DataZen Overview

SECURED CORPORATE AND B2B DATA REPLICATION

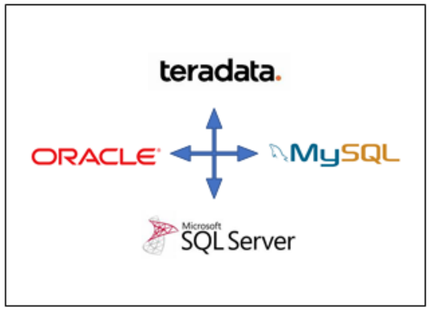
Organizations looking for ways to share their data or files across disparate database systems, remote offices, or between business partners in a secure way now have the option to do so with DataZen. With DataZen, companies can now establish a replication mechanism between disconnected sites and share data or files securely from any source to any target platform.

The Problem Space

DataZen is designed to solve a seemingly simple use case: sharing data. While most organizations share data many different ways, the way this is done usually depends on the recipient. Companies share data through file exports, emails, FTP sites and more. However, sharing becomes hard when the data needs to be shared between companies, or from remote locations to a central site, or even between systems that are not designed to communicate with each other. Let's explore a few scenarios that expose the challenges at hand.

Cross Database Replication and Data Copy

In some cases, organizations need to replicate data between disparate database platforms, such as synchronizing data between an Oracle database and a SQL Server or Azure SQL Database platform, or between MySQL and SQL Server. At times, organizations need to copy full data sets across database engines on a regular schedule, such as daily dashboards, month-end reporting, or general integration purposes. With DataZen, administrators can setup data replication between any number of database platforms, and configure automatic data synchronization when source data changes.



Typical Concerns:

Disparate Platforms

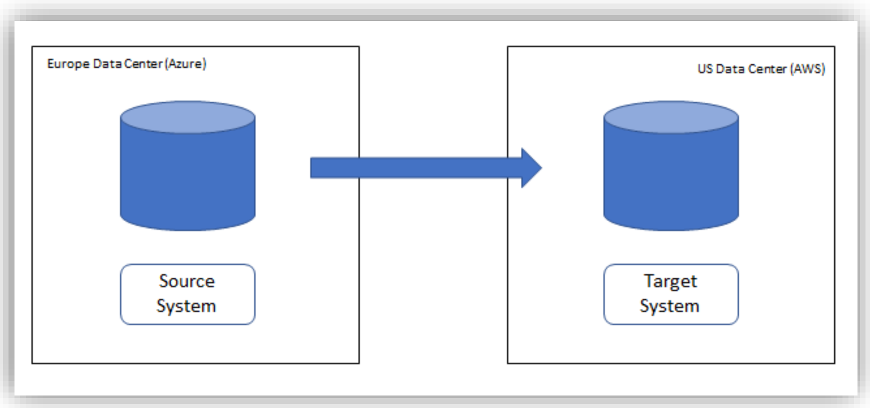
Disparate Data Types

Data Centralization for Reporting

Data Migration / Consolidation

Remote Site Data Replication

Some organizations operate multiple facilities that may not be able to share information easily due to regulatory concerns, security requirements, organizational needs or due to geographic distance. For example, an organization may experience networking limitations that prevent replicating data from a MySQL database located in Europe in the Microsoft Azure Cloud with a central SQL Server database located in the United States running in the Amazon AWS cloud. Another issue may be with unpredictable network availability or speed, preventing long-distance replication from two Oracle databases.



Typical Concerns:

Network Connection

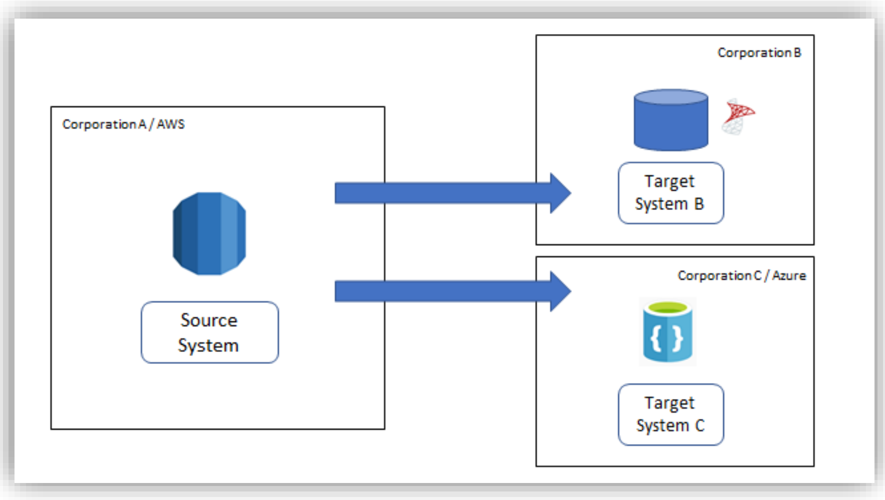
Network Latency

Cross Data Center

Disparate Systems

B2B Data Sharing

Another scenario that be difficult to solve involves sharing data between two corporations. Sharing data between two or more businesses usually involves additional security layers to ensure only the intended recipients can read the data, requires sufficient controls to ensure only the relevant data is shared with business partners, and complete independence so that none of the systems depend on each other's availability. For example, Corporation A may want to share its Oracle data running on AWS RDS with two business partners, one running SQL Server, and another storing the data in a no-sql CosmosDB database in Azure.



Typical Concerns:

Network Connection

Network Latency

Cross Data Center

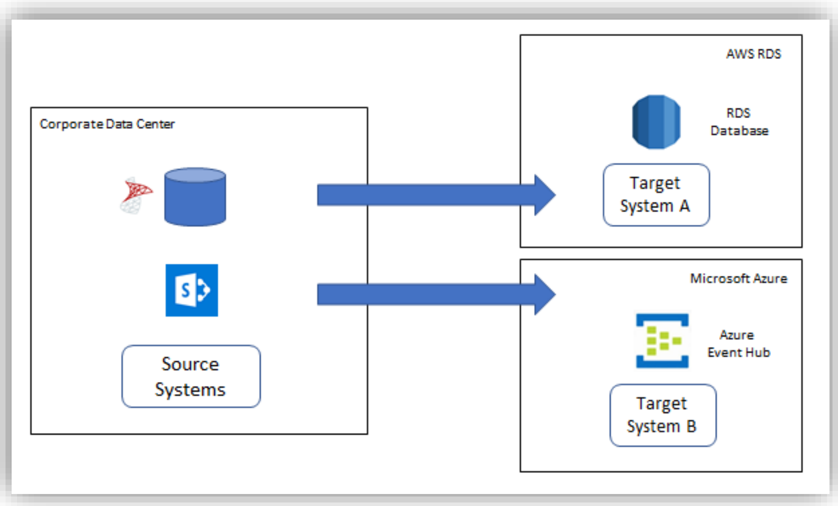
Disparate Systems

Strong Security

System Independence

On-Prem to Cloud Replication

In some cases, organizations that migrate their on-prem systems to the cloud face the issue of data migration. Larger companies will have systems running in their data center (such as DB2, or Teradata), and need to partially replicate this data to the cloud to enable dashboards (such as PowerBI) or feed new cloud-native applications. The challenge is two-fold: the source system is usually different than the target system, and network latency forces companies to only send changes (Change Data Capture, or CDC in short) to the cloud. However few systems support CDC natively; this means that changes must be identified by an external engine. This issue becomes more important as the number of target systems grows over time, and when the number of tables/systems to replicate increases due to evolving business requirements. For example, a company wants to replicate its data, and ongoing changes, from two systems (SQL Server and SharePoint Online), and forward data to both Azure and AWS on different platforms, or forward changes to a cloud messaging hub such as the Azure Event Hub, or the AWS SQS service.



Typical Concerns:

Network Connection

Network Latency

Cross Data Center

Disparate Systems

Change Data Capture

Cloud Messaging

Solution

As discussed previously, sharing data across systems can be challenging for a few reasons:

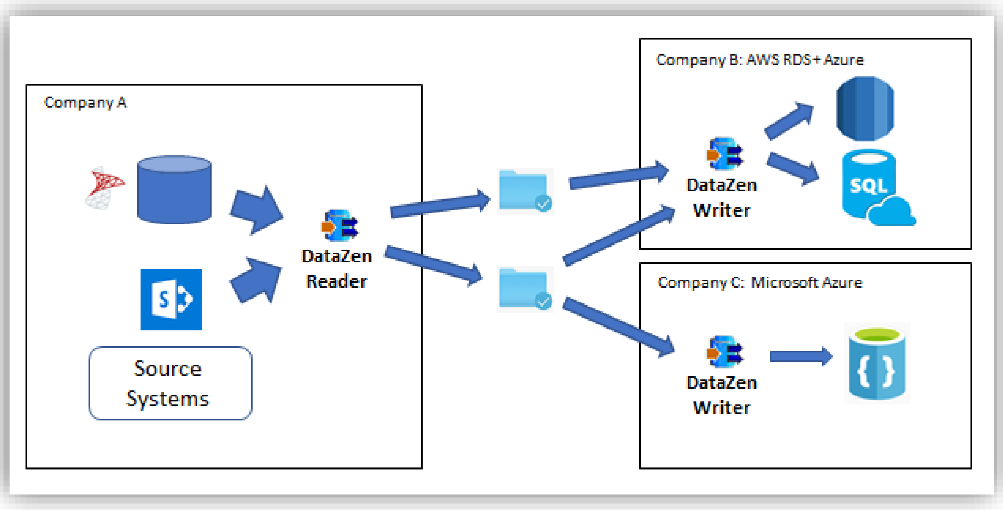
- Network latency: not all systems participating in the replication topology are always connected
- Security: sharing data changes can involve the exchange of sensitive data requiring encryption
- Network bandwidth: identifying and sending changes to target systems (except an initial load)
- Disparate systems: the source systems may be completely different that the target systems

DataZen solves these challenges by decoupling the source and target system(s) entirely. DataZen is made up of two types of agents: Readers and Writers.

A Reader will be configured to extract source data, regardless of the source system, and creates a Change Log (also called a Data Sync File) on a network location, or a cloud drive for example. Then Writer agents run in the target environments; these writers wait for the “next” Change Log to be published. When one becomes available, the Writer Agent pushes changes onto to the target system(s).

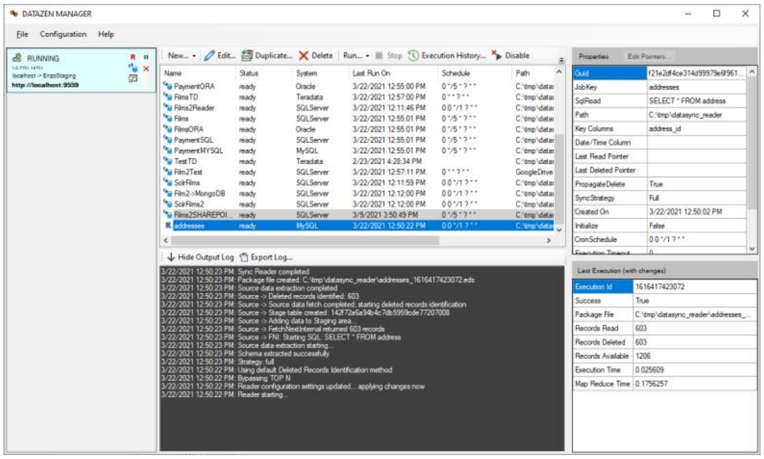
In a simple deployment scenario, the Reader and Writer agents could be running on the same machine, replicating data from two relational databases. In a more complex deployment scenario, Corporation A runs a Reader Agent, shares data through an AWS Bucket using PGP encrypted Change Logs, and

Corporations B and C run an Agent Writer pushing changes to their respective systems at the desired interval. This architecture provides complete autonomy between systems.



Capabilities

At a high level, DataZen is a technology that runs as a windows service; the windows service can be configured as a Reader and/or Writer agent. When running as a Reader, the agent will query the source system to retrieve all its available data (usually a table), and run a CDC engine that will identify changes from the previous successful run. If any changes are identified, a change log is created using a sequence number, allowing Writer agents to play back changes in the correct order.



Name	Status	System	Last Run On	Schedule	Path
PaymentORA	ready	Oracle	3/22/2021 12:55:00 PM	0 */5 * ? * *	C:\tmp\datas
FilmsTD	ready	Teradata	3/22/2021 12:57:00 PM	0 * * ? * *	C:\tmp\datas
Films2Reader	ready	SQLServer	3/22/2021 12:11:46 PM	0 0 */1 ? * *	C:\tmp\datas
Films	ready	SQLServer	3/22/2021 12:55:01 PM	0 */5 * ? * *	C:\tmp\datas
FilmsORA	ready	Oracle	3/22/2021 12:55:01 PM	0 */5 * ? * *	C:\tmp\datas
PaymentSQL	ready	SQLServer	3/22/2021 12:55:01 PM	0 */5 * ? * *	C:\tmp\datas
PaymentMYSQL	ready	MySQL	3/22/2021 12:55:01 PM	0 */5 * ? * *	C:\tmp\datas
TestTD	ready	Teradata	2/23/2021 4:28:34 PM		C:\tmp\datas
Film2Test	ready	SQLServer	3/22/2021 12:57:11 PM	0 * * ? * *	GoogleDrive
SolrFilms	ready	SQLServer	3/22/2021 12:11:59 PM	0 0 */1 ? * *	C:\tmp\datas
Film2->MongoDB	ready	SQLServer	3/22/2021 12:12:00 PM	0 0 */1 ? * *	C:\tmp\datas
SolrFilms2	ready	SQLServer	3/22/2021 12:12:00 PM	0 0 */1 ? * *	C:\tmp\datas
Films2SHAREPOI...	ready	SQLServer	3/9/2021 3:50:49 PM	0 */5 * ? * *	C:\tmp\datas
addresses	ready	MySQL	3/22/2021 12:50:22 PM	0 0 */1 ? * *	C:\tmp\datas

```
3/22/2021 12:50:23 PM: Sync Reader completed
3/22/2021 12:50:23 PM: Package file created: C:\tmp\datasync_reader\addresses_1616417423072.eds
3/22/2021 12:50:23 PM: Source data extraction completed
3/22/2021 12:50:23 PM: Source -> Deleted records identified: 603
3/22/2021 12:50:23 PM: Source -> Source data fetch completed; starting deleted records identification
3/22/2021 12:50:23 PM: Source -> Stage table created: 142f72a6a94b4c7db5959cde77207008
3/22/2021 12:50:23 PM: Source -> Adding data to Staging area...
3/22/2021 12:50:23 PM: Source -> FetchNextInternal returned 603 records
3/22/2021 12:50:23 PM: Source -> FNI: Starting SQL: SELECT * FROM address
3/22/2021 12:50:23 PM: Source data extraction starting...
3/22/2021 12:50:23 PM: Schema extracted successfully
3/22/2021 12:50:23 PM: Strategy: full
3/22/2021 12:50:22 PM: Using default Deleted Records Identification method
3/22/2021 12:50:22 PM: Bypassing TOP N
3/22/2021 12:50:22 PM: Reader configuration settings updated... applying changes now
3/22/2021 12:50:22 PM: Reader starting...
```

Guid	f21e2df4ce314d99979e6f961...
JobKey	addresses
SqlRead	SELECT * FROM address
Path	C:\tmp\datasync_reader
Key Columns	address_id
Date/Time Column	
Last Read Pointer	
Last Deleted Pointer	
PropagateDelete	True
SyncStrategy	Full
Created On	3/22/2021 12:50:02 PM
Initialize	False
CronSchedule	0 0 */1 ? * *
Execution Timeout	0

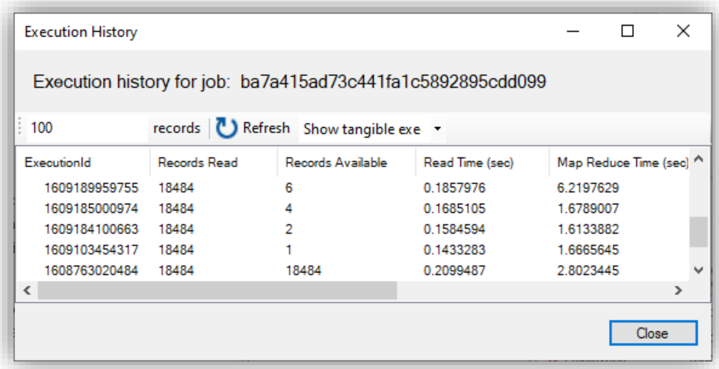
Last Execution (with changes)

Execution Id	1616417423072
Success	True
Package File	C:\tmp\datasync_reader\addresses_...
Records Read	603
Records Deleted	603
Records Available	1206
Execution Time	0.025609
Map Reduce Time	0.1756257

This screenshot shows a list of Reader and Writer jobs; these jobs are configured to listen for changes from source systems, and push those changes to target databases.

Agent History

Each job history can be viewed directly to review the change log summary. For example, below is a change log showing an initial extraction of 18,484 records, and additional changes captured over time, providing a full audit of agent executions.

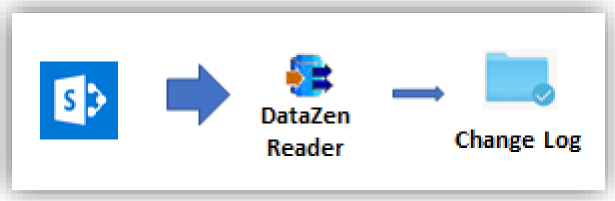


Execution history for job: ba7a415ad73c441fa1c5892895cdd099

ExecutionId	Records Read	Records Available	Read Time (sec)	Map Reduce Time (sec)
1609189959755	18484	6	0.1857976	6.2197629
1609185000974	18484	4	0.1685105	1.6789007
1609184100663	18484	2	0.1584594	1.6133882
1609103454317	18484	1	0.1433283	1.6665645
1608763020484	18484	18484	0.2099487	2.8023445

Changed and Deleted Data

DataZen allows you to setup a two-way replication topology, from any source system to any target, even if the systems are different in nature, such as relational databases and SharePoint, or between GoogleSheet documents and Azure Tables. DataZen has a built-in CDC engine that detects changes from source systems over time, even if the system doesn't provide CDC capabilities. This engine can also detect records that have been deleted; the Change Log file created will contains both changed records (new or updated) and deleted records. Although the Change Log may contain deleted records, the Writer agent has the option not to carry over deletions; this can be useful for auditing purposes.

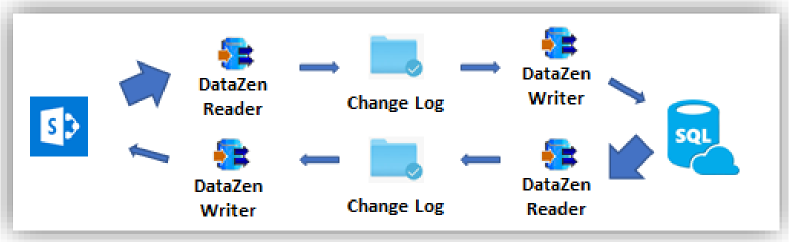


A change log is created when changes (including deletions) are detected

Two-Way Replication

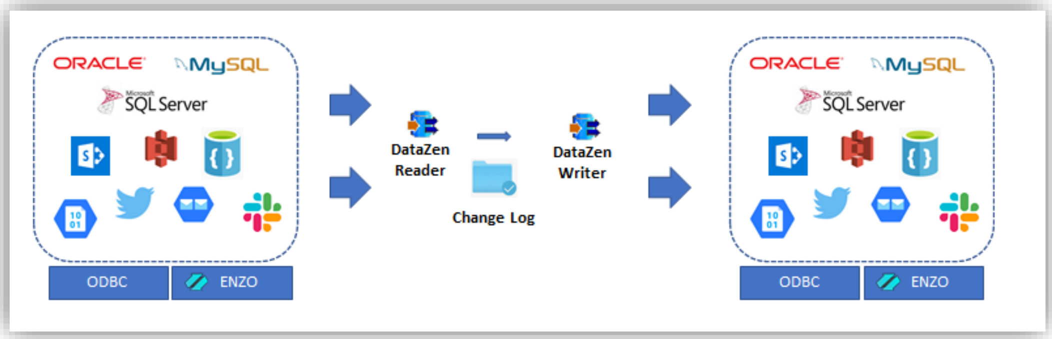
Because DataZen implements a CDC capability by reading from the source system and comparing changes from the previous read operation, it is possible to implement a two-way replication topology in most situations. The main objective of a two-way replication topology is to push changes both ways, but only as many times as necessary to achieve consistency between two systems. Usually this means that "Updated" date fields, or timestamps, should not be replicated when implementing two-way replication.

A two-way replication topology is possible in most scenarios



ODBC and Data Virtualization Compliant

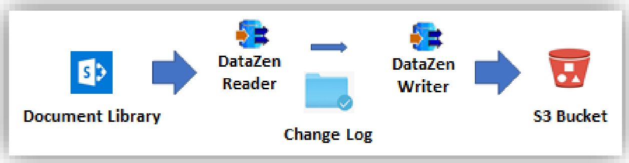
Because DataZen can communicate with native SQL Server databases and ODBC drivers, DataZen can extract and replicate data from virtually any source system to any target platform. Because the Change Log store is not system specific, the target system can be completely different than the source system where the data came from.



Replicate changes from any source system to any target quickly

File Replication

It is possible to replicate files to target locations using DataZen, as long as the ODBC driver or Enzo Server is used to access the drive where the files are located. Using this approach, it is possible to replicate file changes from a SharePoint Document Library to an Azure Container for example, or from an FTP site to an S3 Bucket.

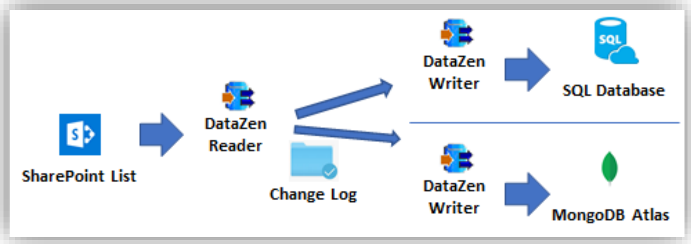


Replicate from a SharePoint Document Library to an S3 Bucket

Multicast Replication

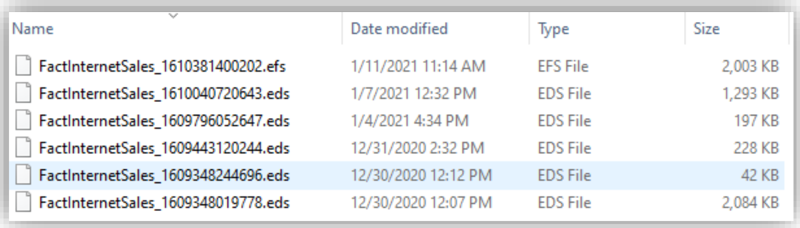
DataZen is designed to create platform-independent Change Log files and creating a strong separation between the source system and the target platforms. As a result, it is possible to apply the same Change Log multiple times, to any number of target systems. For example, a SharePoint List is replicated to both a SQL Database and a MongoDB Atlas database.

Change Logs can be applied to multiple systems independently



Replay and Resync

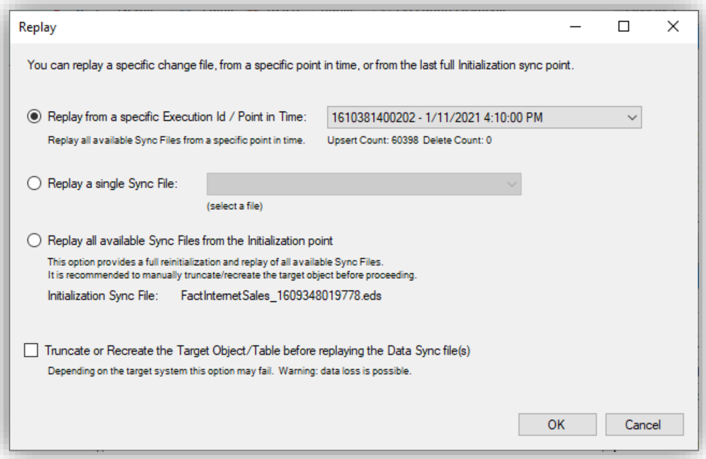
DataZen is designed to create platform-independent Change Log files, using a naming convention that automatically orders the Change Logs so that Writer Agents can determine which change log to apply next, even if the files are encrypted. The Change Logs are created with a sequence number which grows over time to help determine the proper sequence in which to apply the changes. The Writer Agent keeps track of the last applied Change Log and waits for the next one to become available.



Name	Date modified	Type	Size
FactInternetSales_1610381400202.efs	1/11/2021 11:14 AM	EFS File	2,003 KB
FactInternetSales_1610040720643.eds	1/7/2021 12:32 PM	EDS File	1,293 KB
FactInternetSales_1609796052647.eds	1/4/2021 4:34 PM	EDS File	197 KB
FactInternetSales_1609443120244.eds	12/31/2020 2:32 PM	EDS File	228 KB
FactInternetSales_1609348244696.eds	12/30/2020 12:12 PM	EDS File	42 KB
FactInternetSales_1609348019778.eds	12/30/2020 12:07 PM	EDS File	2,084 KB

Change Logs use a specific naming convention to help with proper sequencing

From the management application an administrator can choose to reapply all changes from the beginning, a single Change Log, or from a specific point in time.



You can replay a specific change file, from a specific point in time, or from the last full Initialization sync point.

● Replay from a specific Execution Id / Point in Time: 1610381400202 - 1/11/2021 4:10:00 PM

Replay all available Sync Files from a specific point in time.

Upsert Count: 60398 Delete Count: 0

○ Replay a single Sync File:

(select a file)

○ Replay all available Sync Files from the Initialization point

This option provides a full reinitialization and replay of all available Sync Files.
It is recommended to manually truncate/recreate the target object before proceeding.

Initialization Sync File: FactInternetSales_1609348019778.eds

□ Truncate or Recreate the Target Object/Table before replaying the Data Sync file(s)

Depending on the target system this option may fail. Warning: data loss is possible.

Replay Change Logs with the DataZen management application. Choose a single Change Log, from a specific point in time, or all available.

Conclusion

DataZen is a modern data replication technology that allows companies to securely share and replication data across systems, between remote locations, and between business partners. By leveraging ODBC drivers and Data Virtualization technologies such as Enzo Server, DataZen can replicate data from any source system to any target platform, and offers advanced capabilities such as PGP encryption, replay and resync, multicast and two-way replication.

For more information, or to try DataZen for 30 days free of charge, contact us at info@bluesyntaxconsulting.com.