChecker to Validate SAP ASE Source and Target Environments
• Linking a SAP ASE Data Source
• Sudo File Configurations
Requirements for SAP ASE Target Hosts and Databases

This topic describes the configuration and settings requirements for SAP ASE target hosts and databases.

- **Target Host Requirements**
- **Related Links**

**Target Host Requirements**

- The operating system on the target environment must be the same as, or binary compatible with, the operating system on the source environment.
- The SAP ASE version on the target environment must be the same as the version on the source environment.
- There must be an operating system user, such as `delphix_os`, that meets the following requirements:
  - The `$PATH` environment variable includes the location for the `isql` binary.
  - The `$SYBASE` environment variable is set (via the `.bashrc` configuration file as we're using a non-interactive shell).
  - Can login to the source host via Secure Shell (SSH).
  - Has permission for the mount-point directory.
  - Has permission to run `mount/umount` and `mkdir/rmdir` as the super-user, usually granted through `sudo` permissions. See [Sudo Privilege Requirements](#) for further explanation of this requirement, and [Sudo File Configurations](#) for example file configurations.
  - Disable `tty` for the `delphix_os` user for `mount/umount`.
  - Is in the same group as the operating system user running the ASE process.
- There must be a database user, such as `delphix_db`, with `SA` role on each instance on the target environment.
- There must be a directory on the target environment where you can install the Delphix Engine toolkit, for example `/var/opt/delphix/Toolkit`. This will also be used as the base for mount points that are created when you provision a virtual database (VDB) to the target host.
  - The `delphix_os` user must own the directory.
  - The directory must have permissions 0770, for example, `-rwxrwx--`. However, you can also use more permissive settings.
  - The directory should have 1GB of available storage.
- If the target host will be used as a staging target environment (see [Managing SAP ASE Environments: An Overview](#)), at least one of the following two options must be configured:
  - You must configure the ASE Backup Servers with `sysservers` on the source database ASE instance, or
  - Full and transaction dump files from the source database must be available over NFS on the target host.
- Staging and target ASE instances should have disk mirroring disabled.

```
sp_configure "disable disk mirroring", 1
```

**Related Links**

- [Using HostChecker to Confirm Source and Target Environment Configuration](#)
- [sudoers Manual Page](#)
Network and Connectivity Requirements for SAP ASE Environments

**General Port Allocation**

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

### General Outbound from the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>25</td>
<td>Connection to a local SMTP server for sending email</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>53</td>
<td>Connections to local DNS servers</td>
</tr>
<tr>
<td>UDP</td>
<td>123</td>
<td>Connection to an NTP server</td>
</tr>
<tr>
<td>UDP</td>
<td>162</td>
<td>Sending SNMP TRAP messages to an SNMP Manager</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
<td>SSL connections from the Delphix Engine to the Delphix Support upload server</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>636</td>
<td>Secure connections to an LDAP server</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Connections to a Delphix replication target (see Configuring Replication)</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections to source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>

### General Inbound to the Delphix Engine Port Allocation

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>80</td>
<td>HTTP connections to the Delphix GUI</td>
</tr>
<tr>
<td>UDP</td>
<td>161</td>
<td>Messages from an SNMP Manager to the Delphix Engine</td>
</tr>
<tr>
<td>Protocol</td>
<td>Port(s)</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TCP</td>
<td>443</td>
<td>HTTPS connections to the Delphix GUI</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections from source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>
| TCP/UDP  | 32768 - 65535 | Required for NFS mountd and status services from target environment only if the firewall between Delphix and the target environment does not dynamically open ports. 
**Note:** If no firewall exists between Delphix and the target environment, or the target environment dynamically opens ports, this port range is not explicitly required. |

**Firewalls and Intrusion Detection Systems (IDS)**

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

**SSHD Configuration**

Both source and target Unix environments are required to have sshd running and configured such that the Delphix Engine can connect over ssh.

The Delphix Engine expects to maintain long-running, highly performant ssh connections with remote Unix environments. The following sshd configuration entries can interfere with these ssh connections and are thus disallowed:

**Disallowed sshd Configuration Entries**

- ClientAliveInterval
- ClientAliveCountMax
Connection Requirements for SAP ASE Environments

- The Delphix Engine uses an **SSH** connection to each source environment and **SAP ASE client** connections to the SAP ASE instances on the source environment.
- The Delphix Engine uses an **SSH** connection to each target environment, **NFS** connections from each target environment to the Delphix Engine, and **SAP ASE JDBC** connections to the virtual databases on the target environment.

Port Allocation for SAP ASE Environments

Refer to [Managing SAP ASE Environments](#) for information on SAP ASE environments. The Delphix Engine makes use of the following network ports for SAP ASE dSources and VDBs:

**Outbound from the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>Configuration dependent</td>
<td>JDBC Connections to the SAP ASE instances on the source environments</td>
</tr>
</tbody>
</table>

**Inbound to the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>33434-33464</td>
<td>Traceroute from source and target database servers to the Delphix Engine (optional)</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>111</td>
<td>Remote Procedure Call (RPC) port mapper used for NFS mounts</td>
</tr>
<tr>
<td>TCP</td>
<td>2049</td>
<td>NFS client from target hosts to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>1110</td>
<td>Network Status Monitor (NSM) client from target hosts to Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>4045</td>
<td>Network Lock Manager (NLM) client from target hosts to Delphix Engine</td>
</tr>
</tbody>
</table>

**Port Allocation Between Source and Staging Target Environments**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>Configuration dependent</td>
<td>SAP ASE Remote Backup Server protocol. Applies if linking using the <strong>New Full Backup</strong> option, or if linking with the <strong>Remote Backup Server</strong> option.</td>
</tr>
</tbody>
</table>

**Port Allocation Between Staging Target Environments and Shared Backup Fileserver**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
</table>
| TCP/UDP | NFS and related port numbers:  
- Portmap (111)  
- NFS (typically 2049)  
- Network Lock Manager (NLM)  
- Network Status Monitor (NSM) | NFS mount point exported by an NFS shared backup fileserver. Applies if linking using the **Local Backup Server** option. |
Supported Operating Systems and Database Versions for SAP ASE

Source and Target OS and DBMS Compatibility

The source and target must be running the same DBMS version. For example, if the source is running SAP ASE 15.7, then so must the target. Operating System platform must be the same between the source and target, although the operating system version may differ. For example, if the source is running Red Hat Enterprise Linux 6.2 x86_64 then the target could be running Red Hat Enterprise Linux 6.4 x86_64, but not Solaris 10 SPARC.

Supported DBMS and Operating System Versions

<table>
<thead>
<tr>
<th>DBMS Versions</th>
<th>Operating System Versions</th>
<th>Processor Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Adaptive Server Enterprise (ASE) 15.0.3, 15.5, 15.7</td>
<td>Red Hat Enterprise Linux 6.2, 6.3, 6.4</td>
<td>x86_64</td>
</tr>
<tr>
<td>SAP Adaptive Server Enterprise (ASE) 12.5</td>
<td>Red Hat Enterprise Linux 5.x</td>
<td>x86_64</td>
</tr>
<tr>
<td>SAP Adaptive Server Enterprise (ASE) 12.5, 15.03, 15.5, 15.7</td>
<td>Solaris 10</td>
<td>x86_64</td>
</tr>
<tr>
<td>SAP Adaptive Server Enterprise (ASE) 12.5, 15.03, 15.5, 15.7</td>
<td>Solaris 10</td>
<td>SPARC</td>
</tr>
</tbody>
</table>
Managing SAP ASE Environments

These topics describe special tasks and concepts for working with SAP ASE environments.

- Managing SAP ASE Environments: An Overview
- Using HostChecker to Validate SAP ASE Source and Target Environments
- Adding an SAP ASE Environment
- Editing SAP ASE Environment Attributes
- Changing the Host Name or IP Address of an SAP ASE Environment
- Deleting an SAP ASE Environment
- Managing SAP ASE Environment Users
- Refreshing an SAP ASE Environment
- Enabling Linking and Provisioning for SAP ASE Environments
Managing SAP ASE Environments: An Overview

This topic describes the high-level process for adding SAP ASE environments, linking SAP ASE databases to the Delphix Engine, and provisioning virtual databases.

Environment Setup

SAP ASE dSources are backed by a staging database that runs on a target host, as shown in the diagram. There is no requirement for additional local storage on this host, as the storage is mounted over NFS from the Delphix Engine. At Delphix, we refer to the creation and maintenance of this staging database on the staging host as "validated sync," because it prepares the dSource data on the Delphix Engine for provisioning VDBs later on. After the Delphix Engine creates the staging database, it continuously monitors the source database for new transaction log dumps. When it detects a new transaction log dump, it loads that dump to the staging database. The result is a TimeFlow with consistent points from which you can provision a virtual database (VDB), and a faster provisioning process, because there is no need for any database recovery during provisioning.

When you later provision a VDB, you can specify any environment as a target, including the environment that contains the staging database. However, for best performance, Delphix recommends that you choose a different target environment. The target must have an operating system that is compatible with the one running on the validated host, as described in Requirements for SAP ASE Target Hosts and Databases.

Workflow and Tasks for SAP ASE Environments

1. Add the desired source environments as described in Managing SAP ASE Environments.
2. Add the desired target environments as described in Managing SAP ASE Environments.
3. Link the source database as described in [Linking a SAP ASE Data Source](#).
4. Provision VDBs as described in [Provisioning an SAP ASE VDB](#).

**Related Links**
- [SAP ASE Support and Requirements](#)
Using HostChecker to Validate SAP ASE Source and Target Environments

The HostChecker is a standalone program which validates that host machines are configured correctly before the Delphix Engine uses them as data sources and provision targets.

Please note that HostChecker does not communicate changes made to hosts back to the Delphix Engine. If you reconfigure a host, you must refresh the host in the Delphix Engine in order for it to detect your changes.

You can run the tests contained in the HostChecker individually, or all at once. You must run these tests on both the source and target hosts to verify their configurations. As the tests run, you will either see validation messages that the test has completed successfully, or error messages directing you to make changes to the host configuration.
Adding an SAP ASE Environment

Prerequisites

See Requirements for SAP ASE Source Hosts and Databases.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click the Plus icon next to Environments.
4. In the Add Environment dialog, select Unix/Linux.
5. Select Standalone Host.
6. Enter the Host IP address.
7. Enter an optional Name for the environment.
8. Enter the SSH port.
   The default value is 22.
9. Enter a Username for the environment.
10. Select a Login Type.
11. For Password, enter the password associated with the user in Step 10.

   Using Public Key Authentication
   If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      ii. Run chmod 755 ~ to make your home directory writable only by your user.

   The public key needs to be added only once per user and per environment.

   You can also add public key authentication to an environment user’s profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

12. For Password Login, click Verify Credentials to test the username and password.
13. Enter a Toolkit Path.
   The toolkit directory stores scripts used for Delphix Engine operations. It must have a persistent working directory rather than a temporary one. The toolkit directory will have a separate sub-directory for each database instance. The toolkit path must have 0770 permissions.
14. Click the Discover SAP ASE checkbox.
15. Enter a Username for an instance on the environment.
16. Enter the Password associated with the user in Step 15.
17. Click OK.
Post-Requisites

After you create the environment, you can view information about it by selecting Manage > Environments and then selecting the environment name.
1. Log in to the Delphix Admin application with Delphix Admin credentials or as the owner of an environment.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of an environment to view its attributes.
4. Under Attributes, click the Pencil icon to edit an attribute.
5. Click the Check icon to save your edits.

### Common Editable Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment Users</td>
<td>The users for that environment. These are the users who have permission to ssh into an environment, or access the environment through the Delphix Connector. See the Requirements topics for specific data platforms for more information on the environment user requirements.</td>
</tr>
<tr>
<td>Host Address</td>
<td>The IP address for the environment host.</td>
</tr>
<tr>
<td>Notes</td>
<td>Any other information you want to add about the environment.</td>
</tr>
</tbody>
</table>

### SAP ASE Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB User</td>
<td>User for Delphix to use for ASE database operations</td>
</tr>
<tr>
<td>DB Password</td>
<td>Credentials to use for the DB User</td>
</tr>
</tbody>
</table>
Changing the Host Name or IP Address of an SAP ASE Environment

This topic describes how to change the host name or IP address for source and target environments, and for the Delphix Engine.

- **Procedure**
  - **For Source Environments**
  - **For VDB Target Environments**
  - **For the Delphix Engine**

**Procedure**

**For Source Environments**

1. Disable the dSource as described in [Enabling and Disabling dSources](#).
2. If the Host Address field contains an IP address, edit the IP address.
3. If the Host Address field contains a host name, update your Domain Name Server to associate the new IP address to the host name. The Delphix Engine will automatically detect the change within a few minutes.
4. In the Environments screen of the Delphix Engine, refresh the host.
5. Enable the dSource.

**For VDB Target Environments**

1. Disable the VDB as described in [Enabling and Disabling Virtual Databases](#).
2. If the Host Address field contains an IP address, edit the IP address.
3. If the Host Address field contains a host name, update your Domain Name Server to associate the new IP address to the host name. The Delphix Engine will automatically detect the change within a few minutes.
4. In the Environments screen of the Delphix Engine, refresh the host.
5. Enable the VDB.

**For the Delphix Engine**

1. Stop all running VDBs by clicking the red Stop button on the VDB card.
2. Disable all dSources as described in [Enabling and Disabling dSources](#).
3. You can use either the command line interface or the Server Setup application to change the IP address of the Delphix Engine.
   a. To use the command line interface, press F2 and follow the instructions described in [Setting Up Network Access to the Delphix Engine](#).
   b. To use the Server Setup application, go to System > Server Setup in the Delphix Admin interface, or click Server Setup in the Delphix Engine login screen.
      i. In the Network panel, click Modify.
      ii. Under DNS Services, enter the new IP address.
      iii. Click OK.
4. Refresh all Environments by clicking the Blue/Green Refresh Symbol on the Environments screen.
5. Enable all dSources as described in [Enabling and Disabling dSources](#).
6. Start all VDBs by clicking the Start button on the VDB card.
Deleting an SAP ASE Environment

This topic describes how to delete an environment.

Prerequisites

You cannot delete an environment that has any dependencies, such as dSources or VDBs. These must be deleted before you can delete the environment.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, select the environment you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.
Managing SAP ASE Environment Users

This topic describes how to manage the users associated with an environment. For information on providing Delphix users with privileges for groups and database objects, see the topics under Managing Delphix Users and Managing Policies.

- Prerequisites
- Procedure

Prerequisites

Users that you add to an environment must meet the requirements for that environment as described in the platform-specific Requirements topics.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Environments.
4. Click the name of an environment to open the environment information screen.
5. Under Basic Information, click the green Plus icon to add a user.
6. Enter the Username and Password for the OS user in that environment.

Using Public Key Encryption

If you want to use public key encryption for logging into your environment:

a. Select Public Key for the Login Type.
   
   b. Click View Public Key.
      
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      
      ii. Run chmod 755 ~ to make your home directory writable only by your user.

The public key needs to be added only once per user and per environment.

You can also add public key authentication to an environment user's profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

7. Click the Check icon to save the new user.
8. To change the primary user for this environment, click the Pencil icon next to Environment Users.
9. To delete a user, click the Trash icon next to their username.
Refreshing an SAP ASE Environment

This topic describes how to refresh an environment.

After you make changes to an environment that you have already set up in the Delphix Admin application, such as installing a new database home, creating a new database, or adding a new listener, you may need to refresh the environment to reflect these changes.

**Procedure**

1. Log in to the Delphix Admin application with Delphix Admin credentials.
2. Select Manage > Environments.
3. In the Environments panel, click on the name of the environment to you want to refresh.
4. Click the Refresh icon.

To refresh all environments, click the Refresh icon next to Environments.
Enabling Linking and Provisioning for SAP ASE Environments

This topic describes how to enable and disable provisioning and linking for SAP ASE databases.

Before a database can be used as a dSource, you must first make sure that you have enabled linking to it. Similarly, before you can provision a VDB to a target database, you must make sure that you have enabled provisioning to it.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Environments.
3. Click Databases.
4. Slide the button next to Allow Provisioning to On or Off to enable or disable provisioning for that instance.
5. Click show details for the database and slide the button next to Allow Linking to On or Off to enable or disable linking.
Managing SAP ASE Data Sources

These topics describe special tasks and concepts for working with SAP ASE data sources.

- Linking SAP ASE Data Sources: An Overview
- Linking a SAP ASE Data Source
- Advanced Data Management Settings for SAP ASE dSources
- Using Pre- and Post-Scripts with SAP ASE Data Sources
- Deleting an SAP ASE dSource
- Detaching and Re-Attaching SAP ASE dSources
- Enabling and Disabling SAP ASE dSources
- SAP ASE dSource Icon Reference
Linking SAP ASE Data Sources: An Overview

This topic describes basic concepts behind the creation of dSources from SAP ASE databases.

Initial Linking and Staging Databases

A dSource is the copy of a physical database that is created when the Delphix Engine links to and loads the database from a backup. The database backup can be a new full database backup that the Delphix Engine initiates, the most recent existing database backup, or an existing database backup specified by the user. When loading from an existing backup, the backup should be in a location that the source environment user can access.

After obtaining the initial snapshot and linking the dSource, the Delphix Engine keeps the dSource and the source database in sync by monitoring the source database for new database and transaction backups, and then applying those backups on a standby database. This database is called the staging database. A target environment that hosts one or more staging databases is referred to as a staging target.

After you have linked a database into the Delphix Engine, you can re-initialize it by performing a sync on the dSource.

Related Links

- ASE
- Link an SAP ASE Data Source
- Add an SAP ASE Environment
Linking a SAP ASE Data Source

This topic describes the process of linking to a source database and creating a dSource.

Prerequisites

- Make sure you have correctly set up the source and target environment, as described in Managing SAP ASE Environments.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > Add dSource. Alternatively, on the Environment Management screen, you can click Link next to a database name to start the dSource creation process.
3. In the Add dSource wizard, select the source database.

   - Changing the Environment User
   
   If you need to change or add an environment user for the source database, see Managing SAP ASE Environment Users.

4. Enter your login credentials for the source database.
5. Click Verify Credentials.
6. Click Next.
7. Select a Database Group for the dSource. Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. For more information, see the topics under Users, Permissions, and Policies.
8. Click Next.
9. Select an Initial Load option and enter any additional settings needed. There are three different options for the initial load of the dSource:
   - New Full Backup - Let Delphix create a new full backup file and load it. Note that when Delphix creates the backup, it is dumped to Delphix storage, not the Backup Location specified in the next step.
   - Most Recent Existing Full Backup - Find the most recent existing full backup file in the Backup Location and load it.
   - Specific Existing Full Backup - Specify which backup files in the Backup Location that you want to load.
10. Enter the Backup Location. This is the directory where the database backups are stored.
11. Select whether the data in the database is Masked. This setting is a flag to the Delphix Engine that the database data is in a masked state. Selecting this option will not mask the data.
12. Enable or disable LogSync.
13. Select Backup Location Type.
14. Click Advanced to edit Retention policies, Pre and Post Scripts and External Data Directory.
15. Click Next.
16. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs, DB_Link and DB_Sync, to create the dSource. You can monitor
these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System > Event Viewer**. When the jobs have successfully completed, the database icon will change to a dSource icon on the **Environments > Databases** screen, and the dSource will be added to the list of **My Databases** under its assigned group.

---

**The dSource Card**

After you have created a dSource, you can view information about it on the dSource card. You can also make modifications to its policies and permissions. To view the front of the dSource card, click the **Open** icon in the **Databases** panel. The card will then flip, showing you information such as the **Source Database** and **Data Management** configuration.

---

**Related Links**

- [Requirements for SAP ASE Source Environments](#)
- [Requirements for SAP ASE Target Environments](#)
- [Users, Permissions, and Policies](#)
Advanced Data Management Settings for SAP ASE dSources

- **Accessing Data Management Settings**
- **Retention Policies**
- **SAP ASE Settings**

This topic describes advanced data management settings for dSources

When linking a dSource, you can use default data management settings for Retention, sync validation, and LogSync policies. However, you can also use custom settings to improve overall performance and match the needs of your specific server and data environment.

**Accessing Data Management Settings**

There are three ways you can set or modify data management settings for dSources:

1. During the dSource linking process, click **Advanced** in the **Data Management** panel of the **Add dSource** wizard.
2. Click the **Edit** icon next to a policy or data management setting on the back of the dSource card. For **SnapSync** and **Retention** policies, click the policy name. This will open the **Policy Management** screen.
3. Select **Manage > Policies** in the top menu bar.
   - This will open the **Policy Management** screen. Select the policy for the dSource you want to modify, and click **Modify**.
   - See **Creating Custom Policies** and **Creating Policy Templates**, for more information.

**Retention Policies**

Retention policies determine the length of time that snapshots and log files are retained. The retention time for logs must be equal to or longer than the retention time for snapshots. This policy, in combination with the SnapSync policy, can have a significant impact on the performance of the Delphix Engine.

**SAP ASE Settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staging environment</td>
<td>SAP ASE instance to use for validated sync</td>
</tr>
<tr>
<td>Backup path</td>
<td>Path to the directory, relative to the staging environment, where backups can be found</td>
</tr>
</tbody>
</table>
Using Pre- and Post-Scripts with SAP ASE Data Sources

This topic describes the use of pre- and post-scripts with dSources.

For each script, you can add the script path and its arguments when linking a dSource in one of two ways. Either:

- During the Add dSource wizard process, in the Data Management screen, click Advanced.

or

- On the back of the dSource card, click the Pencil icon next to the Pre Script and Post Script fields.

To update pre- and post-script information:

1. Flip over the dSource card.
2. Click on the Pencil icon next to the Pre Script and Post Script fields.
3. When finished, click the check mark icon.

Note that pre-scripts are executed before the SnapSync policy, and if the script fails, SnapSync will fail as well. In the case of a post-script, the script will execute after SnapSync is complete. If a post-script fails, you will see an error message.

Using Scripts with SAP ASE dSources

- For SAP ASE dSources, linking pre- and post-scripts are run on the environment of the linked dSource.
- Staging pre- and post-scripts are run on the staging environment

Available Variables for SAP ASE dSource Scripts

These environment variables are set by Delphix Engine for scripts running on a SAP ASE dSource or VDB:

<table>
<thead>
<tr>
<th>Environment variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE_ENVUSER</td>
<td>Environment user executing the scripts</td>
</tr>
<tr>
<td>ASE_DBUSER</td>
<td>SAP ASE user name</td>
</tr>
<tr>
<td>ASE_DATABASE</td>
<td>SAP ASE database name</td>
</tr>
<tr>
<td>ASE_INSTANCE</td>
<td>SAP ASE instance name</td>
</tr>
<tr>
<td>ASE_PORT</td>
<td>SAP ASE instance port</td>
</tr>
</tbody>
</table>

Specifying Arguments for SAP ASE Scripts

You can specify multiple arguments for a script. In the Pre or Post Script field, enter the path to the script, and then list the arguments. If the argument contains spaces, enclose it in single or double quotes. You can escape single quotes within the argument with a backslash.

An Example with Three Arguments
An Example with an Apostrophe

```
/opt/app/product/10.2.0.5/db_1/dbs/myscript.sh o

I'd rather be in Hawaii.
```

```
/opt/app/product/10.2.0.5/db_1/dbs/myscript.sh '
```
Deleting an SAP ASE dSource

This topic describes how to delete a dSource.

Prerequisites

- You cannot delete a dSource that has dependent VDBs. Before deleting a dSource, make sure all dependent VDBs have been deleted as described in Deleting a VDB.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. In the Databases panel, select the dSource you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.

Deleting a dSource will also delete all snapshots, logs, and descendant VDB Refresh policies for that database, and the deletion cannot be undone.
Detaching and Re-Attaching SAP ASE dSources

This topic describes how to detach dSources and re-attach them to a different source database.

Each dSource contains an association with the source database, as well as the data it has pulled from the source database up to that point. It is possible to detach, or unlink, a dSource from its source database. This breaks the association with the source database without affecting the data within Delphix. Detached dSources and their source databases have these properties:

- Detached dSources can be used as the source of VDB provisioning operations
- The source database can be re-linked as a different dSource
- The dSource can be attached to a different source database and resume incremental SnapSync, provided the source database is an image of the same logical database (for example, a standby replica). Currently this is only supported through the CLI, and is most often used when performing an initial load from a standby replica and then later attaching the dSource to the production database for subsequent incremental syncs.

Detaching a dSource

1. Log into the Delphix Admin application as a user with OWNER privileges on the dSource, group, or domain.
2. Select Manage > Databases > My Databases.
3. Select the database you want to unlink or delete.
4. Click the Unlink icon.
   A warning message will appear. Click Yes

Attaching a dSource

Rebuilding Source Databases and Using VDBs

In situations where you want to rebuild a source database, you will need to detach the original dSource and create a new one from the rebuilt data source. However, you can still provision VDBs from the detached dSource.

1. Detach the dSource as described in this topic.
2. Re-name the detached dSource by clicking the Edit icon next to its name. This is necessary only if you intend to give the new dSource the same name as the original one. If you attempt to give the new dSource the same name as the detached one, you will see an error message.
3. Create the new dSource from the rebuilt database.

You will now be able to provision VDBs from both the detached dSource and the newly created one, but the detached dSource will only represent the state of the source database prior to being detached.

The attach operation is currently only supported in the CLI. Full GUI support will be added in a future release. Only databases that represent the same physical database can be re-attached.
1. Log into the **Delphix CLI** as a user with **OWNER** privileges on the dSource, group, or domain.
2. Select the dSource by name using `database select <dSource>`
3. Run the `attachSource` command.
4. Set the source config you want to attach to using `set source.config=<newSource>`. **Source configs** are named by their database unique name.
5. Set any other source configuration operations as you would for a normal link operation.
6. Run the `commit` command.
Enabling and Disabling SAP ASE dSources

This topic describes how to enable and disable dSources for operations such as backup and restore.

For certain processes, such as backing up and restoring the source database, you may want to temporarily disable your dSource. Disabling a dSource turns off communication between the dSource and the source database, but does not tear down the configuration that enables communication and data updating to take place. When a disabled dSource is later enabled, it will resume communication and incremental data updates from the source database according to the original policies and data management configurations that you set.

Disabling a dSource is also a pre-requisite for several other operations, like database migration and upgrading the dSource after upgrade of the associated data source. See the topics in the Related Links section for more information.

Procedure

1. Go to Manage > Databases > My Databases.
2. Select the dSource to disable.
3. On the back of the dSource card, move the slider control from Enabled to Disabled.
4. Click Yes to acknowledge the warning.

When you are ready to enable the dSource again, move the slider control from Disabled to Enabled, and the dSource will continue to function as it did previously.
## SAP ASE dSource Icon Reference

This topic illustrates the icons that appear on dSources and Virtual Databases (VDBs) in the Delphix Engine Graphic User Interface, and describes the meaning of each, along with tips for clearing those that represent errors.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Icon" /></td>
<td>There is a critical fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image2.png" alt="Icon" /></td>
<td>There is a warning fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Icon" /></td>
<td>The Delphix Engine is checking the VDB status.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Icon" /></td>
<td>The dSource has been deleted or the Source status is UNKNOWN.</td>
</tr>
<tr>
<td><img src="image5.png" alt="Icon" /></td>
<td>The state of the VDB is unknown. This is often associated with a connection error.</td>
</tr>
<tr>
<td><img src="image6.png" alt="Icon" /></td>
<td>The VDB is inactive.</td>
</tr>
<tr>
<td><img src="image7.png" alt="Icon" /></td>
<td>The dSource has been unlinked from the source database.</td>
</tr>
<tr>
<td><img src="image8.png" alt="Icon" /></td>
<td>The VDB is disabled, is in the process of being created, or the creation process has been canceled or failed. See <a href="#">Enabling and Disabling VDBs</a> for more information.</td>
</tr>
<tr>
<td><img src="image9.png" alt="Icon" /></td>
<td>The VDB is running normally</td>
</tr>
<tr>
<td><img src="image10.png" alt="Icon" /></td>
<td>The dSource is disabled. See <a href="#">Enabling and Disabling dSources</a> for more information.</td>
</tr>
<tr>
<td><img src="image11.png" alt="Icon" /></td>
<td>The dSource or VDB is ready for Linux Transformation.</td>
</tr>
</tbody>
</table>
Provisioning VDBs from SAP ASE dSources

These topics describe special tasks and concepts for provisioning VDBs from SAP ASE dSources.

- Provisioning SAP ASE VDBs: An Overview
- Provisioning an SAP ASE VDB
- Provisioning an SAP ASE VDB from a Replicated VDB or dSource
- Deleting a SAP ASE VDB
- Enabling and Disabling SAP ASE VDBs
- Migrating an SAP ASE VDB
- Refreshing an SAP ASE VDB
- Rewinding an SAP ASE VDB
- SAP ASE VDB Icon Reference
Provisioning SAP ASE VDBs: An Overview

Before you provision SAP ASE virtual databases (VDBs), you must:

- Already have linked a dSource from a source database, as described in Linking an SAP ASE dSource, or have already created a VDB from which you want to provision another VDB
- Make sure you have the required privileges on the target environment and you are provisioning to a different target environment from the one where you set up the staging database, you must make sure that the two environments have compatible operating systems, all described in Requirements for SAP ASE Target Hosts and Databases.

Procedure

1. Log into the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a means of provisioning. See Provisioning by Snapshot and LogSync in this topic for more information.
5. Click Provision. The Provision VDB panel will open, and the Instance and Database Name fields will auto-populate with information from the dSource.
6. Specify any Pre or Post Scripts that should be used during the provisioning process.
7. Click Next.
8. Select a Target Group for the VDB. Click the green Plus icon to add a new group, if necessary.
9. Select a Snapshot Policy for the VDB. Click the green Plus icon to create a new policy, if necessary.
10. Click Next.
11. If your Delphix Engine system administrator has configured the Delphix Engine to communicate with an SMTP server, you will be able to specify one or more people to notify when the provisioning is done. You can choose other Delphix Engine users, or enter email addresses.
12. Click Finish. When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the VDB will be included in the group you designated, and it will be listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.

Provisioning by Snapshot or LogSync

When provisioning by snapshot, you can provision to the start of any particular snapshot.

If LogSync is enabled on the dSource, you can provision by LogSync information. When provisioning by LogSync information, you can provision to any point in time, within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the Slide to Open LogSync control at the top of an individual snapshot card.
Provisioning an SAP ASE VDB

This topic describes how to provision a virtual database (VDB) from a SQL Server dSource.

Prerequisites
- You must have linked a dSource from a source database, as described in Linking a SAP ASE Data Source, or have already created a VDB from which you want to provision another VDB.
- You must have set up target environments as described in Adding an SAP ASE Environment.
- Make sure you have the required privileges on the target environment as described in Requirements for SAP ASE Target Hosts and Databases.
- If you are provisioning to a different target environment from the one where you set up the staging database, you must make sure that the two environments have compatible operating systems, as described in Requirements for SAP ASE Target Hosts and Databases. For more information on the staging database and the validated sync process, see Managing SAP ASE Environments: An Overview.

Procedure
1. Login to the Delphix Admin application.
2. Select Manage > Databases > My Databases.
3. Select a dSource.
4. Select a means of provisioning. See Provisioning by Snapshot and LogSync in this topic for more information.
5. Click Provision. The Provision VDB panel will open, and the Instance and Database Name fields will auto-populate with information from the dSource.
6. Select whether to enable Truncate Log on Checkpoint database option for the VDB.
7. Click Next.
8. Select a Target Group for the VDB. Click the green Plus icon to add a new group, if necessary.
9. Select a Snapshot Policy for the VDB. Click the green Plus icon to create a new policy, if necessary.
10. Click Next.
11. Specify any Hooks to be used during the provisioning process. See Customizing SAP ASE Management with Hook Operations for more information.
12. If your Delphix Engine system administrator has configured the Delphix Engine to communicate with an SMTP server, you will be able to specify one or more people to notify when the provisioning is done. You can choose other Delphix Engine users or enter email addresses.
13. Click Finish. When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the VDB will be included in the group you designated, and it will be listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.

Provisioning by Snapshot

You can provision to the start of any snapshot by selecting that snapshot card from the TimeFlow view, or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.
Provisioning by LogSync

If LogSync is enabled on the dSource, you can provision by LogSync information. When provisioning by LogSync information, you can provision to any point in time within a particular snapshot. The TimeFlow view for a dSource shows multiple snapshots by default. To view the LogSync data for an individual snapshot, use the Slide to Open LogSync control at the top of an individual snapshot card. Drag the red triangle to the point in time from which you want to provision. You can also enter a date and time directly.

Related Links

- Linking a SAP ASE Data Source
- Adding an SAP ASE Environment
- Requirements for SAP ASE Target Hosts and Databases
- Managing SAP ASE Environments: An Overview
- Customizing SAP ASE Management with Hook Operations
Provisioning an SAP ASE VDB from a Replicated VDB or dSource

This topic describes how to provision from a replicated dSource or virtual database (VDB). The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the replica containing the replicated object.

- **Prerequisites**
- **Procedure**
- **Post-Requisites**

**Prerequisites**

- You must have replicated a dSource or a VDB to the target host, as described in [Replication Overview](#).
- You must have added a compatible target environment on the target host

**Procedure**

1. Log into the Delphix Admin application for the target host.
2. Select Manage > Databases > My Databases.
3. In the list of replicas, select the replica that contains the dSource or VDB you want to provision.
4. The provisioning process is now identical to the process for provisioning standard objects.

**Post-Requisites**

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.
Deleting a SAP ASE VDB

This topic describes how to delete a VDB.

Procedure

1. Log into the Delphix Admin application using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. Select the VDB you want to delete.
4. Click the Trash Can icon.
5. Click Yes to confirm.
Enabling and Disabling SAP ASE VDBs

This topic describes how to enable and disable a virtual database (VDB).

Disabling a VDB is a pre-requisite for procedures such as VDB migration or upgrade. Disabling a VDB removes all traces of it, including any configuration files, from the target environment to which it was provisioned. When the VDB is later enabled again, these configuration files are restored on the target environment.

Procedure

1. Go to Manage > Databases > My Databases.
2. Select the VDB to disable.
3. On the back of the dSource card, move the slider control from Enabled to Disabled.
4. Click Yes to acknowledge the warning.

When you are ready to enable the VDB again, move the slider control form Disabled to Enabled, and the VDB will continue to function as it did previously.
Migrating an SAP ASE VDB

This topic describes how to migrate a Virtual Database (VDB) from one target environment to another.

There may be situations in which you want to migrate a virtual database to a new target environment, for example when upgrading the host on which the VDB resides, or as part of a general data center migration. This is easily accomplished by first disabling the database, then using the Migrate VDB feature to select a new target environment.

- **Prerequisites**
- **Procedure**
- **Video**

**Prerequisites**
- You should have already set up a new target environment that is compatible with the VDB that you want to migrate.

**Procedure**
1. Log into your Delphix Engine using Delphix Admin credentials.
2. Select Manage > Databases > My Databases.
3. Select the VDB you want to migrate, and then click the Open icon.
4. Slide the Enable/Disable control to Disabled, and click Yes to confirm. When the VDB is disabled, its icon will turn grey.
5. On the bottom-right corner of the VDB card, click the VDB Migrate icon.
6. Select the new target environment for the VDB, the user for that environment, and the database installation where the VDB will reside.
7. Click the Check icon to confirm your selections.
8. Slide the Enable/Disable control to Enabled, and click Yes to confirm. Within a few minutes your VDB will re-start in the new environment, and you can continue to work with it as you would any other VDB.

**Video**

![Get ADobeFlash Player](image)
Refreshing an SAP ASE VDB

- **Prerequisites**
- **Procedure**

This topic describes how to manually refresh a VDB.

Refreshing a VDB will re-provision the VDB from the dSource. As with the normal provisioning process, you can choose to refresh the VDB from a snapshot or a specific point in time. However, you should be aware that refreshing a VDB will delete any changes that have been made to it over time. When you refresh a VDB, you are essentially re-setting it to the state you select during the refresh process. You can refresh a VDB manually, as described in this topic, or you can set a VDB refresh policy, as described in the topics **Managing Policies: An Overview**, **Creating Custom Policies**, and **Creating Policy Templates**.

Although the VDB no longer contains the previous contents, the previous Snapshots and TimeFlow still remain in Delphix and are accessible through the Command Line Interface (CLI).

**Prerequisites**

To refresh a VDB, you must have the following permissions:

1. **Auditor** permissions on the dSource associated with the VDB.
2. **Auditor** permissions on the group that contains the VDB.
3. **Owner** permissions on the VDB itself.

A user with Delphix Admin credentials can perform a VDB Refresh on any VDB in the system.

**Procedure**

1. Log into the **Delphix Admin** application.
2. Under **Databases**, select the VDB you want to refresh, and then click the **Open** icon to open its card.
3. On the back of the VDB card, click the **Refresh VDB** icon in the lower right corner. This will open the screen to re-provision the VDB.
4. Select the refresh point as a snapshot or a point in time.
5. Click **Refresh VDB**.
6. Click **Provide Privileged Credentials** if you want to use login credentials on the target environment other than those associated with the **Environment User**.
7. Click **Yes** to confirm.
Rewinding an SAP ASE VDB

This topic describes the procedure for rewinding a VDB.

Rewinding a VDB rolls it back to a previous point in its TimeFlow and re-provisions the VDB. The VDB will no longer contain changes after the rewind point.

Although the VDB no longer contains changes after the rewind point, the rolled over Snapshots and TimeFlow still remain in Delphix and are accessible through the Command Line Interface (CLI). See the topic CLI Cookbook: Rolling Forward a VDB for instructions on how to use these snapshots to refresh a VDB to one of its later states after it has been rewound.

Prerequisites

To rewind a VDB, you must have the following permissions:

1. Auditor permissions on the dSource associated with the VDB.
2. Owner permissions on the VDB itself.

You do not need owner permissions for the group that contains the VDB. A user with Delphix Admin credentials can perform a VDB Rewind on any VDB in the system.

Procedure

1. Log into the Delphix Admin application.
2. Under Databases, select the VDB you want to rewind.
3. Select the rewind point as a snapshot or a point in time.
4. Click Rewind.
5. Click Provide Privileged Credentials if you want to use login credentials on the target environment other than those associated with the Environment User.
6. Click Yes to confirm.

TimeFlow bookmarks can be used as the rewind point when using the CLI. Bookmarks can be useful to:

- Mark where to rewind to - before starting a batch job on a VDB for example.
- Provide a semantic point to revert back to in case the chosen rewind point turns out to be incorrect.

See CLI Cookbook: Provisioning a VDB from a TimeFlow Bookmark for a CLI example using a TimeFlow bookmark.

Video

[Get ADOBE FLASH PLAYER]
### SAP ASE VDB Icon Reference

This topic illustrates the icons that appear on dSources and Virtual Databases (VDBs) in the Delphix Engine Graphic User Interface, and describes the meaning of each, along with tips for clearing those that represent errors.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>There is a critical fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>There is a warning fault associated with the dSource or VDB. See the error logs for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The Delphix Engine is checking the VDB status.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource has been deleted or the Source status is UNKNOWN.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The state of the VDB is unknown. This is often associated with a connection error.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The VDB is inactive.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource has been unlinked from the source database.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The VDB is disabled, is in the process of being created, or the creation process has been canceled or failed. See <a href="#">Enabling and Disabling VDBs</a> for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The VDB is running normally</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource is disabled. See <a href="#">Enabling and Disabling dSources</a> for more information.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The dSource or VDB is ready for Linux Transformation.</td>
</tr>
</tbody>
</table>
Customizing SAP ASE Management with Hook Operations

Hook operations allow you to execute an ordered list of custom operations as part of the dSource syncing, provisioning, and refresh process. For details on the types of operations that are available, see children of this page.

**dSource Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Sync</td>
<td>Operations performed before a sync. These operations can quiesce data to capture more information into the sync or stop applications that may interfere with the sync.</td>
</tr>
<tr>
<td>Post-Sync</td>
<td>Operations performed after a sync. These operations can undo any quiescing of data or undo any changes made by the Pre-Sync hook.</td>
</tr>
</tbody>
</table>

**Virtual Dataset Hooks**

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Clone</td>
<td>Operations performed after initial provision or refresh of a virtual dataset. This hook runs when the virtual dataset has been started. During a refresh, the Delphix Engine reaches this hook before the Post-Refresh hook.</td>
</tr>
<tr>
<td>Pre-Refresh</td>
<td>Operations performed before a refresh. These operations can back up any data or configurations from the running source before doing the refresh.</td>
</tr>
<tr>
<td>Post-Refresh</td>
<td>Operations performed after a refresh. You can use these operations to restore any data or configurations backed up by the Pre-Refresh hook. During a refresh, the Delphix Engine runs this after the Configure Clone hook.</td>
</tr>
</tbody>
</table>

**Operation Failure**

If a hook operation fails, it will fail the entire sync, provision, or refresh job. The job will halt immediately and no further hook operations will be run.

**Hook Operations on Cluster Environments**

When linking from, or provisioning to, cluster environments such as Oracle RAC environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine picks a node in the cluster at random and guarantees all operations within any single hook will execute serially on this node. Note that the Delphix Engine does not guarantee the same node is chosen for the execution of every hook.

**Setting Hook Operations**

You can construct hook operation lists through the Delphix Admin application or the command line interface (CLI).
You can either define the operation lists as part of the linking or provisioning process or edit them on dSources or virtual datasets that already exist.

**Setting Hook Operations through the Delphix Admin Application**

To specify hook operations during linking or provisioning, navigate to the **Hooks** tab of the **Linking Wizard** or **Provision Wizard**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Select the **type of operation** or click **Import** to load a hook operation template.
4. Click the **text area** and edit the contents of the operation.
5. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
6. To remove an operation from the list, click the **Trash** icon on the operation.
7. When you have set all hook operations, click **Next** to continue with the provisioning process.

To edit hook operations on an already-existing dSource or virtual dataset, navigate to the **Hooks** tab on the back of the **dSource card** or virtual dataset card.

1. Select the **hook** to edit.
2. The current operations at this hook will be displayed. To edit this list of operations, click the **Pencil** icon in the top right-hand corner of the card.
3. Click the **Plus** icon to add a new operation.
4. Select the **type of operation** or click **Import** to load a hook operation template.
5. Click the **text area** and edit the contents of the operation.
6. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
7. To remove an operation from the list, click the **Trash** icon on the operation.
8. When you have set all hook operations, click **Check** to save the changes.

**Setting Hook Operations through the CLI**

To specify hook operations during linking, edit the relevant hook's array of operations defined on the **LinkingParameters > Source > Operations** object.

To specify hook operations during provisioning, edit the relevant hook's array of operations defined on the **ProvisionParameters > Source > Operations** object.

To edit hook operations on an already-created dSource, edit the relevant hook's array of operations defined on the **Source > Operations** object.

To edit hook operations on an already-created virtual dataset, edit the relevant hook's array of operations defined on the **Source > Operations** object.

For more information about these CLI objects, see the **LinkedSourceOperations**, **VirtualSourceOperations**, **RunCommandOnSourceOperation**, and **RunExpectOnSourceOperation** API documentation in the **Help** menu of the **Delphix Admin** application.

**Example of Editing Hook Operations through the CLI**

1. Navigate to relevant source's **VirtualSourceOperations** object.
2. Select a hook to edit.

```
delphix> source
delphix source> select "pomme"
delphix source "pomme"> update
delphix source "pomme" update *> edit operations
delphix source "pomme" update operations *> edit postRefresh
```

3. Add an operation at index 0.

```
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 0 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 0 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 0 *> ls
Properties
  type: RunCommandOperation (*)
  command: echo Refresh completed. (*)
delphix source "pomme" update operations postRefresh 0 *> commit
```

4. Add another operation at index 1 and then delete it.

```
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 1 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 1 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 1 *> back
delphix source "pomme" update operations postRefresh *> unset 1
delphix source "pomme" update operations postRefresh *> commit
```

Hook Operation Templates

You can use templates to store commonly-used operations, which allows you to avoid repeated work when an operation is applicable to more than a single dSource or virtual dataset. You manage templates through the Delphix Admin application.

**Creating a Hook Operation Template**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Operation Templates.
4. Click the Plus icon to add a new operation template.
5. Enter a Name for the template.
6. Select an operation Type.
7. Enter a Description detailing what the operation does or how to use it.
8. Enter operation Contents to implement the operation partially or fully.
9. Click Create.

**Importing a Hook Operation Template**
To import a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Click Import.
4. Select the template to import.
5. Click Import.
6. When you have set all hook operations, click Check to save the changes.

**Exporting a Hook Operation Template**

To export a hook operation template, navigate to the Hooks tab on the back of the dSource or virtual dataset card.

1. Select the hook to edit.
2. Click the Plus icon to add a new operation.
3. Select the type of operation.
4. Click the text area and edit the contents of the operation.
5. Click Export.
6. Enter a Name for the template.
7. Enter a Description detailing what the operation does or how to use it.
8. Click Create.
SAP ASE Hook Operation Types

RunCommand Operation

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at /bin/sh. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Examples of RunCommand Operations

You can input the full command contents into the RunCommand operation.

```bash
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
fi
exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```bash
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

RunExpect Operation

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as ssh. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Admin application and CLI to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunExpect Operation

Start an ssh session while interactively providing the user's password.
spawn ssh user@delphix.com
expect {
    -re {Password: } {
        send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {
        puts "Timed out waiting for password prompt."
        exit 1
    }
}
exit 0

RunPowershell Operation

The RunPowershell operation executes a Powershell script on a Windows environment. The Delphix Engine captures and logs all output of the script. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunPowershell Operation

You can input the full command contents into the RunPowershell operation.

```powershell
$removedir = $Env:DIRECTORY_TO_REMOVE
if ((Test-Path $removedir) -And (Get-Item $removedir) -is [System.IO.DirectoryInfo]) {
    Remove-Item -Recurse -Force $removedir
} else {
    exit 1
}
exit 0
```

SAP ASE Environment variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set certain environment variables so that the user-provided script can use them to access the dSource or VDB.

dSource Environment Variables

<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE_ENVUSER</td>
<td>Environment username for the dSource</td>
</tr>
<tr>
<td>ASE_DBUSER</td>
<td>Database username for the dSource</td>
</tr>
<tr>
<td>ASE_DATABASE</td>
<td>Database name for the dSource</td>
</tr>
<tr>
<td>ASE_INSTANCE</td>
<td>SAP ASE Instance name for the dSource</td>
</tr>
<tr>
<td>ASE_PORT</td>
<td>SAP ASE Instance port for the dSource</td>
</tr>
</tbody>
</table>

VDB Environment Variables
### Environment Variables

<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE_ENVUSER</td>
<td>Environment username for the VDB</td>
</tr>
<tr>
<td>ASE_DBUSER</td>
<td>Database username for the VDB</td>
</tr>
<tr>
<td>ASE_DATABASE</td>
<td>Database name for the VDB</td>
</tr>
<tr>
<td>ASE_INSTANCE</td>
<td>SAP ASE Instance name for the VDB</td>
</tr>
<tr>
<td>ASE_PORT</td>
<td>SAP ASE Instance port for the VDB</td>
</tr>
</tbody>
</table>
Data Backup and Recovery

These topics describe how to use the Delphix Engine for data backup and recovery, including use of the Replication and Virtual to Physical (V2P) features.

- Backup and Recovery Strategies for the Delphix Engine
  - Backup and Recovery Requirements
  - Deployment Architecture
  - Mapping Requirements to Solutions
  - Backup Solution Implementation

- Replication
  - Replication Overview
  - Replication Use Cases
  - Replication User Interface
  - Configuring Replication
  - Replicas and Failover
  - Failing Over a Replica
  - Updating Replication User Credentials from Previous Versions
  - Provisioning from Replicated Data Sources or VDBs

- V2P: Virtual to Physical
  - Virtual to Physical: An Overview
  - V2P with an Oracle VDB
    - Move an Oracle VDB to a Physical ASM or Exadata Database
  - V2P with a SQL Server VDB
  - V2P with a PostgreSQL dSource or VDB
  - V2P with an SAP ASE dSource or VDB — This topic describes the procedure for exporting a virtual database (VDB) to a physical one, also known as V2P.
  - Manually Recovering a Database after V2P
  - Customizing Target Directory Structure for Database Export
  - V2P with Unstructured Files
Backup and Recovery Strategies for the Delphix Engine

These topics describe backup and recovery options for Delphix Engine.

As a software virtual appliance, Delphix leverages features of the storage, hypervisor, and appliance infrastructure to provide for recovery in the event of failure. These topics walk through the process of evaluating requirements and defining a solution. This process depends on requirements and features of the environment in which Delphix Engine is deployed.

- Backup and Recovery Requirements
- Deployment Architecture
- Mapping Requirements to Solutions
- Backup Solution Implementation
Backup and Recovery Requirements

This topic describes determining requirements around infrastructure failure modes and recovery.

Before devising a strategy, you must first have a set of requirements to evaluate possible solutions. What failures are you trying to protect against, and what are your recovery goals in the event of failure?

Failure Points

Before devising a strategy, you must first have a set of requirements by which the resulting solution can be evaluated. What failures are you trying to protect against, and what are your recovery goals in the event of failure?

Physical Server Failure

The Delphix Engine runs within the VMware ESX hypervisor, which itself is running on a physical machine. Failure of that physical machine will affect the Delphix Engine, as well as any other virtual machines running on that server. The failure is isolated to that particular server, and is not the result of a larger, site-wide failure.

- **Recommendation**: ESX Clustering

Storage Failure

The Delphix Engine uses LUNs from a storage array provided through the VMware hypervisor. The storage array may have redundant disks and/or controllers to protect against single points of failure within the array. However, the Delphix Engine can still be affected by a failure of the entire array, the SAN path between the Delphix Engine and the array, or by a failure of the LUNs in the array that are assigned to the Delphix Engine.

- **Recommendation**: Replication

Site Failure

When an entire site or datacenter goes down, all servers, storage, and infrastructure are lost. This will affect not only the Delphix Engine, but any production databases and target servers in the datacenter.

- **Recommendation**: Replication

Administrative Error

If an administrator mistakenly deletes a VDB or takes some other irreversible action, there is no method of recovery built into the Delphix Engine.

- **Recommendation**: Snapshots

Recovery Objectives

Once infrastructure fails, some amount of work is required to restore the Delphix Engine to an operational state. Clients won't have access to the Delphix Engine during this time, and the point to which the system is recovered is dependent on the mechanism being used. These qualitative aspects of recovery can be captured by the following metrics. As these metrics are often directly associated with cost, it is important to think not just about the desired metrics, but also the minimum viable goals.

Recovery Point Objective (RPO)

The RPO is the acceptable amount of data that can be lost in the event of a failure. For example, if backups are taken once a day, then at most 24 hours of data will be lost if the system fails immediately before a regularly scheduled backup.
Recovery Time Objective (RTO)

The RTO is the time required to restore the system to an operational state after a failure. For example, a recovery may require restoring data from a backup, followed by some number of manual steps to recreate the configuration in the new system. RTO is equivalent to the downtime experienced by clients.

Recovery Time Granularity (RTG)

The granularity of the recovery time is the specificity by which you can select a particular point in time from the past to restore the system. For example, VM snapshots taken every hour provide no way to restore to a point in time between those snapshots.
Deployment Architecture

This topic describes components of the Delphix deployment architecture.

Delphix operates in a virtual environment with several core systems working in concert, each with its own set of capabilities. Understanding this architecture is critical in evaluating how solutions can be applied across the components, and the tradeoffs involved.

Architectural Components

This diagram illustrates Delphix’s recommended best practices for deploying the Delphix Engine in a VMware environment:

![Diagram of Delphix deployment architecture]

This architecture is designed to isolate I/O traffic to individual LUNs while using the most commonly deployed VMware components. In this example each VMDK file is placed in a separate VMFS volume. Each volume is exported to every node in the ESX cluster, allowing the Delphix Engine to run on any physical host in the cluster.

Fault Recovery Features

Across the recommended deployment architecture there are three key components in play: Delphix Engine, VMware, and storage. Each of these provides different failure handling capabilities, which can be roughly grouped into the following areas.

Server Clustering

Clustering provides a standby server that can take over in the event of failure. A given clustering solution may or may not provide high availability guarantees, though all provide failover capabilities, provided that an identical
passive system is available.

**Snapshots**

Snapshots preserve a point-in-time copy of data that can be used later for rollback or to create writable copies. Creating a snapshot is typically low cost in terms of space and time. Because they use the storage allocated to the array, snapshots restore quickly, but they do not protect against failures of the array.

**Replication**

Data replication works by sending a series of updates from one system to another in order to recreate the same data remotely. This stream can be synchronous, but due to performance considerations is typically asynchronous, where some data loss is acceptable. Replication has many of the same benefits of backup, in that the data is transferred to a different fault domain, but has superior recovery time given that the data is maintained within an online system. The main drawback of replication is that the data is always current - any logical data error in the primary system is also propagated to the remote target. The impact of such a failure is less when replication is combined with snapshots, as is often the case with continuous data protection (CDP) solutions.

**Backup**

Like snapshots, backup technologies preserve a point-in-time copy of a storage dataset, but then move that copy to offline storage. Depending on the system, both full and incremental backups may be supported, and the backup images may or may not be consistent. Backup has the advantage that the data itself is stored outside the original fault domain, but comes at high cost in terms of complexity, additional infrastructure, and recovery time.
Mapping Requirements to Solutions

This topic describes how to map from backup and recovery requirements to solutions.

With requirements and detailed knowledge of the deployment architecture, we can map to solutions tailored for the features provided by the underlying infrastructure.

Feature Capabilities

Based on these failure points and recovery features, you can use the following table to map requirements to architectural components: VMware (V), Delphix (D), or storage (S). This can drive implementation based on infrastructure capabilities and recovery objectives.

<table>
<thead>
<tr>
<th>Failure Point</th>
<th>Clustering</th>
<th>Snapshots</th>
<th>Replication</th>
<th>Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Failure</td>
<td>V</td>
<td>-</td>
<td>V S D</td>
<td>-</td>
</tr>
<tr>
<td>Storage Failure</td>
<td>-</td>
<td>-</td>
<td>V S D</td>
<td>V S</td>
</tr>
<tr>
<td>Site Failure</td>
<td>-</td>
<td>-</td>
<td>V S D</td>
<td>V S</td>
</tr>
<tr>
<td>Administrative Error</td>
<td>-</td>
<td>V S</td>
<td>-</td>
<td>V S</td>
</tr>
</tbody>
</table>

Recovery Point Objective (RPO)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>Zero</td>
<td>All changes committed to disk are automatically propagated to the new server. Any pending changes in memory are lost.</td>
</tr>
<tr>
<td>Replication</td>
<td>Near zero</td>
<td>Most solutions offer scheduled replication, but many can offer continuous replication with near-zero data loss.</td>
</tr>
<tr>
<td>Snapshots</td>
<td>Snapshot period (for example, one hour)</td>
<td>Given their relatively low cost, snapshots tend to be taken at a higher frequency than a traditional backup schedule.</td>
</tr>
<tr>
<td>Backup</td>
<td>Backup period (for example, one day)</td>
<td>Backup policies can be configured in a variety of ways, but even with incremental backups, most deployments operate no more frequently than once a day because of the cost of full backups, and the impact of incremental backups on recovery time.</td>
</tr>
</tbody>
</table>

Recovery Time Objective (RTO)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>Near zero</td>
<td>VM clustering with the Delphix Engine provides near zero downtime in the event of failure, but clients may be briefly paused or interrupted.</td>
</tr>
<tr>
<td>Replication</td>
<td>15 minutes</td>
<td>The target side environment is kept in hot standby mode, so it is relatively quick to switch over to the target environment. Depending on the scope of the failure, however, some configuration information may need to be changed on the target side prior to enabling operation.</td>
</tr>
<tr>
<td>Snapshots</td>
<td>15 minutes</td>
<td>The Delphix Engine can be rolled back to a previous state. Changes made to systems external to the Delphix Engine (for example, deleting a VDB) can cause inconsistencies after rollback.</td>
</tr>
<tr>
<td>Backup</td>
<td>Hours or days</td>
<td>Restoring a full backup can be very time consuming. In addition to having to read, transfer, and write all of the data, the same process will need to be run for each incremental backup to reach the objective point.</td>
</tr>
</tbody>
</table>

**Recovery Time Granularity**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Granularity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>None</td>
<td>Only the current system state can be recovered.</td>
</tr>
<tr>
<td>Replication</td>
<td>None</td>
<td>Only the nearest replicated state can be recovered, unless combined with snapshots.</td>
</tr>
<tr>
<td>Snapshots</td>
<td>Snapshot period (for example, one hour)</td>
<td>Determined by the snapshot schedule.</td>
</tr>
<tr>
<td>Backup</td>
<td>Backup period (for example, one day)</td>
<td>Determined by the backup schedule.</td>
</tr>
</tbody>
</table>
Backup Solution Implementation

This topic describes tradeoffs involved with backup and recovery solutions.

With the exception of clustering, solutions can be implemented using features at both the storage and hypervisor layer. Choosing the right technology requires understanding both your requirements and what infrastructure is in use in your environment. The following sections outline some basic choices and the tradeoffs involved.

Clustering

VMware vSphere high availability provides the ability to have a VM configuration shared between multiple physical ESX servers. Once the storage has been configured on all physical servers, any server can run the Delphix Engine VM. This allows ESX clusters to survive physical server failure. In the event of failure, the VM is started on a different server, and appears to clients as an unexpected reboot with non-zero but minimal downtime. Depending on the length of the outage, this may cause a short pause in I/O and database activity, but longer outages can trigger timeouts at the protocol and database layers that result in I/O and query errors. Such long outages are unlikely to occur in a properly configured environment.

Automatic detection of failure in a HA environment does not work in all circumstances, and there are cases where the host, storage, or network can hang such that clients are deprived access, but the systems continue to appear functional. In these cases, a manual failover of the systems may be required.

When configuring a cluster, it is important to provide standby infrastructure with equivalent resources and performance characteristics. Asymmetric performance capabilities can lead to poor performance in the event of a failover. In the worst case of an over-provisioned server, this can cause widespread workload failure and inability to meet performance SLAs.

Snapshots

VMware provides storage-agnostic snapshots that are managed through the VMware Snapshot Manager. Use of VMware snapshots can, however, cause debilitating performance problems for write-heavy workloads due to the need to manage snapshot redo-log metadata. In order to provide an alternative snapshot implementation, while retaining the existing management infrastructure, VMware has created an API to allow storage vendors to supply their own snapshot implementation. This is only supported in ESX 5.1. Furthermore, the array must support the vStorage APIs. Consult the VMware documentation for supported storage solutions and the performance and management implications.

Storage-based snapshots, by virtue of being implemented natively in the storage array, typically do not suffer from such performance problems, and are preferred over VMware snapshots when available. When managing storage-based snapshots, it is critical that all LUNs backing a single VM be part of the same consistency group. Consistency groups provide write order consistency across multiple LUNs and allow snapshots to be taken at the same point in time across the LUNs. This must include all VM configuration, system VMDKs, and VMDKs that hold the dSources and VDBs. Each storage vendor presents consistency groups in a different fashion; consult your storage vendor documentation for information on how to configure and manage snapshots across multiple LUNs.

Replication

Site Recovery Manager (SRM) is a VMware product that provides replication and failover of virtual machines within a vSphere environment. It is primarily an orchestration framework, with the actual data replication performed by a native VMware implementation, or by the storage array through a storage replication adapter (SRA). A list of supported SRAs can be found in the VMware documentation. There is some performance overhead in the native solution, but not of the same magnitude as the VMware snapshot impact. SRAs provide better performance, but require that the same storage vendor be used as both source and target, and require resynchronization when migrating between storage vendors.
Storage-based replication can also be used in the absence of SRM, though this will require manual coordination when re-configuring and starting up VMs after failover. The VM configuration, as well as the storage configuration within ESX, will have to be recreated using the replicated storage.

The Delphix Engine also provides native replication within Delphix. This has the following benefits:

- The target system is online and active
- VDBs can be provisioned on the target from replicated objects
- A subset of objects can be replicated
- On failover, the objects are started in a disabled state. This allows configuration to be adjusted to reflect the target environment prior to triggering policy-driven actions.
- Multiple sources can be replicated to a single target

Note that the Delphix Engine currently only replicates data objects (dSources and VDBs) and environments (source and target services). It does not replicate system configuration, such as users and policies. This provides more flexibility when mapping between disparate environments, but requires additional work when instantiating an identical copy of a system after failover.

**Backup**

There is a large ecosystem of storage and VM-based backup tools, each with its own particular advantages and limitations. VMware provides Data Protector, but there are size limitations (linked to a maximum of 2TB of deduped data) that make it impractical for most Delphix Engine deployments. Most third-party backup products, such as Symantec NetBackup, EMC Networker, and IBM Tivoli Storage Manager, have solutions designed specifically for backup of virtual machines. Because the Delphix Engine is packaged as an appliance, it is not possible to install third party backup agents. However, any existing solution that can back up virtual machines without the need for an agent on the system should be applicable to Delphix as well. Check with your preferred backup vendor to understand what capabilities exist.

Some storage vendors also provide native backup of LUNs. Backup at the storage layer reduces overhead by avoiding data movement across the network, but loses some flexibility by not operating within the VMware infrastructure. For example, recreating the VM storage configuration from restored LUNs is a manual process when using storage based recovery.
Replication

These topics describe concepts and procedures for replicating data from one Delphix Engine to another.

- Replication Overview
- Replication Use Cases
- Replication User Interface
- Configuring Replication
- Replicas and Failover
- Failing Over a Replica
- Updating Replication User Credentials from Previous Versions
- Provisioning from Replicated Data Sources or VDBs
Replication Overview

This topic describes how to backup, restore, and replicate data between Delphix Engines.

Delphix provides the ability to replicate dSources and virtual databases (VDBs) between engines for disaster recovery and remote provisioning of VDBs.

Replication Features

As virtual appliances, it is possible to backup, restore, replicate, and migrate data objects between Delphix Engines using features of VMWare and the underlying storage infrastructure. Data objects include groups, dSources, VDBs, Jet Stream data templates and data containers, and associated dependencies. In addition to the replication capabilities provided by this infrastructure, native Delphix Engine replication provides further capabilities, such as the ability to replicate a subset of objects, replicate multiple sources to a single target, and provision VDBs from replicated dSources and VDBs without affecting ongoing updates. The topics under Backup and Recovery Strategies for the Delphix Engine provide more information on how to evaluate features of the Delphix Engine in relation to your backup and recovery requirements.

Replication is configured on the source Delphix Engine and copies a subset of dSources and VDBs to a target Delphix Engine. It then sends incremental updates manually or according to a schedule. For more information on configuring replication, see Configuring Replication.

You can use replicated dSources and VDBs to provision new VDBs on the target side. You can refresh these VDBs to data sent as part of an incremental replication update, as long as you do not destroy the parent object on the replication source. For more information, see the Provisioning from a Replicated dSource or VDB topic.

During replication, replicated dSources and VDBs are maintained in an alternate replica and are not active on the target side. In the event of a disaster, a failover operation can break the replication relationship. For more information on how to activate replicated objects, see Replicas and Failover.

How Replication Works

Delphix allows data objects to be replicated between Delphix Engines. These engines must be running identical Delphix versions, but otherwise they can be asymmetric in terms of engine configuration. In the event of a failure that destroys the source engine, you can bring up the target engine in a state matching that of the source. In addition, you can provision VDBs from replicated objects, allowing for geographical distribution of data and remote provisioning.

Replication can be run ad hoc, but it is typically run according to a predefined schedule. After the initial update, each subsequent update sends only the changes incurred since the previous update. Replication does not provide synchronous semantics, which would otherwise guarantee that all data is preserved on the target engine. When there is a failover to a replication target, some data is lost, equivalent to the last time a replication update was sent.

Replication is not generally suited for high-availability configurations where rapid failover (and failback) is a requirement. Failing over a replication target requires a non-trivial amount of time and is a one-way operation; to fail back requires replicating all data back to the original source. For cases where high availability is necessary, it is best to leverage features of the underlying hypervisor and storage platform. For more information on how to evaluate the use of Delphix Engine replication for your data recovery requirements, see the topics under Backup and Recovery Strategies for the Delphix Engine.

What Is Copied

Only database objects and their dependencies are copied as part of a backup or replication operation, including:

- dSources
When you select objects for replication, the engine will automatically include any dependencies, including parent objects, such as groups, and data dependencies such as VDB sources. This means that replicating a VDB will automatically include its group, the parent dSource, and the group of the dSource, as well as any environments associated with those databases. When replicating an entire engine, all environments will be included. When replicating a database or group, only those environments with the replicated databases are included.

**What Is Not Copied**

The following objects are not copied as part of a backup or replication operation:

- Users and roles
- Policies
- VDB (init.ora) configuration templates
- Events and faults
- Job history
- System services settings, such as SMTP

After failover, you must recreate these settings on the target.

**SSL Support**

The Delphix Engine includes Secure Sockets Layer (SSL) support for replication operations. During replication, the Delphix Engine will negotiate with its server peer to use SSL_RSA_WITH_RC4_128_MD5 as the cipher suite, and TLSv1 as the protocol.

**Resumable Replication**

Resumable replication enhances the current replication feature by allowing you to restart large, time-consuming initial replication or incremental updates from an intermediate point. A single replication instance can fail for a number of environmental and internal reasons. Previously, when you restarted a failed replication instance, replication required a full resend of all data transmitted prior to the failure. With resumable replication, no data is retransmitted. Replication is resumable across machine reboot, stack restart, and network partitions.

For example, suppose a replication profile has already been configured from a source to a target. A large, full send begins between the two that is expected to take weeks to complete. Halfway through, a power outage at the datacenter that houses the source causes the source machine to go down and only come back up after a few hours. On startup, the source will detect that a replication was ongoing, automatically re-contact the target, and resume the replication where it left off. In the user interface (UI) on the source, the same replication send job will appear as active and continue to update its progress. However, in the UI of the target, a new replication receive job will appear but will track its progress as a percentage of the entire replication.

In 4.1 and earlier releases, the replication component would always clean up after failed jobs to ensure that the Delphix Engine was kept in a consistent state and that no storage was wasted on unused data. With the addition of resumability, the target and source can choose to retain partial replication state following a failure to allow future replications to complete from that intermediate point. In the current release, the target and source will only choose to retain partial replication state following failures that leave them out of network contact with each other – for example, source restart, target restart, or network partition. Once network contact is re-established, the ongoing
replication will be automatically detected and resumed. The resumable replication feature is fully automated and does not require or allow any user intervention.

Replication will not resume after failures that leave the source and target connected. For example, if a storage failure on the target, such as out-of-space errors, causes a replication to fail, the source and target remain connected. As a result, the target will conservatively throw away all MDS and ZFS data associated with the failed replication. Nonetheless, resumable replication would begin during a source reboot, a target reboot and a network partition.

Related Topics

- Backup and Recovery Strategies for the Delphix Engine
- Replication User Interface
- Configuring Replication
- Provisioning from a Replicated dSource or VDB
- Replicas and Failover
Replication Use Cases

This topic describes use cases for Delphix replication.

Replication is a flexible tool that allows you to move dSources and virtual databases (VDBs) between Delphix Engines. These topics describe the ways in which you can use replication to meet different use cases.

Disaster Recovery

Replication is traditionally used to provide recovery in the event of disaster, where a datacenter or site is completely destroyed. Delphix replication may not be the only recovery solution in this scenario; consult the Backup and Recovery Strategies for the Delphix Engine topic to determine if it meets your requirements.

In a disaster recovery scenario, the target is kept in a passive state until the source system is lost. At this point, a failover is performed that breaks subsequent replication updates and activates objects so that they can be managed on the target side.

You can reconfigure environments on the target prior to failover if the infrastructure uses a different network topology or set of systems. Whether or not this is required depends on the nature of the failure at the primary site. If only the Delphix Engine is affected, and all of the source databases and target environments are unaffected, then the target can enable dSources and VDBs and reconnect to the original systems. If, on the other hand, the failure also destroyed the source and target systems, then those environments will have to be adjusted to point to the new systems on the target side. If there is not a 1:1 mapping, then you can migrate the VDBs to new systems on the target, and you can detach dSources and attach them to the standby system in the target environment.

Follow the best practices below to simplify failover and meet performance expectations in the event of a disaster:

- The environment should be as close to identical when it comes to available resources
  - Target hosts and systems should exist at the target that match those at the source
- The Delphix Engine should be provisioned with identical resources
- The network and storage topologies should be the same
- A 1:1 relationship between source and targets should be maintained
- The target should remain passive and not be actively used for other workloads
- Configuration of non-replicated objects, such as policies and users, should be retrieved via the command line interface (CLI) and saved so that they can be recreated after failover.

Geographically Distributed Development

The Delphix Engine allows for VDBs to be provisioned from replicated dSources and VDBs, as described in the Provisioning from a Replicated dSource or VDB topic. This allows dSources to be linked in a single central location and geographically distributed so that developers can provision VDBs remotely without having to sync from the source database in multiple locations.

In this environment, replication is never broken and failover never performed, unless the motivation for distribution is eliminated but remote VDBs need to be preserved. You can refresh remote VDBs as long as the parent objects continue to exist on the source. If they are deleted, then remote VDBs will continue to function but cannot be refreshed.

![Replication Diagram]

Because there is no failover, this topology can support more complex topologies such as 1-to-many and many-to-1. Chained replication (replicating from Site A -> Site B -> Site C) is not supported.

For geographical distribution, follow these best practices:

- Because each replication stream induces load on the source system:
  - Minimize the number of simultaneous replication updates
  - If possible, avoid heavy VDB workloads on the source
- Provision only from sources that are effectively permanent. Otherwise, remote VDBs cannot be refreshed once the source is deleted.
- Provision additional storage capacity on the target
  - Remotely provisioned VDBs can consume shared storage on the target even when the parent is deleted on the source

Migration

You can use replication to perform one-time migration of resources from one Delphix Engine to another. While the hypervisor provides tools to move virtual appliances between physical systems, there are times when migration is necessary, such as:

- Migrating between different physical storage
- Consolidating or distributing workloads across Delphix Engines

In these cases, replication can be used to copy a subset of objects across asymmetric topologies.

For migration, follow these best practices:

- Send full updates, followed by incremental updates, until the time required for incremental updates meets your downtime window
- Disable all objects to be migrated on the source, to ensure that they are not actively changing
- Send a final incremental update before failing over the target
- After failover, destroy any migrated objects on the source, or the entire engine
Replication User Interface

- Understanding Replication Functionality and User Interface
  - Replication in the Delphix Engine before 4.2 release
  - Replication in the Delphix Engine after 4.2 release
  - Understanding Sources for Replication
  - Understanding Targets for Replication
- User Interface Replication Screen
  - Interacting with the Replication Profiles Section
  - Interacting with the Create New Replication Profile Section
  - Interacting with the Replica Section
- Getting Started: Working in the Replication User Interface
  - Configuring Replication Profiles
  - Viewing and Editing an Existing Replication Profile
  - Viewing and Editing a Replica
- Related Links

Understanding Replication Functionality and User Interface

Replication in the Delphix Engine before 4.2 release

Prior to the 4.2 release, replication was managed by two separate dialog windows in the user interface (UI). The first, the Replication window, handled the creation and manipulation of a replication spec (replication on the source). The Namespaces window allowed you to view existing namespaces (replication on the target) and initiate failover.

Replication in the Delphix Engine after 4.2 release

In the 4.2 release, these two separate user interfaces have been collapsed into a single UI which is responsible for managing replication on both the source and the target. Moreover, the concept of a replication spec is now referred to as a "replication profile" (on the source), and namespaces are referred to as "replicas." Thus, replication consists of a profile-replica pair. As before, you can view and edit the profile on the source engine, whereas you can view the replica on the target engine.

<table>
<thead>
<tr>
<th>Version 4.1 and Earlier</th>
<th>Version 4.2 and Later</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replication Spec</td>
<td>Replication Profile</td>
</tr>
<tr>
<td>Namespace</td>
<td>Replicas</td>
</tr>
</tbody>
</table>

Understanding Sources for Replication

The Replication Profile section continues to handle the existing functionality but allows for the configuration of multiple replication profiles. This makes it possible to replicate objects from a single source to multiple targets. Each profile defines the set of data objects and the associated configuration between a single source and target.

Understanding Targets for Replication

The Replica section shows you the set of all objects in the replica and allows you to initiate failover. As in the old UI,
one Delphix Engine can have multiple “received replicas.” This means that the engine is the target of multiple sources of replication, or profiles.

**User Interface Replication Screen**

There are three interconnected sections that have been introduced to make the replication workflow easier: Replication Profile, Create New Replication Profile, and Replica.

**Interacting with the Replication Profiles Section**

The following screenshot and the descriptions below illustrate the capabilities in the Replication Profiles section. Click the screenshot for an enlarged view.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
</table>
| 1 | **Create Profile Button**  
Allows you to configure a Replication Profile |
| 2 | **Replication Profiles List**  
Provides a list of existing replication profiles. Click a profile in this list to view its details. |
| 3 | **Received Replicas List**  
Provides a list of all existing replicas on this Delphix Engine. Click a replica in this list to view its details. |
Status Box
Shows the replication status of the selected profile, including:

- The result of the most recent or current replication event
- Statistics for the replication run such as data transferred, duration, and average throughput

In the top left-hand corner, an icon summarizes the replication status. There are four possible status icons:

- ![Green Circle](image) This icon appears after a successful replication job.
- ![Blue Cloud](image) This icon appears while a replication job is in progress
- ![Red Square](image) This icon appears when a replication job has failed.
- ![Orange Triangle](image) This icon appears when a replication job was cancelled.

Configuration Options
Configuration Options for the selected replication profile. These include:

- Description – Free text description of the profile
- Target Engine – The Delphix Engine on the receiving end of this replication pair
- Automatic Replication – If enabled, shows the frequency and time that regular replication will be run
- Traffic options – Summarizes the traffic options with which this profile has been configured

Object Selection Tree
Shows all of the objects, such as groups, dSources, VDBs, and Jet Stream data layouts, that you have selected for replication in this replication profile. If you select Entire Delphix Engine, all objects on the engine will be replicated, and thus the tree is collapsed.

Replicate Now Button
Begins the replication process
Delete Button

Allows you to delete the current profile

Interacting with the Create New Replication Profile Section

Click the screenshot below for an enlarged view of the Create New Replication Profile section. The descriptions below provide more details of the functionality of this section.

Create New Profile Configuration Fields

Configuration fields include a replica profile name, a description of the replica profile, the name of the target engine you would like to use, the user name, and password. There are also automatic configurations available.

Object Selection Tree

Shows all of the objects, such as groups, dSources, VDBs, and Jet Stream data layouts, that you have selected to be replicated in this replication profile.

Create Profile Tab Button

Saves the new Replication Profile

Interacting with the Replica Section

Click the screenshot below for an enlarged view of the Replica section. The descriptions below provide more details of the functionality of this section.
Getting Started: Working in the Replication User Interface

To navigate to the Replication UI screen:

1. In the top navigation menu, click System.
2. Select Replication.
Configuring Replication Profiles

1. In the left-hand navigation section, click **Create Profile**.
2. Enter the **name** of the replication profile and an optional **description**.
3. For Target Engine, enter the **hostname** or **IP address** for the target Delphix Engine.
4. Enter the **username** and **password** of a user who has Delphix Admin-level credentials on the target Delphix Engine. If the username and password change on the target Delphix Engine, you must update these settings on the source Delphix Engine.
5. By default, automatic replication is disabled, meaning that you must trigger replication updates manually. To enable automatic replication, click the **Enabled** checkbox.
6. In the **Automatic Replication** field, enter the **Frequency** and **Starting Time** for replication updates to the target Delphix Engine. Once you have entered and saved your replication settings, you will also see an option to trigger replication immediately with the **Replicate Now** button.
   a. Note: Automatic replication uses Quartz for scheduling. Starting with Delphix version 4.2, the Quartz-formatted string is editable via the **Advanced** option. Please refer to the screenshot below:

![Screenshot of Replication Settings]

7. Under **Traffic Options**, select whether you want to **Encrypt** traffic, **Compress** traffic, or **Limit bandwidth** during replication updates.
   a. Note: the **Compress** option is enabled by default.
8. In the right-hand column, under **Objects Being Replicated**, click the boxes next to the objects you want to replicate.
Selected Objects

- Some selected objects may have dependencies – other objects that will be pulled into replication because they share data. For more details, see [http://docs.delphix.com/display/DOCS42/Replication+Overview#ReplicationOverview-WhatsCopied](http://docs.delphix.com/display/DOCS42/Replication+Overview#ReplicationOverview-WhatsCopied). Objects that will be replicated are confirmed with a blue chain link icon.
  - Note that this is not guaranteed to be the full set of dependent objects, but rather is a best guess. The full set of objects and their dependents will be calculated at the time of replication.

- When selecting objects, you can select the entire server (Entire Delphix Engine) or a set of groups, dSources, VDBs, and Jet Stream data layouts.
- When replicating a group, all dSources and VDBs currently in the group, or added to the group at a later time, will be included.
- If you select a Jet Stream data template, all data containers created from that template will be included. Likewise, if you select a data container, its parent data template will be included.
- If you select the entire server, all groups and Jet Stream objects will be included.
- Regardless of whether you select a VDB individually or as part of a group, the parent dSource or VDB (and any parents in its lineage) are automatically included. This is required because VDBs share data with their parent object. In addition, any environments containing database instances used as part of a replicated dSource or VDB are included as well.
- When replicating individual VDBs, only those database instances and repositories required to represent the replicated VDBs are included. Other database instances that may be part of the environment, such as those for other VDBs, are not included.

10. Click **Create Profile** to submit the new profile. This saves the replication profile details. If you leave the Create page prior to submitting the profile, the draft replication profile will be discarded.

**Viewing and Editing an Existing Replication Profile**

1. In the Replication Profiles screen, click a profile in the left-hand navigation area to select it. Replication configuration fields for the selected Profile will appear. These include Replica Profile Name, Description, Target Engine, User Name, Password, Automatic Replication fields, and Traffic Options.

2. To edit fields, click the edit icon next to the corresponding field or group of fields.

3. You will also see ✗ ✓, which allow you to edit or cancel changes.
   a. To commit to the edits and/or selections, click the Green Checkmark icon.
   b. To cancel the edits and/or selections, click the Red X icon.

**Viewing and Editing a Replica**

1. To view and edit existing replicas, select a replica from the left-hand navigation area under the Received Replicas section. You can edit the Name and Description fields. All other fields are view-only.

2. To edit fields, click the edit icon next to the corresponding field or group of fields.

3. You will also see ✗ ✓, which allow you to edit or cancel changes.
   a. To commit to the edits and/or selections, click the Green Checkmark icon.
   b. To cancel the edits and/or selections, click the Red X icon.
Related Links

- CLI Cookbook: Replication
- Configuring Replication
- Replicas and Failover
- Failing Over a Replica
Configuring Replication

This topic describes how to configure data replication between Delphix Engines.

**Prerequisites**

- The replication source and the replication target must be identical versions of the Delphix Engine (for example, Delphix Engine version 3.2)

**Configuring the Network**

Replication operates using a private network protocol between two Delphix Engines. Apart from standard network considerations for performance, no additional configuration is required for replication. Replication can run over dedicated networks by configuring routing to direct traffic destined for the target IP address over a specific interface. The replication process can recover from transient network outages, but extended outages may cause the process to start from the previous update.

**Enabling Port 8415 for Firewalls**

If there is a firewall between the source and target, port 8415 must be enabled to allow connections from the source to the target. A reverse path through the firewall is not required.

**Configuring the Replication Source Delphix Engine**

1. On the source Delphix Engine, go to **System > Replication**.
2. In the left-hand navigation section, click **Create Profile**.
3. Enter the **name** of the replication profile and an optional **description**.
4. For Target Engine, enter the **hostname** or **IP address** for the target Delphix Engine.
5. Enter the **username** and **password** of a user who has Delphix Admin-level credentials on the target Delphix Engine. If the username and password change on the target Delphix Engine, you must update these settings on the source Delphix Engine.
6. By default, automatic replication is disabled, meaning that you must trigger replication updates manually. To enable automatic replication, click the **Enabled** checkbox.
7. In the **Automatic Replication** field, enter the **Frequency** and **Starting Time** for replication updates to the target Delphix Engine. Once you have entered and saved your replication settings, you will also see an option to trigger replication immediately with the **Replicate Now** button.

**Encrypting Traffic**

Automatic replication uses Quartz for scheduling. Starting with Delphix version 4.2, the Quartz-formatted string is editable via the **Advanced** option.

8. Under **Traffic Options**, select whether you want to **Encrypt** traffic, **Compress** traffic, or **Limit bandwidth** during replication updates.
8. Encrypting Traffic

By default, replication streams are sent unencrypted. This provides maximum performance on a secure network. If the network is insecure, encryption can be enabled. Note that encrypting the replication stream will consume additional CPU resources and may limit the maximum bandwidth that can be achieved.

9. Compressing Traffic

By default, replication streams are compressed. In environments where network bandwidth is a constrained resource, compression has been shown to conserve bandwidth and optimize overall throughput achieved by replication. Enabling this option will consume additional CPU on the Delphix Engine during replication updates. Testing within Delphix environments has shown that enabling compression provides optimal throughput for bandwidth constrained environments with less than 1GigE of end-end bandwidth.

10. Enabling Compression with Encryption

In general, it is a good practice to enable compression together with encryption. Compression is applied to the data prior to encryption. Because compression is both faster and less expensive than encryption, it will typically lead to higher throughput with less CPU overhead.

11. Limiting Bandwidth

By default, replication will run at the maximum speed permitted by the underlying infrastructure. In some cases, particularly when a shared network is being used, replication can increase resource contention and may impact the performance of other operations. This option allows administrators to specify maximum bandwidth that Replication can consume.

9. In the right-hand column, under **Objects Being Replicated**, click the **checkboxes** next to the objects you want to replicate.
Selected Objects

- Some selected objects may have dependencies – other objects that will be pulled into replication because they share data. For more details, see [http://docs.delphix.com/display/DOCS42/Replication+Overview#ReplicationOverview-Whatiscopied](http://docs.delphix.com/display/DOCS42/Replication+Overview#ReplicationOverview-Whatiscopied). Objects that will be replicated are confirmed with a blue chain link icon.
- Note that this is not guaranteed to be the full set of dependent objects, but rather is a best guess. The full set of objects and their dependents will be calculated at the time of replication.

- When selecting objects, you can select the entire server (Entire Delphix Engine) or a set of groups, dSources, VDBs, and Jet Stream data layouts.
- When replicating a group, all dSources and VDBs currently in the group, or added to the group at a later time, will be included.
- If you select a Jet Stream data template, all data containers created from that template will be included. Likewise, if you select a data container, its parent data template will be included.
- If you select the entire server, all groups and Jet Stream objects will be included.
- Regardless of whether you select a VDB individually or as part of a group, the parent dSource or VDB (and any parents in its lineage) are automatically included. This is required because VDBs share data with their parent object. In addition, any environments containing database instances used as part of a replicated dSource or VDB are included as well.
- When replicating individual VDBs, only those database instances and repositories required to represent the replicated VDBs are included. Other database instances that may be part of the environment, such as those for other VDBs, are not included.

10. Click Create Profile to submit the new profile. This saves the replication profile details. If you leave the Create page prior to submitting the profile, the draft replication profile will be discarded.

Configuring Replication and Multiple Target Engines through the CLI

You can also configure replication on the Source Delphix Engine by using the replication spec in the command line interface. See the topics under CLI Cookbook: Replication for more information.

Enabling Configuration of Multiple Replication Profiles

Learn how to configure and use functionality for multiple replication profiles on the source using the replication profiles in the Replication User Interface.

Configuring the Target Delphix Engine

No additional configuration on the target is needed. Replicated objects will appear in an alternate replica that mirrors the original object layout. These replicas can be viewed through the System > Replication screen under the Received Replicas section (or namespace in the CLI). All replicated objects are read-only until the replica is failed over. For more information about managing replicas and how to activate a replica, see the topics Replicas and Failover and Failing Over a Replica.

Objects can be created and managed on the target server without affecting subsequent updates, though this can
cause conflicts on failover that require additional time to resolve. For disaster recovery use cases, it is recommended to keep the target passive and not create any local objects. This will avoid conflicts and guarantee a smooth failover operation.

Multiple sources can replicate to the same target, allowing for flexible geographical distribution of data. For disaster recovery this is not a recommended practice, as it increases the probability of conflicts on failover, and may oversubscribe resources on the target if multiple replicas are failed over and there is insufficient infrastructure to support the combined workloads.

**Related Links**

- [CLI Cookbook: Replication](#)
- [Replication User Interface](#)
- [Replicas and Failover](#)
- [Failing Over a Replica](#)
Replicas and Failover

This topic describes failover of replicated state.

Replication recreates objects on the target system in a replica that preserves object relationships and naming on the target server without interfering with active objects on the system. Objects within a replica are read-only and disabled until a replica is failed over, at which point they can be activated. VDBs and dSources within a replica can be used as the source for provisioning new VDBs.

- **Replicas**
- **Failover and Conflict Resolution**
- **Enabling Databases and Environments**
- **Restoring Policies and Users**

**Replicas**

A replica contains a set of replicated objects. These objects are read-only and disabled while replication is ongoing. To view replicated objects, select the **System > Replication** menu item and select the replica under **Received Replicas** (or **namespace** in the CLI). On this screen you can browse the contents of replicas, as well as fail over or delete individual replicas. As described in the **Replication Overview** topic, databases (dSources and VDBs) and environments are included within the Replica.

Deleting or failing over a replica will sever any link with the replication source. Subsequent incremental updates will fail, requiring the source to re-establish replication. Failover should only be triggered when no further updates from the source are possible (as in a disaster scenario).

Multiple replicas can exist on the system at the same time. Active objects can exist in the system alongside replicas without interfering with replication updates. VDBs and dSources within a replica can also be used as a source when provisioning. For more information see the **Provisioning from a Replicated dSource or VDB** topic.

**Failover and Conflict Resolution**

To activate the objects in a replica, you must first fail over the replica. This will sever replication and move the objects to the live system, where they can be manipulated in the same fashion as other objects on the system. Any active objects with conflicting names will cause an error at the time of failover. The error message will indicate which object(s) have conflicting names. The active objects must be renamed, since the replica objects are read-only, before the failover operation can complete successfully.

> Given that conflicting names prevent failover from succeeding, best practices in a disaster recovery situation are to leave the target system completely passive with no active objects until the time that a failover is required.

Once a replica is failed over, the objects are active but will be automatically disabled.

**Enabling Databases and Environments**

Objects may refer to states (IP addresses, mount paths, etc) that differ between the source and target system. Because of this, all databases and objects within a replica automatically start in the disabled state after a failover. This allows the administrator to alter configuration prior to enabling databases and environments, without the system inadvertently connecting to invalid systems.

After failover is complete, all dSources, VDBs, and environments must be explicitly enabled. Should any
configuration need to change for the target environment, this can be done prior to enabling the objects.

*Restoring Policies and Users*

Policies and users are not replicated or backed up, requiring that they be recreated on the target after a failover. There is no built-in mechanism to automate this within Delphix. It is recommended that all users and policies be backed up (using the CLI, web services or other manual means) so that they can be recreated on the target system after failover.
Failing Over a Replica

This topic describes the process of failing over a replica. Objects stored in a replica are read-only, and failing over a replica moves the replicated objects to the live system. After a failover all of the objects will appear in the system as if they had been created locally.

Prerequisites

A Delphix system that contains a replica is required, see the Replicas and Failover topic for an overview of what replicas are and what failover implies. For more information on configuring replication please refer to the Configuring Replication topics.

Procedure

1. Locate the replica to failover. The list of replicas can be accessed via the Received Replicas section of the System > Replication screen. Each replica has a default name that is the hostname of the source that sent the update. These names may be customized if desired. Each replica will list the databases and environments it contains.

   If this replica is the result of a replication update, check to see if the source Delphix appliance is still active. If so, then disable any dsoure or VDB that is part of the replica being failed over to ensure that only one instance is enabled. Dsources and VDBs can be disabled by going to the Databases > My Databases screen, finding the appropriate database, and toggling the enabled slider.

2. Click Failover and confirm the dialog that appears. This will pause while the replica is failed over.

3. Apply any configuration changes that are required to customize the objects for the system. This might include updating object state such as IP addresses, mount paths, or credentials. See the Replicas and Failover topic for more details.

4. Enable the environments that were failed over. The environments can be found by selecting Manage > Environments. The environments will be disabled, toggle the disabled slider to enable them.

5. Enable the dsources and VDBs that were failed over. The databases can be found by selecting Databases > My Databases. Each database will have a card, select the desired database card and toggle the disabled slider to enable them.

   Policies and users are not replicated, be sure to recreate them after you fail over.

Related Topics

- Replication User Interface
- Replicas and Failover
- Configuring Replication
Updating Replication User Credentials from Previous Versions

This topic describes how to update the replication user credentials from previous versions of the Delphix Engine.

Previous versions of the Delphix Engine performed authentication through an NDMP user that was configured separately from the normal Delphix users. With Delphix Engine 3.2, authentication is performed against native Delphix users.

Prerequisites

- A source and target replication host that were configured with a release prior to 3.2 and subsequently upgraded.
- On the target system, a Delphix user with domain privileges is required.

Procedure

1. On the source host, select System > Replication.
2. Enter the name and password for a Delphix user on the target who has domain privileges.
3. Save the configuration.

Related Links

- Configuring Replication
Provisioning from Replicated Data Sources or VDBs

This topic describes how to provision from a replicated dSource or VDB. The process for provisioning from replicated objects is the same as the typical VDB provisioning process except for the need to first select the namespace containing the replicated object.

Prerequisites
- You will need to have replicated a dSource or a VDB to the target host, as described in Replication Overview.
- You will need to have added a compatible target environment on the target host as described in Provisioning VDBs: An Overview.

Procedure
1. Login to the Delphix Admin application for the target host.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. In the list of replicas, select the replica that contains the dSource or VDB to be provisioned.
6. The provisioning process is now identical to the process for provisioning standard objects. For the details of how to provision VDBs for specific platforms, consult Provisioning VDBs: An Overview.

Post-Requisites
Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.

Related Links
- Replication Overview
- Database Provisioning Overview
- Provisioning VDBs: An Overview
- Provisioning VDBs from Oracle and Oracle RAC dSources
- Provisioning VDBs from SQL Server dSources
- Provisioning VDBs from PostgreSQL dSources
V2P: Virtual to Physical

These topics describe the concepts and procedures for exporting a virtual database (VDB) to a physical one, otherwise known as V2P.

- Virtual to Physical: An Overview
- V2P with an Oracle VDB
  - Move an Oracle VDB to a Physical ASM or Exadata Database
- V2P with a SQL Server VDB
- V2P with a PostgreSQL dSource or VDB
- V2P with an SAP ASE dSource or VDB — This topic describes the procedure for exporting a virtual database (VDB) to a physical one, also known as V2P.
- Manually Recovering a Database after V2P
- Customizing Target Directory Structure for Database Export
- V2P with Unstructured Files
Virtual to Physical: An Overview

This topic describes the basic concepts behind exporting a virtual database to a physical one, also known as V2P.

After you have created a dSource or a VDB, you can export its contents and log files to a physical database. This process, referred to as V2P, creates a set of directories in the target environment and populates them with the database data, log files, and scripts that are used to recover the physical database. You can automatically start the physical database recovery process as part of the V2P export, or you can use the scripts for a manual recovery. When the export process completes, the target environment will contain a copy of the database in its unvirtualized size, so before you begin the process, make sure the target directories you specify in the V2P wizard have enough capacity to hold the unvirtualized database.

V2P can be initiated by users with delphix_admin credentials, and object and group owners.
V2P with an Oracle VDB

This topic describes the procedure for exporting a virtual database (VDB) to a physical one, also known as V2P.

**Procedure**

1. Login to the Delphix Admin application as a Delphix Admin user, or as a group or object owner.
2. Select the dSource or VDB you want to export.
3. Select the snapshot of the dSource or VDB state you want to export.
4. If you want to export the state of the database from a specific point in time, slide the LogSync slider on the top of the snapshot card to the right, and then select the point in time from which you want to create the export.
5. Click V2P.
6. Select the target environment.
7. Enter the Target Directory for the export.
   The target directory you enter here must exist in the target environment, and you must have permission to write to it. For more information on user requirements for target environments, see Requirements for Oracle Target Hosts and Databases.
8. Select whether or not to Open Database After Recovery.
   If you do not select this option, the Oracle database will not undergo open resetlogs, and the database will not be available for read/write access. This can be useful if the files are to be used to restore an existing data file for recovery purposes. You can use the scripts that are created in the target environment to complete the database open process at a later time. For more information, see Manually Recovering a Database after V2P.
9. Click Advanced to customize data transfer settings, customize the target directory layout, enter any database configuration parameters or enter file mappings from the source environment to the target. For more information, see Customizing Target Directory Structure for Database Export, Customizing Oracle VDB Configuration Settings and Customizing VDB File Mappings. The data transfer settings are described below:
   - **Compression** – Enable compression of data sent over the network. Default is Off.
   - **Encryption** – Enable encryption of data sent over the network. Default is Off.
   - **Bandwidth Limit** – Select the network bandwidth limit in units of megabytes per second (MB/s) between the Delphix Engine and the target environment. Default is 0, which means no bandwidth limit is enforced.
   - **Number of Connections** – Select the number of transmission control protocol (TCP) connections to use between the Delphix Engine and the target environment. Multiple connections may improve network throughput, especially over long-latency and highly-congested networks. Default is 1.
   - **Number of Files to Stream Concurrently** – Select the number of files that V2P should stream concurrently from the Delphix Engine to the target environment. Default is 3.
10. Click Next.
11. Select whether you want to have an email sent to you when the export process completes.
12. Click Finish.

**Post-Requisites**

If you did not select for Open Database After Recovery, follow the instructions in Manually Recovering a Database after V2P to complete the database open process.

**Resumable V2P**
Resumable V2P is a capability that allows you to suspend a V2P operation and then resume it at a later time, without redoing any of the work already completed. For example, any portion of a file that has already been transferred to the target environment is not re-sent. For an entire file that has already been transferred, no part is re-sent.

The image below presents a progress bar, a stop button and a pause button while a V2P is running. To manually suspend a V2P operation:

1. Click the **pause** button.

![Image of pause button](image1.png)

The next image represents a message alert generated after a V2P job has been suspended. To manually resume the job:

1. Click the **play** button.

![Image of resume button](image2.png)

**Recoverable Errors**

Broadly speaking, a "recoverable error" is an error condition caused by a disruption in the environment or on the target host, not errors in the actual V2P operation. Examples of recoverable errors include:

- A timeout due to a network outage
- Running out of disk space on the target environment
- An inability to create directories or files on the target environment

You can often address recoverable errors by taking some action to fix the problem, such as freeing up space on the target environment.

**Auto-Suspend**

A V2P operation that encounters a recoverable error is auto-suspended: it appears as a suspended job in the user interface (UI), with a message detailing the error condition. Once you have fixed the error, you can simply resume the job. Alternatively, you can cancel the job. Just as when you manually suspend and resume a job, any portion of a file that has been transferred, including possibly the entire file itself, is not re-sent when the job resumes.

**Video**
Related Links

- Requirements for Oracle Target Hosts and Databases
- Manually Recovering a Database after V2P
- Customizing Oracle VDB Configuration Settings
- Customizing VDB File Mappings
- Move an Oracle VDB to a Physical ASM or Exadata Database
Move an Oracle VDB to a Physical ASM or Exadata Database

This topic describes how to move a virtual database (VDB) into a physical database stored on Oracle Automatic Storage Management (ASM) disk groups. This is a scripted procedure that is assisted, but not fully automated, by Delphix. A full restore of the original source ASM database from a Delphix dSource can be achieved using this procedure. No intermediate storage is needed; the database files are moved directly from Delphix into the source database ASM diskgroups.

This procedure applies to stand-alone and RAC databases residing on ASM disk groups, including databases residing in an Oracle Exadata machine.

Prerequisites

- Provision a VDB on the target machine that is running Oracle ASM or Exadata
- Create an ASM disk group that will contain all the database files. Optionally create a separate disk group for redo log files.
- Where multiple disk groups are used for datafiles, the reference move-to-asm.sh script will need to be modified. Oracle best practices recommend a single datafile disk group.

Oracle Versions

This procedure applies to all Oracle RDBMS Versions supported by Delphix.

Procedure

1. Download the reference shell script move-to-asm.sh on the target machine where the VDB instance and ASM instance are running.
2. Ensure that Oracle environment variables ORACLE_HOME ORACLE_SID and CRS_HOME (RAC only) are correctly set for the VDB that needs to be moved.
3. Execute the script move-to-asm.sh as the Environment User who provisioned the single instance VDB. For a RAC VDB, the Environment User selected to execute move-to-asm.sh must be the Oracle installation owner. This is due to an Oracle restriction that only the installation owner can invoke srvctl to add or remove database configurations.

```
move-to-asm.sh [-noask] [-parallel #] [-dbunique db_unique_name] <data_diskgroup> [<redo_diskgroup>]
```

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-noask [optional]</td>
<td>Do not prompt for confirmation before moving the VDB. Default is to prompt.</td>
</tr>
<tr>
<td>-parallel [optional]</td>
<td>Number of RMAN channels used to move the VDB to ASM. Default is 8.</td>
</tr>
<tr>
<td>-dbunique [optional]</td>
<td>Database unique name for the resulting physical database. Default is VDB unique name.</td>
</tr>
<tr>
<td>&lt;data_diskgroup&gt; [required]</td>
<td>Target ASM disk group for data, server parameter and control files.</td>
</tr>
<tr>
<td>&lt;redo_diskgroup&gt; [optional]</td>
<td>Target ASM disk group for redo log files. Default is data_diskgroup.</td>
</tr>
</tbody>
</table>
$ /home/oral120/scripts/delphix/move-to-asm.sh
Usage: move-to-asm.sh [-noask] [-parallel #] [-dbunique db_unique_name] <data_diskgroup> [<redo_diskgroup>]
$ /home/oral120/scripts/delphix/move-to-asm.sh -noask -dbunique davis +DATA +LOG

Virtual-to-ASM script (move-to-asm.sh v1.5) for Delphix 3.x
Copyright (c) 2013 by Delphix.

Moving database db52temp to ASM: started at Mon Jun 10 11:48:31 EDT 2013
db_unique_name => db52
ORACLE_SID => db52
ORACLE_HOME => /opt/app/oracle/product/11.2.0/dbhome_1
Datafile diskgroup => +VIIL
RMAN Channels => 8

Generate script to move tempfiles to ASM
Generate script to drop old tempfiles
Generate script to drop offline tablespace
Generate script to make read-only tablespace read-write
Make read-only tablespace read-write
Remove offline tablespaces
Updating server parameter file with ASM locations
Move spfile to ASM
Move datafiles to ASM: started at Mon Jun 10 11:49:25 EDT 2013
Move datafiles to ASM: completed at Mon Jun 10 11:56:09 EDT 2013
Startup database with updated parameters
Move tempfiles into ASM
Move Online logs
Restore any read-only tablespace
Remove old tempfiles
Database db52 moved to ASM: completed at Mon Jun 10 11:57:19 EDT 2013

Final Steps to complete the move to ASM:
1) Delete VDB on Delphix.
2) Copy new init.ora: cp
/home/oral120/scripts/delphix/initdb52_run8396_moveasm.ora
/opt/app/oracle/product/11.2.0/dbhome_1/dbs/initdb52.ora
3) Startup database instance.
4) Modify initialization parameters to match source and restart.
Source parameters are restored at
/home/oral120/scripts/delphix/source_initdb52.ora

4. Alternatively, enter move-to-asm.sh as a Post Script when provisioning the VDB. This will provision and move the VDB into ASM diskgroups in a single flow.

You must specify the -noask option to execute in non-interactive mode. For example:
/delphix/scripts/move-to-asm.sh -noask -parallel 10 +DATA +REDO

See Using Pre- and Post-Scripts with dSources and SQL Server VDBs for more information.
CRS_HOME Environment Variable for Releases Prior to 3.1.4

For releases prior to 3.1.4, move-to-asm.sh must be modified to add the CRS_HOME environment variable for RAC VDB target environments.

Example: Restoring a RAC ASM Database from a dSource TimeFlow

- Source Database: db_unique_name = proddb, db_name = proddb
- 4 RAC Instance ORACLE_SID = prod1, prod2, prod3, prod4.
- All datafiles are contained in ASM diskgroup +DATA
- Redo log files are in diskgroup +LOG.

1. Shut down all RAC instances for proddb. The entire database must be shut down before a full restore.
   Issue srvctl stop database -d proddb
2. Remove all files in the +DATA and +LOG diskgroups using asmcmd.
3. Provision a VDB from the proddb dSource timeflow to RAC node 1 (SID prod1) with a Post Script to move into ASM:
   
   a. In the Provision VDB wizard, set Database Unique Name to proddbtemp, set SID to prod1, set Database Name to proddb.
   b. Set Post Script to /delphix/scripts/move-to-asm.sh -noask -dbunique proddb +DATA +LOG

This method preserves the Database Unique Name, Database Name and SID of the original source database when restoring from a dSource timeflow.

Script Output

The move-to-asm.sh script generates several output files. These files are all written to the working directory of the script:

1. move-to-asm.sh_<oracle_sid>_run<process-id>.log – the log file for the operation
2. init<oracle_sid>_run<process-id>_moveasm.ora – the init.ora parameter file created for the ASM DB instance
3. source_init<oracle_sid>.ora – the init.ora for the source database (from which the VDB was created)

Post-Requisites

Final steps to be manually executed are displayed when the script completes and are written to the execution output log.

1. Delete the Delphix VDB that was moved.
2. For Single Instance only: copy generated init.ora parameter file to the default $ORACLE_HOME/dbs/init<$ORACLE_SID>.ora
3. For Single Instance only: startup the physical database that will now run on ASM.
A RAC database is automatically started up by the `move-to-asm.sh` script using `srvctl`.

4. Modify initialization parameters to match the original source database parameters, if necessary. As a convenience to assist with this step, the source database parameters are restored as `source_init<$ORACLE_SID>.ora`

**Related Links**

- [Requirements for Oracle Target Hosts and Databases](#)
V2P with a SQL Server VDB

This topic describes how to perform the Virtual to Physical (V2P) process with a SQL Server virtual database (VDB).

Procedure

1. Log into the Delphix Admin application as a Delphix Admin user, or group or object owner.
2. Select the dSource or VDB you want to export.
3. Select the snapshot of the dSource or VDB state you want to export.
4. If you want to export the state of the database from a specific point in time, slide the LogSync slider on the top of the snapshot card to the right, and then select the point in time from which you want to create the export.
5. Click V2P.
6. Select the target environment.
7. Enter the Target Directory for the export.
   The target directory you enter here must exist in the target environment, and you must have permission to write to it. See Requirements for SQL Server Target Hosts and Databases for more information on user requirements for target environments.
8. Select an option for Run recovery after V2P.
   If you select No, you can use the scripts that are created in the target environment to manually recover the database at a later time. See Manually Recovering a Database after V2P for more information.
9. Click Advanced to customize the target directory layout. See Customizing Target Directory Structure for Database Export for more information.
10. Click Next.
11. Select whether you want to have an email sent to you when the export process completes, and then click Finish.

Post-Requisites

If you selected No for Run Recovery after V2P, follow the instructions in Manually Recovering a Database after V2P to complete the V2P process.

Related Links

- Requirements for SQL Server Target Hosts and Databases
- Manually Recovering a Database after V2P
V2P with a PostgreSQL dSource or VDB

This topic describes the procedure for performing virtual to physical (V2P) operations with a PostgreSQL VDB.

Procedure

1. Log into the Delphix Admin application as a Delphix Admin user, or as a group or object owner.
2. Select the dSource or VDB you want to export.
3. Select the snapshot of the dSource or VDB state you want to export.
4. Click V2P.
5. Select the target environment.
6. Enter the Target Directory for the export. The target directory you enter here must exist in the target environment, and you must have permission to write to it. See Requirements for PostgreSQL Target Hosts and Databases for more information on user requirements for target environments.
7. Enter a Port Number. This is the TCP port the exported database will listen on.
8. Click Advanced to customize the target directory layout, or enter any database configuration parameters. See Customizing Target Directory Structure for Database Export, Customizing PostgreSQL VDB Configuration Settings for more information.
9. Click Next.
10. Review the Target Environment configuration information, and then click Finish.

Related Links

- Requirements for PostgreSQL Target Hosts and Databases
- Customizing Target Directory Structure for Database Export
- Customizing VDB Configuration Settings
V2P with an SAP ASE dSource or VDB

This topic describes the procedure for exporting a virtual database (VDB) to a physical one, also known as V2P.

- **Requirements**
- **Procedure**
- **Related Links**

Requirements

Before you perform the V2P operation, you must have created a database on the target instance into which you will load the exported data. It must be sufficiently large, and you must have created it with the **for load** SAP ASE option.

⚠️ The Delphix Engine will initiate a load command using the database specified. The V2P operation will overwrite any existing data in this database.

Procedure

1. Login to the **Delphix Admin** application as a **Delphix Admin** user, or as a group or object owner.
2. Select the **dSource** or **VDB** you want to export.
3. Select the **snapshot** of the dSource or VDB state you want to export.
4. Click **V2P**.
5. Select the **target environment**.
6. Under **Installation**, select which **instance** that you want to export to.
7. Enter the **Name** of the database on the target instance into which you want to load the exported data.
8. Select whether or not to **Run Recovery After V2P**. When this option is set, the Delphix Engine will online the database when the export is done.
9. Click **Next**.
10. Select whether you want to have an email sent to you when the export process completes.
11. Click **Finish**.

Related Links

- **Requirements for SAP ASE Environments**
Delphix Engine 4.2 User Guide © 2015 Delphix

Manually Recovering a Database after V2P
This topic describes how to manually recover a database after the V2P process.
If you select No as the option for Run recovery after V2P during the V2P export process, you can use the scripts
that are created in the script directory for your database instance in the target environment to manually recover
the database.
V2P for PostgreSQL VDBs
This does not apply for PostgreSQL since after V2P PostgreSQL VDBs are automatically
recovered by the Delphix Engine.

Procedure
1. In the V2P target environment, navigate to the scripts directory for your exported database instance.
You can find the scripts in a sub-directory named for that specific database instance. For Oracle databases,
the path is <target_directory>/<db_unique_name>/script/<instance name> . For SQL Server
databases, the path is <target_directory>\<db_name>\scripts.
2. For Oracle databases, locate the scripts recover-vdb.sh and open-vdb.sh. Run them in that order.
For SQL Server databases, locate the script Provision.ps1 and run it.
3. For SQL Server databases, when the script completes, Refresh the target environment for it to discover the
recovered database.
For Oracle databases, add the recovered database to /etc/oratab and Refresh the target environment for
it to discover the recovered database.

583


Customizing Target Directory Structure for Database Export

This topic describes how to customize the target directory layout for database export.

In the V2P export process, it may be necessary to customize the target directory structure which the files will be exported to. The following is the default directory structure:

- **Data files**: `<target directory>/data`
- **Archive files**: `<target directory>/archive`
- **Temp files**: `<target directory>/temp`
- **External files**: `<target directory>/external`
- **Script files**: `<target directory>/script`

Note: The example on this page uses `/` for file separators which is relevant for Unix and Linux environments. If the target environment is Windows, the file separator will be `\`.

The following procedure describes how to customize the directory layout.

**Procedure**

1. During the virtual to physical export process, click **Advanced** in the V2P Wizard to see the target directory options.
2. You can customize any of the following:
   - **Data Directory**
   - **Archive Directory**
   - **Temp Directory**
   - **External Directory**
   - **Script Directory**
3. Each directory will then be concatenated to the **Target Directory** separated by the appropriate separator.

Any one of **Target Directory, Data Directory, Archive Directory, Temp Directory, External Directory, Script Directory** can be blank. However, the combination of the fields must form an absolute path.

- **Data files**: `<target directory>/<data directory>`
- **Archive files**: `<target directory>/<archive directory>`
- **Temp files**: `<target directory>/<temp directory>`
- **External files**: `<target directory>/<external directory>`
- **Script files**: `<target directory>/<script directory>`

**Examples**

*Target directory is not empty*

This means all target directories have a common root.

<table>
<thead>
<tr>
<th>Input</th>
<th>Final Directory Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Target directory is empty

All target directories may not have a common root. Note that external files and temp files share the same common root.

<table>
<thead>
<tr>
<th>Input</th>
<th>Final Directory Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Directory:</td>
<td>Data files: /mytarget/mydata</td>
</tr>
<tr>
<td>Data Directory:</td>
<td>Archive files: /mytarget/myarchive</td>
</tr>
<tr>
<td>Archive Directory:</td>
<td>Temp files: /mytarget/mytemp</td>
</tr>
<tr>
<td>Temp Directory:</td>
<td>External files: /mytarget/myexternal</td>
</tr>
<tr>
<td>External Directory:</td>
<td>Script files: /mytarget/myscript</td>
</tr>
<tr>
<td>Script Directory:</td>
<td></td>
</tr>
</tbody>
</table>

Target directory is empty and data directory /

Combined with **Customizing VDB File Mappings**, exporting data files to separate file systems is possible. In this example, a.dbf and b.dbf can be exported to /filesystem1 and /filesystem2 respectively.

<table>
<thead>
<tr>
<th>Input</th>
<th>Final Directory Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Directory:</td>
<td>Data files:/filesystem1/a.dbf</td>
</tr>
<tr>
<td>Data Directory:</td>
<td>/filesystem2/b.dbf</td>
</tr>
<tr>
<td>Archive Directory:</td>
<td>Archive files: /myarchive</td>
</tr>
<tr>
<td>Temp Directory:</td>
<td>Temp files: /mytarget/temp</td>
</tr>
<tr>
<td>External Directory:</td>
<td>External files: /mytarget/myexternal</td>
</tr>
<tr>
<td>Script Directory:</td>
<td>Script files: /myscript</td>
</tr>
<tr>
<td>File mappings:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• a.dbf: /filesystem1/a.dbf</td>
</tr>
<tr>
<td></td>
<td>• b.dbf: /filesystem2/b.dbf</td>
</tr>
</tbody>
</table>

Target directory is empty and one of the sub directories is empty would result in error

<table>
<thead>
<tr>
<th>Input</th>
<th>Final Directories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Directory:</td>
<td></td>
</tr>
<tr>
<td>Data Directory:</td>
<td></td>
</tr>
<tr>
<td>Archive Directory:</td>
<td></td>
</tr>
<tr>
<td>Temp Directory:</td>
<td></td>
</tr>
<tr>
<td>External Directory:</td>
<td></td>
</tr>
<tr>
<td>Script Directory:</td>
<td></td>
</tr>
<tr>
<td>Target Directory:</td>
<td>INVALID</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Data Directory:</td>
<td>/mydata</td>
</tr>
<tr>
<td>Archive Directory:</td>
<td></td>
</tr>
<tr>
<td>Temp Directory:</td>
<td>/mytarget/temp</td>
</tr>
<tr>
<td>External Directory:</td>
<td>/mytarget/myexternal</td>
</tr>
<tr>
<td>Script Directory:</td>
<td>/myscript</td>
</tr>
</tbody>
</table>
V2P with Unstructured Files

This topic describes the procedure for performing virtual to physical (V2P) operations with unstructured files.

V2P Not Supported for Unstructured Files on Windows

V2P is not supported for unstructured files on Windows environments. Similar results to V2P may be achieved by provisioning a vFiles and copying data out of the vFiles to the local machine.

Procedure

1. Log into the Delphix Admin application as a Delphix Admin user, or as a group or object owner.
2. Select the dataset you want to export.
3. Select the snapshot you want to export.
4. Click V2P.
5. Select the target environment.
6. Enter the Mount Path for the export. The directory you enter here must exist in the target environment, and you must have permission to write to it. See Managing Unix Environments for more information on user requirements for target environments.
7. Click Next.
8. Review the Target Environment configuration information, and then click Finish.

Related Links

- Managing Unix Environments
- Virtual to Physical: An Overview
Cross-Platform Provisioning of Oracle Data Sources

These topics describe how to use the cross-platform provisioning feature of the Delphix Engine to enable agile data migration of Oracle data sources.

- Cross-Platform Provisioning of Oracle dSources: Overview
- Enabling Oracle dSources for Cross-Platform Provisioning
- Provisioning an Oracle dSource across Platforms
- Creating Scripts for Cross-Platform Provisioning
Cross-Platform Provisioning of Oracle dSources: Overview

This topic describes basic concepts behind migrating Oracle databases from Unix to Linux platforms. This feature is available in the Delphix Agile Migration Package. Contact sales@delphix.com for more information about obtaining this package.

Delphix Engine cross-platform provisioning automates existing Oracle best practices to create a Linux Oracle database with the same structure and data as an existing Unix Oracle database. The process uses the underlying Oracle Transportable Tablespaces and RMAN CONVERT DATAFILES technologies to efficiently convert Oracle datafiles to the Linux format. In addition, the Delphix Engine cross-platform provisioning functionality utilizes algorithms that are unique to the Delphix File System (DxFS) to detect similarities between the Unix datafiles and converted Linux datafiles, allowing the converted database to be stored in less than 5/100 of the space that would normally be required.

Requirements

The underlying Oracle technology used to transform to Linux imposes several requirements, including:

- Encryption can not be used
- Tablespace Transport Set must be self-contained
- Tablespaces with XML types can not be used before Oracle version 11.2
- Advanced queues versions 8 or later
- Spatial indexes can not be used before Oracle version 11.2

These requirements are checked by Transformation Validation, as described in Enabling Oracle dSources for Cross-Platform Provisioning. Creating Scripts for Cross-Platform Provisioning describes how to modify the database to meet these requirements.

Related Links

- Enabling Oracle dSources for Cross-Platform Provisioning
- Creating Scripts for Cross-Platform Provisioning
Enabling Oracle dSources for Cross-Platform Provisioning

This topic describes how to enable a Unix Oracle database for conversion to Linux.

- **Prerequisites**
- **Procedure**
- **Related Links**

**Prerequisites**

- A source Unix Oracle database
  This can be a dSource or a VDB.
- A Unix staging environment
  This environment must be the same platform and Oracle version as the source database. See Enabling Validated Sync for Oracle for information on designating a staging environment.
- The default OS user for the staging host must have access to the Oracle installation that will be used as the staging environment.
- A Linux provisioning environment
  This environment must be the same Oracle version as the source database. We recommend that this environment have a fast network link to the Delphix Engine, because it needs to process all blocks in the database when converting a database to Linux. See General Network and Connectivity Requirements and Network Performance Configuration Options for general information about network requirements and configuration for the Delphix Engine.

**Procedure**

1. Log into the Delphix Admin application using delphix_admin credentials
2. In the Manage menu, select Databases > My Databases.
3. Select the Oracle dSource that you want to use for cross-platform provisioning.
4. Click the dSource's Expand icon to open the dSource card, then click the Flip icon on the card to view the back.
5. On the back of the dSource card, click the Linux tab.
6. In the lower right corner of the dSource card, click the green Validate Transformation button. The validation process will create a temporary VDB on the Unix staging environment, and run SQL commands against it to verify that the database structure meets the requirements of the underlying Oracle platform conversion technology. Depending on the size of the dSource, this may take several minutes. See Cross-Platform Provisioning of Oracle dSources: Overview for more information about the specific database requirements that will checked during this process.
7. If the validation process is successful, green check marks will appear next to each validation requirement, and a gold database icon will appear next to the dSource name in the Databases panel. If the dSource does not pass the validation process, a red X will appear next to the requirement. See Creating Scripts for Cross-Platform Provisioning for more information on how to correct these violations of the cross-platform provisioning requirements.

**Related Links**

- Enabling Validated Sync for Oracle
- General Network and Connectivity Requirements
• Network Performance Configuration Options
• Creating Scripts for Cross-Platform Provisioning
Provisioning an Oracle dSource across Platforms

This topic describes how to provision a Unix Oracle virtual database (VDB) to a Linux environment.

Prerequisities

- A Unix Oracle dSource or VDB that has passed the validation checks for cross-platform provisioning as described in Enabling Oracle dSources for Cross-Platform Provisioning
- A Unix staging environment
  This environment must be the same platform and Oracle version as the source database. See Enabling Validated Sync for Oracle for information on designating a staging environment.
- A Linux provisioning environment
  This environment must be the same Oracle version as the source database. We recommend that this environment have a fast network link to the Delphix Engine, because it needs to process all blocks in the database when converting a database to Linux. See Network and Connectivity Requirements and Network Performance Configuration Options for general information about network requirements and configuration for the Delphix Engine.

Procedure

1. Log into the Delphix Admin application using delphix_admin credentials.
2. In the Manage menu, select Databases > My Databases if the Databases panel is not visible.
3. In the Databases panel, select an Oracle dSource that has passed the validation checks for cross-platform provisioning.
   Eligible dSources will have a gold database icon next to the dSource name, as shown in the dSource Icon Reference.
4. Select a provision point for the virtual database.
   See Provisioning an Oracle VDB for information on using Snapshots, LogSync, and SCN Numbers as provision points.
5. Click Transform to Linux.
6. In the Linux Transformation VDB wizard, select a Linux environment where you want to provision the VDB, and follow the steps for configuring the new VDB as described in Provisioning an Oracle VDB.
   When the Linux transformation process completes, a VDB will be created with the transformed database running on Linux. You should be aware that the transformation process can be time and resource intensive, because Oracle must read and convert all blocks in the database.

Related Links

- Enabling Oracle dSources for Cross-Platform Provisioning
- Enabling Oracle Pre-Provisioning
- Network and Connectivity Requirements
- Network Performance Configuration Options
- dSource Icon Reference
- Provisioning an Oracle VDB
Creating Scripts for Cross-Platform Provisioning

This topic describes the process for using scripts to correct violations that occur when a dSource does not pass validation checks for cross-platform provisioning.

As described in Enabling Oracle dSources for Cross-Platform Provisioning, a dSource may not pass all the validation checks that are necessary for it to be eligible for cross-platform provisioning. Failure to pass these checks is usually the result of an issue with objects in the database itself. For example, there cannot be any user objects in system tablespaces, and other requirements as described in Cross-Platform Provisioning of Oracle dSources: Overview. These validation violations must be corrected by modifying the database objects, which can be accomplished by uploading a Transformation Script. This script is a series of SQL commands that will be run against the temporary virtual database (VDB) before the validation process executes. For example, the Transformation Script might move user objects from the system tablespace to a user tablespace. The Delphix Customer Services group can help you with the initial creation of these scripts.

- **Procedure**
- **Related Links**

**Procedure**

1. Log into the Delphix Admin application using delphix_admin credentials.
2. If the Databases panel is not visible, select Manage > Databases > My Databases.
3. In the Databases panel, select the dSource that did not pass the cross-platform provisioning validation checks.
4. Click the Expand icon for the dSource to view its card.
5. Click the Flip icon to view the back of the card.
6. Click the Linux tab.
7. Click the Upload Transformation Script icon in the lower-right corner of the card.
8. Click Choose a File to Upload and navigate to the location of the script, then click Choose. The file will automatically upload when you click Choose.

⚠️ The Transformation Script must be an SQL or plain text file otherwise the upload will fail.

9. Click the Validate Transformation icon to execute the script against the temporary virtual database.

**Related Links**

- **Enabling Oracle dSources for Cross-Platform Provisioning**
Unstructured Files
Unstructured Files: Getting Started

This topic describes basic concepts for creating and managing unstructured files sources.

The term “unstructured files" refers to data stored in a filesystem that is NOT usually accessed by a DBMS or similar software. Unstructured files can consist of anything from a simple directory to the root of a complex application like Oracle Enterprise Business Suite.

Like with other data types, you can configure a dSource to sync periodically with a set of unstructured files external to the Delphix Engine. The dSource is a copy of these physical files stored on the Delphix Engine. On Unix platforms, dSources are created and periodically synced by an implementation of the rsync utility. On Windows, files are synced using the robocopy utility, which is distributed with Windows.

From dSources, you can provision “vFiles," which are virtual copies of data that are fully functional read write copies of the original files source. You can mount vFiles across one target environment or many.
Unstructured Files Environment Requirements

This section describes the environment requirements that must be satisfied before you can link, virtualize, or provision unstructured files. These requirements include expectations for operating system and network configuration.

- **Unstructured Files on Unix Environments**
  - Requirements for Unix Environments
  - Network and Connectivity Requirements for Unix Environments
  - Sudo Privilege Requirements and File Configurations
  - Managing Unix Environments
    - Adding a Unix Environment
    - Managing Unix Environment Users

- **Unstructured Files on Windows Environments**
  - Requirements for Windows Environments
  - Network and Connectivity Requirements for Windows Environments
  - Managing Windows Environments
    - Adding a Windows Environment
    - Managing Windows Environment Users
Unstructured Files on Unix Environments

These topics describe requirements for Unix source and target environments hosting unstructured files.

- Requirements for Unix Environments
- Network and Connectivity Requirements for Unix Environments
- Sudo Privilege Requirements and File Configurations
- Managing Unix Environments
  - Adding a Unix Environment
  - Managing Unix Environment Users
### Requirements for Unix Environments

This topic outlines the supported operating systems (OSs) for use on Unix source and target environments.

### Supported Operating Systems

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Version</th>
<th>Processor Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris</td>
<td>9, 10, 11</td>
<td>SPARC</td>
</tr>
<tr>
<td>Solaris</td>
<td>10, 11</td>
<td>x86_64</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux</td>
<td>4.7, 4.8, 4.9</td>
<td>x86_64</td>
</tr>
<tr>
<td></td>
<td>5.3 - 5.10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.0 - 6.5</td>
<td></td>
</tr>
<tr>
<td>Oracle Enterprise Linux</td>
<td>5.3 - 5.10</td>
<td>x86_64</td>
</tr>
<tr>
<td></td>
<td>6.0 - 6.5</td>
<td></td>
</tr>
<tr>
<td>Novell SUSE Linux Enterprise</td>
<td>10, 10SP1, 10SP2, 10SP3</td>
<td>x86_64</td>
</tr>
<tr>
<td>Server</td>
<td>11, 11SP1</td>
<td></td>
</tr>
<tr>
<td>AIX</td>
<td>5.3, 6.1, 7.1</td>
<td>Power</td>
</tr>
<tr>
<td>HP-UX</td>
<td>11i v2 (11.23)</td>
<td>IA64</td>
</tr>
<tr>
<td></td>
<td>11i v3 (11.31)</td>
<td></td>
</tr>
</tbody>
</table>

Delphix supports all 64-bit OS environments for source and target, though 64-bit Linux environments also require that a 32-bit version of glibc is installed.

**Required HP-UX patch for Target Servers**

PHNE_37851 - resolves a known bug in HP-UX NFS client prior to HP-UX 11.31.

### Additional Source Environment Requirements

- There must be an operating system user (**delphix_os**) with these privileges:
  - Ability to login to the source environment via SSH
  - There must be a directory on the source environment where you can install the Delphix Engine Toolkit – for example, `/var/opt/delphix/toolkit`.
    - The **delphix_os** user must own the directory
    - The directory must have permissions `-rwxrwx---` (0770), but you can also use more permissive settings
    - The **delphix_os** user must have read and execute permissions on each directory in the path leading to the toolkit directory. For example, when the toolkit is stored in `/var/opt/delphix/toolkit`, the permissions on `/var`, `/var/opt`, and `/var/opt/delphix` should allow read and execute for “others,” such as `-rwxr-xr-x`.
    - The directory should have 1.5GB of available storage: 400MB for the toolkit and 400MB for the set of logs generated by each client that runs out of the toolkit.
• On a Solaris host, `gtar` must be installed. Delphix uses `gtar` to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The `gtar` binary should be installed in one of the following directories:

  • `/bin:/usr`
  • `/bin:/sbin:/usr`
  • `/sbin:/usr/contrib`
  • `/bin:/usr/sf`
  • `/bin:/opt/sfw`
  • `/bin:/opt/csw/bin`

• The Delphix Engine must be able to initiate an SSH connection to the source environment

**Additional Target Environment Requirements**

• There must be an operating system user (**delphix_os**) with these privileges:

  • Ability to login to the target environment via SSH

  • The following permissions are usually granted via `sudo` authorization of the commands. See **Sudo Privilege Requirements** for further explanation of the commands and for examples of the `/etc/sudoers` file on different operating systems.

    • Permission to run `mount`, `umount`, `mkdir`, and `rmdir` as a super-user

    • If the target host is an AIX system, permission to run the `nfso` command as a super-user

• There must be a directory on the source environment where you can install the Delphix Engine Toolkit – for example, `/var/opt/delphix/toolkit`.

  • The **delphix_os** user must own the directory

  • The directory must have permissions `-rwxrwx---` (0770), but you can also use more permissive settings

  • The **delphix_os** user must have read and execute permissions on each directory in the path leading to the toolkit directory. For example, when the toolkit is stored in `/var/opt/delphix/toolkit`, the permissions on `/var`, `/var/opt`, and `/var/opt/delphix` should allow read and execute for “others,” such as `-rwxr-xr-x`.

  • The directory should have a total of at least 800MB of storage, plus 1MB of storage per vFile that will be provisioned to the target

• On a Solaris host, `gtar` must be installed. Delphix uses `gtar` to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The `gtar` binary should be installed in one of the following directories:

  • `/bin:/usr`
  • `/bin:/sbin:/usr`
  • `/sbin:/usr/contrib`
  • `/bin:/usr/sf`
  • `/bin:/opt/sfw`
  • `/bin:/opt/csw/bin`
• There must be an empty directory (/delphix) that will be used as a container for the mount points that are created when provisioning a vFile to the target environment. The group associated with the directory must be the primary group of the delphix_os user. Group permissions for the directory should allow read, write, and execute by members of the group.

• The Delphix Engine must be able to initiate an SSH connection to the target environment

• NFS client services must be running on the target environment
Network and Connectivity Requirements for Unix Environments

This topic outlines the network and connectivity requirements for the Delphix Engine and Unix source and target environments.

- **Port Allocations Specific to Unstructured Files**
  - Inbound to the Delphix Engine Port Allocation
  - Outbound from a Source Environment Port Allocation
  - Inbound to a Source Environment Port Allocation
  - Outbound from a Target Environment Port Allocation
  - Inbound to a Target Environment Port Allocation

- **General Port Allocation**
  - General Outbound from the Delphix Engine Port Allocation
  - General Inbound to the Delphix Engine Port Allocation

- **Firewalls and Intrusion Detection Systems (IDS)**

- **SSHD Configuration**

Port Allocations Specific to Unstructured Files

The Delphix Engine makes use of the following network ports for unstructured files dSources and vFiles:

**Inbound to the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>873</td>
<td>Rsync connections used for communication to rsyncd during SnapSync</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>111</td>
<td>Remote Procedure Call (RPC) port mapper used for NFS mounts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: RPC calls in NFS are used to establish additional ports, in the high range 32768-65535, for supporting services. Some firewalls interpret RPC traffic and open these ports automatically. Some do not.</td>
</tr>
<tr>
<td>TCP</td>
<td>1110</td>
<td>NFS Server daemon status and NFS server daemon keep-alive (client info)</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>2049</td>
<td>NFS Server daemon from vFiles to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>4045</td>
<td>NFS lock daemon/manager</td>
</tr>
<tr>
<td>UDP</td>
<td>33434 - 33464</td>
<td>Traceroute from source and target database servers to the Delphix Engine (optional)</td>
</tr>
</tbody>
</table>
UDP/TCP | 32768 - 65535 | NFS mountd and status services, which run on a random high port. Necessary when a firewall does not dynamically open ports.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>873</td>
<td>Rsync connections used during SnapSync</td>
</tr>
<tr>
<td>TCP</td>
<td>xxxx</td>
<td>DSP connections used for monitoring and script management during SnapSync. Typically DSP runs on port 8415.</td>
</tr>
</tbody>
</table>

**Outbound from a Source Environment Port Allocation**

**Inbound to a Source Environment Port Allocation**

**Outbound from a Target Environment Port Allocation**

**Inbound to a Target Environment Port Allocation**

**General Port Allocation**

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

**General Outbound from the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Port Number</td>
<td>Use</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>TCP</td>
<td>25</td>
<td>Connection to a local SMTP server for sending email</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>53</td>
<td>Connections to local DNS servers</td>
</tr>
<tr>
<td>UDP</td>
<td>123</td>
<td>Connection to an NTP server</td>
</tr>
<tr>
<td>UDP</td>
<td>162</td>
<td>Sending SNMP TRAP messages to an SNMP Manager</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
<td>SSL connections from the Delphix Engine to the Delphix Support upload server</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>636</td>
<td>Secure connections to an LDAP server</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Connections to a Delphix replication target (see Configuring Replication)</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections to source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>

**General Inbound to the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>80</td>
<td>HTTP connections to the Delphix GUI</td>
</tr>
<tr>
<td>UDP</td>
<td>161</td>
<td>Messages from an SNMP Manager to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>443</td>
<td>HTTPS connections to the Delphix GUI</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.</td>
</tr>
</tbody>
</table>
TCP | 50001 | Connections from source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)

TCP/UDP | 32768 - 65535 | Required for NFS mountd and status services from target environment only if the firewall between Delphix and the target environment does not dynamically open ports.

**Note:** If no firewall exists between Delphix and the target environment, or the target environment dynamically opens ports, this port range is not explicitly required.

### Firewalls and Intrusion Detection Systems (IDS)

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

### SSHD Configuration

Both source and target Unix environments are required to have sshd running and configured such that the Delphix Engine can connect over ssh.

The Delphix Engine expects to maintain long-running, highly performant ssh connections with remote Unix environments. The following sshd configuration entries can interfere with these ssh connections and are thus disallowed:

<table>
<thead>
<tr>
<th>Disallowed sshd Configuration Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientAliveInterval</td>
</tr>
<tr>
<td>ClientAliveCountMax</td>
</tr>
</tbody>
</table>
Sudo Privilege Requirements and File Configurations

This topic describes sudo file privilege configurations necessary for interacting with the Delphix Engine.

Sudo Privilege Rationale

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Sources</th>
<th>Targets</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>mkdir/rmdir</td>
<td>Not Required</td>
<td>Optional</td>
<td>Delphix dynamically makes and removes directories under the provisioning directory during vFiles operations. This privilege is optional, provided the provisioning directory permissions allow the delphix_os user to make and remove directories.</td>
</tr>
<tr>
<td>mount/umount</td>
<td>Not Required</td>
<td>Required</td>
<td>Delphix dynamically mounts and unmounts directories under the provisioning directory during vFiles operations. This privilege is required because mount and umount are typically reserved for a super-user.</td>
</tr>
<tr>
<td>nfsO (AIX only)</td>
<td>Not Required</td>
<td>Required</td>
<td>Delphix monitors NFS read and write sizes on an AIX target host. It uses the nfs command to query the sizes in order to optimize NFS performance for vFiles running on the target host. Only a super-user can issue the nfs command.</td>
</tr>
</tbody>
</table>

Configuring sudo Access on Solaris SPARC Target Environments

On a Solaris target, sudo access to mount, umount, mkdir, and rmdir is required. In this customer example, super-user privilege is restricted to the virtual dataset mount directory /delphix.
Example: Solaris /etc/sudoers entries for a Delphix Target

User_Alias DELPHIX_USER=delphix_os

Cmnd_Alias DELPHIX_CMDS= \
/usr/sbin/mount /delphix/*, \
/usr/sbin/mount 100.245.235.12:* /delphix/*, \
/usr/sbin/mount -o* 100.245.235.12:* /delphix/*, \
/usr/sbin/umount /delphix/*, \
/usr/sbin/umount -f /delphix/*, \
/usr/bin/mkdir /delphix/*, \
/usr/bin/mkdir -p /delphix/*, \
/usr/bin/rmdir /delphix/*

DELPHIX_USER ALL=(ALL) NOPASSWD: DELPHIX_CMDS

Configuring sudo Access on Linux for Target Environments

On a Linux target, sudo access to mount, umount, mkdir, and rmdir is required. In this customer example, super-user privilege is restricted to the virtual database mount directory /delphix.

Example: Linux /etc/sudoers file for a Delphix Target

Defaults:delphix_os !requiretty

Cmnd_Alias DELPHIX_ADMIN_CMDS= \
/bin/mount /delphix/*, \
/bin/mount * /delphix/*, \
/bin/umount /delphix/*, \
/bin/umount * /delphix/*, \
/bin/mkdir -p -m 755 /delphix/*, \
/bin/mkdir -p /delphix/*, \
/bin/mkdir /delphix/*, \
/bin/rmdir /delphix/*

Host_Alias DELPHIX_HOSTS=delphix001, delphix002
delphix_os DELPHIX_HOSTS=NOPASSWD:DELPHIX_ADMIN_CMDS

Configuring sudo Access on AIX for Target Environments

In addition to sudo access to the mount, umount, mkdir, and rmdir commands on AIX target hosts, Delphix also requires sudo access to nfso. This is required on target hosts for Delphix to monitor the NFS read / write sizes configured on the AIX system. Super-user access level is needed to run the nfso command.

Example: AIX /etc/sudoers File for a Delphix Target

Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/bin/mount, \
/bin/umount, \
/bin/mkdir, \
/bin/rmdir, \
/usr/sbin/nfso
Configuring `sudo` Access on HP-UX for Target Environments

On the HP-UX target, as with other operating systems, `sudo` access to `mount`, `umount`, `mkdir`, and `rmdir` is required.

### Example: HP-UX /etc/sudoers file for a Delphix Target

```plaintext
[Defaults:delphix_os] !requiretty
delphix_os ALL=NOPASSWD:/sbin/mount, /sbin/umount, /bin/mkdir, /bin/rmdir
```

### Considerations for `sudo` access and account locking

The Delphix Engine tests its ability to run the `mount` command using `sudo` on the target environment by issuing the `sudo mount` command with no arguments. Many of the examples shown in this topic do not allow that, and in those cases the attempt will be blocked. In most situations, this does not cause a problem.

However, some users configure the security on the target environments to monitor `sudo` failures and lock out the offending account after some threshold. In those situations, the `delphix_os` account can become locked. One work-around for this situation is to increase the threshold for locking out the user account. Another option is to modify `/etc/sudoers` to permit the `delphix_os` user to run `mkdir`, `rmdir`, and `mount` commands without parameters.
Managing Unix Environments

These topics describe special tasks and concepts for working with Unix environments containing unstructured files.

- Adding a Unix Environment
- Managing Unix Environment Users
Adding a Unix Environment

This topic describes how to add a new Unix environment.

**Prerequisites**
- See the topic [Requirements for Unix Environments](#).

**Procedure**
1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Environments.
4. Click the Plus icon next to Environments.
5. In the Add Environment dialog, select Unix/Linux.
6. Select Standalone Host.
7. Enter the Host IP address.
8. Enter an optional Name for the environment.
9. Enter the SSH port. The default value is **22**.
10. Enter a Username for the environment.
11. Select a Login Type.
   - For **Password**, enter the password associated with the user in Step 10.

<table>
<thead>
<tr>
<th>Using Public Key Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you want to use public key encryption for logging into your environment:</td>
</tr>
<tr>
<td>a. Select <strong>Public Key</strong> for the <strong>Login Type</strong>.</td>
</tr>
<tr>
<td>b. Click <strong>View Public Key</strong>.</td>
</tr>
<tr>
<td>c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.</td>
</tr>
<tr>
<td>i. Run <code>chmod 600 authorized_keys</code> to enable read and write privileges for your user.</td>
</tr>
<tr>
<td>ii. Run <code>chmod 755 ~</code> to make your home directory writable only by your user.</td>
</tr>
</tbody>
</table>

   The public key needs to be added only once per user and per environment.

   You can also add public key authentication to an environment user’s profile by using the command line interface, as explained in the topic [CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users](#).

12. For **Password Login**, click **Verify Credentials** to test the username and password.
13. Enter a **Toolkit Path**.
   - The toolkit directory stores scripts used for Delphix Engine operations. It should have a persistent working
directory rather than a temporary one.
14. Click **OK**.

**Post-Requisites**

After you create the environment, you can view information about it by doing the following:

1. Click **Manage**.
2. Select **Environments**.
3. Select the **environment name**.

**Related Links**
- [Requirements for Unix Environments](#)
Managing Unix Environment Users

This topic describes how to manage the users associated with an environment. For information on providing Delphix users with privileges for groups and database objects, see the topics under Managing Delphix Users and Managing Policies.

- Prerequisites
- Procedure

Prerequisites

Users that you add to an environment must meet the requirements for that environment as described in the platform-specific Requirements topics.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Environments.
4. Click the name of an environment to open the environment information screen.
5. Under Basic Information, click the green Plus icon to add a user.
6. Enter the Username and Password for the OS user in that environment.

Using Public Key Encryption

If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      ii. Run chmod 755 ~ to make your home directory writable only by your user.

The public key needs to be added only once per user and per environment.

You can also add public key authentication to an environment user's profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

7. Click the Check icon to save the new user.
8. To change the primary user for this environment, click the Pencil icon next to Environment Users.
9. To delete a user, click the Trash icon next to their username.
Unstructured Files on Windows Environments

These topics describe requirements for Windows source and target environments hosting unstructured files.

- **Requirements for Windows Environments**
- **Network and Connectivity Requirements for Windows Environments**
- **Managing Windows Environments**
  - **Adding a Windows Environment**
  - **Managing Windows Environment Users**
Requirements for Windows Environments

This topic outlines the supported operating systems (OSs) for use on Windows source and target environments.

**Supported Operating Systems**

- Windows Server 2003 SP2, 2003 R2
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012, 2012 R2

*Delphix supports only 64-bit versions of Windows on target hosts.*

Target hosts running Windows Server 2003 SP2 or 2003 R2 must install the hotfix documented in [KB 943043](#).

**Additional Source Environment Requirements**

- The Delphix Connector must be installed on the source environment. The Delphix Connector must have been used to register this environment with the Delphix Engine.

- The `robocopy` utility must be installed on the source Windows environment. `robocopy` is installed by default on Windows Server 2008, Windows Vista, Windows 7, and Windows 8. For other versions of Windows, it is available by downloading a resource kit from Microsoft.

**Additional Target Environment Requirements**

- The Delphix Connector must be installed on the target environment. The Delphix Connector must have been used to register this environment with the Delphix Engine.
Network and Connectivity Requirements for Windows Environments

This topic outlines the network and connectivity requirements for the Delphix Engine and Windows source and target environments.

- **Port Allocations Specific to Unstructured Files**
  - Outbound from the Delphix Engine Port Allocation
  - Inbound to the Delphix Engine Port Allocation
  - Outbound from a Source Environment Port Allocation
  - Inbound to a Source Environment Port Allocation
  - Outbound from a Target Environment Port Allocation
  - Inbound to a Target Environment Port Allocation
- **General Port Allocation**
  - General Outbound from the Delphix Engine Port Allocation
  - General Inbound to the Delphix Engine Port Allocation
- **Firewalls and Intrusion Detection Systems (IDS)**
- **SSHD Configuration**

Port Allocations Specific to Unstructured Files

The Delphix Engine makes use of the following network ports for unstructured files dSources and VDBs:

**Outbound from the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>xxxx</td>
<td>Delphix Connector connections to source and target environments. Typically the Delphix Connector runs on port 9100.</td>
</tr>
</tbody>
</table>

**Inbound to the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Number</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>3260</td>
<td>iSCSI target daemon for connections from iSCSI initiators on the target environments to the Delphix Engine</td>
</tr>
</tbody>
</table>

**Outbound from a Source Environment Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>80</td>
<td>The Delphix Connector registers source environments over HTTP</td>
</tr>
</tbody>
</table>

Firewalls and Intrusion Detection Systems (IDS)

Firewalls and Intrusion Detection Systems (IDS) need to have the following ports open:

- TCP 3260
- TCP 80

SSHD Configuration

- TCP 3260
- TCP 22
- TCP 9100
TCP | xxxx | DSP connections used for monitoring and script management during SnapSync. Typically DSP runs on port 8415.

**Inbound to a Source Environment Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>xxxx</td>
<td>Delphix Connector connections to source environments. Typically the Delphix Connector runs on port 9100.</td>
</tr>
</tbody>
</table>

**Outbound from a Target Environment Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>80</td>
<td>The Delphix Connector registers target environments over HTTP</td>
</tr>
<tr>
<td>TCP</td>
<td>xxxx</td>
<td>DSP connections used for monitoring and script management. Typically DSP runs on port 8415.</td>
</tr>
</tbody>
</table>

**Inbound to a Target Environment Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>xxxx</td>
<td>Delphix Connector connections to source environments. Typically the Delphix Connector runs on port 9100.</td>
</tr>
</tbody>
</table>

**General Port Allocation**

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

**General Outbound from the Delphix Engine Port Allocation**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Numbers</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>25</td>
<td>Connection to a local SMTP server for sending email</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>53</td>
<td>Connections to local DNS servers</td>
</tr>
<tr>
<td>UDP</td>
<td>123</td>
<td>Connection to an NTP server</td>
</tr>
<tr>
<td>Protocol</td>
<td>Port Number</td>
<td>Use</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>-----</td>
</tr>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH connections to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>80</td>
<td>HTTP connections to the Delphix GUI</td>
</tr>
<tr>
<td>UDP</td>
<td>161</td>
<td>Messages from an SNMP Manager to the Delphix Engine</td>
</tr>
<tr>
<td>TCP</td>
<td>443</td>
<td>HTTPS connections to the Delphix GUI</td>
</tr>
<tr>
<td>TCP</td>
<td>8415</td>
<td>Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.</td>
</tr>
<tr>
<td>TCP</td>
<td>50001</td>
<td>Connections from source and target environments for network performance tests via the Delphix CLI (see Network Performance Tool)</td>
</tr>
</tbody>
</table>

*General Inbound to the Delphix Engine Port Allocation*
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/UDP</td>
<td>32768 - 65535</td>
<td>Required for NFS mountd and status services from target environment only if the firewall between Delphix and the target environment does not dynamically open ports.</td>
</tr>
</tbody>
</table>

**Note:** If no firewall exists between Delphix and the target environment, or the target environment dynamically opens ports, this port range is not explicitly required.

---

**Firewalls and Intrusion Detection Systems (IDS)**

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

**SSHD Configuration**

Both source and target Unix environments are required to have sshd running and configured such that the Delphix Engine can connect over ssh.

The Delphix Engine expects to maintain long-running, highly performant ssh connections with remote Unix environments. The following sshd configuration entries can interfere with these ssh connections and are thus disallowed:

<table>
<thead>
<tr>
<th>Disallowed sshd Configuration Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientAliveInterval</td>
</tr>
<tr>
<td>ClientAliveCountMax</td>
</tr>
</tbody>
</table>
Managing Windows Environments

These topics describe special tasks and concepts for working with Windows environments containing unstructured files.

- Adding a Windows Environment
- Managing Windows Environment Users
Adding a Windows Environment

This topic describes how to add a Windows environment to the Delphix Engine for use with unstructured files.

All Windows source and target environments containing unstructured files must have the Delphix Connector installed to enable communication between the environment and the Delphix Engine. The instructions in this topic cover initiating the Add Target process in the Delphix Engine interface, running the Delphix Connector installer on the environment, and verifying that the environment has been added in the Delphix Engine interface.

Prerequisites

- Make sure your source and target environment meets the requirements described in **Requirements for Windows Environments**.

Procedure

1. From the machine that you want to use, login to the Delphix Admin application.
2. Click **Manage**.
3. Select **Environments**.
4. Next to **Environments**, click the green **Plus** icon.
5. In the operating system menu, select **Windows**.
6. Select **Target**.
7. Select **Standalone**.
8. Click the download link for the **Delphix Connector Installer**. The Delphix Connector will download to your local machine.
9. On the Windows machine that you want to use, run the Delphix Connector installer. Click **Next** to advance through each of the installation wizard screens.

   - The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.

   a. For **Connector Configuration**, make sure there is no firewall in your environment blocking traffic to the port on the target environment to which the Delphix Connector service will listen.

   b. For **Select Installation Folder**, either accept the default folder or click **Browse** to select another.

   c. Click **Close** to complete the installation process.

   d. Run this batch script as Administrator: `<Delphix Connector installation folder>\Delphix\DelphixConnector\connector\addhostgui.cmd`.

      When the **Add Windows Target Environment Wizard** launches, enter the **Target Host IP Address**, **Delphix Engine IP Address**, your **login credentials**, and the **environment user** on the Windows host.

   e. After providing this information, click **Submit**.

   f. Click **Yes** to confirm the target environment addition request.

10. In the Delphix Engine interface, you will see a new icon for the environment and two jobs running in the **Delphix Admin Job History**, one to **Create and Discover** an environment, and another to **Create** an environment. When the jobs are complete, click the icon for the new environment, and you will see the details for the environment.
**Post-Requisites**

1. On the Windows environment, in the **Windows Start Menu**, select **Services**.
2. Select **Extended Services**.
3. Make sure that the **Delphix Connector** service has a Status of **Started**.
4. Make sure that the **Startup Type** is **Automatic**.

**Related Links**

- [Requirements for Windows Environments](#)
Managing Windows Environment Users

This topic describes how to manage the users associated with an environment. For information on providing Delphix users with privileges for groups and database objects, see the topics under Managing Delphix Users and Managing Policies.

- Prerequisites
- Procedure

Prerequisites

Users that you add to an environment must meet the requirements for that environment as described in the platform-specific Requirements topics.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Environments.
4. Click the name of an environment to open the environment information screen.
5. Under Basic Information, click the green Plus icon to add a user.
6. Enter the Username and Password for the OS user in that environment.

Using Public Key Encryption

If you want to use public key encryption for logging into your environment:
   a. Select Public Key for the Login Type.
   b. Click View Public Key.
   c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.
      i. Run chmod 600 authorized_keys to enable read and write privileges for your user.
      ii. Run chmod 755 ~ to make your home directory writable only by your user.

The public key needs to be added only once per user and per environment.

You can also add public key authentication to an environment user's profile by using the command line interface, as explained in the topic CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users.

7. Click the Check icon to save the new user.
8. To change the primary user for this environment, click the Pencil icon next to Environment Users.
9. To delete a user, click the Trash icon next to their username.
Linking Unstructured Files

This topic describes the process of linking to a set of unstructured files and creating a dSource.

Prerequisites

- The source environment must meet the requirements outlined in Unstructured Files Environment Requirements.
- The Delphix Engine must have access to an environment user. This user should have read permissions on all files to be cloned.

Unstructured Files on Cluster Environments

Unstructured files cannot be linked from, or provisioned to, any form of cluster environment, such as an Oracle RAC environment. To link or provision unstructured files from a host that is part of a cluster, add the host as a standalone environment. Then, link from or provision to this standalone host.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Select Manage.
3. Select Environments.
4. Select the environment containing the unstructured files you want to link.
   If you have not already added the environment, see the Managing Unix Environments and Managing Windows Environments topics for more information about adding environments.
5. Click the Environment Details tab.
6. If the environment user described in the Prerequisites section is not already added to the Delphix Engine, add the user.
   See the Managing Unix Environments and Managing Windows Environments topics for more information about adding environment users.
7. Click the Databases tab.
8. Click the Plus icon next to Add Dataset Home.
   Adding the files as a dataset home will register the type and location of the files with the Delphix Engine.
9. Select Unstructured Files as the Dataset Home Type.
10. Enter a Name to help identify the files.
11. Enter the Path to the root directory of the files. On Windows, this may be a local path or an UNC name.
12. Click the Check icon to save your dataset home. Scroll down the list of dataset homes to view and edit this dataset home if necessary.
13. Click Manage.
15. Select **Add dSource**. Alternatively, on the **Environment Management** screen, you can click **Link** next to a dataset name to start the dSource creation process.

16. In the **Add dSource** wizard, select the **files source**.

17. Select the **Environment User** outlined in the Prerequisites section.

18. Click **Advanced**.

19. Enter **Paths to Exclude**. These paths are relative to the root path of the dataset home path and will not be linked by the Delphix Engine. This feature is most commonly used to exclude directories containing log files. Wildcard (*) pattern matching is supported to exclude all the contents of a directory, without excluding the directory itself. For example, specifying `/dir/*` will exclude all contents of `/dir` but still link `/dir` as an empty directory.

   **Retroactive Edits to Exclude Paths on Windows**

   After creating a dSource, you can edit the set of **Paths to Exclude** from syncing at any time on the back of the **dSource card**. For Unix environments, retroactively adding a path to exclude will result in the next SnapSync deleting the newly-excluded files. However, for Windows environments, retroactively adding a path to exclude will result in the next SnapSync ignoring newly-excluded files. Stale versions of these files will still exist in all future snapshots.

20. If you are linking files from a Unix environment, enter **Paths of Symlinks to Follow**. These paths are relative to the root path of the dataset home path and will be followed to gather additional files to copy.

   **Paths of Symlinks to Follow - Caveats**

   - This feature can only be used to follow symlinks to directories. Symlinks to files will be ignored.
   - This feature is not available for files on Windows environments

21. Click **Next**.

22. Enter a **dSource Name**.

23. Select a **Database Group** for the dSource.

24. Click **Next**. Adding a dSource to a database group enables you to set Delphix Domain user permissions for that dSource's objects, such as snapshots. For more information, see the topics under **Users, Permissions, and Policies**.

25. Select a **SnapSync** policy.

26. Click **Advanced** to edit Retention policies.
27. Click Next.

28. Enter any operations that should be run at **Hooks** during the sync process (or any future sync processes). For more information, see [Customizing vFiles Management with Hook Operations](#).

29. Click Next.

30. Review the **dSource Configuration** and **Data Management** information.

31. Click Finish.

The Delphix Engine will initiate two jobs to create the dSource, **DB_Link** and **DB_Sync**. You can monitor these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System > Event Viewer**. When the jobs have completed successfully, the files icon will change to a dSource icon on the **Environments > Databases** screen, and the dSource will be added to the list of **My Databases** under its assigned group.

**The dSource Card**

After you have created a dSource, you can view information about it and make modifications to its policies and permissions on the **dSource card**. In the **Databases** panel, click the **Open** icon to view the front of the **dSource card**. The card will then flip, showing you information such as the **Source** and **Data Management** configuration.

**Related Links**

- [Unstructured Files - Getting Started](#)
- [Provisioning Unstructured Files as vFiles](#)
- [Customizing vFiles with Hook Operations](#)
Provisioning Unstructured Files as vFiles

This topic describes the process of provisioning to a set of unstructured files as vFiles.

Prerequisites

- You will need an unstructured files dSource, as described in Linking Unstructured Files, or an existing vFiles from which you want to provision another.
- The target environment must meet the requirements outlined in Unstructured Files Environment Requirements.

Unstructured Files on Cluster Environments

Unstructured files cannot be linked from, or provisioned to, any form of cluster environment, such as an Oracle RAC environment. To link or provision unstructured files from a host that is part of a cluster, add the host as a standalone environment. Then, link from or provision to this standalone host.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select a dSource or vFiles.
6. Select a snapshot.
7. Click Provision.
   The Provision vFiles panel will open, and the field Mount Path will auto-populate with the path to the files on the source environment.
8. Select a target environment.
   If you need to add a new target environment for the vFiles, click the green Plus icon next to Filter Target to add the environment.

Environment Compatibility

You can only target a Unix environment when provisioning from a Unix dSources or vFiles. You can only target a Windows environment when provisioning from a Windows dSources and vFiles.

9. If necessary, modify the Mount Path.
10. Click Advanced.
11. Enter Additional Mount Points.
    When it is mounted to the target environment, the vFiles will be mounted to any additional mount points you provide.
12. Select an Environment User to own the mounted files.
   If the files are being mounted to multiple environments, ensure this user is available across all environments.

13. Click Next.

14. Enter a vFiles Name.

15. Select a Target Group for the vFiles.
   If necessary, click the green Plus icon to add a new group.

16. Select a Snapshot Policy for the vFiles.
   If necessary, click the green Plus icon to create a new policy.

17. Click Next.

18. Enter any operations that should be run at Hooks during the lifetime of the vFiles.
   See Customizing Oracle VDB Configuration Settings for more information.

19. Click Next.

20. Click Finish.
   When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the vFiles will be included in the group you designated and listed in the Databases panel. If you select the vFiles in the Databases panel and click the Open icon, you can view its card, which contains information about the vFiles and its Data Management settings.

Related Links

- Linking Unstructured Files
- Managing Data Operations for vFiles
- Creating Empty vFiles from the Delphix Engine
Customizing vFiles with Hook Operations

Hook operations allow you to execute an ordered list of custom operations as part of the dSource syncing, provisioning, and refresh process. For details on the types of operations that are available, see children of this page.

### dSource Hooks

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Sync</td>
<td>Operations performed before a sync. These operations can quiesce data to capture more information into the sync or stop applications that may interfere with the sync.</td>
</tr>
<tr>
<td>Post-Sync</td>
<td>Operations performed after a sync. These operations can undo any quiescing of data or undo any changes made by the Pre-Sync hook.</td>
</tr>
</tbody>
</table>

### Virtual Dataset Hooks

<table>
<thead>
<tr>
<th>Hook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Clone</td>
<td>Operations performed after initial provision or refresh of a virtual dataset. This hook runs when the virtual dataset has been started. During a refresh, the Delphix Engine reaches this hook before the Post-Refresh hook.</td>
</tr>
<tr>
<td>Pre-Refresh</td>
<td>Operations performed before a refresh. These operations can back up any data or configurations from the running source before doing the refresh.</td>
</tr>
<tr>
<td>Post-Refresh</td>
<td>Operations performed after a refresh. You can use these operations to restore any data or configurations backed up by the Pre-Refresh hook. During a refresh, the Delphix Engine runs this after the Configure Clone hook.</td>
</tr>
</tbody>
</table>

⚠️ Operation Failure

If a hook operation fails, it will fail the entire sync, provision, or refresh job. The job will halt immediately and no further hook operations will be run.

⚠️ Hook Operations on Cluster Environments

When linking from, or provisioning to, cluster environments such Oracle RAC environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine picks a node in the cluster at random and guarantees all operation within any single hook will execute serially on this node. Note that the Delphix Engine does not guarantee the same node is chosen for the execution of every hook.

Setting Hook Operations

You can construct hook operation lists through the Delphix Admin application or the command line interface (CLI).
You can either define the operation lists as part of the linking or provisioning process or edit them on dSources or virtual datasets that already exist.

**Setting Hook Operations through the Delphix Admin Application**

To specify hook operations during linking or provisioning, navigate to the **Hooks** tab of the **Linking Wizard** or **Provision Wizard**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Select the **type of operation** or click **Import** to load a hook operation template.
4. Click the **text area** and edit the contents of the operation.
5. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
6. To remove an operation from the list, click the **Trash** icon on the operation.
7. When you have set all hook operations, click **Next** to continue with the provisioning process.

To edit hook operations on an already-existing dSource or virtual dataset, navigate to the **Hooks** tab on the back of the **dSource card** or **virtual dataset card**.

1. Select the **hook** to edit.
2. The current operations at this hook will be displayed. To edit this list of operations, click the **Pencil** icon in the top right-hand corner of the card.
3. Click the **Plus** icon to add a new operation.
4. Select the **type of operation** or click **Import** to load a hook operation template.
5. Click the **text area** and edit the contents of the operation.
6. You can reorder operations either through drag-and-drop or by clicking the **arrow** icons.
7. To remove an operation from the list, click the **Trash** icon on the operation.
8. When you have set all hook operations, click **Check** to save the changes.

**Setting Hook Operations through the CLI**

To specify hook operations during linking, edit the relevant hook's array of operations defined on the **LinkingParameters > Source > Operations** object.

To specify hook operations during provisioning, edit the relevant hook's array of operations defined on the **ProvisionParameters > Source > Operations** object.

To edit hook operations on an already-created dSource, edit the relevant hook's array of operations defined on the **Source > Operations** object.

To edit hook operations on an already-created virtual dataset, edit the relevant hook's array of operations defined on the **Source > Operations** object.

For more information about these CLI objects, see the **LinkedSourceOperations**, **VirtualSourceOperations**, **RunCommandOnSourceOperation**, and **RunExpectOnSourceOperation** API documentation in the Help menu of the Delphix Admin application.

**Example of Editing Hook Operations through the CLI**

1. Navigate to relevant source's **VirtualSourceOperations** object.
2. Select a **hook** to edit.

```bash
delphix> source
delphix source> select "pomme"
delphix source "pomme"> update
delphix source "pomme" update *> edit operations
delphix source "pomme" update operations *> edit postRefresh
```

3. Add an operation at index 0.

```bash
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 0 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 0 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 0 *> ls
Properties
    type: RunCommandOperation (*)
    command: echo Refresh completed. (*)
delphix source "pomme" update operations postRefresh 0 *> commit
```

4. Add another operation at index 1 and then delete it.

```bash
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 1 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 1 *> set command="echo Refresh completed."
delphix source "pomme" update operations postRefresh 1 *> back
delphix source "pomme" update operations postRefresh *> unset 1
delphix source "pomme" update operations postRefresh *> commit
```

**Hook Operation Templates**

You can use templates to store commonly-used operations, which allows you to avoid repeated work when an operation is applicable to more than a single dSource or virtual dataset. You manage templates through the Delphix Admin application.

**Creating a Hook Operation Template**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Operation Templates.
4. Click the Plus icon to add a new operation template.
5. Enter a Name for the template.
6. Select an operation Type.
7. Enter a Description detailing what the operation does or how to use it.
8. Enter operation Contents to implement the operation partially or fully.
9. Click Create.

**Importing a Hook Operation Template**
To import a hook operation template, navigate to the **Hooks** tab on the back of the **dSource** or **virtual dataset card**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Click **Import**.
4. Select the **template** to import.
5. Click **Import**.
6. When you have set all hook operations, click **Check** to save the changes.

**Exporting a Hook Operation Template**

To export a hook operation template, navigate to the **Hooks** tab on the back of the **dSource** or **virtual dataset card**.

1. Select the **hook** to edit.
2. Click the **Plus** icon to add a new operation.
3. Select the **type of operation**.
4. Click the **text area** and edit the contents of the operation.
5. Click **Export**.
6. Enter a **Name** for the template.
7. Enter a **Description** detailing what the operation does or how to use it.
8. Click **Create**.

**Related Links**
Working with vFiles Hook Operation Types

- **RunCommand Operation**
  - Examples of RunCommand Operations
- **RunExpect Operation**
  - Example of a RunExpect Operation
- **RunPowershell Operation**
  - Example of a RunPowershell Operation

Unstructured Files Environment Variables
- **dSource Environment Variables**
- **vFiles Environment Variables**

**Related Links**

**RunCommand Operation**

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at /bin/sh. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

**Examples of RunCommand Operations**

You can input the full command contents into the RunCommand operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
fi

exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" "third argument in single quotes"
```

**RunExpect Operation**

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as ssh. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Admin
application and CLI to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunExpect Operation

Start an ssh session while interactively providing the user's password.

```
spawn ssh user@delphix.com
expect {
    -re {Password: } {
        send "\$\{env(PASSWORD_ENVIRONMENT_VARIABLE)\}\\n"
    }
    timeout {
        puts "Timed out waiting for password prompt."
        exit 1
    }
}
exit 0
```

RunPowershell Operation

The RunPowershell operation executes a Powershell script on a Windows environment. The Delphix Engine captures and logs all output of the script. If the script fails, the output is displayed in the Delphix Admin application and command line interface (CLI) to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunPowershell Operation

You can input the full command contents into the RunPowershell operation.

```
$removedir = $Env:DIRECTORY_TO_REMOVE
if ((Test-Path $removedir) -And (Get-Item $removedir) -is [System.IO.DirectoryInfo]) {
    Remove-Item -Recurse -Force $removedir
} else {
    exit 1
}
exit 0
```

Unstructured Files Environment Variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or vFiles, the Delphix Engine will always set certain environment variables so that the user-provided script can use them to access the dSource or vFiles.

dSource Environment Variables

<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLPX_DATA_DIRECTORY</td>
<td>The absolute path of the data directory synced</td>
</tr>
</tbody>
</table>

vFiles Environment Variables
<table>
<thead>
<tr>
<th>Environment Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLPX_DATA_DIRECTORY</td>
<td>The absolute path of the virtual data directory on the target host created by the Delphix Engine</td>
</tr>
</tbody>
</table>

*Related Links*
Managing Data Operations for vFiles

These topics describe how to run and manage various data operations for vFiles once they have been provisioned.

- Enabling and Disabling vFiles
- Rewinding vFiles
- Refreshing vFiles
- Deleting vFiles
- Migrating vFiles
Enabling and Disabling vFiles

This topic describes the process of enabling and disabling a vFiles.

**Prerequisites**

- You must have already provisioned a vFiles. For more information, see Provisioning Unstructured Files as vFiles.

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the vFiles to disable.
6. On the back of the card, move the slider control from Enabled to Disabled.

When you are ready to enable the vFiles again, move the slider control from Disabled to Enabled, and the vFiles will continue to function as it did previously.

![Warning]

Disabling the vFiles will unmount it from target environments. This unmount will fail if there are processes still accessing the vFiles.

**Related Links**

- Managing Data Operations for vFiles
- Provisioning Unstructured Files as vFiles
Rewinding vFiles

This topic describes how to rewind a vFiles.

Prerequisites

- You must have already provisioned a vFiles. For more information, see Provisioning Unstructured Files as vFiles.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the vFiles to rewind.
6. Select a past snapshot.
7. Click Rewind below the snapshots.

Related Links

Managing Data Operations for vFiles

Provisioning Unstructured Files as vFiles
Refreshing vFiles

This topic describes how to manually refresh a vFiles.

Refreshing a vFiles will re-provision the vFiles from its parent. As with the normal provisioning process, you can choose to refresh the vFiles from any snapshot available in its parent. However, you should be aware that refreshing a vFiles will delete any changes that have been made to it over time. When you refresh a vFiles, you are essentially resetting it to the state you select during the Refresh process. You can refresh a vFiles manually, as described in this topic, or you can set a vFiles refresh policy, as described in the topics Managing Policies: An Overview, Creating Custom Policies, and Creating Policy Templates.

Although the VDB no longer contains the previous contents, the previous Snapshots and TimeFlow still remain in the Delphix Engine and are accessible through the Command Line Interface (CLI).

Prerequisites

- You must have already provisioned a vFiles. For more information, see Provisioning Unstructured Files as vFiles.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the vFiles to refresh.
6. On the back of the vFiles card, click the Refresh icon in the lower right-hand corner.
7. Select a snapshot from which to refresh.

Related Links

Managing Data Operations for vFiles
Provisioning Unstructured Files as vFiles
Deleting vFiles

This topic describes how to delete a vFiles.

**Prerequisites**

- You must have already provisioned a vFiles. For more information, see [Provisioning Unstructured Files as vFiles](#).

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the vFiles you want to delete.
6. Click the Trash icon.
7. Click Yes to confirm.

⚠️ Deleting a vFiles may fail if it cannot be unmounted successfully from all target environments. You can use the Force Delete option to ignore all failures during unmount.

**Related Links**

- [Managing Data Operations for vFiles](#)
- [Provisioning Unstructured Files as vFiles](#)
Migrating vFiles

This topic describes how to migrate a vFiles from one target environment to another.

Prerequisites

- You must have already provisioned a vFiles. For more information, see Provisioning Unstructured Files as vFiles.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the vFiles you want to migrate.
6. Click the Open icon.
7. Move the slider control from Enabled to Disabled to unmount the files.
   For more information, see Enabling and Disabling vFiles.
8. In the bottom right-hand corner of the vFiles card, click the vFiles Migrate icon.
9. Select a new target environment.
10. Click the Check icon to confirm your selection.
11. Move the slider control to Enabled.
12. Click Yes to confirm.
   Within a few minutes, your vFiles will re-start in the new environment, and you can continue to work with it as you would any other vFiles.

Related Links

- Enabling and Disabling vFiles
- Rewinding vFiles
- Refreshing vFiles
- Deleting vFiles
Creating Empty vFiles from the Delphix Engine

This topic describes the procedure for creating empty vFiles. This feature is available for unstructured files. “Unstructured files” are a dataset that is treated as simply a directory tree full of files. It is not a database, and it does not receive any special treatment or processing by Delphix. It is just a set of files. You can create vFiles in two ways: by provisioning from an existing dataset – that is, from a dSource or from another vFiles – or by creating an empty vFiles and filling it with data.

Creating an empty vFiles places an initially-empty mount on target environments, hence the term "empty vFiles." This mount is useful when you have no existing files to copy into the Delphix Engine, but you do have files which you will generate, track, and copy with vFiles.

vFiles created without dSources are almost identical to those created by provisioning. The only thing you cannot do with them is refresh. Refreshing a dataset means overwriting the dataset’s content with new data that is pulled in from the dataset’s parent. If you create new vFiles from scratch, that newly-created dataset will not have a parent. Therefore, it cannot be refreshed. All other functionality is identical – you can provision from such a dataset, rewind, take snapshots, and so forth.

Prerequisites

- The target environment must meet the requirements outlined in Unstructured Files Environment Requirements.

Unstructured Files on Cluster Environments

You cannot create vFiles on any form of cluster environment, such as an Oracle RAC environment. To create a vFiles on a host that is part of a cluster, add the host as a standalone environment. Then, create the vFiles on this standalone host.

Procedure

To create new vFiles without provisioning:

1. Login to the Delphix Admin application.
2. Click Manage.
3. Select Database.
4. Select Create vFiles as seen below.

Related Links

Managing Data Operations for vFiles
Provisioning Unstructured Files as vFiles
Virtualizing Oracle Enterprise Business Suite
Virtualizing Oracle Business Suite: Getting Started

This topic describes basic concepts for linking and virtualizing an instance of Oracle Enterprise Business Suite (EBS) with the Delphix Engine.

Oracle Enterprise Business Suite is primarily comprised of a database service and a plethora of application services. The Delphix Engine supports the linking and provisioning of all EBS data, including the database technology stack (dbTechStack), database, and application files (appsTier). Note that the Delphix Engine can also manage custom extensions and plug-ins.

The process of linking EBS data involves creating multiple dSources:

- a dSource for the dbTechStack
- a dSource for the Oracle database used by EBS
- a dSource for the appsTier

These dSources are collectively referred to as EBS dSources. They are also the sources from which you can provision virtual EBS instances.

The process of provisioning a virtual EBS instance involves provisioning each of these dSource separately to the proper environments. You can add custom configuration logic per-EBS instance to the Delphix Engine such that the linking, provisioning, and refresh processes are as automated as possible.

**dbTechStack dSource**

The source dbTechStack is linked using the Delphix Engine’s EBS support: the linking process automatically runs pre-clone logic to ensure EBS configuration is always appropriately staged at the time of data capture. When you provision EBS, the Delphix Engine automates post-clone configuration such that a copy of the dbTechStack is available for use on the target dbTier server with no additional effort. You can add this copy of the dbTechStack to the Delphix Engine as an Oracle installation home and use it to host an EBS virtual database (VDB).

**Database dSource**

The database dSource is linked using the Delphix Engine’s support for Oracle databases. This dSource contains database data files that EBS is currently using. For more information about managing Oracle databases, see [Managing Oracle, Oracle RAC, and Oracle PDB Data Sources](#). When you provision EBS, you will use the Delphix Engine to set up a copy of the EBS database on the target dbTier server. This copy of the database will be used by a virtual EBS instance’s appsTier.

**appsTier dSource**

The appsTier is linked using the Delphix Engine’s EBS support: the linking process automatically runs pre-clone logic to ensure EBS configuration is always appropriately staged at the time of data capture. When you provision EBS, the Delphix Engine will automate post-clone configuration such that a copy of the appsTier is available for use on the target appsTier server. This virtual copy of the appsTier will connect to the provisioned EBS virtual database (VDB).
Oracle Enterprise Business Suite Environment Requirements

This section describes the environment requirements that must be satisfied before you can link, virtualize, or provision an instance of Oracle Enterprise Business Suite (EBS). These requirements include expectations for operating system and network configuration.

- Supported EBS Versions
- Source EBS Environment Requirements
- Target EBS Environment Requirements
Supported EBS Versions

The Delphix Engine offers first-class support for the following versions of Oracle Enterprise Business Suite (EBS). Note that minor releases of EBS are not certified for compatibility individually: major release support implies support for any minor release.

Please contact Delphix if you have concerns about compatibility with your specific version of EBS.

### Supported EBS Versions

<table>
<thead>
<tr>
<th>Supported EBS Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Enterprise Business Suite 11i</td>
</tr>
<tr>
<td>▪ Both 32-bit and 64-bit architectures are supported for both the dbTier and appsTier.</td>
</tr>
<tr>
<td>▪ The EBS database must be patched to version 9.2.0.8 or greater. The Delphix Engine does not support Oracle databases patched below this version.</td>
</tr>
<tr>
<td>Oracle Enterprise Business Suite R12.1</td>
</tr>
<tr>
<td>Oracle Enterprise Business Suite R12.2</td>
</tr>
</tbody>
</table>

#### Custom Context Variables Not Supported

Deployments of Oracle Enterprise Business Suite utilizing custom context variables (custom variables maintained in the EBS context file) are not supported by the workflow outlined in this documentation. Please contact Delphix if you utilize custom context variables in your EBS deployment.

#### Provisioning From a Virtual EBS Instance Not Supported

Provisioning virtual EBS instances from virtual EBS instances is not supported by the Delphix Engine. Please contact Delphix if you plan to provision a virtual EBS instance from a virtual EBS instance.

### Related Links

- [Source EBS Environment Requirements](#)
- [Target EBS Environment Requirements](#)
Source EBS Environment Requirements

This topic outlines the environment requirements for linking an Oracle Enterprise Business Suite (EBS) instance to the Delphix Engine.

**Source EBS Environment Requirements**

The Delphix Engine expects EBS source environments to meet the following requirements:

- The source database servers must meet the requirements outlined in [Requirements for Unix Environments](#) and [Oracle Support and Requirements](#).

- The source application servers must meet the requirements outlined in [Requirements for Unix Environments](#).

- The Delphix Engine must have access to an `oracle` user on the source dbTier. This user should be a member of both the EBS `dba` and `oinstall` groups. The user should have read permissions on all `dbTechStack` and database files to be cloned.

- The Delphix Engine must have access to an `applmgr` user on the source appsTier. This user should be a member of the `oinstall` group. The user should have read permissions on all appsTier files to be cloned.

**Supported Source EBS Topologies**

You can configure Oracle Enterprise Business Suite in a variety of topologies such that compute and storage resources are distributed across multiple machines. The Delphix Engine offers first-class support for linking the following topologies:

**Source dbTier Supported Topologies**

- Oracle SI dbTechStack and Database

- Oracle RAC Database

  **Oracle RAC dbTechStack Not Supported**

  The Delphix Engine does not support linking an Oracle RAC dbTechStack or cluster services. However, the Delphix Engine will support linking and provisioning the Oracle RAC database itself. If your source EBS instance uses a RAC dbTechStack and database, you will have to manage cloning the dbTechStack manually outside of Delphix; however, the database and appsTier virtualization are still supported.

**Source appsTier Supported Topologies**

- Single-node appsTier

Multi-node appsTier with a shared APPL_Top

  **Non-shared APPL_TOP Not Supported**

  The Delphix Engine does not provide first-class support for linking a multi-node appsTier where the APPL_TOP is not shared between nodes. Please contact Delphix if you have concerns about linking an appsTier with this configuration.

**Related Links**
- **Supported EBS Versions**
- **Target EBS Environment Requirements**
Target EBS Environment Requirements

This topic outlines the environment requirements for provisioning a virtual Oracle Enterprise Business Suite (EBS) instance from the Delphix Engine.

Target EBS Environment Requirements

The Delphix Engine expects EBS target environments to meet the following requirements:

- The target database servers must meet the requirements outlined in Requirements for Unix Environments and Oracle Support and Requirements.
- The target application servers must meet the requirements outlined in Requirements for Unix Environments.
- The Delphix Engine must have access to an oracle user on the target dbTier. This user should be a member of both the EBS dba and oinstall groups. This user will be given proper permissions to manage the target dbTechStack and database.
- The Delphix Engine must have access to an applmgr user on the target appsTier. This user should be a member of the oinstall group. This user will be given proper permissions to manage the target appsTier.

**oralinventory, oraTab and oraInst.loc**

Note that an oralinventory and oraTab file do not need to exist on the target dbTier or target appsTier environments prior to provisioning. These files are created during the provision process if they do not already exist.

However, you should create an oraInst.loc file on your target environments prior to provisioning. This file will specify where the oralinventory is created. The oraInst.loc file is typically located at /etc/oraInst.loc or /var/opt/oracle/oraInst.loc. Consults Oracle EBS documentation to determine where to place this file on your target environment and for more information about what this file should contain.

Supported Target EBS Topologies

You can configure Oracle Enterprise Business Suite in a variety of topologies such that compute and storage resources are distributed across multiple machines. The Delphix Engine offers first-class support for provisioning the following topologies:

Target dbTier Supported Topologies

- Oracle SI dbTechStack and Database
- Oracle RAC Database

**Oracle RAC dbTechStack Not Supported**

The Delphix Engine does not support linking an Oracle RAC dbTechStack or cluster services. Therefore, the Delphix Engine also does not support provisioning a RAC dbTechStack. You can provision a RAC database for a target EBS instance only if you have cloned dbTechStack and cluster services to the target servers manually outside of Delphix.
Target appsTier Supported Topologies

- Single-node appsTier

⚠️ Multi-node appsTier Not Supported

The Delphix Engine does not provide first-class support for provisioning a multi-node appsTier. Please contact Delphix if you have concerns about provisioning an appsTier with this configuration.

Related Links

- [Supported EBS Versions](#)
- [Source EBS Environment Requirements](#)
Linking and Provisioning Oracle Enterprise Business Suite 11i

This section describes linking and provisioning Oracle Enterprise Business Suite 11i.

- Linking EBS 11i Sources
- Provisioning EBS 11i Sources
Linking EBS 11i Sources

This topic describes the process of linking an Oracle Enterprise Business Suite (EBS) 11i instance and creating the necessary dSources.

Prerequisites

- Ensure that your EBS environment meets the criteria outlined in Oracle Enterprise Business Suite Environment Requirements

Oracle EBS 11i Database Version

The EBS database must be patched to version 9.2.0.8 or greater. The Delphix Engine does not support Oracle databases patched below this version.

Procedure

Link the Oracle Database

1. Link the Oracle database used by EBS, as outlined in Linking an Oracle Data Source.

EBS SnapSync Conflicts

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

2. Set up Before Sync hook operations to run any pre-clone steps necessary and specific to your EBS database.
   Normally, these steps will include running Oracle's adpreclone tool. Below is an example of a simple Run Shell Command hook operation:

EBS 11i Database Before Sync Script

```bash
NOTE: Ensure the below environment variables will be set up correctly by the shell. If not, hardcode or generate the values below.
ORACLE_HOME=/u01/oracle/visdb/9.2.0
CONTEXT_NAME=VIS_rh511-ebs11-db
APPS_PASSWD=apps

. ${ORACLE_HOME}/${CONTEXT_NAME}.env
perl ${ORACLE_HOME}/appsutil/scripts/${CONTEXT_NAME}/adpreclone.pl database <<-EOF
${APPS_PASSWD}
EOF
```

Link the EBS dbTechStack
1. Login to the Delphix Admin application using Delphix Admin credentials.

2. Click Manage.

3. Select Environments.

4. Select the source dbTier environment containing the source dbTechStack.
   If you have not already added the environment, see the Managing Unix Environments topics for more information about adding environments.

5. Click the Environment Details tab.

6. If the oracle environment user described in the Oracle Enterprise Business Suite Environment Requirements section is not already added to Delphix, add the user.
   For more information about adding environment users, see the Managing Unix Environment Users topics.

7. Click the Databases tab.

8. Click the Plus icon next to Add Dataset Home.
   Adding the files as a dataset home will register the type and location of the files with the Delphix Engine.

9. Select E-Business Suite 11i dbTechStack as your Dataset Home Type.
   When you select this type of dataset home, the Delphix Engine will know to automate pre-clone logic. Specifically, adpreclone.pl dbTechStack will be run prior to every SnapSync of the dbTechStack. During dSource creation, you will be able to enter additional pre-clone steps as Before Sync hook operations.

10. Enter an Installation Home.
    This path should be the directory above $ORACLE_HOME on your source environment. Commonly, this path looks like /u01/oracle/VIS.

11. Click the Check icon to save your dataset home.
    Scroll down the list of dataset homes to view and edit this dataset home if necessary.

12. Click Manage.

13. Select Databases.

14. Select Add dSource.

15. In the Add dSource wizard, select the dbTechStack files source you just created.

16. Enter the EBS-specific parameters for your dbTechStack.
    These parameter values will be used when adpreclone.pl is run.
    Ensure the Context Name uses the uppercase SID and the short hostname.

17. Click Advanced.

18. Exclude the EBS database's data files if they are stored underneath the dbTechStack root.
    This data was linked with the database instead of with the dbTechStack. Add the relative path to the data files to the Paths to Exclude list.

19. Click Next.

20. Enter a dSource Name.

21. Select a Database Group for the dSource.

22. Click Next.
    Adding a dSource to a database group enables you to set Delphix Domain user permissions for that
dSource’s objects, such as snapshots. For more information, see the topics under Users, Permissions, and Policies.

23. Select a SnapSync policy.

**EBS SnapSync Conflicts**

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

24. Click Next.

25. Enter any custom pre-clone logic as Before Sync or After Sync hook operations. Remember that adpreclone.pl dbTechStack is already run prior to every SnapSync of the dbTechStack.

The Before Sync hook operations will be run prior to running the adpreclone.pl tool.

For more information, see Customizing vFiles Management with Hook Operations.

26. Click Next.

27. Review the dSource Configuration and Data Management information, and then click Finish.

The Delphix Engine will initiate two jobs to create the dSource, DB_Link and DB_Sync. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have completed successfully, the files icon will change to a dSource icon on the Environments > Data bases screen, and the dSource will be added to the list of My Databases under its assigned group.

**Link the EBS appsTier**

1. Login to the Delphix Admin application using Delphix Admin credentials.

2. Click Manage.

3. Select Environments.

4. Select the source appsTier environment.

If you have not already added the environment, see the Managing Unix Environments topics for more information about adding environments.

5. Click the Environment Details tab.

6. If the applmgr environment user described in the Oracle Enterprise Business Suite Environment Requirements section has not already been added to Delphix, add the user.

For more information about adding environment users, see the Managing Unix Environment Users topics.

7. Click the Databases tab.

8. Click the Plus icon next to Add Dataset Home.

Adding the files as a dataset home will register the type and location of the files with the Delphix Engine.
9. Select **E-Business Suite 11i appsTier** as your **Dataset Home Type**.
   When you select this type of dataset home, the Delphix Engine will know to automate pre-clone logic.
   Specifically, adpreclone.pl appsTier will be run prior to every SnapSync of the appsTier. During
dSource creation, you will be able to enter additional pre-clone steps as Before Sync hook operations.

10. Enter an **Installation Home**.
    This path should be the directory above $APPL_TOP on your source EBS environment. Commonly, this path
    looks like /u01/oracle/VIS/.

11. Click the **Check** icon to save your dataset home.
    Scroll down the list of dataset homes to view and edit this dataset home if necessary.

12. Click **Manage**.
13. Select **Databases**.
14. Select **Add dSource**.
15. In the **Add dSource** wizard, select the **appsTier files source** you just created.

16. Enter the **EBS-specific parameters** for your appsTier.
    These parameter values will be used when adpreclone.pl is run.
    Ensure the **Context Name** uses the uppercase SID and the short hostname.

17. Click **Next**.

18. Enter a **dSource Name**.

19. Select a **Database Group** for the dSource.

20. Click **Next**.
    Adding a dSource to a database group enables you to set Delphix Domain user permissions for that
dSource's objects, such as snapshots. For more information, see the topics under **Users, Permissions, and Policies**.

21. Click **Next**.

22. Select a **SnapSync** policy.

---

**EBS SnapSync Conflicts**

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

---

23. Click **Next**.

24. Enter any **custom pre-clone logic** as Before Sync or After Sync hook operations.
    Remember that adpreclone.pl appsTier is already run prior to every SnapSync of the appsTier.
    The Before Sync hook operations will be run prior to running the adpreclone.pl tool.

25. Click **Next**.
26. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs to create the dSource, DB_Link and DB_Sync. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have completed successfully, the files icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of My Databases under its assigned group.

Related Links

Oracle Enterprise Business Suite Environment Requirements
Linking an Oracle Data Source
Provisioning EBS 11i Sources
Managing Data Operations of Virtual EBS Instances
Provisioning EBS 11i Sources

This topic describes the process of provisioning a virtual instance of Oracle Enterprise Business Suite (EBS) 11i.

**Prerequisites**

- Ensure that your EBS environment meets the criteria outlined in Oracle Enterprise Business Suite Environment Requirements.
- You must have linked a source instance of Oracle Enterprise Business Suite 11i. For more information, see Linking EBS 11i Sources.

**Snapshot Coordination**

Changes applied to EBS and picked up only in certain dSource snapshots may make certain combinations of snapshots across the appsTier and dbTier incompatible. When provisioning, refreshing or rewinding a virtual EBS instance, be sure the points in time you select for each dataset are compatible with each other.

**Procedure**

Provision the EBS dbTechStack

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the dbTechStack dSource.
6. Select a dSource snapshot.
   All snapshots will have staged configuration prepared by adpreclone.pl and any hook operations placed on the dSource.
7. Click Provision.
   The Provision vFiles wizard will open, and the field Mount Path will auto-populate with the path to the dbTechStack on the source dbTier environment. Change the value of Mount Path if you want to provision to another location on the target dbTier environment.
8. Select an Environment User.
   This user should be the oracle user outlined in Oracle Enterprise Business Suite Environment Requirements.
9. Enter the EBS-specific parameters for your target dbTier.
   These parameter values will be used to run adcfgclone.pl dbTechStack during the provision process. Ensure that the Target System Database Hostname is the short hostname, not the fully-qualified hostname.
Source APPS Password

The source APPS password is required to configure the dbTechStack. This password is encrypted when stored within Delphix and is available as an environment variable to the a dcfgclone process.

10. Click Next.

11. Enter a vFiles Name.

12. Select a Target Group for the vFiles.
   If necessary, click the green Plus icon to add a new group.

13. Select a Snapshot Policy for the vFiles.
   If necessary, click the green Plus icon to create a new policy.

14. Click Next.

15. Enter any custom hook operations that are needed to help manage the dbTechStack files correctly.
   For more information about these hooks, when they are run, and how operations are written, see Customizing vFiles with Hook Operations.
   The Configure Clone hook will be run after the dcfgclone.pl tool has both mounted and configured the dbTechStack.

16. Click Next.

17. Click Finish.
   When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the dbTechStack vFiles will be included in the group you designated and listed in the Databases panel. If you select the dbTechStack vFiles in the Databases panel and click the Open icon, you can view its card, which contains information about the virtual files and its Data Management settings.

Register the EBS dbTechStack

Register the freshly-provisioned dbTechStack with the Delphix Engine.

1. Login to the Delphix Admin application using Delphix Admin credentials.

2. Click Manage.

3. Select Environments.

4. Select the target dbTier environment.

5. Click the Databases tab.

6. Click the Plus icon next to Add Dataset Home.

7. Select Oracle as your Dataset Home Type.

8. Enter an Installation Home.
   This path should be the value of $ORACLE_HOME on your target EBS environment. Commonly, this path looks like /u01/oracle/VIS/visdb/9.2.0.
9. Click the **Check** icon to save your dataset home. Scroll down the list of dataset homes to view and edit this dataset home if necessary.

10. Refresh the dbTier environment.
    Refreshing the environment will ensure that the EBS database listener is available for selection when provisioning the EBS database.

    a. Click the **Refresh** button in the bottom right-hand corner of the environment card.

**Provision the Oracle Database**

1. Provision the EBS database to the target dbTier environment by following the steps outlined in **Provisioning an Oracle VDB**.

2. Select the correct **Installation Home**. This selection should be the virtual dbTechStack you just added to the Delphix Engine.

3. Select the correct **Oracle Node Listeners** value. This selection should be the listener corresponding to the virtual dbTechStack you just added to the Delphix Engine.

**Provision the EBS appsTier**

1. Login to the **Delphix Admin** application using **Delphix Admin** credentials.

2. Click **Manage**.

3. Select **Databases**.

4. Select **My Databases**.

5. Select the **appsTier dSource**.

6. Select a dSource **snapshot**. All snapshots will have staged configuration prepared by **adpreclone.pl** and any hook operations placed on the dSource.

7. Click **Provision**. The **Provision vFiles** wizard will open, and the field **Mount Path** will auto-populate with the path to the appsTier on the source appsTier environment. Change the value of **Mount Path** if you want to provision to another location on the target appsTier environment.

8. Select an **Environment User**. This user should be the **app1mgr** user outlined in **Oracle Enterprise Business Suite Environment Requirements**.

9. Enter the **EBS-specific parameters** for your target appsTier. These parameter values will be used to run **adcfgclone.pl** appsTier during the provision process. Ensure the **Target System Application Hostname** and **Target System Database Hostname** are the short hostnames, not the fully-qualified hostnames.
10. Click Next.

11. Enter a **vFiles Name**.

12. Select a **Target Group** for the vFiles.
   If necessary, click the green **Plus** icon to add a new group.

13. Select a **Snapshot Policy** for the vFiles.
   If necessary, click the green **Plus** icon to create a new policy.

14. Click Next.

15. Enter any **custom hook operations** that are needed to help manage the appsTier files correctly.
   For more information about these hooks, when they are run, and how operations are written, see [Customizing vFiles with Hook Operations](#).
   The Configure Clone hook will be run after the `adcfgclone.pl` tool has both mounted and configured the appsTier.

16. Click Next.

17. Click Finish.

---

**dbTier Must Be Accessible During appsTier Provisioning**

Post-clone configuration will fail if the appsTier cannot connect to the database. Ensure that the target dbTier is accessible to the appsTier during the provisioning process. Ensure that both the target database and database listener are running.

When provisioning starts, you can review progress of the job in the **Databases** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the appsTier vFiles will be included in the group you designated and listed in the **Databases** panel. If you select the appsTier vFiles in the **Databases** panel and click the **Open** icon, you can view its card, which contains information about the virtual files and its Data Management settings.

Once all three EBS virtual datasets have been provisioned successfully, your virtual EBS instance should be running and accessible.

**Related Links**

- [Oracle Enterprise Business Suite Environment Requirements](#)
- [Linking an Oracle Data Source](#)
- [Linking EBS 11i Sources](#)
Managing Data Operations of Virtual EBS Instances
Linking and Provisioning Oracle Enterprise Business Suite R12.1

This section describes linking and provisioning Oracle Enterprise Business Suite R12.1.

- [Linking EBS R12.1 Sources](#)
- [Provisioning EBS R12.1 Sources](#)
Linking EBS R12.1 Sources

This topic describes the process of linking an Oracle Enterprise Business Suite (EBS) R12.1 instance and creating the necessary dSources.

Prerequisites

- Ensure that your EBS environment meets the criteria outlined in [Oracle Enterprise Business Suite Environment Requirements](#).

Procedure

**Link the Oracle Database**

1. Link the Oracle database used by EBS as outlined in [Linking an Oracle Data Source](#).

**EBS SnapSync Conflicts**

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

2. Set up a Before Sync hook operation to run any pre-clone steps necessary and specific to your EBS database.

   Normally, these steps will include running Oracle's `adpreclone` tool. Below is an example of a simple Run Shell Command hook operation:

   ```bash
   # NOTE: Ensure the below environment variables will be set up correctly by the shell. If not, hardcode or generate the values below.
   # ORACLE_HOME=/u01/oracle/VIS/db/tech_st/11.1.0
   # CONTEXT_NAME=${ORACLE_SID}_${hostname -s}
   # APPS_PASSWD=<passwd>
   . ${ORACLE_HOME}/${CONTEXT_NAME}.env
   perl ${ORACLE_HOME}/appsutil/scripts/${CONTEXT_NAME}/adpreclone.pl
   ${APPS_PASSWD}
   EOF
   
   EOF
   ```

**Link the EBS dbTechStack**

1. Login to the Delphix Admin application using Delphix Admin credentials.

2. Click Manage.

3. Select Environments.

4. Select the source dbTier environment containing the source dbTechStack.
If you have not already added the environment, see the Managing Unix Environments topics for more information about adding environments.

5. Click the **Environment Details** tab.

6. If the oracle environment user described in the Oracle Enterprise Business Suite Environment Requirements section is not already added to Delphix, add the user.
   For more information about adding environment users, see the Managing Unix Environment Users topics.

7. Click the **Databases** tab.

8. Click the **Plus** icon next to Add Dataset Home.
   Adding the files as a dataset home will register the type and location of the files with the Delphix Engine.

9. Select **E-Business Suite R12.1 dbTechStack** as your **Dataset Home Type**.
   When you select this type of dataset home, the Delphix Engine will know to automate pre-clone logic.
   Specifically, **adpreclone.pl dbTechStack** will be run prior to every SnapSync of the dbTechStack.
   During dSource creation, you will be able to enter additional pre-clone steps as Before Sync hook operations.

10. Enter an **Installation Home**.
    This path should be the Oracle base install directory. For example, if the value of $ORACLE_HOME on your source environment is /u01/oracle/VIS/db/tech_st/11.1.0, the Oracle base install directory is /u01/oracle/VIS.

11. Click the **Check** icon to save your dataset home.

12. Scroll down the list of dataset homes to view and edit this dataset home if necessary.

13. Click **Manage**.

14. Select **Databases**.

15. Select **Add dSource**.

16. In the Add dSource wizard, select the dbTechStack files source you just created.

17. Enter the EBS-specific parameters for your dbTechStack.
   These parameter values will be used when adpreclone.pl is run.
   Ensure that the **Context Name** uses the uppercase SID and the short hostname.

18. Click **Advanced**.

19. Exclude the EBS database’s data files if they are stored underneath the dbTechStack root.
   This data was linked with the database instead of with the dbTechStack. Add the relative path to the data files to the **Paths to Exclude** list.

20. Click **Next**.

21. Enter a **dSource Name**.

22. Select a **Database Group** for the dSource.

23. Click **Next**.
   Adding a dSource to a database group enables the ability for you to set Delphix Domain user permissions for that dSource’s objects, such as snapshots. For more information, see the topics under Users, Permissions, and Policies.

24. Select a **SnapSync** policy.
24. EBS SnapSync Conflicts

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

25. Click Next.

26. Enter any custom pre-clone logic as Before Sync or After Sync hook operations.
Remember that adpreclone.pl dbTechStack is already run prior to every SnapSync of the dbTechStack.
The Before Sync hook operations will be run prior to running the adpreclone.pl tool.

For more information, see Customizing vFiles Management with Hook Operations.

27. Click Next.

28. Review the dSource Configuration and Data Management information, and then click Finish.
The Delphix Engine will initiate two jobs to create the dSource, DB_Link and DB_Sync. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have completed successfully, the files icon will change to a dSource icon on the Environments > Data bases screen, and the dSource will be added to the list of My Databases under its assigned group.

Link the EBS appsTier

1. Login to the Delphix Admin application using Delphix Admin credentials.

2. Click Manage.

3. Select Environments.

4. Select the source appsTier environment.
If you have not already added the environment, see the Managing Unix Environments topics for more information about adding environments.

5. Click the Environment Details tab.

6. If the applmgr environment user described in the Oracle Enterprise Business Suite Environment Requirements section is not already added to Delphix, add the user.
For more information about adding environment users, see the Managing Unix Environment Users topics.

7. Click the Databases tab.

8. Click the Plus icon next to Add Dataset Home.
Adding the files as a dataset home will register the type and location of the files with the Delphix Engine.

9. Select E-Business Suite R12.1 appsTier as your Dataset Home Type.
When you select this type of dataset home, the Delphix Engine will know to automate pre-clone logic. Specifically, adpreclone.pl appsTier will be run prior to every SnapSync of the appsTier. During
dSource creation, you will be able to enter additional pre-clone steps as Before Sync hook operations.

10. Enter an **Installation Home**.  
    This path should be the Oracle base install directory. For example, if the value of `$APPL_TOP` on your source environment is `/u01/oracle/VIS/apps/apps_st/appl`, the Oracle base install directory is `/u01/oracle/VIS`.

11. Click the **Check** icon to save your dataset home.

12. Scroll down the list of dataset homes to view and edit this dataset home if necessary.

13. Click **Manage**.

14. Select **Databases**.

15. Select **Add dSource**.

16. In the **Add dSource wizard**, select the **appsTier files source** you just created.

17. Enter the **EBS-specific parameters** for your appsTier.  
    These parameter values will be used when `adpreclone.pl` is run.  
    Ensure that the **Instance Home Directory** specifies an absolute path.

18. Click **Next**.

19. Enter a **dSource Name**.

20. Select a **Database Group** for the dSource.

21. Click **Next**.  
    Adding a dSource to a database group enables you to set Delphix Domain user permissions for that dSource's objects, such as snapshots. For more information, see the topics under **Users, Permissions, and Policies**.

22. Click **Next**.

23. Select a **SnapSync** policy.

    **EBS SnapSync Conflicts**

    When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

    To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

24. Click **Next**.

25. Enter any **custom pre-clone logic** as Before Sync or After Sync hook operations.  
    Remember that `adpreclone.pl` appsTier is already run prior to every SnapSync of the appsTier.  
    The Before Sync hook operations will be run prior to running the `adpreclone.pl` tool.

26. Click **Next**.

27. Review the **dSource Configuration** and **Data Management** information, and then click **Finish**.  
    The Delphix Engine will initiate two jobs to create the dSource, **DB_Link** and **DB_Sync**. You can monitor
these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System > Event Viewer**. When the jobs have successfully completed, the files icon will change to a dSource icon on the **Environments > Data bases** screen, and the dSource will be added to the list of **My Databases** under its assigned group.

**Related Links**

- Oracle Enterprise Business Suite Environment Requirements
- Linking an Oracle Data Source
- Provisioning EBS R12.1 Sources
- Managing Data Operations of Virtual EBS Instances
Provisioning EBS R12.1 Sources

This topic describes the process of provisioning a virtual instance of Oracle Enterprise Business Suite (EBS) R12.1.

Prerequisites

- Ensure that your EBS environment meets the criteria outlined in Oracle Enterprise Business Suite Environment Requirements
- You must have linked a source instance of Oracle Enterprise Business Suite R12.1. See Linking EBS R12.1 Sources for more information.

Snapshot Coordination

Changes applied to EBS and picked up only in certain dSource snapshots may make certain combinations of snapshots across the appsTier and dbTier incompatible. When provisioning, refreshing, or rewinding a virtual EBS instance, be sure the points in time you select for each dataset are compatible with each other.

Procedure

Provision the EBS dbTechStack

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the dbTechStack dSource.
6. Select a dSource snapshot.
   All snapshots will have staged configuration prepared by adpreclone.pl and any hook operations placed on the dSource.
7. Click Provision.
   The Provision vFiles wizard will open, and the field Mount Path will auto-populate with the path to the dbTechStack on the source dbTier environment. Change the value of Mount Path if you want to provision to another location on the target dbTier environment.
8. Select an Environment User.
   This user should be the oracle user outlined in Oracle Enterprise Business Suite Environment Requirements.
9. Enter the EBS-specific parameters for your target dbTier.
   These parameter values will be used to run adcfgclone.pl dbTechStack during the provision process. Ensure that the Target System Database Hostname is the short hostname, not the fully-qualified hostname.
   Ensure that the Target System Database SID is in upper case.
9. The source APPS password is required to configure the dbTechStack. This password is encrypted when stored within Delphix and is available as an environment variable to the `adcfgclone` process.

10. Click **Next**.

11. Enter a **vFiles Name**.

12. Select a **Target Group** for the vFiles.
   - If necessary, click the green **Plus** icon to add a new group.

13. Select a **Snapshot Policy** for the vFiles.
   - If necessary, click the green **Plus** icon to create a new policy.

14. Click **Next**.

15. Enter any **custom hook operations** that are needed to help correctly manage the dbTechStack files.
   - For more information about these hooks, when they are run, and how operations are written, see [Customizing vFiles with Hook Operations](#).
   - The Configure Clone hook will be run after the `adcfgclone.pl` tool has both mounted and configured the dbTechStack.

16. Click **Finish**.

17. Click **Finish**.
   - When provisioning starts, you can review progress of the job in the **Databases** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the dbTechStack vFiles will be included in the group you designated and listed in the **Databases** panel. If you select the dbTechStack vFiles in the **Databases** panel and click the **Open** icon, you can view its card, which contains information about the virtual files and its Data Management settings.

Register the EBS dbTechStack

Register the freshly-provisioned dbTechStack with the Delphix Engine.

1. Login to the **Delphix Admin** application using **Delphix Admin** credentials.

2. Click **Manage**.

3. Select **Environments**.
4. Select the **target dbTier environment**.

5. Click the **Databases** tab.

6. Click the **Plus** icon next to **Add Dataset Home**.

7. Select **Oracle** as your **Dataset Home Type**.

8. Enter an **Installation Home**.

   This path should be the value of $ORACLE_HOME on your target EBS environment. Commonly, this path looks like /u01/oracle/VIS/db/tech_st/11.1.0.

9. Click the **Check** icon to save your dataset home.

10. Scroll down the list of dataset homes to view and edit this dataset home if necessary.

11. Refresh the dbTier environment.

    Refreshing the environment will ensure that the EBS database listener is available for selection when provisioning the EBS database.

    a. Click the **Refresh** button in the bottom right-hand corner of the environment card.

**Provision the Oracle Database**

1. Provision the EBS database to the target dbTier environment by following the steps outlined in [Provisioning an Oracle VDB](#).

   Ensure that the **SID** is in upper case.

   ![EBS SnapSync Conflicts](image)

   When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

   To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

   ![EBS SnapSync Conflicts](image)

   2. Select the correct **Installation Home**.

      This selection should be the virtual dbTechStack you just added to the Delphix Engine.

   3. Select the correct **Oracle Node Listeners** value.

      This selection should be the listener corresponding to the virtual dbTechStack you just added to the Delphix Engine.

   4. Add a RunCommand operation to the **Configure Clone** hook that to ensure adcfgclone is run against the newly provisioned database. Typically, this operation will look similar to the below script.
# NOTE: Ensure the below environment variables will be set up correctly by the shell. If not, hardcode or generate the values below.

```bash
# CONTEXT_NAME=${ORACLE_SID}_${hostname -s}
# APPS_PASSWORD=<passwd>

. ${ORACLE_HOME}/${CONTEXT_NAME}.env

sqlplus "/ as sysdba" <<EOF
@${ORACLE_HOME}/appsutil/install/${CONTEXT_NAME}/adupdlib.sql so
EOF

perl ${ORACLE_HOME}/appsutil/clone/bin/adcfgclone.pl dbconfig
${ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml <<EOF
${APPS_PASSWORD}
EOF
```

Provision the EBS appsTier

1. Login to the Delphix Admin application using Delphix Admin credentials.

2. Click Manage.

3. Select Databases

4. Select My Databases.

5. Select the appsTier dSource.

6. Select a dSource snapshot.
   All snapshots will have staged configuration prepared by adpreclone.pl and any hook operations placed on the dSource.

7. Click Provision.
   The Provision vFiles wizard will open, and the field Mount Path will auto-populate with the path to the appsTier on the source appsTier environment. Change the value of Mount Path if you want to provision to another location on the target appsTier environment.

8. Select an Environment User.
   This user should be the applmgr user outlined in Oracle Enterprise Business Suite Environment Requirements.

9. Enter the EBS-specific parameters for your target appsTier.
   These parameter values will be used to run adcfgclone.pl appsTier during the provision process.
   Ensure that the Target System Application Hostname and Target System Database Server Node are the short hostnames, not the fully-qualified hostnames.
   Ensure that the Target System Database SID is in upper case.

   **Source APPS Password**
   The source APPS password is required to configure and manage the appsTier. This password is encrypted when stored within Delphix and is available as an environment variable to the adcfgclone, adstartl, and adstpal processes.

10. Click Next.
11. Enter a **vFiles Name**.

12. Select a **Target Group** for the vFiles.
   If necessary, click the green **Plus** icon to add a new group.

13. Select a **Snapshot Policy** for the vFiles.
   If necessary, click the green **Plus** icon to create a new policy.

---

**EBS SnapSync Conflicts**

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

---

14. Click **Next**.

15. Enter any **custom hook operations** that are needed to help correctly manage the appsTier files.
   For more information about these hooks, when they are run, and how operations are written, see [Customizing vFiles with Hook Operations](#).
   The Configure Clone hook will be run after the `adcfgclone.pl` tool has both mounted and configured the appsTier.

16. Click **Next**.

17. Click **Finish**.

---

**dbTier Must Be Accessible During appsTier Provisioning**

Post-clone configuration will fail if the appsTier cannot connect to the database. Ensure the target dbTier is accessible to the appsTier during the provisioning process. Ensure both the target database and database listener are running.

When provisioning starts, you can review progress of the job in the **Databases** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the appsTier vFiles will be included in the group you designated and listed in the **Databases** panel. If you select the appsTier vFiles in the **Databases** panel and click the **Open** icon, you can view its card, which contains information about the virtual files and its Data Management settings.

Once all three EBS virtual datasets have been provisioned successfully, your virtual EBS instance should be running and accessible.

**Related Links**

- [Oracle Enterprise Business Suite Environment Requirements](#)
- [Linking an Oracle Data Source](#)
Linking EBS R12.1 Sources

Managing Data Operations of Virtual EBS Instances
Linking and Provisioning Oracle Enterprise Business Suite R12.2

This section describes linking and provisioning Oracle Enterprise Business Suite R12.2.

- Linking EBS R12.2 Sources
- Provisioning EBS R12.2 Sources
**Linking EBS R12.2 Sources**

This topic describes the process of linking an Oracle Enterprise Business Suite (EBS) R12.2 instance and creating the necessary dSources.

**Prerequisites**

- Ensure that your EBS environment meets the criteria outlined in [Oracle Enterprise Business Suite Environment Requirements](#).

**Procedure**

**Link the Oracle Database**

1. Link the Oracle database used by EBS as outlined in [Linking an Oracle Data Source](#).

**EBS SnapSync Conflicts**

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

2. Set up a Before Sync hook operation to run any pre-clone steps necessary and specific to your EBS database. Normally, these steps will include running Oracle's `adpreclone` tool. Below is an example of a simple Run Shell Command hook operation:

```
# NOTE: Ensure the below environment variables will be set up correctly by the shell. If not, hardcode or generate the values below.
# ORACLE_HOME=/u01/oracle/VIS/11.2.0
# CONTEXT_NAME=${ORACLE_SID}_${hostname -s}
# APPS_PASSWD=<passwd>

. ${ORACLE_HOME}/${CONTEXT_NAME}.env
perl ${ORACLE_HOME}/appsutil/scripts/${CONTEXT_NAME}/adpreclone.pl database
<<-EOF
${APPS_PASSWD}
EOF
```

**Link the EBS dbTechStack**

1. Login to the [Delphix Admin](#) application using [Delphix Admin](#) credentials.

2. Click [Manage](#).

3. Select [Environments](#).

4. Select the [source dbTier environment](#) containing the source dbTechStack.
If you have not already added the environment, see the Managing Unix Environments topics for more information about adding environments.

5. Select the Environment Details tab.

6. If the oracle environment user described in the Oracle Enterprise Business Suite Environment Requirements section is not already added to Delphix, add the user.
   See the Managing Unix Environment Users topics for more information about adding environment users.

7. Click the Databases tab.

8. Click the Plus icon next to Add Dataset Home.
   Adding the files as a dataset home will register the type and location of the files with the Delphix Engine.

9. Select E-Business Suite R12.2 dbTechStack as your Dataset Home Type.
   When you select this type of dataset home, the Delphix Engine will know to automate pre-clone logic.
   Specifically, adpreclone.pl dbTechStack will be run prior to every SnapSync of the dbTechStack.
   During dSource creation, you will be able to enter additional pre-clone steps as Before Sync hook operations.

10. Enter an Installation Home.
    This path should be the directory above $ORACLE_HOME on your source environment. Commonly, this path looks like /u01/oracle/VIS.

11. Click the Check icon to save your dataset home.
12. Scroll down the list of dataset homes to view and edit this dataset home if necessary.
13. Click Manage.
15. Select Add dSource.
16. In the Add dSource wizard, select the dbTechStack files source you just created.

17. Fill in the EBS-specific parameters for your dbTechStack.
    These parameter values will be used when adpreclone.pl is run.
    Ensure that the Context Name uses the uppercase SID and the short hostname.

18. Click Advanced.

19. Exclude the EBS database's data files if they are stored underneath the dbTechStack root.
    This data was linked with the database instead of with the dbTechStack. Add the relative path to the data files to the Paths to Exclude list.

20. Click Next.

21. Enter a dSource Name.

22. Select a Database Group for the dSource.
    Click Next.
    Adding a dSource to a database group enables you to set Delphix Domain user permissions for that dSource's objects, such as snapshots. For more information, see the topics under Users, Permissions, and Policies.

23. Select a SnapSync policy.
EBS SnapSync Conflicts

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

24. Click Next.

25. Enter any custom pre-clone logic as Before Sync or After Sync hook operations. Remember that adpreclone.pl dbTechStack is already run prior to every SnapSync of the dbTechStack. The Before Sync hook operations will be run prior to running the adpreclone.pl tool.

For more information, see Customizing vFiles Management with Hook Operations.

26. Click Next.

27. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs to create the dSource, DB_Link and DB_Sync. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have completed successfully, the files icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of My Databases under its assigned group.

Link the EBS appsTier

1. Login to the Delphix Admin application using Delphix Admin credentials.

2. Click Manage.

3. Select Environments.

4. Select the source appsTier environment.
   If you have not already added the environment, see the Managing Unix Environments topics for more information about adding environments.

5. Click the Environment Details tab.

6. If the applmgr environment user described in the Oracle Enterprise Business Suite Environment Requirements section is not already added to Delphix, add the user. See the Managing Unix Environment Users topics for more information about adding environment users.

7. Click the Databases tab.

8. Click the Plus icon next to Add Dataset Home.
   Adding the files as a dataset home will register the type and location of the files with the Delphix Engine.

9. Select E-Business Suite R12.2 appsTier as your Dataset Home Type.
   When you select this type of dataset home, the Delphix Engine will know to automate pre-clone logic. Specifically, adpreclone.pl appsTier will be run prior to every SnapSync of the appsTier. During
dSource creation, you will be able to enter additional pre-clone steps as Before Sync hook operations.

10. Enter an Installation Home.
    This path should be the one level above Run/Patch edition file system base directory. For example, if the value of $RUN_BASE on your source environment is /u01/oracle/VIS/fs1, the install directory is /u01/oracle/VIS.

11. Click the Check icon to save your dataset home.
12. Scroll down the list of dataset homes to view and edit this dataset home if necessary.
13. Click Manage.
15. Select Add dSource.
16. In the Add dSource wizard, select the appsTier files source you just created.
17. Enter the EBS-specific parameters for your appsTier.
    These parameter values will be used when adpreclone.pl is run.
18. Click Advanced.
19. Add the relative paths of files to exclude to the Paths to Exclude list.
    Exclude the non-edition file system, instance directories, and FMW home directories which will be recreated when provisioning a new virtual EBS appsTier. These paths should typically include:
    - fs_ne
    - fs1/inst
    - fs2/inst
    - fs1/FMW_Home
    - fs2/FMW_Home
20. Click Next.
21. Enter a dSource Name.
22. Select a Database Group for the dSource.
23. Click Next.
    Adding a dSource to a database group enables you to set Delphix Domain user permissions for that dSource's objects, such as snapshots. For more information, see the topics under Users, Permissions, and Policies.
24. Click Next.
25. Select a SnapSync policy.
EBS SnapSync Conflicts

When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

26. Click Next.

27. Enter any custom pre-clone logic as Before Sync or After Sync hook operations. Remember that adpreclone.pl appsTier is already run prior to every SnapSync of the appsTier. The Before Sync hook operations will be run prior to running the adpreclone.pl tool.

28. Click Next.

29. Review the dSource Configuration and Data Management information, and then click Finish. The Delphix Engine will initiate two jobs to create the dSource, DB_Link and DB_Sync. You can monitor these jobs by clicking Active Jobs in the top menu bar, or by selecting System > Event Viewer. When the jobs have completed successfully, the files icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of My Databases under its assigned group.

Related Links

Oracle Enterprise Business Suite Environment Requirements

Linking an Oracle Data Source

Provisioning EBS R12.2 Sources

Managing Data Operations of Virtual EBS Instances
Provisioning EBS R12.2 Sources

This topic describes the process of provisioning a virtual instance of Oracle Enterprise Business Suite R12.2.

Prerequisites

- Ensure that your EBS environment meets the criteria outlined in Oracle Enterprise Business Suite Environment Requirements
- You must have linked a source instance of Oracle Enterprise Business Suite R12.2. For more information, see Linking EBS R12.2 Sources.

Snapshot Coordination

Changes applied to EBS and picked up only in certain dSource snapshots may make certain combinations of snapshots across the appsTier and dbTier incompatible. When provisioning, refreshing or rewinding a virtual EBS instance, be sure the points in time you select for each dataset are compatible with each other.

Procedure

Provision the EBS dbTechStack

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the dbTechStack dSource.
6. Select a dSource snapshot. All snapshots will have staged configuration prepared by adpreclone.pl and any hook operations placed on the dSource.
7. Click Provision. The Provision vFiles wizard will open, and the field Mount Path will auto-populate with the path to the dbTechStack on the source dbTier environment. Change the value of Mount Path if you want to provision to another location on the target dbTier environment.
8. Select an Environment User. This user should be the oracle user outlined in Oracle Enterprise Business Suite Environment Requirements.
9. Enter the EBS-specific parameters for your target dbTier. These parameter values will be used to run adcfgclone.pl dbTechStack during the provision process. Ensure that the Target System Database Hostname is the short hostname, not the fully-qualified hostname. Ensure that the Target System Database SID is in upper case.
10. Click **Next**.

11. Enter a **vFiles Name**.

12. Select a **Target Group** for the vFiles.
   If necessary, click the green **Plus** icon to add a new group.

13. Select a **Snapshot Policy** for the vFiles.
   If necessary, click the green **Plus** icon to create a new policy.

14. Click **Next**.

15. Enter any **custom hook operations** that are needed to help correctly manage the dbTechStack files.
   For more information about these hooks, when they are run, and how operations are written, see **Customizing vFiles with Hook Operations**.
   The Configure Clone hook will be run after the `adcfgclone.pl` tool has both mounted and configured the dbTechStack.

16. Click **Next**.

17. Click **Finish**.
   When provisioning starts, you can review progress of the job in the **Databases** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the dbTechStack vFiles will be included in the group you designated and listed in the **Databases** panel. If you select the dbTechStack vFiles in the **Databases** panel and click the **Open** icon, you can view its card, which contains information about the virtual files and its Data Management settings.

**Register the EBS dbTechStack**

Register the freshly-provisioned dbTechStack with the Delphix Engine.

1. Login to the **Delphix Admin** application using **Delphix Admin** credentials.

2. Click **Manage**.

3. Select **Environments**.
4. Select the target dbTier environment.

5. Click the Databases tab.

6. Click the Plus icon next to Add Dataset Home.

7. Select Oracle as your Dataset Home Type.

8. Enter an Installation Home.
   This path should be the value of SORACLE_HOME on your target EBS environment. Commonly, this path looks like /u01/oracle/VIS/11.2.0.

9. Click the Check icon to save your dataset home.

10. Scroll down the list of dataset homes to view and edit this dataset home if necessary.

11. Refresh the dbTier environment.
    Refreshing the environment will ensure the EBS database listener is available for selection when provisioning the EBS database.

    a. Click the Refresh button in the bottom right-hand corner of the environment card.

Provision the Oracle Database

1. Provision the EBS database to the target dbTier environment by following the steps outlined in Provisioning an Oracle VDB.
   Ensure that the SID is in upper case.

   - **EBS SnapSync Conflicts**
     When SnapSync is run against the dbTechStack, database, or appsTier, the Delphix Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple SnapSyncs are run against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.

     To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

2. Select the correct Installation Home.
    This selection should be the virtual dbTechStack you just added to the Delphix Engine.

3. Select the correct Oracle Node Listeners value.
    This selection should be the listener corresponding to the virtual dbTechStack you just added to the Delphix Engine.

4. Add a RunCommand operation to the Configure Clone hook that to ensure adcfgclone is run against the newly provisioned database. Typically, this operation will look similar to the below script.
Provision the EBS appsTier

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the appsTier dSource.
6. Select a dSource snapshot.
   All snapshots will have staged configuration prepared by adpreclone.pl and any hook operations placed on the dSource.
7. Click Provision.
   The Provision vFiles wizard will open, and the field Mount Path will auto-populate with the path to the appsTier on the source appsTier environment. Change the value of Mount Path if you want to provision to another location on the target appsTier environment.
8. Select an Environment User.
   This user should be the applmgr user outlined in Oracle Enterprise Business Suite Environment Requirements.
9. Enter the EBS-specific parameters for your target appsTier.
   These parameter values will be used to run adcfgclone.pl appsTier during the provision process. Ensure the Target System Application Hostname and Target System Database Hostname are the short hostnames, not the fully-qualified hostnames.
   Ensure that the Target System Database SID is in upper case.

   Source APPS Password

   The source APPS password is required to configure and manage the appsTier. This password is encrypted when stored within Delphix and is available as an environment variable to the adcfgclone, adstrtal, and adstpall processes.
10. Click Next.
11. Enter a vFiles Name.

12. Select a Target Group for the vFiles.
    If necessary, click the green Plus icon to add a new group.

13. Select a Snapshot Policy for the vFiles.
    If necessary, click the green Plus icon to create a new policy.

14. Click Next.

15. Enter any custom hook operations that are needed to help correctly manage the appsTier files.
    For more information about these hooks, when they are run, and how operations are written, see Customizing vFiles with Hook Operations.
    The Configure Clone hook will be run after the adcfgclone.pl tool has both mounted and configured the appsTier.

16. Click Next.

17. Click Finish.

When provisioning starts, you can review progress of the job in the Databases panel, or in the Job History panel of the Dashboard. When provisioning is complete, the appsTier vFiles will be included in the group you designated and listed in the Databases panel. If you select the appsTier vFiles in the Databases panel and click the Open icon, you can view its card, which contains information about the virtual files and its Data Management settings.

Once all three EBS virtual datasets have been provisioned successfully, your virtual EBS instance should be running and accessible.

Related Links

Oracle Enterprise Business Suite Environment Requirements
Linking an Oracle Data Source

Linking EBS R12.2 Sources

Managing Data Operations of Virtual EBS Instances
Managing Data Operations of Virtual EBS Instances

This section describes how to perform various data operations for virtual Oracle Enterprise Business Suite (EBS) instances once they have been provisioned.

- Starting and Stopping a Virtual EBS Instance
- Rewinding a Virtual EBS Instance
- Refreshing a Virtual EBS Instance
- Enabling and Disabling a Virtual EBS Instance
- Deleting a Virtual EBS Instance
Starting and Stopping a Virtual EBS Instance

This topic describes the process of starting and stopping a virtual Oracle Enterprise Business Suite (EBS) instance.

Virtual EBS instances are started and stopped through the Delphix Admin application or through the standard Oracle command line interface (CLI) utilities, adstrtal and adstpall. The Delphix Engine will show the dbTechStack and appsTier as running as long as there are processes using the dbTechStack and appsTier file system mounts on the target environments.

Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

Prerequisites

- You must have already provisioned a virtual EBS instance. For more information, see Virtualizing Oracle Business Suite: Getting Started.

Procedure

Stopping

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the appsTier vFiles for your EBS instance.
6. On the back of the card, move the slider control from Enabled to Disabled.
7. In the bottom right-hand corner, click Stop.
   Stopping the appsTier will run Oracle's adstpall.sh utility.

   ![](danger.png) Stopping the EBS application may take a long time. The Delphix Engine will wait for all Oracle application processes to exit before declaring the application as stopped.

8. Select the VDB utilized by your EBS instance.
9. In the bottom right-hand corner, click Stop.
   This action will shutdown the database instance.
10. Select the dbTechStack vFiles hosting your virtual EBS database.
11. In the bottom right-hand corner, click Stop.
    Stopping the dbTechStack will shut down the database listener.

Starting

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the dbTechStack vFiles hosting your virtual EBS database.
6. In the bottom right-hand corner, click Start.
Starting the dbTechStack will start the database listener.

7. Select the VDB utilized by your EBS instance.

8. In the bottom right-hand corner, click Start.
   Starting the database will open the database.

9. Select the appsTier vFiles for your EBS instance.

10. In the bottom right-hand corner, click Start.
    Starting the appsTier will run Oracle’s adstrtal.sh utility.

Related Links

- [Rewinding a Virtual EBS Instance](#)
- [Refreshing a Virtual EBS Instance](#)
- [Enabling and Disabling a Virtual EBS Instance](#)
- [Deleting a Virtual EBS Instance](#)
Rewinding a Virtual EBS Instance

This topic describes the process of rewinding a virtual Oracle Enterprise Business Suite (EBS) instance.

Changes applied to EBS and picked up only in some dSource snapshots may make certain combinations of snapshots across the appsTier and dbTier incompatible. When provisioning, refreshing, or rewinding a virtual EBS instance, be sure the points in time you select for each dataset are compatible with each other.

Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

Prerequisites

- You must have already provisioned a virtual EBS instance. For more information, see Virtualizing Oracle Business Suite: Getting Started.

Procedure

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the appsTier vFiles for your EBS instance.
6. Click the Stop icon to shutdown the appsTier services.
7. Select the VDB utilized by your EBS instance.
8. Click the Stop icon to shutdown the database.
9. Select the dbTechStack vFiles hosting your virtual EBS database.
10. Click the Stop icon to shutdown the database listener.
11. Rewind the dbTechStack vFiles.
   a. Select a snapshot.
   b. Click the Rewind button below the snapshots.
12. Rewind the EBS VDB.
   a. Select a snapshot.
   b. Click the Rewind button below the snapshots.
13. Rewind the appsTier vFiles.
   a. Select a snapshot.
   b. Click the Rewind button below the snapshots.

Once all three EBS virtual datasets have been rewound successfully, your virtual EBS instance should be running and accessible.

Related Links
• Starting and Stopping a Virtual EBS Instance
• Refreshing a Virtual EBS Instance
• Enabling and Disabling a Virtual EBS Instance
• Deleting a Virtual EBS Instance
Refreshing a Virtual EBS Instance

This topic describes the process of refreshing a virtual Oracle Enterprise Business Suite (EBS) instance.

Changes applied to EBS and picked up only in some dSource snapshots may make certain combinations of snapshots across the appsTier and dbTier incompatible. When provisioning, refreshing, or rewinding a virtual EBS instance, be sure the points in time chosen for each dataset are compatible with each other.

Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

Prerequisites
- You must have already provisioned a virtual EBS instance. For more information, see Virtualizing Oracle Business Suite: Getting Started.

Procedure
1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage
3. Select Databases.
4. Select My Databases.
5. Select the appsTier vFiles for your EBS instance.
6. Click the Stop icon to shutdown the appsTier services.
7. Select the VDB utilized by your EBS instance.
8. Click the Stop icon to shutdown the database.
9. Select the dbTechStack vFiles hosting your virtual EBS database.
10. Click the Stop icon to shutdown the database listener.

Clean Up Required Prior to Refresh

Before refreshing, make sure to clean up any files the virtual EBS instance might have created outside of the Delphix mount points. These would typically include the instance-specific directories and oraInventory files.

If you are refreshing a virtual EBS 11i instance, you must clean up stale entries in the oraTab file and oraInventory before performing a refresh. If you do not do this, adcfgclo ne will complain on refresh that a conflicting EBS instance is already installed.

11. Refresh the dbTechStack vFiles.
   a. On the back of the vFiles card, click the Refresh icon in the lower right-hand corner.
   b. Select a snapshot from which to refresh.
12. Refresh the EBS VDB.
a. On the back of the VDB card, click the **Refresh** icon in the lower right-hand corner.
b. Select a **snapshot** from which to refresh.

13. Refresh the appsTier vFiles.
   a. On the back of the vFiles card, click the **Refresh** icon in the lower right-hand corner.
   b. Select a **snapshot** from which to refresh.

Once all three EBS virtual datasets have been refreshed successfully, your virtual EBS instance should be running and accessible.

**Related Links**
- Starting and Stopping a Virtual EBS Instance
- Rewinding a Virtual EBS Instance
- Enabling and Disabling a Virtual EBS Instance
- Deleting a Virtual EBS Instance
Enabling and Disabling a Virtual EBS Instance

This topic describes the process of enabling and disabling a virtual Oracle Enterprise Business Suite (EBS) instance. An enabled virtual EBS instance will be running and fully available to end users. A disabled virtual EBS instance will be neither running nor mounted to the target environments.

⚠️ Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

**Prerequisites**

- You must have already provisioned a virtual EBS instance. For more information, see [Virtualizing Oracle Business Suite: Getting Started](#).

**Procedure**

**Procedure for Disabling**

1. Login to the **Delphix Admin** application using **Delphix Admin** credentials.
2. Click **Manage**.
3. Select **Databases**.
4. Select **My Databases**.
5. Select the **appsTier vFiles** for your EBS instance.
6. On the back of the card, move the slider control from **Enabled** to **Disabled**.
   Disabling the appsTier vFiles will stop the appsTier services and unmount the appsTier files.
7. Select the **VDB** utilized by your EBS instance.
8. On the back of the card, move the slider control from **Enabled** to **Disabled**.
   Disabling the VDB will stop the database instance and unmount the data files.
9. Select the **dbTechStack vFiles** hosting your virtual EBS database.
10. On the back of the card, move the slider control from **Enabled** to **Disabled**.
    Disabling the dbTechStack vFiles will stop the database listener and unmount the dbTechStack files.

Once all three EBS virtual datasets have been disabled successfully, your virtual EBS instance should be fully removed from the target environment.

⚠️ **Stopping EBS Might Be Slow**

In order to disable the virtual EBS instance, the Delphix Engine will stop it. Stopping the EBS application may take a long time. The Delphix Engine will wait for all Oracle application processes to exit before declaring the application as stopped.

⚠️ **Clean Up Required to Delete Virtual EBS**

If you plan to delete the virtual EBS instance, make sure to clean up any files the virtual EBS instance might have created outside of the Delphix mount points. These would typically include the instance-specific directories, oraInventory files and oraTab entries.

**Procedure for Enabling**

1. Login to the **Delphix Admin** application using **Delphix Admin** credentials.
2. Click **Manage**.
3. Select **Databases**.
4. Select **My Databases**.
5. Select the **dbTechStack vFiles** hosting your virtual EBS database.
6. On the back of the card, move the slider control from **Disabled** to **Enabled**.
   Enabling the dbTechStack vFiles will mount the dbTechStack files and start the database listener.
7. Select the **VDB** utilized by your EBS instance.
8. On the back of the card, move the slider control from **Disabled** to **Enabled**.
   Enabling the VDB will mount the data files and start the database instance.
9. Select the **appsTier vFiles** hosting your virtual EBS database.
10. On the back of the card, move the slider control from **Disabled** to **Enabled**.
    Disabling the dbTechStack vFiles will mount the appsTier files and start the application services.

Once all three EBS virtual datasets have been enabled successfully, your virtual EBS instance should be running and accessible.

**Related Links**
- [Starting and Stopping a Virtual EBS Instance](#)
- [Rewinding a Virtual EBS Instance](#)
- [Refreshing a Virtual EBS Instance](#)
- [Deleting a Virtual EBS Instance](#)
Deleting a Virtual EBS Instance

This topic describes the process of deleting a virtual Oracle Enterprise Business Suite (EBS) instance.

⚠️ Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

**Prerequisites**

- You must have already provisioned a virtual EBS instance. For more information, see [Virtualizing Oracle Business Suite: Getting Started](#).

**Procedure**

1. Login to the Delphix Admin application using Delphix Admin credentials.
2. Click Manage.
3. Select Databases.
4. Select My Databases.
5. Select the appsTier vFiles for your EBS instance.
6. Delete the appsTier vFiles by clicking the Trash Can icon in the bottom left-hand corner.
7. Select the VDB utilized by your EBS instance.
8. Delete the VDB by clicking the Trash Can icon in the bottom left-hand corner.
9. Click Manage.
10. Select Environments.
11. Select the target dbTier environment.
12. Click the Databases tab.
13. In the list of Installation Homes on the environment, click the Trash Can icon next to the dbTechStack you want to delete.
14. Click Manage.
15. Select Databases.
17. Select the dbTechStack vFiles for your EBS instance.
18. Delete the dbTechStack vFiles by clicking the Trash Can icon in the bottom left-hand corner.
19. Clean up any files the virtual EBS instance might have created outside of the Delphix mount points on the target environments. These would typically include the instance-specific directories, oraInventory files and oraTab entries.

Once all three EBS virtual datasets have been deleted successfully, your virtual EBS instance should be fully removed from the target environments.

**Related Links**
• Starting and Stopping a Virtual EBS Instance
• Rewinding a Virtual EBS Instance
• Refreshing a Virtual EBS Instance
• Enabling and Disabling a Virtual EBS Instance
_Compliance Engine
Agile Data Masking Admin Guide

As a Delphix Agile Data Masking Administrator, you specify the information (data elements) to be masked, how to mask the data (the algorithms to use), the location of the data to mask (regular expressions and profiler settings), and the roles or privileges for Delphix Agile Data Masking users. You perform all of these steps within Delphix Agile Data Masking and you can then propagate it across all of your organization's departments.

A **domain** is a virtual representation of a data element. An integral part of the data masking process is to use algorithms to mask each data element. The way you specify which algorithm to use on each individual data element is by creating a unique domain for each element. You do this on the Domains tab. You define a unique domain for each element and then associate the classification and algorithm you want to use for each domain.

In addition to using the Domain settings to determine your inventory of what to mask, a Profiling job uses expressions to identify the data you are seeking. A **regular expression** is a special text string that defines a search pattern. You can also group expressions into **profiler sets**, which are defined for a given target, such as financial services or health care.

Delphix Agile Data Masking has a built-in **Administrator** role, which gives a user complete access to Delphix Agile Data Masking functions. You can also add roles to the Roles Settings. For example, you can create an analyst role to allow a user to create masking jobs or an operator role to make sure jobs are run consistently.

**Related Links**
- Managing Settings
- Managing Users
Accessing Information About Your Software

To access the about screen first click the Admin tab at the top of the screen and then the About tab on the left of the screen.

From the About screen you can access information such as:

- The Delphix Agile Data Masking Version number
- The server’s Operating System
- The Application Server
- The installed Database type
- The installed Masking library
- The installed Java version
- The license Expiration Date
- The list of all the Licensed Data Sources
Agile Masking Security

The following sections describe security actions:

- Storing Database Passwords
- Authenticating Users
- Authorizing Users ( Roles)
Authenticating Users

If you choose to use Delphix Agile Data Masking internal authentication, Delphix Agile Data Masking uses encryption and stores passwords for each user encrypted in the Delphix Agile Data Masking relational repository.

When a user logs in to Delphix Agile Data Masking and enters their username and password, Delphix Agile Data Masking verifies that the user is an active user with Delphix Agile Data Masking, and then authenticates their password.

Optionally, Delphix Agile Data Masking can integrate with external authentication software (Microsoft Active Directory, CA SiteMinder, or LDAP) to authenticate users. If you integrate with external authentication software, Delphix Agile Data Masking will validate that the user has rights to access the application and will log in the user automatically. (No additional Delphix Agile Data Masking password will be required.)
Authorizing Users (Roles)

With the built-in Delphix Agile Data Masking Administrator role, which is similar to a superuser role, the administrator can add roles (Managing Roles Settings) and assign the roles to users (Managing Users). By creating specific roles and assigning them, the administrator can control which users are authorized to perform various tasks (privileges).
Storing Database Passwords

Delphix Agile Data Masking uses encryption and stores all passwords encrypted in the application's repository database.
Configuration
Configuring Delphix Agile Data Masking to Use Active Directory

Delphix Agile Data Masking can be configured to use the Active Directory environment to manage the login process. This is accomplished by modifying one of the Delphix Agile Data Masking property files with the appropriate information to communicate with the Active Directory infrastructure.

Configuration Steps

1. The first step, before you configure Delphix Agile Data Masking to use AD is to create a user in Delphix Agile Data Masking using your AD username. The Delphix Agile Data Masking username must match exactly your AD username as this is what we will be sending to AD for validation. You will have to put in a password, but this will not be used once AD is turned on. This user should be an administrator in Delphix Agile Data Masking as this will be the only valid user until more AD users are created.

2. Once this user is created, bring down Delphix Agile Data Masking.

3. Once Delphix Agile Data Masking is stopped, you need to edit the `dm-util.properties` file. This is located in the `<Delphix Agile Data Masking_home>/conf` directory.

4. Scroll down in the file until you come to the following section:

   ```
   #LDAP CONFIGURATION.
   LDAP_ENABLE=0
   LDAP_HOST=10.10.10.31
   LDAP_PORT=389
   LDAP_BASEDN=DC=tbspune,DC=com
   LDAP_FILTER=((&(objectClass=person)(sAMAccountName=?)) LDAP_ANONYMOUS=false
   MSAD_DOMAIN=AD
   LDAP_KERBEROS_AUTH=true
   LDAP_USERID_ATTR=msfwid
   ```

5. Set the following entries:

   ```
   LDAP_ENABLE=1
   LDAP_HOST=xxx.xxx.xxx.xxx (your AD host IP address) LDAP_PORT=389 (your AD host port, this is normally 389)
   ```

6. Save the file

7. Restart Delphix Agile Data Masking

Once Delphix Agile Data Masking comes up you should be able to login to Delphix Agile Data Masking using your AD login and password.

If this does not work, a few things are the possible cause:

1. You did not enter in you username in Delphix Agile Data Masking exactly the way AD expects it. In order to fix this, you will have to bring Delphix Agile Data Masking down. Edit the dm-util.properties file and change LDAP_ENABLE=0, and save the file. Restart Delphix Agile Data Masking and login as axistech, correct the AD user. Edit the property file again setting LDAP_ENABLE=1, and save the file. Bring Delphix Agile Data Masking down, and restart Delphix Agile Data Masking and then try the login again.

2. It is possible that your Active Directory environment is customized, we have run into this before and then you will need to open a support ticket and have your Active Directory support people available for consultation.
Configuring Log File Locations

Application log files and masking log files and reports are copied to a location determined by a path in a properties file. Over time, these locations might become full so, at some point, you might want to change the default locations.

- You MUST restart your application server after making changes to any properties file for the changes to take effect.

Changing the Application Log File Location

1. Open the /conf/log4j.properties file.
   For example:
   log4j.appender.R.File = C:/Tomcat 6.0/logs/Delphix Agile Data Masking/Delphix Agile Data Masking.log
3. Save and close the properties file.
4. Restart your application server.

Changing the Masking Log File and Report Locations

1. Open the /conf/dm-util.properties file.
2. Modify these keys in the file:
   - MASKING_LOG_PATH =
   - REPORT_PATH =
   For example:
   MASKING_LOG_PATH = D:/logs
   REPORT_PATH = D:/reports
3. Save and close the properties file.
4. Restart your application server.
Configuring the Default Port

The default port for application servers is 8080. Follow this procedure if you want to change this default.

- The example in this procedure is specific to Tomcat. Substitute values as appropriate for your application server.

Changing the Default Port for Application Servers

1. Go to the following location, where `<tomcat_home>` is the directory with the tomcat installation:
   `/<tomcat_home>/conf`
   For example:
   Delphix Agile Data Masking/apache_tomcat_6.0.18/conf

2. The `conf` folder is at the same level as the bin folder.

3. Modify this line in the `server.xml` file:
   `<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />`
   In this example, the default port is changed to 8443.

4. Save and close the file.

5. Restart your application server.
Restarting Delphix Agile Data Masking

In case of a power outage or other failure, you will need to restart your Delphix Agile Data Masking application. The procedure you follow differs depending on your application server.

Restarting Your Delphix Agile Data Masking Application for Tomcat/JBoss

1. Go to this directory, where `<tomcat_home>` is the directory with the tomcat installation:
   `/<tomcat_home>/bin`
   For example:
   Delphix Agile Data Masking/apache_tomcat_7.0.27/bin

2. Run the `startup.bat` file.

Restarting Your Delphix Agile Data Masking Application for WebLogic Server

1. Go to this directory, where `<bea_server_root>` is the location of your application server root folder:
   `<bea_server_root>/userprojects/domains/Delphix Agile Data Masking_domain/bin`
   
   For example:
   bea_Delphix Agile Data Masking/apache_tomcat_6.0.18/userprojects/domains/Delphix Agile Data Masking_domain/bin

2. Run the `startupWebLogic.cmd` file.

Restarting Your Delphix Agile Data Masking Application for IBM WebSphere

- Select Programs > IBM WebSphere > Application Server ... > Profiles > newly created profile > Start the server.

For example, if the default profile created when you installed WebSphere was AppSrv01, your newly created profile might be AppSrv02:
Programs > IBM WebSphere > Application Server ... > Profiles > AppSrv02 > Start the server.
Managing Algorithm Settings

An integral part of the data masking process is to use algorithms to mask each data element. You specify which algorithm to use on each individual data element (domain) on the Masking's tab. There, you define a unique domain for each element and then associate the classification and algorithm you want to use for each domain. Use the Algorithm settings tab to create or delete algorithms.
Adding New Delphix Agile Data Masking Algorithms

You might want to create a new algorithm if none of the default Delphix Agile Data Masking algorithms meet your needs

1. Click **Add Algorithm** at the top right of the Algorithm tab.
2. Choose one of the following:
   - **Secure Lookup Algorithm**
   - **Segmented Mapping Algorithm**
   - **Mapping Algorithm**
   - **Binary Lookup Algorithm**
3. Complete the form to the right (corresponding to your selected Algorithm)
4. Click **Save**.
Algorithm Settings Tab

The Algorithm tab displays algorithm Names along with Type and Description. This is where you add (or create) new algorithms. The default Delphix Agile Data Masking algorithms and any algorithms you have defined appear on this tab.

- All algorithm values are stored encrypted. These values are only decrypted during the masking process.
Binary Lookup Algorithm

A Binary Lookup Algorithm is much like the Secure Lookup Algorithm, but is used when entire files are stored in a specific column. This is useful for masking binary columns (e.g. blob, image, varbinary, etc).

1. Click Add Algorithm at the top right of the Algorithm tab.
2. Choose Binary Lookup Algorithm.
   The Binary SL Rule pane appears.
3. Enter a Rule Name.
4. Enter a Description.
5. Select a Binary Lookup File on your filesystem.
6. Click Save.
Mapping Algorithm

A mapping algorithm sequentially maps original data values to masked values that are pre-populated to a lookup table through the Delphix Agile Data Masking user interface. With the mapping algorithm, you must supply at minimum, the same number of values as the number of unique values you are masking, more is acceptable. For example, if there are 10000 unique values in the column you are masking you must give the mapping algorithm at least 10000 values.

Adding a Mapping Algorithm

1. Click **Add Algorithm** at the top right of the Algorithm tab.
2. Choose **Mapping Algorithm**.
   The MAPPING Rule pane appears.
3. Enter a **Rule Name**. (This name must be unique.)
4. Enter a **Description**.
   The value file must have NO header. Make sure there are no spaces or returns at the end of the last line in the file.
5. The following is sample file content (notice there's no header and only a list of values):

   Smallville
   Clarkville
   Farmville
   Townville
   Cityname
   Citytown
   Towneaster

6. Specify a **Lookup File (.txt){*}**.
7. To ignore specific characters, enter one or more characters in the **Ignore Character List** box. Separate values with a comma.

   To ignore the comma character (,), select the **Ignore comma (,)** check box.
8. When you are finished, click **Save**.
9. Before you can use the algorithm (specify it in a profiling or masking job), you must add it to a domain. If you are not using the Delphix Agile Data Masking Profiler to create your inventory, you do not need to associate the algorithm with a domain. See **Adding New Domains**.
Secure Lookup Algorithm

A secure lookup algorithm is a proprietary encrypt/hash/modulus algorithm that is repeatable but unbreakable. It lets you assign a realistic value from a list of predefined values. Use a secure lookup algorithm when you do not need unique values.

Adding a Secure Lookup Algorithm

1. Click Add Algorithm at the top right of the Algorithm tab.

2. Choose Secure Lookup Algorithm.
   The Create SL Rule pane appears.

3. Enter a Rule Name. (This name must be unique.)
4. Enter a Description.
5. Specify a Lookup File.

   This file is a single list of values. It does not require a header. Make sure there are no spaces or returns at the end of the last line in the file.

   The following is sample file content:

   - Smallville
   - Clarkville
   - FarmvilleTownville
   - Cityname
   - Citytown
   - Towneaster

   Delphix Agile Data Masking supports lookup files saved in ASCII or UTF-8 format only. If the lookup file contains foreign alphabet characters, the file must be saved in UTF-8 format for Delphix Agile Data Masking to read the Unicode text correctly.

6. When you are finished, click Save.
7. Before you can use the algorithm (specify it in a profiling or masking job), you must add it to a domain. See Adding New Domains.
Segmented Mapping Algorithm

Segmented mapping algorithms let you create unique masked values by dividing a target value into separate segments and masking each segment individually. Optionally, you can preserve the semantically rich part of a value while providing a unique value for the remainder. This is especially useful for primary keys or columns that need to be unique because they are part of a unique index.

- When using segmented mapping algorithms for primary and foreign keys, to make sure they match, you must use the same segmented mapping algorithm for each.
**Segmented Mapping Example**

Perhaps you have an account number for which you need to create a segmented mapping algorithm. You can separate the account number into segments, preserving the first two-character segment, replacing a segment with a specific value, and preserving a hyphen.

The following is a sample value for this account number: NM831026-04

Where:
- **NM** is a plan code number that you want to preserve, always a two-character alphanumeric code.
- **831026** is the uniquely identifiable account number. To ensure that you do not inadvertently create actual account numbers, you can replace the first two digits with a sequence that never appears in your account numbers in that location. (For example, you can replace the first two digits with 98 because 98 is never used as the first two digits of an account number.) To do that, you want to split these six digits into two segments.
- **04** is a location code. You want to preserve the hyphen and you can replace the two digits with a number within a range (in this case, a range of 1 to 77).

1. Choose 3 for **No. of Segment**. (Remember, you do not count the segment(s) you want to preserve.)
2. Preserve the first two characters (**NM** in sample value). Under **Preserve Original Values**:
   - For **Starting position**, enter 1.
   - For **length**, enter 2.
3. Define the next two-digit segment (**83** in sample value) to always be 98 or 99:
   - For **Segment 1**, choose **Type > Numeric**.
   - Choose **Length > 2**.
   - For **Mask Values Range#**, specify 98,99.
4. Define the next four-digit segment (**1026** in sample value):
   - For **Segment 2**, choose **Type > Numeric**.
   - Choose **Length > 4**.
   - Leave range fields empty.
   - Click **Add** to the right of **Preserve Original Values**.
5. Preserve the hyphen:
   - For **Starting position**, enter 9.
   - For **length**, enter 1.
6. Define the last two-digit segment (**04** in sample value):
   - For **Segment 3**, choose **Type > Numeric**.
   - Choose **Length > 2**.
   - For **Mask Values Min#**, specify 1.
   - For **Mask Values Max#**, specify 77.

The sample value NM831026-04 might be masked to NM98129177.
Segmented Mapping Procedure

1. Click **Add Algorithm** at the top right of the Algorithm tab.
2. Choose **Segmented Mapping Algorithm**.
   The Segmented Mapping pane appears.
3. Enter a **Rule Name**.
4. Enter a **Description**.
5. Select how many segments you want to mask (not counting the values you want to preserve) from the **No. of Segment** dropdown. (The minimum number of segments is 2; the maximum is 9.)
6. A box appears for each segment.
7. For each segment, choose the **Type** of segment from the dropdown: **Numeric** or **AlphaNumeric**.
   * **Numeric segments** are masked as whole segments. **Alphanumeric segments** are masked by individual character.
8. For each segment, choose the **Length** of the segment (number of characters) from the dropdown (maximum is 4).
9. Optionally, for each segment, specify range values. (You might need to specify range values to satisfy particular application requirements, for example.)

You can specify ranges for **Real Values** and **Mask Values**. With Real Values ranges, you can specify all the possible real values to map to the ranges of masked values. Any values *not* listed in the Real Values ranges would then mask to themselves.

- Specifying range values is optional. If you need unique values (for example masking a unique key column) you *must* leave the range values blank. If you plan to certify your data, you must specify range values.
- **Numeric** segment type:
  - **Min**—A number; the first value in the range. (Value can be 1 digit or up to the length of the segment. For example, for a 3-digit segment, you can specify 1, 2, or 3 digits. Acceptable characters: 0-9.)
  - **Max**—A number; the last value in the range. (Value should be the same length as the segment. For example, for a 3-digit segment, you should specify 3 digits. Acceptable characters: 0-9.)
  - **Range**—A range of numbers; separate values in this field with a comma (,). (Value should be the same length as the segment. For example, for a 3-digit segment, you should specify 3 digits. Acceptable characters: 0-9.)

If you do not specify a range, Delphix Agile Data Masking uses the full range. For example, for a 4-digit segment, Delphix Agile Data Masking uses 0-9999.

- **Alpha-Numeric** segment type:
  - **Min**—A number from 0 to 9; the first value in the range.
  - **Max**—A number from 0 to 9; the last value in the range.
  - **MinChar**—A letter from A to Z; the first value in the range.
  - **MaxChar**—A letter from A to Z; the last value in the range.
  - **Range**—A range of alphanumeric characters; separate values in this field with a comma (,). Individual values can be a number from 0 to 9 or an uppercase letter from A to Z. (For example, B,C,J,K,Y,Z or AB,DE.)

If you do not specify a range, Delphix Agile Data Masking uses the full range (A-Z, 0-9). If you do not know the format of the input, leave the range fields empty. If you know the format of the input (for example, always alphanumeric followed by numeric), you can enter range values such as A2 and S9.
• When determining a numeric or alphanumeric range, remember that a narrow range will likely generate duplicate values, which will cause your job to fail.
• To ignore specific characters, enter one or more characters in the **Ignore Character List** box. Separate values with a comma.
• To ignore the comma character (,), select the **Ignore comma (,)** check box.
• To ignore control characters, select **Add Control Characters**.

10. The Add Control Characters window appears.
11. Select the individual control characters that you would like to ignore, or choose **Select All** or **Select None**.
12. When you are finished, click **Save**.
   You are returned to the Segmented Mapping pane.
13. **Preserve Original Values** by entering **Starting position** and **length** values. (Position starts at 1.)
   For example, to preserve the second, third, and fourth values, enter Starting position 2 and length 3.
   If you need additional value fields, click **Add**.
14. When you are finished, click **Save**.
15. Before you can use the algorithm (specify it in a profiling or masking job), you must add it to a domain. If you are not using the Delphix Agile Data Masking Profiler to create your inventory, you do not need to associate the algorithm with a domain. See **Adding New Domains**.
Managing Domain Settings

The Domains tab is where you define Domains along with their Classification and default Masking Algorithm.

Delphix Agile Data Masking includes several default domains and algorithms. These appear the first time you display the Masking Settings tab. Each domain has a classification and masking method assigned to it. You might choose to assign a different algorithm to a domain, but each domain name is unique and can only be associated with one algorithm.

If you create additional algorithms, they will appear in the Algorithms dropdown. Because each algorithm used must have a unique domain, you need to add a domain (or reassign an existing domain) to use any other algorithms.
Adding New Domains

1. Click Add Domain at the Top of the Domains tab. A new domain will be created in-line.
2. Name the new Domain Name. The domain name you specify will appear as a menu option on the Inventory screen elsewhere in Delphix Agile Data Masking. Domain names must be unique.
3. Specify the Classification (informational only); for example, customer-facing data, employee data, or company data.
4. Select a default Masking Algorithm for the new domain. For information about algorithm settings, see Managing Algorithm Settings.
5. Click Save.

Deleting a Domain

- Click the Delete icon to the far right of the domain name.
Managing File Formats

A file format is a template that you can use again and again. This saves you time when creating an inventory for files of the same type. Instead of recreating an inventory for each file, you can create a file format to associate with the files.

The File Format tab displays the file formats that you have created. (If you have not created any file formats, the list will be empty.)
Creating a New File Format

1. Click Create Format at the upper right. The Create File Format window appears.

2. Enter a File Format Name.

3. Choose a File Format Type:
   - Delimited File
   - Excel Sheet
   - Fixed Width File
   - Type 19

4. Optionally enter a Description.

5. Click Submit.
Deleting a File Format

1. Click the **Delete** icon to the right of the File Format name.

2. File inventory is based on file format. Therefore, if you make a change to a file inventory, that change applies to all files that use that format.

3. You can only add or delete a file format, you cannot edit one.
Importing a New File Format

1. Click Import Format at the upper right The Import File Format window appears
2. Select an Import File Type

For a Format Type of Copybook or XML:

1. Select a Connection Mode
2. Fill out the required fields of the selected Connection Mode
3. Click Browse
4. Click the Select button to the right of the desired import file format
5. Enter a Logical Name
6. Click Submit

For a Format Type of Delimited File, Excelsheet, or Fixed Width File:

1. Click the Select… button
2. Browse for the file to import fields from
3. Click Save

The file must have NO header. Make sure there are no spaces or returns at the end of the last line in the file. The field names must be in the same order they are in the file to be masked.

The following is sample file content for Delimeted or Excel file formats. With these formats just the field name is provided.(notice there's no header and only a list of values):

First_Name Last_NameDOBSSNAddress
CityState
Zip_Code

The following is sample file content for Fixed Width format. In this format the field name is followed by the length of the field, separated by a comma. (notice there's no header and only a list of values):

First_Name,20Last_Name,30DOB,10SSN,11Address,30
City,20State,2
Zip_Code,10
Managing Mapping Settings
Adding Mappings

1. Click Add Mapping at the upper right. The Add Mapping Rule window appears.
2. Select a Mapping Type.
3. Enter a Mapping Name.
4. Enter values for Input and Output.
5. Select a Mapping File from the filesystem.
6. Click Submit.
Managing Profiler Settings

The Delphix Agile Data Masking profiler uses two different methods to identify the location of sensitive data:

- **At the metadata level**—searches through the column names in the target database, by querying the database catalog, looking for specific words in column names (for example, column names with "name" in them).
- **At the data level**—looks at the data itself using a sampling algorithm, to see whether there is any sensitive data.

Delphix Agile Data Masking then uses that profile information to generate the appropriate jobs that will mask the target database. The user defines the connections to the databases to profile and then uses the Delphix Agile Data Masking software to perform the Profiling. When the profiling is complete, the information is stored as profile metadata for Delphix Agile Data Masking processing in the locally hosted or network Delphix Agile Data Masking database.
Adding New Expressions

Expressions let you specify how you want to profile data by letting you determine the data to profile based on the criteria you enter in the expressions. For example, you can define an expression that looks for a name or partial name for a column and only profiles data in columns that match that name or partial name. The following table shows some sample expressions.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Column Description</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i:ad(dress)_line 1</td>
<td>ad(dress)1</td>
<td>city_ad(dress)</td>
</tr>
<tr>
<td>([s]+b(ou)?l(e)?v(ar)?d[[s].+)([s] +st[word]?[[s].+)([s]+ave[[s]?[nu e]?[[s].+)([s]+r(oa)?d[[s].+)([s]+ l(a)?n(e)?[[s].+)([s]+cir(cle)?[[s].+ )</td>
<td>Looks for address line information in data</td>
<td></td>
</tr>
<tr>
<td>(?i)([s]*ap(ar)?t(ment)?[[s].+)([s ]*s(ui)?te[[s].+)(c(are)?[[s][\]]?o(f)?[[s].+ )</td>
<td>Looks for address line 2 information in the data</td>
<td></td>
</tr>
</tbody>
</table>

For sample expressions and tools, see [http://www.regular-expressions.info/](http://www.regular-expressions.info/) or perform an Internet search for “regular expressions”. (Disclaimer: We have provided this resource as a suggestion. Delphix does not endorse this or any other related site.)

Add an Expression

1. Click Add Expression at the top of the Profiler tab.
   - A new expression will be created in-line.

2. Select a domain from the Domain dropdown.
   Note: Only the default Delphix Agile Data Masking domains and the domains you have defined appear in this dropdown. If you need to add a domain, see Adding New Domains.

3. Enter the following information for that domain:
   - **Expression Name**—The field name used to select this expression as part of a profiler set.
   - **Expression Text**—The regular expression used to identify the location of the sensitive data.

4. Select an Expression Level for the domain:
• **Column Level**—To identify sensitive data based on column names.
• **Data Level**—To identify sensitive data based on data values, not column names.

2. When you are finished, click **Save**.

**Delete an Expression**

• Click the **Delete** icon to the far right of the name.
Adding or Editing a Profiler Set

- You can define Profiler Sets in Delphix Agile Data Masking. A profiler set is a grouping of expressions for a particular purpose. For instance, First Name, Last Name, Address, Credit Card, SSN, and Bank Account Number could constitute a Financial Profiler Set. For information about creating a profiling job, see "Creating a New Profiling Job" in Delphix Agile Data Masking User's Guide.

- Delphix Agile Data Masking comes with two predefined profiler sets: Financial and Healthcare vertical. A Delphix Agile Data Masking administrator (a user with the appropriate role privileges) can create/add/update/delete these profiler sets.

- If you don't choose a profiler set as part of the Profiler job, Delphix Agile Data Masking profiles data based on all the expressions defined on the Profiler Settings page.

- If you want to edit or add a profiler set, click **Profiler Set** at the top of the Profiler tab. The Profiler Set screen appears, listing the profiler sets along with their Purpose and Date Created.

Editing a Profiler Set

- Click the **Edit** icon to the right of the Profiler Set name.

Deleting a Profiler Set

- Click the **Delete** icon to the right of the Profiler Set name.

Adding a Profiler Set

1. Click **Add Set**.
   - The Create Profile Set window appears.
2. Enter a profile **Set Name**.
3. Optionally, enter a **Purpose** for this profile set.
4. Enter/select which **Domains** to include in this set.
5. When you are finished, click **Submit**.
Practical Profiling Example

This section provides an example of how you might define the data you want to profile.

Starting on the Profiler Settings page, you might want to look for First Name. Specify a regular expression to specify how to look for it. If the expression is column-name specific, Delphix Agile Data Masking will identify which column names match the pattern specified in the expression. If Delphix Agile Data Masking finds a match, it will tag it as a sensitive column. If an expression matches multiple columns in a table, Delphix Agile Data Masking tags all the columns for which it finds a match, not just the first column in the table.

However, if multiple expressions match one column, Delphix Agile Data Masking tags the first match in that column.

Profiling data takes a sample against the column. (Data sampling does not apply to mainframe processing.) Delphix Agile Data Masking does not look at all rows, but the first \( n \) (\( n \) being 10,000 rows, 100,000 rows, and so on). (The value of \( n \) is set in the kettle-profiling.properties file by the NO_OF_ROWS property.)

So, if you want to look for First Names across all of your databases, specify the following expression on the Profiler Settings page:

\[ [Nn][Aa][Mm][Ee] \]

If the expression is at a data level, you can look for common names such as John and Mary:

\[ (([Jj][Oo][Hh][Nn])|([Mm][Aa][Rr][Yy])) \]

This expression looks for the names John and Mary in the database. If Delphix Agile Data Masking finds any, it identifies that as a First Name column.

You can also search based on format. For instance you can look for a social security number by looking for nine digits of data, with two hyphens (at positions 4,1 and 7,1): \(^\text{^d(3)}-\text{^d(2)}-\text{^d(4)}\)$
Profiler Settings Tab

You can add regular expressions and profiler sets to the Profiler Settings. In addition to using the Masking settings to determine your inventory of what to mask, a Profiling job uses expressions to identify the data you are seeking. For more information about profiling, see "About Profiling Data" in Delphix Agile Data Masking User's Guide.

The Profiler displays Domains along with their Expression Text, Expression Name, and Expression Level.
Managing Remote Servers

- This feature is an add-on for Delphix Agile Data Masking Standard Edition.

Delphix Agile Data Masking typically executes jobs on a local instance. To execute jobs on a remote server, use this tab to define the server(s).

If you have already defined remote server(s), they appear on the Remote Server tab, in the Remote Server Name column.

Defined remote servers also appear in a dropdown on the Create Job window when you create any new job.
Defining a Remote Server

1. Click **Create Remote Server** to the upper right, below the tabs. The New Remote Server window appears.
2. Enter a name for the remote server in the **Remote Server Name** field.
3. In the **Host Name/IP** field, enter the name of the remote server host or the IP address of the remote server.
4. Enter the **Port** on which the remote client is listening for job requests.
5. Enter a **User Name** to access the remote server.
6. Enter the **Password** for the specified User Name.
7. In the **Remote Application Home** field, enter the path on the remote server to the home directory for the *Delphix Agile Data Masking* client.
8. Click **Submit**.
Deleting a Remote Server

- Click the **Delete** icon to the right of the Remote Server Name.
Editing a Remote Server

- Click the **Edit** icon to the right of the Remote Server Name.
Managing Roles Settings

Delphix Agile Data Masking has a built-in Administrator role, which gives a user complete access to Delphix Agile Data Masking functions. This is similar to a superuser role. The administrator can access, update, and delete all environments, and all objects within those environments. The administrator can also add roles to the Roles Settings. The following are some sample roles and associated tasks:

- **Analyst role**—Can profile data and update inventories (but not create environments or connections)
- **Developer role**—Can create masking jobs and view reports
- **Operator role**—Can execute jobs (but cannot update inventories)
- **Application owner role**—Can define connections

Each username can only have one role assigned to it.
Adding Roles
1. Click **Add Roles** near the top of the Roles tab.
2. Enter a **Role Name**.
   - The types of privileges appear across the top of the table, corresponding to the columns of check boxes:
     - View
     - Add
     - Update
     - Delete
     - Copy
     - Import
     - Export
   - The far-left column lists the items for which you can set privileges.
3. Select the check boxes for the corresponding privileges that you want to apply. If there is no check box, that privilege is not available.
   - For example, if you want this Role to have View, Add, Update, and Run privileges for Masking jobs, select the corresponding check boxes in the Masking Job row.
4. When you are finished assigning privileges for this Role, click **Submit**.
Roles Settings Tab

The Roles tab displays the Roles that you have created. (If you have not created any roles, the list will be empty.) If you do not have the appropriate privileges, the Roles tab will not appear. When you click on a Roles Name or Add Roles, a screen similar to the following appears:
Managing Users
Creating and Editing Users

Managing Users

Click the Admin tab at the top and the then the Users tab on the left of the screen to display the list of users defined in the Delphix Agile Data Masking installation.

Creating a New User

- Click Add User at the upper right of the Users screen.

You will be prompted for the following information:

- **First Name**—The user’s given name.
- **Last Name**—The user’s surname.
- **User Name**—The login name for the user.
- **Email**—The user’s e-mail address (mailable from the Delphix Agile Data Masking server for purposes of job completion e-mail messages).
- **Password**—The password Delphix Agile Data Masking uses to authenticate the user on the login page. The password must be at least six characters long, and contain a minimum of one uppercase character, one wild character (’@#$%^&*’), and one number.
- **Confirm Password**—Confirm the password with double-entry to avoid data entry error.
- **Administrator**—(Optionally) Select the Administrator check box if you want to give this user Administrator privileges. (Administrator privileges allow the user to perform all Delphix Agile Data Masking tasks, including creating and editing users in Delphix Agile Data Masking.)
- If you select the Administrator check box, the Roles and Environments fields disappear because Administrator privileges include all roles and environments.
- **Role**—Select the role to grant to this user. The choices here depend on the custom roles that you have created. (For information about creating custom roles, see Managing Roles Settings on page .)
- **Environment**—Enter as many environments that this user will be able to access
- Granting a user access to a given environment does not give them unlimited access to that environment. The user's access is still limited to their assigned role.

When you are finished, click Save.

Editing a User

1. Click on the User Name in the user list. The User Profile pane appears.
2. Modify the settings as you would for a new user.
3. Click Save.

Deleting a User

1. Click the Delete icon.
The Settings Screen

Display the Settings screen by clicking the **Settings** tab at the top of any Delphix Agile Data Masking screen.

- You must have the appropriate user privileges to see this screen.

The Settings Screen has the following tabs:

- **Algorithm**—Define the algorithms to use to mask your data.
- **Domains**—Define domains and choose their classification and default masking algorithm.
- **Profiler**—Define expressions and groupings of expressions used to create your inventory.
- **Roles**—Define user roles and privileges, such as edit and delete.
- **Mapping**—Define both Delphix Agile Data Masking and Informatica mapping rules
- **File Format**—Define the file format definitions and format types.
- **Remote Server**—(This is an add-on feature for Delphix Agile Data Masking Standard Edition.) Define the remote server(s) that will execute jobs.
Troubleshooting
**Application Server Down**

If the application server is down, the application is also down. The repository might still be available, if the database server is still up. If the server is down, do the following:

1. Restart the database server, if necessary.
2. Restart the application server.
3. Ensure that both servers are up and running.
4. If the problem persists, contact Customer Support.
Database Server Down

If the database server is down, do the following:

1. Restart the database server.
2. Restart the application server, if necessary.
3. Ensure that both servers are up and running.
4. If the problem persists, contact Customer Support.
Memory Usage

Delphix Agile Data Masking masking operations can be memory- and processor-intensive. Therefore, the number of jobs that can run in parallel and the speed with which they run varies depending on processor and RAM.

Initially, we recommend that you allocate at least 1 GB for the Tomcat application server instance. Other application servers might require more memory; follow the suggested guidelines for your server. If you encounter memory issues, you might need to increase your memory allocation.

- 32-bit Java Virtual Machines (JVMs) have a maximum memory setting (1.5 GB) that you cannot exceed. 64-bit JVMs do not have this restriction.

If you do not allocate enough memory initially, you could have issues if you try to allocate memory as needed. To avoid this problem, we suggest that you set your Java Xms and Xmx values to the same number. This ensures that all necessary memory is reserved and available for the job at the beginning. Otherwise, your operating system might attempt to terminate some lower priority processes to free up memory, which could halt your higher priority processes. We recommend allocating 1 GB per job.

For information on system requirements, see Delphix Agile Data Masking System Requirements.
Stack Traces

If an unhandled exception occurs in code, you might get a stack trace. If this happens, do the following:

1. Restart the server.
2. Ensure that the database is up.
3. If the problem persists, contact Customer Support.
Utilization Reports
The Utilization Screen

Click the Admin tab at the top and then the Utilization tab on the left to bring up the utilization screen.

Generating a Utilization Report

1. Select the Environment you want a utilization report for.
2. Select the Start Date of the report.
3. Select the End Date of the report.
4. Click Create at the top right.
Delphix Agile Data Masking User Guide

Data masking is a technique used to secure your nonproduction environments (such as Development and QA environments) by replacing sensitive data with fictitious but realistic data, eliminating the risk of exposing your sensitive data to unauthorized parties. The Delphix Agile Data Masking unique approach to data masking breaks the association between unique identifying information (for example, social security numbers) and personally identifiable data (for example, name and address). Delphix Agile Data Masking, using an Extract-Transform-Load (ETL) process, reads data from a source, transforms the data in memory, and loads data to the target. Delphix Agile Data Masking processes for protecting sensitive data are measurable, documented, and repeatable, providing clearly auditable results.

Delphix Agile Data Masking is a multi-user web application that provides a complete, secure, and scalable solution for your data masking needs, while meeting enterprise-class infrastructure requirements. There are several advantages to using the Delphix Agile Data Masking solution to mask your data:

- **Risk reduction**—By creating masked data that is suitable for a large number of business activities, potential sources for breaches are greatly reduced.
- **Automated identification of sensitive data**—By identifying the location of sensitive data automatically across databases, copybooks, and files, the time-consuming work associated with a data masking project is reduced significantly, thereby reducing time and costs as well.
- **Automated creation and masking of sensitive data**—By quickly and consistently creating masked data automatically, no programming is required. This also results in significant time savings. With Delphix Agile Data Masking, you create XML that is automatically imported into your Informatica Client (via the PMCMD interface), no programming is required.
- **Maintain referential integrity**—Repeatable algorithms automatically maintain referential integrity in Delphix Agile Data Masking, not only within a database but across databases and platforms.
- **Leverage investment in Informatica**—Delphix Agile Data Masking can automatically create mappings in Informatica based on the Sensitive Data inventory. This allows you to use the powerful Extract-Transform-Load (ETL) engine in Informatica PowerCenter, within your existing infrastructure, without needing to code and maintain hundreds or thousands of Informatica jobs. Another major advantage of Delphix Agile Data Masking is that you can use the masking algorithms deployed as mapplets, use mapplets provided by Informatica, or write your own mapplets to use with Delphix Agile Data Masking.
- **Enabled data sharing**—By sharing masked data safely and easily, relationships with partners, third parties, and outside vendors are improved.
- **Lower costs**—By automating the entire masking process and securing only essential data (thereby reducing how much data to secure), and provisioning only the data you need, costs to secure sensitive data are greatly reduced.
**About the Delphix Agile Data Masking Solution and This Guide**

Delphix Agile Data Masking uses an Extract-Transform-Load (ETL) process to read data from a source, transform the data in memory, and then load data to the target. With Delphix Agile Data Masking, you can have multiple connections to databases and files. You can create one application environment and then copy it, so you can replicate the inventory for different environments for that application (QA, Development, Testing, and so on). For example, you can create one environment for an application, such as Development, and replicate it for QA, UAT, SIT, and so on, without having to duplicate the effort of setting up multiple environments. Delphix Agile Data Masking makes this entire process simple.

The Delphix Agile Data Masking solution consists of the following process:

- **Identify sensitive data**—Assess where your existing sensitive data is stored.
- **Select data inventory**—Choose the data that you want to mask, based on rule sets.
- **Validate data inventory**—Confirm that the data inventory specifies all your sensitive data.
- **Implement data masking**—Generate code to perform masking such that the masked data is usable as production data. This code is copied to the mainframe so you can run the masking jobs from the mainframe as required. Delphix Agile Data Masking generates XML that is automatically imported into your PowerCenter repository via your Informatica Client (via the PMCMD interface).
- **Certify data masking**—Perform certification on an ongoing basis to ensure that you have not introduced unmasked data.

Delphix Agile Data Masking uses the Informatica PowerCenter data integration platform as an *applicator* to perform the data transformation. Delphix Agile Data Masking seamlessly performs data transformation, thereby letting you maximize the potential of your applicator software without doing any programming. Delphix Agile Data Masking does that for you automatically.

The purpose of this document is to walk you through a Quick-Start Scenario (see "Getting Started"), explain the various processes involved in data masking (see "Chapter 2: About the Major Delphix Agile Data Masking Components"), and document the software itself (see "Chapter 3: Using Delphix Agile Data Masking").
About the Major Delphix Agile Data Masking Components

Delphix Agile Data Masking consists of the following major components:

1. **Profiling**
2. **Provisioning**
3. **Masking**
4. **Certification**

The following sections describe each component.
About Certifying Data

After profiling and masking data, you want to monitor or audit the process (also known as certifying your data). This alerts you if unmasked data is introduced to a masked database.

For example, if you mask your master customer database once a week, and an input file of unmasked is introduced by mistake, you want to be able to detect that. The purpose of the Delphix Agile Data Masking certification module is to identify such a situation. To do so, you create a Certification job against that database (see "Creating a New Certify Job").

The Certifying job goes through every row in the tables in a rule set and verifies that every value designated for masking in the inventory is masked. The Certification job output lists the fields designated for masking, along with the result of the certification: Clean, Polluted, or Not Applicable. Polluted data indicates that Delphix Agile Data Masking encountered a value that could potentially be an unmasked production value. Not Applicable indicates that Delphix Agile Data Masking was unable to determine whether the value is masked.
A Practical Certification Example

When a certification job runs, it looks at the inventory defined for that database. If the inventory indicates to mask the Customer.First_Name column with the First Name Secure Lookup algorithm, the certification job ensures that there are only masked values for that column.

The First Name Secure Lookup algorithm uses a look-up file that contains all the first names used to mask a column. When the certification job runs, it compares the values in the Customer.First_Name column with the look-up file, which contains only the mask values.

If any names in the column do not match the names in the look-up file, Delphix Agile Data Masking indicates that it found polluted data in the Customer.First_Name column.

You can use the Delphix Agile Data Masking Certification module to certify all data masked with Secure Lookup Algorithms.

You can also certify data masked with Segmented Mapping Algorithms, keeping in mind that the segmented mapping algorithm must specify a range for each segment.

For example, if the SSN Segmented Mapping algorithm first segment has a range from 800-899, and there is SSN data in the Customer.SSN column with the first three digits of 435, the certification job will tag the Customer.SSN column as containing polluted rows.

With the Telephone algorithm, the middle segment is the exchange. Because the exchange is always masked to an exchange of 555, any other exchange will indicate polluted data.

Axis Technology recommends that you run certification jobs at least once in between refreshes, when the refreshes for a database are scheduled 7 days or more apart.
Certification and Delta Masking

As part of the certification process for databases, you can specify to mask rows that are not masked but are identified as part of the certification process as polluted. In other words, if the certification process finds "polluted rows," *Delta Masking* instructs Delphix Agile Data Masking to mask the polluted records to ensure that those polluted rows get masked.
About Masking Data

After you create an environment, connection, rule set, and inventory, you mask data.

To maintain Referential Integrity (RI), Delphix Agile Data Masking masks each field on itself. This repeatable masking automatically maintains RI (for verbatim matches), even if it's between applications or platforms.

For example, if you want to match the values between a parent and children, simply select the same algorithm to mask them. This ensures that referential integrity is maintained within the same database. Furthermore, Delphix Agile Data Masking creates the integrity across database platforms (between SQL Server and DB2, for example) or across files (tab-delimited files) and relational data (a column in a SQL Server database)—just select the same masking algorithm.

As a practical example, assume you have an SSN column in a Microsoft SQL Server database, an SSN column in a DB2 database, and an SSN field in a tab-delimited file. If the SSN value was 111111111 across the two databases and the file, and you use the same SSN algorithm for all three, the masked value (for example, 801-01-0838) will be the same for all three.

There are two ways to mask data. You can mask data on-the-fly or you can provision it first and then mask it. The following sections explain these two options.
About Masking In-Place

With in-place masking, production data that already exists in a nonproduction environment is masked, in place.

Advantages/Disadvantages:
The main advantage to in-place masking is when you have provisioned data to a non-production environment that contains some production data. Delphix Agile Data Masking can mask the data in those existing environments. In-place masking masks only the columns you flag in the inventory, leaving the other columns alone.

The main disadvantage is that production data is copied potentially into a nonproduction environment while the masking takes place, so sensitive data might exist in the nonproduction environment until the masking is complete. DMsuite Premier Edition generates XML that is automatically imported into your PowerCenter repository via your Informatica Client. The XML files create your mappings and your workflow.
About On-The-Fly Masking

With on-the-fly masking, you specify the source of the information to be masked, and where the masked data will be loaded. On-the-fly masking is an Extract Transform Load (ETL) process.

Delphix Agile Data Masking extracts the data from a source environment, such as a production copy, gold copy, or disaster recovery copy (only read from a database not an archived file).

Delphix Agile Data Masking transforms, or masks, the data in the memory of the application server on which it resides, and then loads the masked data to the target environment. Delphix Agile Data Masking does not modify the original source data; only the target data changes.

Advantages/Disadvantages:

One advantage to on-the-fly masking is that sensitive production data doesn't get persisted in any nonproduction environment. This method only requires a production source and nonproduction target environment. Because on-the-fly masking uses all insert statements, it typically performs better than in-place masking, which uses updates.

The main disadvantage to on-the-fly masking is that it requires an active connection to a source production environment or copy. DMsuite Premier Edition generates XML that is automatically imported into your PowerCenter repository via your Informatica Client. The XML files create your mappings and your workflow.
Masking a Primary Key Column

Because primary keys require unique values, you must mask those columns using a Delphix Agile Data Masking algorithm that can guarantee uniqueness. You apply the same mapping algorithm to both the primary key column and the foreign key column so the values between the columns will match. For information about creating algorithms, see Delphix Agile Data Masking Administrator's Guide.
About Profiling Data

Profiling is one of the major components in Delphix Agile Data Masking. The objective of profiling is to identify the location of Non-Public Information (NPI) or sensitive data.

The Delphix Agile Data Masking profiler uses two different methods to identify the location of sensitive data:

1. Search through the column names in the target database, by querying the database catalog (metadata).
2. Look at the data itself using a sampling algorithm, to see whether there is any sensitive data. (This is especially useful for files, and comment and notes fields in a database, for example.)

After you have defined an environment and a connection for your data source, you can profile the data. To do so, you create a profiling job (see “Creating a New Profiling Job”).
About Provisioning (Subsetting Data)

Delphix Agile Data Masking allows you to provision a subset of data, which provides your development and testing teams with the ability to procure data from a source environment (such as gold copy or masked copy) to their nonproduction environment.

Provisioning performs Extract and Load functions only. Delphix Agile Data Masking does not transform data as part of this process.
Practical Subset Example

This section provides an example of when you might want to subset data. All the functionality for provisioning is done at the individual table level.

If you want to filter information based on customers, as part of the rule set you can filter as follows:

1. Columns—Select a column or columns from a table when you don't want to load data to all the columns in a table.
2. Filter—Specify a filter to run on the data before loading it to the target database.
3. Join Table—Specify a SQL join condition so you can define primary key/foreign key relationships between tables.
4. List—Select a list to use for filtering data in a table.
5. Custom SQL—Use SQL statements to filter data for a table.

To change these filters, see Modifying Tables in a Rule Set.
Getting Started with Agile Data Masking

This section describes how to use Delphix Agile Data Masking and outlines the steps you need to follow to mask your data or provision and mask your data.

- Starting Delphix Agile Data Masking
- Basic Tasks for a Database File
- Create or Import an Environment
- Create and Define Connections
- Create and Define a Rule Set
- Profile Data and Define Inventory
- Provision Data
- Mask Data
- Certify Data
Getting Started with Delphix Agile Data Masking

Getting Started - User Work Flow Summary for Agile Data Masking

Delphix Agile Data Masking is a Web application that you use within a browser window. The documentation that follows summarizes the common user work flow to provision and mask data in the Delphix Engine.

- **Activity One: Provision an Oracle Virtual Database in the Delphix Engine**
- **Activity Two: Mask Data in Agile Masking Interface**
- **Activity Three - Refresh Data from the Source in Delphix Engine**

Activity One: Provision an Oracle Virtual Database in the Delphix Engine

1. Login to the Delphix Engine.

After you are logged in, follow the instructions to **Provision an Oracle VDB**.

Once you have completed provisioning an oracle VDB in the Delphix Engine, you will need to complete the following activities in the DMSuite, another user interface to aid in agile masking. The five primary tasks to be completed include adding an Oracle Application in DMSuite, adding an Oracle Environment, adding a connector to the newly provisioned VDB in the Delphix Engine, defining the rules and profiles for masking and finally creating the masking job.

Activity Two: Mask Data in Agile Masking Interface

**Step 1 - Login to DMSuite**

1. Login to http://<hostname>:8282/dmsuite

2. Enter default Username: Axistech

3. Enter default User password: Axis_123

Once logged into DMSuite, a user will navigate to three areas inside the DMSuite interface to complete activities needed for Agile Masking. These pages can be found under the Environments Tab as seen below:
Step 2 Add and Create an Application and Environment

The first task you need to perform is to create or import an application environment. Environments define the scope of work in DMsuite. In order to mask or provision databases and files within DMsuite, you first need to create an environment in which DMsuite will store the connection information and masking and provisioning rules for those data stores. An environment can contain multiple database connections and multiple file connections. For a step by step procedure, go to Add and Create an Application and Environment

Step 3 Add and Create a Connector

After you create an environment, you need to create connections, as seen below. Go to Add a Connector to the VDB and follow the step by step procedure.

After you create an environment and connectors, you need to define a rule set.

Step 4 - Add, configure and Define what to mask

1. Add and Configure what to mask:

2. Navigate to Create a Rule Set as seen below:
3. For a step by step procedure go to Create and Define a Rule Set.

Afterwards, configure what you want to mask with profile data.

After selecting tables (connectors) and defining a rule set, which you did in the preceding sections, you create a profiling job. (When you defined a rule set, an empty inventory was automatically created.) A profiling job updates or populates an inventory of your data and identifies potential sensitive data elements based on the profiling set (rule sets) you use. When you examine your inventory, you might determine that you only want to select certain columns to mask, to finalize your inventory.

In the Environment Overview screen, locate the jobs icons to create a new profile or to create a masking job as seen below.
4. For a step by step procedure to create a new profile, go to **Profile Data**

**Step 5 - Mask your Data**

1. For a step by step procedure go to **Mask (Use In Place Option)**
2. **Choose appropriate rule set settings**

3. **Run and Confirm the Masking Job**

Activity Three - Refresh Data from the Source in Delphix Engine

After a VDB is initially masked, a user may want the latest data from the source. Since the masking job has already been defined, Delphix allows the user to invoke this masking job every time a VDB is refreshed. This can be done by calling the masking job via a post clone hook. Now every time the VDB is refreshed, the masking job will be invoked before making it available for users. Paste the following link as a post clone hook for a VDB to execute masking job.
Create or Import an Environment

The first task you need to perform is to create or import an application environment. *Environments* define the scope of work in Delphix Agile Data Masking. In order to mask or provision databases and files within Delphix Agile Data Masking, you first need to create an environment in which Delphix Agile Data Masking will store the connection information and masking and provisioning rules for those data stores. An environment can contain multiple database connections and multiple file connections.

Creating a New Environment

1. Click **Add Environment** at the upper right of the screen.
   
The screen prompts you for the following items:
   
   - **Application Name**: The name of the application associated with this environment, for informational purposes. (An integrated test environment can have multiple applications.)
   - **Environment Name**: The display name of the new environment.
   - **Purpose**: The way the environment is used in the development process: Development, Gold Copy, QA, Training, and so on.

2. Either click **Save** to return to the Environments List/Summary screen, or click **Save & View** to display the Environment Overview screen (see "The Environment Overview Screen").

Importing an Environment

1. Click **Import Environment** at the upper right of the screen.
   
The screen prompts you for the following items:
   
   - **Application Name**: The name of the application associated with this environment, for informational purposes. (An integrated test environment can have multiple applications.)
   - **Environment Name**: The name of the environment that you want to import.
   - **Purpose**: The way the environment is used in the development process: Development, Gold Copy, QA, Training, and so on.
   - **Select...**: Use to browse for the XML file that contains the information you want to import. (This file must be a previously exported Delphix Agile Data Masking environment.)
   - Either click **Save** to return to the Environments List/Summary screen, or click **Save & View** to display the Environment Overview screen (see "The Environment Overview Screen").
Create and Define Connections

After you create an environment, you need to create connections.

- With Delphix Agile Data Masking, the connections you define in Delphix Agile Data Masking must match connections you have defined in Informatica PowerCenter. So, each connector you define in Delphix Agile Data Masking must have a connection with the same name already defined in PowerCenter.

To define connectors:

1. Click an Environment name.
2. From the Environment Overview screen, click the Connector tab. The Connector List Screen appears.
3. Click Create Connection to the upper right of the Connectors tab. The Create Connection Window appears, prompting you for connection information for the database or file:
   - Type—Select the appropriate database type (for example, Database - Oracle) or file type (for example, File - Delimited).
   - For descriptions of the database connector fields, see "Database Connectors."
   - For descriptions of the file connector fields, see "File Connectors."
   - For descriptions of the mainframe connector fields, see "Mainframe Connectors."
4. Enter the required information and click Save. You are returned to the Connectors screen where you may add additional connectors if necessary.

For detailed information about connections, see "Managing Connectors" on page 1.
Create and Define a Rule Set

After you create an environment and connectors, you need to define a rule set.

**Defining a Rule Set for a Database or File**

1. Click on the name of an Environment, and then click the **Rule Set** tab.

   The Rule Set screen appears.

2. Click **Create Rule Set** to the upper right of the Rule Set screen.

   The Create Rule Set screen appears (Figure 1.5 on next page). This screen lets you specify which tables belong in the Rule Set.
   
   a. Enter a **Name** for your Rule Set.
   
   b. Select a **Connector** name from the dropdown.
   
   c. The list of tables/files for that connector appears.
   
   d. Click individual tables/file names in the list to the right to select them, or click **Select All** in the bottom left to select all the tables. Click **Save**.
   
   e. You are returned to the Rule Set screen.

3. When you are finished, click **Save**.

4. You may then need to define the Rule Set by modifying the table settings as described in "Managing Rule Sets".

For example:

- For a table, you may want to filter data from the table.
- For a file, you must select a File Format to use.
Profile Data and Define Inventory

- Before performing any profiling, be sure that your administrator has set up the site profiles according to *DMsuite Administrator's Guide*.

The purpose of profiling data is to determine what data you want to mask by updating an inventory of your data with sensitive data elements identified. You can then review and edit that inventory.

If you are using a source file, instead of a source database, you have the choice to populate your inventory manually or use the Delphix Agile Data Masking Profiling module to profile your data. Because profiling assigns algorithm information to fields or columns, you might prefer not to profile your data if you would rather manually assign these properties. If you choose not to profile your data, skip this section.

You can mask data on-the-fly or you can provision the data (using Delphix Agile Data Masking or other tools) and then mask it (in-place masking). For a mainframe environment, code is generated for provisioning and masking on the mainframe system. All activity prior to code generation is performed in DMsuite; after code generation, all activity occurs on the mainframe system, outside of DMsuite. DMsuite Premier Edition generates XML that is automatically imported into your PowerCenter repository via your Informatica Client (via the PMCMD interface). Whichever form of masking you choose, you need to profile your data first.

After selecting tables (connectors) and defining a rule set, which you did in the preceding sections, you create a profiling job. (When you defined a rule set, an empty inventory was automatically created.) A profiling job updates or populates an inventory of your data and identifies potential sensitive data elements based on the profiling set (rule sets) you use. (For information about the inventory, see "Managing Masking Inventory"). When you examine your inventory, you might determine that you only want to select certain columns to mask, to finalize your inventory.

**Profiling Data**

1. Create a profiling job as described in "Creating a New Profiling Job."
2. Run the profiling job you just created. When you run this profiling job, it updates/populates an Inventory.
3. To view the inventory, click the Inventory tab while in an Environment Overview.
4. Examine the inventory to ensure that the profiling job has included everything you want to mask. (For example, if you selected a First Name field, you probably want the Last Name field as well.) You can see which columns were selected for masking by selecting the associated rule set. Make sure you have included all sensitive data elements (for example, personal identifying information) from the table that you want to mask.
5. Modify the inventory, if necessary.

When a profiling job runs, it automatically updates the inventory for the given rule set. If you do not want the Profiler to automatically update the inventory, change the ID Method to User. For detailed information about profiling data, see "About Profiling Data."

What you do next depends on how you plan to mask your data. If you plan to mask your data in-place and you want to use Delphix Agile Data Masking to provision your data, proceed with "Provision Data" next. If you plan to mask data on-the-fly, or have already provisioned your data outside of Delphix Agile Data Masking, continue with "Mask Data."
Mask Data

You can mask data on-the-fly or you can provision it first and then mask it (in-place masking). If you plan to use Delphix Agile Data Masking to provision your data before masking it, you must first follow the procedure in the preceding section, "Provision Data".

Masking Your Data

1. Create a Masking job, as described in "Creating a New Masking Job."

2. Run the Masking job.
   If you are using Informatica PowerCenter, this step generates XML that is automatically imported into your PowerCenter repository via your Informatica Client (via the PMCMD interface). The XML files create your mappings and your workflow (how you apply the mappings). If you chose Import as your Mask Method (rather than Import and Run), you need to log in to PowerCenter to execute the workflow before you continue with this procedure.

3. Confirm that you have masked data/generated the masking job:
   a. Examine the data in the database columns or the fields in the file to determine whether the data is masked.
   b. Run the application that uses the data to ensure that you do not see any unmasked data and that there are no issues in running the application.

4. For detailed information about masking data, see "About Masking Data."
Basic Tasks for a Database File

Whether you only plan to profile, provision and mask data, or just mask data, you need to perform certain tasks. Databases and files are dealt with in a similar but slightly different way.

You need to perform the following tasks if you are working with a source. The sections following this list describe the tasks.

1. Create or Import an Environment
2. Define Connections
3. Define a Rule Set
4. Profile Data and Define Inventory
5. Provision Data (Databases only)
6. Mask Data
7. Certify Data (Databases only)
Certify Data

After completing the other tasks and masking your data, you should create a job to certify your data on an ongoing basis. This alerts you if unmasked data is introduced to a masked database.

- You need to certify your data on a regular basis.

Certifying Data

1. Create a Certification job, as described in "Creating a New Certify Job" on page .
2. Run the Certify job.
3. Confirm that no unmasked data has been introduced.

For detailed information about certifying data, see "About Certifying Data."
Provision Data

If you are using a source file, instead of a source database, you do NOT provision your data. Skip this section.

Before you can provision or subset data, you must first create your source environment in Delphix Agile Data Masking (Create or Import an Environment), define connections (Define Connections), define a rule set (Define a Rule Set), and create a target environment (Create or Import an Environment).

Provisioning Data

1. Create a provisioning job as described in "Creating a New Provisioning Job."

2. Run the provisioning job.

3. Ensure you have the information you need.

For detailed information about provisioning data, see "About Provisioning (Subsetting Data)."
Using Delphix Agile Data Masking

The following sections detail the various tabs and functions in Delphix Agile Data Masking, presented in the order in which they appear in the Delphix Agile Data Masking application:

- Managing Environments with Agile Data Masking
- Managing Connectors
- Managing Rule Sets
- Managing Masking Inventory
- Managing Jobs
Admin Tab

**Note:** For more detailed Admin information, see the Delphix Agile Data Masking Administrator's Guide. Click the **Admin** tab at the top of the screen for administrator settings and information.
About

From the Admin tab, if you click **About** to the left, the list of information about your current Delphix Agile Data Masking installation is shown:

- Delphix Agile Data Masking Version
- Operating System
- Application Server
- Database
- Masking
- Java Version
- Expiration Date
- Licensed Data Sources
Users

From the Admin tab, if you click Users to the left, the list of users will be displayed. From here users can be added, edited, or deleted.

Along with regular user information (name, username, email etc.), users have permissions. These permissions can be set by the User's role and what environments they can access.
Managing Connectors

Delphix Agile Data Masking stores database connection information in an object called a **Connector**. When in an Environment Overview, click the **Connector** tab to view the list of connectors within an environment.

- For each Connection, in Informatica PowerCenter you must manually define a corresponding connector with the same name.
Creating or Editing a Connector

Editing a Connector

1. Click the **Edit** icon to the right of the connector.

Creating a New Connector

1. Click **Create Connection** to the upper right of the Connectors Editor. The Create Connection window appears, prompting you for connection information for a database or file:

2. **DB Type**—Select the appropriate database type (for example, *Database - Oracle*) or file type (for example, *File - Delimited*).
   
   For database connectors, see "Database Connectors."
   
   For file connectors, see "File Connectors."
   
   For mainframe connectors, see Mainframe C.

3. Each new connector uses the source system access credentials so appropriate access is granted when reading source data.
Database Connectors

The fields that appear are specific to the DBMS Type you select. If you need assistance determining these values, please contact your database administrator. All required fields are marked with an asterisk on the screen.

- You only can create connectors for the databases and/or files listed. If your database or file type is not listed here, you cannot create a connector for it.
- **Kerberos Authentication**—(Sybase, Oracle, or DB2 only, optional) Whether to use a Kerberos connection to the database. This box is clear by default. If this box is checked, the application code makes a Kerberos connection to the database instead of using a login/password.
- **Connection Type**—(Oracle or MS SQL Server only) Choose a connection type:
  - **Basic**—Basic connection information.
  - **Advanced**—The full JDBC connect string.
- **Connection Name**—The name of the database connector (specific for your Delphix Agile Data Masking application).
- For each Connection Name, in Informatica PowerCenter you must manually define a corresponding connector with the same name.
- **Schema Name**—The schema that contains the tables that this connector will access.
- **Database Name**—The name of the database to which you are connecting.
- **Host Name / IP or Hostname/IP**—The network host name or IP address of the database server.
- **Username**—(Oracle only)
- **ODBC DNS Name**—(ODBC and Microsoft Access only)
- **Login ID**—The user login this connector will use to connect to the database.
- **Password**—The password associated with the Login ID or Username. (This password is stored encrypted.)
- **System Number**—(SAP only)
- **SAP Client**—(SAP only)
- **Language**—(SAP only)
- **Port**—The TCP port of the server.
- **SID**—(Oracle only) Oracle System ID (SID).
- **Instance Name**—(MS SQL Server only) The name of the instance
- **Server Name**—(Informix only) The name of the Informix server.
- **Custom Driver Name**—(Adabas and SQL Anywhere only) The name of the custom driver.
- **Custom JDBC URL**—(Adabas and SQL Anywhere only) The name of the custom JDBC URL.

All database types have a **Test Connection** button at the bottom left of the New Connector window. We highly recommend that you test your connection before you save it. Do so before you leave this window. When you click the **Test Connection** button, Delphix Agile Data Masking uses the information in the form to attempt a database connection. When finished, a status message appears indicating success or failure.
Deleting Connectors

Procedure

1. Click the Delete icon to the far right of the connector name.
2. When you delete a connector, you also delete its rule sets and inventory data.
File Connectors

The values that appear correlate to the File Type you select. All required fields are marked with an asterisk on the screen.

- **Connector Name**—The name of the file connector (specific to your Delphix Agile Data Masking application and unrelated to the file itself).
- **Connection Mode**—Local Files, SFTP, FTP, HTTP & HTTPS.
- **Path**—The path to the directory where the file(s) are located.
- **Operating System**—Choose the operating system on which the file resides: Windows or Linux. (This value does not appear for Mainframe Copybooks.)

If you select SFTP or FTP for **Connection Mode**, the following additional values appear:

- **Server Name**—The name of the server used to connect to the file.
- **User Name**—The User Name to connect to the server.
- **Public Key Authentication**—(Optional) (Only appears for SFTP.) Check this box to specify a public key.

When you check this box, the Available Keys dropdown appears. Choose a key from the dropdown. (The path on the server to the location that contains the keys is configured in a Delphix Agile Data Masking property files.)

- **Password**—The associated Password for the server.
- **Port**—The Port used to connect to the server.
Mainframe Connectors

The fields that appear are correlate to the File Type you select. If you need assistance determining these values, please contact your MVS administrator. All required fields are marked with an asterisk on the screen.

- **Connection Name**—The name of the file connector (specific to your DMsuite application and unrelated to the file itself).
- **Host Name / IP**—The network host name or IP address of the PDS server.
- **FileName**—The source file fully qualified data set name, including "(0)" for generation data group files.
- **FileType**—The source file type: normal, VSAM, or GDG.
- **UserID**—The user login this connector will use to connect to the mainframe host system to access the PDS copybook files. For VSAM files, use the cluster name.
- **Password**—The password associated with the Login ID or Username. (This password is stored encrypted.)
- **File DCB RECFM**—The source file record format; possible values: F, FB, FBA, V, VB, VBA.
- **BLKSIZE**—The source file block size, from 1 to 32760.
- **File DCB LRECL**—The source file logical record length, from 1 to 32760. If record format is fixed, must be a divisor of block size.
- **Header/Trailer Code**—The number of records to skip (copy and not mask) at the beginning and end of the source file. The format is in the form "H,x,T,y", "H,x", or "T,y" where:
  - H and T are constants.
  - x is the number of header rows to skip.
  - y is the number of trailer rows to skip.

Normal file types have a **Test Connection** button at the bottom left of the New Connector window. VSAM and GDG file types do not have a **Test Connection** button.

We highly recommend that you test your connection. Do so before you leave this window. When you click the **Test Connection** button, Delphix Agile Data Masking uses the information in the form to attempt a mainframe connection. When finished, a status message appears indicating whether the attempt was successful or failed.

- VSAM files are treated sequentially as ordinary files. IDCAMS uses the cluster name to create an ordinary file. This GOLDCOPY is used to load the target dataset later.

To allocate the new file, the process uses record format, record length, and block size, as follows:

- If the VSAM source record is fixed length, use FB and record length, and make block size the largest multiple of record length less than 27,998 (half-track blocking).
- If the VSAM record is variable length, use VB, and use the maximum record length +4 as the record length, and use block size of 27,990.

This information pertains to processing the source records and allocating the GOLDCOPY; it does not apply to system records about the original source.
The Connector List

The Connector List Screen shows the connectors that have been created within the environment.
Managing Environments with Agile Data Masking

Environments define the scope of work in Delphix Agile Data Masking. In order to mask or provision databases and files within Delphix Agile Data Masking, you first need to create an Environment in which Delphix Agile Data Masking will store the connection information and masking and provisioning rules for those data stores. An environment can contain multiple database connections and multiple file connections.
Copying and Deleting Environments

You can copy and delete environments from the Environment List/Summary screen.

Copying Environments

The copy feature is a very powerful tool. If you have multiple similar environments, you can create one, copy it multiple times, and make necessary changes without having to start from scratch. Each environment will need its own connections, but you can update the connections that get copied. However, you will need to create new jobs for each copied environment.

When you copy an environment, all of its connectors, rule sets, and inventory are copied with it. To avoid confusion when using copied environments, we recommend that you change descriptive information to be different from the original.

1. Click the Copy icon to the right of the environment status.
   The popup prompts you for the following items: Name, Purpose, and Application Name
2. Click Save.
   The environment definition is copied to a new name.

Deleting an Environment

- Click the Delete icon to the right of the environment status and copy icon.
- Clicking the Delete icon deletes *everything* for that environment: connections, inventory, rule sets, and so on.
Creating or Importing an Environment

Creating a New Environment

1. Click **Add an Environment** at the upper right of the screen. The screen prompts you for the following items:
   a. **Application Name**: The name of the application associated with this environment, for informational purposes. (An integrated test environment can have multiple applications.)
   b. **Environment Name**: The display name of the new environment.
   c. **Purpose**: The way the environment is used in the development process: Development, Gold Copy, QA, Training, and so on.

2. Either click **Save** to return to the Environments List/Summary screen, or click **Save & View** to display the Environment Overview screen (see "The Environment Overview Screen").

Importing an Environment

1. Click **Import an Environment** at the upper right of the screen. The screen prompts you for the following items:
   a. **Application Name**: The name of the application associated with this environment, for informational purposes. (An integrated test environment can have multiple applications.)
   b. **Environment Name**: The name of the environment that you want to import.
   c. **Purpose**: The way the environment is used in the development process: Development, Gold Copy, QA, Training, and so on.
   d. **Select...**: Use to browse for the XML file that contains the information you want to import. (This file must be a previously exported Delphix Agile Data Masking environment.)

2. Either click **Save** to return to the Environments List/Summary screen, or click **Save & View** to display the Environment Overview screen (see "The Environment Overview Screen").
**Exporting an Environment**

You can export an environment from the Environment List/Summary screen. You can later import that environment to a different instance of Delphix Agile Data Masking, such as a development test instance or a production instance.

1. Click the **Export** icon.
   The popup fills in the following items: Environment Name and File Name.

2. Click **Export**.
   All the information for the specified environment (connectors, rule sets, inventory, jobs, and so on) is exported to an XML file.

3. A status popup appears. When the export operation is complete, you can click on the **Download file** name to access the XML file.
The Environment List/Summary Screen

The Environment List/Summary screen provides a dashboard overview of what Delphix Agile Data Masking is doing. It is the first screen that appears when a user logs in to Delphix Agile Data Masking. You can also reach it by clicking the Environments tab or by clicking the Environment List link elsewhere in the application.

To search for an environment, enter an environment name in the Search field and click Search.

The Environments List/Summary screen contains the following columns:

- **Application**—The name of the application.
- **Environment**—The name of the environment.
- **Purpose**—The purpose of this environment.
- **Jobs Running**—The number of running jobs.
- **Export**—Export the environment.
- **Copy**—Copy the environment.
- **Delete**—Delete the environment.
The Environment Overview Screen

You can reach the Environment Overview screen by clicking an environment name from the Environments List/Summary screen.

This screen gives an overview of the Environment and the Environment Status. The left of the screen displays the environment Name, Purpose (for example, DEV or QA), and the Application Name. The Environment Status lists the Current Status, and dates for Last Data Refresh, Last Masked, Last Certified, and Last Profiled.

The files listed on the right side of the window are pdfs of the last certification job (C) and the last masking job inventory (M).

The body of the page displays all jobs currently defined for this environment, along with the status of the jobs (created, running, succeeded, or failed). For information about Jobs and the icons on this screen, see "Managing Jobs."

You can use the icons in the Jobs heading to create new jobs. See "Monitor Jobs."
Managing Jobs

Delphix Agile Data Masking creates jobs to profile, provision, and mask, and certify data.
Creating a New Certify Job

1. Select **Certify**.
   The Create Certify Job window appears.

You will be prompted for the following information:

- **Job Name**—A free-form name for the job you are creating. Must be unique.
- **Multi Tenant**—Check box if the job is for a multi-tenant database.
- **Rule Set**—Select a rule set that this job will execute against.
- **Generator**—The default value is **Delphix Agile Data Masking**.
- **No. of Streams**—The number of parallel streams to use when running the jobs. For example, you can select two streams to run two tables in the ruleset concurrently in the job instead of one table at a time.
- **Remote Server**—(optional) The remote server that will execute the jobs. This option lets you choose to execute jobs on a remote server, rather than on the local Delphix Agile Data Masking instance.
  Note: This is an add-on feature for Delphix Agile Data Masking Standard Edition. (This option only appears if you select DMsuite as the Generator.)
- **Min Memory (MB)**—(optional) Minimum amount of memory to allocate for the job, in megabytes.
- **Max Memory (MB)**—(optional) Maximum amount of memory to allocate for the job, in megabytes.
- **Feedback Size**—(optional) The number of rows to process before writing a message to the logs. Set this parameter to the appropriate level of detail required for monitoring your job. For example, if you set this number significantly higher than the actual number of rows in a job, the progress for that job will only show 0 or 100%.
- **Prescript**—(optional) Specify the full pathname of a file that contains SQL statements to be run before the job starts, or click **Browse** to specify a file. If you are editing the job and a prescript file is already specified, you can click the **Delete** button to remove the file. (The Delete button only appears if a prescript file was already specified.) For information about creating your own prescript files, see "Creating SQL Statements to Run Before and After Jobs" on page .
- **Postscript**—(optional) Specify the full pathname of a file that contains SQL statements to be run after the job runs, or click **Browse** to specify a file. If you are editing the job and a postscript file is already specified, you can click the **Delete** button to remove the file. (The Delete button only appears if a postscript file was already specified.) For information about creating your own postscript files, see "Creating SQL Statements to Run Before and After Jobs" on page .
- **Delta Masking**—Check this box to mask rows that are not masked and are identified as part of the certification process as polluted.
- **Batch Update**—(optional) Enable or disable use of a batch for updates. A job's statements can either be executed individually, or can be put in a batch file and executed at once, which is faster.
- **Disable Trigger**—(optional) Whether to automatically disable database triggers. The default is for this check box to be clear and therefore not perform automatic disabling of triggers.
- **Comments**—(optional) Add comments related to this certification job.
- **Email**—(optional) Add e-mail address(es) to which to send status messages.

2. When you are finished, click **Save**.

For information about running a job, see "Running and Stopping Jobs from the Environment Overview Screen."
Creating a New Masking Job

1. Select **Mask**.
   The Create Masking Job window appears.
   You will be prompted for the following information:
   - **Job Name**—A free-form name for the job you are creating. Must be unique across the entire application.
   - **Masking Method**—Select either **In-Place** or **On-The-Fly**. For more information on masking type, see "Mask Data".
   - **Multi Tenant**—Check box if the job is for a multi-tenant database.
   - **Rule Set**—Select a rule set that this job will execute against.
   - If you choose **Informatica**, the screen changes:
     - **Generator**—The default value is **Delphix Agile Data Masking**.
     - **Informatica Repository Folder name**—The folder name in the Informatica repository where the objects should be imported. (This option only appears if you select Informatica as the Generator.)
     - **Parameter File Path**—(optional) If checked, this tells DMsuite to configure the Informatica sessions and workflows to use a parameter file that contains the source and target connection information. If unchecked, DMsuite will generate sessions/workflows that use the connector names as defined within DMsuite, which will require connections with the same names defined within the Informatica repository. (This option only appears if you select Informatica as the Generator.)
     - **Import Mapplet**—(optional) if checked, this tell DMsuite to import mapplets that are assigned to columns in the inventory along with the mappings/sessions/workflows. If unchecked, DMsuite will not attempt to import any mapplets that are assigned in the inventory. (This option only appears if you select Informatica as the Generator.)
   - **Mask Method**—(Only appears if you choose Informatica as the Generator.) Choose either of the following:
     - **Import**—When you click the Run icon, creates the mappings in Informatica but does not execute the workflow. You later run the job from the Informatica client.
     - **Import and Run**—When you click the Run icon, creates the mappings in Informatica and executes the workflow.
   - **No. of Streams**—The number of parallel streams to use when running the jobs. For example, you can select two streams to run two tables in the Rule Set concurrently in the job instead of one table at a time.
   - **Remote Server**—(optional) The remote server that will execute the jobs. This option lets you choose to execute jobs on a remote server, rather than on the local Delphix Agile Data Masking instance. Note: This is an add-on feature for Delphix Agile Data Masking Standard Edition. (This option only appears if you select DMsuite as the Generator.)
   - **Min Memory (MB)**—(optional) Minimum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
   - **Max Memory (MB)**—(optional) Maximum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
   - **Update Threads**—The number of update threads to run in parallel to update the target database. (This option only appears if you select Delphix Agile Data Masking as the Generator and should only be used if the Batch option is not checked.)
   - For databases using T-SQL, multiple update/insert threads can cause deadlocks. If you see this type of error, reduce the number of threads that you specify in this box.
   - **Commit Size**—(optional) The number of rows to process before issuing a commit to the database.
   - **Feedback Size**—(optional) The number of rows to process before writing a message to the logs. Set
this parameter to the appropriate level of detail required for monitoring your job. For example, if you set this number significantly higher than the actual number of rows in a job, the progress for that job will only show 0 or 100%.

- **Bulk Data**—(optional) For In-Place masking only. The default is for this check box to be clear. If you are masking very large tables in-place and require performance improvements, check this box. Delphix Agile Data Masking will mask data to a flat file, and then use inserts instead of updates to bulk load the target table. (This option only appears if you select DMsuite as the Generator.)

- **Disable Constraint**—(optional) Whether to automatically disable database constraints. The default is for this check box to be clear and therefore not perform automatic disabling of constraints. For more information about database constraints, see “Enabling and Disabling Database Constraints” on page.

- **Batch Update**—(optional) Enable or disable use of a batch for updates. A job's statements can either be executed individually, or can be put in a batch file and executed at once, which is faster.

- **Disable Trigger**—(optional) Whether to automatically disable database triggers. The default is for this check box to be clear and therefore not perform automatic disabling of triggers.

- **Prescript**—(optional) Specify the full pathname of a file that contains SQL statements to be run before the job starts, or click Browse to specify a file. If you are editing the job and a prescript file is already specified, you can click the Delete button to remove the file. (The Delete button only appears if a prescript file was already specified.) For information about creating your own prescript files, see “Creating SQL Statements to Run Before and After Jobs.”

- **Postscript**—(optional) Specify the full pathname of a file that contains SQL statements to be run after the job finishes, or click Browse to specify a file. If you are editing the job and a postscript file is already specified, you can click the Delete button to remove the file. (The Delete button only appears if a postscript file was already specified.) For information about creating your own postscript files, see “Creating SQL Statements to Run Before and After Jobs.”

- **Comments**—(optional) Add comments related to this masking job.

- **Email**—(optional) Add e-mail address(es) to which to send status messages.

2. When you are finished, click Save.

For information about running jobs, see “Running and Stopping Jobs from the Environment Overview Screen.”
Creating a New Profiling Job

- You can create Profiling jobs for databases, copybooks, delimited files, fixed-width, and Excel files.

Creating a New Profiling Job

1. Click the Profile button.

   A Profiling job for a mainframe system cannot assign groups because it does not have the heuristics needed to determine sensitive elements per group. The Profiler assigns group-sensitive elements to a single group. Then, in inventory, groups are updated as needed to establish the sensitive element field group sets. The Create Profiling Job window appears. You will be prompted for the following information:

   - **Job Name**—A free-form name for the job you are creating. Must be unique.
   - **Multi Tenant**—Check box if the job is for a multi-tenant database.
   - **Rule Set**—Select a rule set that this job will execute against.
   - **Generator**—The default value is Delphix Agile Data Masking.
   - **No. of Streams**—The number of parallel streams to use when running the jobs. For example, you can select two streams to run two tables in the ruleset concurrently in the job instead of one table at a time.
   - **Remote Server**—(optional) The remote server that will execute the jobs. This option lets you choose to execute jobs on a remote server, rather than on the local Delphix Agile Data Masking instance. Note: This is an add-on feature for Delphix Agile Data Masking Standard Edition. (This option only appears if you select DMsuite as the Generator.)
   - **Min Memory (MB)**—(optional) Minimum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
   - **Max Memory (MB)**—(optional) Maximum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
   - **Feedback Size**—(optional) The number of rows to process before writing a message to the logs. Set this parameter to the appropriate level of detail required for monitoring your job. For example, if you set this number significantly higher than the actual number of rows in a job, the progress for that job will only show 0 or 100%.
   - **Profile Sets**—(Optional) The name of a profiler set, which is a subset of expressions (for example, a subset of financial expressions). (See Delphix Agile Data Masking Administrator's Guide.)

   **Note:** If you do not select a profile set, Delphix Agile Data Masking will use all defined expressions instead of just a subset.

   - **Comments**—(optional) Add comments related to this job.
   - **Email**—(optional) Add e-mail address(es) to which to send status messages. Separate addresses with a comma (,).

2. When you are finished, click **Save**.

For information about running jobs, see “Running and Stopping Jobs from the Environment Overview Screen.”
Creating a New Provisioning Job

- When you provision data, you define the provisioning job within the source environment (from source to target). On the other hand, a masking job is defined from the target environment.

Creating a New Provisioning Job

1. Select **Provision**.
   The Create Provisioning Job window appears.
   You will be prompted for the following information:

   - **Job Name**—A free-form name for the job you are creating.
   - **Target Environment**—The environment into which the data will be loaded.
   - **Multi Tenant**—Check box if the job is for a multi-tenant database.
   - **Rule Set**—Select a rule set that this job will execute against.
   - **Target Connector**—The database connector into which the data will be loaded.
   - **Generator**—Choose **DMsuite** or **Informatica**.
     - If you choose **Informatica**, the screen changes:
       - **Informatica Generator**—The default value is **Delphix Agile Data Masking**.
       - **Informatica Repository Folder name**—The folder name in the Informatica repository where the objects should be imported. (This option only appears if you select Informatica as the Generator.)
       - **No. of Streams**—The number of parallel streams to use when running the jobs. For example, you can select two streams to run two tables in the ruleset concurrently in the job instead of one table at a time. (This option only appears if you select DMsuite as the Generator.)
       - **Remote Server**—(optional) The remote server that will execute the jobs. This option lets you choose to execute jobs on a remote server, rather than on the local Delphix Agile Data Masking instance. Note: This is an add-on feature for Delphix Agile Data Masking Standard Edition. (This option only appears if you select DMsuite as the Generator.)
       - **Min Memory (MB)**—(optional) Minimum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
       - **Max Memory (MB)**—(optional) Maximum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
       - **Commit Size**—(optional) The number of rows to process before issuing a commit to the database.
       - **Feedback Size**—(optional) The number of rows to process before writing a message to the logs. Set this parameter to the appropriate level of detail required for monitoring your job. For example, if you set this number significantly higher than the actual number of rows in a job, the progress for that job will only show 0 or 100%.
   - **Disable Constraint**—(optional) Whether to automatically disable database constraints. The default is for this check box to be clear and therefore not perform automatic disabling of constraints. For more information about database constraints, see “Enabling and Disabling Database Constraints” on page .
   - **Batch Update**—(optional) Enable or disable use of a batch for updates. A job's statements can either be executed individually, or can be put in a batch file and executed at once, which is faster.
   - **Truncate**—(optional) Whether to truncate target tables before loading them with data. If this box is selected, the tables will be "cleared" before the operation. If this box is clear, data is appended to tables, which potentially can cause primary key violations. This box is clear by default.
   - **Disable Trigger**—(optional) Whether to automatically disable database triggers. The default is for this check box to be clear and therefore not perform automatic disabling of triggers.
   - **Prescript**—(optional) Specify the full pathname of a file containing SQL statements to be run before the job starts, or click **Browse** to specify a file. If you are editing the job and a prescript file is already specified, you can click the **Delete** button to remove the file. (The Delete button only appears if a
prescript file was already specified.) For information about creating your own prescript files, see "Creating SQL Statements to Run Before and After Jobs" on page.

- **Postscript**—(optional) Specify the full pathname of a file containing SQL statements to be run after the job finishes, or click **Browse** to specify a file. If you are editing the job and a postscript file is already specified, you can click the **Delete** button to remove the file. (The Delete button only appears if a postscript file was already specified.) For information about creating your own postscript files, see "Creating SQL Statements to Run Before and After Jobs" on page.

- **Comments**—(optional) Add comments related to this provisioning job.
- **Email**—(optional) Add e-mail address(es) to which to send status messages.

2. When you are finished, click **Save**.
Creating New Jobs

In the Environment Overview screen, select one of the jobs icons to create the corresponding job.
Jobs on the Environment Overview Screen

The Environment Overview screen provides status of any jobs running or previously run within the environment. When you create jobs for an environment, you must create them from the Environment Overview screen of the environment they mask (or the environment they use as a source for data provisioning).

The following columns appear in the Jobs section of the screen:

- Name
- Rule Set
- Status
- Action
- Edit
- Delete
Running and Stopping Jobs from the Environment Overview Screen

Running or Re-Running a Job from the Environment Overview Screen
1. Click the **Run** icon (play icon) in the Action column for the desired job.
   
   If you are using Informatica as the Generator and chose Import as your Mask Method (rather than Import and Run), Delphix Agile Data Masking just creates the mappings in Informatica. You need to log in to PowerCenter to actually execute the workflow.

   The Run icon changes to a Stop icon while the job is running.

   When the job is complete the Status changes.

Stopping a Running Job from the Environment Overview Screen
1. Click the **Stop** icon in the Action column for the running job.
   
   A popup appears asking **Are you sure you want to stop job?**

   2. Click **OK**.

   When complete the Status changes.
Managing Masking Inventory

Delphix Agile Data Masking stores the masking rules for each sensitive column in the Delphix Agile Data Masking repository database in the environment's inventory.
Importing and Exporting Inventory

- The format of an imported.csv file must exactly match the format of the exported inventory. If you plan to import an inventory, before importing the inventory, you should export it and then update the exported file as needed before you import it.

Exporting an Inventory

1. Click the Export icon at the upper right.
   The Export Inventory popup appears with the name of the currently selected Rule Set as the Inventory Name and a corresponding .csv File Name.

2. Click Save.
   A status popup appears. When the export operation is complete, you can click on the Download file name to access the inventory file.

Importing an Inventory

1. Click the Import icon at the upper right.
   The Import Inventory popup appears.

2. Click Select to browse for the name of a comma-separated (.csv) file.

3. Click Save.
   The inventory you imported appears in the Rule Set list for this environment.
Inventory Settings

Specifying Inventory Settings

1. On the left of the screen, select a Rule Set from the dropdown. Below this, Contents lists all the tables or files defined for the rule set.

2. Select a table or file for which you would like to create or edit the inventory of sensitive data. The Columns or Fields for that specific table or file appear.
   - If a column is a primary key (PK) or an index (IDX), an icon indicating this will appear to the left of the column name. If there is a Note for the column, a Note icon will appear. To read the note, click the icon.
   - If a table, metadata for the column appears: Data Type and Length (in parentheses). This information is read-only.

3. Choose how you would like to view the inventory:
   - All Fields—Displays all columns in the table or all fields in the file (allowing you to mark new columns or fields to be masked).
   - Masked Fields—Filters the list to just those columns or fields that are already marked for masking.

4. Choose how to determine whether to mask/unmask a column:
   - Auto—The default value. The profiling job can determine or update the algorithm assigned to a column and whether to mask the column.
   - User—The user’s choice overrides the profiling job. The user manually updates the algorithm assignment, mask/unmask option of the column. The Profiler will ignore the column, so it will not be updated as part of the Profiling job.
Managing a Database Inventory

The following sections apply to databases:

- Setting Column Criteria for a Table
- Row Type and Creating New Row Types for Tables

Setting Column Criteria for a Table

Note: a database must be selected from the Select Rule Set dropdown menu on the left, not a file system.

1. Click the green edit icon to the right of a name of a column.
2. To mask the selected column, check the Mask check box.
   Clear this check box if you do not want to mask this column.
3. Choose the appropriate sensitive data element type for the column from the Domain dropdown.
   Delphix Agile Data Masking defaults to a Masking Algorithm as specified in the Settings screen. If necessary, you may override the default algorithm for a column.
4. To select a different masking algorithm, choose one from the Algorithm drop down.

If you are using Informatica PowerCenter, you can create Mapplets to use in Delphix Agile Data Masking or use the mapplet-based algorithms that are provided with Delphix Agile Data Masking for the Informatica Generator. For information about how to use these Mapplets, see DMsuite Administrator's Guide.

- When selecting masking algorithms for your domains, be sure not to select BOTH algorithms created in DMsuite and mapplet-based algorithms generated with PowerCenter. When you later create your masking job, the Generator you specify (DMsuite or Informatica) determines which class of algorithm to use. The masking job cannot combine the DMsuite and Informatica PowerCenter algorithm classes, because the job will fail.
- For detailed descriptions of these algorithms, see "Algorithms." For information about how to create your own algorithms and how to add them to domains, see Delphix Agile Data Masking Administrator's Guide.
- If you select a DATESHIFT algorithm and you are not masking a datetime or timestamp column, you must specify a Date Format. (This field only appears if you select a DATESHIFT algorithm from the Masking Algorithm dropdown.) For a list of acceptable formats, click the Help link for Date Format. The default format is yyyy-MM-dd.

5. Select the Row Type according to its purpose, using "All Row" as a convention for all rows.
   If you need to create a row type (for example, if filter conditions are required), see Row Types and Creating New Row Types for Tables next.
6. Select an ID Method:
   - Auto—The default value. The profiling job can determine or update whether to mask a column.
   - User—The user decides whether to mask/unmask a column. The user's choice overrides the profiling job. (The user masking is done after the profiling job is finished.)
7. Notes can be added/removed in the Notes text field.
8. Click Save when you are finished.

- You must click Save for any edits to take effect.

Row Type and Creating New Row Types for Tables
Delphix Agile Data Masking provides a feature called **Row Types** that limits masking of a given column to a specific subset of rows.

**Note:** a database must be selected from the Select Rule Set dropdown menu on the left, not a file system.

1. From an Environment's Inventory tab, click **+Row Types** in the upper right.
   The Row Type window appears, listing existing row types.

2. Click **+ Add a Row Type**.
   The Add Row Type window appears.

3. Name the **Row Type** according to its purpose (for example, if you want to subset the rows to only take rows with addresses, you can name this row type "Address Rows").

4. To limit the masking to a subset of rows, specify an appropriate **Where Clause**.

5. Click **Save**.
Managing a File Inventory

The following sections apply to files:

- **Set Criteria for Sensitive Fields**
- **Create New Fields**
- **Add a New Record Format**

File inventory is based on file format, not on file content. Therefore, if you make a change to a file inventory, that change applies to all files that use that format.

**Set Criteria for Sensitive Fields**

*Note:* a file system must be selected from the Select Rule Set dropdown menu on the left, not a database.

1. From an Environment's Inventory tab, click the green edit icon to the right of the field you want.
2. To mask this field, check the **Mask** check box (in the View Inventory pane). Clear this check box if you do not want to mask this field.
3. Choose the appropriate sensitive data element type for the field from the **Domain** dropdown. Delphix Agile Data Masking defaults to a masking **Algorithm** as specified in the Settings screen. If necessary, you may override the default algorithm for a field.
4. To select a different masking algorithm, choose one from the **Algorithm** dropdown. For descriptions of these algorithms, see "[Algorithms](#)". Choose a **Record Type** from the drop down.
   - When selecting masking algorithms for your domains, be sure not to select BOTH algorithms created in DMsuite and mapplet-based algorithms generated with Informatica PowerCenter. When you later create your masking job, the Generator you specify (DMsuite or Informatica) determines which class of algorithm to use. The masking job cannot combine the DMsuite and Informatica PowerCenter algorithm classes, because the job will fail.
   - If you select a DATESHIFT algorithm, you must specify a **Date Format**. (This field only appears if you select a DATESHIFT algorithm from the Masking Algorithm dropdown.) For a list of acceptable formats, click the **Help** link for Date Format. The default format is yyyy-MM-dd.
5. If you are masking a delimited or fixed file, specify a **Position**: the field number (for delimited files) or the character position (for fixed files) of the beginning of the field within the data record.
6. If you are masking a fixed file, specify a **Length** (the length of this field within the data record).
7. Notes can be added/removed in the **Notes** text field. (Fields with notes display a Note icon in the Notes column.)
8. Click **Save** when you are finished.
   - You must click **Save** for any edits to take effect.

**Create New Fields**

*Note:* a file system must be selected from the Select Rule Set dropdown menu on the left, not a database.

1. From an Environment's Inventory tab, click **Define fields** to the far right. The Edit Fields window appears.
2. Edit the fields as described in "[Setting Field Criteria for a File.](#)"
3. When you are finished, click **New** to create a new field, or click **Save** to update an existing field.
Add a New Record Format

**Note**: a file system must be selected from the Select Rule Set dropdown menu on the left, not a database.

1. From an Environment's Inventory tab, click **Record Types** towards the upper right. The Record Type window appears.
2. Click **+Add a Record Type** towards the bottom of the window. The Add Record Type window appears.
3. Enter values for the following fields:
   - **Record Type Name**—A free-form name for this record format.
   - **Header/Body/Trailer**—If the file has header or trailer records, you will need to create file formats for them. Select the appropriate type. Delphix Agile Data Masking allows for masking of multiple headers, multiple trailers, and multiple types of body records.
   - **Record Type ID**—(optional) For body records, specify the value of the record type code or other identifier that allows Delphix Agile Data Masking to identify records that qualify as this record type.
   - **Position #**—(optional) Specify the field number (for delimited files) or the character position number (for fixed files) of the beginning of the Record Type Identifier within the data record.
   - **Length #**—(optional) For fixed files, specify the length of the Record Type Identifier within the data record.
4. Click **Save** when you are finished.
The Inventory Screen

From anywhere within an environment, click the **Inventory** tab to display the inventory for the rule sets for the environment and the Inventory Screen appears (Figure 3.7).
Managing Rule Sets

A Rule Set points to a collection of tables or flat files that Delphix Agile Data Masking uses for profiling, provisioning, and masking, and certifying data. For mainframe systems, the rule set represents a copybook definition for a file.
Creating and Copying a Rule Set

Creating a Rule Set for a Mainframe Environment

1. Click **Create Rule Set** to the upper right of the Rule Set screen. The Create Rule Set screen appears. This screen lets you specify which tables belong in the rule set. (Note: A Similar screen appears when you edit a Rule Set.) The Edit rule set feature is disabled for a mainframe environment.

2. Enter a name for the new rule set.

3. Select a **Connector** name from the dropdown. The list of tables for that connector appears. (If you haven't yet created any connectors, the list is empty.)

4. Click individual table names to select them, or click **Select All** (bottom left) to select all the tables in the connector.

5. Click **Save**. You are returned to the Rule Set screen.

6. Select a **Connection** name for this rule set.

7. Name and browse the PDS where copybooks are located. The PDS connection must use the same host information as the file connection.

8. Select a copybook to add to the rule set.

9. Click **Save**.

Copying a Rule Set

This feature is disabled for a mainframe environment.

When you copy a rule set, you also copy the ’ for that rule set.

1. Click the **Copy** icon to the right of the rule set on the Rule Set screen. The Copy Rule Set window appears.

2. Specify a **Name** for the new rule set.

3. Click **Save**.

Modify the rule set as you want, using the preceding procedures.
Deleting a Rule Set

If you delete a Rule Set, any inventory associated with that Rule Set will also be deleted. Also, any filter conditions defined for that Rule Set will be deleted.

**Deleting a Rule Set**

1. Click the **Delete** icon to the right of the rule set on the Rule Set screen.
Editing a Rule Set

This feature is disabled for a mainframe environment.

**Editing a Rule Set**

1. Click the **Edit** icon to the right of the rule set on the Rule Set screen.

2. Click the **Edit Rule Set** button towards the top.
   The Create Rule Set screen appears. This screen lets you specify which tables belong in the rule set.

3. Edit the rule set as you want, using the preceding procedures.

If you have tables with names that change monthly, for example tables that are appended with the current date, you can set a table suffix for a rule set.
Modifying Tables in a Rule Set (For Distributed Environment)

1. The features in this section are disabled for a mainframe environment.
2. For additional information about any of the features in this section, see "About Provisioning (Subsetting Data)."

You can modify tables in a rule set as follows:

- **Entering a Logical Key**
- **Adding or Editing a Filter**
- **Using Custom SQL**
- **Setting a Table Suffix**
- **Adding a Column**
- **Joining a Table**
- **Selecting a List**
- **Removing a Table**

### Entering a Logical Key

If your table has no primary keys defined in the database, and you are using an In-Place strategy, you must specify an existing column or columns to be a *logical key*. This logical key does not change the target database; it only provides information to Delphix Agile Data Masking. For multiple columns, separate each column using a comma.

1. From the Rule Set screen, click the name of the desired Rule Set.
2. Click the green edit icon to the right of the table you wish to edit the filter of.
3. On the left, select Logical Key.
4. Edit the text for this property.
5. To remove any existing code, click Delete.
6. Click Save.

### Adding or Editing a Filter

Use this function to specify a filter to run on the data before loading it to the target database.

1. From the Rule Set screen, click the name of the desired Rule Set.
2. Click the green edit icon to the right of the table you want.
3. On the left, select Edit Filter.
4. Edit the properties of this filter by entering or changing values in the field.

Be sure to specify column name with table name prefix (for example, customer.cust_id <1000).

1. To remove an existing filter, click Delete.
2. Click Save.

### Using Custom SQL

Use this function to use SQL statements to filter data for a table.
1. From the Rule Set screen, click the name of the desired Rule Set.
2. Click the green edit icon to the right of the table you want.
3. On the left, select Custom SQL.
4. Enter custom SQL code for this table.

Delphix Agile Data Masking will run the query to subset the table based on the SQL you specify.

1. To remove any existing code, click **Delete**.
2. Click **Save**.

**Setting a Table Suffix**

1. In the Rule Set screen, click the name of the desired Rule Set.
2. Click the green edit icon to the right of the table you wish to set the suffix for.
3. On the left, select Table Suffix.
4. The **Original Table Name** will already be filled in.
5. (Optional) Enter a **Suffix date Pattern** (for example, mmyy).
6. (Optional) Enter a **Suffix Value**, if you want to append a specific value.
7. (Optional) Enter a **Separator** (for example, _). This value will be inserted before the suffix value (for example, tablename_0131).
8. Click **Save**.

**Adding a Column**

Use this function to select a column or columns from a table when you don’t want to load data to all the columns in a table.

1. From the Rule Set screen, click the name of the desired Rule Set.
2. Click the green edit icon to the right of the table you want.
3. On the left, select Add Column.
4. Select one or more column names to include in the table. To remove a column, deselect it.
   You can also choose **Select All** or **Select None**.
5. Select **Save**.

**Joining a Table**

Use this function to specify a SQL join condition so you can define primary key/foreign key relationships between tables.

1. From the Rule Set screen, click the name of the desired Rule Set.
2. Click the green edit icon to the right of the table you want.
3. On the left, select Join Table.
4. Edit the properties for this join condition.
5. To remove an existing join condition, click **Delete**.
6. Click **Save**.

**Selecting a List**

Use this function to select a list to use for filtering data in a table.
1. From the Rule Set screen, click the name of the desired Rule Set.
2. Click the green edit icon to the right of the table you want.
3. On the left, select List.
4. Edit the text file properties for this list.
   - Select a column.
   - Enter or browse for a filename.
   - Files that have already been specified appear next to Existing File.
5. To remove an existing list file, click **Delete**.
6. Click **Save**.

**Removing a Table**

1. From the Rule Set screen, click the name of the desired Rule Set.
2. Click the red delete icon to the right of the table you wish to remove.
   - If you remove a table from a rule set and that table has an inventory, that inventory will also be removed.
The Rule Set Screen

From anywhere within an Environment, click the **Rule Set** tab to display the Rule Sets associated with that environment. The Rule Set screen appears. If you haven't yet created any rule sets, the Rule Set list is empty.
Monitor Jobs

Click the Monitor tab at the top of the screen to display all of the jobs defined to Delphix Agile Data Masking. This screen provides an overview of job activity within the entire Delphix Agile Data Masking application, and also provides a mechanism to view execution results and to run or rerun jobs. You will only see jobs associated with environments for which you have the appropriate role definition. If any job does not succeed, you can correct the errors and then rerun the job.

The following columns appear in the Jobs Editor:

- Environment
- Job Name
- Type
- Progress
- Status

Procedure

1. To go to the Environment Overview screen for any environment, click the Environment name.
2. To see detailed information for any job, click the Job Name.
3. Enter a job name in the Search Filter field and press Enter to search for a job.
Scheduler Tab

Click the Scheduler tab at the top of the screen to display the list of jobs scheduled to run. This screen provides an overview of scheduled jobs and lets the user configure schedules for jobs to run. The following columns appear on the Scheduler screen:

- Groups
- Status
- Start
- End
- Frequency
- Edit
- Delete

Enter a job group name in the Search field and click Search to search for a job group.
Creating SQL Statements to Run Before and After Jobs (For Distributed Environment)

**Not applicable if you chose Informatica as the Generator.**
When you create a masking job or a certification job, you can specify SQL statements to run before (prescript) you run a job and/or after (postscript) the job has completed.

For example, if you want to provision a schema from the source to a target, you would use a prescript (SQL statements) to disable constraints and truncate data on the target.

You create prescript and postscript by creating a text document with the SQL statement(s) to execute. If the text file contains more than one SQL statement, each statement must be separated by a semicolon (except when variables are being used in the script. Any time variables are used, a semicolon should not be used between statements until those variables are no longer needed.

For example:

```sql
DECLARE @Path VARCHAR(250) [no semicolon after this statement]
Set @Path='C:\temp\file.bak' [no semicolon after this statement]
RESTORE DATABASE Delphix Agile Data Masking
FROM DISK = @Path WITH FILE = 1; [semicolon at the end]
```
Enabling and Disabling Database Constraints

Not applicable if you chose Informatica as the Generator.

Depending on the type of target database you are using, Delphix Agile Data Masking can automatically enable and disable database constraints.

The ability to enable and disable constraints ensures that Delphix Agile Data Masking can update columns that have primary key/foreign key relationships.

You can set Delphix Agile Data Masking to handle constraints automatically.

- Support for disabling constraints is dependent on your RDBMS. If your RDBMS doesn't support disabling of foreign keys, for example, constraints must be added or dropped using prescript or postscript.
**Job Completion E-mail Message**

Upon completion of each job, an e-mail message that contains job start and end times, along with the completion status, is sent to the user whose e-mail address is specified in the **E mail** field (in the Edit Job window).
Scheduling Job(s) to Run

Scheduling New Jobs

1. Click **Create Scheduler** at the upper right of the screen. The Schedule Jobs screen appears. You will be prompted for the following information:
   - **Group Name**—A free-form name for this job schedule.
   - **Scheduled Date**—Enter the date when you want to run the job group, in the form mm/dd/yyyy.
   - **Scheduled Time**—Enter the time when you want to run the job group, in the form hh:mm.
   - **Frequency**—(Optional) Select the frequency at which you want to run this job group: Daily, Weekly, Monthly, Yearly. Default is daily if none is chosen.
   - **Jobs**—Jobs are grouped by their Environment. Expand Environments and use the check boxes to add jobs to this group.

2. When you are finished, click **Save**.

   - Jobs will run serially (one after the other). If you want to run jobs simultaneously, create two schedules with the same start time.

All of the jobs you specified for this job group will be run, serially, beginning at the appointed time.

Editing a Schedule

- From the Scheduler tab, click the **Edit** icon to the right of the schedule you want.

Deleting a Schedule

- Click the **Delete** icon to the right of the schedule you want.
Settings Tab

**Note:** This user guide only gives an overview of the Delphix Agile Data Masking settings. For more detailed information, see the Delphix Agile Data Masking Administrator's Guide.

Click the **Settings** tab at the top of the screen to view or change Delphix Agile Data Masking settings.

There are several areas to which settings are applied:

- **Algorithms**
- **Domains (Masking)**
- **Profiler**
- **Mapping**
- **File Formats**
- **Remote Server**
Algorithms

Note: This user guide only gives an overview of the Delphix Agile Data Masking settings. For more detailed information, see the Delphix Agile Data Masking Administrator's Guide.

The main methods used by Delphix Agile Data Masking algorithms are secure lookup and segmented mapping. Delphix Agile Data Masking also includes some algorithms for specific types of data (for example, for phone numbers and dates). These standard Delphix Agile Data Masking algorithms are available if you select Delphix Agile Data Masking as the Generator.

- Delphix Agile Data Masking includes mapplets created specifically for use with Informatica PowerCenter. These mapplet-based algorithms are available if you select Informatica as the Generator. In addition, you can create your own PowerCenter mapplets to use with Delphix Agile Data Masking. For information about how to integrate mapplets into Delphix Agile Data Masking, see Delphix Agile Data Masking Administrator's Guide.

From the Settings Tab screen, if you click Algorithm to the left, the list of algorithms will be displayed.

Types of Algorithms

If you then click Add Algorithm in the upper right, a dialogue box will appear. On the left of this dialogue box are the 4 types of algorithms:

1. Secure Lookup Algorithm: Uses a lookup file to assign masked values in a consistent manner. The design of the algorithm introduces intentional collisions.

2. Segmented Mapping Algorithm: Replaces data values based on segment definitions. For example, an ACCOUNT NUMBER algorithm might keep the first segment of an account number, but replace the remainder or remaining segments with a random number.

3. Mapping Algorithm: Sequentially maps original data values to masked values that are pre-populated to a lookup table through the Delphix Agile Data Masking user interface.

4. Binary Lookup Algorithm: Much like the Secure Lookup Algorithm, but used when entire files are stored in a specific column.

The Delphix Agile Data Masking Generator algorithms give you the ability to quickly and easily define the algorithms you want, directly on the Settings page. Then, you can immediately propagate them. Anyone in your organization who has Delphix Agile Data Masking can then access the info.

For information about creating your own algorithms with Delphix Agile Data Masking or Informatica PowerCenter, see Delphix Agile Data Masking Administrator's Guide.
Domains (Masking)

**Note:** This user guide only gives an overview of the Delphix Agile Data Masking settings. For more detailed information, see the Delphix Agile Data Masking Administrator's Guide.

Domains specify certain data to be masked with a certain algorithm.

From the Settings Tab screen, if you click **Domains** to the left, the list of domains will be displayed. From here, Domains can be added, edited, or deleted. A Domain's algorithm can be set here; to change a Domain's selection expressions or to group Domains into sets, continue to the next section.
File Format

**Note:** This user guide only gives an overview of the Delphix Agile Data Masking settings. For more detailed information, see the Delphix Agile Data Masking Administrator's Guide.

File Formats are a way of organizing the types of files to be masked. Before a file can be masked, it needs to have a File Format assigned to it (This can be done in the Rule Set Screen by selecting a Rule Set, then clicking the green edit icon to the right of a file).

From the Settings Tab screen, if you click **File Format** to the left, the list of File Formats will be displayed. From here File Formats can be added or deleted.
Mapping

From the Settings Tab screen, if you click **Mapping** to the left, the list of Mappings will be displayed. From here Mappings can be added edited or deleted.
Profiler

**Note:** This user guide only gives an overview of the Delphix Agile Data Masking settings. For more detailed information, see the Delphix Agile Data Masking Administrator's Guide.

The profiler is used to group domains into Profile Sets (aka Profiles) and assign expressions to domains. Profile Sets can be used in Profiling Jobs.

From the Settings Tab screen, if you click **Profiler** to the left, a list of expressions will be displayed. From here, you can work with:

- **Expressions:** Expressions are used to specify what data is sensitive. This can be done at either the column or data level. Each expression is assigned to a domain (and domains can be grouped into Profile Sets). Expressions can be added, edited or deleted from the Profiler Settings page.

- **Profile Sets (Profiles):** A Profile Set is a group of Domains. Said another way, it is a group of certain data to be masked in certain ways. Profiles can be added, edited, or deleted by clicking **+ Profile Set** at the top of the Profiler Settings page.
Remote Server

Note: This user guide only gives an overview of the Delphix Agile Data Masking settings. For more detailed information, see the Delphix Agile Data Masking Administrator's Guide.

Delphix Agile Data Masking typically executes jobs on a local instance. Remote Servers are for executing jobs elsewhere.

From the Settings Tab screen, if you click Remote Server to the left, the list of Remote Servers will be displayed. From here, Remote Servers can be added, edited, or deleted.
Jet Stream Admin Guide

Table of Contents

Getting Started Topics
Welcome to Jet Stream
User Roles and Permissions
Admin User
Jet Stream Data User
Login

Navigating the Jet Stream Admin Interface Topics
Jet Stream Administrator Home Page
Jet Stream Data Template Management
Jet Stream User Roles and Permissions in Admin App
Jet Stream Data Management User Interface for Delphix Administrators

Jet Stream Concept Topics
Data Sources
Data Templates
Data Containers
Jet Stream Data Flow

Understanding Jet Stream Data Templates Topics
Jet Stream Data Templates: An Overview
Jet Stream Data Template Activities
Creating a Data Template
Managing Data Template Notes
Editing a Data Template's Name
Deleting a Data Template

Understanding How to Manage Data Template Details Topics
Viewing and Working with the Data Template Details Page
Summary
Containers
Sources
Properties
Bookmarks
Capacity

Understanding Jet Stream User Management Topics
Jet Stream User Management Activities
Creating a Jet Stream User
Assigning a Jet Stream User to a Data Container
User Details Page

Understanding Jet Stream Data Container Topics
Jet Stream Data Container Overview
Refresh
Restore
Reset
Branch
Activate
Bookmark
Share
Jet Stream Data Container Activities
Configuring Jet Stream Data Containers
Data Management Operations

Understanding Jet Stream Usage Management Topics
Jet Stream Usage Management Dashboard Overview
Template Usage Overview
User Usage Overview
Template Usage (Containers) Overview
Template Usage (Bookmarks) Overview
Container Usage (Branches) Overview

Understanding Bookmark Topics
Bookmarks Overview

Resources

Support
Getting Started with Jet Stream

Welcome to Jet Stream

The Delphix agile data platform has greatly improved the speed at which end users can get the data that they need. While end users reap the benefits, they do not typically interact with the Delphix Engine directly, nor are they necessarily even aware that they are using it. End users most commonly file tickets for data management operations and wait for the tickets to be serviced by their IT organization. Delphix data management workflows allow database administrators (DBAs) to respond to these tickets much more quickly and reliably, but DBAs are often overloaded, and resolving high-priority issues takes precedence over requests from users. Requiring interactions between users and IT for every data operation is inefficient and can lead to unwanted delays.

The goal of Jet Stream is to create a clear separation of IT infrastructure and data management. As with the current Delphix platform, IT administrators and DBAs continue to control decisions about how resources such as virtual databases (VDBs) and vFiles are allocated. However, with Jet Stream, administrators can also assign these resources directly to a user. A Jet Stream user has the ability to control what data these resources should make available, even though the details of the physical resources are hidden from them. This separation of roles empowers Jet Stream users to get the data they need, when they need it, while providing administrators with the controls to ensure resources are accounted for appropriately.

User Roles and Permissions

Jet Stream has two types of users:

Admin User

Admin users have full access to all report data and can configure Jet Stream. Additionally, they can use the Delphix data platform to add/delete Delphix Engines, add/delete reports, add/delete users, change tunable settings, add/delete tags, and create and assign data templates and containers.

Jet Stream Data User

Jet Stream data users have access to production data provided in a data container. The data container provides these users with a playground in which to work with data using the Self-Service Toolbar.

Login

1. Access Jet Stream by opening a web browser and using the IP address or DNS qualified host name.
2. Login with the Delphix Admin User ID and Password provided for you.
Jet Stream Concepts

Data Sources

A data source in Delphix can represent a database, an application, or a set of unstructured files. Delphix administrators configure the Delphix Engine to link to data sources, which pulls the data of these sources into Delphix. The Delphix Engine will periodically pull in new changes to the data, based on a specific policy. This, in turn, begins building a custom timeline for each data source. Additionally, the Delphix Engine can rapidly provision new data sources that are space-efficient copies, allowing users to work in parallel without impacting each other.

Data Templates

Data templates are the backbone of the Jet Stream data container. They are created by you, the Delphix administrator, and consist of the data sources users need in order to manage their data playground and their testing and/or development environments. Data templates serve as the parent for a set of data containers that the administrator assigns to Jet Stream users. Additionally, data templates enforce the boundaries for how data is shared. Data can only be shared directly with other users whose containers were created from the same parent data template.

Data Containers

A Jet Stream data container allows data users to access and manage their data in powerful ways. Their data can consist of your application binaries, supporting information, and even the entire database(s) that underlie it. The Jet Stream data container allows users to:

- Undo any changes to their application data in seconds or minutes
- Have immediate access to any version of their data over the course of their project
- Share their data with other people on their team, without needing to relinquish control of their own container
- Refresh their data from production data without waiting for an overworked DBA

A Jet Stream data container consists of one or more data sources, such as databases, application binaries, or other application data. The user controls the data made available by these data sources. Just like data sources in a template, changes that the user makes will be tracked, providing the user with their own data history. The Jet Stream Data Container Interface lets users view the details and status of their data container and its associated data sources, as well as manipulating which data is in those sources. The Data Container Interface includes a section called the Data Container Report Panel, which displays details about each source, including the connection information needed to access it - for example, the java database connectivity (JDBC) string for a database. This connection of information is persistent and stable for the life of the data container, regardless of what data the resources are hosting.

Jet Stream Data Flow

The Jet Stream data flow diagram below demonstrates how a Jet Stream data user accesses data sources. Data sources are connected to a Delphix Engine, which is controlled by the Delphix administrator. The Delphix administrator will connect all data sources that developers and quality assurance (QA) teams need to a Jet Stream data template. This data template acts as a parent source to create the data containers that the administrator will assign to Jet Stream data users. Data sources flow from the Delphix Engine into a data template and downstream into a data container, where a Jet Stream data user will use the data sources to complete tasks. The data container acts as a self-contained testing environment and a playground for the Jet Stream data user. Additionally, Jet Stream data users are able to set, bookmark, and share data points in their container with Jet Stream data users using other data containers, as long as all the data containers were created from the same parent data template.
JetStream Data Flow, Version 1.0.0
Navigating the Jet Stream Admin Interface

The following screenshots provide a roadmap for how to navigate the primary screens and places a user will go within the Jet Stream Admin Interface. The interface includes screens such as the Jet Stream Administrator Home Page, Jet Stream Data Platform Management, Jet Stream Users and Permissions, and the Data User Management Interface.

JetStream Delphix Admin User Interface Drop Down Menu, Version 1.0.0

Jet Stream Administrator Home Page

The Jet Stream Administrator Home Page is the home page of Jet Stream once an admin user has selected Jet Stream in the Admin App user log in dropdown menu. On this page you can add and view existing data templates and data containers. You can also view users, whom you can then assign to data containers that you create from existing data templates.

JetStream Administrator Home Page, Version 1.0.0

Jet Stream Data Template Management

The Jet Stream Data Template Management page contains a view panel of 6 tiles on the left-hand side of the screen. Each tile reports on a variety of useful information, such as user activity, data sources, data capacity, specific details about data containers, and data templates. They also help you navigate to areas where you can complete specific tasks, such as creating a new template or container, working with data timeflows, assigning users to containers, and bookmarking important points in time.
Jet Stream User Roles and Permissions in Admin App

The Jet Stream User Roles and Permissions Page is found under Resources-Users in the Admin App. Here you can add new Jet Stream users by selecting the Jet Stream users only in the user details box, as seen in the screenshot below.

Jet Stream Data Management User Interface for Delphix Administrators

The Jet Stream Data Management User Interface is the only interface to which Jet Stream data users have access and with which they interact. The user interface is the environment in which a data user works with data in an assigned data container, using data sources from a data template. The user interface is divided into two work areas. The upper half allows the user a workspace to complete tasks using self-service operations. The lower half provides users with a summary of important details about the data container and offers bookmark management. For more details about how to use this interface, please refer to the Jet Stream Data User Guide. The screenshot below illustrates the data user interface.
JetStream Data Management User Interface, Version 1.0.0
Understanding Jet Stream User Management

Jet Stream User Management Activities

This document describes the process of creating a Jet Stream user and assigning that user to a data container. It also provides an overview of the Jet Stream User Details page.

Creating a Jet Stream User

Follow the same process when creating a new user or modifying an existing Delphix user. Jet Stream users do not have access to the existing admin UI, and they can only access the Jet Stream Data Container page for containers they own.

1. From the Admin App, select Manage -> Users.

   1. Click Add User.
      a. To make an existing user a Jet Stream user, select the user from the list.

2. Enter the appropriate information.

3. Select the JS-Only User checkbox.

   1. Press Save.

The user is now a Jet Stream user! This means that the user can now login to the Jet Stream user interface, and you can make the user the owner of a data container.

Notes

- Jet Stream users will only be able to access the Jet Stream Data Management page. They will not be able to access the other portions of the Jet Stream interface, nor the Admin App.
- A Delphix admin user cannot be made a JS-Only User. However, admins can still use Jet Stream and own a data container. Admins are also able to manage all data containers.
- A user who owns one or more data containers cannot be deleted.
Assigning a Jet Stream User to a Data Container

This section describes how to assign a Jet Stream user (created in the previous section) to a data container. Making a Jet Stream user the owner of a data container allows them to perform operations such as Refresh on that data container. Jet Stream users cannot see or manipulate data containers that they do not own. You can either assign a user when creating a new data container, or modify the owner of an existing data container.

Case 1: Data Container Creation

1. On the Create Data Container page, select the desired owner from the drop-down owner menu.

Case 2: Changing the Owner of an Existing Data Container

1. On the Management Overview page, select the data template from which the data container was provisioned.
2. Click the Containers tile in the left-hand panel.
3. Click the Edit icon next to the name of the data container’s owner.

1. Select the desired owner from the drop-down list.
   a. To remove the current owner, select <None> from the list.
2. When you are finished editing, click the checkmark to the right.

The user you selected is now the owner of the data container and can perform operations on that data container.

Note

- A data container can only have a single owner at a time
User Details Page

This section provides an overview of the Jet Stream User Details page. This page displays graphs related to the user's Jet Stream activity, as well as a list of all of the data containers that the user owns.

1. On the Management Overview page, click the Users tab.

1. Select the name of the desired user to go to their User Details page.

The user details page looks like this:

Jet Stream User Details, version 1.0.0

The Operation Counts By Week graph shows the aggregate of all Jet Stream operations performed by this user on all of their containers. The Container Age Distribution graph shows the average time since a data operation was performed on all of the user's containers. Each container that the user owns will appear in the Containers section.
Understanding Jet Stream Data Templates

Jet Stream Data Templates: An Overview

A Jet Stream data template represents a collection of data sources that you can provision to a Jet Stream user. A data source can be a dSource, a VDB, or vFiles. These sources can be used in multiple data templates. Once you have created a data template, the set of data sources associated with it is fixed; you cannot add data sources to an existing template, nor can you remove data sources from it. In addition to data sources, you can define the set of metadata that is relevant for a given template – for example, notes, descriptions, names for sources that are relevant to an end user, and other configuration details. Once you have created the template, it provides a stencil for provisioning data containers. This, in turn, enables Jet Stream users to have self-service access to a space-efficient copy of the data sources defined in the data template.

Jet Stream Data Template Activities

Data templates are managed by a Delphix admin. The admin can provision data containers from the data template and assign a data container to an end user. The admin can also create bookmarks on the data template timeline in order to mark meaningful points in time.

When creating a data template, it is important to consider the set of users who will own data containers provisioned from it. In Jet Stream, templates effectively define the boundaries of the data that users can share directly with each other. Only owners of data containers created from the same data template are able to share data using bookmarks.

Creating a Data Template

A data template consists of an arbitrary set of dSources, VDBs, and vFiles. These are created and managed in the Delphix Admin interface, and can be used in Jet Stream as data sources. You can use any data type supported by the Delphix Engine as a data source in Jet Stream. For more information, refer to the Linking/Provisioning documentation for the standard Delphix Engine. The following is an example of the many kinds of data sources you can use to create a data template.

Multiple JetStream Data Sources, Version 1.0.0

To create a data template:

1. Select Jet Stream from the drop-down menu in the upper right-hand corner of the Delphix UI.
JetStream Log In, Version 1.0.0

1. On the Mgmt Overview page, click Add Template.

JetStream Add Template, Version 1.0.0

This will send you to the Create Template page.

1. Enter a Name for the data template.
2. Optionally, enter a description for the data template.
3. Click Add Data Source to add data sources to the template.
   a. Use the drop-down menu to select the source you want to include. The drop-down menu will display all dSources in the system and all VDBs and vFiles that are not already assigned to a Jet Stream data container.
   b. Enter a Jet Stream-specific name for the data source.
   c. Optionally, enter a description in the Notes section. Jet Stream users see a copy of these notes in the data containers they own.
   d. Click Add Data Source to continue to add and configure more data sources to the data template. You can remove data sources using the Delete icon, highlighted below.
Adding Data Sources to a Data Template, Version 1.0.0

1. Click **Create** to finish creating the data template.

**Note:** Once you have created a template, you cannot change the set of data sources in it. Any vdb's or dSources being used as data sources in Jet Stream will appear with a special badge in the Admin App.

Managing Data Template Notes

After you have created the data template, it will be visible from the **Management Overview** page under the **Data Templates** tab, which is the default tab.

**Notes**

- Each tile corresponds to a data template and contains high-level information about that data template. For instance, the number of child data containers is visible under the name of the container. Note that these graphs are not very interesting at the moment, because you have not created any data containers yet.
- You can sort and filter the data template tiles, making it easy to manage a large number of data templates in Jet Stream.
Editing a Data Template’s Name

1. Click the Edit icon next to the data template name.

   1. Enter the new name.
   2. Click the checkmark icon to confirm changes.

Deleting a Data Template

1. Select the data template you would like to delete.
2. Click the Delete icon in the lower right-hand corner of the tile.

Note: If there are any data containers provisioned from the data template, you must remove them before you can delete the data template. See instructions in the Managing Data Containers section of this guide.
Understanding How to Manage Data Template Details

Viewing and Working with the Data Template Details Page

1. In the Data Management Page, under the Templates Tab, Select and click the data template's name. This will direct you to the Data Template Details page. You can use this page to view and configure details of an individual data template. It consists of a number of tiles:

**Summary**

Use this tile to get an overview of the data template and its child data containers.

JetStream Summary Details, Version 1.0.0

**Notes**

- The graphs labeled **Operation Counts By Week** and **Container Data Age Distribution** give a sense of the amount of activity in the data template over time
- **Top Checkouts** shows at a glance which bookmarks have been used most frequently as part of a **Restore** or **Branch** operation
- **Top Users by Data Operations** shows at a glance which users are the most active in Jet Stream

**Containers**

Use this tile to create, view, and delete child data containers from this data template.
JetStream Container Details, Version 1.0.0

Sources

In this tile, you can view the data sources that this data template uses. Each data source has a Jet Stream user-visible name, a description, and a set of properties that consist of arbitrary key/value pairs. This information will be included in the data containers provisioned from this template.

JetStream Sources Details, Version 1.0.0

Properties

Use this tile to edit the data template's properties. Properties are arbitrary key/value pairs associated with the data template. These values will be propagated to all data containers provisioned from this template. This provides a way for you to annotate data templates and data containers with whatever information is relevant to their use case.
JetStream Properties Details, Version 1.0.0

**Bookmarks**

Use this tile to create and manage bookmarks on the data template. A bookmark represents a given point in time that is protected against retention. Bookmarks created on a data template are visible to all of the data containers provisioned from it. For more details, refer to the Bookmarks section in the Jet Stream Data User Guide.

Jet Stream Bookmarks Details, Version 1.0.0

**Capacity**

Use this tile to get information about the storage associated with the data template and its child containers.
Jet Stream Capacity Details, Version 1.0.0

<table>
<thead>
<tr>
<th>Branch</th>
<th>Unique</th>
<th>Shared (others data)</th>
<th>Shared (self data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>220.8MB</td>
<td>72.00MB</td>
<td>0B</td>
</tr>
</tbody>
</table>
Understanding Jet Stream Data Containers

Jet Stream Data Container Overview

Data containers are provisioned from data templates by administrators and assigned to a Jet Stream user. A data container represents a socket that is capable of making any data within the data template accessible. The Jet Stream user controls what data they want to access. Jet Stream users have effectively been provisioned a set of "physical" resources, such as a database on a host that consumes some set of resources. A data container is comprised of a VDB or vFiles provisioned from each source in the data template from which it is created. The data container manages these VDBs, and the data operations performed on a data container will only impact these VDBs. Data containers represent the separation between IT infrastructure and end users. IT determines the set of VDBs or vFiles to allocate to a data container, and Jet Stream users determine the data that they want accessible in the containers allocated to them.

Data containers can be used to access any data within a single data template, but not across templates. Jet Stream users have the ability to populate the data within their data container from any point in time on the data template, the data container's history, or shared bookmarks from other data containers. Although operations are all accomplished by performing timeflow operations on the underlying VDBs, the data containers hide the VDBs and their underlying properties from Jet Stream users. None of the data container operations require provisioning additional VDBs; everything is accomplished using the resources assigned when the data container is created.

Refresh

This is the same basic concept as Refresh in VDBs today. In Jet Stream, Refresh will update the data on the active branch of a user's data container. The user will then have the latest data in the sources of the data template from which the container was provisioned.

Restore

Restore allows a Jet Stream user to update the data on the active branch of their data container to any point in time on the data container, the data template from which the container was provisioned, or a bookmark. This operation effectively means, "Take me to the data at this time."

Reset

Reset is a simplified version of Restore built to support the notion of "undo." It allows a user to reset the state of their application container to the latest operation. This can be useful for testing workflows where, after each test, users want to reset the state of their environment.

Branch

A Jet Stream branch represents a logical timeline, effectively a task on which a user is working. Only one branch can be active at a time, but a user can use multiple branches to track logically separate tasks. Jet Stream branches do not require the allocation of a new VDB; instead, they are comprised of a collection of timeflows within a VDB.

Activate

This allows the user to select which branch they want to be active. Only a single branch within a data container can be active at a time.

Bookmark
This creates a semantic name for a point in time and prevents this data from being removed by the retention policy. Bookmarks can be annotated with tags to make them easier to search for. In addition to tags, bookmarks allow a user to enter a description of what the bookmark represents.

**Share**

Bookmarks can be shared, which allows them to be seen by users who own data containers that have been provisioned from the same data template. This allows users to share data, providing a way for other users to either restore their existing timeline or create a new branch from these shared points.

**Jet Stream Data Container Activities**

**Configuring Jet Stream Data Containers**

A Jet Stream data container is comprised of a set of VDBs, where each VDB is a direct child of the dSource, VDB, or vFiles in the data template's data sources. Jet Stream does not automatically provision VDBs when creating a data container; a Delphix admin must create the required VDBs via the existing Admin App. Once the data container has been created, these VDBs are managed exclusively through Jet Stream.

1. From the **Management Overview** page, select a **template** from which you want to create the data container.

Jet Stream Management Overview for Templates, Version 1.0.0

This will take you to the Data Template Page. Click **Add Container** in the upper right-hand corner of the screen.
Jet Stream Details Panel and Dashboard. Version 1.0.0

This will take you the Create Data Container page.

Jet Stream Data Container Page, Version 1.0.0

Enter information about the data container, such as the **Name** and **Description** (optional).
Select the **Owner** of the data container from the drop-down menu. **Note:** Any Delphix administrator is able to manage all containers, so the owner should be the end user. Refer to the User Management section in this guide for details.

**Jet Stream User Drop Down**

1. Select the **VDBs** to use for this container's data sources.
   a. The available VDBs have the following constraints:
      i. They have been provisioned from the dSources/VDBs belonging to the parent data template
      ii. They are not already part of another Jet Stream data template or container

**Note:** If there are no VDBs that meet these constraints, you may see a message informing you that you do not have any compatible VDBs. Click **Create.**

**Delete a Data Container**

By default, all data sources (VDBs and vFiles) in a Jet Stream Data Container are deleted as part of the Jet Stream Data Container deletion process.

When performing the Delete Container operation, you can uncheck the **Delete associated VDBs and vFiles** box in the dialog window to keep these data sources intact after the Data Container is deleted.
Data Management Operations

Start a Data Container

Starting a Data Container does the following:

- Starts the data sources. This means that each data source listed in the Source Details section of the Data Container page will start using CPU and network resources on the host system it is running.
- Puts a copy of the data from the active branch into those data sources

On the Self-Service Toolbar, click Start.

Stop a Data Container

Stopping a data container does the following:

- If not already done, copies the current data in the data sources into the active branch of the data container
- Shuts down the data sources. This means each data source listed in the Source Details section of the Data Container page will stop using CPU and network resources on the host system.

On the Self-Service Toolbar, click Stop.

Jet Stream Self Service Toolbar, Version 1.0.0

Other operations on the data container, such as Stop, Reset, and Refresh, must be performed from the Data Management page:

Jet Stream Data Management Interface Shortcut in Jet Stream Data Template, Version 1.0.0
Understanding Bookmarks

Bookmarks Overview

Bookmarks are a way to mark and name a particular moment of data on a timeline. You can restore the active branch's timeline to the moment of data marked with a bookmark. You can also share bookmarks with other Jet Stream users, which allows them to restore their own active branches to the moment of data in your container. The data represented by a bookmark is protected from retention, meaning it will not be deleted until the bookmark is deleted. Once created, you can easily locate a bookmark through one of the bookmark viewers in the interface. To understand how to use bookmarks in Jet Stream, please refer to the Jet Stream Data User Guide.

Using Bookmarks in Data Templates

An admin user can create a bookmark on a template that will then be propagated to all containers created from the template. Additionally, an admin user can create a bookmark on the master template timeline with the point of time you are interested in. The bookmark will always be saved from retention policies and a new branch can be created from this bookmark.
Understanding Jet Stream Usage Management

Jet Stream Usage Management Dashboard Overview

Jet Stream data templates are comprised of dSources, virtual databases (VDBs), and vFiles. These data sources are controlled by the standard policies configured in the Admin App of the Delphix Engine. As with existing containers, space will be reclaimed by the retention policy over time. As retention cleans up historical data, users will no longer be able to use those points in time to restore or branch. In Jet Stream, an admin can create a bookmark on the data template timeline, which will prevent retention from cleaning up the data that a bookmark references.

Jet Stream data containers are comprised of VDBs provisioned from the sources defined in the data template. Similar to VDBs in the existing Admin App, data containers' VDBs will share blocks with the source from which they are provisioned. This prevents the referenced data on the source from being cleaned up by retention. Retention for these VDBs is controlled by the standard Delphix retention policies. As on templates, bookmarks in data containers will prevent storage from being reclaimed by retention. In addition, Jet Stream will ensure that the latest data on each Jet Stream branch is never removed.

The Usage pages of the data templates and data containers provide information that can help you understand how storage is being used, how to reclaim space, and how much space you are able to reclaim.

Usage Overview is a top-level page, along with the Data Mgmt and Mgmt Overview pages. It contains the space usage breakdowns by data templates and users.

Template Usage Details

You can locate the Usage tile at the bottom of the Jet Stream navigation sidebar, as seen in the image below. Usage summaries are available for templates, containers, and users. For example, when you click the Usage tile on the Template Details page, the usage details you interact with will be in the context of the selected data template. The same is true when you are navigating the Data Management page for the data containers, and the User Details page for users.
The Usage tile in the Jet Stream navigation sidebar, Version 4.2

Template Usage Overview

The Template Usage Overview page, seen in the image below, contains the usage breakdowns for data templates and users. The interface is interactive and allows you to visualize data by interacting with pie charts, bar graphs and tables. The pie chart contains information about the top 10 space consumers; the table at the bottom contains information about all of the templates and/or users. The table below the charts includes category fields. You can find corresponding descriptions by hovering over the names of the fields in the table:

Additionally, the table allows you to sort, navigate, and interact by clicking the field category of interest. For example, to sort the table, click a column header such as Unvirtualized and the table will sort by that category. To navigate to a particular data template or user, you can click either the pie slice or the name of the template/user in the table.
Table of templates/users, Version 4.2

The field categories display the following information:

- **Total** – The sum of the space used by the data containers provisioned from this data template and by the bookmarks created on this template. This is the space that will be freed if you delete the template.
- **Containers** – The amount of space used by the data containers provisioned from this data template. This is the space that will be freed if you delete or purge all of the data containers.
- **Bookmarks** – The amount of space used by the bookmarks on this data template. This is the space that will be freed if you delete all bookmarks on the template.
- **Unvirtualized** – The amount of space that would be used by the data in this template and its child data containers without Delphix virtualization.

The pie chart and table graphs can help you analyze storage usage information.

**User Usage Overview**

The User Usage Overview page provides graphical visualizations of space used by the Jet Stream users assigned to data containers. The two category fields include the number of containers owned and the amount of space being referenced by a user.

![User Usage Overview](image)

The User Usage Overview page, Version 4.2

The field categories display the following information:

- **Referenced** – The amount of space used by data containers that are owned by this user. This excludes the space that this user is sharing with other users.
- **Containers Owned** – The number of data containers owned by this user.

**Template Usage (Containers) Overview**

The Template Usage Details page, as seen below, shows the space used by data containers provisioned from the template and the bookmarks created on the template.
The Template Usage Details page, Version 4.2

The stacked bar graph shows information about the top 10 space users. You can re-sort the graph based on the fields in the **Sort by** legend on the top right-hand corner of the screen as seen in the image above. For example, if you want to know which data containers are sharing the most data with others, you can un-select **Shared (others data)** and **Unique** by clicking them in the legend.

**Note:** When the legend items are not selected, their corresponding colored boxes turn gray and the data is removed from the chart. The data and name will reappear when you re-select by clicking the grayed-out category you want.

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this data container. This assumes that also delete underlying data sources.
- **Shared (others data)** – The amount of space that cannot be freed on the parent data template (or sibling data containers) because it is also being referenced by this data container due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up the space.
- **Shared (self data)** – The amount of space that cannot be freed on this data container because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks
- **Unvirtualized** – The amount of space that would be used by the data in this container without Delphix virtualization

**Template Usage (Bookmarks) Overview**

As shown in the image below, the **Template Usage Details** page provides the usage information about bookmarks created on a template. The primary categories of information include **Unique, Shared (others data)** and **Shared (self data).**
Template Usage (Bookmarks), Version 4.2

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this bookmark
- **Shared** – The amount of space referenced by this bookmark that cannot be freed by deleting this bookmark because it is also referenced by neighboring bookmarks or branches that have been created or restored from this bookmark
- **Externally Referenced** – The amount of space referenced by this bookmark that cannot be freed by deleting this bookmark because it is also being referenced outside of Jet Stream – for example, by a retention policy.

**Container Usage (Branches) Overview**

As detailed in the image below, the **Container Usage Details** page shows the usage information about the branches and bookmarks created on a container. The primary categories of information include **Unique, Shared (others data)**, and **Shared (self data)**.

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this branch
• **Shared (others data)** – The amount of space that cannot be freed on the parent data template or sibling branches because it is also being referenced by this branch due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up the space.

• **Shared (self data)** – The amount of space that cannot be freed on this branch because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks.
Resources

Access more resources at http://docs.delphix.com/display/DOCS42/Delphix+Engine+4.1+Documentation
Support

Ask the community for support at [https://community.delphix.com/delphix](https://community.delphix.com/delphix). If you are seeing an issue that cannot be resolved with help from the community, file a support case as appropriate.
Jet Stream Data User Guide

Table of Contents

**Getting Started Topics**
Welcome to Delphix Jet Stream
User Roles and Permissions
Admin User
Jet Stream Data User
Login

**Jet Stream User Interface Topics**
Data Container Workspace (Top Half of the Jet Stream Interface)
Data Container Workspace
Branch Timeline
Data Container Self-Service Toolbar
Data Container View Panel
Jet Stream User Log In and Settings Drop Down Menu
Data Container Drop Down Menu
Data Container Report Panel (Bottom Half of the Jet Stream Interface)
Data Container Report Panel
Summary
Sources
History
Bookmarks
Usage

**Jet Stream Concepts Topics**
Understanding Data Sources
Understanding Data Templates
Understanding Data Containers
Jet Stream Data Flow
Understanding Branches
Understanding Timelines
Branch Timeline
Container Timeline
Select a Point in Time with the Time Selector
Select a Point in Time with the Time Selector Calendar
Understanding the Self-Service Toolbar
Branch Timeline Segments
Working with Multiple Branches and Timelines
Understanding How to Preserve Data in a Point of Time
Understanding Bookmarks
Bookmarks Tab in the Data Container View Panel
Bookmarks Tile in the Data Container Report Panel
Bookmark Sharing Permissions
Bookmark Appearance
Data Container Storage and Retention for Branches and Timelines
Working with Data Sources in a Container Topics
Getting Started
Activity One: How to Start and Stop a Data Container
Working with a Branch, a Branch Timeline, and the Self-Service Toolbar
Activity Two: Using Reset from a Bookmark to Facilitate Destructive Testing
Activity Three: Using Refresh to Get the Latest Data From a Data Template
Activity Four: Using Restore to Return Data Back to a Point in Time
Activity Five: Create a New Branch and Switch Between Branches
Activity Six: Rename and/or Delete a Branch
Working with Bookmarks
Activity Seven: Share a Bookmark with Other Jet Stream Users
Activity Eight: Editing Bookmarks
Activity Nine: Filter and View Bookmarks

Understanding Jet Stream Usage Management Topics
Jet Stream Usage Management Dashboard Overview
Container Usage Overview
Bookmarks Usage Overview
Branches Usage Overview

Resources

Support
Getting Started

Welcome to Delphix Jet Stream

Jet Stream grants access to the data that users need, whenever they need it. Once users have been assigned a Jet Stream data container, they can control the data available within it. This means they can refresh to the latest production data, roll back to a previous point in the data container’s timeline, and share data with another Jet Stream user without requiring any involvement from Information Technology or database administrators (DBAs). Self-service data management allows developers to be more productive while using fewer resources, dramatically improving operational efficiency.

User Roles and Permissions

Jet Stream has two types of users:

Admin User

Admin users have full access to all report data and can configure Jet Stream. Additionally, they can use the Delphix data platform to add/delete Delphix Engines, add/delete reports, add/delete users, change tunable settings, add/delete tags, and create and assign data templates and containers.

Jet Stream Data User

Jet Stream data users have access to production data provided in a data container. The data container provides these users with a playground in which to work with data using the self-service toolbar.

Login

1. Access Jet Stream by opening a web browser using the IP address or DNS qualified host name.
2. Login with the User ID and Password the Delphix Administrator has provided for you.
Jet Stream User Interface

The Jet Stream User Interface is organized within a single web browser page. The upper half of the screen represents an interactive data container workspace, and the bottom half of the screen serves as a data container report and management panel. The diagram below provides a visual orientation along with descriptive narratives to navigate a user to Jet Stream activities and viewing panels.

Data Container Workspace (Top Half of the Jet Stream Interface)
Data Container Workspace

The Data Container Workspace contains all the tools, actions, and view panels needed to begin using Jet Stream features. For example, the workspace allows a user to view the history of their data on a branch, and to refresh, reset, and restore that data.

Branch Timeline

Use this to view the timeline associated with a branch. Note that this only shows the timeline for a single branch. The branch timeline is how a user interacts with data in the container to mark, stamp, and perform tasks that occur at various points in time.

Data Container Self-Service Toolbar

The Data Container Self-Service Toolbar allows you to perform tasks and activities with data in the current container, by clicking on the following user action icons:

- **Activate** will make a branch active
- **Bookmark** will mark an interesting point of data on a branch timeline
- **Branch** will create a branch that supports one task. A branch is a group of data time segments called a "timeline."
- **Share** will share a bookmark with users of other data containers from the same template
- **Refresh** will refresh each source in the data container on a branch timeline to the latest data in the corresponding source of the data template.
- **Restore** will restore the data to a point in time from the template, the container, or a shared bookmark.
- **Reset** will reset to the last interesting moment of data time on the current data timeline
Stop will stop a data container
Start will start a data container

Data Container View Panel

The Data Container View Panel, found in the upper left-hand side of the screen, is divided into three tabular sections: time, branches, and bookmarks. These tabs allow you to find and select data that you are interested in. Based on user selections made in the view panel, the corresponding branch timeline can change.

Jet Stream User Log In and Settings Drop Down Menu

The user login icon in the upper right-hand corner of the screen provides a drop-down menu with options to change your password and/or log out.

Data Container Drop Down Menu

The Container drop-down menu in the upper right-hand region of the screen allows you to change which data container (or data template) is shown in the page. Users can own multiple data containers and can select whichever data containers they want to browse.

Data Container Report Panel (Bottom Half of the Jet Stream Interface)
The Data Container Report Panel consists of a series of tile buttons to help report on activities being completed in the Data Container. They are summarized below as **Summary, Sources, History, Bookmarks, and Capacity**.

### Summary

The **Summary** tile allows you to see an overview identifying what data sources are in the data container, properties associated with the data container, and information about operations performed in the data container.

### Sources

The **Sources** tile in the upper left-hand panel bar provides information about each data source, such as the description, name, and properties that the administrator has placed inside the data container. In particular, you can get the connection information to access them from here.

### History

The **History** tile reveals a list of actions performed in this data container. Using the **filter control** on the upper right-hand side of the page is an easy way to find specific activities completed over time.

### Bookmarks

The **Bookmarks** tile allows you to view and edit details about bookmarks within this data container and bookmarks accessible from it.

### Usage

The **Usage** tile allows you to view information about how much storage capacity this container has used.
Jet Stream Data Concepts

Understanding Data Sources

A "data source" in Delphix can represent a database, an application, or a set of unstructured files. Delphix administrators configure the Delphix Engine to link to data sources, which pulls the data of these sources into Delphix. The Delphix Engine will periodically pull in new changes to data, based on a specific policy. This, in turn, begins building a custom timeline for each data source. Additionally, the Delphix Engine can rapidly provision new data sources that are space efficient copies, allowing users to work in parallel without impacting each other.

Understanding Data Templates

"Data templates" are the backbone of the Jetstream data container. They are created by the Delphix administrator and consist of the data sources you need in order to manage your data playground and your testing and/or development environments. Data templates serve as the parent for a set of data containers assigned to Jet Stream users. Additionally, data templates enforce the boundaries for how data is shared. Data can only be shared directly with other users whose containers were created from the parent data template.

Understanding Data Containers

A Jet Stream data container allows you to access and manage your data in powerful ways. Your data can consist of your application binaries, supporting information, and even the entire database(s) that underlie it. The Jet Stream data container allows you to:

- Undo any changes to your application data in seconds or minutes
- Have immediate access to any version of your data over the course of your project
- Share your data with other people on your team, without needing to relinquish control of your own container
- Refresh your data from production data without waiting for an overworked DBA

A Jet Stream data container consists of one or more data sources, such as databases, application binaries, or other application data. The user controls the data made available by these data sources. Just like data sources in a template, changes that the user makes will be tracked, providing the user with their own data history. The Jet Stream Data Container Interface lets you view the details and status of your data container and its associated data sources, as well as manipulating which data is in those sources. The Data Container Interface includes a section called the Data Container Reporting Panel, which displays details about each source, including the connection information needed to access it - for example, the java database connectivity (JDBC) string for a database. This connection information is persistent and stable for the life of the data container, regardless of what data the resources are hosting.

Jet Stream Data Flow

The Jet Stream data flow diagram below demonstrates how a Jet Stream data user accesses data sources. Data sources are connected to a Delphix Engine, which is controlled by the Delphix administrator. The Delphix administrator will connect all data sources that developers and quality assurance (QA) teams need to a Jet Stream data template. This data template acts as a parent source to create the data containers that the administrator will assign to Jet Stream data users. Data sources flow from the Delphix Engine into a data template and downstream into a data container, where a Jet Stream data user will use the data sources to complete tasks. The data container acts as a self-contained testing environment and playground for the Jet Stream data user. Additionally, Jet Stream data users are able to set, bookmark, and share data points in their container with other Jet Stream data users of other data containers, as long as all the data containers were created from the same parent data template.
Understanding Branches

You can organize data in the data container into task-specific groupings, called "branches." For example, you can use a branch to group all the data you have used while addressing a particular bug, testing a new feature in an application, or exploring a business analytics scenario. By default, Jet Stream automatically creates the first branch of source data for you when you login to Jet Stream for the first time. You can view the default branch and any additional branches that you create over time by clicking the Branch tab. Additionally, to the right of the default branch, you will see an interconnected branch timeline unique to whichever branch is currently active. The illustration below displays both the default branch in the Branch tab of the Data Container View Panel and the default branch timeline.

A branch is used to track a logical task, and contains a timeline of the historical data for that task. One branch is the "active" branch, which means that it is the branch that is currently being updated with new data from the data
sources. At any time, you can change which branch is active and thus change which data is in the associated data sources.

Understanding Timelines

Branch Timeline

A branch timeline acts as a dynamic point-in-time interface for user actions within the branch. You can interact with the source data in the active branch by using both the branch timeline and icons along the Self-Service Toolbar at specific points in time. Common activities include re-setting data sources to run a test, refreshing the data container with the most current source data, and bookmarking data to share or track interesting moments of time along the branch timeline. Users work with one branch at a time to perform a series of actions related to a particular testing or debugging task such as data updates or starting and stopping data. As you work within your data container, you can create more branches over time to run or complete separate tasks. Additionally, the data container tracks each branch and the corresponding actions you perform on the branches. To view the actions completed over the life of a branch, see the container timeline in the Time tab of the Data Container View Panel.

Container Timeline

The Time tab displays the data container's timeline, which acts as a wall clock of time. It shows continuous real time across all branches and timeline segments. You can scroll up and down in the container timeline to find the point of time that interests you.
Clicking on a point in time in the container timeline will display the corresponding branch timeline capturing any actions performed on the branch. Additionally, should you need to select a time between tic-marks, you can use the time input field in the time selector on the left side of the screen.

**Selecting a Point in Time with the Time Selector**

1. In the time selector, type in a date and time with the following format:
   
   Month/Day/Year Hour:Minute:Second{am|pm}. For example: 1/26/2015 1:14:13pm.

2. Press Enter.

The time input field will show the selected time. Now that you have entered the specific time you want, you can use the toolbar to select the data operation that you want performed at this point in time. Data operations can include Create Bookmark, Create Branch, and Restore.

**Note:** If you type in an invalid time value, or a time that is out of range, the value you typed in will revert to the previous default that existed before.

**Selecting a Point in Time with the Time Selector Calendar**

1. Locate and Click on the calendar icon on the left of the input field in the time selector.

2. From the flyout that appears, click the date and select a time that you wish to use.

3. Click on the data operation button on the toolbar that you want to perform at this point in time. Data operations can include Reset, Create Branch, and Create a Bookmark.

**Note:** The flyout will not let you pick a date that is before the first point of data time in the container, or after the present moment.

**Understanding the Self-Service Toolbar**
The Jet Stream **Self-Service Toolbar** contains self-service action icons that represent available actions a Jet Stream data user can perform. You can distinguish between available and unavailable icon actions by the use of color on the toolbar. Actions available to you will be red, and actions that are unavailable will be grey. All actions are dynamic, and availability will change based on how you use and work with data in both the branches and data container(s) that are assigned to you.

Jet Stream Self-Service Toolbar, Version 1.0.0

For example, your options for actions on the **Self-Service Toolbar** can change if the branch of the branch timeline you are working with is activated. In the illustration below, the screen shows a user working in an active branch. Notice the bright red star at the end of the timeline. This indicates that the branch is active. Also notice which actions are and are not available to the user on the **Self-Service Toolbar**.

Jet Stream Self Service Toolbar with a Point In Time selected on an Active Branch Timeline, Version 1.0.0

The **Self-Service Toolbar** is dynamic and will change based on tasks a user performs in Jet Stream. These workflows will influence how and when self service actions become available on the self-service toolbar.

**Branch Timeline Segments**

A branch timeline with segments is a visual representation of actions taken on a branch timeline over a time span. The timeline segments represents data in time that is no longer contiguous once a user clicks **Create Branch, Refresh, Reset**, or **Restore** on the **Self-Service Toolbar**. A vertical bar between each of the segments appears to remind a Jet Stream user that the data in one timeline segment is a completely new data start. In other words, while the data within one segment is logically contiguous, the data is never contiguous across segments. For example, the following image shows a timeline with multiple segments:

Segmented Branch Timeline, Version 1.0.0
As mentioned above, the branch timeline becomes segmented after you have performed a specific action or task, such as **Refresh**. Based on the action, two red bubbles will appear in the time segment. The top bubble indicates where the data used for this action came from, for example the data template, a different branch, or a shared bookmark. The second red bubble appears on the timeline as the actual data stream in a point of time from the parent data. It appears because of actions such as **Refresh**, **Reset**, **Restore**, **Create Branch**, and **Bookmark**. Clicking the second bubble will show you specific details of the action, such as the specifics of the action including its name, the time the action occurred, and the data sources used at a point in time. This is illustrated below:

![Branch Timeline Example](image)

**Working with Multiple Branches and Timelines**

As you work in your data container, you can switch between branches at various times to work on resolving a bug or to test a new application feature. For example, consider what occurs on two different branches in a container:

**Branch 1:**

![Branch 1 Timeline](image)

**Branch 2:**

![Branch 2 Timeline](image)

The Jet Stream user may have actually worked with these branches in the following order over time:

**Branch 1**: Create a branch and use
Branch 2: Create another branch and use

Branch 1: Activate branch, Restore the data source and use

Branch 2: Activate branch and create bookmarks

Branch 2: Refresh the data source from a particular point in time

Branch 2: Reset a branch to the last action (e.g., refresh) on the timeline, and use

In the above illustrations, an individual branch’s timeline shows all actions performed on the branch while the branch was active. The active branch timeline can be interrupted and deactivated when a user chooses to perform actions such as switching to another branch, **Create Branch, Activate**, or **Stop** a data container. Additionally, a user will only be able to view actions on a single branch at a time. A better way to manage multiple branches is to go to the **Time** tab in the **Data Container View Panel**. The **Time** tab allows you to access the **container timeline**, which becomes useful as you toggle back and forth between branches to complete different tasks. The **container timeline** allows you to view all the continuous data points of time, with all actions taken on all branches in a single data container.
Understanding How to Preserve Data in a Point of Time

The following illustration shows that on 8/27/14, at 9:33:09am, data was reset to the parent data branch (master) at 9:28:48am, capturing data points from 9:33:06am.

The black arrows above point to a tick, (representing a point in time) clicked on the branch timeline. This represents the time the **Reset** action was performed on the data container. The red arrows point to when time was captured in a data source using the **Reset** action on the branch timeline. When clicked, the reset bubble provides more details with a flyout, indicating where the data comes from and the time that the data represents. Additionally, the reset bubble detail flip card provides additional information about each data source. Specifically, the blue arrows point to the time used for each data source at this point in the data container. **NOTE:** This does not show the time that was used for each source that pulled the data.

Time represented on the branch timeline varies based on many factors. For example, after selecting a specific point in time on the branch timeline, the Delphix Engine will map that point in time to the closest usable point in time for each data source. Based on the properties of the underlying data sources, these times may be different. Not all data sources track changes at the same granularity, as illustrated below.
While a branch timeline can follow a continuous time flow, the data sources being selected for each time segment may not be continuous.

Understanding Bookmarks

Bookmarks are a way to mark and name a particular moment of data on a timeline. Once created, you can easily locate a bookmark through one of the bookmark viewers in the interface. You can restore the active branch’s timeline to the point of data marked with a bookmark. You can also share bookmarks with other Jet Stream users, which allows them to restore their own active branches to the point of data in your container.

Bookmarks Tab in the Data Container View Panel

The Bookmarks tab is the third tab in the Data Container View Panel within the data container workspace of the Jet Stream interface. It allows you to find a bookmark that is within your data container and view the branch where the bookmark has been placed.

Bookmarks Tile in the Data Container Report Panel

The Bookmarks tile in the Data Container Report Panel allows you to see all bookmarks within your container and all bookmarks that other users have made available to you. Here you can also edit details about bookmarks, create new branches, and restore the active branch to the bookmark’s point of data time.
Bookmark Sharing Permissions

When you first create them, bookmarks are private to your data container, but you can share a bookmark with other Jet Stream data users. Bookmarks that other users have shared with you are called "available" bookmarks. Your bookmarks will only be shared with Jet Stream data users in data containers created from the same data template. This is because all data containers created from the same data template have a compatible set of data sources.

Bookmark Appearance

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Private Bookmark]</td>
<td>A bookmark that is private</td>
</tr>
<tr>
<td>![Shared Bookmark]</td>
<td>A bookmark you have shared</td>
</tr>
<tr>
<td>![Available Bookmark]</td>
<td>A bookmark that has been shared with you</td>
</tr>
</tbody>
</table>

Data Container Storage and Retention for Branches and Timelines

Bookmarks mark a moment of data. Jet Stream will never automatically delete the data marked by a bookmark. However, it may delete data from any time in the past on your branches, depending on the retention policies configured by your administrator. If you select a moment of data that has been deleted, the flyout will indicate that retention has removed data for this point in time.
Working with Data Operations and Sources in a Container

Getting Started

Activity One: How to Start and Stop a Data Container

Starting a Data Container does the following:

- Starts the data sources
  - This means that each data source listed in the Source Details section of the Data Container page will start using CPU and network resources on the target system it is running on
- Makes the data in the active branch available
  - Once the container has been started, the data represented by the active branch is available

Note: By default, when a Jet Stream data user logs in, the data container and data sources are automatically started.

Stopping a Data Container does the following:

- Shuts down the data sources
  - This means each data source listed in the Source Details section of the Data Container page will stop using CPU and network resources on the target system.

1. To start a Data Container, click Start on the Self-Service Toolbar.
2. To stop a Data Container, click Stop on the Self-Service Toolbar.

Working with a Branch, a Branch Timeline, and the Self-Service Toolbar

Activity Two: Using Reset from a Bookmark to Facilitate Destructive Testing

Reset is a Jet Stream data user workflow that is optimized to enable destructive testing. Reset automatically restores the data to the last operation conducted in the data container, which can include creating a bookmark, resetting, or restoring data. As an example, you can do a refresh and then get your data into a state required for testing. Once you are satisfied with the state of your data, you can create a bookmark, which will preserve the data at this point in time. You can then run destructive tests on the data. When you are done, you can select the Reset icon on, which will automatically restore the state of the container to the last operation – in this case, the bookmark. This workflow ensures that each test has a clean copy of the data and is not impacted by the results of other tests. You only need to create a bookmark and click Reset on the Self-Service Toolbar. The following steps indicate how to do this:

Create a Bookmark

1. Select a Data Point on a branch's timeline.
2. Click the Bookmark icon on the Self-Service Toolbar.

- Type a new name in the Bookmark Window.
1. Optionally, fill in a description.
2. Optionally, add one or more tags.
   a. These can be used to help filter a set of bookmarks.
3. Click Create.

After the bookmark has been created, you will see the bookmark icon appear on the timeline. When you click the Reset button, all data will be reset to that point of time.

Reset to Data from a Bookmark

1. Click the Reset icon.

Update Data with Reset, Version 1.0.0

This action reflects the moment of data marked by the closest operation bubble (Refresh, Restore, Reset, or Bookmark) into a new timeline segment on the active branch. It also copies the moment of data into the data sources.

Activity Three: Using Refresh to Get the Latest Data From a Data Template

Start a new timeline segment with the most recent point of data from the Data Container’s Data Template.

1. Click the Refresh icon.
Update Data with Refresh, Version 1.0.0
Refresh creates a new timeline segment on the active branch. This refreshes each source in the data container to the latest data in the corresponding source of the data template.

Activity Four: Using Restore to Return Data Back to a Point in Time
This starts a new timeline segment on the active branch with the selected point of data.

1. Select one of the following:
   a. A point of data on a timeline.
   b. A bookmark on a timeline.
   c. A bookmark under the Bookmarks tile in the Data Container Report Panel.

2. Click the Restore icon.

If you restore data back to a point in time on the data template master timeline, Jet Stream will ask you which data container to restore into. It will then:

- Reflect the selected point of data into a new timeline segment on the active branch
- Copy the moment of data into the data sources

If the timeline segment on a branch timeline was created by a Restore operation, then the segment starts with the moment of data from the branch that was selected when the Restore operation was done. This is illustrated below:

Note: The parent branch for this segment can be the same branch of which this segment is a part. It is possible to restore the active branch from a point in time on the same branch.
Activity Five: Create a New Branch and Switch Between Branches

Developers and QA teams can have multiple branches that can represent data from different points in time or different sources. You have many options for how you create a new branch. These include:

- A **point of data** time on a data timeline within the Jet Stream data container, or
- A **bookmark** bubble on the timeline, or
- A **bookmark** in the Bookmarks tile in the Data Container Report Panel

1. Click the Branch icon to create a new branch.
2. Enter a **name** for the new branch.
3. Click **OK**.
4. Click the **Activate** icon in the Self-Service Toolbar.

If the inactive branch is not showing in the data container workspace:

1. Find the **branch** in the **Branch** tab.

Selection of Branches in Branch Tab, Version 1.0.0

1. Click the **Activate** icon.
2. After a moment, the branch will become active.

Active Branch

Within a single data container, only one branch is active at any given time. The data at the newest end of the active branch’s timeline is the newest copy of the data from the data container’s data sources. The active branch is distinguished by a red star, which appears at the far right of the timeline, alongside its name in the **Branch Name** area, and in the **Branch** tab.

Active Branch Inactive Branch

Activity Six: Rename and/or Delete a Branch

Rename the Default Branch

1. Select the **Default Branch** in the **Branch** tab.
2. Click the **Pencil** icon to the right of the name.
3. Enter the **new name**.
4. Click the **Checkmark** icon.

**Delete a Newly-Created Branch**
1. Select the **branch** in the **Branch** tab.

2. Click the **Delete** icon to the right of the name.

3. Click **Delete** in the confirmation window that appears.

**Working with Bookmarks**

Working with bookmarks is an easy way to share data in points of time with other Jet Stream users assigned to the same data container. By sharing with others, you can integrate testing, development and QA needs. Sharing a bookmark allows users to work with data as they see fit. Bookmarks mark a moment of data. Jet Stream will never automatically delete the data marked by that bookmark.

![Bookmarks Management in the Data Report View Panel, version 1.0.0](image)

**Activity Seven: Share a Bookmark with Other Jet Stream Users**

**Share a Bookmark**
1. Select a **bookmark** by clicking one of the following:
   a. The bookmark's **bubble** on the **branch timeline**.
   b. The **Bookmarks** tab in the data container workspace.
   c. The **Bookmarks** tile in the **Data Container Report Panel**.

2. Click the **Share** icon.

**Note**: You cannot share a bookmark that you or another user have already shared.

**Un-share a Bookmark**
1. Select a **bookmark** by clicking one of the following:
   a. The bookmark's **bubble** on the **branch timeline**.
b. The **Bookmarks** tab in the data container workspace.

c. The **Bookmarks** tile in the **Data Container Report Panel**.

2. Click the **Unshare** icon.

**Note:** You cannot unshare a bookmark that is already private or a bookmark which someone else has shared.

**Delete a Bookmark**

1. Select a **bookmark** by clicking one of the following:
   a. The bookmark's **bubble** on the **branch timeline**.
   b. The **Bookmarks** tile in the **Data Container Report Panel**.

2. Click the **Delete** icon.

**Activity Eight: Editing Bookmarks**

**Rename a Bookmark**

1. Click the **Bookmarks** tile found in the **Data Container Report Panel**. A selection of bookmarks will appear based on whether you have chosen to view private, shared, and/or available bookmarks. Whichever bookmark tile you click, a bookmark tile will appear in the **Data Container Report Panel**.

2. In the **detail bookmarks** window, click the **Edit** icon to the right of its name.

3. Enter the **new name** in the edit field and click the **checkmark** to the right of the field to accept and save the new name.

**Edit the Description of a Bookmark**

1. Select a **bookmark** by clicking the **Bookmarks** tile in the **Data Container Report Panel**.

2. Click the **Edit** icon to the right of its name.

**Activity Nine: Filter and View Bookmarks**

**View Only Your Created Bookmarks**

In the **Bookmarks** tile in the **Data Container Report Panel**, bookmarks that belong to you are shown. To see only your own bookmarks:

1. Click the **Bookmarks** tile in the **Data Container Report Panel**.

2. De-select **Available**.

[Bookmark view options: Private, Shared, and/or Available. JetStream Version 1.0.0]
**View Bookmarks You Have Shared with Others**
1. Click the Bookmarks tile in the Data Container Report Panel.
2. De-select Private.
3. De-select Available.
4. Only your shared bookmarks will be shown.

**View Bookmarks That Others Have Shared with You**
1. Click the Bookmarks tile in the Data Container Report Panel.
2. De-select Private.
3. De-select Shared.
4. Select Available.
5. These are the bookmarks that have been shared with you.

**Adding Tags To Your Bookmark**
1. Click the Bookmarks tile in the Data Container Report Panel.
2. Select the bookmark to which you want to add tags.
3. Click Add a Tag.
4. Enter the tag name.
5. Click the Accept icon.

Your tags will be shown at the bottom of the Bookmarks tile in the Data Container Report Panel.

**Finding Bookmarks**
In either the Bookmarks tab in the data container workspace or the Bookmarks tile in the Data Container Report Panel:

1. Type into the Filter field.

This will only show bookmarks that have names or tags that match the text you have entered.
Understanding Jet Stream Usage

Jet Stream Usage Management Dashboard Overview

Jet Stream data templates are comprised of dSources, virtual databases (VDBs), and vFiles. These data sources are controlled by the standard policies configured in the Admin App of the Delphix Engine. As with existing containers, space will be reclaimed by the retention policy over time. As retention cleans up historical data, users will no longer be able to use those points in time to restore or branch.

Jet Stream data containers are comprised of VDBs provisioned from the sources defined in the data template. Similar to VDBs in the existing Admin App, data containers' VDBs will share blocks with the source from which they are provisioned. This prevents the referenced data on the source from being cleaned up by retention. Retention for these VDBs is controlled by the standard Delphix retention policies. As on templates, bookmarks in data containers will prevent storage from being reclaimed by retention. In addition, Jet Stream will ensure that the latest data on each Jet Stream branch is never removed.

The Usage pages of the data templates and data containers provide information that can help you understand how storage is being used, how to reclaim space, and how much space you are able to reclaim.

Container Usage Overview

The Usage Details page, as seen below, shows the space used by data containers provisioned from the template and the bookmarks created on the template.

![Container Usage, Version 4.2](image)

The stacked bar graph shows information about the top 10 space users. You can re-sort the graph based on the fields in the Sort by legend on the top right-hand corner of the screen as seen in the image above. For example, if you want to know which data containers are sharing the most data with others, you can un-select Shared (others data) and Unique by clicking them in the legend.
The Sort by legend, Version 4.2

**Note:** When the legend items are not selected, their corresponding colored boxes turn gray and the data is removed from the chart. The data and name will reappear when you re-select by click on the preferred grayed-out category.

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this data container. This assumes that also delete underlying data sources.
- **Shared (others data)** – The amount of space that cannot be freed on the parent data template (or sibling data containers) because it is also being referenced by this data container due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up the space.
- **Shared (self data)** – The amount of space that cannot be freed on this data container because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks
- **Unvirtualized** – The amount of space that would be used by the data in this container without Delphix virtualization

**Bookmarks Usage Overview**

As shown in the image below, the Container Usage page provides the usage information about bookmarks created on a template. The primary categories of information include **Unique**, **Shared (others data)**, and **Shared (self data)**.

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this bookmark
- **Shared** – The amount of space referenced by this bookmark that cannot be freed by deleting this bookmark because it is also referenced by neighboring bookmarks or branches that have been created or restored from this bookmark
- **Externally Referenced** – The amount of space referenced by this bookmark that cannot be freed by deleting this bookmark because it is also being referenced outside of Jet Stream – for example, by a retention policy.

### Branches Usage Overview

As detailed in the image below, the **Container Usage Details** page shows the usage information about the branches and bookmarks created on a container. The primary categories of information include **Unique, Shared (others data), and Shared (self data)**.

![Branches Usage, Version 4.2](image)

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this branch
- **Shared (others data)** – The amount of space that cannot be freed on the parent data template or sibling branches because it is also being referenced by this branch due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up the space.
- **Shared (self data)** – The amount of space that cannot be freed on this branch because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks.
Jet Stream Resources

Access more resources at http://docs.delphix.com/display/DOCS42/Delphix+Engine+4.1+Documentation
Jet Stream Support

Ask the community for support @ {+}https://community.delphix.com/delphix+. If you are seeing an issue that cannot be resolved with help from the community, contact your in-house Delphix administrator and have them file a support case as appropriate.
Getting Started with Mission Control

Welcome to Delphix Mission Control

Mission Control is a centralized reporting and auditing tool for overseeing multiple Delphix Engine deployments. Mission Control allows administrators to combine, sort, and audit data from multiple Delphix Engines. A centralized view of data from many Delphix Engines allows administrators to identify promptly any issues with the deployment, determine where resources are allocated, quickly locate deployed databases, and properly size future additions.

User Roles and Permissions

Mission Control has two types of users:

Admin User
Admin users have full access to all report data and can configure the Mission Control appliance. For example, they can add/delete Delphix Engines, add/delete reports, add/delete users, change tunable settings, and add/delete tags.

Auditor User
Auditor users can only view report data. Admin users can also assign auditor users a set of tags (arbitrary text strings) to restrict which report data they can view. There is no default auditor account. The first Delphix Administrator will need to create the auditor users and will be responsible for creating their User IDs and Passwords.

System Requirements

The VM guest where you install Mission Control has the following requirements:

- VMware ESX: 4.x or greater
- Two Virtual CPUs
- 4 GB of Memory
- 50 GB of Storage

Mission Control supports Delphix Engine 4.0 or later.

Supported Browsers

Accessing the Mission Control console is supported on the following browser versions:

- Chrome 37
- Safari 7
- Firefox 32
- Internet Explorer 11

Activity One: Install Mission Control

1. Using the vSphere client, log into the vSphere server where you want to install Mission Control.
2. Select File > Deploy OVA Template.

By default, Mission Control is configured to use DHCP to acquire an IP address.
Logging In

1. Access Mission Control by opening a web browser using the IP address or DNS qualified host name. Mission Control does not currently support SSL connections, so you should use http, not https.

2. Mission Control ships with one generic Delphix Admin User. The User ID is "delphix_admin" and the password is "delphix."

Once logged in as the Delphix Admin User, change your password. You can find instructions to do this in the "Change a User Password" section found below.

Navigating the Mission Control Toolbar and User Interface

The Mission Control Toolbar appears after logging into Mission Control. The navigation bar enables you to analyze, manage, and configure data reporting for a Delphix deployment. The View Report functionality appears on the left-hand side of the toolbar. The configuration, help, and login buttons appear on the right-hand side of the toolbar. Below is a screenshot of the toolbar key functionality as well as a brief summary of the functionality available.

Viewing Reports

The View Report tab provides aggregated data across all connected Delphix Engines and presents it as a set of different reports. You can select these reports from the drop-down menu. Mission Control has automated features that check for updates across all Delphix Engines and sync these updates into reports every 10 minutes. To refresh the currently displayed report manually, click Refresh.
Interactive reports such as **Storage Breakdown** and **History** display interactive graphical representations of historical and current storage usage across all Delphix Engines you are monitoring. These visualizations of storage and disk capacity enable you to analyze and mediate storage across Delphix Engines from multiple perspectives.

**Configure Mission Control**

Clicking on the configuration icon in the upper right-hand corner of the screen navigates you to four configuration tabs, including **Reports, Engines, Users**, and **System**. Read below for more details.
Delphix Engine Configuration

Configure Reports

To navigate to the Report Scripts screen, as seen below:

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click Reports.

The Report Scripts screen provides settings to:

- Enable/disable individual reports to determine which ones are available in the reports drop-down menu
- Delete reports
  - Deleted reports are no longer generated in Mission Control
- Upload new reports
  - This is an experimental feature. Please contact Delphix if you are interested in customizing existing reports or creating new ones.
- Configure tunable parameters for specific reports
  - Click the field in the value column to make it editable
- View current version of Mission Control

Configure Engines

To navigate to the Engines screen, as seen below:

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click **Engines**.

Configure Engines Tab - Version 1.2.

The **Engines** tab lists all Delphix Engines that you have added to Mission Control. The **Status** column shows whether Mission Control is connected to each Engine; it will prompt a specific error message if it is unable to connect. To remove an engine from Mission Control:

1. Click the X icon next to the engine you wish to delete.
2. In the confirmation dialog, click **OK**.

**Configure Users**

To navigate to the **Users** screen, as seen below:

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click **Users**.

Users Tab - Version 1.2.

The **Users** tab displays the set of user accounts that have permission to access Mission Control. You can assign
tags to auditor users to restrict which Delphix Engines and containers they can see. For more information, refer to the "How to Assign Tags" activity in a later section.

Managing the Operating System

Mission Control runs as an open virtual appliance. The underlying operating system, CentOS, can be modified or patched as needed. For best practices, please refer to the official documentation: https://www.centos.org/docs/

*Note: Any changes to files related to Delphix may result in an unusable system. These files are stored in the following locations:*

- /opt/delphix
- /var/delphix
- /var/log/delphix

To manage the appliance or the Operating System or to upgrade, you must have root access. The default root password is "delphix." You should change this as soon as possible.

To navigate to the **System** screen, as seen below:

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click **System.**

![System Tab - Version 1.2](image)

**Activity Two: Add Delphix Engines to Mission Control**

1. Access Mission Control with a supported web browser using its IP address or DNS qualified host name.
   Supported web browsers include Chrome 37, IE 11, Safari 7, and Firefox 32.
2. Click the **Engines** tab in the **Mission Control Toolbar.**
3. Click **Add Engine**.
4. In the **Hostname** field, enter the Delphix Engine's **IP address** or **hostname**.
5. Enter a **username** and **password**. The user must have at least an auditor role on the target Delphix Engine.

![Image of Delphix Engine configuration screen]

Adding a Delphix Engine – Version 1.2.

**Activity Three: Adding Users**

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click **Users**.
3. Click **add user**.
4. Enter a **username** and **password**.
5. Select **auditor** or **admin**.
6. Inform the newly-created user of their user ID and password login credentials.

**Activity Four: Change a User Password**

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click **Users**.
3. Click the **name** in the upper right-hand corner.
4. Click **change password**.
Search and Run Reports

Reports are the heart of Mission Control. Reports gather data that spans all connected Delphix Engines and presents the data in a single location.

Activity Five: Search and View a Report

1. Click View Report.
2. Click the report drop-down menu to reveal a selection of report options, as seen below.

Five new category sections appear to help you quickly locate the report of your choice. These include Engine Activity, Health, Source, Storage, and VDB.

Reports of interest may include the Storage Summary report found under Storage, allowing you to view, compare, and analyze storage usage across Delphix Engines. Other useful reports include Active Faults, SnapSync, and Replication Summary. Below is a summary list of all reporting categories found under View Reports and a description of the reports found under each category.

Engine Activity Reports

Audit Log: Provides a view of all actions that have been performed on all Delphix Engines. This includes both user- and system-executed actions.
Recent Jobs: Covers all jobs across all engines, such as provisioning or refreshing a VDB.

Replication Status: Allows you to validate that replication is running successfully across all Delphix Engines where it has been configured. If replication is not running successfully, it allows you to determine the cause.

Health Reports

Active Faults: Presents a consolidated view of faults across all Delphix Engines, along with suggested actions (in the "Action" column) to resolve the fault. When you have identified and fixed a fault, an administrator can go to the affected engine and mark the fault as resolved through the GUI or CLI.

Engine Summary: A high-level overview report of all Delphix Engines. It contains version and platform information, the number of faults and recent jobs, and critical alerts for each engine.

Recent Alerts: Alerts are events that have occurred, their severity, and the point in time at which they occurred. The Recent Alerts report combines the alerts and their associated information across engines.

Source Reports

dSource Usage: Shows a list of dSources with the following information for each:

- Actual disk capacity the dSource uses
- Unvirtualized capacity – that is, the disk space that would be required if not using Delphix Engines
- Percentage storage saved
- Number of VDBs that are currently provisioned from the dSource
**SnapSync Summary:** Allows you to validate that SnapSync is occurring as expected and to compare the current and average duration of SnapSync operations. The duration of SnapSync operations may vary based on the size of the database, available network bandwidth, and database configuration – for example, whether change block tracking (CBT) is enabled. You can use this report to easily find the dSources for which SnapSyncs take the longest.


**Storage Reports**

**Storage Breakdown**

Using the information displayed with the **Total** button, you can:

- Determine which engines have the most free space and identify good candidates for new dSources/VDBs
- Determine which engines have the least free space, identify which engines need additional storage or require storage to be freed, and identify which engines may require different retention policies
- Determine which engines have the most space used by VDBs and take actions such as refreshing VDBs or removing unneeded VDBs and/or VDB snapshots
- Determine which engines have the most space used by dSources and identify source breakdown to see how capacity is used for dSource data. If needed, you can make appropriate changes to free up space.

Using the information displayed with the **Source** button, you can determine which engines have the most space used for logs and snapshots and modify retention policies or refresh VDBs to release old snapshots.


**Storage History:** Clicking the **Historical** tab summarizes total storage usage of all monitored engines for the past 30 days.

**Storage Summary:** Shows the total, used, and available storage for each engine, which allows you to identify when you are approaching a low-storage availability state.
VDB Reports

**VDB Inventory**: Shows a consolidated list of all virtual datasets (VDBs and vFiles) that have been provisioned from a data source using the Delphix Engine. You can use this report to easily identify where each virtual database is located.

**VDB Refresh Summary**: Allows you to verify that refreshes are happening as expected and to compare the current and average duration of refresh operations.

**VDB Usage**: Shows a list of VDBs with the following information for each:

- Actual disk capacity the VDB uses
- Unvirtualized capacity
- Percentage of storage saved
- Date of most recent refresh operation or, if never refreshed, date when VDB was provisioned
Filter, Organize, and Extract Reports

Tagging

You can tag Delphix Engines in Mission Control with a set of arbitrary text strings. You can then filter reports to show only data from Delphix Engines with a certain tag. You can also use tags to restrict auditor users so that they can only view data from Delphix Engines with that tag.

Activity Six: Apply Tags

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click Engines.
3. Click the space under the Tag headline.
4. Enter any text string.
5. Click OK.

Applying Tags – Version 1.2.

The screenshot below illustrates how to use a tag to filter the kinds of data and reports an Auditor User can access.

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click Users.
3. Click in space under the Tag headline.
4. Enter the tag category configured for the Auditor User.
5. Click OK.
Applying Tags to Users - Version 1.2.

Once you apply the tag filter, the Auditor User will only have access to reports and data associated with that tag.

**Filtering**

Each report contains a free text filter field. Using this filter allows you to search all displayed columns and returns all rows that have at least one match. Examples of report filtering include:

- Identifying certain types of faults
- Identifying all assets related to an engine
- Locating a virtual database by name

**Activity Seven: Extracting Data from Reports**

Once you have selected the report you are interested in viewing, you will be able to extract the report data with a variety of options. Extract and save the report data by clicking **Copy**, **CSV**, **PDF**, or **Print** on the right-hand side. Click the option of your choice to extract and save the report. The screenshot below highlights the sequence of steps to extract reports.
Understanding Interactive Storage Graphs

The View Report drop-down menu on the Mission Control Toolbar includes interactive graphical representations of historical and current storage usage across all Delphix Engines you are monitoring. These include visualizations of Storage Breakdown and Storage History. Storage Breakdown provides different ways of viewing data, including Total to interact with graphs containing the current total storage on all engines and Source to view the breakdown of disk capacity currently used for dSource data. Storage History provides a graphical interface to analyze all historical storage data collected by monitored engines in the past 30 days.

Activity Eight: Viewing Storage Breakdown Reports

Selecting Storage Breakdown allows you to analyze current storage usage by Total and Source.

Working with Total Storage Graphs

Clicking the Total button presents an interactive data visualization of total storage on each Delphix Engine. The bar graphs show the breakdown of all disk capacity between:

- **Source** – The capacity used for dSource data
- **Virtual** – The capacity used for VDBs
- **Free space** – The amount of available free space on the engine

The Category Legend Key on the right-hand side, shown in the screenshot below, helps you see how your current storage is distributed.
To display engines according to a particular category:

1. Click a **category** in the **Category Legend Key**.

The engines will appear in order according to the category you chose to prioritize. In the screenshot above, **Total** has been prioritized.

Using the information displayed with the **Total** button, you can:

- Determine which engines have the most free space and identify good candidates for new dSources/VDBs
- Determine which engines have the least free space, identify which engines need additional storage or require storage to be freed, and identify which engines may require different retention policies
- Determine which engines have the most space used by VDBs and take actions such as refreshing VDBs or removing unneeded VDBs and/or VDB snapshots
- Determine which engines have the most space used by dSources and identify source breakdown to see how capacity is used for dSource data. If needed, you can make appropriate changes to free up space.

**Working with Source Usage Graphs**

Selecting the **Source** button presents an interactive data visualization from a source perspective. The bar graphs show how much of each engine's storage space is used by:

- **Active Source Data** – The capacity used for current active copy of dSources
- **Manually Retained Snapshots** – The capacity held exclusively due to manually retained snapshots
- **Snapshot Retention** – The capacity held due to snapshot retention policy, either exclusively or in combination with manual settings
- **Snapshots for Dependent VDBs** – The capacity of snapshots held due to VDBs that have been provisioned from them, either exclusively or in combination with policy or manual settings
- **Log Retention** – The capacity held due to LogSync retention policy.

The **Category Legend Key** on the right-hand side helps you see how your current storage is distributed.
To display engines according to a particular category:

1. Click a category in the **Category Legend Key**.

The engines will appear in order according to the category you chose to prioritize. In the screenshot above, **Active Source Data** has been prioritized.

Using the information displayed with the **Source** button, you can determine which engines have the most space used for logs and snapshots and modify retention policies or refresh VDBs to release old snapshots.

**Activity Nine: Viewing Storage History Reports**

Clicking the **Storage History Report** visually summarizes total storage usage of all monitored engines for the past 30 days. In this tab, a line graph appears with the historical storage data for the selected Delphix Engine(s) and a drop-down list of Delphix Engines from which to choose, as seen in the screenshot below.

By default, the above graph shows historical details of the top five engines, based on the most recent data point. However, you can choose which engines' details to display by selecting it from the drop-down menu.
The screenshot below illustrates using the scroll bar at the bottom to hone in on a particular time and date of capacity use. Use your mouse and hover over interesting points on the graph for specific storage information. A rollover box will appear with specific information.

Specific Historical Storage Capacity Details, Mission Control version 1.2.
Mission Control Maintenance

Activity Ten: Self-Service Upgrade of Mission Control

Upgrading Mission Control

When a new version of Mission Control is available, download the upgrade script from Delphix.

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click System.
3. Scroll down to the Upgrade section.
4. Click Choose file.
5. Select the upgrade script.
6. Click Upload & Install.

Activity Eleven: Generate and Upload MC Support Bundles

You only need to do this when Delphix support requests that you send a Mission Control support bundle.

1. Click the configuration icon on the right-hand side of the toolbar.
2. Click System.
3. Scroll down to the Support section.
4. Enter the case number if provided by Delphix support.
5. Click Submit.
Mission Control Support

To file support requests from the support portal please go to [https://support.delphix.com](https://support.delphix.com). Additional support is available with the Delphix community at [https://community.delphix.com/delphix under the "Mission Control" category](https://community.delphix.com/delphix under the "Mission Control" category).
Command Line Interface Guide

- Command Line Interface Overview
  - Connecting to the CLI
  - CLI Contexts
  - Managing Objects
  - Managing Properties
  - Array Properties
  - Untyped Object Properties
  - CLI Automation
- Delphix Objects
  - Object Type Hierarchy
  - Object Names and References
  - Databases and Environments
  - Asynchronous Jobs
- Command Reference
  - CLI Help and Display Commands
  - CLI Context Commands
  - CLI Object Commands
  - CLI Property Commands
  - CLI Miscellaneous Commands
- CLI Cookbook: Common Workflows, Tasks, and Examples
  - CLI Cookbook: Authentication and Users
    - CLI Cookbook: Configuring Key-Based SSH Authentication for Automation
    - CLI Cookbook: Third Party SSH Key Generation Example
  - CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users
- CLI Cookbook: System Administration
  - CLI Cookbook: Configuring A Second Network Interface
  - CLI Cookbook: Adding A Static Route
  - CLI Cookbook: Changing the Default Group Name
  - CLI Cookbook: How to Change a Delphix User Password
  - CLI Cookbook: Creating Alert Profiles
- CLI Cookbook: Hosts and Environments
  - CLI Cookbook: Adding a UNIX Host
  - CLI Cookbook: Adding a SQL Server Source Environment
  - CLI Cookbook: Setting Multiple Addresses for a Target Host
  - CLI Cookbook: How to Change Environment User
- CLI Cookbook: Source Databases and dSources
  - CLI Cookbook: Detaching and Attaching a dSource
  - CLI Cookbook: Detaching and Attaching an Oracle dSource
  - CLI Cookbook: Disabling LogSync for a dSource
  - CLI Cookbook: Enabling Oracle Validated Sync
  - CLI Cookbook: Linking a Microsoft SQL Server Database Loading from a Specific Full Backup of the Source Database
  - CLI Cookbook: Linking a Microsoft SQL Server Database Loading from the Last Full Backup of the Source Database
• **CLI Cookbook: VDBs**
  • CLI Cookbook: Changing the SID of Oracle RAC VDBs
  • CLI Cookbook: Oracle VDB Migration
  • CLI Cookbook: Provisioning a Single Instance Oracle VDB
  • CLI Cookbook: Provisioning a SQL Server VDB
  • CLI Cookbook: Provisioning a VDB from a TimeFlow Bookmark
  • CLI Cookbook: Rolling Forward a VDB
  • CLI Cookbook: Refresh a VDB from a Specific Timepoint or Latest
  • CLI Cookbook: Repairing a Timeflow
  • CLI Cookbook: Changing SGA Parameter
  • CLI Cookbook: Rolling Back a VDB
  • CLI Cookbook: Creating a VDB Config Template
  • CLI Cookbook: Creating a Policy
  • CLI Cookbook: Provisioning a SAP ASE VDB
  • CLI Cookbook: Taking a Snapshot
  • CLI Cookbook: Determining the Snapshot used to provision a VDB

• **CLI Cookbook: Replication**
  • CLI Cookbook: Adding a Replication Spec
  • CLI Cookbook: Deleting a Replication Spec
  • CLI Cookbook: Failing Over a Namespace
  • CLI Cookbook: Triggering Immediate Execution of a Replication Spec
Command Line Interface Overview

This topic provides an overview of the Delphix Engine command line interface, and links to additional topics.

The Delphix Engine provides a native command line interface (CLI) accessible over SSH. This CLI provides an interactive layer on top of the public web service APIs, and is intended for users that wish to automate interactions with the Delphix Engine, or simply prefer a text based interface. All of the functionality available in the CLI is also available through the public stable web service APIs should more full-featured automation be required. For more information on automation using the web service APIs, see the Web Service API Guide.

The CLI has an internal help system and supports tab completion to help guide users. Running the help command will display a list of valid commands and properties, if applicable. Specifying the command or property as an argument to help will display more specific information about that command or property. This guide serves as an overview of CLI operation and examples of some basic tasks, and is not a reference for all CLI commands or properties. As the CLI content is identical to the public web services, complete information about particular commands, properties, or other operations can be found in the API documentation delivered with each server instance, found at:

`http://<server>/api`

The API documentation is guaranteed to be consistent with the set of APIs exported by that particular server. All of the APIs used by the GUI will be supported by the CLI, though as of version 3.0 this conversion is still in progress. While all the database and environment APIs are available, most of the system-oriented APIs (such as those required to do initial setup) will be made available in a later release.

- Connecting to the CLI
- CLI Contexts
- Managing Objects
- Managing Properties
- Array Properties
- Untyped Object Properties
- CLI Automation
Connecting to the CLI

This topic describes how to connect to the Delphix Engine command line interface.

The CLI is available over SSH or the terminal console on any Delphix Engine version 3.0 or later. To connect, use any SSH client appropriate for your workstation environment and connect to the Delphix Engine by IP or hostname on the standard SSH port (22). Enter a username for either a domain or system user followed by the namespace appropriate to that user (either DOMAIN or SYSTEM). For example:

- `ssh delphix_admin@DOMAIN@delphix-server.example.com`
- `ssh sysadmin@SYSTEM@delphix-server.example.com`

At the prompt, enter your user password. Once connected, you will be placed at the CLI prompt:

```
  delphix>
```

While both delphix_admin and sysadmin produce the same prompt once logged in, be aware that the two users have different menus and different functional areas.

---

<table>
<thead>
<tr>
<th>Sysadmin Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>delphix&gt; ls</td>
</tr>
<tr>
<td>Children</td>
</tr>
<tr>
<td>network</td>
</tr>
<tr>
<td>service</td>
</tr>
<tr>
<td>storage</td>
</tr>
<tr>
<td>system</td>
</tr>
<tr>
<td>user</td>
</tr>
<tr>
<td>Operations</td>
</tr>
<tr>
<td>version</td>
</tr>
<tr>
<td>Operations</td>
</tr>
<tr>
<td>version</td>
</tr>
<tr>
<td>delphix&gt;</td>
</tr>
</tbody>
</table>
Individual commands passed as arguments to the SSH client will be interpreted as if they had been read from the terminal. More complex scripts can be passed as input to the SSH command. When running SSH in non-interactive mode via these mechanisms, the command line prompt will be suppressed, as will terminal font decorations such as underline and bold.

The CLI is also available from the serial terminal console should the network be unavailable. Consult your VM platform documentation for information on how to connect to the terminal console. Once connected, log in using your Delphix user credentials just as you would over SSH.

If the management service is unavailable due to a software bug or other problem, the CLI can still be accessed as a system user provided that user is locally authenticated (not via LDAP) and has logged in at least once before. While in this state, only the system commands are available, including restart, which will attempt to restart the management service without rebooting the entire server. If this problem persists, please contact Delphix support.

The topic [CLI Cookbook: Configuring Key-Based SSH Authentication for Automation](#) shows an example of how to connect to the CLI using SSH key exchange instead of the standard password-based authentication.
**CLI Contexts**

This topic explains the concept of contexts within the Delphix Engine command line interface.

The CLI is built on the concept of modal “contexts” that represent an administrative point for interacting with the web service APIs. These contexts can be divided into the following types:

<table>
<thead>
<tr>
<th>Context</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Children</td>
<td>These contexts exist for the purpose of navigating between points in the hierarchy, but have no properties of their own and do not correspond to any server side object. The root context is an example of this, as are most of the top level contexts such as database or group.</td>
</tr>
<tr>
<td>Object</td>
<td>These contexts represent an object on the server, either a specific object (such as databases) or system wide state (such as SMTP configuration). These contexts have properties that can be retrieved via the get command.</td>
</tr>
<tr>
<td>Operation</td>
<td>These contexts represent a request to the server. Commands may or may not require input and may or may not change state on the server, but in all cases require an explicit commit operation to execute the command. When in command context, the prompt includes a trailing asterisk (*) to indicate that commit or discard is required before exiting the context.</td>
</tr>
</tbody>
</table>

User can move between contents by typing the name of the context. To move to a previous context, the `up` or `back` commands can be used. In addition, the CLI supports UNIX-like aliases for `cd` and `ls`, allowing navigation similar to a UNIX filesystem. For more information on these commands, see the [Command Reference](#) section.
Managing Objects

This topic describes the use of objects in the Delphix Engine command line interface, and provides a list of the object management operations.

The Delphix Engine represents state through objects. These objects are typically managed through the following operations, covered in more detail in the Command Reference topics.

The topic CLI Cookbook: Changing the Default Group Name illustrates the use of object management commands such as list and get.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>For a given object type (represented by a static context such as database), list the objects on the system, optionally constrained by some set of attributes. Some objects are global to the system and do not support this operation.</td>
</tr>
<tr>
<td>select</td>
<td>Select a particular object by name to get properties or perform an operation on the object. See the “Delphix Objects” section for more information on object naming.</td>
</tr>
<tr>
<td>get</td>
<td>Display all or some of the properties of an object after selecting it.</td>
</tr>
<tr>
<td>update</td>
<td>Enter a command context to change one or more properties of an object after selecting. Not all objects support this operation, and only properties that can be edited are shown when in the update command context.</td>
</tr>
<tr>
<td>create</td>
<td>Create a new instance of the object type from the root static context. Not all objects can be created in this simplified fashion. Databases, for example, are created through the link and provision commands.</td>
</tr>
<tr>
<td>delete</td>
<td>Deletes an object that has been selected. Not all objects can be deleted.</td>
</tr>
</tbody>
</table>

In contexts where there are multiple objects of a given type, the list command can be used to display available objects, and the select command can select an object for subsequent operation.

When listing objects, each context has its own set of default columns to display. The display option can be used to control what columns are displayed to the user. This is a comma-separated list of property names as they would be retrieved by the get command. It is possible to specify properties that do not exist in order to accommodate lists of objects of varying types, and untyped objects.

The topic CLI Cookbook: Listing Data Source Sizes provide an example of using the list command.
Managing Properties

This topic describes the use of properties in relation to objects in the Delphix Engine command line interface.

Object properties are represented as a hierarchy of typed name/value pairs. The `get` command by itself will display the complete hierarchy for a particular object. This hierarchy is displayed with each nested object indented by an additional level. The set of available properties depends on the command context, and may change if the type of an object is changed.

Property State

Properties are typically set to a specific value, but they can also be `unset`. Unset properties indicate there is no known value, either because it hasn’t been provided yet, or it has been explicitly removed. Properties in this state are displayed via the following means:

- `(unset)` – The property is not currently set. It may never have been given a value or it may have been explicitly unset through the `unset` command.
- `(required)` – This has the same underlying semantics as `(unset)`, but indicates that the property must be set before the current command can be committed. Failure do so will result in a validation error at the time the commit operation is attempted. Required properties are displayed in bold.

In addition, all objects have a default state when in command context. A property that has been modified is noted with an asterisk (*), and can be reverted to its default state through the `revert` command.

When updating properties, only those properties are sent to the server. The exception is arrays and untyped objects, covered in **Array Properties** and **Untyped Object Properties**. These objects are always sent in their entirety, so changing any one element will send the entire object.

Basic Properties

Most properties are displayed and input as a string, though the underlying type may be more specific. The following are some of the basic types:

- **String** – An arbitrary string. This may be subject to additional validation (such as an IP address) that is enforced at the time the property is set.
- **Number** – An integer number.
- **Boolean** – Either “true” or “false”.
- **Enumeration** – A string that must be chosen from a known set of options.

Nested Properties

Some properties are in fact other objects, and are represented as a nested set of properties. These properties can be manipulated in one of two ways: by specifying a dot-delimited name, or changing the context via the `edit` command.

A dot (.) in a property name indicates that the portion to the left of the dot is the parent object name, and the portion to the right is a child of that object. For example, `sourcingPolicy.logsyncDisabled` denotes the `logsyncDisabled` property within the `sourcingPolicy` property. These dots can be arbitrarily nested. An alternative syntax of using brackets to enclose property names (`sourcingPolicy[logsyncDisabled]`) is also supported for familiarity with other programming languages.

The `edit` command, in contrast, will change the current context such that all properties are relative to the specified object. This can be useful when changing many nested properties at once, or when the complete set of properties...
can be confusing to manage all at once.

The topic [CLI Cookbook: Disabling LogSync for a dSource](#) provides an example of manipulating nested properties.
Array Properties

This topic describes the use of array properties in the Delphix Engine command line interface.

Some Delphix objects represent properties as arrays. Arrays are effectively objects whose namespace is a contiguous set of integers. While they behave like objects and their properties can be referenced via the same object property notation, they differ in several key areas.

Arrays can be divided into two types: arrays of primitive types (strings, integers, etc.) and arrays of objects. Arrays of objects can be managed like other objects via nested property names and the `edit` command, but differ in the following respects:

- When an array element is `unset`, it removes the element from the array and shifts all other elements down to preserve the contiguous index space.
- New array elements can only be appended to the end of the array by specifying an index that is one more than the maximum index of the array.
- When displaying a property that is an array, if the length is greater than 3, then it is displayed only as “[ … ]”. The complete contents of the array can be displayed by getting or editing that particular property.

Arrays of primitive types can be managed as arrays of objects, but also support an inline notation using comma-separated notation. This allows single-element arrays to be set as a standard property, and for arrays of strings to be set on a single line instead of having to edit each element.

Regardless of element type, arrays are sent as complete objects when updated. When any array element is changed and subsequently committed, the complete array is sent to the server. When a single array element is reverted, the entire contents of the array are reverted.

The topic [CLI Cookbook: Setting Multiple Addresses for a Target Host](#) provides an example of working with a property that is an array of strings.
Untyped Object Properties

This topic describes the use of the \texttt{type} field in the Delphix Engine command line interface object model, and the use of untyped objects.

Most Delphix objects are typed, meaning they have a \texttt{type} field that controls what properties are available and their types. Object types and their associated hierarchy are described in more detail in the topic \textit{Object Type Hierarchy} topic. In contrast, some properties are “untyped” objects, which means that there are no constraints on the property namespace, and all properties are plain strings. These objects are used for database configuration templates and other scenarios where the property namespace is unbounded or under the control of the user.

Untyped objects are always sent in their entirety when making updates. This means that when any one value is changed and then committed, all values are sent. In addition, when reverting a single value within an untyped object, the entire parent object is reverted to its default state.
CLI Automation

This topic describes using automation with both the Delphix Engine command line interface and the API web service.

The CLI is designed for simple automation, while the web services API upon which it is built is designed for more complex automation. The CLI provides facilities for both enabling simple automation and understanding the underlying infrastructure to aid in the development of other web service API consumers.

The default text output of the CLI is unstable. Any attempt to parse the output is certain to run into difficulties in repeatable results for unknown input, as well as instability as the text output is changed in subsequent releases. To alleviate this problem, the CLI provides an option \texttt{(setopt format=\texttt{json})} that changes the CLI to output all results in parseable JSON (javascript object notation). The JSON format has wide support in a variety of programming languages; see \url{http://www.json.org} for more information.

JSON output is useful when the automation is primarily focused on consuming output. With no built-in scripting functionality to parse and take action on this output, building robust SSH-based automation can be slow and overly complicated due to the many round trips needed to control the automation logic. In these cases, the web service APIs are the preferred method for building complex automation. The web service APIs are covered in more detail in the \texttt{Web Service API Guide}, but the CLI provides one useful option \texttt{(setopt trace=true)} that will display the underlying HTTP calls being made with each operation and their JSON payload. This allows users to experiment with the CLI to leverage its integrated property facilities, but then copy the underlying API content to a custom system for better control over behavior.

Both the web services and API are versioned to support backwards compatibility. Future Delphix versions are guaranteed to support clients that explicitly express a version provided the major version identifier is compatible. For more information, see the \texttt{Web Service API Guide}. The CLI will always connect with the latest version, but the version command can be used to both display the current version and explicitly bind to a supported version. Users building a stable set of scripts can run \texttt{version} to get the current version. Scripts can then run the \texttt{version <id>} command to guarantee that their scripts will be supported on future versions. For more information on the different API versions, see the \texttt{API Version Information} section.
Delphix Objects

These topics describe the object model for the Delphix Engine command line interface.

The Delphix object model is a flexible system for describing arbitrary hierarchies and relationships of objects. In order to enable current and future functionality of the system, the relationship between objects is not always immediately obvious. The CLI is merely a veneer atop the web services layer to ensure that the full complement of functionality expressed by the API is always available, but this requires users to have some understanding of how objects are represented in the system.

- Object Type Hierarchy
- Object Names and References
- Databases and Environments
- Asynchronous Jobs
Object Type Hierarchy

This topic describes the object type hierarchy for the Delphix Engine command line interface.

All Delphix objects have an associated type. This type determines what properties are available for a particular object, the format of those properties, and controls how the system interprets objects and commands. The type hierarchy uses polymorphic inheritance to allow for common properties and behavior to be defined at a single point, while permitting dramatically different types of objects to co-exist without requiring a completely separate API for each. For example, the SourceConfig object is the base type for all external database configurations, but it has children that include OracleSIConfig and OracleRACConfig types that refer to single instance and RAC databases, respectively.

When specifying input types, the system will attempt to determine types appropriate for the current operation, but there are times when the type must be explicitly set, either because the operation supports multiple possible inputs, or the object can embed an abstract type. In these cases, it may be necessary to explicitly set the type property. Setting the type may change the set of visible properties and the resulting validation that is performed, but it will not affect any properties that are already set.
Object Names and References

This topic describes the use of object names and references in the Delphix Engine command line interface.

Most Delphix objects are persistent objects in that they have a well-known identity on the server and associated persistent state. The exceptions are objects used only as input to other operations, or global objects that have persistent state but don’t require any explicit identity since they always exist.

Persistent objects have both a name and a reference. The reference is the canonical identifier for the object, and remains valid even if the object is renamed on the server. It is an opaque token that should never be interpreted by the client; the format may change in future releases though backwards compatibility with current references will be maintained. All web service APIs operate using references. References can be used in the CLI when selecting objects, but given that they are a programmatically generated internal concept, they are difficult for most users to use.

The object name, on the other hand, is a much more convenient way to refer to objects, but suffers from the fact that it is not guaranteed to be globally unique. When displaying or setting references, the CLI will convert to or from the ‘canonical name’ based on the type of the reference and the current set of objects on the system. The canonical name has the form:

<Type>:/<Parent>/<Object>@<Namespace>

The type, parent, and namespace are only included if the local object name conflicts with other objects on the system that would otherwise be valid for the given type specification. Not all objects have names relative to their parent; groups, environments, users, and many other objects are globally unique on the system. This “best fit” method is used both when displaying references as well as when setting properties that are references. If the given name potentially matches multiple objects when attempting to set a reference property, then an error is displayed that includes a list of possible names to clarify which object is being referred to. The conversion from name to reference when setting properties always occurs. Consumers can use references, optionally prefixed with a backtick (‘) character to signify they are references in the unlikely event that someone has created an object with the same name as a valid reference.

Providing unique names for objects without the use of forward slashes (‘/’) and at signs (‘@’) will provide the simplest CLI experience when referencing objects.

Here are some scenarios for databases and groups and their resulting behavior:

No conflicting database name

The local name will be used when displaying references to the object, and can be used when setting references:

set container=example

Databases with the same name in different groups

The parent group name must be used when displaying references to the object and when setting references to the object:

set container=group1/example

Databases with the same name in different namespaces
The namespace name must be used when displaying references to the object and when setting references to the object:

```
set container=example@namespace
```

**Objects of different types but with the same name**

This conflict is exceptionally rare, as the reference context typically constrains the set of possible objects to be a single type, but there are cases (such as alerts, or policy targets) that can be applied to any object. In these cases, the type name must be included to uniquely identify the object:

```
set target=Container:/group1/example
```

In the event that one of the named components contains a slash ('/') or an at sign ('@'), single quotes must be used to disambiguate the name from its parent or namespace.
Databases and Environments

This topic describes the relationship between database container objects and environments in the Delphix Engine object model.

The core Delphix objects revolve around the notion of environments and databases, known at the API layer as containers. Understanding how these objects relate to each other is crucial to operating effectively within the CLI. This section provides an overview of these objects; for more information about a particular representation such as Oracle RAC, see the Web Service API Guide.

Environment Components

An environment is the root of the representation of external state that manages database instances. An environment could be a single host (UnixHostEnvironment) or an Oracle cluster (OracleClusterEnvironment). Environments exist to contain repositories, and each environment may have any number of repositories associated with it. A repository is the entity that contains database instances. Repositories are typically installation directories (OracleInstall) within an environment. Within each repository of any number of SourceConfig objects, which represent known database instances. The source config exists independent of Delphix, and could represent a possible dSource (in which case there is no associated database object), or could be managed entirely by Delphix (for VDBs). The source config contains intrinsic properties of the database instance, while the source (described below) contains information specific to Delphix and only exists when the source config is linked to a dSource or VDB.

Most environment objects are created through the act of discovery. By specifying a host, Delphix will attempt to automatically discover all environments, repositories, and source configs. These objects can also be added manually after the fact in cases where discovery fails.

The environment hierarchy can be represented this way:

```
Environment
  UnixHostEnvironment
      SourceRepository
          OracleInstall
      SourceConfig
          OracleSIConfig
```

The generic type is listed in the top portion of each box, with an example of the Oracle single instance objects in the lower portion of each box. Each of these objects can contain multiple child objects with it.
Database Components

The core of all databases within Delphix is the **Container** that contains all the physical data associated with the database, whether it is a dSource or VDB. Within each container is a **Timeflow**, which represents a single timeline of change within the database history. Currently, a container can only have one timeflow, though this limitation may be relaxed in a future release. Within a timeflow are two important object: **TimeflowSnapshot** objects and **TimeflowRange** objects. Timeflow ranges represent the provisionable ranges within the history of the timeflow, while timeflow snapshot represent a point at which at snapshot was taken and therefore more likely to provision in a short amount of time. The resulting data hierarchy can be represented this way:

```
Container
  OracleDatabaseContainer

Timeflow
  OracleTimeflow

TimeflowSnapshot
  OracleSnapshot
```

Each container may be associated with a **Source**. A source is the Delphix representation of an external database when it is associated with a container, and contains information specific to managing that source. Not all source configs within an environment have a source associated with them (as is the case with linkable databases), but all sources must have a source config. Containers may have no sources associated with them if they are unlined; sources can be manually attached at a later point. Currently, each container can have at most once source associated with it, though this may change in a future release.
Asynchronous Jobs

This topic describes conditions under which command line interface operations may spawn jobs that run in the background, and using the `wait` option to wait for job completion.

Not all operations can be performed in the context of a single web service API call. For cases where there is a long running operation that cannot be executed quickly and transactionally, a job may be dispatched to do the remaining work in the background. For more information on jobs and their semantics, see the topic Viewing Action Status. Within the CLI, any command can potentially result in an asynchronous operation. The default behavior is to wait for any such job to complete, and display its progress in the CLI.

In the event that you do not want to wait for the operation to complete, the global wait option can be set (setopt `wait=false`). If disabled, the CLI will display the reference to any job that was dispatched, but not wait for it to complete.
Command Reference

These topics describe the core built-in commands within the CLI. It is not an exhaustive list of all commands in all contexts. For object or type specific commands, consult the API documentation.

- CLI Help and Display Commands
- CLI Context Commands
- CLI Object Commands
- CLI Property Commands
- CLI Miscellaneous Commands
**CLI Help and Display Commands**

This topic describes help and display commands for the Delphix Engine command line interface.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>children</td>
<td>Display all statically defined children valid for the current context. These children can be targets of the <code>cd</code> command.</td>
</tr>
<tr>
<td>commands</td>
<td>Display all build in commands valid for this context.</td>
</tr>
<tr>
<td>help</td>
<td>Display all commands and properties valid for the current context. Specifying a command or property will provide more information about that command or object. When nested properties are present, only top-level properties are displayed by default, though specifying a particular property will display the entire hierarchy.</td>
</tr>
<tr>
<td>ls</td>
<td>Display children, commands, objects, and operations valid in the current context. Only those sections that are relevant in the current context are displayed.</td>
</tr>
<tr>
<td>operations</td>
<td>Display available context-specific operations. These operations require an explicit <code>commit</code> command to execute the operation, or <code>discard</code> to abort it.</td>
</tr>
</tbody>
</table>
## CLI Context Commands

This topic describes context commands for the Delphix Engine command line interface.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>back</td>
<td>Return to the previous visited valid context. This history only tracks contexts that were actually visited, so running <code>database example</code> followed by <code>back</code> will return you to the root context, not the database (because the two were executed as part of one action and never actually visited). If a previous context was deleted or is no longer valid, this command will skip over it.</td>
</tr>
<tr>
<td>cd</td>
<td>Switch to the given child. This is identical to typing the name of the child itself, but also support UNIX-style directory structures, such as <code>./</code> and <code>../</code>. This allows for contexts to be chained such as <code>cd ../database/template</code>.</td>
</tr>
<tr>
<td>history</td>
<td>Display the history of input to the shell. The shell supports the ability to move back and forth in the history using the up and down arrows.</td>
</tr>
<tr>
<td>up</td>
<td>This is an alias for <code>cd ..</code> for the benefit of those less familiar with UNIX filesystem navigation. Unlike <code>back</code>, which only returns to the previous context only if it was visited, and may return to a child context, this command will always return to the immediate parent context.</td>
</tr>
</tbody>
</table>
### CLI Object Commands

This topic describes object commands for the Delphix Engine Command Line interface.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>list</code></td>
<td>List all objects of a particular type when in the appropriate root context. Different contexts may support different options to the list command to constrain the output; run <code>help list</code> to see possibilities.</td>
</tr>
<tr>
<td><code>select</code></td>
<td>Select an object by name within a list.</td>
</tr>
</tbody>
</table>
## CLI Property Commands

This topic describes property commands for the Delphix Engine command line interface.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commit</td>
<td>When in operation context, commit the changes and execute the operation.</td>
</tr>
<tr>
<td>discard</td>
<td>When in operation context, discard any changes and abort the operation.</td>
</tr>
<tr>
<td>edit</td>
<td>Change the current context to be relative to a particular object property when in operation context.</td>
</tr>
<tr>
<td>get</td>
<td>Get all properties (with no arguments) or a particular property of the current object.</td>
</tr>
<tr>
<td>revert</td>
<td>Revert a particular property to its default value, either the value of the underlying object during an update, or the default command input value.</td>
</tr>
<tr>
<td>set</td>
<td>Set the value of one or more properties. These properties can be specified as <code>name=value</code>, or as simply the property name. When only the property name is specified the CLI will prompt for the value to use, optionally obscuring the input if the property is a password.</td>
</tr>
<tr>
<td>unset</td>
<td>Clear the current value of a property. This is not the same as reverting the property, though this can have semantically identical behavior in the case that the default value is unset.</td>
</tr>
</tbody>
</table>
# CLI Miscellaneous Commands

This topic describes miscellaneous commands for the Delphix Engine command line interface.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>echo</td>
<td>Print the input arguments.</td>
</tr>
<tr>
<td>exit</td>
<td>Exit from the current CLI session. This is equivalent to sending the EOF control character (typically Ctrl-D) or closing your client SSH application.</td>
</tr>
<tr>
<td>getopt</td>
<td>Get the current value of a global configuration option. The list of global options can be retrieved by running <code>help getopt</code>, but include options for controlling JSON output (format), tracing HTTP calls (trace), and enabling synchronous job semantics (wait).</td>
</tr>
<tr>
<td>setopt</td>
<td>Set the value of a global configuration option.</td>
</tr>
<tr>
<td>version</td>
<td>Display the current API version or bind to a particular version. See the <a href="/CLI-Automation">CLI Automation</a> section for more information.</td>
</tr>
</tbody>
</table>
CLI Cookbook: Common Workflows, Tasks, and Examples

This section walks through some of the basic command line interface workflows to add an environment, link to a database, provision a new copy of a database, as well as a few other common operations. This is not meant to be an exhaustive list of possible workflows or cover every possible linking or provisioning scenario. The built-in API documentation should be used once the basics are understood.

⚠️ Case Sensitivity

All CLI parameter names and variables submitted (dSource, environment name, listener, etc) are case sensitive. For example, fullbackupUUID is incorrect and will generate an error, while fullBackupUUID is correct.

- CLI Cookbook: Authentication and Users
- CLI Cookbook: System Administration
- CLI Cookbook: Hosts and Environments
- CLI Cookbook: Source Databases and dSources
- CLI Cookbook: VDBs
- CLI Cookbook: Replication
**CLI Cookbook: Authentication and Users**

These topics describe command line interface procedures for authentication and managing users.

- [CLI Cookbook: Configuring Key-Based SSH Authentication for Automation](#)
- [CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users](#)
CLI Cookbook: Configuring Key-Based SSH Authentication for Automation

This topic describes how to use CLI commands to configure individual users with SSH keys to allow for password-less authentication from a remote host to the CLI in an automated environment.

- What is SSH Key-based Authentication?
- Procedure

What is SSH Key-based Authentication?

Secure SHell (SSH) is a connection method used to log into UNIX or Linux servers remotely. With Delphix, it is used to connect to the Delphix Command Line Interface (CLI) from a remote computer. This normally requires a password on each connection; however, it is possible to use Key-based Authentication to avoid the password requirement and allow automation of Delphix commands.

Key-based Authentication relies on a public/private key pair generated on the client system. The private key allows access to any server acknowledging the matching public key as being authorized to login. In order to configure this, a public/private key pair must be created, and the resulting public key should be added to the Delphix server using the CLI.

Procedure

1. Consult your client documentation for information on generating a public/private key pair. The `ssh-keygen` program is typical on UNIX platforms. If you need details on ssh-keygen usage or have unique requirements (such as named RSA keys), see Third Party SSH Key Generation Example. If you already have a public/private key pair generated on your system, you can skip to step 2.

2. Connect as the user you wish to configure or as a Delphix administrator.

   **Connecting to Namespaces**

   When you connect to the Delphix Engine with the CLI, you should specify the appropriate namespace (either DOMAIN or SYSTEM). See Connecting to the CLI for more information.

3. Select the current user, or select a specific user if configuring another user as an administrator.

   ```
   delphix> user current
   ```

4. Update the user and set the SSH key.

   ```
   delphix user "delphix_admin"> update
   delphix user "delphix_admin" update *> set publicKey
   Enter publicKey:
   ```

   **Wrap the key in double-quotes. The equals sign in the key will cause a syntax error unless the full key is wrapped in double-quotes.**
Avoid Newline Characters with Public Key Entry

The public key value, which can be quite long, must be entered as a single string with no newlines. When copying and pasting the public key be sure to avoid introducing any newline characters.

5. Paste the contents of the public key configured on your client and commit the result.

```
delphix user "delphix_admin" update *> commit
```  
delphix>

6. Verify you can authenticate through the Delphix CLI without a passphrase.

**Example Using Default SSH Key**

```
ssh delphix_admin@DOMAIN@delphix-server.example.com
Last login: Thu Dec 13 22:16:28 2012 from 192.168.0.2
```

delphix>

**Example Using a Non-default SSH Key File Located at path/to/delphix_key**

```
ssh -i path/to/delphix_key delphix_admin@DOMAIN@delphix-server.example.com
Last login: Thu Dec 13 22:16:28 2012 from 192.168.0.2
```

delphix>
Third Party SSH Key Generation Example

This topic describes Example SSH key generation for password-less login to the Delphix CLI

Prerequisites

These operations are performed as a command line user on a non-Dephix host, where SSH is installed. In the remainder of the document we will use:

- **username** - to refer to the existing command line user the non-Delphix host
- **host name** - to refer to the existing non-Delphix host

These example here should work with a variety of SSH distributions, however your distribution may behave differently. If you are unable to follow these instructions successfully, consult with your system administrator, and/or your operating system or SSH client vendor.

Procedure

This example details three ways to set up password-less authentication:

1. Using an existing SSH key with no passphrase
2. Generating a new SSH key stored in the default location
3. Generating a new SSH key stored in a non-default location

Using an Existing SSH Key With No Passphrase

Note: To use an existing SSH key for password-less authentication, that key must have been established with no passphrase. If the existing SSH key has a passphrase associated with it, instead follow the instructions for generating a new SSH key stored in a non-default location below.

1. Log in as **username** to **hostname**
2. Within the **~username/.ssh/** directory there will be one or more files whose names end in the **.pub** extension, for example:
   a. **id_dsa.pub**
   b. **id_rsa.pub**
   c. **identity.pub**

   (the differing names correspond to different types of keys SSH may use)
3. Input the contents of the desired **.pub** file into the publicKey field as described in [CLI Cookbook: Configuring Key-Based SSH Authentication for Automation](

Generating a New SSH Key Stored in the Default Location

1. Log in as **username** to **hostname**
2. Verify there are no files with the **.pub** extension in the **~username/.ssh/** directory. If there are you must either use the existing SSH key as described above, or generate a new SSH key in a non-default location as described below.
3. Choose the type of key you wish to create: RSA or DSA for SSH protocol 2, or RSA for SSH protocol 1. If you have no preference, RSA for SSH protocol 2 is most commonly used.
4. Create your keys with no passphrase, by running:
   a. For RSA protocol 2: `ssh-keygen -t rsa`
      i. This command creates two new files in **~username/.ssh/id_rsa** and **id_rsa.pub**
b. For DSA protocol 2: `ssh-keygen -t dsa`
   i. This command creates two new files in `~username/.ssh/id_dsa` and `id_dsa.pub`

c. For RSA protocol 1: `ssh-keygen -t rsa1`
   i. This command creates two new files in `~username/.ssh/identity` and `identity.pub`

5. You will be **prompted to enter a passphrase**, and to confirm the passphrase - **simply hit enter twice** to create a key with no passphrase for password-less authentication.

6. Input the contents of the created `.pub` file in the `~username/.ssh/` directory into the `publicKey` field as described in **CLI Cookbook: Configuring Key-Based SSH Authentication for Automation**

**Generating a New SSH Key Stored in a Non-Default Location**

1. Log in as `username` to `hostname`

2. Chose the filename for your new SSH key files, for example: `delphix_key` and `delphix_key.pub`

3. Choose the type of key you wish to create: RSA or DSA for SSH protocol 2, or RSA for SSH protocol 1. If you have no preference, RSA for SSH protocol 2 is most commonly used.

4. Create your keys with no passphrase, by running:
   a. For RSA protocol 2: `ssh-keygen -t rsa -f delphix_key`
   b. For DSA protocol 2: `ssh-keygen -t dsa -f delphix_key`
   c. For RSA protocol 1: `ssh-keygen -t rsa1 -f delphix_key`

   This command creates two new files `delphix_key` and `delphix_key.pub`

5. You will be **prompted to enter a passphrase**, and to confirm the passphrase - **simply hit enter twice** to create a key with no passphrase for password-less authentication.

6. Input the contents of the created `.pub` file in the `~username/.ssh/` directory into the `publicKey` field as described in **CLI Cookbook: Configuring Key-Based SSH Authentication for Automation**

7. On subsequent login attempts to the Delphix CLI, specify the private SSH key to be used with the `-i` argument, for example:
   a. `ssh -i ~/delphix_key username@delphixserver.company.com`

**Related Links**

**CLI Cookbook: Configuring Key-Based SSH Authentication for Automation**
CLI Cookbook: Setting Up SSH Key Authentication for UNIX Environment Users

This topic describes adding public key authentication for a UNIX environment user, thus allowing the Delphix server to connect to your UNIX Environments without an explicit password. This method uses the Delphix CLI in order to set up the environment user and gather SSH public keys. It is also possible to perform these actions in the Delphix Engine Admin interface by navigating to Manage > Environments and selecting Public Key as the Login Type for the environment (see Managing Environments with Agile Data Masking for details).

UNIX host environments (and Oracle cluster environments) can have users configured to use SSH-key based public key authentication instead of the traditional password authentication method. Within Delphix, there is a per-system SSH public key that can be placed into the ~/.ssh/authorized_keys file of the remote user. Once this has been done, the Delphix environment user can be configured to use the private key instead of an explicit password.

Prerequisites
- You must be able to log into the remote host (or all hosts of an Oracle cluster) and have write access to the ~/.ssh/authorized_keys file within the desired user's home directory.

Procedure
1. Get the current system public key:

```
$ delphix> system get sshPublicKey
ssh-rsa
AAAAB3NzaC1yc2EAAAABIAwAAAQEAse1M7uJX441VPBljhnxB6MZUTx8VF6cupaVATg1201QonIqx291P+Mwp0AWh7C983IDOyDc+AY7RXpcFP9nKksiJnS5cIK6wo9RIiqSnFlx/VXNkTc2/67RVofoiui4W5fuxD4hOIr0rUr7Bgihh9L6hP0nUvS/rusHFJ+ogxGM46mW9aUJUGmLTNaoyW0YU693HLuKch01t4k6o1VGaC0eLjYlgBf025XiicBX62WqVHAnwMinVjAvmfQhirAgCI7gYrd5/PwN1/DC6xyhWuxdZjgA7sSPeRqWY0JHt/x4oiIAp7xTwtxQLKTnFxrQd+14uf6LKxr5g7w== root@delphix
```

2. Add this key (starting with ssh-rsa) to the remote user's ~/.ssh/authorized_keys file. You will need to get access to this file using an alternate authentication mechanism (such as logging in as the user with a password, or logging in as an administrator). Depending on the target OS, you may need to do the following:
   a. If the directory does not exist:

```
$ mkdir ~/.ssh
```

   b. If creating the file or directory as an administrator:

```
# chown -R <username> <home>/ssh
```

   c. If required by the host SSH configuration, ensure the directory is not world readable:

```
$ chmod 600 ~/.ssh/authorized_keys
$ chmod 755 ~
```

3. Create a new environment user:

```
$ delphix> environment user create
```

4. Set the user environment and name:
5. Set the user credential type to `SystemKeyCredential`:

```
$ delphix environment user create *> set credential.type=SystemKeyCredential
```

6. Commit the results:

```
$ delphix environment user create *> commit
```

**Related Topics**

- [Managing Environments with Agile Data Masking](#)
- [Managing Environment Users](#)
CLI Cookbook: System Administration

These topics describe various system administration tasks that can be performed with the command line interface, such as changing the name of the <default> group and setting up network connectivity.

- CLI Cookbook: Configuring A Second Network Interface
- CLI Cookbook: Adding A Static Route
- CLI Cookbook: Changing the Default Group Name
- CLI Cookbook: How to Change a Delphix User Password
- CLI Cookbook: Creating Alert Profiles
CLI Cookbook: Configuring A Second Network Interface

This topic describes how to configure a static IP address on a second network interface.

**Procedure**

1. Add a VMXNET3 virtual network adapter to the Delphix VM and reboot the VM.
   A reboot is required because the Delphix Engine does not dynamically recognize newly added network devices.

2. Log in to the Delphix Engine as the sysadmin user and switch to the network interface context. Then use the `list` command to view the available network interfaces, and select the new interface to be configured.

   ```
   delphix network interface> list
   NAME
   vmxnet3s0
   vmxnet3s1
   delphix network interface> select vmxnet3s1
   fender.talisker network interface "vmxnet3s1"> get
   type: NetworkInterface
   name: vmxnet3s1
   addresses: (empty)
   device: vmxnet3s1
   macAddress: 0:c:29:e5:4c:c1
   mtu: 1500
   mtuRange: 60-9000
   reference: NETWORK_INTERFACE-vmxnet3s1
   state: DOWN
   ```

3. Run the `update` command and configure a static address.

   ```
   delphix network interface "vmxnet3s1"> update
   delphix network interface "vmxnet3s1" update *> edit addresses.0
   delphix network interface "vmxnet3s1" update addresses.0 *> set
   address=10.1.2.3/24
   delphix network interface "vmxnet3s1" update addresses.0 *> get
   type: InterfaceAddress (*)
   address: 10.1.2.3/24 (*)
   addressType: STATIC (*)
   ```

4. Commit the operation.
delphix network interface "vmxnet3s1" update addresses.0 *> commit
delphix network interface "vmxnet3s1"> get
  type: NetworkInterface
  name: vmxnet3s1
  addresses:
    0:
      type: InterfaceAddress
      address: 10.1.2.3/24
      addressType: STATIC
      state: OK
      device: vmxnet3s1
      macAddress: 0:c:29:e5:4c:c1
      mtu: 1500
      mtuRange: 60-9000
      reference: NETWORK_INTERFACE-vmxnet3s1
      state: OK
CLI Cookbook: Adding A Static Route

This topic describes how to add a static route.

**Procedure**

1. Log in to the Delphix Engine as the sysadmin user and switch to the network route context.

```
   delphix network route> list
   DESTINATION      GATEWAY       OUTINTERFACE
   default          172.16.0.1    vmxnet3s0
   10.1.2.0/24      -             vmxnet3s1
   172.16.0.0/24    -             vmxnet3s0
```

   Run the `add` command to add a new route.

```
   delphix network route> add
   delphix network route add *> set destination=192.168.11.0/24
   delphix network route add *> set gateway=10.1.2.1
   delphix network route add *> get
   type: NetworkRoute
   destination: 192.168.11.0/24 (*)
   gateway: 10.1.2.1 (*)
   outInterface: (unset)
```

   **Optional outInterface Property**

   Setting the `outInterface` property is optional, as the system will automatically determine the output interface based on the gateway address provided, as shown below.

2. Run the `add` command to add a new route.

```
   delphix network route> add
   delphix network route add *> set destination=192.168.11.0/24
   delphix network route add *> set gateway=10.1.2.1
   delphix network route add *> get
   type: NetworkRoute
   destination: 192.168.11.0/24 (*)
   gateway: 10.1.2.1 (*)
   outInterface: (unset)
```

3. Commit the operation.

```
   delphix network route add *> commit
   delphix network route> list
   DESTINATION      GATEWAY       OUTINTERFACE
   default          172.16.0.1    vmxnet3s0
   10.1.2.0/24      -             vmxnet3s1
   172.16.0.0/24    -             vmxnet3s0
   192.168.11.0/24  10.1.2.1      vmxnet3s1
```
CLI Cookbook: Changing the Default Group Name

This topic describes how to change the name of the default group `<New Group>` on the Delphix Engine as a simple example of CLI interactions. You must have delphix_admin credentials to perform this procedure.

**Procedure**

1. Switch to the group context and list groups on the system.

   ```
   delphix> group
   delphix group> list
   NAME         DESCRIPTION
   <New Group>  -
   ```

2. Select the default group and show current properties.

   ```
   delphix group> select "<New Group>"
   delphix group "<New Group>" get
   name: <New Group>
   type: Group
   description: (unset)
   reference: GROUP-1
   ```

3. Run the `update` command and set the name.

   ```
   delphix group "<New Group>" update
   delphix group "<New Group>" update *> set name=default
   delphix group "<New Group>" update *> get
   name: default (*)
   type: Group
   description: (unset)
   reference: GROUP-1
   ```

4. Commit the operation.

   ```
   delphix group "<New Group>" update *> commit
   delphix group "default>
CLI Cookbook: How to Change a Delphix User Password

1. ssh into your engine with a user that has Admin privileges
   
   ```
   ssh delphix_admin@delphix
   ```

2. Go to Users and select the User you would like to change the password of
   
   ```
   delphix > user
   delphix user > ls
   delphix user > select example_user
   delphix user "example_user" > ls
   ```

3. Select `updateCredential` to allow you to change password and set new password
   
   ```
   delphix user "example_user" > updateCredential
   delphix user "example_user" updateCredential *> set newCredential.password=<new password>
   ```

4. Commit the operation
   
   ```
   delphix user "example_user" updateCredential *> commit
   ```

Example:

```
ssh delphix_admin@delphixengine
delphixengine > user
delphixengine user > ls

Objects

<table>
<thead>
<tr>
<th>NAME</th>
<th>EMAILADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysadmin</td>
<td>-</td>
</tr>
<tr>
<td>delphix_admin</td>
<td><a href="mailto:no@delphix.com">no@delphix.com</a></td>
</tr>
<tr>
<td>test_user</td>
<td><a href="mailto:no@delphix.com">no@delphix.com</a></td>
</tr>
</tbody>
</table>

Operations

create
current

delphixengine user > select test_user
delphixengine user "test_user" > ls

Properties

    type: User
```
name: test_user
authenticationType: NATIVE
credential:
  type: PasswordCredential
  password: ********
emailAddress: no@delphix.com
enabled: true
firstName: (unset)
homePhoneNumber: (unset)
isDefault: true
lastName: (unset)
locale: en_US
mobilePhoneNumber: (unset)
passwordUpdateRequested: false
principal: test_user
publicKey: (unset)
reference: USER-2
sessionTimeout: 30min
userType: DOMAIN
workPhoneNumber: (unset)

Operations
delete
update
disable
enable
updateCredential
delphixengine user "test_user" > updateCredential
delphixengine user "test_user" updateCredential *> set newCredential.password=<new password>
delphixengine user "test_user" update *> commit
CLI Cookbook: Creating Alert Profiles

This article describes how to create alert profiles.

Delphix generates alerts for different events. Users may want to be notified of events based on certain criteria such as the type of event or severity. An alert profile allows a user or group of users to be notified of the desired event.

Procedure

1. ssh into your engine using your delphix_admin username and password

   ssh delphix_admin@yourdelphixengine

2. Go into your alerts and list the alerts you already have

   delphix > alert
   delphix alert > ls

3. Create your profile

   delphix alert > profile
   delphix alert profile > create
   delphix alert profile create * > ls

4. Set Actions and Severity Filter

   delphix alert profile create *> set actions.0.type=<AlertActionEmailList or AlertActionEmailUser>
   delphix alert profile create *> set actions.0.addresses.0=<email address to send to>
   delphix alert profile create *> set actions.0.addresses.1=<additional email address>
   delphix alert profile create *> set actions.0.addresses.2=<additional email address>
   delphix alert profile create *> ls
   delphix alert profile create *> set severityFilter=
   delphix alert profile create *> set severityFilter=<WARNING|CRITICAL|INFORMATIONAL>
Valid Values for Parameters

actions.0.type:

- **AlertActionEmailList:** This type of alert is used to create an alert for any number of users. When this type is selected, an email address may be defined in each element of the "actions.0.addresses" array as illustrated above.
- **AlertActionEmailUser:** This type of alert is created for the email address of the user currently logged into the command line interface. The "actions.0.addresses" array is not available for this type.

actions.0.severityFilter:

- INFORMATIONAL
- WARNING
- CRITICAL

5. Commit your changes

```
delphix alert profile create *> commit
```

**Example:**

```
ssh delphix_admin@yourengine
delphix > alert
delphix alert> ls
Objects
REFERENCE   TIMESTAMP               TARGETNAME
EVENTTITLE
ALERT-102  2015-01-14T21:00:04.380Z  ASE/pubs2
Job complete
Job complete
ALERT-100  2015-01-14T19:35:32.958Z  ASE/pubs2VDB
Job complete
ALERT-99   2015-01-14T19:35:32.850Z  ASE/pubs2VDB
Job complete
ALERT-98   2015-01-14T19:34:58.744Z  ASE/pubs2
Error during job execution
ALERT-97   2015-01-14T18:12:01.928Z  ASE/pubs2
Job complete
ALERT-96   2015-01-14T18:03:10.664Z  ASE/pubs2
Job complete
ALERT-95   2015-01-14T17:16:07.464Z  ASE/pubs2
Job complete
ALERT-94   2015-01-14T17:15:55.298Z  ASE/market
Job complete
ALERT-93   2015-01-14T17:15:45.995Z  ASE/pubs2VDB
Job complete
Job complete
```
ALERT-91  2015-01-14T16:38:33.719Z  nstacksolase2.acme.com
Job complete
ALERT-90  2015-01-14T15:47:35.005Z  market
Validated sync failed for dSource
ALERT-89  2015-01-14T15:45:40.895Z  pubs2
Validated sync failed for dSource
ALERT-88  2015-01-15T15:02:14.874Z  ASE/market
Job complete
Job complete
Job complete
Job complete
ALERT-84  2015-01-13T07:04.385Z  pubs2
Backup detection failed
ALERT-83  2015-01-12T22:35:18.774Z  pubs2
Backup detection failed
ALERT-82  2015-01-12T11:30:00.063Z  ASE/pubs2VDB
Unable to connect to remote database during virtual database policy enforcement
ALERT-81  2015-01-12T11:30:00.054Z  ASE/pubs2
Unable to connect to remote database during dSource policy enforcement
ALERT-80  2015-01-12T08:38:26.983Z  pubs2
Backup detection failed
ALERT-79  2015-01-12T06:04.34.666Z  pubs2
Validated sync failed for dSource
ALERT-78  2015-01-11T11:30:03.393Z  ASE/pubs2VDB
Job complete
Children
profile
delphix alert> select ALERT-98
delphix alert "ALERT-98" > ls
Properties
  type: Alert
  event: alert.jobs.failed.object
  eventAction: Create the database on the target host.
  eventDescription: DB_EXPORT job for "ASE/pubs2" failed due to an error during execution: Could not find database "pubs2VDB" on target instance "SRC_157_4K", environment "ASE".
  eventSeverity: CRITICAL
  eventTitle: Error during job execution
  reference: ALERT-98
  target: ASE/pubs2
  targetName: ASE/pubs2
  targetObjectType: ASEDBContainer
  timestamp: 2015-01-14T19:34:58.744Z
delphix alert> profile
delphix alert profile> select ALERT_PROFILE-1
delphix alert profile "ALERT_PROFILE-1" > ls
Properties
  type: AlertProfile
  actions:
    0:
      type: AlertActionEmailList
      addresses: sys_admin@acme.com
      format: HTML
      eventFilter: (empty)
      reference: ALERT_PROFILE-1
      severityFilter: CRITICAL,WARNING
targetFilter: (empty)
Operations
delete
update
delphix alert profile> create
delphix alert profile create *> set actions.0.type=AlertActionEmailList
delphix alert profile create *> set actions.0.addresses.0=johndoe@acme.com
delphix alert profile create *> set actions.0.addresses.1=samsmith@acme.com
delphix alert profile create *> set severityFilter=INFORMATIONAL

*The last piece of the alert profile that needs to be configured is the "targetFilter". This is an array so you can define multiple targets. In the following example, there is a dSource named "pubs2" the user wants to define an alert on. If they try to set the filter to just the name of the dSource itself ("pubs2"), it will warn them that this is ambiguous and gives a hint on how to fully qualify it:

```
delphix alert profile create *> set targetFilter=pubs2
The name 'pubs2' is ambiguous, specify one of: [ "ASE/pubs2", "pubs2/pubs2", "SRC_157_4K/pubs2" ].
```

delphix alert profile create *> set targetFilter.0=pubs2/pubs2
delphix alert profile create *> set targetFilter.1=ASE/pubs2

delphix alert profile "ALERT_PROFILE-34" update *> commit

⚠️ Note on names used in the example

SRC_157_4K: Repository (entity containing the database instances)
ASE: Group name
pubs2: Name of individual database instance

The user set the targetFilter to be equal to "pubs2/pubs2" and "ASE/pubs2" because if you review the "TARGETNAME" column from step 1 above, you will see alerts generated for both of these targets.
CLI Cookbook: Hosts and Environments

These topics describe command line interface procedures for working with database hosts and environments.

- CLI Cookbook: Adding a UNIX Host
- CLI Cookbook: Adding a SQL Server Source Environment
- CLI Cookbook: Setting Multiple Addresses for a Target Host
- CLI Cookbook: How to Change Environment User
CLI Cookbook: Adding a UNIX Host

This topic describes the process of adding a UNIX host using the 3.0 command line interface.

Within Delphix, there are both hosts and host environments. A host represents a remote system, but may or may not be a source or target for linking or provisioning. For example, in an Oracle RAC cluster, the cluster environment represents the location of the Oracle installation(s), and while there are hosts within that cluster they are not individually manageable as environments.

Procedure

1. Create a new environment and set the parameter type to be a UNIX host.
   The default is a UNIX host, but for completeness this demonstrates how one would add another type of environment (Oracle cluster or Windows host).

   ```
   delphix> environment create
   delphix environment create *> set type=HostEnvironmentCreateParameters
   delphix environment create *> set hostEnvironment.type=UnixHostEnvironment
   delphix environment create *> set hostParameters.type=UnixHostCreateParameters
   delphix environment create *> set primaryUser.credential.type=PasswordCredential
   delphix environment create *> get
   type: HostEnvironmentCreateParameters (*)
   hostEnvironment: (unset)
   type: UnixHostEnvironment (*)
   name: (unset)
   description: (unset)
   hostEnvironment: (unset)
   hostParameters: (unset)
   type: UnixHostCreateParameters (*)
   host: (unset)
   type: UnixHost
   addresses: (required)
   sshPort: 22
   toolkitPath: (required)
   primaryUser: (unset)
   type: EnvironmentUser
   name: (required)
   credential:
   type: PasswordCredential
   password: (required)
   environment: (unset)
   ```

2. Set the host address.
   The name of the environment is derived from the address used, though you can provide a more descriptive name if desired. The addresses can be DNS names, IP addresses, or a comma separated list of the above.

   ```
   delphix environment create *> set hostParameters.host.addresses=192.168.1.2
   ```

3. Set the toolkit path.
   This is where Delphix will store temporary binaries used while the host is configured as part of Delphix.

   ```
   delphix environment create *> set hostParameters.host.toolkitPath=/var/delphix
   ```

4. Set the username and password to use when connecting over SSH.
This user must have the privileges described in the Delphix Adminsitration Guide. To configure a SSH user, change the credential type to SystemKeyCredential.

```
delphix environment create *> set primaryUser.name=oracle
delphix environment create *> set primaryUser.credential.password
Enter primaryUser.credential.password: ********
```

5. Commit the result.
The environment discovery process will execute as an asynchronous job. The default behavior is to wait for the result, so progress will be updated until the discovery process is complete or fails.

```
delphix environment create *> commit
UNIX_HOST_ENVIRONMENT-4
Dispatched job JOB-39
ENVIRONMENT_CREATE_AND_DISCOVER job started for "192.168.1.2".
ENVIRONMENT_CREATE_AND_DISCOVER job for "192.168.1.2" completed successfully.
delphix>
```
CLI Cookbook: Adding a SQL Server Source Environment

This topic describes how to add a SQL Server source environment using the command line interface.

Since SQL Server source environments do not have the Delphix Connector running on them, you must use a target environment as a proxy when adding source environments. Delphix uses the connector running on the proxy environment to run commands against the source environment. See Setting Up SQL Server Environments: An Overview for more information.

Procedure

Enter these commands through the command line interface:

```
/environment;
create;
set type=HostEnvironmentCreateParameters;
set hostEnvironment.type=WindowsHostEnvironment;
set hostEnvironment.name=<Source environment name>;
set hostEnvironment.proxy=<target host name>;
set hostParameters.type=WindowsHostCreateParameters;
set hostParameters.host.type=WindowsHost;
set hostParameters.host.addresses="<Source host IP address or hostname>";
set primaryUser.name="<domain\username>";
set primaryUser.credential.type=PasswordCredential;
set primaryUser.credential.password=<password>;
commit;
```

Example

The CLI commands for adding source host "mssql_source_1" using target host "mssql_target_1" as proxy and environment user "ad\delphix_user" would be:

```
/environment;
create;
set type=HostEnvironmentCreateParameters;
set hostEnvironment.type=WindowsHostEnvironment;
set hostEnvironment.name="mssql_source_1";
set hostEnvironment.proxy="mssql_target_1";
set hostParameters.type=WindowsHostCreateParameters;
set hostParameters.host.type=WindowsHost;
set hostParameters.host.addresses="mssql_source_1";
set primaryUser.name="ad\delphix_user";
set primaryUser.credential.type=PasswordCredential;
set primaryUser.credential.password="i_am_the_password";
commit;
```
Related Links

- Setting Up SQL Server Environments: An Overview
CLI Cookbook: Setting Multiple Addresses for a Target Host

This topic is an example of using arrays to configure a target host to support multiple IP addresses. The addresses property is an array of strings.

Procedure

1. Select the host to update

   ```
   delphix> host
   delphix host> select example
   delphix host "example"> update
   ```

2. Set the address:

   ```
   delphix host "example" update *> set addresses=192.168.1.1,192.168.1.2
   ```

3. Get the current addresses, both as a string and as an array object.

   ```
   delphix host "example" update *> get addresses
   192.168.1.1,192.168.1.2 (*)
   delphix host "example" update *> get addresses[0]
   192.168.1.1 (*)
   delphix host "example" update *> edit addresses
   delphix host "example" update addresses *> get
   0: 192.168.1.1 (*)
   1: 192.168.1.2 (*)
   ```

4. Commit the result:

   ```
   delphix host "example" update addresses *> commit
   delphix host "example">
CLI Cookbook: How to Change Environment User

1. ssh into your engine using Admin privileges

   
   ```
   ssh delphix_admin@delphix
   ```

2. Go to Environment and find the Environment you would like to update

   
   ```
   delphix > environment
   delphix environment > ls
   delphix environment > select test_env
   ```

3. Select Environment updating and Update

   
   ```
   delphix environment "test_env" > update
   delphix environment "test_env" update *> ls
   ```

4. Set `primaryUser` to new user you would like to use for the Environment

   
   ```
   delphix environment "test_env update" *> set primaryUser=<new user>
   ```

5. Commit the change

   
   ```
   delphix environment "test_env" update *> commit
   ```

Example:
ssh delphix_admin@delphix
delphix > environment
delphix environment > ls

Objects

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td></td>
</tr>
</tbody>
</table>

Children

oracle
user

Operations

create

delphix environment > select Demo
delphix environment "Demo" > update
delphix environment "Demo" update *> ls

Properties

  type: UnixHostEnvironment

  name: Demo

  description:

      primaryUser: delphix

  delphix environment "Demo" update *> set primaryUser=<new user>
delphix environment "Demo" update *> commit
CLI Cookbook: Source Databases and dSources

These topics describe command line interface procedures for working with dSources.

- CLI Cookbook: Detaching and Attaching a dSource
- CLI Cookbook: Detaching and Attaching an Oracle dSource
- CLI Cookbook: Disabling LogSync for a dSource
- CLI Cookbook: Enabling Oracle Validated Sync
- CLI Cookbook: Linking a Microsoft SQL Server Database Loading from a Specific Full Backup of the Source Database
- CLI Cookbook: Linking a Microsoft SQL Server Database Loading from the Last Full Backup of the Source Database
- CLI Cookbook: Linking to a Single Instance Oracle Database
- CLI Cookbook: Listing Data Source Sizes
- CLI Cookbook: Detaching and Attaching a SQL Server dSource
- CLI Cookbook: How to Change Database User Password
CLI Cookbook: Detaching and Attaching a dSource

This topic describes how to attach a dSource to a different data source.

Prerequisites

A dSource can only be attached to a new data source once it has been unlinked.

When attaching an Oracle dSource to a new data source, the new data source must be the same logical database satisfying the following constraints:

- Same dbid
- Same dbname
- Same creation time
- Same resetlogs SCN
- Same resetlogs time
- Same redo stream, where a log must exist with
  - Same sequence
  - Same thread
  - Same end SCN

For Oracle dSources, this procedure can be used to initially link from a standby server that is faster or less disruptive, unlink the dSource, and then attach it to the production server for subsequent incremental SnapSync operations. When you perform the attach operation, you will need the source config name of an unlinked database.

When attaching a SQL Server dSource to a new data source, the new data source must be the same database satisfying the following containts:

- Same database name
- Same recovery fork UUID

Procedure

1. Select the dSource.

   delphix> database "dexample"

2. Run the detachSource command, specifying the currently active source. This step can be skipped if the dSource has already been detached through the GUI.

   delphix database "dexample"> detachSource
delphix database "dexample" detachSource *> set source=dexample
delphix database "dexample" detachSource *> commit

3. Run the attachSource command.

   delphix database "dexample"> attachSource

4. Set the config to point to an unlinked source.

   The following is an example to attach to an Oracle data source:
The following is an example to attach to a SQL Server data source:

```plaintext
delphix database "dexample" attachSource *> set source.config=SQLSERVER/dexample
delphix database "dexample" attachSource *> set source.sharedBackupLocation=\SERVER1\Backups
delphix database "dexample" attachSource *> set pptRepository=SQL2008R2
```

5. Commit the operation.

```plaintext
delphix database "dexample" attachSource *> commit
```
CLI Cookbook: Detaching and Attaching an Oracle dSource

This topic describes how to attach a dSource to a different data source.

Prerequisites

A dSource can only be attached to a new data source once it has been unlinked.

When attaching an Oracle dSource to a new data source, the new data source must be the same logical database satisfying the following constraints:

- Same dbid
- Same dbname
- Same creation time
- Same resetlogs SCN
- Same resetlogs time
- Same redo stream, where a log must exist with
  - Same sequence
  - Same thread
  - Same end SCN

For Oracle dSources, this procedure can be used to initially link from a standby server that is faster or less disruptive, unlink the dSource, and then attach it to the production server for subsequent incremental SnapSync operations. When you perform the attach operation, you will need the source config name of an unlinked database.

Procedure

1. Select the dSource.

   ```delphix> database "dexample"
   ```

2. Run the `detachSource` command, specifying the currently active source. This step can be skipped if the dSource has already been detached through the GUI.

   ```delphix database "dexample"> detachSource
delphix database "dexample" detachSource *> set source=dexample
delphix database "dexample" detachSource *> commit```

3. Run the `attachSource` command.

   ```delphix database "dexample"> attachSource```

4. Set the config to point to an unlinked source.
   The following is an example to attach to an Oracle data source:

   ```delphix database "dexample" attachSource *> set source.name=dexamplePrimary
delphix database "dexample" attachSource *> set source.config=example2
delphix database "dexample" attachSource *> set environmentUser=myuser
delphix database "dexample" attachSource *> set dbUser=orauser
delphix database "dexample" attachSource *> set dbCredentials.password=orauserpwd```
5. Commit the operation.

delix database "dexample" attachSource *> commit
CLI Cookbook: Disabling LogSync for a dSource

This topic provides a simple example of how nested state is represented and manipulated. The LogSync state is maintained in the sourcingPolicy property of dSources, itself an object with several different fields.

**Procedure**

1. Select the dSource to be changed and run the `update` command.

   ```
   delphix> database "example"
   delphix "example"> update
   ```

2. Get the current property using dot-delimited notation.

   ```
   delphix "example" update *> get sourcingPolicy.logsyncEnabled
   true
   ```

3. The property could also be set using dot-delimited notation, but for illustrative purposes we can also use the `edit` command and set it directly.

   ```
   delphix "example" update *> edit sourcingPolicy
   delphix "example" update sourcingPolicy *> set logsyncEnabled=false
   ```

4. Commit the state, either from within the editing context or after running `back` to return to the parent context.

   ```
   delphix "example" update sourcingPolicy *> commit
   delphix "example">
   ```
CLI Cookbook: Enabling Oracle Validated Sync

This topic describes how to designate a staging host and enable validated sync for Oracle data sources.

Prerequisite - Designating a Staging Host

In order to validate an Oracle dSource snapshot for syncing, the Delphix Engine requires a host with an Oracle installation that is compatible with the dSource. This machine is known as the staging host. You must explicitly designate which machines you want the Delphix Engine to use as staging hosts. All machines that have been marked as staging sites are added to a pool. During validated sync, the Delphix Engine will select a compatible host from the pool, export the requisite archived redo logs and datafiles, and execute Oracle media recovery on the host. Follow these steps to designate a staging host.

1. Select the repository you want to designate as staging.

   ```shell
   delphix>/repository/select '/u01/app/ora10205/product/10.2.0/db_1'
   ```

2. Execute the `update` command.

   ```shell
   delphix repository "'/u01/app/ora10205/product/10.2.0/db_1'">update
   ```

3. Set staging to `true`.

   ```shell
   delphix repository "'/u01/app/ora10205/product/10.2.0/db_1'" update *>set staging=true
   ```

4. Commit the operation to designate the repository as staging.

   ```shell
   delphix repository "'/u01/app/ora10205/product/10.2.0/db_1'" update *> commit
   ```

To configure validated sync for multiple dSources with different Oracle versions, you must designate a compatible staging source for each. If multiple compatible staging sites exist, the Delphix Engine will select one at random.

- The validated sync process will consume some resources on the staging host when snapshots are taken. Designating a performance-critical host as a staging host is not recommended.

Procedure - Enabling Validated Sync

1. Select the dSource for which you want to enable validated sync.

   ```shell
   delphix>/database/select redsox1
   ```

2. Execute the `update` command.

   ```shell
   delphix database "redsox1">update
   ```

3. Set `preProvisioningEnabled` to `true`.

   ```shell
   delphix database "redsox1">update
   ```
3. `delphix database "redsox1" update *\>set preProvisioningEnabled=true`

4. **Commit** the operation to enable validated sync.

    `delphix database "redsox1" update *\>commit`
CLI Cookbook: Linking a Microsoft SQL Server Database Loading from a Specific Full Backup of the Source Database

This topic describes how to use the command line interface to link a SQL Server database by loading from a specific full backup of the source database as indicated by the backup UUID.

Prerequisites

- You can find the `fullBackupUUID` referenced in the last command line in the `msdb.dbo.backupset` on the source database, for example using the following query

```sql
USE master
SELECT backupset.database_name, 
    backupset.type, 
    backupset.backup_set_id, 
    backupset.backup_set_uuid, 
    backupset.family_guid, 
    backupset.position, 
    backupset.first_lsn, 
    backupset.last_lsn, 
    backupset.database_backup_lsn, 
    backupset.name, 
    backupset.has_bulk_logged_data, 
    backupset.is_damaged, 
    backupset.begins_log_chain, 
    backupset.is_copy_only, 
    backupset.backup_finish_date, 
    backupset.database_version, 
    backupset.database_guid, 
    mediafamily.logical_device_name, 
    mediafamily.physical_device_name
FROM msdb.dbo.backupmediafamily mediafamily 
JOIN msdb.dbo.backupset backupset 
ON mediafamily.media_set_id = backupset.media_set_id 
WHERE backupset.database_name = N'<Database Name>'
ORDER BY backupset.backup_finish_date desc
```

Procedure

Enter these commands through the Delphix Engine command line interface:
link;
set type=MSSqlLinkParameters;

set container.type=MSSqlDatabaseContainer;
set container.name=<dSource name>;
set container.group=<group name>;
set container.sourcingPolicy.loadFromBackup=true;

set source.type=MSSqlLinkedSource;
set source.config=<source database>;
set source.sharedBackupLocation="<source database backup location>";

set pptRepository=<SQL instance on the staging server>;

set container.sourcingPolicy.type=SourcingPolicy;

set dbUser=<source database login with SQL authentication>;

set dbCredentials.type=PasswordCredential;
set dbCredentials.password=<password for the database login>;

set fullBackupUUID=859FD1F1-1590-4FCB-A341-5D2D13852E2E;

commit;
## CLI Cookbook: Linking a Microsoft SQL Server Database Loading from the Last Full Backup of the Source Database

This topic describes how to use the command line interface to link a SQL Server database by loading from the last full backup of the source database.

### Procedure

Enter the following commands in the Delphix Engine command line interface:

```plaintext
/database;
link;
set type=MSSqlLinkParameters;

set container.type=MSSqlDatabaseContainer;
set container.name=<dSource name>;
set container.group=<group name>;
set container.sourcingPolicy.loadFromBackup=true;

set source.type=MSSqlLinkedSource;
set source.config=<source database>;
set source.sharedBackupLocation="<source database backup location>";

set pptRepository=<SQL instance on the staging server>;
set container.sourcingPolicy.type=SourcingPolicy;

set dbUser=<source database login with SQL authentication>;

set dbCredentials.type=PasswordCredential;
set dbCredentials.password=<password for the database login>;

commit;
```
CLI Cookbook: Linking to a Single Instance Oracle Database

This topic describes how to link to a single instance Oracle database using the Delphix Engine command line interface.

Prerequisites

You will need the following information:

- The name of the dSource you want to create.
- The group in which you want to create the dSource.
- The database unique name of the Oracle database you want to link to.
- The database username/password with sufficient privileges as described in the Delphix User Guide.
- The host environment user with sufficient privileges as described in the Delphix User Guide.

Procedure

1. Execute the `database link` command.

   ```
   delphix> database link
   delphix database link>
   ```

2. The default link operation is `OracleLinkParameters`, but you can confirm that by getting the input type:

   ```
   delphix database link *> get type
   OracleLinkParameters
   ```

3. Set the name for the dSource and the group in which you want to create it.

   ```
   delphix database link *> set container.name=dexample
   delphix database link *> set container.group="<New Group>"
   ```

4. Set the source configuration.
   
   For Oracle databases, these are identified by the database unique name. If you are unsure of the set of available databases, you can list available source configurations.

   ```
   delphix database link *> /sourceconfig list
   NAME      REPOSITORY            LINKINGENABLED
   example1  '/opt/ora/dexample1'  true
   example2  '/opt/ora/dexample1'  true
   delphix database link *> set source.config=example1
   ```

5. Set the privileged database username/password.
   
   The password can be set like other properties, or the value can be omitted so that it can be manually input without exposing the password.

   ```
   delphix database link *> set dbUser=delphix
   delphix database link *> set dbCredentials.password
   Enter dbCredentials.password: ********
6. Set the privileged environment user. This user must be from the same environment as the associated source config set in step 4. You can list the set of available users through the `environment user list` command.

```bash
delphix database link *> /environment/user list
NAME 172.168.1.2/oracle
delphix database link *> set environmentUser=172.168.1.2/oracle
```

7. Adjust any other properties you may want, such as RMAN tunables, description, and whether to link now. The full set of options is described in the API documentation for the `OracleLinkParameters` type. If you set the `linkNow` property, then this operation will wait for the sync to complete, otherwise you can perform the initial link by running the sync command at a later point.

```bash
delphix database link *> set masked=true
```

8. Commit the result.

```bash
delphix database link *> commit
    ORACLE_DB_CONTAINER-1
delphix>
```
CLI Cookbook: Listing Data Source Sizes

This topic describes a basic use of the CLI `list` command.

1. Switch to the source view and view the default list.

   delphix> source
   delphix source> list
   NAME      CONTAINER  VIRTUAL  CONFIG
   example   example    false    example
   vexample  vexample   true     vexample

2. List sources with their database size (in MB).

   delphix> source
   delphix source> list display=name,virtual,run.time.databaseSize
   NAME      VIRTUAL  RUNTIME.DATABASESIZE
   example   false    12784
   vexample  true     12842
CLI Cookbook: Detaching and Attaching a SQL Server dSource

**Prerequisites**

A dSource can only be attached to a new data source once it has been **unlinked**.

When attaching a SQL Server dSource to a new data source, the new data source must be the same database satisfying the following containts:

- Same database name
- Same recovery fork UUID
- pptRepository needs to be set to the name of the SQL instance on the staging server. The unlink operation removes the database from the SQL instance on the staging server and unmounts the iscsi luns, reattaching the dSource via the CLI will remount the iscsi luns and puts the database back.

**Procedure**

1. Select the dSource.

   ```
   delphix> database "dexample"
   ```

2. Run the `detachSource` command, specifying the currently active source. This step can be skipped if the dSource has already been detached through the GUI.

   ```
   delphix database "dexample"> detachSource
   delphix database "dexample" detachSource *> set source=dexample
   delphix database "dexample" detachSource *> commit
   ```

3. Run the `attachSource` command.

   ```
   delphix database "dexample"> attachSource
   ```

4. Set the following for SQL Server:

   You can also type `help pptRepository` to see what is wanted
   You can also set pptRepository=<then press tab> to list all values.

   ```
   delphix database "dexample" attachSource *> set source.config=SQLSERVER/dexample
   delphix database "dexample" attachSource *> set source.sharedBackupLocation=\SERVER1\Backups
   delphix database "dexample" attachSource *> set pptRepository=SQL2008R2
   delphix database "dexample" attachSource *> set dbUser=duser
   delphix database "dexample" attachSource *> set dbCredentials.password=dbuserpwd
   ```

5. Commit the operation.

   ```
   delphix database "dexample" attachSource *> commit
   ```
CLI Cookbook: How to Change Database User Password

1. ssh into your engine using Admin privileges

   ```
   ssh delphix_admin@delphixengine
   ```

2. Go to sourceconfig and find the Database that you need to update the password on

   ```
   delphix > sourceconfig
   delphix sourceconfig > ls
   delphix sourceconfig > select <yourdatabase>
   ```

3. Update the password

   ```
   delphix sourceconfig "yourdatabase" > update
   delphix sourceconfig "yourdatabase" update *> ls
   delphix sourceconfig "yourdatabase" update *> credentials.password=<new password>
   ```

4. Commit the change

   ```
   delphix sourceconfig "database" update *> commit
   ```

Example:

```
ssh delphix_admin@example
delphix > sourceconfig
delphix sourceconfig > ls

Objects

<table>
<thead>
<tr>
<th>NAME</th>
<th>REPOSITORY</th>
<th>LINKINGENABLED</th>
</tr>
</thead>
<tbody>
<tr>
<td>metal</td>
<td>'/u01/oracle/10.2.0.4/eel'</td>
<td>true</td>
</tr>
</tbody>
</table>

Operations

create

delphix sourceconfig > select metal
delphix sourceconfig "metal" > ls

Properties

  type: OracleSIConfig

  name: metal

  credentials:

    type: PasswordCredential

    password: ********
```
databaseName: metal

discovered: true

environmentUser: delphix

instance:

    type: OracleInstance

    instanceName: metal

    instanceNumber: 1

linkingEnabled: true

nonSysCredentials: (unset)

nonSysUser: (unset)

reference: ORACLE_SINGLE_CONFIG-1

repository: '/u01/oracle/10.2.0.4/ee1'

services:

    0:

        type: OracleService

        discovered: true

        jdbcConnectionString: jdbc:oracle:thin:@172.16.100.69:1525:metal

    1:

        type: OracleService

        discovered: true

        jdbcConnectionString: jdbc:oracle:thin:@172.16.100.69:1521:metal

    uniqueName: metal

    user: delphix


Operations

delete

update

validateCredentials
delphix sourceconfig "metal" > update
delphix sourceconfig "metal" update * > credentials.password=<new password>
delphix sourceconfig "metal" update * > commit
## CLI Cookbook: VDBs

These topics describe command line interface procedures for working with virtual databases (VDBs).

- CLI Cookbook: Changing the SID of Oracle RAC VDBs
- CLI Cookbook: Oracle VDB Migration
- CLI Cookbook: Provisioning a Single Instance Oracle VDB
- CLI Cookbook: Provisioning a SQL Server VDB
- CLI Cookbook: Provisioning a VDB from a TimeFlow Bookmark
- CLI Cookbook: Rolling Forward a VDB
- CLI Cookbook: Refresh a VDB from a Specific Timepoint or Latest
- CLI Cookbook: Repairing a Timeflow
- CLI Cookbook: Changing SGA Parameter
- CLI Cookbook: Rolling Back a VDB
- CLI Cookbook: Creating a VDB Config Template
- CLI Cookbook: Creating a Policy
- CLI Cookbook: Provisioning a SAP ASE VDB
- CLI Cookbook: Taking a Snapshot
- CLI Cookbook: Determining the Snapshot used to provision a VDB
CLI Cookbook: Changing the SID of Oracle RAC VDBs

This topic describes how to change the SID of instances in an Oracle RAC VDB.

This example demonstrates how to switch the instance name and number between two different hosts, from

```
SQL> select * FROM V$ACTIVE_INSTANCES;
INST_NUMBER INST_NAME
--------- -------------------------------
 1 cnrac3:VchiBEB1
 2 cnrac4:VchiBEB2
```
to

```
SQL> select * FROM V$ACTIVE_INSTANCES;
INST_NUMBER INST_NAME
--------- -------------------------------
 1 cnrac4:VchiBEB1
 2 cnrac3:VchiBEB2
```

Procedure

1. Stop the VDB through the GUI and login to the Delphix CLI

2. Select the sourceconfig of the RAC VDB whose instances you would like to rename.

   ```
   kfc-manual.dcenter> sourceconfig
   kfc-manual.dcenter sourceconfig> select Vchicago_BEB
   ```

3. Use the update command to change the properties of the sourceconfig

   ```
   kfc-manual.dcenter sourceconfig "Vchicago_BEB"> update
   ```

4. Type 'ls' to view the complete list of properties associated with the VDB's sourceconfig
5. Use the Set command to change the values for instanceName and instanceNumber for each instance.

```
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update => set
  instances.0.instanceName=VchiBEB2
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update => set
  instances.0.instanceNumber=2
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update => set
  instances.1.instanceName=VchiBEB1
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update => set
  instances.1.instanceNumber=1
```

6. Finally, commit the changes.

```
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update => commit;
```

7. Restart the VDB through the GUI for the changes to take effect on the VDB.
CLI Cookbook: Oracle VDB Migration

This topic describes moving a VDB from one environment or installation to another.

VDBs can be moved (or migrated) between hosts by changing the source repository associated with the VDB source config. This operation is currently only available via the CLI.

Restrictions

The following restrictions apply when migrating a VDB between repositories:

- When migrating a RAC VDB, the host of each OracleRACInstance must be updated as well.
- The mount point of the VDB source cannot be changed.
- The database_unique_name and db_name cannot be changed.
- The new environment and repository must be a compatible target environment.

Procedure

1. Select the source associated with the VDB. By default, sources are named the same as the VDB.

   ```
   delphix> source "vexample"
   ```

2. Disable the source by running the disable command and committing the operation.

   ```
   delphix source "vexample"> disable
   delphix source "vexample" disable *> commit
   Dispatched job JOB-171
   SOURCE_DISABLE job started for "vexample".
   Starting disable of virtual database.
   Unexporting storage.
   Virtual database disable successful.
   SOURCE_DISABLE job for "vexample" completed successfully.
   delphix source "vexample">
   ```

3. Select the source config associated with the source. By default this is also the same name as the VDB.

   ```
   delphix source "vexample"> get config
   vexample
   delphix source "vexample"> /sourceconfig "vexample"
   delphix sourceconfig "vexample">
   ```

4. Update the repository and repository user associated with the source config.

   ```
   delphix sourceconfig "vexample"> update
   delphix sourceconfig "vexample" update *> set
   repository=192.168.100.247/"/opt/oracle/product/10.2.0.4/db_1"
   delphix sourceconfig "vexample" update *> set
   environmentUser=192.168.100.247/ora1024
   delphix sourceconfig "vexample" update *> commit
   delphix sourceconfig "vexample">
   ```

5. Enable the source.

   ```
   delphix sourceconfig "vexample"> enable
   ```
delphix sourceconfig "vexample"> /source "vexample" enable
delphix source "vexample" enable *> commit
  Dispatched job JOB-173
  SOURCE_ENABLE job started for "vexample".
  Starting enable of virtual database.
  Exporting storage.
  Virtual database enable successful.
  SOURCE_ENABLE job for "vexample" completed successfully.
delphix sourceconfig "vexample">
CLI Cookbook: Provisioning a Single Instance Oracle VDB

This topic describes how to provision a single instance Oracle VDB using the Delphix Engine command line interface.

Prerequisites

You will need the following information:

- The name of the VDB you want to create
- The group in which to create the VDB
- The Oracle database name
- The Oracle database unique name
- The Oracle database instance number
- The Oracle database instance name
- The source dSource or VDB from which you wish to provision
- The semanticLocation, SCN, or timestamp of the point you want to provision from. You can run these commands to get the list of snapshots or timeflow ranges:

```bash
snapshot list database=dexample
timeflow "dexample" timeflowRanges; commit
```

- The base mountpoint on the target server where VDB data should be mounted
- The source repository (oracle install) in which to create the VDB. These can be listed with the `/repositor` y list command.

- If you are using a VDB template, the name of the template to use. See the Customizing Oracle VDB Configuration Settings topic for information on how templates affect configuration parameters on refresh.

Procedure

1. Execute the database provision command.

   ```bash
delphix> database provision
   ```

2. Execute the defaults command.

   ```bash
delphix database provision> defaults
   ```

3. Set the timeflow point source timeflow and location.

   ```bash
delphix database provision defaults *> set type=TimeflowPointSemantic
delphix database provision defaults *> set container=dexample
delphix database provision defaults *> set location=LATEST_SNAPSHOT
   ```

4. Commit the operation to populate the defaults, as provided by the browser interface. At this point, the operation can be committed, though you will likely need to change the defaults to match the information.

   ```bash
delphix database provision defaults *> commit
   ```
5. Set the name and group for the new VDB.

   delphix database provision *> set container.name=vexample
   delphix database provision *> set container.group="<New Group>"

6. Set the base mountpoint.

   delphix database provision *> set source.mountBase=/mnt

7. Set the source config type to be single instance Oracle, and set the database name and database unique name. When provisioning from a RAC or single instance oracle source, the default type will match that of the repository selected by the defaults operation.

   delphix database provision *> set sourceConfig.type=OracleSIConfig
   delphix database provision *> set sourceConfig.databaseName=vexample
   delphix database provision *> set sourceConfig.uniqueName=vexample123

8. Set the instance name and number.

   delphix database provision *> edit sourceConfig.instance
   delphix database provision sourceConfig.instance *> set instanceNumber=1
   delphix database provision sourceConfig.instance *> set instanceName=vexample
   delphix database provision sourceConfig.instance *> back

9. Set the target repository.

   delphix database provision *> set sourceConfig.repository=env'/opt/oracle'

10. Configure the Oracle database parameters. If you are using manually specified parameters, you can set the contents of source.configParams. If you want to use a template, you can set source.configTemplate.

    delphix database provision *> set source.configTemplate=exampleTemplate

11. Commit the result.

    delphix database provision *> commit
CLI Cookbook: Provisioning a SQL Server VDB

This topic describes how to provision a SQL Server VDB using the command line interface.

**Prerequisites**

You will need the following information:

- The name of the VDB you want to create
- The group in which to create the VDB
- The SQL Server database name for the VDB
- The source dSource or VDB from which you wish to provision
- The semanticLocation, LSN, or timestamp of the point you want to provision from. You can run these commands to get the list of snapshots or timeflow ranges:

```
snapshot list database=dexample
```

```
timeflow "dexample" timeflowRanges; commit
```

- The target host on which you want to create the VDB. You can list the hosts with the `/host list` command.
- The source repository (SQL Server instance on the target host) in which to create the VDB. These can be listed with the `/repository list` command.

**Procedure**

1. Execute the `database provision` command.

   ```
delphix> database provision
   ```

2. Execute the `defaults` command.

   ```
delphix database provision> defaults
   ```

3. Set the timeflow point source timeflow and location.

   ```
delphix database provision defaults *> set type=TimeflowPointSemantic
delphix database provision defaults *> set container=dexample
delphix database provision defaults *> set location=LATEST_SNAPSHOT
   ```

4. Commit the operation to populate the defaults, as provided by the browser interface. At this point, the operation can be committed, though you will likely need to change the defaults to match the information.

   ```
delphix database provision defaults *> commit
   ```

5. Set the name and group for the new VDB.

   ```
delphix database provision *> set container.name=vexample
delphix database provision *> set container.group="<New Group>"
   ```
6. Set the database name for the VDB on the target SQL Server instance.

```
delphix database provision *> set sourceConfig.databaseName=vexample
```

7. Set the target host.

```
delphix database provision *> edit sourceConfig.instance
delphix database provision sourceConfig.instance *> set host=targethost
delphix database provision sourceConfig.instance *> back
```

8. Set the target repository.

```
delphix database provision *> set sourceConfig.repository=targetEnv\SQLServer2008
```

9. Commit the result.

```
delphix database provision *> commit
```
CLI Cookbook: Provisioning a VDB from a TimeFlow Bookmark

This topic describes how to create a TimeFlow bookmark and use it to provision a single instance Oracle VDB using the Delphix Engine command line interface.

You can create TimeFlow bookmarks to give a semantically meaningful name to a TimeFlow point (scn, location or timestamp within a TimeFlow). You can then use the bookmarks you created to execute the following database operations:

- Provision
- Refresh
- Export
- Test file mappings
- VDB Rewind

Prerequisites

You will need the following information:

- The name of the timeflow bookmark you want to create
- The name of the VDB you want to create
- The group in which to create the VDB
- The Oracle database name
- The Oracle database unique name
- The Oracle database instance number
- The Oracle database instance name
- The source dSource or VDB from which you wish to provision
- The SCN, or timestamp of the point you want to provision from. You can run these commands to get the list of snapshots or timeflow ranges:

```
snapshot list database=dexample
timeflow "dexample" timeflowRanges; commit
```

- The base mountpoint on the target server where VDB data should be mounted
- The source repository (oracle install) in which to create the VDB. These can be listed with the `/repository list` command.

Creating the TimeFlow Bookmark

1. Execute the `timeflow bookmark create` command.

```
delphix> timeflow bookmark create
```

2. Set the timeflow point to be Oracle timeflow point.

```
delphix> timeflow bookmark create *> set timeflowPoint.type=OracleTimeflowPoint
```

3. Set the timeflow point timeflow and location.
Provisioning from a TimeFlow Bookmark

1. Execute the `database provision` command.

   delphix> database provision

2. Set the `timeflowPointParameters.type` to be `TimeflowPointBookmark`.

   delphix database provision *> set timeflowPointParameters.type=TimeflowPointBookmark

3. Set the `timeflow bookmark`.

   database provision *> set timeflowPointParameters.bookmark=myTimeFlowBookmark

4. Set the name and group for the new VDB.

   delphix database provision *> set container.name=vexample
delphix database provision *> set container.group="Untitled"

5. Set the base mountpoint.

   delphix database provision *> set source.mountBase=/mnt
6. Set the source config type to be single instance Oracle, and set the database name and database unique name.

   delphix database provision *> set sourceConfig.type=OracleSIConfig
   delphix database provision *> set sourceConfig.databaseName=vexample
   delphix database provision *> set sourceConfig.uniqueName=vexample123

7. Set the instance name and number.

   delphix database provision *> edit sourceConfig.instance
   delphix database provision sourceConfig.instance *> set instanceNumber=1
   delphix database provision sourceConfig.instance *> set instanceName=vexample
   delphix database provision sourceConfig.instance *> back

8. Set the target repository.

   delphix database provision *> set sourceConfig.repository=env/"/opt/oracle"

9. Commit the result.

   delphix database provision *> commit
CLI Cookbook: Rolling Forward a VDB

This topic describes how to roll forward a virtual database after it has been rewound, as described in Rewinding a VDB.

Once a VDB has rewound to a specific TimeFlow point, the snapshots of its previous states are still available in Delphix Engine storage, and be accessed via the command line interface to restore those previous states. This is referred to as "rolling forward" a VDB.

Procedure

1. Use the ls command to find the VDB you want to roll forward.
   In this example the TimeFlows and their associated containers are listed. The VDB called PVDB will be the one to roll forward.

```
delphix timeflow> ls
Objects
NAME                                CONTAINER  PARENTPOINT.TIMEFLOW
PARENTPOINT.LOCATION  PARENTPOINT.TIMESTAMP
hrprod/default                      hrprod     -                     -
erpprod/default                     erpprod    -                     -
'DB_PROVISION@2013-11-25T17:37:06'  PVDB       erpprod/default
657925                -                     -
'DB_ROLLBACK@2013-11-25T18:24:16'   PVDB       'DB_PROVISION@2013-11-25T17:37:06'
678552
```

2. Use the Select command to select the database.

```
delphix database> select PVDB
```

3. Use the rollback command to roll forward the VDB.

```
delphix database "PVDB"> rollback
```

4. Use the ls command to display options for selecting TimeFlow parameters.

```
delphix database "PVDB" rollback *> ls
Properties
  type: OracleRollbackParameters
  credential: (unset)
  timeflowPointParameters:
    type: TimeflowPointSemantic
    container: (required)
    location: LATEST_POINT
    username: (unset)
```

5. Because this VDB was rolled back, two TimeFlows now exist for it. To rollback the VDB and roll it forward, select the original TimeFlow, because the original snapshots are associated with that TimeFlow.
6. Use the `ls` command to view the parameter options for the TimeFlow you selected.

```bash
delphix database "PVDB" rollback *> ls
Properties
    type: OracleRollbackParameters
    credential: (unset)
    timeflowPointParameters:
        type: TimeflowPointLocation (*)
        location: LATEST_POINT
        timeflow: 'DB_PROVISION@2013-11-25T17:37:06' (*)
        username: (unset)
```

7. Set the TimeFlow location to rollback the VDB to a particular Oracle SCN.

```bash
delphix database "PVDB" rollback *> set timeflowPointParameters.location=678994
```

8. Use the `ls` command to review all the options you selected before executing the commit.

```bash
delphix database "PVDB" rollback *> ls
Properties
    type: OracleRollbackParameters
    credential: (unset)
    timeflowPointParameters:
        type: TimeflowPointLocation (*)
        location: 678994 (*)
        timeflow: 'DB_PROVISION@2013-11-25T17:37:06' (*)
        username: (unset)
```

9. Commit the changes.
delphix database "PVDB" rollback => commit
Dispatched job JOB-369
DB_ROLLBACK job started for "ERP/PVDB".
Starting provision of the virtual database "PVDB".
Creating new TimeFlow.
Generating recovery scripts.
Exporting storage.
Validating user environment settings on target host.
Mounting filesystems for the virtual database instance "1".
Mounting read-only archive log filesystem for the virtual database instance "1".
Running user-specified pre-provisioning script.
Recovering Oracle database.
Opening the virtual database "PVDB".
Opening Oracle database.
Oracle recovery was successful.
Unmounting read-only archive log filesystem for the virtual database instance "1".
Running user-specified post-provisioning script.
The virtual database "PVDB" was successfully provisioned.
Starting snapshot of virtual database.
Processing database files of virtual database.
Creating snapshot of virtual database.
Finalizing snapshot of virtual database.
Virtual database "PVDB" snapshot successful.
DB_ROLLBACK job for "ERP/PVDB" completed successfully.

Related Links

- [Rewinding a VDB](#)
CLI Cookbook: Refresh a VDB from a Specific Timepoint or Latest

This topic describes the steps to Refresh a VDB from a specific Timepoint or from Latest.

You can refresh from any point on Timeflow using SCN, location, or timestamp.

Prerequisites

You will need the following information:

- The VDB name
- The Timeflow location, SCN, or timestamp of the point you want to provision from.

Login to CLI:

$ ssh delphix_admin@<delphixengine>

Determine the TimeFlow

Run:

> timeflow "<dSource>" timeflowRanges
> commit
> cd

Perform the Refresh from specific Timepoint

> database
> select <VDB name>
> refresh
> set timeflowPointParameters.type= (one of TimeflowPointBookmark, TimeflowPointBookmarkTag, TimeflowPointLocation, TimeflowPointSemantic, TimeflowPointTimestamp as appropriate)
> set timeflowPointParameters.location= (the location, timestamp, or bookmark you wish to refresh to)
> set timeflowPointParameters.timeflow= (the timeflow associated with location)
> commit

Perform the Refresh from Latest

> database
> select <yourdatabase>
> refresh
> set timeflowPointParameters.container= (Parent of VDB)
> commit
CLI Cookbook: Repairing a Timeflow

Prerequisites

- Know the dSource and Group you need to repair from

Procedure

1. Log into the Delphix Engine as an Admin user. Go to timeflow and then list. Find the timeflow that needs to be repaired.

   ssh delphix_admin@<yourengine>
   delphix > timeflow
   delphix timeflow > ls

2. Go to oracle/log and list the missing logs for the timeflow. Note the start and end scn of the missing log.

   delphix timeflow oracle log> list timeflow=example missing=true

3. Run the fetch command and set all of the following fields

   delphix timeflow oracle log > fetch
   delphix timeflow oracle log fetch *> set type=TimeflowLogFetchParameters
   delphix timeflow oracle log fetch *> set timeflow=example
   delphix timeflow oracle log fetch *> set directory=[directory where you restored the log file]
   delphix timeflow oracle log fetch *> set endLocation=[end SCN of the sequence]
   delphix timeflow oracle log fetch *> set startLocation=[start SCN of the sequence]
   delphix timeflow oracle log fetch *> set host=[hostname or IP of the host you restored the file to]
   delphix timeflow oracle log fetch *> set username=[a user that can read the file]
   delphix timeflow oracle log fetch *> edit credentials
   delphix timeflow oracle log fetch *> set type=PasswordCredential
   delphix timeflow oracle log fetch *> set password=[password for this user]

4. Commit the Changes

   delphix timeflow oracle log fetch *> commit

   Note: Only do ONE repair job at a time

NOTE -

It is possible there may be more than one timeflow visible for a given container/source, if so you can verify the current timeflow being used with:
look for the "currentTimeflow" value.
CLI Cookbook: Changing SGA Parameter

Below outlines the procedure to change SGA parameter setting on a provisioned VDB.

Procedure

1. Log into the Delphix Engine as `delphix_admin` or a user with Admin privileges.
2. Go to `source` and then `select` the name of the VDB that you would like to change the parameters of
3. You are then going to `update` and edit the `configParams`
4. Next you are going to set the `sga_target` to the correct value
5. Commit the operation so that it saves

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
</table>
| ssh delphix_admin@engine  
delphix > source  
delphix source > select "vdb_example"  
delphix source "vdb_example" > update  
delphix source "vdb_example" *> set sga_target=new value  
delphix source "vdb_example" *> commit |
CLI Cookbook: Rolling Back a VDB

Requirements
1. Know the bookmark you want to rollback to

Rolling Back or Rewinding to a Timeflow Bookmark

1. Log into the Delphix Engine

   ssh delphix@<yourengine>

2. Retrieve database and time flow information that you would like to rewind/rollback to

   delphix > ls
   delphix database > select "dexample"
   delphix database "dexample" > get currentTimeflow

3. Rollback/Rewind VDB

   delphix database "dexample" > rollback
   delphix database "dexample" rollback *> ls
   delphix database "dexample" rollback *> set
timeflowPointParameters.type=TimeflowPointBookmark
   delphix database "dexample" rollback *> set timeflowPointParameters.bookmark="bookmark example"
   delphix database "dexample" rollback *> commit

Related Support or Delphix Documentation Topics

Creating a Bookmark in the CLI

Rewinding a VDB Using the GUI
CLI Cookbook: Creating a VDB Config Template

This topic will address how to create a VDB Config Template in the CLI; this functionality is also available in the GUI

**Procedure**

1. ssh into your delphix engine using delphix_admin credentials

   ```
   ssh delphix_admin@<yourdelphixengine>
   ```

2. Go to `database and then template and then create`

   ```
   delphix > database template
   delphix database template > create
   delphix database template create *> set name=
   delphix database template create *> set parameters.<set parameters you want>
   delphix database template create *> set sourceType=
   ```

3. Verify information and commit

   ```
   delphix database template create *> ls
   delphix database template create *> commit
   ```

**Example:**

```
ssh delphix_admin@testengine
testengine > database template
testengine database template > create
testengine database template create *> set name=test_template
testengine database template create *> set parameters.none
testengine database template create *> set sourceType=OracleVirtualSource
testengine database template create *> ls
```

**Properties**

- **type**: DatabaseTemplate
- **name**: test_template (*)
- **description**: (unset)
- **parameters**:
  - none: none (*)
  - sourceType: OracleVirtualSource (*)

```
testengine database template create *> commit
```
CLI Cookbook: Creating a Policy

This will outline how to create a policy in the CLI, please note that you can also do this in the GUI

Procedure

1. ssh into your Delphix Engine using delphix_admin credentials

```bash
ssh delphix_admin@delphixengine
delphix > ls
```

2. Go to policies and createAndApply (please note that you cannot just create a policy, you must createAndApply, in the GUI you have the option to just create) and set your policy parameter

```bash
delphix > policy
delphix policy > createAndApply
delphix policy createAndApply *> set policy.type=< choose from QuotaPolicy, RefreshPolicy, RetentionPolicy, SnapshotPolicy or SyncPolicy>
delphix policy createAndApply *> set policy.name=< name your policy>
delphix policy createAndApply *> set policy.customized=true
delphix policy createAndApply *> set policy.provisionSource=(LATEST_SNAPSHOT or LATEST_TIME_FLOW_LOG)
```

*i* *if doing a RefreshPolicy, SyncPolicy or SnapshotPolicy you are also going to need to add the following:*

```bash
delphix policy createAndApply *> edit policy.scheduleList
delphix policy createAndApply policy.scheduleList *> add
delphix policy createAndApply policy.scheduleList *> set
cronString=
delphix policy createAndApply policy.scheduleList *> set
cutoffTime=
delphix policy createAndApply policy.scheduleList *> back
```

3. Set your target parameters which are going to be a container, group etc

```bash
delphix policy createAndApply *> set target=
```

4. Verify and commit

```bash
delphix policy createAndApply *> ls
delphix policy createAndApply *> commit
```
CLI Cookbook: Provisioning a SAP ASE VDB

This topic describes how to provision an SAP ASE VDB using the command line interface.

- **Prerequisites**
- **Procedure**

**Prerequisites**

You will need the following information:

- The name of the VDB you want to create
- The group in which to create the VDB
- The SAP ASE database name for the VDB
- The source dSource or VDB from which you wish to provision
- The semanticLocation, LSN, or timestamp of the point you want to provision from (if not using the most recent). You can run these commands to get the list of snapshots or timeflow ranges:

```
snapshot list database=dexample
timeflow "dexample" timeflowRanges; commit
```

- The target host on which you want to create the VDB. You can list the hosts with the /host list command.
- The source repository (SAP ASE instance on the target host) in which to create the VDB. These can be listed with the /repository list command.

**Procedure**

1. Execute the database provision command.

```
delphix> database provision
```

2. Set the type for the new VDB

```
delphix database provision *> set type=ASEProvisionParameters
```

3. Set the name and group for the new VDB.

```
delphix database provision *> set container.name=<vexample>
delphix database provision *> set container.group="<New Group>"
```

4. Set the name of the new VDB.

```
delphix database provision *> set sourceConfig.databaseName=<vexample>
```

5. Set the source configuration properties on the target SAP ASE instance
6. Set the target Dataset Home.
   ```
   delphix database provision *> set sourceConfig.repository=<Dataset Home>
   ```

7. Set the source container from which to provision.
   ```
   delphix database provision *> set timeflowPointParameters.container=<dexample>
   ```

8. Set the desired value for truncateLogOnCheckpoint
   ```
   delphix database provision *> set truncateLogOnCheckpoint=false
   ```

9. Commit the configuration and start the DB_PROVISION job
   ```
   delphix database provision *> commit
   ```
CLI Cookbook: Taking a Snapshot

This article is to document how to take a Snapshot outside of the normal snapshot policy time using the CLI, you can also do this in the GUI using the camera icon. A Snapshot of a VDB is similar to bookmarking a point in time in the life of the VDB.

Procedure:

1. ssh into the delphix engine using delphix_admin credentials
2. Go into databases and select the VDB or dSource you would like to take a Snapshot of

   ```
   ssh delphix_admin@dengine
   delphix > database
   delphix database > select vdb_test
   ```

3. You are now going to sync and commit the operation

   ```
   delphix database "vdb_test" > sync
   delphix database "vdb_test" sync *> commit
   ```

4. You can verify the snapshot by going to snapshots and listing them

   ```
   delphix database "vdb_test" > /snapshot
   delphix snapshot > ls
   ```

Related Articles:

CLI Cookbook: Creating Policies
CLI Cookbook: Determining the Snapshot used to provision a VDB

 Procedure:

 *In Delphix 3.0 and higher, the parent snapshot can be determined using the CLI as follows:

 1. Log into the server as a Delphix administrator:

     
     ```
     ssh delphix_admin@<server_ip>
     ```

 2. Select the VDB.

     ```
     delphix> database
     delphix database> ls
     Objects
     NAME          PARENTCONTAINER DESCRIPTION
     dSource1      -
     dSource2      -
     VDB1          dSource1        -
     VDB2          dSource2        -
     VDB3          dSource1        -
     delphix database> select VDB1
     ```

 3. List the VDB parameters, and make a note of the currentTimeflow value.

     ```
     delphix database "VDB1"> ls
     Properties
     type: OracleDatabaseContainer
     name: VDB1
     currentTimeflow: VDB1/default
     description: (unset)
     diagnoseNoLoggingFaults: true
     endianness: BIG_ENDIAN
     group: <New Group>
     masked: false
     os: HP-UX
     parentContainer: dSource1
     performanceMode: false
     processor: ia64
     reference: ORACLE_DB_CONTAINER-10
     runtime:
     type: OracleDBContainerRuntime
     logSyncActive: true
     sourcingPolicy:
     type: OracleSourcingPolicy
     encryptedLinkingEnabled: false
     logsyncEnabled: true
     logsyncInterval: 300
     logsyncMode: ARCHIVE_ONLY_MODE
     version:
     ```

 4. Select the Timeflow listed for the VDB.
5. List the Timeflow parameters. The Snapshot used to provision the VDB is listed as parentSnapshot.

delphix timeflow "VDB1/default"> ls
Properties

type: OracleTimeflow
   name: VDB1/default
   container: VDB1
   parentPoint:
       type: OracleTimeflowPoint
       location: 141285148
       timeflow: dSource1/default
       parentSnapshot: @2013-02-14T15:07:28.491Z
       reference: ORACLE_TIMEFLOW-92572
CLI Cookbook: Replication

These topics describe how to use the command line interface for replication tasks.

- [CLI Cookbook: Adding a Replication Spec](#)
- [CLI Cookbook: Deleting a Replication Spec](#)
- [CLI Cookbook: Failing Over a Namespace](#)
- [CLI Cookbook: Triggering Immediate Execution of a Replication Spec](#)
CLI Cookbook: Adding a Replication Spec

This topic describes how to use the command line interface to add a replication specification to the Delphix Engine.

Unlike the GUI, the CLI supports the ability to manage multiple replication specifications within a single system. This allows updates to be sent to multiple systems from a single point.

Prerequisites

- You should review the topic Replication Overview to understand which objects are copied as part of a backup or restore operation, as well as the dependencies between objects.

Procedure

1. Switch to the replication spec context.

   ```
   delphix> cd replication/spec
   delphix replication spec> ls
   Operations
   create
   ```

2. Create a new replication spec.

   ```
   delphix replication spec> create
   delphix replication spec create *> ls
   Properties
   type: ReplicationSpec
   name: (unset)
   bandwidthLimit: (unset)
   compressed: (unset)
   enabled: (unset)
   encrypted: (unset)
   objects: (required)
   schedule: (unset)
   targetCredential:
      type: PasswordCredential
      password: (required)
   targetHost: (required)
   targetPrincipal: (required)
   ```

3. Specify the target host name, user, and credentials.

   ```
   delphix replication spec create *> set targetHost=exampleHost.mycompany.com
   delphix replication spec create *> set targetPrincipal=delphix_admin
   delphix replication spec create *> set targetCredential.password=password
   ```

   **Target Principal**

   The target principal must be a Delphix user on the target host who has domain privileges.

4. Specify the set of objects to replicate.

   a. To replicate all dSources and VDBs on the system, specify `DOMAIN`as the list of objects.
4. a. To replicate a subset of Groups, VDBs and dSources, specify their names as a comma-separated list.

```
delphix replication spec create *> set objects=`DOMAIN
```

```
delphix replication spec create *> set objects=dExample1,dExample2
```

**Name Completion**

The CLI will provide possible completions for all objects in the system, but only groups, dSources and VDBs can be specified. Attempts to replicate other types of objects will generate an error when the operation is committed.

5. Commit the operation.

```
delphix replication spec create *> commit
`REPLICATION_SPEC-1
```
CLI Cookbook: Deleting a Replication Spec

This topic describes how to use the command line interface to delete a replication spec.

Procedure

1. Switch to the replication spec context and list the specs on the system.

   ```
   delphix> cd replication/spec
   delphix replication spec> ls
   Objects
   REFERENCE TARGETHOST
   REPLICATION_SPEC-1 exampleHost.mycompany.com
   Operations
   create
   ```

2. Select the replication spec to remove.

   ```
   delphix replication spec> select REPLICATION_SPEC-1
   delphix replication spec "exampleHost.mycompany.com">
   ```

3. Remove the spec.

   ```
   delphix replication spec "exampleHost.mycompany.com"> delete
   delphix replication spec "exampleHost.mycompany.com" delete *> commit
   ```
CLI Cookbook: Failing Over a Namespace

This topic describes how to use the command line interface to fail over a namespace.

Procedure

1. Switch to the namespace context and list the namespaces on the system

```
   delphix> cd namespace
   delphix namespace> ls
   Objects
   NAME
   [172.16.203.93]
   Operations
   lookup
```

2. Select the namespace to failover

```
   delphix namespace> select [172.16.203.93]
   delphix namespace "[172.16.203.93]"> 
```

3. Failover the namespace

```
   delphix namespace "[172.16.203.93]"> failover
   delphix namespace "[172.16.203.93]" failover *> commit
```

Failover

Failover will sever the replication connection and make objects in the namespace part of the live system. This will prevent the target from receiving subsequent incremental updates.
CLI Cookbook: Triggering Immediate Execution of a Replication Spec

This topic describes how to use the command line interface to trigger an immediate execution of a replication spec in the Delphix Engine.

**Procedure**

1. Switch to the replication spec context and list the specs on the system.
   
   ```
   delphix> cd replication/spec
   delphix replication spec> ls
   Objects
   REFERENCE            TARGETHOST
   REPLICATION_SPEC-1    exampleHost.mycompany.com
   
   Operations
   create
   ```

2. Select the replication spec to execute.
   
   ```
   delphix replication spec> select REPLICATION_SPEC-1
   delphix replication spec "exampleHost.mycompany.com">
   ```

3. Execute the spec.
   
   ```
   delphix replication spec "exampleHost.mycompany.com"> execute
   delphix replication spec "exampleHost.mycompany.com" execute *> commit
   Dispatched job JOB-7
   REPPLICATION_SEND job started.
   Connecting to target "exampleHost.mycompany.com".
   Preparing replication update.
   Starting incremental replication update.
   Sending metadata.
   Sending data for "Untitled".
   Sending data for "Untitled/redsox1".
   Transfer completed in 0:00:01, sent 1.39MB (1.39MB/s).
   Committing serialization state.
   REPPLICATION_SEND job completed successfully.
   ```
Web Service API Guide

These topics describe interfacing with the public web service APIs, building automation facilities and integrating with third party orchestration tools.

- API Version Information
- Web Service Object Model
- Web Service Protocol
- CLI to Web Services Transition
- GUI API Mapping
- API Cookbook: Common Tasks, Workflows, and Examples
  - API Cookbook: Authentication
  - API Cookbook: Host Environment Details
  - API Cookbook: List Alerts and List Jobs
  - API Cookbook: List dSources and VDBs
  - API Cookbook: List Snapshots
  - API Cookbook: Refresh VDB
  - API Cookbook: Example Provision Of An Oracle VDB
  - API Cookbook: Stop/Start a VDB
# API Version Information

This topic describes API version information for each release of the Delphix Engine, including schema changes and links to the relevant version of the schema.

<table>
<thead>
<tr>
<th>Delphix Engine Version</th>
<th>API Version</th>
<th>Link to Schema within the Appliance</th>
<th>Major Changes</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0.x.x</td>
<td>1.0.0</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.0.0/delphix.json</td>
<td>Complete architectural change from 2.7 CLI. See <a href="#">Migrating from the Delphix Engine 2.7 CLI</a> and related topics for more information.</td>
<td></td>
</tr>
<tr>
<td>3.1.0.x - 3.1.1.x</td>
<td>1.1.0</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.1.0/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1.2+</td>
<td>1.1.1</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.1.1/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2.0.x</td>
<td>1.2.0</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.2.0/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2.1.x</td>
<td>1.2.1</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.2.1/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2.2.x - 3.2.3.x</td>
<td>1.2.2</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.2.2/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2.4+</td>
<td>1.2.3</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.2.3/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0.0.x</td>
<td>1.3.0</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.3.0/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0.1.x - 4.0.2.x</td>
<td>1.3.1</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.3.1/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0.3+</td>
<td>1.3.2</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.3.2/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.0.x</td>
<td>1.4.0</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.4.0/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Version</td>
<td>URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.1.x</td>
<td>1.4.1</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.4.1/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.2.x - 4.1.3.x</td>
<td>1.4.2</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.4.2/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.4+</td>
<td>1.4.3</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.4.3/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2.1.0</td>
<td>1.5.0</td>
<td>http://&lt;engine-address&gt;/api/json/versions/1.5.0/delphix.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2.2.0</td>
<td>1.5.1</td>
<td>http://&lt;engine-address&gt;/api/json/delphix.json</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Related Links**
- [Migrating from the Delphix Engine 2.7 CLI](#)
Web Service Object Model

This topic describes the Delphix object model as exported over the web services.

Object Types

All objects in the Delphix API are “typed objects.” All such objects have a type field that indicates the type of the object and its associated semantics. This allows for object inheritance and polymorphism without requiring separate APIs for each type, and allows generic client-specific semantic encoding and decoding without having to be aware of the context. This means that even APIs that operate only a specific type (such as the Group API) still require a type field to be specified as part of the input, and will continue to report the type of objects when listing or retrieving objects. This allows the APIs to evolve in a backwards-compatible fashion through the introduction of new types.

Certain "root" object types (groups, containers, sources, etc) have an associated API. This API is rooted at a particular point under /resources/json/delphix, but all APIs follow a standard format beneath this namespace. The CLI namespace is a direct reflection of this URL, and the API reference has an index both by object type as well as by object (CLI) path. These APIs may operate on different extended types (such as OracleSIConfig and OracleRACConfig), but the base operations remain the same regardless of input type.

Object References

Some objects returned by the APIs are pure typed objects, in that they don’t represent persistent state on the Delphix Engine but are rather calculated and returned upon request. Most of the objects in the system, however, are “persistent objects.” Persistent objects (of type PersistentObject) have a stable reference that uniquely identifies the object on the system. This reference is separate from its name, so that objects can be renamed without affecting the programmatic API. More information about object names and how they can be represented generically can be found in the CLI documentation. Object references are opaque tokens; while they can be stored and re-used for later use, an interpretation of their contents is unstable and may break in a future release.

The Delphix object hierarchy is stitched together by these object references. When fetching an object that refers to another object, the member will be returned as a reference, rather than being inserted directly within the parent object. This allows consumers to control when and how links are resolved, and makes it clear when an object may change independently from its parent. The per-object APIs outlined below all operate on object references.

Note that some Delphix objects are singleton objects, and there is only one such object on the system. These objects do not have references because the API URL uniquely identifies the object on the system.

API Operations

All APIs optionally support the following operations:

- CREATE - Create a new instance of the given object type. This is used for most objects, but more complicated objects, such as dSources and VDBs, must be created through a dedicated link or provision operation. The input to this operation is typically the object itself, though some objects may have specialized parameters used during the creation of objects. An example of this is HostEnvironmentCreateParameters.
- UPDATE - Update properties of the given object, specified as an object reference in the URL.
- DELETE - Delete a particular object, specified as an object reference in the URL. These operations are typically done as HTTP DELETE operations, but it is also possible to do a POST operation to the /delete API to do the same thing. The POST form allows for delete-specific parameters, such as OracleDeleteParameters.
- GET - Get the properties for a particular object, specified as an object reference in the URL.
• **LIST** - List all objects of the given type within the system. These APIs typically take optional query parameters that allow the set of results to be constrained, filtered, paginated, or sorted.

In addition, the following non-CRUD operations may be supported:

- **Root Operation** - A POST or GET operation to the root of an API namespace, not associated with a particular object. This can be used for singleton objects, such as `NDMPConfig`, operations that create objects, such as `link`, and operations that operate on multiple objects at once.

- **Per-object Operation** - A POST operation to a particular object reference. These operations are typically read-write, but are not required to be so. These would include read-only operations that cannot be modeled as CRUD operations or require complex input use per-object operations.

**Database Object Model**

In order to support a wide variety of databases and database configurations, the database object model is more complex than it may initially appear after having used the Delphix GUI. For example, there is no such thing as a "dSource" or "VDB" object, only data "containers" with attached "sources". More information about how Database objects are modeled within Delphix can be found in the [CLI documentation](#).

**Asynchronous Execution**

All APIs are designed to be transactionally safe and "quick." However, there are operations that may take a long period of time, or may need to reach out to external hosts or databases such that they cannot be done safely within the context of a single API call. Such operations will dispatch a `Job` to handle asynchronous execution of the operation. Any API can potentially spawn a job, and which APIs spawn jobs and which do not may differ between object types or releases. If you are developing a full-featured automation system, it is recommended that you build a generic infrastructure to handle job monitoring, rather than encoding the behavior of particular APIs that may change over time.

Every operation, except for LIST and GET, which are guaranteed to be read-only, can potentially spawn a job. This is represented by the `job` field of the `APIResult` object. If this field is null, then the action can be completed within the bounds of the API call. Otherwise, a reference to a dispatched job is returned.

Jobs can spawn other jobs for especially complex operations, such as provisioning to an Oracle cluster environment. The job returned in the API invocation is the root job, and overall success or failure of the operation is determined by the state of this job. Some operations may succeed even if a child job fails. An example would be provisioning to an Oracle cluster where one node failed, but others were successful.

Progress can be monitored by examining the `JobEvent` objects in the `Job` object returned through the job API.
Web Service Protocol

This topic describes an overview of the web service API and available facilities.

Introduction

The Delphix Engine provides a set of public stable web service APIs (application programming interfaces). The web services form the basis upon with the GUI and CLI are built, and are designed to be public and stable. This guide covers the basic operation of the protocol, concepts, and considerations when building layered infrastructure. It is not a reference for all available APIs. For more information on available APIs, go to the '/api' URL of a Delphix appliance, which will provide a complete reference of all available APIs for the version of Delphix running on that system.

The CLI is a thin veneer over the web services. If you are new to the web services, it is recommended you first test out operations with the CLI, and use the `setopt trace=true` option to dump the raw data being sent and received to see the API in action.

Protocol Operation

The Delphix web services are a RESTful API with loose CRUD semantics using JSON encoding.

The following HTTP methods are supported:

- **GET** - Retrieve data from the server where complex input is not needed. All GET requests are guaranteed to be read-only, but not all read-only requests are required to use GET. Simple input (strings, number, boolean values) can be passed as query parameters.
- **POST** - Issue a read/write operation, or make a read-only call that requires complex input. The optional body of the call is expressed as JSON.
- **DELETE** - Delete an object on the system. For languages that don't provide a native wrapper for DELETE, or for delete operations with optional input, all delete operations can also be invoked as POST to the same URL with `/delete` appended to it.

Regardless of the operation, the result is returned as a JSON encoded result, the contents of which are covered below. For example, the following invocation of `curl` demonstrates establishing a new Session (pretty-printing the result):
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/session \
-H "Content-Type: application/json" <<EOF
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
        "major": 1,
        "minor": 0,
        "micro": 0
    }
}
EOF
{
    "status": "OK",
    "result": {
        "type": "APISession",
        "version": {
            "type": "APIVersion",
            "major": 1,
            "minor": 0,
            "micro": 0
        },
        "locale": "en_US",
        "client": null
    }
},
"job": null
}

Session Establishment

Login involves establishing a session and then authenticating to the Delphix Engine. Only authenticated users can access the web APIs. Each user must establish a session prior to making any other API calls. This is done by sending a Session object to the URL /resources/json/delphix/session. This session object will specify the APIVersion to use for communication between the client and server. If the server doesn't support the version requested due to an incompatible change reflected in the API major version number, an error will be returned.

The resulting session object encodes the native server version, which can be different than the version requested by the client. If the server is running a more recent but compatible version, any translation of input and output to the native version is handled automatically. More information on versioning can be found in the documentation for the APIVersion object within the API reference on a Delphix system. If the client supports multiple versions, the appropriate type can be negotiated by trying to establish a session with each major version supported, and then inspecting the version returned.

The session will also return an identifier through browser cookies that can be reused in subsequent calls to use the same session credentials and state without having to re-authenticate. The format of this cookie is private to the server and may change at any point. Sessions do not persist across a server restart or reboot. The mechanism by which this cookie is preserved and sent with subsequent requests is client-specific. The following demonstrates invoking the session login API call using curl and storing the cookies in the file ~/cookies.txt for later use:
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/session \
   -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
   "type": "APISession",
   "version": {
      "type": "APIVersion",
      "major": 1,
      "minor": 0,
      "micro": 0
   }
}
EOF

Authentication

Once the session has been established, the next step is to authenticate to the server by executing the LoginRequest API. Unauthenticated sessions are prohibited from making any API calls other than this login request. The username can be either a system user or domain user, and the backend will authenticate using the appropriate method. This example illustrates logging in via curl using cookies created when the session was established:

$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/login \
   -b ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
   "type": "LoginRequest",
   "username": "delphix_username",
   "password": "delphix_password"
}
EOF

The login API currently only supports authentication by password. There is no way to authenticate using any shared key or alternate authentication strategy.
CLI to Web Services Transition

This topic describes using the CLI to understand the public web service APIs.

The command line interface is a direct translation of the web services API to an interactive environment. This allows you to use the CLI to explore functionality with tab completion, integrated help, stronger type checking, and indication of expected types and required fields. When trying to determine how to invoke an operation through the web services or interpret the results, it is recommended that you first learn how to do the same through the CLI, and then use the provided tools to translate that into web services call.

CLI Translation to HTTP

The CLI namespace is identical to the the web service URLs for each base object and operation type. The root of all web services is /resources/json/delphix. Any additional CLI context is appended to this URL, joined by slashes. For example:

```
  delphix> database provision
```

Is equivalent to:

```
  POST /resources/json/delphix/database/provision
```

All operations in the CLI (those that require an explicit commit command) are modeled as POST HTTP calls. This is an example of a "root operation", as they are invoked not on any particular object, but across the system as a whole. Operations that are invoked on a particular object use a URL specific to that object:

```
  delphix> database "dexample" refresh
```

Is equivalent to:

```
  POST /resources/json/delphix/database/ORACLE_DB_CONTAINER-3/refresh
```

While the CLI uses names to refer to objects, at the API level we use references. Persistent objects (those with a permanent unique identity) have a reference field that is used in all cases to refer to the object. This allows references to remain constant even if objects are renamed.

For the standard CRUD (create, read, update, delete) operations, the mapping is slightly different:

<table>
<thead>
<tr>
<th>CLI Operation</th>
<th>HTTP API</th>
</tr>
</thead>
<tbody>
<tr>
<td>database list</td>
<td>GET /resources/json/delphix/database</td>
</tr>
<tr>
<td>database create</td>
<td>POST /resources/json/delphix/database</td>
</tr>
<tr>
<td>database &quot;dexample&quot; get</td>
<td>GET /resources/json/delphix/database/&lt;reference&gt;</td>
</tr>
<tr>
<td>database &quot;dexample&quot; update</td>
<td>POST /resources/json/delphix/database/&lt;reference&gt;</td>
</tr>
</tbody>
</table>
The **DELETE** operation has an optional **POST** form that can take complex input for clients that don't support sending a payload as part of a **DELETE** operation.

**Tracing HTTP Calls**

The CLI also provides facilities to see the raw HTTP calls being made as part of any operation. To start with, viewing data in JSON format (**setopt format=json**) will provide an example of what the raw output looks like from the server. In its raw form, the CLI does not make any attempt to interpret the results, so references are displayed as references (and not names), and sizes are displayed as their raw numeric value.

This is helpful for scripting, but the CLI also has a mode to display the requests being sent to the server, the responses received, and the URLs used. To enable this mode, run **setopt trace=true**. Once you have determined how to do something through the CLI, you can use this mode as the basis for building direct integration with the raw HTTP APIs.
When using trace mode within the context of a specific object, we refresh the contents of the object before executing each command. This results in the GET request before the delete command in the above example.
GUI API Mapping

This topic describes how to map from GUI operations to the corresponding CLI operation.

It is not always straightforward to convert from the visual layout of the GUI to the corresponding CLI operations. This topic outlines some common operations and indicates how they are represented in the CLI, web services, and the API documentation.

dSource Operations

<table>
<thead>
<tr>
<th>GUI</th>
<th>CLI</th>
<th>API Topic</th>
<th>Input Object</th>
<th>Web Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>database link</td>
<td>Container</td>
<td>LinkParameters</td>
<td>POST /resources/json/delphix/databasename/link</td>
</tr>
<tr>
<td>Show config.</td>
<td>database &quot;name&quot;</td>
<td>Container</td>
<td>Source</td>
<td>GET /resources/json/delphix/databasename/{ref}</td>
</tr>
<tr>
<td></td>
<td>get</td>
<td></td>
<td></td>
<td>GET /resources/json/delphix/source/{ref}</td>
</tr>
<tr>
<td>Sync</td>
<td>database &quot;name&quot;</td>
<td>Container</td>
<td>SyncParameters</td>
<td>POST /resources/json/delphix/databasename/{ref}/sync</td>
</tr>
<tr>
<td></td>
<td>sync</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update</td>
<td>database &quot;name&quot;</td>
<td>Container</td>
<td>Container</td>
<td>POST /resources/json/delphix/databasename/{ref}</td>
</tr>
<tr>
<td></td>
<td>update</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>database &quot;name&quot;</td>
<td>Container</td>
<td>DeleteParameters</td>
<td>POST /resources/json/delphix/databasename/{ref}/delete</td>
</tr>
<tr>
<td></td>
<td>delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detach</td>
<td>database &quot;name&quot;</td>
<td>Container</td>
<td>DetachSourcePar</td>
<td>POST /resources/json/delphix/databasename/{ref}/detach</td>
</tr>
<tr>
<td></td>
<td>detachSource</td>
<td></td>
<td>ameters</td>
<td></td>
</tr>
<tr>
<td>Attach</td>
<td>database &quot;name&quot;</td>
<td>Container</td>
<td>AttachSourcePar</td>
<td>POST /resources/json/delphix/databasename/{ref}/attach</td>
</tr>
<tr>
<td></td>
<td>attachSource</td>
<td></td>
<td>ameters</td>
<td></td>
</tr>
<tr>
<td>Disable</td>
<td>source &quot;name&quot; disable</td>
<td>Source</td>
<td>SourceDisableParameters</td>
<td>POST /resources/json /delphix/source/{ref}/disable</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------</td>
<td>--------</td>
<td>-------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>Enable</td>
<td>source &quot;name&quot; enable</td>
<td>Source</td>
<td>SourceEnableParameters</td>
<td>POST /resources/json /delphix/source/{ref}/enable</td>
</tr>
</tbody>
</table>

**VDB Operations**

<table>
<thead>
<tr>
<th>GUI</th>
<th>CLI</th>
<th>API Topic</th>
<th>Input Object</th>
<th>Web Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision</td>
<td>database provision</td>
<td>Container</td>
<td>ProvisionParameters</td>
<td>POST /resources/json /delphix/database/provision</td>
</tr>
<tr>
<td>V2P</td>
<td>database export</td>
<td>Container</td>
<td>ExportParameters</td>
<td>POST /resources/json /delphix/database/export</td>
</tr>
<tr>
<td>Refresh</td>
<td>database &quot;name&quot; refresh</td>
<td>Container</td>
<td>RefreshParameters</td>
<td>POST /resources/json /delphix/database/{ref}/refresh</td>
</tr>
<tr>
<td>Snapshot</td>
<td>database &quot;name&quot; sync</td>
<td>Container</td>
<td>SyncParameters</td>
<td>POST /resources/json /delphix/database/{ref}/sync</td>
</tr>
<tr>
<td>Update</td>
<td>database &quot;name&quot; update</td>
<td>Container</td>
<td>Container</td>
<td>POST /resources/json /delphix/database/{ref}</td>
</tr>
<tr>
<td>Delete</td>
<td>database &quot;name&quot; delete</td>
<td>Container</td>
<td>DeleteParameters</td>
<td>POST /resources/json /delphix/database/{ref}/delete</td>
</tr>
<tr>
<td>Start</td>
<td>source &quot;name&quot; start</td>
<td>Source</td>
<td>StartParameters</td>
<td>POST /resources/json /delphix/source/{ref}/start</td>
</tr>
<tr>
<td>Stop</td>
<td>source &quot;name&quot; stop</td>
<td>Source</td>
<td>StopParameters</td>
<td>POST /resources/json /delphix/source/{ref}/stop</td>
</tr>
<tr>
<td>Environment Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUI</td>
<td>CLI</td>
<td>API Topic</td>
<td>Input Object</td>
<td>Web Services</td>
</tr>
<tr>
<td>Add environment</td>
<td>environment create</td>
<td>SourceEnvironme</td>
<td>SourceEnvironme</td>
<td>POST /resources/json /delphix/environment/ {ref}</td>
</tr>
<tr>
<td>Update</td>
<td>environment &quot;name&quot; update</td>
<td>SourceEnvironme</td>
<td>Environment</td>
<td>POST /resources/json /delphix/environment/ {ref}</td>
</tr>
<tr>
<td>Delete</td>
<td>environment &quot;name&quot; delete</td>
<td>SourceEnvironme</td>
<td>-</td>
<td>DELETE /resources/json /delphix/environment/ {ref}</td>
</tr>
<tr>
<td>Refresh</td>
<td>environment &quot;name&quot; refresh</td>
<td>SourceEnvironme</td>
<td>-</td>
<td>POST /resources/json /delphix/environment/ {ref}/refresh</td>
</tr>
<tr>
<td>Enable</td>
<td>environment &quot;name&quot; enable</td>
<td>SourceEnvironme</td>
<td>-</td>
<td>POST /resources/json /delphix/environment/ {ref}/enable</td>
</tr>
<tr>
<td>Disable</td>
<td>environment &quot;name&quot; disable</td>
<td>SourceEnvironme</td>
<td>-</td>
<td>POST /resources/json /delphix/environment/ {ref}/disable</td>
</tr>
<tr>
<td>Add manual repository</td>
<td>repository create</td>
<td>SourceRepository</td>
<td>SourceRepository</td>
<td>POST /resources/json /delphix/repository</td>
</tr>
<tr>
<td>Action</td>
<td>Method</td>
<td>Endpoint</td>
<td>Resource</td>
<td>Request Type</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
<td>--------------</td>
</tr>
<tr>
<td>Update repository</td>
<td>POST</td>
<td>/resources/json/delphix/repository/{ref}</td>
<td>SourceRepository</td>
<td>repository &quot;name&quot; update</td>
</tr>
<tr>
<td>Remove manual repository</td>
<td>DELETE</td>
<td>/resources/json/delphix/repository/{ref}</td>
<td>SourceRepository</td>
<td>repository &quot;name&quot; delete</td>
</tr>
<tr>
<td>Show host details</td>
<td>GET</td>
<td>/resources/json/delphix/host/{ref}</td>
<td>Host</td>
<td>host &quot;name&quot; get</td>
</tr>
<tr>
<td>Add cluster node</td>
<td>POST</td>
<td>/resources/json/delphix/environment/oracle/clusternode</td>
<td>OracleClusterNode</td>
<td>environment oracle clusternode create</td>
</tr>
<tr>
<td>Update cluster node</td>
<td>POST</td>
<td>/resources/json/delphix/environment/oracle/clusternode/{ref}</td>
<td>OracleClusterNode</td>
<td>environment oracle clusternode &quot;name&quot; update</td>
</tr>
<tr>
<td>Remove cluster node</td>
<td>DELETE</td>
<td>/resources/json/delphix/environment/oracle/clusternode/{ref}</td>
<td>OracleClusterNode</td>
<td>environment oracle clusternode &quot;name&quot; delete</td>
</tr>
<tr>
<td>Add listener</td>
<td>POST</td>
<td>/resources/json/delphix/environment/oracle/listener</td>
<td>OracleListener</td>
<td>environment oracle listener create</td>
</tr>
<tr>
<td>Update listener</td>
<td>POST</td>
<td>/resources/json/delphix/environment/oracle/listener/{ref}</td>
<td>OracleListener</td>
<td>environment oracle listener &quot;name&quot; update</td>
</tr>
<tr>
<td>Remove listener</td>
<td>DELETE</td>
<td>/resources/json/delphix/environment/oracle/listener/{ref}</td>
<td>OracleListener</td>
<td>environment oracle listener &quot;name&quot; delete</td>
</tr>
</tbody>
</table>
API Cookbook: Common Tasks, Workflows, and Examples

These topics describe approaches to common tasks and workflows using the Delphix Engine API.

- API Cookbook: Authentication
- API Cookbook: Host Environment Details
- API Cookbook: List Alerts and List Jobs
- API Cookbook: List dSources and VDBs
- API Cookbook: List Snapshots
- API Cookbook: Refresh VDB
- API Cookbook: Example Provision Of An Oracle VDB
- API Cookbook: Stop/Start a VDB
**API Cookbook: Authentication**

This API cookbook recipe describes how to create an authenticated session for using the Delphix Sever web services.

Before you can use any Delphix Web Service API’s you need to create a session, and then authenticate the session by providing valid Delphix account credentials.

<table>
<thead>
<tr>
<th>Create Delphix API Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ curl -s -X POST -k --data @- <a href="http://delphix-server/resources/json/delphix/session">http://delphix-server/resources/json/delphix/session</a> -c ~/cookies.txt -H &quot;Content-Type: application/json&quot; &lt;&lt;EOF</td>
</tr>
<tr>
<td>`{</td>
</tr>
<tr>
<td>&quot;type&quot;: &quot;APISession&quot;,</td>
</tr>
<tr>
<td>&quot;version&quot;: {</td>
</tr>
<tr>
<td>&quot;type&quot;: &quot;APIVersion&quot;,</td>
</tr>
<tr>
<td>&quot;major&quot;: 1,</td>
</tr>
<tr>
<td>&quot;minor&quot;: 1,</td>
</tr>
<tr>
<td>&quot;micro&quot;: 0</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>EOF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delphix Login</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ curl -s -X POST -k --data @- <a href="http://delphix-server/resources/json/delphix/login">http://delphix-server/resources/json/delphix/login</a> -b ~/cookies.txt -H &quot;Content-Type: application/json&quot; &lt;&lt;EOF</td>
</tr>
<tr>
<td>`{</td>
</tr>
<tr>
<td>&quot;type&quot;: &quot;LoginRequest&quot;,</td>
</tr>
<tr>
<td>&quot;username&quot;: &quot;delphix_username&quot;,</td>
</tr>
<tr>
<td>&quot;password&quot;: &quot;delphix_password&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>EOF</td>
</tr>
</tbody>
</table>
API Cookbook: Host Environment Details

This API cookbook recipe describes how to obtain host environment details using the Delphix Engine API.

To obtain details about target host environments, list available Environment objects on the system. These environments can represent either a single host, or an Oracle cluster.

```
List Environment

curl -X GET -k http://delphix-server/resources/json/delphix/environment \\
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For single-host environments, the reference can be used to get information about the associated host. It is also possible to get information about all hosts (regardless of whether they are in a single-host environment or cluster) by omitting the environment query parameter.

```
List UNIX Environment

curl -X GET -k http://services.cloud.skytap.com:23173/resources/json/delphix/host?environment=UNIX_HOST_ENVIRONMENT-1 \\
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information about the content of host objects, see the /api/#Host reference on your local Delphix Engine. Depending on the type of the host, additional information may be available through the following types:

- UnixHost
- WindowsHost
API Cookbook: List Alerts and List Jobs

This API cookbook recipe describes how to obtain lists of jobs and alerts using the Delphix Engine API.

The List Alerts and List Jobs API calls can both accept the toDate and fromDate query parameters to limit rows returned. These parameters require the date to be expressed in ISO 8601 format.

### List Alerts

```bash
$ curl -X GET -k http://delphix-server/resources/json/delphix/alert \
   -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information about the structure of an alert object, see the "/api/#Alert" link on your local Delphix Engine.

### List Jobs (using fromDate)

```bash
$ curl -X GET -k \
   http://delphix-server/resources/json/delphix/job?addEvents=true&fromDate=2012-11-08T00:00:00.0000Z \
   -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information about the structure of a job object, see the "/api/#Job" link on your local Delphix Engine.
API Cookbook: List dSources and VDBs

This API cookbook recipe describes how to obtain a list of dSources and VDBs using the Delphix Engine API.

To obtain a list of dSources and VDBs, list available Container (also known as database) objects on the system:

```
$ curl -X GET -k http://delphix-server/resources/json/delphix/database  \
  -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information on the structure of a database object, see the /api/#Container reference on your local Delphix Engine. The following sub-types are available depending on the type of database:

- OracleDatabaseContainer
- MSSqlDatabaseContainer

Each database has zero or one source associated with it. This source could be a linked source, indicating that the database is a dSource, or it could be a virtual source, indicating that it is a VDB. If there are no sources, it is a detached dSource. The parentContainer property indicates the reference to the parent container, also indicating that the database is a VDB. To get runtime information about the source associated with the dSource or VDB, use the Source API with a database parameter set to the reference of the database in question.

```
$ curl -X GET -k  
  http://delphix-server/resources/json/delphix/source?database=DB_CONTAINER_13  \
  -b ~/cookies.txt -H "Content-Type: application/json"
```

If the virtual flag is true, the source is a VDB, otherwise it is a dSource. For more information about the contents of a source object, see the /api/#Source reference on your local Delphix Engine. The following sub-types are available depending on the type of source:

- OracleSource
  - OracleLinkedSource
  - OracleVirtualSource
- MSSqlSource
  - MSSqlLinkedSource
  - MSSqlVirtualSource
API Cookbook: List Snapshots

This API cookbook recipe describes how to obtain a list of available snapshots for a VDB or dSource.

Snapshots represent points in time where a sync operation has occurred on either a dSource or VDB. Provisioning from snapshots is much faster than provisioning between snapshots, as the latter requires replaying LogSync records to arrive at the requested point. Given a reference to a database, the snapshot API can be used to retrieve the set of snapshots within the database. See the topic API Cookbook: List dSources and VDBs for information on how to obtain the database reference.

List Snapshots

```
```

For more information about the structure of a snapshot object, see the /api/#TimeflowSnapshot reference on your local Delphix Engine. Snapshots, while representing points where provisioning will be most efficient, are not the only provisionable points within a database. To get a list of all provisionable points, use the timeflowRange API. This API is based on a timeflow, which is the representation of one timeline within a database. Currently, all databases have a single timeflow, though this may change in the future. To query for the ranges for a particular database, you will need to use the currentTimeflow member of the target database.

List Timeflow Ranges

```
```
API Cookbook: Refresh VDB

This API cookbook recipe describes how to refresh a VDB using the Delphix Engine API.

To refresh a VDB you need a reference to the Database object, and the location of the point you wish to refresh one. See the topic API Cookbook: List dSources and VDBs for information on how to obtain the database reference and current timeflow. The timeflow point can be specified either by timestamp, by location (SCN), semantic location or timeflow bookmark. The TimeflowPointSemantic type allows you to specify a semantically meaningful timeflow location (i.e. the latest snapshot or the latest timeflow point). The TimeflowPointBookmark type allows you to reference a previously created timeflow bookmark. See the API Cookbook: List Snapshots topic for information on how to determine provisionable points in the parent database.

```
curl -v -X POST -k --data @-
http://delphix-server/resources/json/delphix/database/ORACLE_DB_CONTAINER-13/refresh
\n   -b ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
   "type": "OracleRefreshParameters",
   "timeflowPointParameters": {
      "type": "TimeflowPointSemantic",
      "timeflow": "ORACLE_TIMEFLOW-13",
      "location": "LATEST_SNAPSHOT"
   }
}
EOF
```

For more information about the content of refresh parameters, see the /api/#RefreshParameters reference on your local Delphix Engine. Depending on the type of the database, the following parameter types are available:

- OracleRefreshParameters
- MSSqlRefreshParameters
API Cookbook: Example Provision Of An Oracle VDB

This API cookbook recipe demonstrates how to provision an Oracle VDB using the Delphix Engine API.

In order to provision an Oracle VDB using the API, you need to provide a set of parameters of type OracleProvisionParameters (having already authenticated as per API Cookbook: Authentication).

There are a number of parameters you will need to know:

- **Group reference** - See the list operation in "/api/#group" on your Delphix Engine.
- **VDB name** - The name you want the new VDB to be called.
- **Mount path** - Where to mount datasets on the target host.
- **DB/unique names** - The Oracle DB and unique names, often the same as the VDB name.
- **Instance name/number** - The Oracle instance name and number to use (dictated by your environment, but often VDB name and 1).
- **Repository reference** - See the list operation in "/api/#repository" on your Delphix Engine.
- **Timeflow point** - See API Cookbook: List Snapshots for information on finding a timeflow point, as well as the reference at "/api/#TimeflowPointParameters".

You will need to use the structure of the OracleProvisionParameters object to fill it out, see "/api/#OracleProvisionParameters" for details on which fields are mandatory/optional.
Here is a minimal example using curl to communicate with the API, provisioning a VDB called "EGVDB" (authentication omitted)

curl -X POST -k --data @-
http://delphix1.company.com/resources/json/delphix/database/provision \
-b cookies.txt -H "Content-Type: application/json" <<EOF
{
  "container": {
    "group": "GROUP-2",
    "name": "EGVDB",
    "type": "OracleDatabaseContainer"
  },

  "source": {
    "type": "OracleVirtualSource",
    "mountBase": "/mnt/provision"
  },

  "sourceConfig": {
    "type": "OracleSIConfig",
    "databaseName": "EGVDB",
    "uniqueName": "EGVDB",
    "repository": "ORACLE_INSTALL-3",
    "instance": {
      "type": "OracleInstance",
      "instanceName": "EGVDB",
      "instanceNumber": 1
    }
  },

  "timeflowPointParameters": {
    "type": "TimeflowPointLocation",
    "timeflow": "ORACLE_TIMEFLOW-123",
    "location": "3043123"
  },

  "type": "OracleProvisionParameters"
}
EOF
API Cookbook: Stop/Start a VDB

This API cookbook recipe describes how to stop and start a VDB using the Delphix Engine API.

To stop or start a VDB, you need a reference to the `Database` object. See the topic, API Cookbook: List dSources and VDBs, for information on how to obtain the database reference. The following script example includes working examples for creating a session, authenticating to the Delphix Engine, and stopping or starting a VDB. Please update the script variables to match your environment before using. This script requires a single argument which is 'start' or 'stop'.

```python
# Import required modules
import delphix_engine_api as de_api

# Create a session
session = de_api.Session()

# Authenticate to the Delphix Engine
session.auth(username='your_username', password='your_password', server='your_server')

# Get a reference to the database
database = de_api.Database(session, database_name='your_db_name')

# Stop the VDB
if sys.argv[1] == 'stop' or sys.argv[1] == 'Stop':
    database.stop()

# Start the VDB
if sys.argv[1] == 'start' or sys.argv[1] == 'Start':
    database.start()
```
#!/bin/bash
# sample script to start or stop a VDB.
#
# set this to the FQDN or IP address of the Delphix Engine
DE="192.168.2.131"
# set this to the Delphix admin user name
DELPHIX_ADMIN="delphix_admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# set this to the object reference for the VDB
VDB="ORACLE_VIRTUAL_SOURCE-5"
#
# create our session
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/session \
   -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
   "type": "APISession",
   "version": {
      "type": "APIVersion",
      "major": 1,
      "minor": 4,
      "micro": 1
   }
}
EOF

# authenticate to the DE
echo
#
# start or stop the vdb based on the argument passed to the script
fail() { echo "Unknown option: $1" }
case $1 in
  start)
    curl -s -X POST -k http://${DE}/resources/json/delphix/source/${VDB}/start \
       -b ~/cookies.txt -H "Content-Type: application/json"
    ;;
  stop)
    curl -s -X POST -k http://${DE}/resources/json/delphix/source/${VDB}/stop \
       -b ~/cookies.txt -H "Content-Type: application/json"
    ;;
  *)
    echo "Unknown option: $1"
    ;;
esac

echo