

# Dell Compellent Storage Center DB2 UDB backup and recovery with CML data instant replay

---

Best Practices



Compellent

## Document revision

Date	Revision	Description
5/20/2010	A	Preliminary

THIS BEST PRACTICES GUIDE IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2011 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

*Dell*, the *DELL* logo, the *DELL* badge, and Compellent are trademarks of Dell Inc. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

# Contents

- Document revision ..... 2
- Contents ..... 3
  - General syntax ..... 5
  - Conventions ..... 5
- Preface ..... 6
  - Scope ..... 6
  - Audience ..... 6
  - Customer support ..... 6
- DB2 UDB overview ..... 7
  - DB2 instances ..... 7
  - Databases ..... 8
  - Table spaces ..... 8
    - Catalog table space* ..... 8
    - Regular table spaces* ..... 9
    - Long table spaces* ..... 9
    - System temporary table spaces* ..... 9
    - User temporary table spaces* ..... 9
    - Table space management* ..... 9
- Containers ..... 10
- DB2 components used for backup and recovery ..... 11
  - Transaction log files* ..... 11
  - Logging modes* ..... 11
  - Recovery history File* ..... 11
  - Recover methods* ..... 11
  - Non-recoverable database* ..... 12
  - Recoverable database* ..... 12
  - Database consistency (suspend I/O)* ..... 12
  - The db2inidb and db2rfpen commands* ..... 12
- Dell Compellent overview ..... 13
  - Dell Compellent consistency group ..... 13
  - Dell Compellent data instant replay ..... 13

Dell Compellent Storage Center DB2 UDB backup and recovery  
with CML data instant replay Best Practices

- Create consistency group profile ..... 14
  - Creating volumes ..... 14
  - Creating consistency group replay profile ..... 15
- Create DB2 database backups ..... 19
  - Create an offline backup ..... 19
  - Create an online backup..... 19
- Restore DB2 databases ..... 20
  - Restore a non-recoverable database from an offline backup ..... 20
  - Restore a non-recoverable database from an online backup ..... 20
  - Restore a recoverable database from an offline backup ..... 21
  - Restore a recoverable database from an online backup ..... 22
- Conclusion ..... 23

**Tables**

- Table 1. Document syntax..... 5

## General syntax

**Table 1. Document syntax**

Item	Convention
Menu items, dialog box titles, field names, keys	<b>Bold</b>
Mouse click required	Click:
User Input	Monospace Font
User typing required	Type:
Website addresses	<a href="http://www.compellent.com">http://www.compellent.com</a>
Email addresses	<a href="mailto:info@compellent.com">info@compellent.com</a>

## Conventions



Note

Notes are used to convey special information or instructions.



Timesaver

Timesavers are tips specifically designed to save time or reduce the number of steps.



Caution

Caution indicates the potential for risk including system or data damage.



Warning

Warning indicates that failure to follow directions could result in bodily harm.

## Preface

### Scope

This document describes the best practices on how to use Dell™ Compellent™ Storage Center™ Data Instant Replay feature with DB2 UDB databases for backup and recovery. It shows the step by step on how to create a database backup using Data Instant Replay (snapshot) technology. It is also assumed that readers are familiar with DB2 ESE V9 and its backup and recovery mechanisms and tested before implementing in production.

### Audience

The primary target audience for this best practice is database administrators, system administrators, storage administrators, and architects who analyze, design, and maintain a robust database and storage system. Readers should be familiar with DB2 UDB table space containers and how they can affect the performance of a DB2 UDB database.

### Customer support

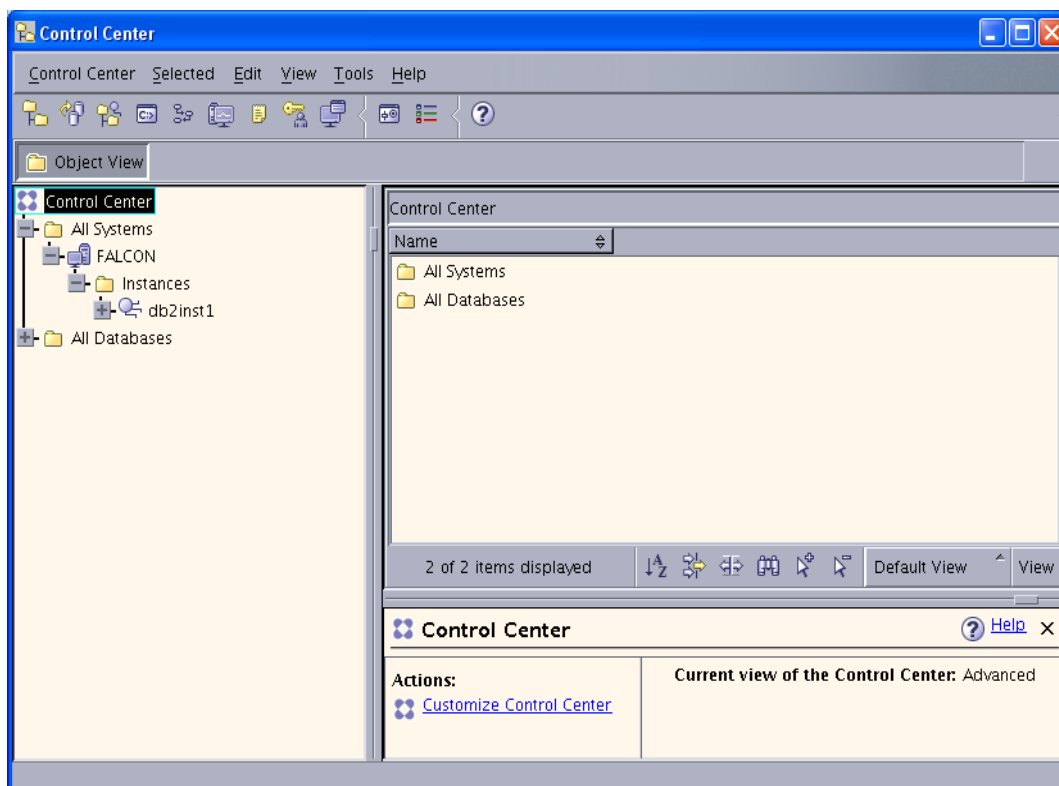
Dell Compellent provides live support 1-866-EZSTORE (866.397.8673), 24 hours a day, 7 days a week, 365 days a year. For additional support, email Dell Compellent at [support@compellent.com](mailto:support@compellent.com). Dell Compellent responds to emails during normal business hours.

## DB2 UDB overview

### DB2 instances

An instance in DB2 for LUW is like a copy of RDBMS including all the processes that run DB2 and memory associated with that instance and some configuration parameters to control that instance. Anytime a new instance is created, that instance references the DB2 Database Manager program files that were stored on that server during the installation process; thus, each instance behaves like a separate installation of DB2 UDB.

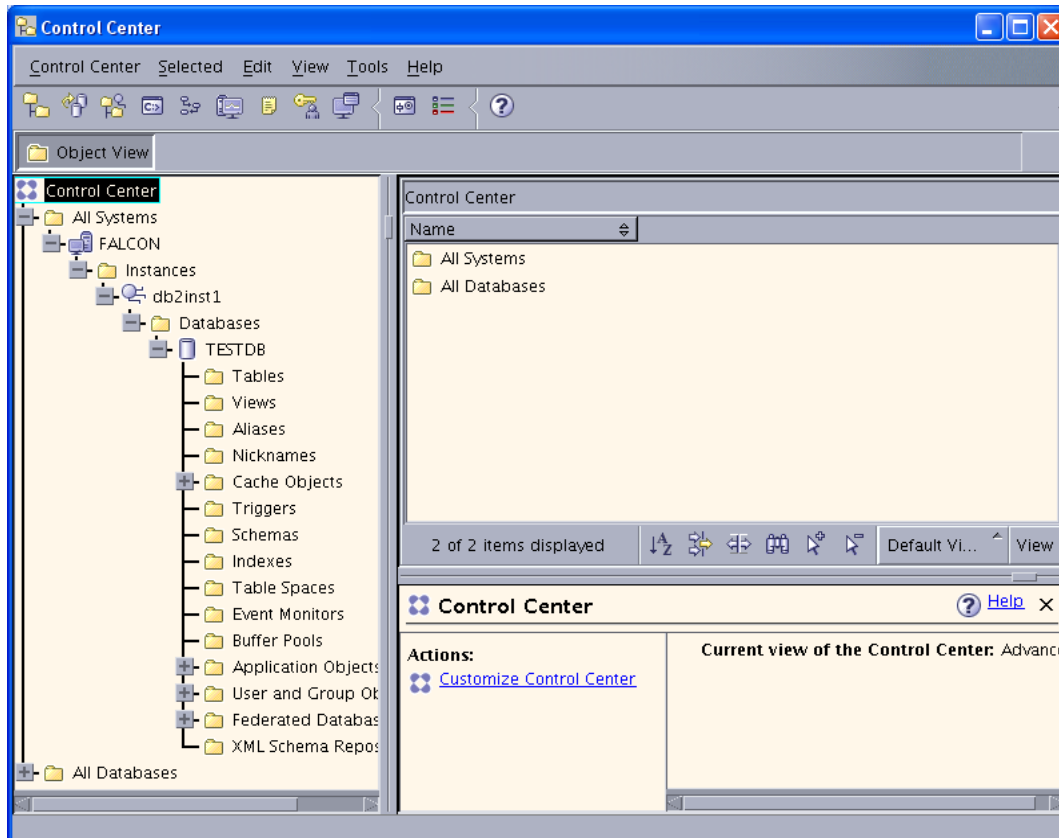
Every instance controls access to one or more databases. Every database within an instance is assigned a unique name, has its own set of system catalog tables, and has its own configuration file. Below is an example of an instance named db2inst1.



## Databases

A DB2 UDB database is a set of related objects. When you create a DB2 UDB database, you are establishing an administrative relational database entity that provides an underlying structure for an eventual collection of related objects such as tables, views, indexes, table spaces, etc.

Below is an example of a database named TESTDB.



## Table spaces

All data for a database is stored in a number of table spaces. You can think of a table space as being a child and a database as its parent, where the table space (child) cannot have more than one database (parent). Because there are different uses for table spaces, they are classified according to their usage and how they will be managed. There are five different table spaces by usage:

### Catalog table space

There is only one catalog table space per database, and it is created when the CREATE DATABASE command is issued. Named SYSCATSPACE by DB2, the catalog table space holds the system catalog tables. This table space is always created when the database is created.



### **Regular table spaces**

Regular table spaces hold table data and indexes. It can also hold long data such as large objects (LOBs) unless they are explicitly stored in long table spaces. A table and its indexes can be segregated into separate regular table spaces, if the table spaces are database managed space (DMS). We will define the differences between DMS and system managed space (SMS) later in this white paper. At least one regular table space must exist for each database. The default is named USERSPACE1 when the database is created.

### **Long table spaces**

Long table spaces are used to store long or LOB table columns and must reside in DMS table spaces. They can also store structured type columns or index data. If no long table space is defined, then LOBs will be stored in regular table spaces. Long table spaces are optional and none will be created by default.

### **System temporary table spaces**

System temporary table spaces are used to store internal temporary data required during SQL operations such as sorting, reorganizing tables, creating indexes, and joining tables. At least one must exist per database. The default created with the database is named TEMPSPACE1.

### **User temporary table spaces**

User temporary table spaces store declared global temporary tables. No user temporary table spaces exist when a database is created. At least one user temporary table space should be created to allow definition of declared temporary tables. User temporary table spaces are optional and none will be created by default.

### **Table space management**

#### **System Managed Space (SMS)**

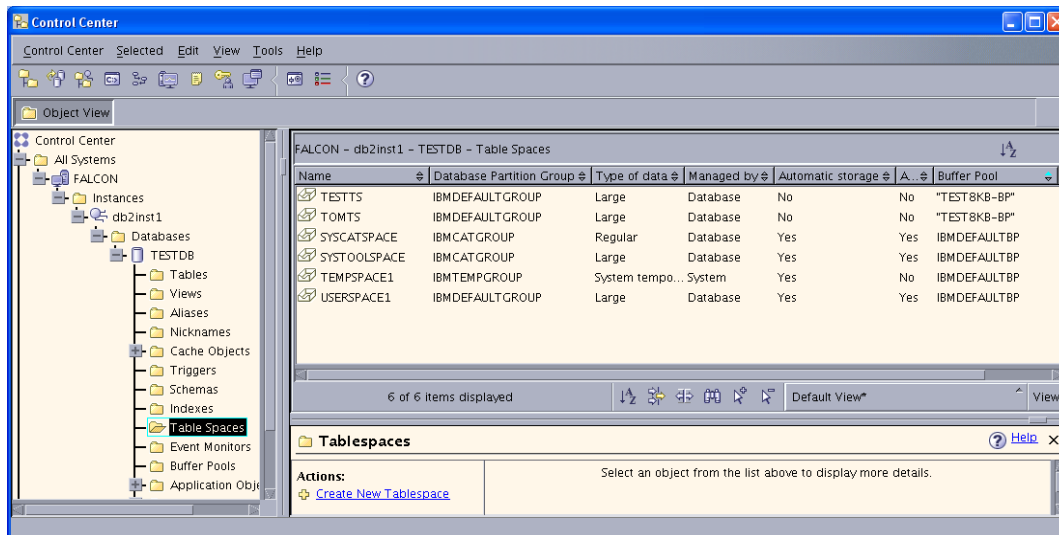
SMS table spaces are managed by the operating system. Containers are defined as regular operating system files and they are accessed via operating system calls. This means that all the regular operating system functions will handle the following: I/O will be buffered by the operating system, space will be allocated according to the operating system conventions, and the table space is automatically extended when it is necessary. However, containers cannot be dropped from SMS table spaces, and adding new ones is restricted to partitioned databases. The three default table spaces explained in the previous section are SMS.

#### **Database Managed Space (DMS)**

DMS table spaces are managed by DB2. Containers can be defined either as files (which will be fully allocated with the size given when the table space is created) or devices. DB2 will manage as much of the I/O as the allocation method and the operating system will allow. Extending the containers is possible by using the ALTER TABLESPACE command. Unused portions of DMS containers can be also released (starting with version 8).

## DELL Compellent Storage Center DB2 UDB backup and recovery with CML data instant replay Best Practices

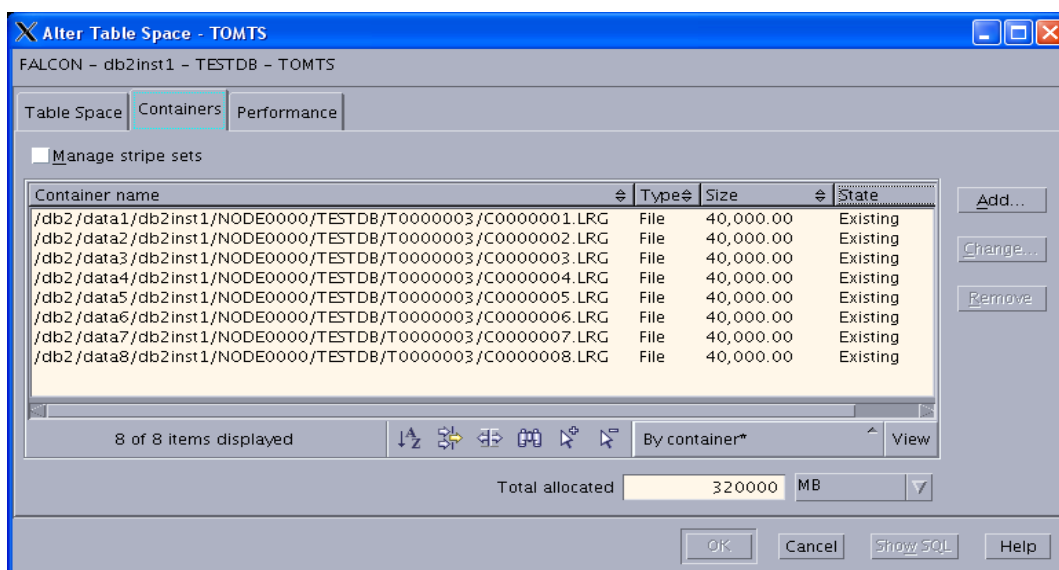
Below is an example of the table spaces.



## Containers

Every table space has one or more containers. Again, you might think of a container as being a child and a table space as its parent. Each container can only belong to a single table space but a table space can have many containers. Containers can be added to or dropped from a DMS table space, and their sizes can be modified. Containers can only be added to SMS table spaces on partitioned databases in a partition, which does not yet have a container allocated for the table space. When new containers are added, an automatic rebalancing will start to distribute the data across all containers. Rebalancing will not prevent concurrent access to the database.

Below is an example of containers.



## DB2 components used for backup and recovery

### Transaction log files

- **Active/online logs:** Log files with contents that have not yet been applied to the database's data files are called active or online log files. The online log files are created in the active database log directory and are required to perform crash as well as roll-forward recovery.
- **Archive/offline logs:** Log files with contents that have been applied to the database's data files. They are not required for crash recovery but are crucial for roll-forward recovery. An active log file becomes a candidate for archiving as soon as its contents are written to the data files and may be moved automatically from the active log directory to an archive location.

### Logging modes

- **Circular logging:** Only online logs are retained and log files are used in round-robin fashion. The online logs are used for crash recovery only. The primary means of database recovery is version recovery from a full database backup. Circular logging overwrites logs that are committed.
- **Archive logging:** Both online and the archived logs are retained. Online logs are used for crash recovery. For roll-forward recovery, both online logs and archive logs are used. Upon commit, log files are closed and become available for offline archiving. You can enable archive logging by setting the *logarchmeth1* and *logarchmeth2* parameters to a value other than OFF in the database configuration.

### Recovery history File

- DB2 uses a special file called Recovery History File as a repository to keep track of database activities such as database or tablespace backups, database or tablespace restores, roll-forward, alter, created, quiesced, renamed, dropped, loaded, or table re-org. The command to execute is DB2 LIST HISTORY.

### Recover methods

- **Crash recovery:** Database transactions (units of work) can be interrupted unexpectedly. If a failure occurs before all of the changes that are part of the unit of work are completed and committed, the database is left in an inconsistent and unusable state. Crash recovery is the process by which the database is returned to a consistent and usable state. This is done by rolling back incomplete transactions and completing committed transactions that were still in memory when the crash occurred. (Transaction records and corresponding commit records stored in transaction log files are used to return the database to a consistent state.) The order in which statements are rolled back is the reverse of the order in which they were originally executed.
- **Version recovery:** This type of recovery allows for the restoration of a previous version of a database using a backup copy of the database that was created by the DB2 backup utility. This method of recovery is normally used for restoring a non-recoverable database. You can also use this method to restore a recoverable database if the DB2 restore utility is invoked with the WITHOUT ROLLING FORWARD option specified. A version recovery reconstructs an entire database using a backup copy created earlier.

- **Roll-forward recovery:** Roll-forward recovery extends version recovery by using full database backups in conjunction with archive and active log files. When a roll-forward recovery operation is performed, the database must first be restored from a backup copy, and then transaction records stored in log files are applied to the restored database. This procedure returns a database or a tablespace to the state it was in at a particular point in time. The roll-forward recovery requires archival logging to be enabled.

### **Non-recoverable database**

- A DB2 database is considered to be non-recoverable if it is using circular logging. For circular logging, only active logs required for crash recovery are retained. Version recovery is the primary method used to restore a damaged database; roll-forward recovery is not possible, and such a database can be restored only to the state it was in at the time the backup image being used for recovery was created.

### **Recoverable database**

- A DB2 database is considered to be recoverable if it has archive logging enabled and the *logarchmeth1* and *logarchmeth2* configuration parameters have been set to a value other than the value OFF. The active logs as well as archive logs are available for recovery; active logs are used for crash recovery and archive logs make roll-forward recovery possible. If a database is recoverable, tablespace level backup and recovery are possible as well.

### **Database consistency (suspend I/O)**

- To create a consistent online backup for a recoverable database using Dell Compellent data instant replay, all write operations must be suspended temporarily before invoking the Replay command. You can suspend I/O by issuing DB2 SET WRITE SUSPEND FOR DATABASE. After issuing the suspend command, all writes to the database and its logs are suspended. Only read operations are allowed. Any transaction requires disk I/O during the suspend mode will proceed normally once the write operation on the database is resumed.

### **The db2inidb and db2rfpen commands**

- The Replay (snapshot) created using the suspended I/O feature continues to stay in a write suspended mode until it is initialized to a usable state. To initialize the Replay (snapshot), a tool called DB2INIDB can be executed to perform a crash recovery. To place the database that has been restored from an offline backup into roll-forward state, you can use DB2RFPEN command.

## Dell Compellent overview

### Dell Compellent consistency group

Dell Compellent consistency group feature allows storage administrators to take a snapshot of a DB2 UDB database atomically. When creating a snapshot of a running DB2 database using storage functionality, you must ensure that all storage volumes (LUNs) that make up your database be atomically snapped because of multiple containers and transaction log files. Remember that DB2 writes to multiple container files, so without consistency group you cannot create a usable snapshot of a running database that spans multiple containers.

Without consistency group, in order to create a usable snapshot of an online database, the database table space must be configured with all container files in one volume or with just one container. Consistency Group feature gives you the ability to create a usable snapshot of an online database with multiple container files and transaction log files spread across volumes for performance.

Also, you can create a re-startable copy of a DB2 database with consistency group without having to suspend IO to database. This scenario is similar to having a power outage on the database server. At restart, DB2 performs crash recovery, rolling forward any changes that did not make it to the data files and rolling back changes that had not committed. However, roll-forward recovery using archive logs to a point-in-time after the re-startable copy is created is NOT supported.

### Dell Compellent data instant replay

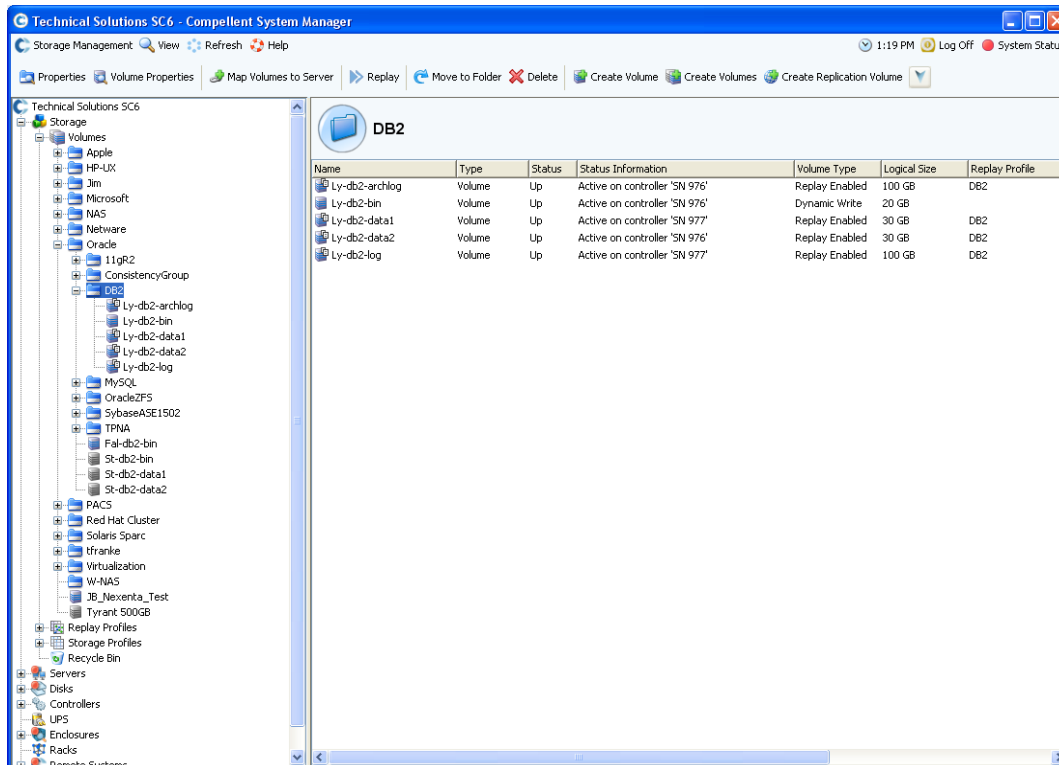
Dell Compellent storage delivers continuous data protection using space-efficient snapshots called Replays. With data instant replay, once an initial snapshot of a volume is taken, only incremental changes in data need to be captured. This not only saves disk space, but speeds local recovery of lost or deleted files. An unlimited number of Replays can be scheduled for near-instant recovery to virtually any point in time.

## Create consistency group profile

### Creating volumes

Creating volumes for your DB2 database is simple. Once the volumes are created, you map the volumes to your server and configure the volumes at the operating system level.

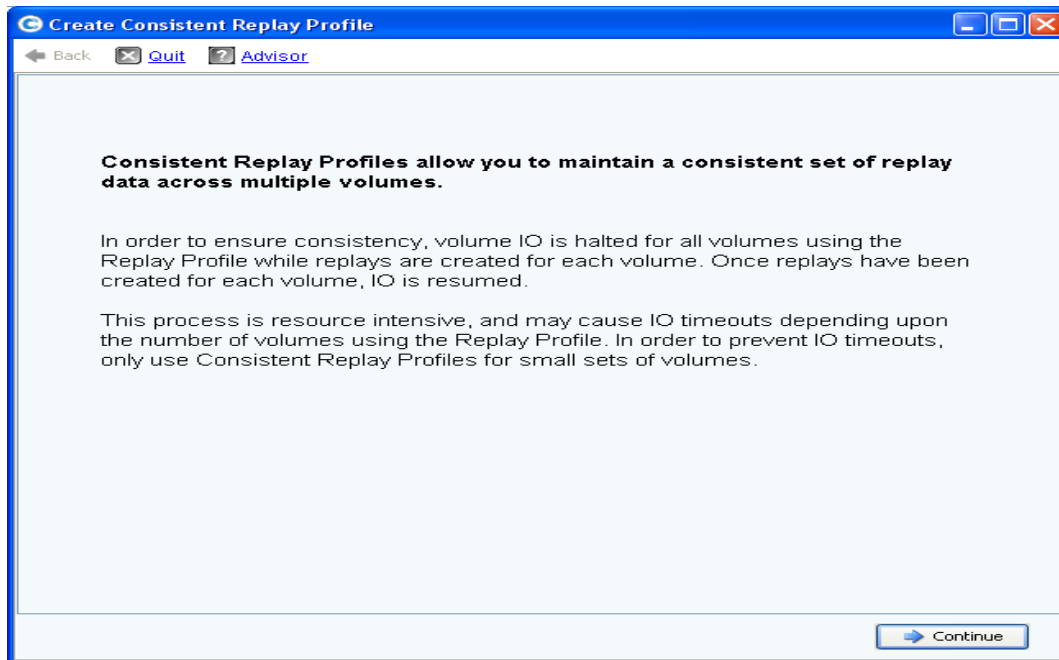
Example: I have created four volumes for my database.



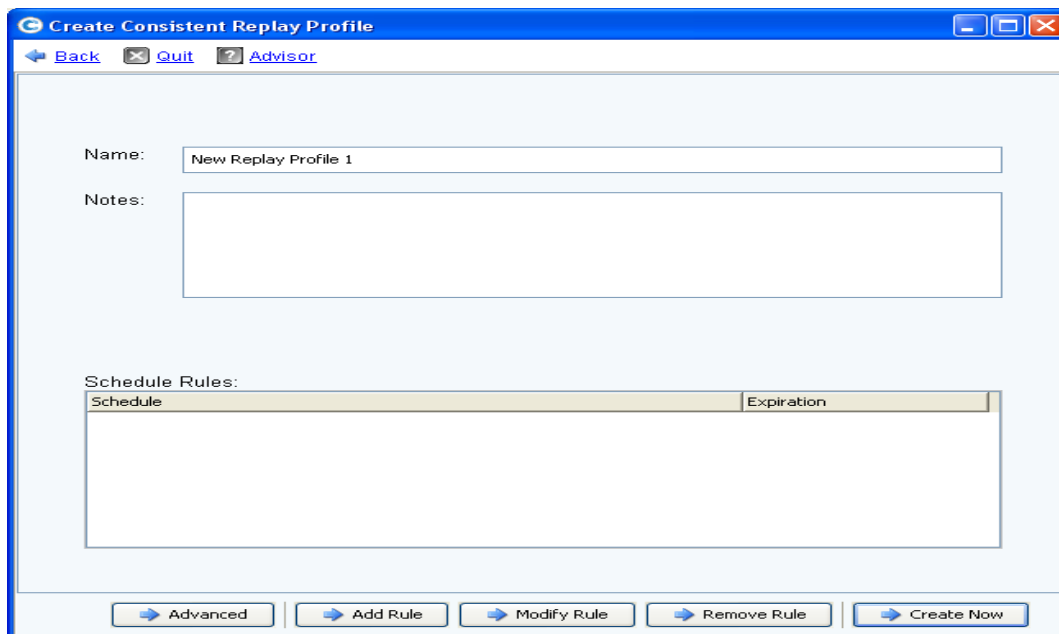
## Creating consistency group replay profile

After creating the volumes for your database, you can create a consistency group replay profile and apply the profile to the volumes. The below steps show how to create this profile and apply it to the volumes:

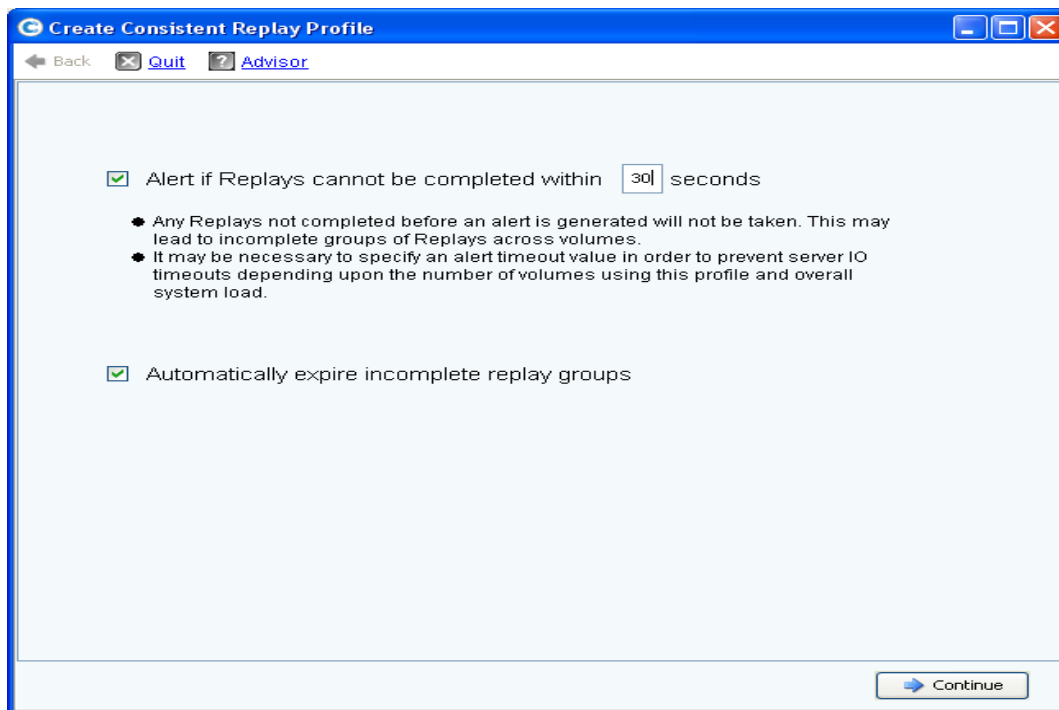
1. Right click on Replay Profiles and select Create Consistent Replay Profile. Click "Continue" button.



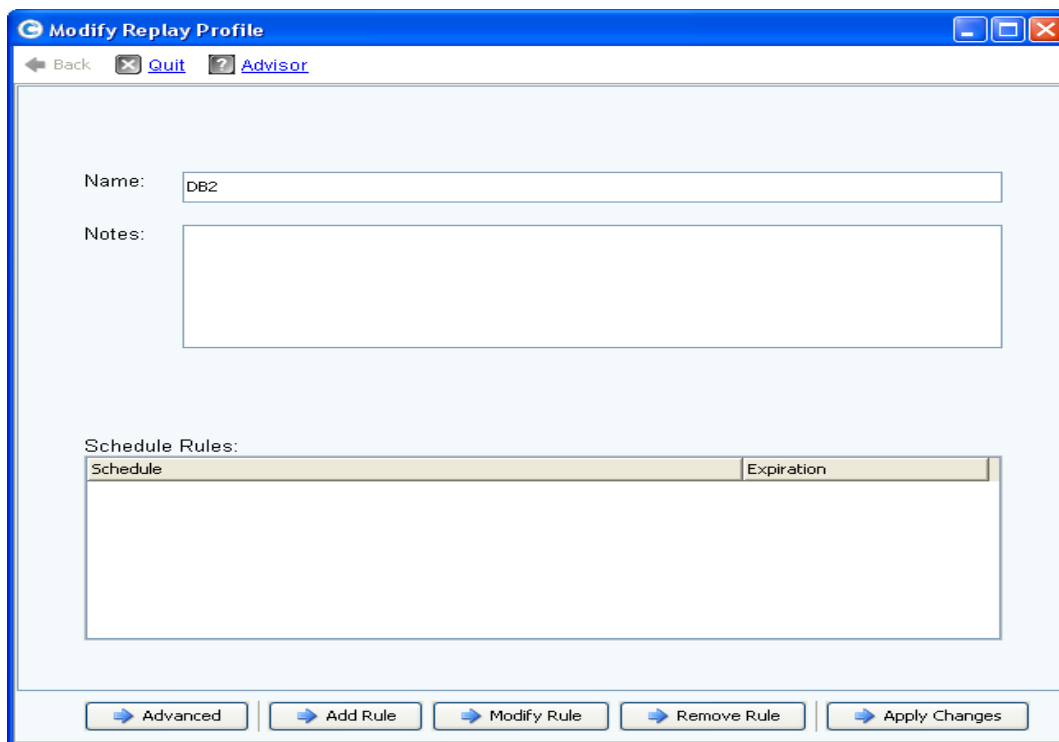
2. Click the "Advanced" button



3. Select both check boxes and enter 30 seconds and click "Continue" button.



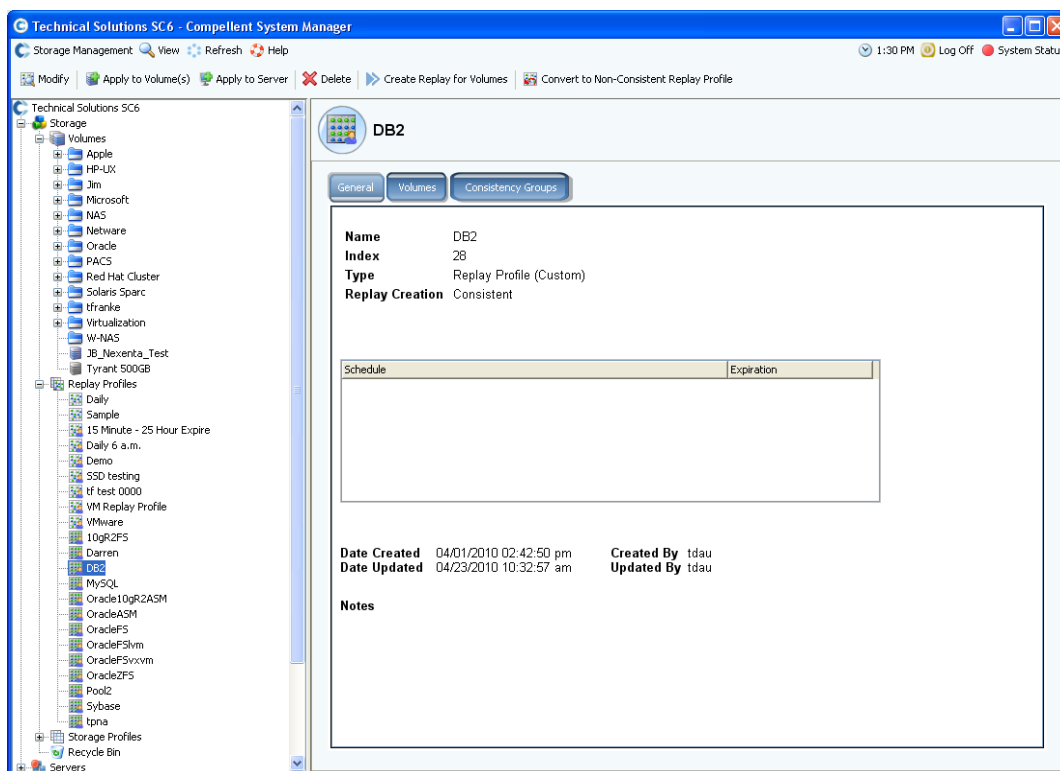
4. Enter a name for this replay profile. The name of this replay profile will be used when creating replays of the database volumes. Click "Create Now" button.



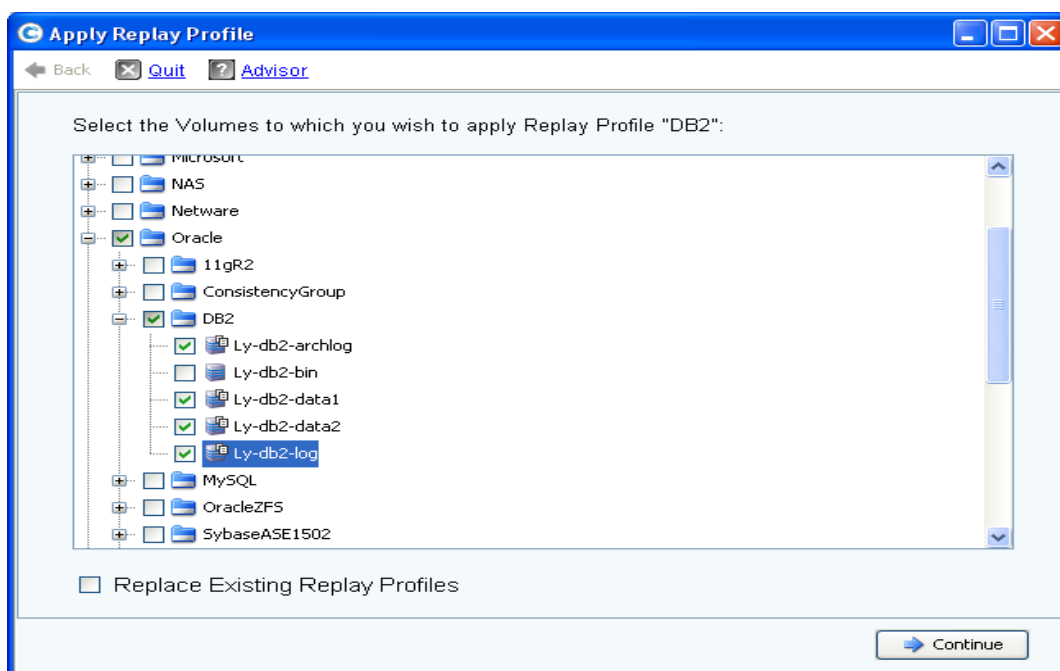


# DELL Compellent Storage Center DB2 UDB backup and recovery with CML data instant replay Best Practices

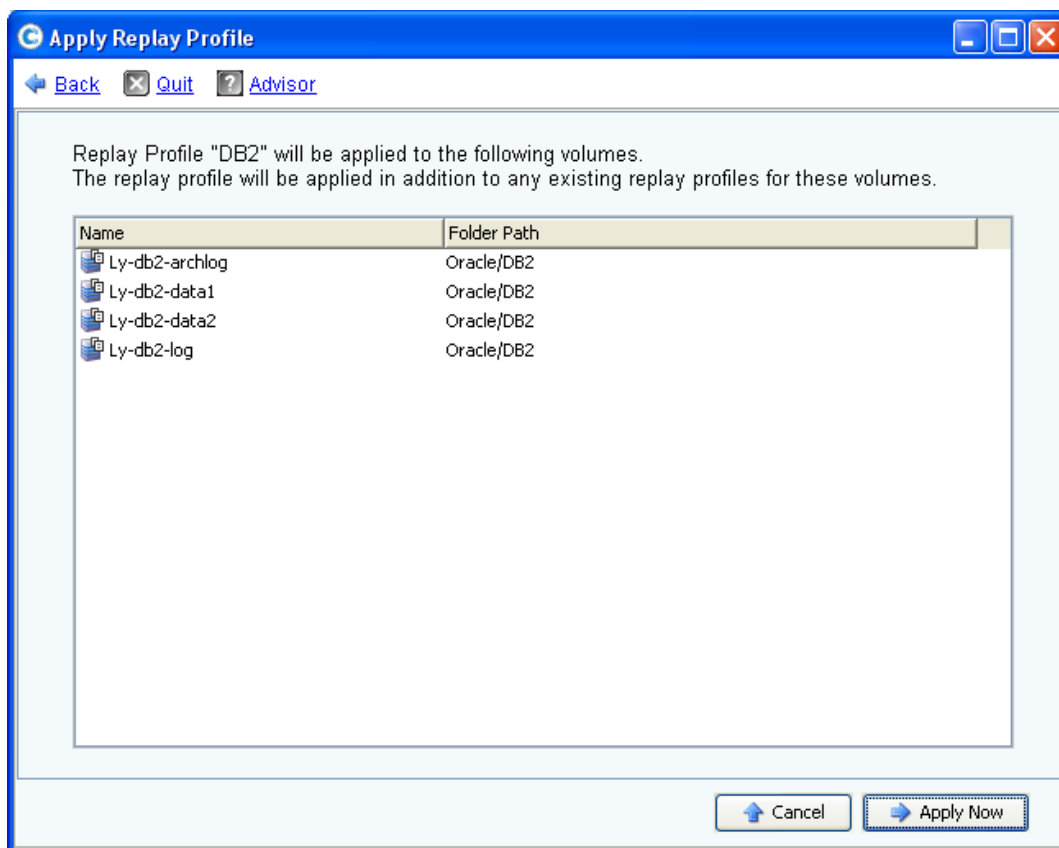
5. DB2 replay profile has been created as shown. Right click on the DB2 replay profile and select "Apply to Volume(s)".



6. Expand the volume folder and select all volumes that make up the database and click "Continue".



7. A confirmation window appears to allow you to verify the information. Click on "Apply Now" button to finish the process.



## Create DB2 database backups

### Create an offline backup

The database is considered offline when all connections to the database have been terminated and the database is no longer active. The following steps must be executed:

1. [db2inst1@production ~]\$ **db2 force applications all**
2. Take a replay (snapshot) of all volumes containing your database

### Create an online backup

The database is considered online when one or more connections have been established to it. To take an online backup, the following steps must be executed:

1. [db2inst1@production ~]\$ **db2 connect to proddb user db2inst1 using db2inst1**
2. [db2inst1@production ~]\$ **db2 set write suspend for database**
3. **sync**
4. Take a replay (snapshot) of all volumes containing your database
5. [db2inst1@production ~]\$ **db2 set write resume for database**

## Restore DB2 databases

### Restore a non-recoverable database from an offline backup

This type of recovery is called version recovery. The following steps must be executed:

1. [db2inst1@production ~]\$ **db2 force applications all**
2. [db2inst1@production ~]\$ **db2stop**
3. [db2inst1@production ~]\$ **db2\_kill**
4. [db2inst1@production ~]\$ **ipclean db2inst1**
5. Mount replays on new mountpoint
6. Copy all database files from the new mountpoints to their original locations
7. Restart your database

**Tip:** If you don't want to copy the files from the new mountpoints to their original locations, you can un-map the original volumes and then map the replays and mount on the same mountpoints and restart your database.

### Restore a non-recoverable database from an online backup

This type of recovery is call version recovery. The following steps must be executed:

1. [db2inst1@production ~]\$ **db2 force applications all**
2. [db2inst1@production ~]\$ **db2stop**
3. [db2inst1@production ~]\$ **db2\_kill**
4. [db2inst1@production ~]\$ **ipclean db2inst1**
5. Mount replays on new mountpoint
6. Copy all database files from the new mountpoints to their original locations
7. [db2inst1@production ~]\$ **db2 restart db proddb write resume**

**Tip:** If you don't want to copy the files from the new mountpoints to their original locations, you can un-map the original volumes and then map the replays and mount on the same mountpoints and restart your database.

## Restore a recoverable database from an offline backup

To recover the database to its original state *without rolling forward*, the following steps must be executed:

1. [db2inst1@production ~]\$ **db2 force applications all**
2. [db2inst1@production ~]\$ **db2stop**
3. [db2inst1@production ~]\$ **db2\_kill**
4. [db2inst1@production ~]\$ **ipclean db2inst1**
5. Mount replays on new mountpoint
6. Copy all database files from the new mountpoints to their original locations
7. Restart your database

**Tip:** If you don't want to copy the files from the new mountpoints to their original locations, you can un-map the original volumes and then map the replays and mount on the same mountpoints and restart your database.

To recover the database to its original state *with rolling forward*, the following steps must be executed:

1. [db2inst1@production ~]\$ **db2 force applications all**
2. [db2inst1@production ~]\$ **db2stop**
3. [db2inst1@production ~]\$ **db2\_kill**
4. [db2inst1@production ~]\$ **ipclean db2inst1**
5. Mount replays on new mountpoint
6. Copy all database files from the new mountpoints to their original locations
7. [db2inst1@production ~]\$ **db2rfpen on proddb**
8. [db2inst1@production ~]\$ **db2 rollforward database proddb to end of logs and complete**

**Tip:** If you don't want to copy the files from the new mountpoints to their original locations, you can un-map the original volumes and then map the replays and mount on the same mountpoints and restart your database.

## Restore a recoverable database from an online backup

To recover the database to its original state *with rolling forward*, the following steps must be executed:

1. [db2inst1@production ~]\$ **db2 force applications all**
2. [db2inst1@production ~]\$ **db2stop**
3. [db2inst1@production ~]\$ **db2\_kill**
4. [db2inst1@production ~]\$ **ipclean db2inst1**
5. Mount replays on new mountpoint
6. Copy all database files from the new mountpoints to their original locations
7. [db2inst1@production ~]\$ **db2inidb proddb as snapshot**
8. [db2inst1@production ~]\$ **db2 rollforward database proddb to end of logs and complete**

**Tip:** If you don't want to copy the files from the new mountpoints to their original locations, you can un-map the original volumes and then map the replays and mount on the same mountpoints and restart your database.

## Conclusion

With Dell Compellent Data Instant Replay technology, the Dell Compellent Storage Center offers a compelling advantage for database administrators in terms of backup and recovery. By using Dell Compellent data instant replay, database administrators can perform backups online or offline without adding additional disk space compare to other traditional snapshot technologies. Database administrators can also keep the backup replays on line for quick recoveries. Additionally, backup and recovery performance is dramatically improved compared to other conventional methods.