HITACHI
Inspire the Next

Hitachi Dynamic Replicator - Scout Solution for MySQL (Windows)

Hitachi Data Systems

MK-98DF8220-00

Contents

Conventions

Keywords, command buttons and other such fields are enclosed in " " while being bold (for example, to denote 

"**Next**" is used).

Inputs for commands and Variables are shown in *Italics*
File names and paths are shown in **bold**
Commands are shown in **`Courier new font`**
Optional keywords and arguments are enclosed within [ ].

 **Notes:**

Contain suggestions or tips.

 **Caution:**

Contains critical information

Purpose of the document

This document attempts to uncover MySQL application support for Windows-based platforms.

Assumptions

- This document is specific to Non-Clustered Windows operating systems.
- VX agent installed on production (source) and DR Server (target)
- FX agents are installed on production (source) and DR servers (target)
- CX server is up and running
- All agents are pointed to the same CX server with valid licenses

1 Introduction to the Solution

This document explains possible solutions for Windows based MySQL server by using Hitachi Dynamic Replicator - Scout. The first approach is to replace the production server with a backup MySQL server in case of an outage. This process can be called as a failover. Failover can be performed under a variety of conditions such as logical corruption or a hardware error or any other situation where the production server is down. Failing over to the backup server is a transparent process to end users. Given below is a picture of a failover process.

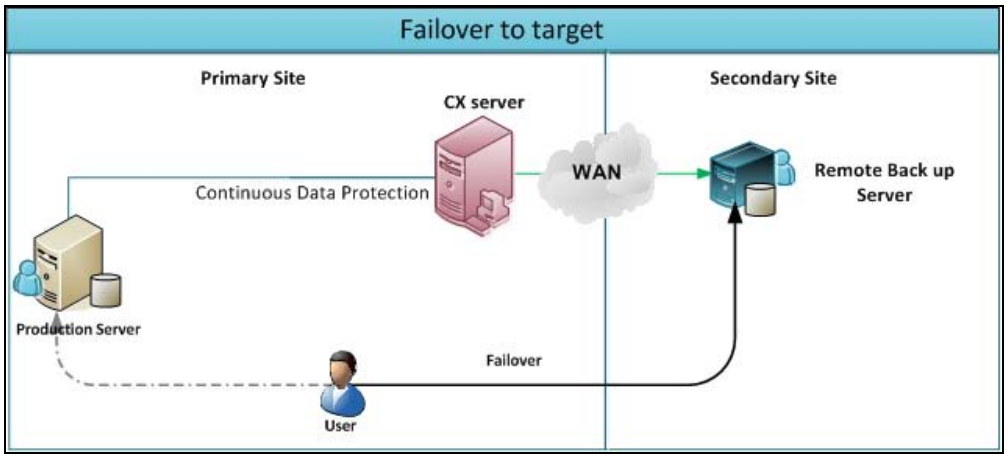


Figure 1

A failback is initiated when the production server back and ready to resume its activities. Depending on the time of outage and data changes, administrators can choose to reverse replicate i.e. update the production server with all the data changes (occurred during its outage) from the backup server and then replace the backup server (acting in place of the production server) with the production server.

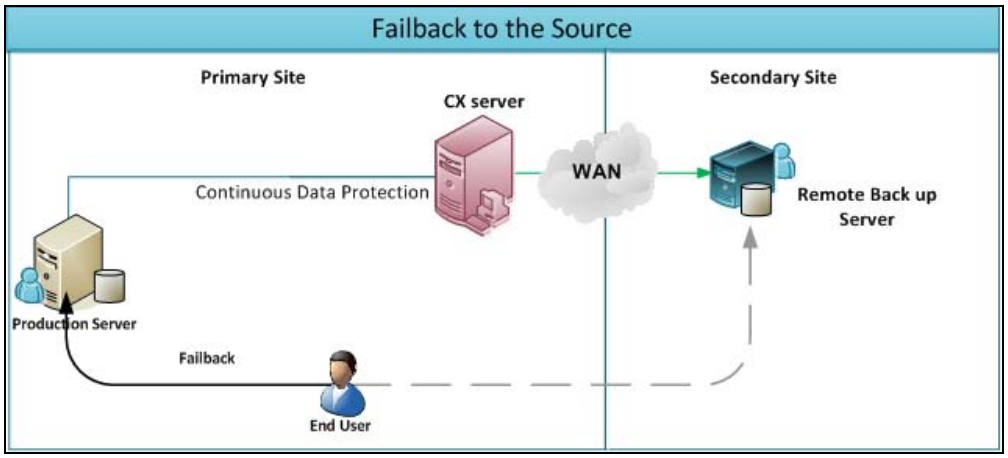


Figure 2

2 How this Solution Works

This solution is divided into three steps; Prepare MySQL server, Protect MySQL and Recover it.

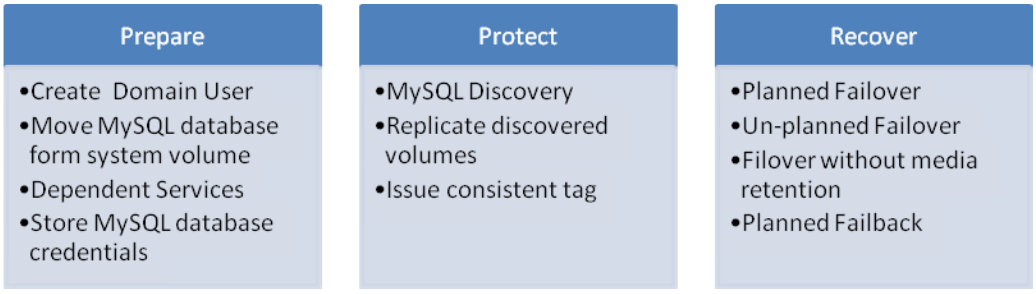


Figure 3

Each of these steps has sequence of steps to be followed to achieve the goal.

Prepare: There are three steps for preparing MySQL Server to be recovered i.e. Create Domain User, Move MySQL Database from System Database, and Dependent Services. Refer Section Prepare on Page 9

Protect: MySQl server is protected in three steps, i.e. Discover MySQL, Replicate the Discovered Volume, and Issuing Consistency Tag. Refer Section Protect on Page 22

Recover: MySQL Server can be recovered through Planned Failover, Un-Planned Failover or Failback. Refer Section Recover on Page 31

3 Solution Workflow

Workflow for Protecting MySQL

MySQL is protected in three steps, discovery, Protect and Consistency.

- MySQL Discovery: This can be executed either from the console or through an FX job
- Protect MySQL: Replicate discovered volumes through VX
- Consistency: Issue MySQL consistency markers through FX job.

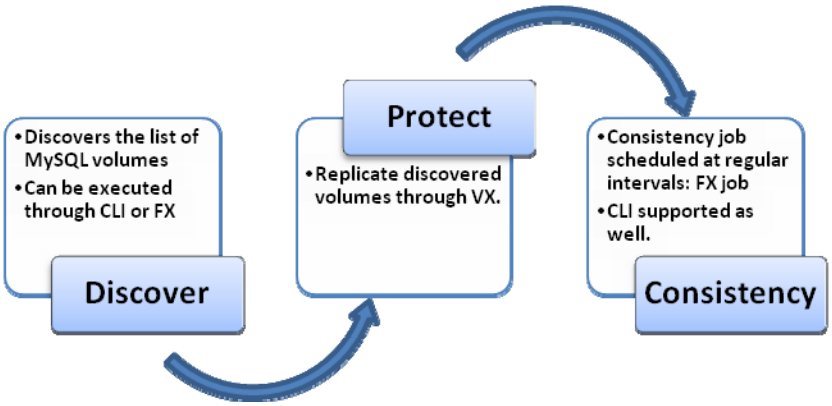


Figure 4:

MySQL recovery is divided into two sections, either MySQL failover or through mounting the database virtual snapshot created on the target host.

Workflow of Planned failover

A planned failover is preferred to test the setup or to perform DR drills. While performing a planned failover a consistency tag is issued after the application services are stopped. This is to ensure zero data loss. Planned failover can be performed through the CX UI and through CLI.

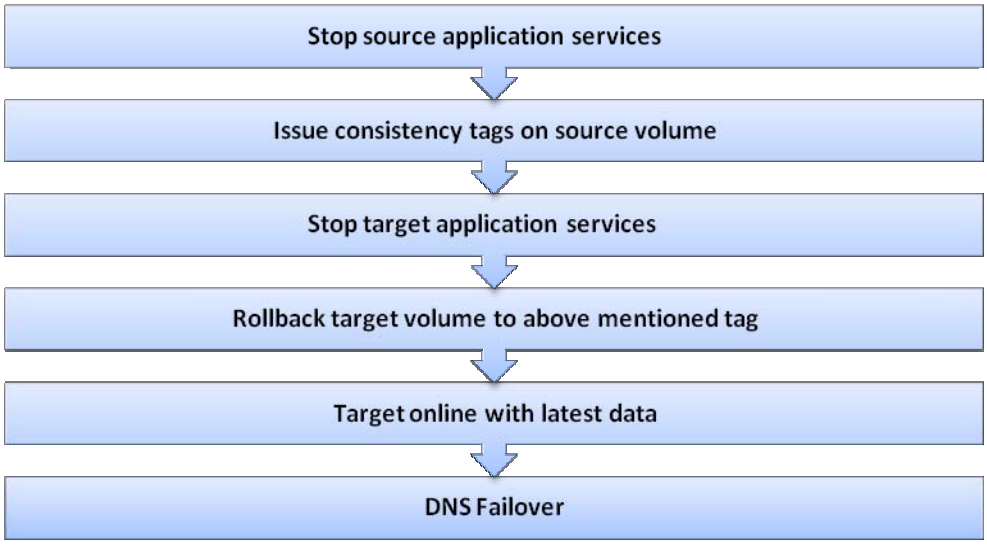


Figure 5

Workflow of Unplanned Failover

Unplanned failover is performed when there is a logical corruption on the source host or if the source host is completely down. This can be performed through the CX UI if the source is available. If the source host is down then unplanned failover can be performed through the target host's console.

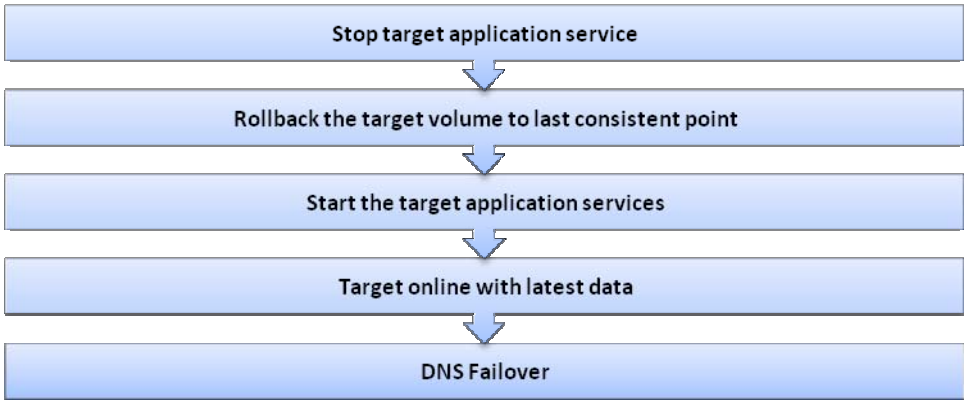


Figure 6:

MySQL Failback

Once the production server is back online and ready, a failback is performed which is similar to a planned failback. We set a reverse replication i.e. from DR server to production server (DR Server as Source and Production Server as Target). This is done to ensure that the production will be updated with all the data changes during its outage.

Failback can be performed either through CX UI or CLI.

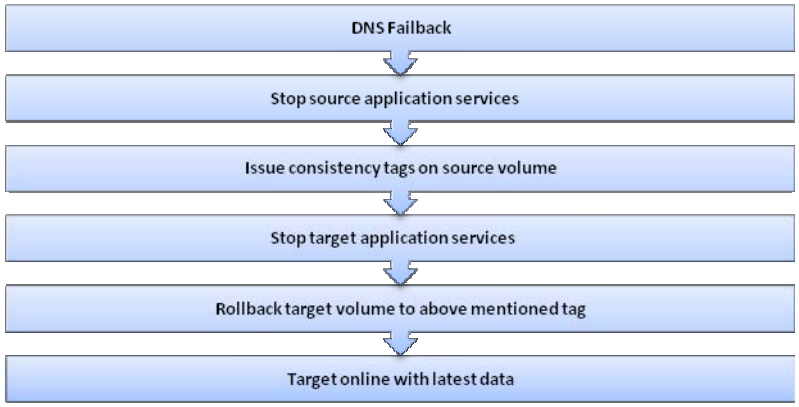


Figure 7

Caution:

Ensure that FX agent service should be running under domain administrator privileges as well as local administrator privileges or else the application.exe will crash.
Before starting Discovery ensure that MYSQL browser service is up and running.

4 Prepare

4.1 Create a Domain User

The FX agent needs to be up and running with domain administrator privileges. To create a domain administrator follow the below steps.

Step 1. Log on to the domain controller as an administrator where production server and DR Server are part of the domain.

Step 2. Navigate to **"Start →Programs →Administrative Tools →Active Directory Users and Computers"**.

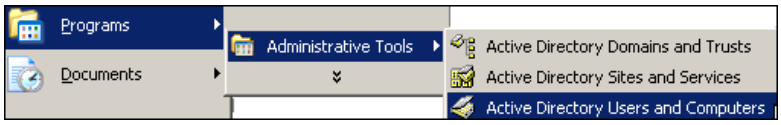


Figure 8:

Step 3. You should see the **"Active Directory Users and Computers"** screen. Under "**Active Directory Users and Computers".** Expand the domain and select **"Domain Users"** to display the list of users on the right hand side. Right click and select **"New →User"**.

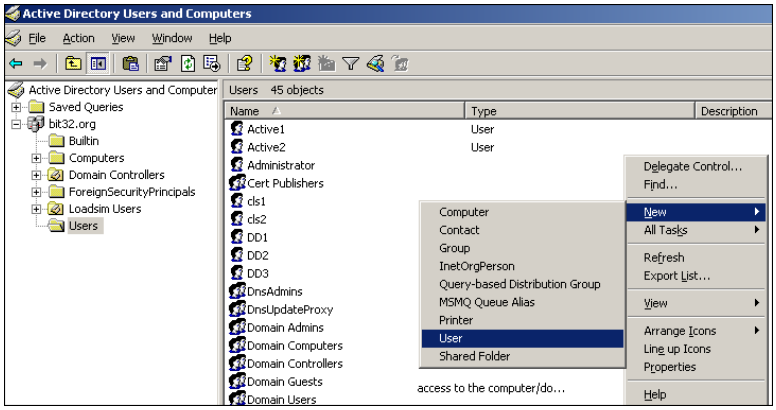


Figure 9:

Step 4. You should be able to see **"New Object-User"** UI. Enter the **"Name"** and **"User Logon Name**" and click "**Next"**.

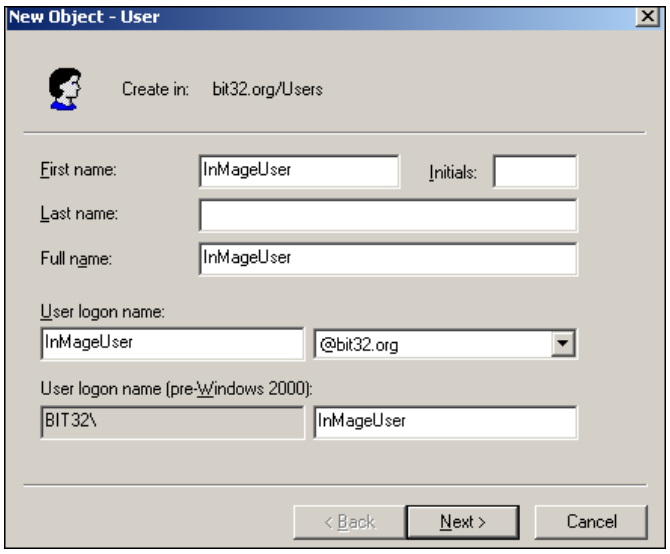


Figure 10:

Step 5. You will need to enter the **"Password"** and reenter to confirm it. Be sure to select the "**Password Never Expires"** option. A message box appears "The user will not required to change the password at next logon", click "**OK"** then click **"Next"**.

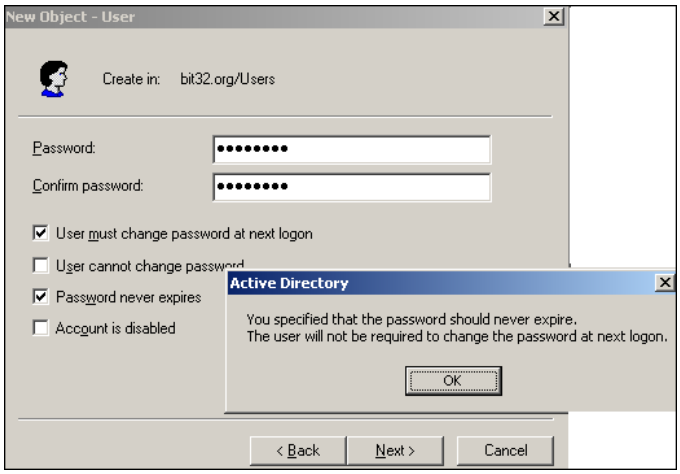


Figure 11:

Step 6. The **"New Object-User"** appears, click on "**Finish**" to continue.

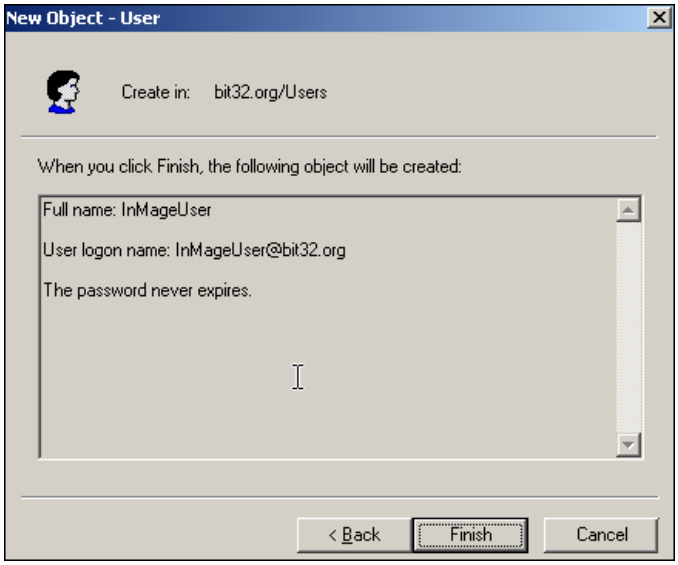


Figure 12:

Step 7. Logon to **"Production Server"** with the domain administrator. Open Computer management console, navigate to **"Local Users and Groups →Groups →Administrators**".

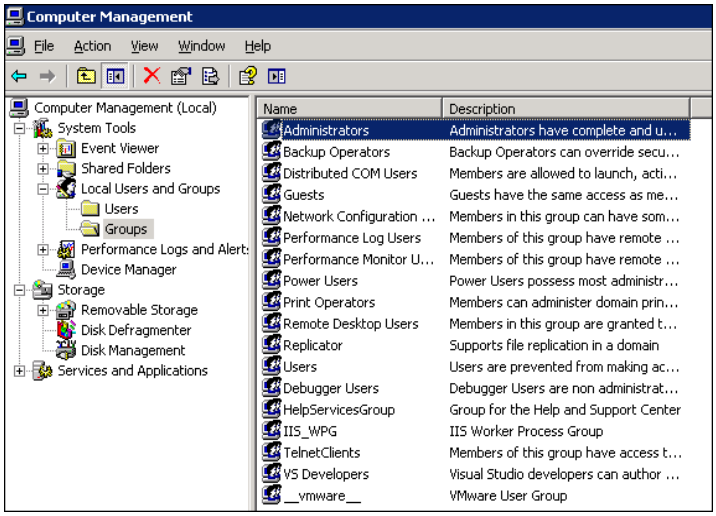


Figure 13:

Step 8. You should be able to see administrator's properties UI. Provide the user name to whom you want to assign administrator privileges and Click "**Add"**.

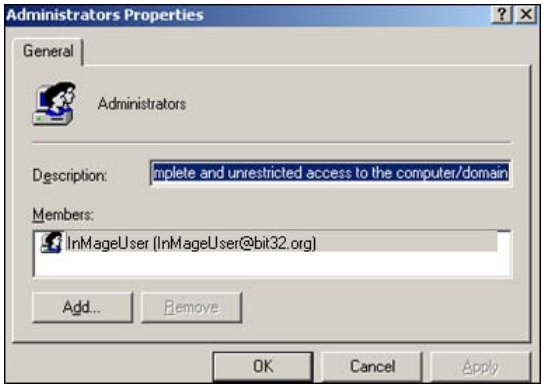


Figure 14:

Step 9. Then the **"Select Users, Computers, or Groups"** UI appears, Select the domain user name (InMageUser) created in the previous steps and check if this username is available by clicking on the **"Check Names"**. Click "**OK**".

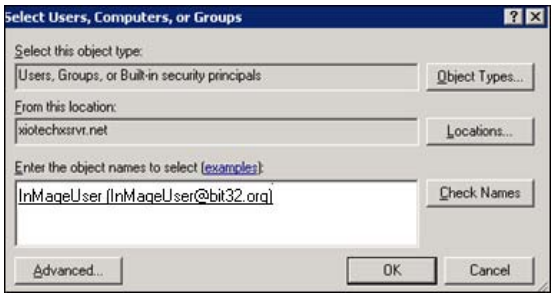


Figure 15:

Step 10. Navigate to "**Start→Control Settings→Administrative Tools**". You should be able to see **"Administrator's Properties"** UI with the new domain user (InMageUser). Click on **"Apply"** and then on **"OK"**.

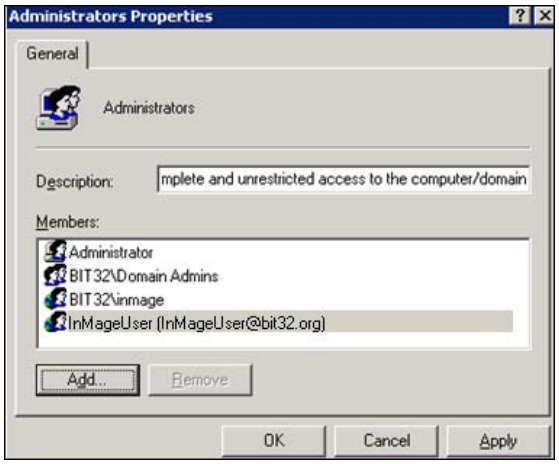


Figure 16:

Step 11. Navigate to **"Start→Program Files→Administrative Tools→Local Security Policy".**

Figure 17:

Step 12. You should see the **"Local Security Settings"** UI. Select **"User Rights Assignment"** and select **"Log on as a Service"**.

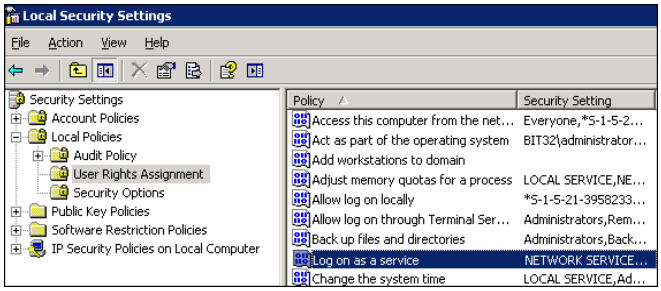


Figure 18:

Step 13. You should see the **"Log on as a Service Properties"** UI. Click "**Add User or Group"**, and click **"OK"**.

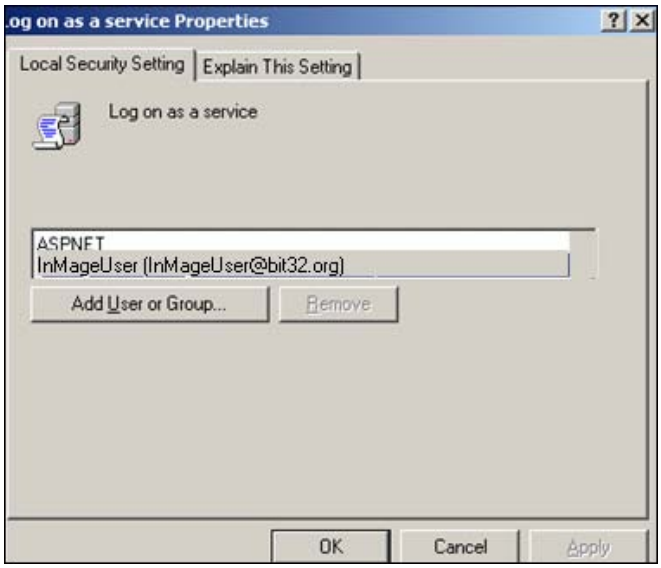


Figure 19:

Step 14. In the **"Select Users, Computers, or Groups"** UI select the domain user name created in the previous steps and check if this username is available or not by clicking on the **"Check Names"**. Click "**OK**".

Figure 20:

Step 15. You will be returned back to the previous UI. Click on **"Apply"**, and then on **"OK"**.

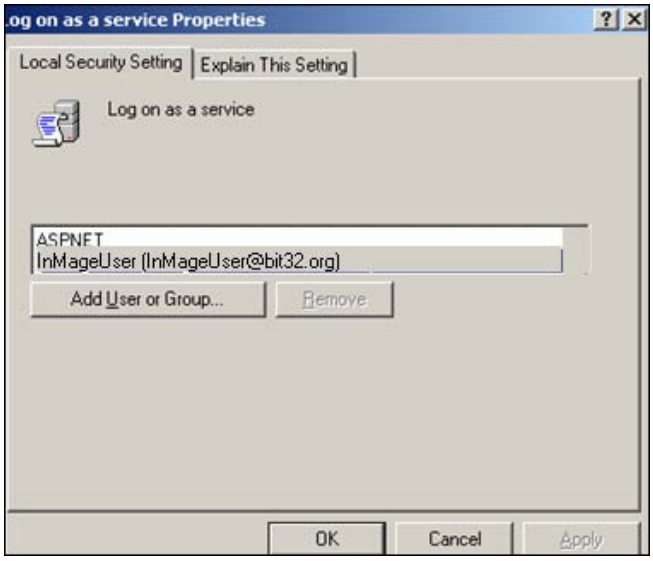


Figure 21:

Step 16. Log on to **"Production Server"** with the domain user created above. Navigate to "**Start->Run->Services.msc"**, select the **"FX agent service"**, right click, and click on the **"Properties"**.

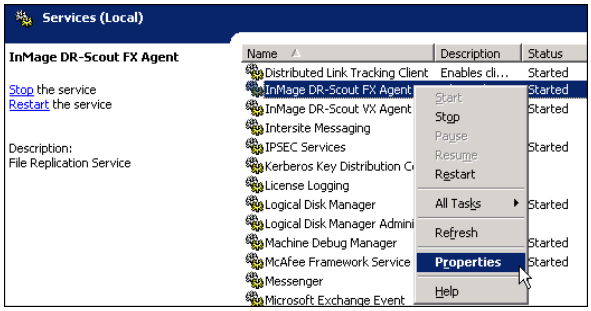


Figure 22:

Step 17. In the **"FX agent Properties"** UI enter the **"Domain Username"** and password details. Then, click **"Apply"** and then **"OK".**

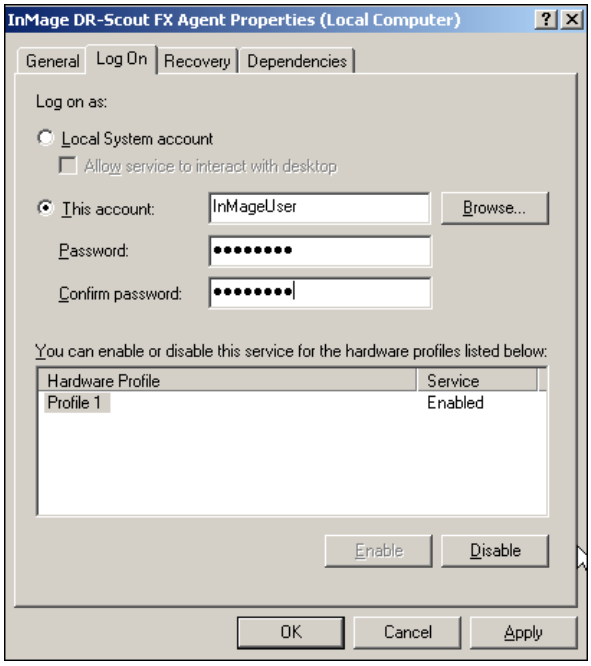


Figure 23:

Step 18. Restart the **"FX Agent Service"**. And repeat the above steps for "**DR-Server".**

Notes:

If FX Services are not running under administrator privileges as well as local administrator privileges, then FX jobs will not be successful. To add and run it with administrator privileges refer to Section Create a Domain User on Page 9 of this current document.

4.2 Move MySQL Databases from System Drive

To move MySQL System Database from boot or system volumes, follow the below steps:

4.2.1 Identify Database path

To get MySQL Database path, follow the following steps:

Step 1. Navigate to **"Start→All Programs→MySQL→MySQL Server 5.1→MySQL Command Line Client"** on the production server.

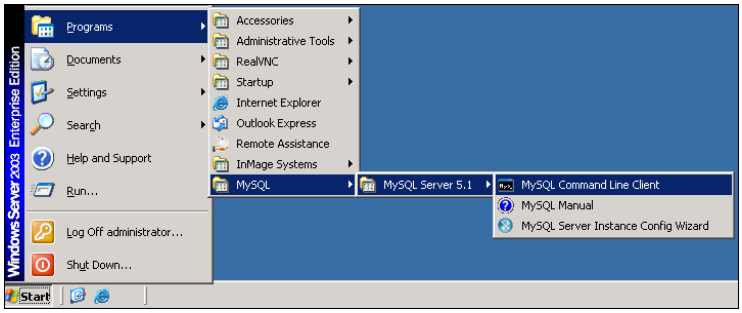


Figure 24

Step 2. You should be able to see **"MySQL command Prompt"**. Enter the password to start "MySQL".

Figure 25

Step 3. Execute the command **"SHOW VARIABLES"** in the command prompt. It displays a list with two columns i.e. the first column is "variable_name"and second column "value". In that we will get the path for the default database location for MySQL.

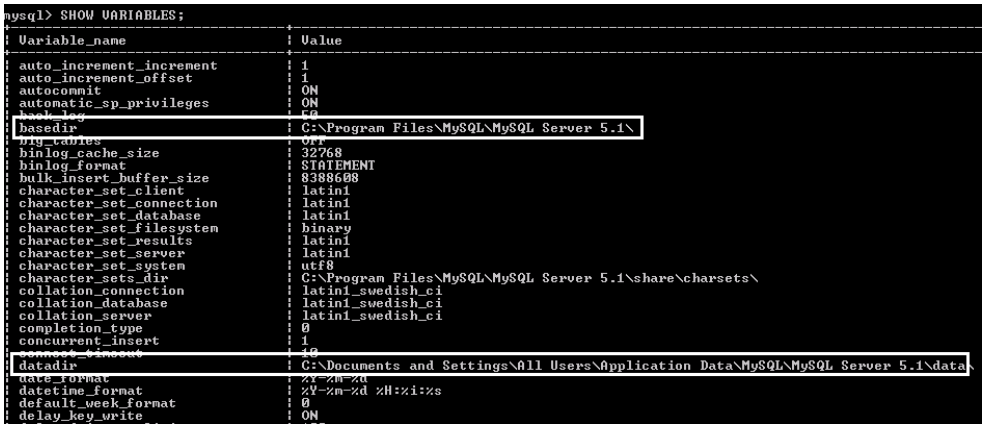


Figure 26

Step 4. Check the variable_name column for "Datadir" & the Value for it. The Value part will display the Datadir location where the MySQL database is stored.

Step 5. Also check the variable_name column for "Basedir" & the Value for it. The Value part will display the Basedir location where the MySQL related files are present. From this location we will get the **"my.ini"** file

4.2.2 Shutdown MySQL Service

To shutdown MySQL service on local machine, follow the following steps:

Step 1. Navigate to **"Start→Run→Services.msc"** on the production server.

Step 2. You should be able to see **"Services" UI**. Select **"MySQL"** under "**Services (Local)**".

Step 3. The status of "**MySQL"** will appear on UI. If the "MySQL" service status is "start" then "stop" the service by clicking on "stop".

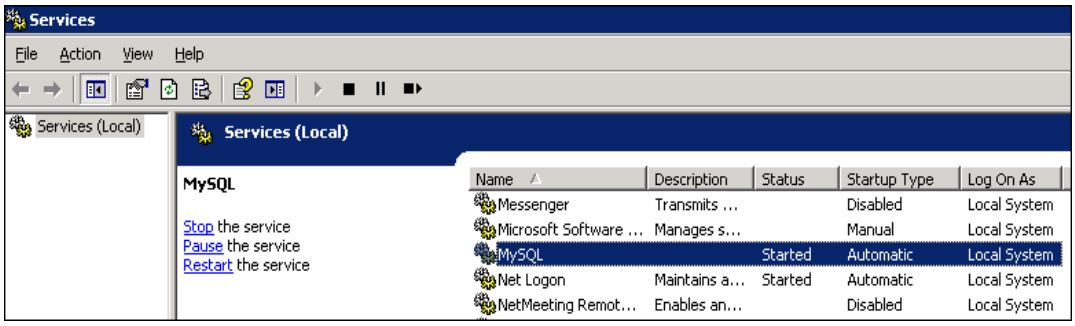


Figure 27

4.2.3 Move MySQL Database

To change the MySQL database location, follow the following steps:

Step 1. Go to the "BaseDir" location and open the file **"my.ini"** to edit it.

In the **"my.ini"** file check for "Datadir" variable & change the database path location in datadir (give the location of the destination folder where the database files are copied) & save the file.

Step 2. Choose a destination folder where you intend to move the MySQL database.

Step 3. Copy all the files from the "Datadir" location to the destination folder.

4.2.4 Start MySQL Service

To change the MYSQL database location, follow the following steps:

Step 1. Navigate to **"Start→Run→ Services.msc"** on the production server.

Step 2. You should be able to see **"Services"** UI. Select **"MYSQL"** under **"Services (Local)"**.

The status of "**MySQL"** should be stopped**.** Now start the service by clicking on **"start".**

Check the datadir path in the mysql command prompt by typing the command "SHOW VARIABLES". The Datadir value should point to your new database location.

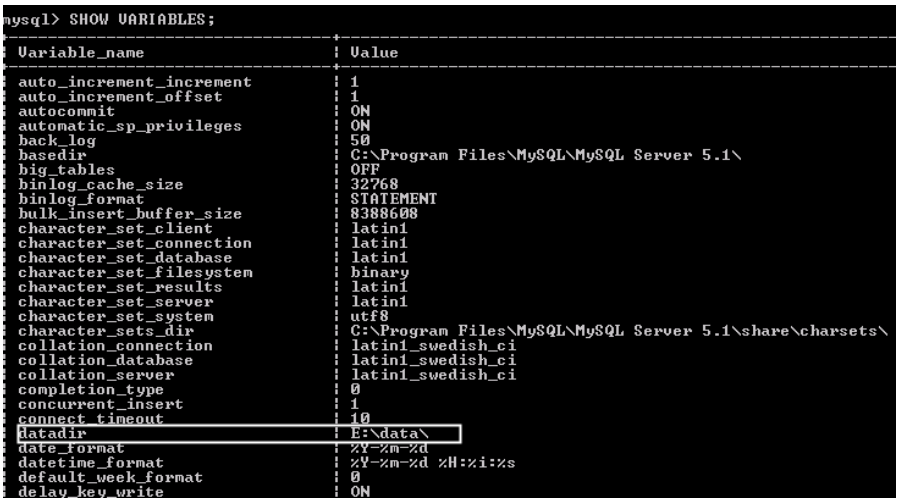


Figure 28

Notes:

If MySQL DB directory is on your system volume then you will needs to move the database to a different volume. Refer to Section Move MySQL from System Directory on Page 16 of this current document.

4.3 Dependent Services

For MySQL Server, Hitachi Dynamic Replicator - Scout does not start or stop all the dependent services automatically during failover or failback operation. By default, it starts and stops SV Agent services. To start and stop other dependent services during failover and failback operation, use Failoverservices.conf file. The "failoverservices.conf" file is located under the "consistency" folder (under VX installation path).

To stop and start the dependent services, create a section [MySQL51] for MySQL 5.1 in failoverServices.conf file. In the section, add two keys START and STOP. Write dependent service name that should be started in START key and write dependent service that should be stopped in STOP key. The service name should NOT be placed in double quotes even if there is any space in the service name. Hitachi Dynamic Replicator - Scout stops the listed dependent service before stopping default MySQL services and will start the listed dependent service after starting default MySQL services. This ensures that these services are stopped and started while performing failover.

Format of the section appears as below.

[MySQL51]
START=<ServiceName>
STOP=<ServiceName>

Example,
[MySQL51]
START=MySQL
STOP= MySQL

4.4 Store MySQL Credentials

You will need to store MySQL database logon credentials using the winop.exe commandline utility. These credentials are used while discovering MySQL, performing failover or even a failback.
Run the below command to store the MYSQL5.1 username and password for further processing. For this we are using "**Winop.exe".** We have to do this before starting the MySql Discovery Job. This is a onetime thing to be performed by the end user. By running the WinOp utility, we can store the MySql Server User Name and Password in an encrypted form in the registry. We have to run this in both source and target server.
Access the source host's console and navigate to the InMage agent installation folder, then issue the following command:

```
Winop.exe  SECURITY -encrypt –mysql
```

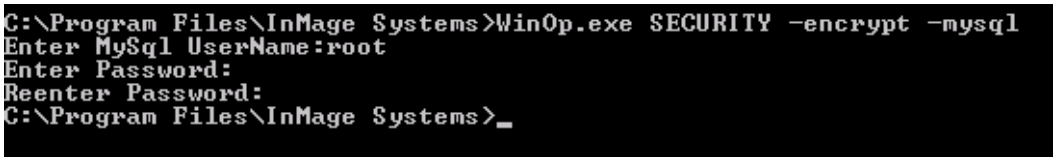


Figure 29

This solution supports only for one instance of MySql5.1, but it can support any number of databases in a single instance. In MYSQL all the databases reside only in a single volume. So here we are replicating only the discovered volume (discovers only one volume which one contains the databases). Also this solution only supports for non-cluster standalone system.

 **Notes:**

If dependent services of the MySQL Server are running and those are not listed in failoverservices.conf file, then MySQL Server service cannot be stopped.

5 Protect

5.1 Discover MySQL

Discovery can be performed either through CX UI or through the CLI.

5.1.1 Through CX UI

Follow the following steps to discover through CX UI.

Step 1. Navigate to **"File Protection"** and click **"New Job Wizard".**

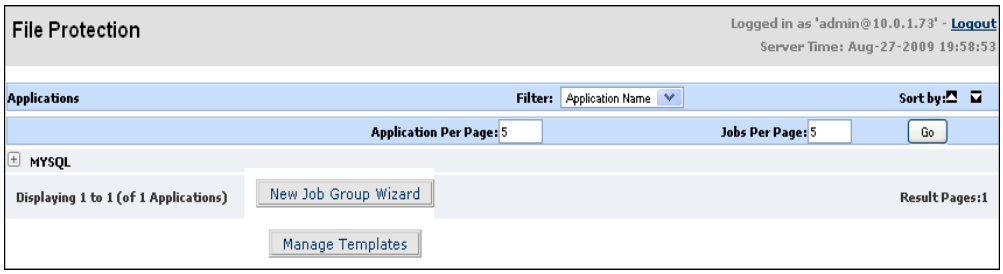


Figure 30:

Step 2. Click **"Add Job"** to create a new FX job.

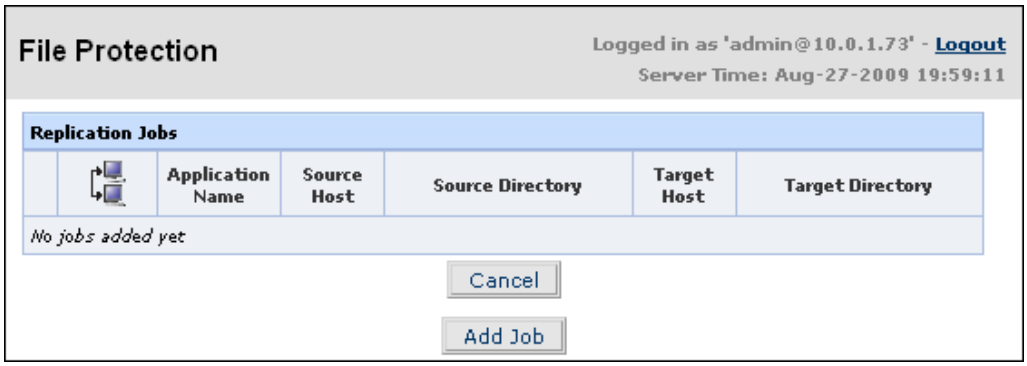


Figure 31:

Step 3. Enter the "**Application Name"** and "**Job Description".** Select the source volume and destination volume. Then, select the template as **"MYSQL(windows) Discovery"** and click on "**Next"**

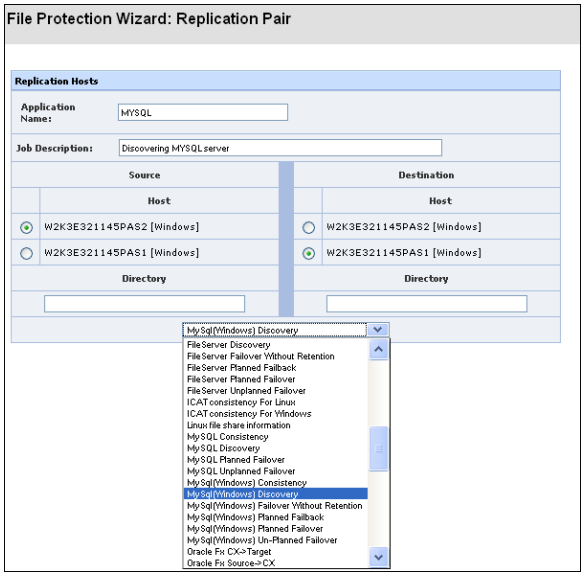


Figure 32

Step 4. Configure the required **"FX Job Options"** and click "**Submit"**.

Step 5. Schedule the above FX job as required. Select the required job from **"Replication Jobs"** list and click **"Submit".**

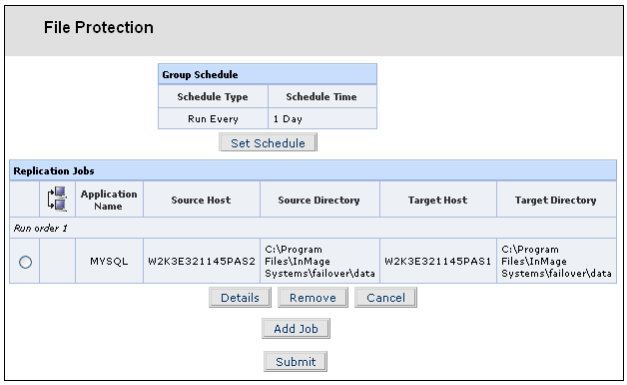
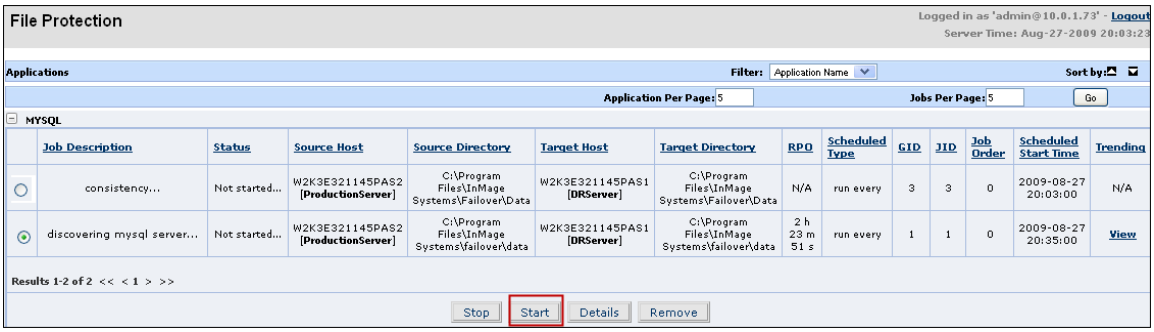


Figure 33

Step 6. Start the above scheduled FX job through **"File Protection"**. Select the above job and click the **"Start"** button.



Step 7. You can check the progress of the FX job through "**Protection Status**". Once the job is complete successfully the Status will change to "**Completed**"

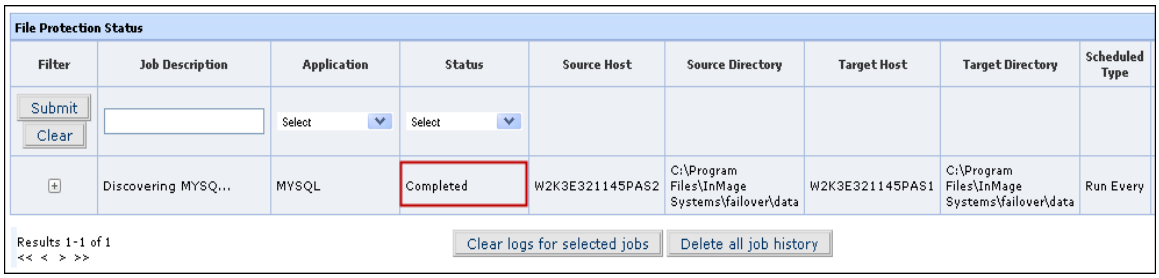


Figure 34:

Notes:

You may choose to click on "Set Schedule" to alter the job execution time.
While using MYSQL the FX template for discovery is "MYSQL(windows) Discovery"

5.1.2 Through CLI

Access the production server and navigate to the InMage agent installation folder to issue the following command:

```
Application.exe –discover –app mysql
```

Copy the "*<installation folder>***Failover\data**"* on the source host to the target host manually. For MYSQL no need to do discovery at the target side.

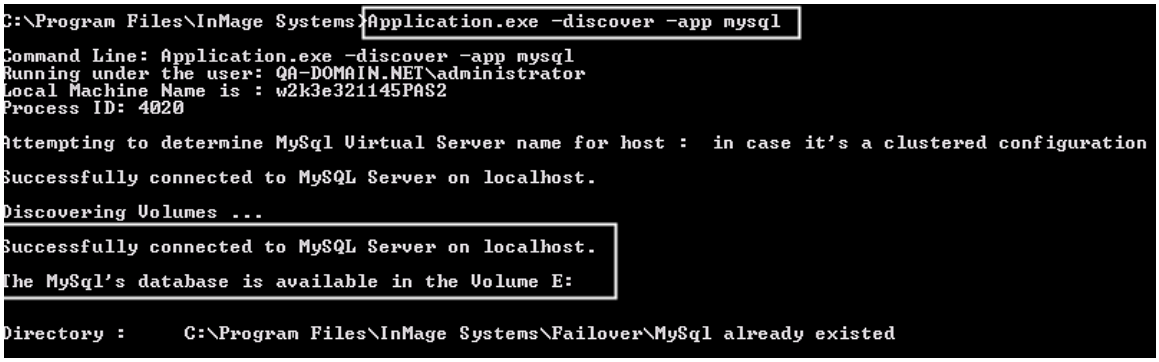


Figure 35

5.2 Set Replication

Set a replication pair form MySQL Production Server to DR Server with same drive letters. To do so follow the following steps:

Step 1. To start replication pair, navigate to **"Volume Protection→Host Drives".** Choose the volume where MySQl exists. Click **"Start Replication"** button to choose DR volume.

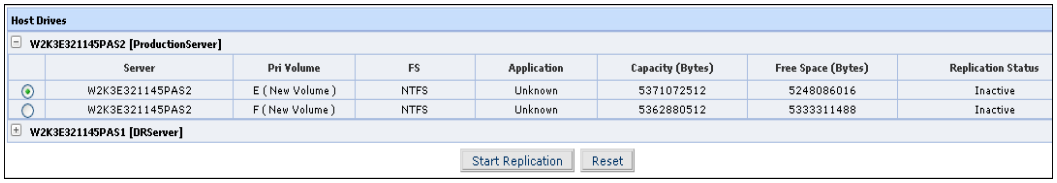


Figure 36

Step 2. Select the DR volume from the **"Select a Target Volume"** UI. The default options should remain as it is. Enable **"CDP Retention"** option. Click "Submit" button. The "Retention Options" UI appears after clicking on submit button.

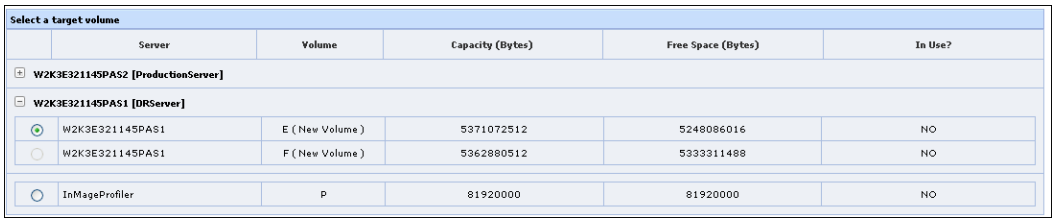


Figure 37

Step 3. Choose the retention log options and click **"Submit"** button.

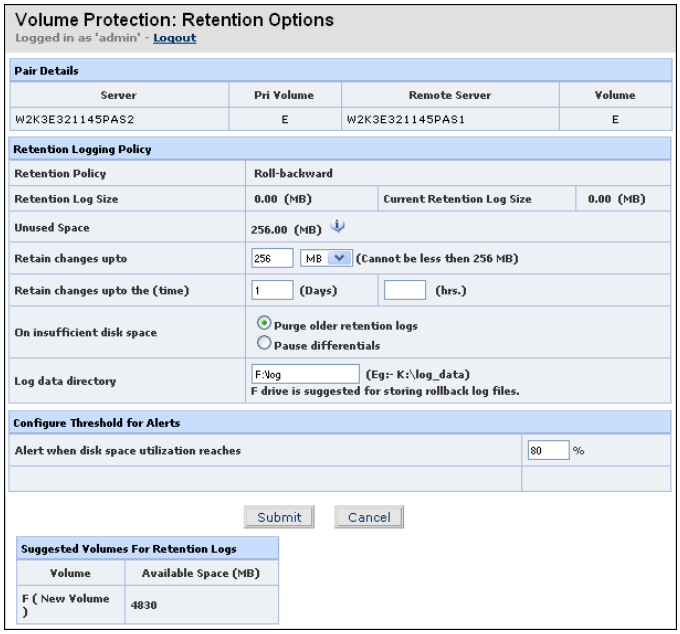


Figure 38

Step 4. Check the status of this replication pair, through **"Protection Status".** The replication pair should come to **"Differential Sync"** mode.

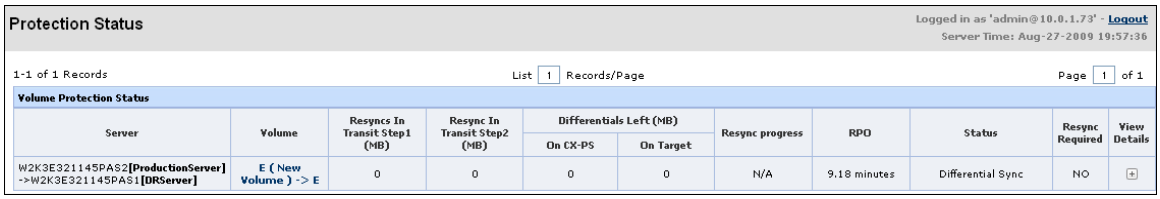


Figure 39

5.3 Issuing Consistency Tag

You can issue a consistency tag through CX UI, and CLI also.

5.3.1 Through CX UI

Step 1. Navigate to **"File Protection→New Job Group Wizard"**, to set a FX job.

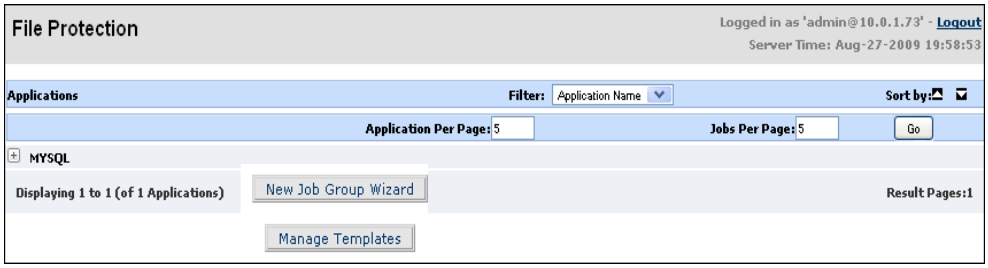


Figure 40

Step 2. Click "**Add Job".**

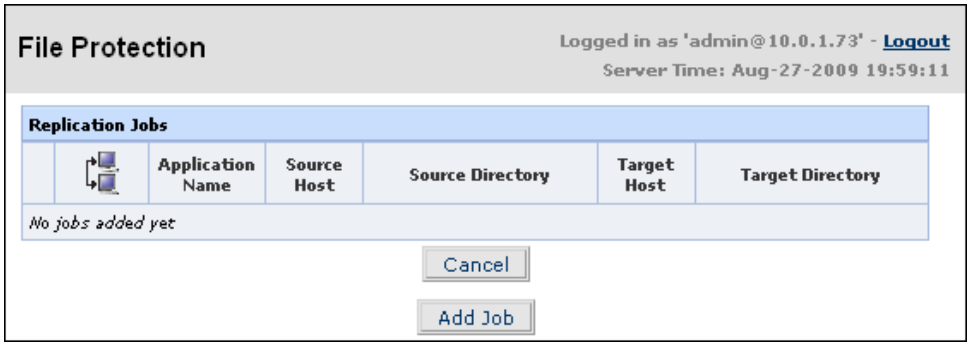


Figure 41

Step 3. Provide **"Application Name"** and **"Job Description"**. Choose Production Server volume and DR Server volume. Provide the source and target directory for installation path for "Failover\Data". Choose **"MySQL (Windows) Consistency"** template from the drop down menu. Click **"Next".**

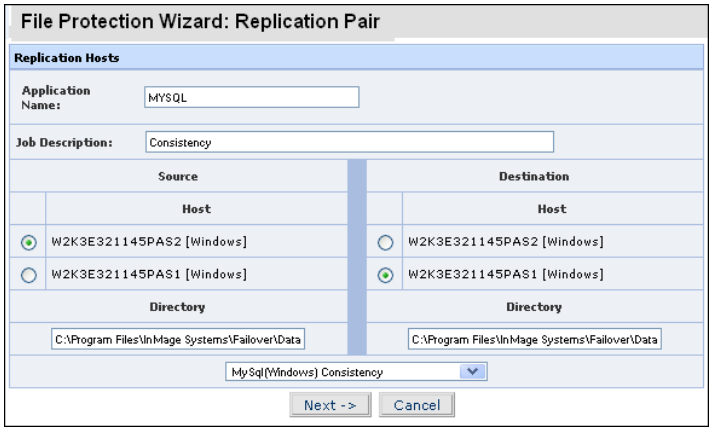


Figure 42

Step 4. Choose the required **"FX job options"**. Click **"Submit"** button.

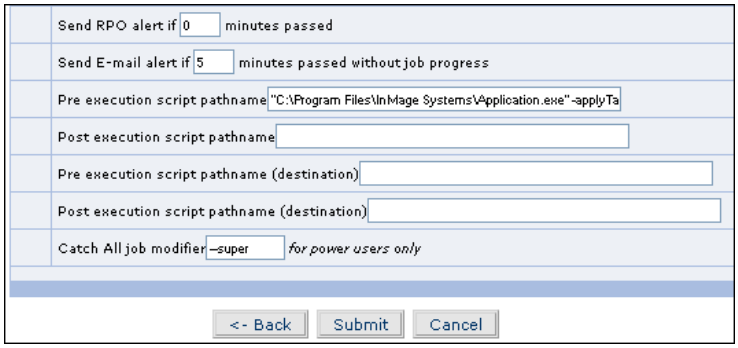


Figure 43

Step 5. Schedule the above FX Job through **"Scheduling"** UI.

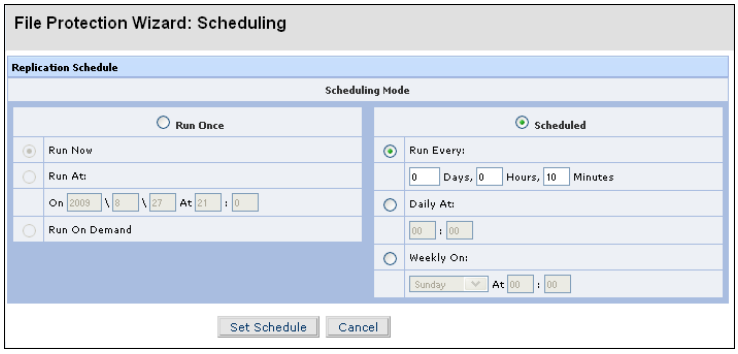


Figure 44

Step 6. Select the FX job and click **"Submit"** button.

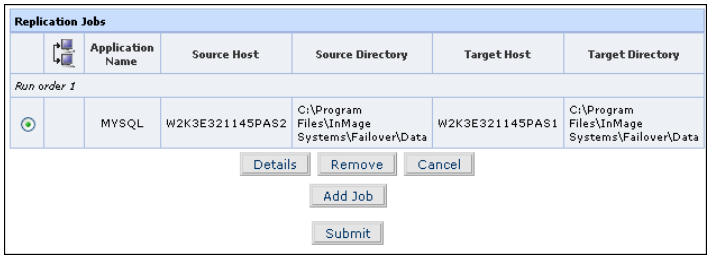


Figure 45

Step 7. Start the above scheduled FX job through **"File Protection"**. Select the above job and click the **"Start"** button.

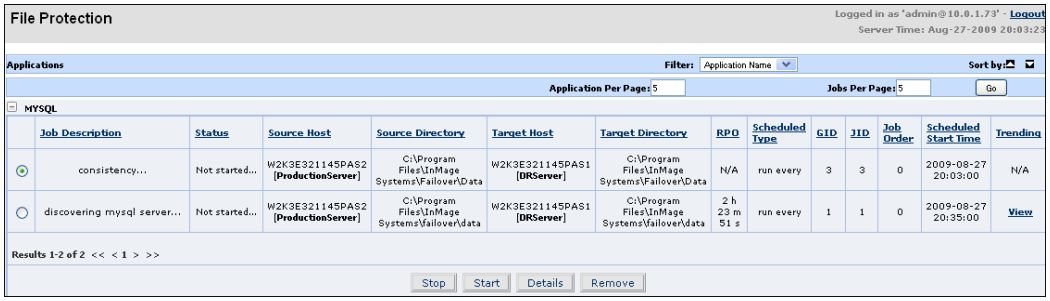


Figure 46

Step 8. You can check the status of the above job through **"Protection Status"**. The FX job status should show **"Completed"**.

Figure 47

5.3.2 Consistency through CLI

Consistency tags can also be issued through the CLI. Access the source MYSQL server's console and navigate to the InMage installation folder, then use the following command for Filesystem consistency.

Application.exe -applyTag -app MySql -s <Source Host Name> -t <Target Host Name>

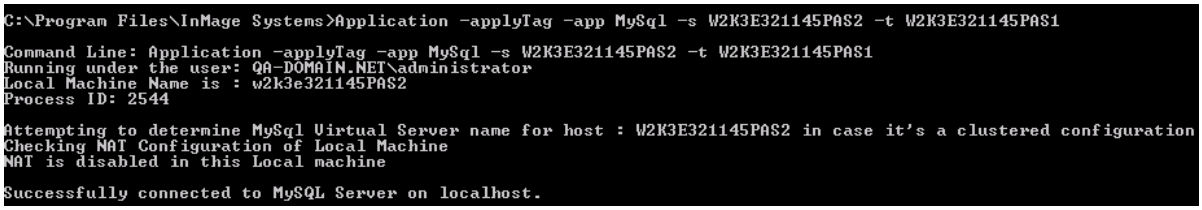


Figure 48

Caution:

While protecting large number of databases, application consistency tags may fail. You can overcome this by using the following command

Application.exe -applytag -app MYSQL -tag <*tag name*>

Additionally you can also write a batch script and automate this through the FX job.

6 Recover

MySQL Server can be recovered by performing Planner Failover, Un-Planned Failover or Failback.

6.1 Planned Failover

Failover can be a planned one or un-planned also.

Table 1: Differences between unplanned failover and planned failover

Activity	Unplanned Failover	Planned Failover
When to Use	Real-time or unexpected outages	Mock drills, To test the setup. Any expected outages
Recovered to	To latest consistent tag common for all MySQL replication pairs	Issues a consistency tag on the MySQL volumes. Target is recovered to the same consistent tag.
Performed Through CX UI	Yes, Unless the CX is down	Yes
Performed Through CLI	Yes	Yes

6.1.1 Through CX UI

You can perform a planned failover by setting a replication pair through CX UI. Follow the following steps for planned failover:

Step 1. To set a FX job, navigate to **"File Protection".** Click **"New Job Wizard".** The **"Add Job"** UI appears. Click **"Add Jobs".**

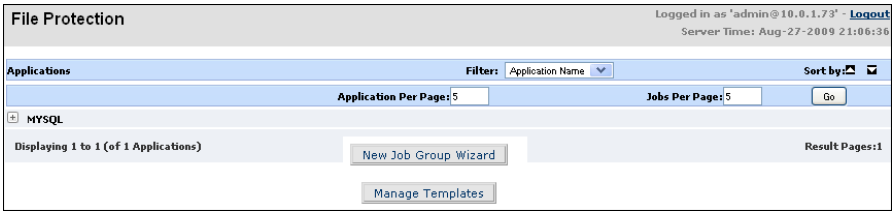


Figure 49

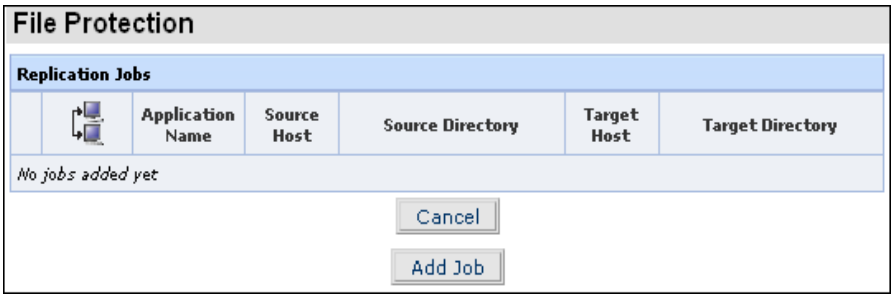


Figure 50

Step 2. Provide **"Application Name"** and **"Job Description"** for the FX job. Select Production Server volume and DR volume. Provide the source and target directory for installation path for "Failover\Data". Choose "MySQL (Windows) Planned Failover" template from the drop down menu. Click **"Next".**

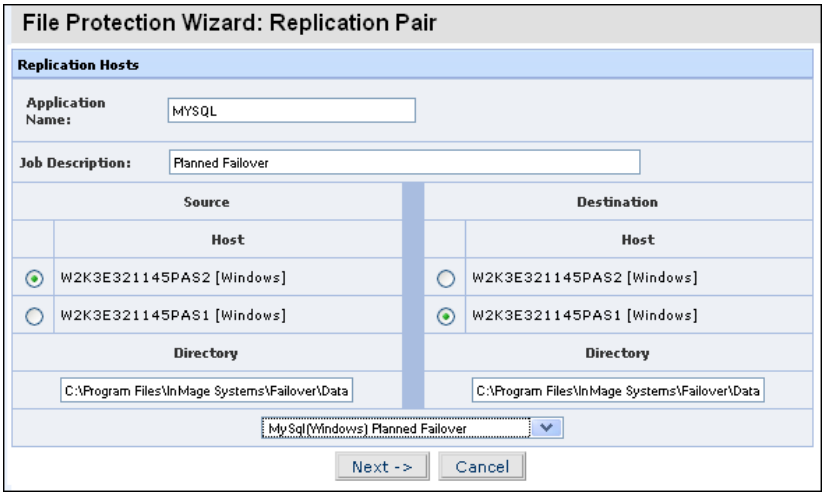


Figure 51

Step 3. Choose the required **"FX job options"** and click **"Submit"** button.

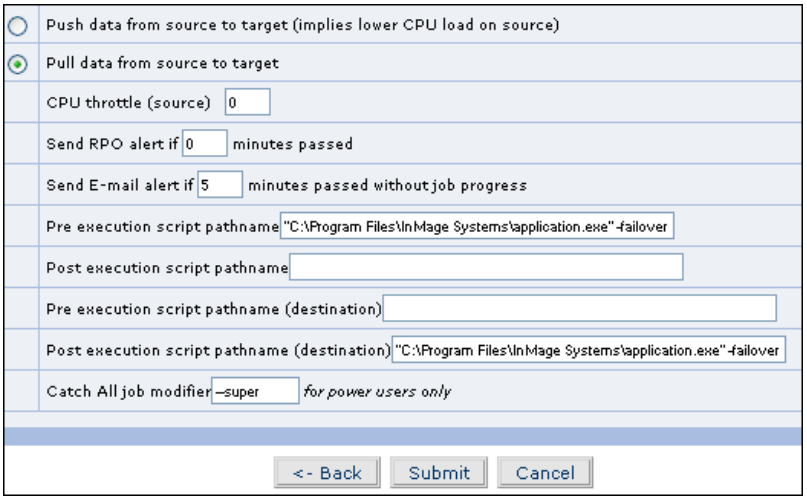


Figure 52

Step 4. Set the schedule for this job as **"On Demand".** Select the FX job and click **"Submit"** button.

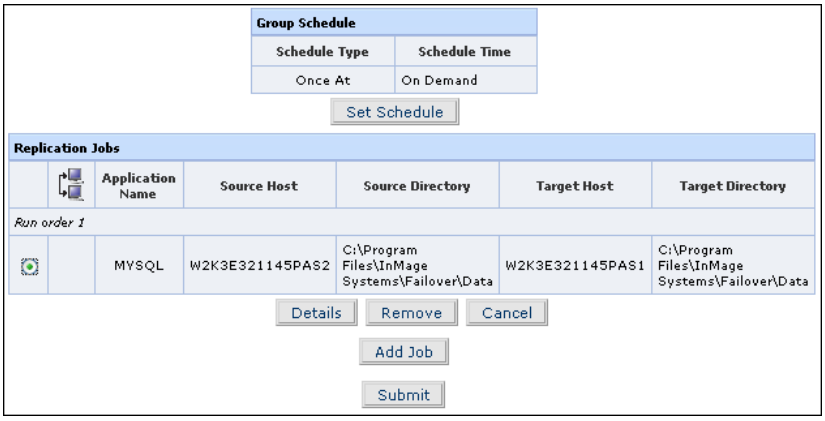


Figure 53

Step 5. Navigate to **"File Protection"** and select the required FX job and click **"Start"** button. Once the job completes execution, the MYSQL server replications will disappear from the CX UI. This is because the target volumes are rolled back to a consistent point. This concludes Planned MYSQL server failover through CX UI.

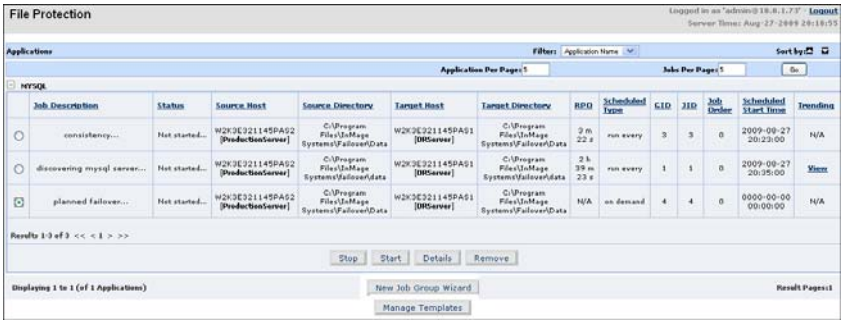


Figure 54

Step 6. You can check the status of the FX job through "Protection Status".

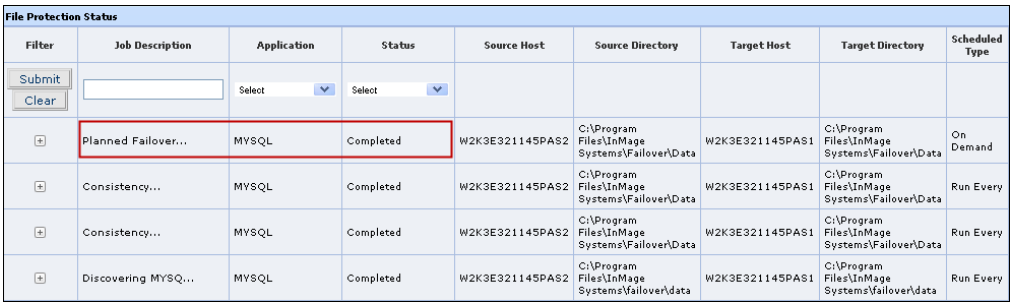


Figure 55

Notes:

On a NAT environment, always append the –nodnsfailover switch on the target post script. This will skip all DNS related operations

By default AD will not be replicated during File server failover. However if you need AD replication as part of File Server failover edit the command line/pre and post script with the "–doadreplication" switch.
OR
You may run "WinOp.exe" from command line to perform AD replication and DNS Update

6.1.2 Through CLI

Planned failover can be performed through CLI as well. Proceed as follows to do a planned failover through CLI.

Step 1. Access the source MYSQL server's console and navigate to the InMage installation folder to issue the following command

`Application –failover –planned –app MySql –s <source MYSQL server name> -t <target MYSQL server name> -builtin –tag NONE`

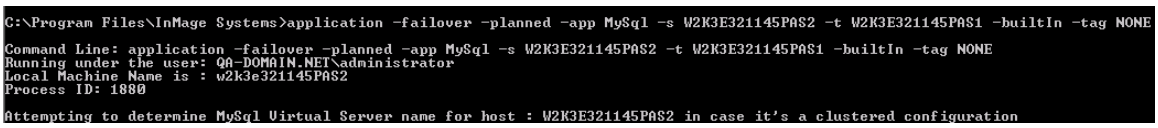


Figure 56

Step 2. Copy "Failover\Data" from Installation folder to DR Server with same directory structure.

Step 3. At the end of the output another command is displayed within the "**important information**". Execute this command from the target MYSQL server console.

Step 4. Now switch to the target console, navigate to InMage installation folder and issue the above mentioned command. This concludes planned failover through CLI.

Figure 57

Notes:

On a NAT environment, always append the –nodnsfailover switch on the source command. This will skip all DNS related operations

The command under the IMPORTANT INFORMATION will also show the same "-nodnsfailover" switch to the end

For MYSQL the –app switch will be MySql

To failover to an application tag or user defined consistency tag, append the command with -tag <name of the tag> -tagtype <type of the tag>.

If the tagtype is not mentioned, by default the FS tagtype is considered

6.2 Un-Planned Failover

6.2.1 Through CX UI

You can perform an un-planned failover by setting a replication pair through CX UI. Follow the following steps for planned failover:

Step 1. To set a FX job, navigate to **"File Protection".** Click **"New Job Wizard".** The **"Add Job"** UI appears. Click **"Add Jobs".**

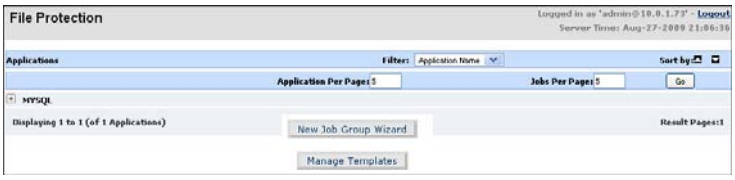


Figure 58

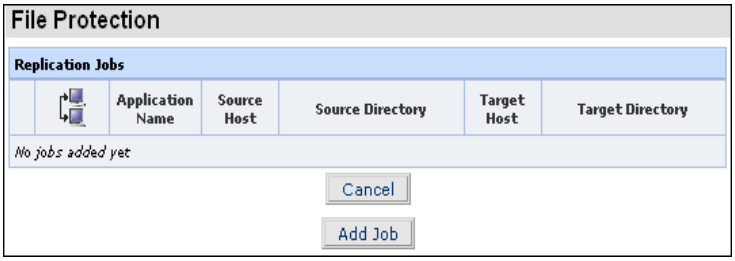


Figure 59

Step 2. Provide **"Application Name"** and **"Job Description"** for the FX job. Select source volume and target volume. Provide the source and target directory for installation path for **"Failover\Data"**. Choose **"MySQL (Windows) Un-Planned Failover"** template from the drop down menu. Click **"Next".**

Figure 60

Step 3. Choose the required "FX job options" and click **"Submit"** button.

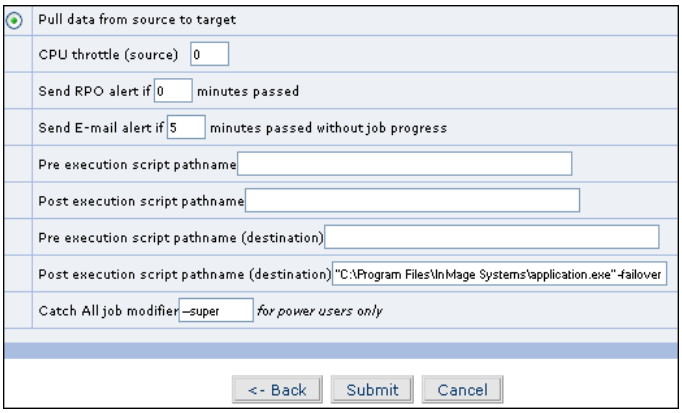


Figure 61

Step 4. Set the schedule for this job as **"On Demand".** Select the FX job and click **"Submit"** button.

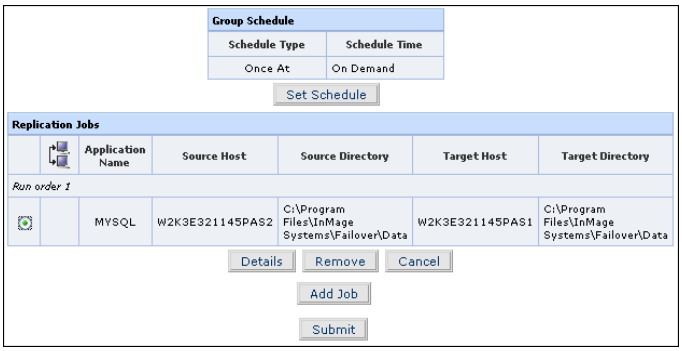


Figure 62

Step 5. Navigate to **"File Protection"** and select the required FX job and click **"Start"** button. Once the job completes execution, the MYSQL server replications will disappear from the CX UI. This is because the target volumes are rolled back to a consistent point. This concludes un-planned MYSQL server failover through CX UI.

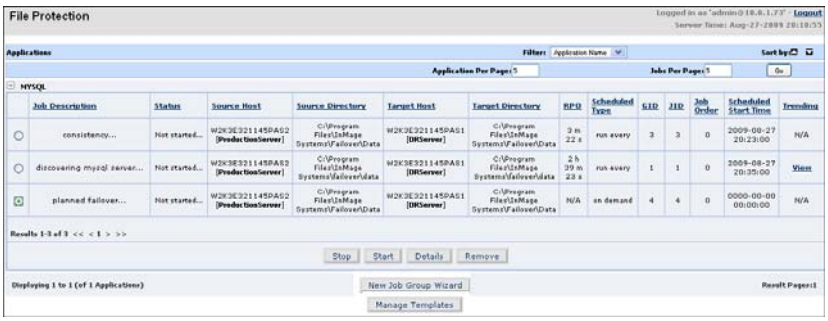
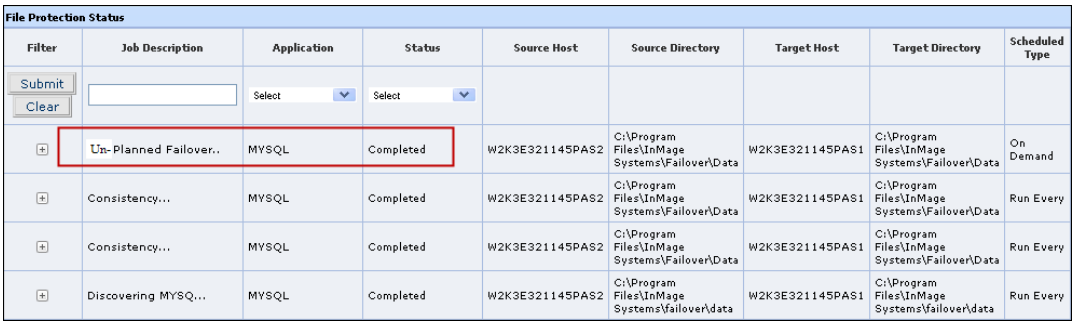


Figure 63

Step 6. You can check the status of the above FX job through "Protection Status".



File Protection Status

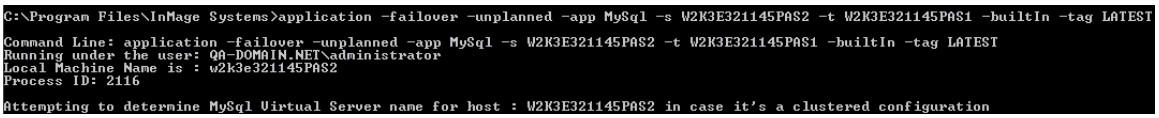
Filter	Job Description	Application	Status	Source Host	Source Directory	Target Host	Target Directory	Scheduled Type
Submit / Clear		Select	Select					
⊞	Un-Planned Failover..	MYSQL	Completed	W2K3E321145PAS2	C:\Program Files\InMage Systems\Failover\Data	W2K3E321145PAS1	C:\Program Files\InMage Systems\Failover\Data	On Demand
⊞	Consistency...	MYSQL	Completed	W2K3E321145PAS2	C:\Program Files\InMage Systems\Failover\Data	W2K3E321145PAS1	C:\Program Files\InMage Systems\Failover\Data	Run Every
⊞	Consistency...	MYSQL	Completed	W2K3E321145PAS2	C:\Program Files\InMage Systems\Failover\Data	W2K3E321145PAS1	C:\Program Files\InMage Systems\Failover\Data	Run Every
⊞	Discovering MYSQ...	MYSQL	Completed	W2K3E321145PAS2	C:\Program Files\InMage Systems\failover\data	W2K3E321145PAS1	C:\Program Files\InMage Systems\failover\data	Run Every

Figure 64

6.2.2 Through CLI

You can perform an un-planned failover through CLI. If the CX and the production servers are both down, then to perform a unplanned failover, access the target host's console to issue the following command

`Application –failover –unplanned –app MySql –s <source MYSQL server name> -t <target MYSQL server name> -builtin –tag LATEST`



```
C:\Program Files\InMage Systems>application -failover -unplanned -app MySql -s W2K3E321145PAS2 -t W2K3E321145PAS1 -builtIn -tag LATEST

Command Line: application -failover -unplanned -app MySql -s W2K3E321145PAS2 -t W2K3E321145PAS1 -builtIn -tag LATEST
Running under the user: QA-DOMAIN.NET\administrator
Local Machine Name is : w2k3e321145PAS2
Process ID: 2116

Attempting to determine MySql Virtual Server name for host : W2K3E321145PAS2 in case it's a clustered configuration
```

Figure 65

6.3 Failover without CDP Retention Option

6.3.1 Through CX UI

To perform a failover for replication pairs without CDP retention through CX UI, set a FX Job as in Section Planned Failover on Page 32 of this current document, while choosing template as "**MYSQL(windows) Failover without Retention**".

6.3.2 Through CLI

To perform Failover without CDP retention option through the CLI, access the target console and navigate to the agent installation folder and issue the following command:

`Application.exe –failover –unplanned –app MySql –s <source host name> -t <target host name> -builtin –tag NONE`

Here the target volume will be made visible and the replication will pause. For pairs with CDP retention options enabled the replication pairs will be stopped.

```
C:\Program Files\InMage Systems>application  -failover -unplanned -app MySql -s
W2K3E321145PAS2 -t W2K3E321145PAS1 -builtIn -tag NONE

Command Line: application  -failover -unplanned -app MySql -s W2K3E321145PAS2 -t
 W2K3E321145PAS1 -builtIn -tag NONE
Running under the user: Administrator
Local Machine Name is : w2k3e321145PAS1
Process ID: 1376

Attempting to determine MySql Virtual Server name for host : W2K3E321145PAS2 in
case it's a clustered configuration
```

Figure 66

6.4 Failback

The Failback process is always planned one.

6.4.1 Discovery MySQL

Through CX UI

To discover MySQL database set a reverse replication pair i.e. the current target will behave as a source and current source will behave as a target. To set the replication pair through CX UI, refer to Through CX UI on Page 22 of this current document.

Once the job is executed successfully, proceed to perform a VX replication.

Through CLI

To discover MySQL database set a reverse replication pair i.e. the current target will behave as a source and current source will behave as a target. To set the replication pair through CLI, refer to Through CLI on Page 25 of this current document.

6.4.2 Replicate Discovered Volume

Step 1. Navigate to "**Volume Protection"** select the source volume (i.e. DR server), select the target volume (i.e. Production Server), and then click on "**Start Replication**".

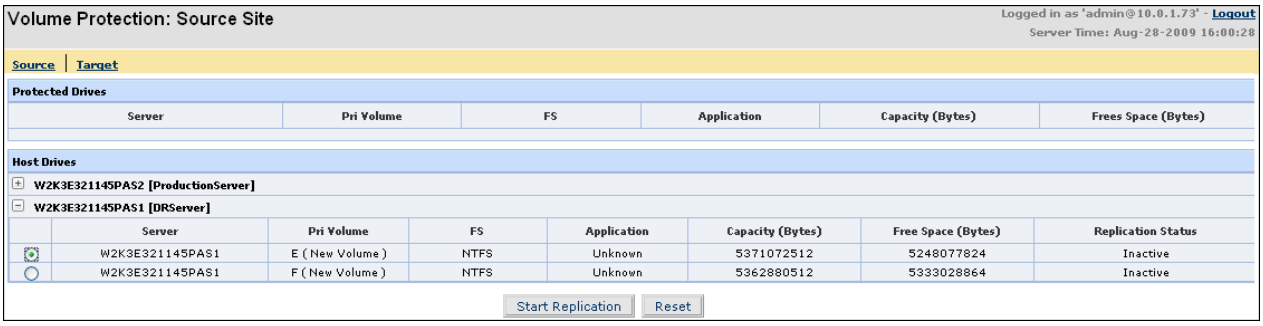


Figure 67

Step 2. Select the target (i.e. source Production server), and select the "**Replication Options".** Check the option "**Enable CDP Retention option",** the rest are optional, click on "**Submit**"

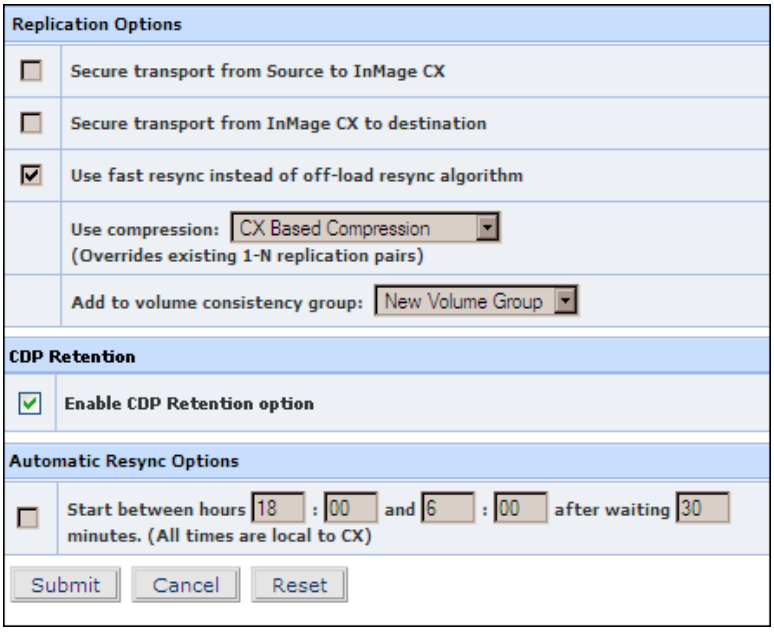


Figure 68

Step 3. Define the type of retention policy, and then click on "**Submit"** to start the replication pair.

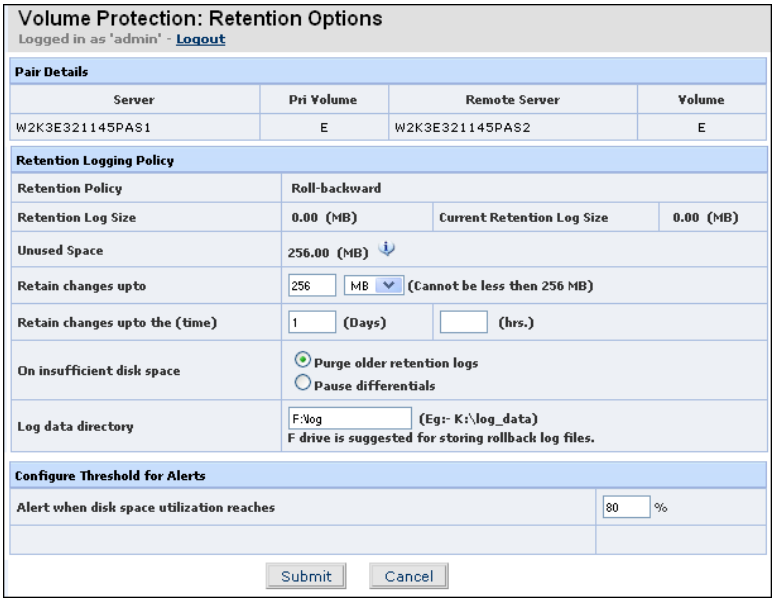


Figure 69:

Step 4. The replication starts and this can be monitored through "**Protection Status"** on the CX UI.

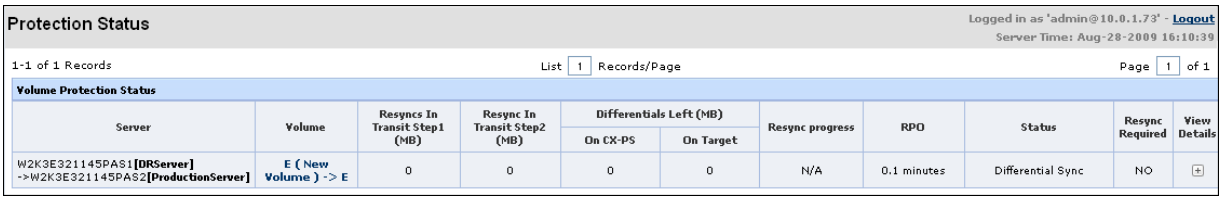


Figure 70:

6.4.3 Issue Consistency Tags

To issue consistency tags through CX UI and CLI also. Refer Issue Consistency Tags on Page 27 of this document.

Caution:

While protecting large number of databases, application consistency tags may fail. You can use the file system tag to overcome this error.

Additionally you can also write a batch script and automate this through the FX job.

6.4.4 MYSQL Failback

Through CX UI

You can perform MySQL Failback through CX UI. Once the consistency job is complete, perform a failback to the production server (source host). This is done by executing the "**MYSQL(windows) planned failback**" FX template.

Follow the following steps for MySQL Failback:

Step 1. To set a FX job, navigate to **"File Protection".** Click **"New Job Wizard".** The **"Add Job"** UI appears. Click **"Add Jobs".**

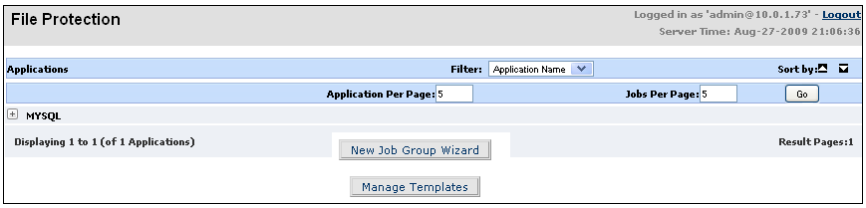


Figure 71

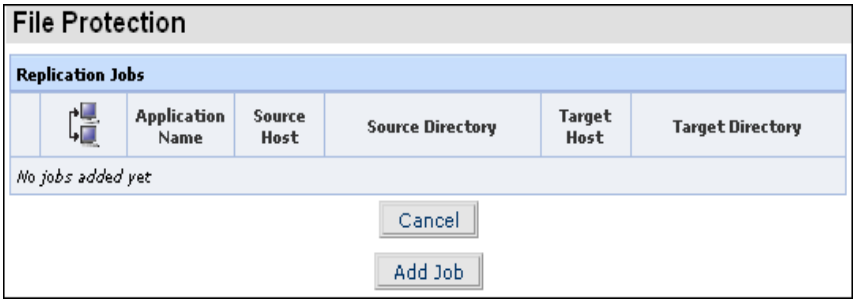


Figure 72

Step 2. Provide "Application Name" and "Job Description" for the FX job. Select source volume and target volume. Provide the source and target directory for installation path for "Failover\Data". Choose "MySQL (Windows) Planned Failback" template from the drop down menu. Click **"Next".**

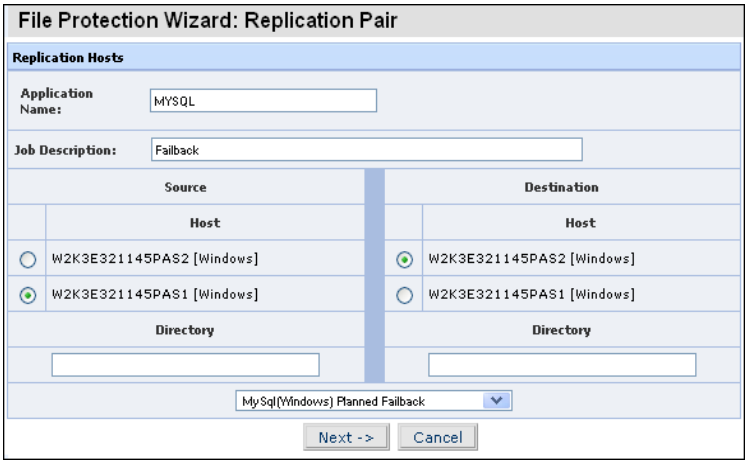


Figure 73

Step 3. Choose the required FX job options from respective "Job Options" and click **"Submit"** button.

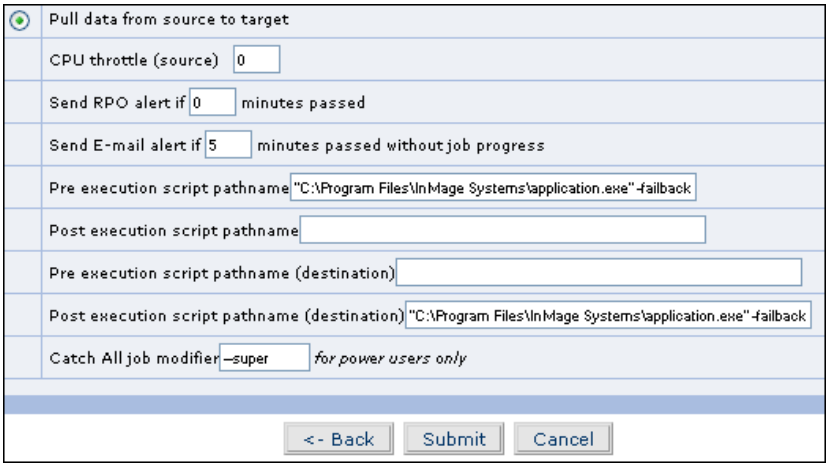


Figure 74

Step 4. Set the schedule for this job as **"On Demand".** Select the FX job and click **"Submit"** button.

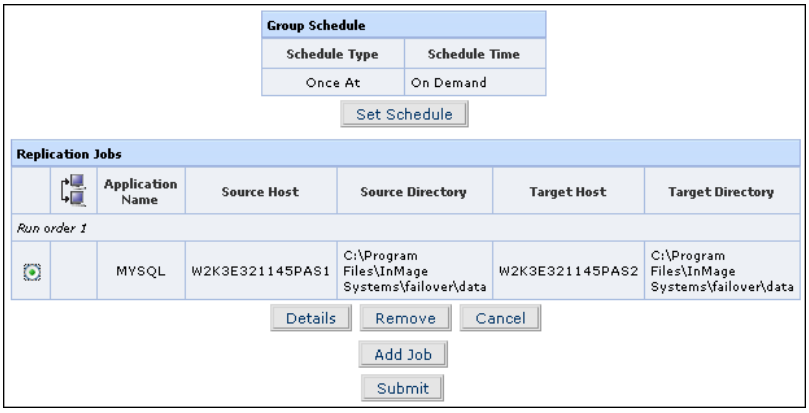
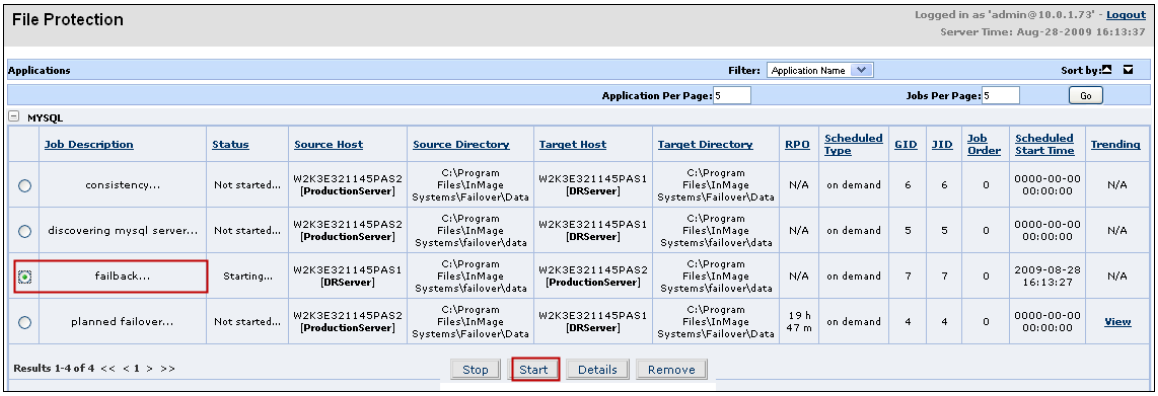
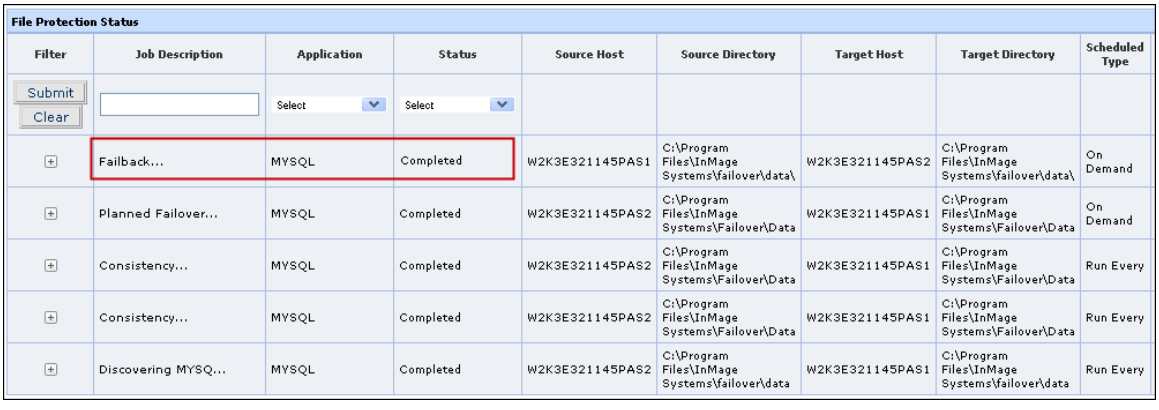


Figure 75:

Step 5. Start the Failback FX Job through File Protection. All the MySQL server replication pairs between the original target and original source stops.



Step 6. You can check the status of Failback FX job through "Protection Status".



Through CLI

To perform Failback through CLI, proceed as follows:

Step 1. Switch to original target's console (now the source) and issue the following command **`Application-failover –planned –app MySql –s <source server, in this case original target> -t <target server, in this case original source> -builtin –tag NONE`**

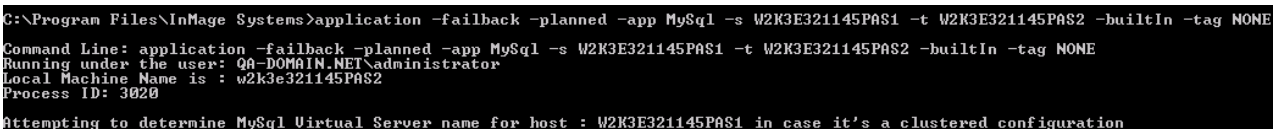


Figure 76:

Step 2. Copy "Failover\Data" from Installation folder to DR server.

Step 3. The above command ends with a resultant command that is it to be executed on the target server, copy the command and execute it on the target console.

Step 4. After accessing the target console, navigate to the InMage installation folder then issue the command as given in the previous step's output.

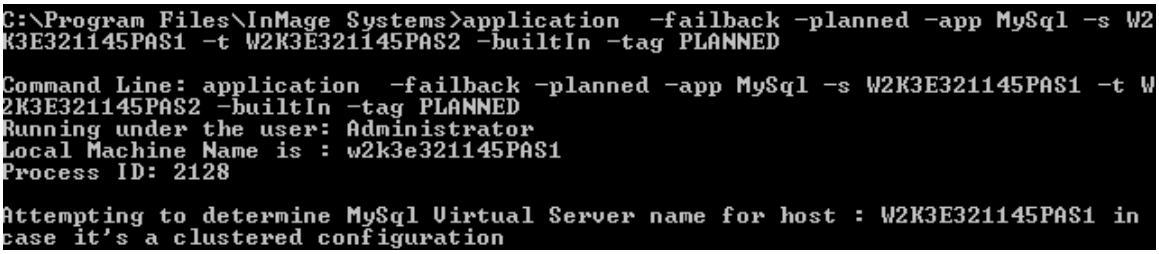


Figure 77

7 Failover and failback using crash consistency tag

Hitachi Dynamic Replicator - Scout uses VSS snapshot to issue application or consistency tag but in setup when it is not possible to issue VSS consistency tag, you can use crash consistency tag. The crash consistency tag can be used for failover and failback. When crash consistency tag is used it always generates USERDEFINED tag. It will not generate any filesystem or application consistency tag.

7.1 Discovery

Use the application discovery template as used in planned failover. No changes are required.

7.2 Consistency

To set crash consistency FX job use the ***MYSQL51_consistency_fstag.bat*** file and pass the required argument as given below.

Syntax:

```
MYSQL51_consistency_fstag.bat "volume Name" -CrashConsistency
```

For example, MYSQL51 has databases in G volume. Use the following command in prescript to set the crashconsistency job or run it through CLI.

```
"C:\Program Files (x86)\InMage
Systems\consistency\SQL2005_consistency_fstag.bat” "G:" -CrashConsistency
```

7.3 Planned Failover

7.3.1 Planned failover through CX UI

When you set the planned failover job, in the planned failover template, add the prescript with "**–crashconsistency**" and postscript with-"**-tagtype USERDEFINED**"
For example,

Pre script: "C:\Program Files (x86)\InMage Systems\application.exe" -failover -planned -app MySql -s W2K3-SRC -t W2K3-TGT -builtIn -tag NONE **-CrashConsistency**

Post script: "C:\Program Files (x86)\InMage Systems\application.exe" -failover -planned -app MySql -s W2K3-SRC -t W2K3-TGT -builtIn -tag PLANNED **-tagtype USERDEFINED**

7.3.2 Planned Failover through CLI

When you run through CLI follow the same procedure as in Section Planned Failover Through CLI on page 35 The only change is add "**–crashconsistency**" switch to the source command and target command with "**-tagtype USERDEFINED**".

Source Command:
"C:\Program Files (x86)\InMage Systems\application.exe" -failover -planned -app MySql -s W2K3-SRC -t W2K3-TGT -builtIn -tag NONE **-CrashConsistency**

Target command:
"C:\Program Files (x86)\InMage Systems\application.exe" -failover -planned -app MySql -s W2K3-SRC -t W2K3-TGT -builtIn -tag <User Defined Tag> **-tagtype USERDEFINED**

7.4 Failback

7.4.1 Failback through CX

Use the same failback template as in Section Failback Through CX UI on page 40. The only change is add"**–crashconsistency**" switch to the source command and target command with "**-tagtype USERDEFINED**".

For example,
Pre script: "C:\Program Files (x86)\InMage Systems\application.exe" -failback -planned -app MySql -s W2K3-TGT t W2K3-SRC -builtIn -tag NONE **-CrashConsistency**

Post script: "C:\Program Files (x86)\InMage Systems\application.exe" - failback -planned -app MySql -s W2K3-TGT -t W2K3-SRC -builtIn -tag PLANNED **-tagtype USERDEFINED**

7.4.2 Failback through CLI

When you run through CLI follow the same procedure as in Section Failback Through CLI on page 40 . The only change is add "**–crashconsistency**" switch to the source command and target command with "**-tagtype USERDEFINED**".

Source command:
"C:\Program Files (x86)\InMage Systems\application.exe" - failback -planned -app MySql -s W2K3-TGT -t W2K3-SRC -builtIn -tag NONE **-CrashConsistency**

Target command:
"C:\Program Files (x86)\InMage Systems\application.exe" - failback -planned -app MySql -s W2K3-TGT -t W2K3-SRC -builtIn -tag <User Defined Tag> **-tagtype USERDEFINED**

7.5 Unplanned Failover

When you set the unplanned failover job, in the unplanned failover template, add the postscript with-"**-tagtype USERDEFINED**". The same can be used for the CLI

For example,
"C:\Program Files (x86)\InMage Systems>application.exe -failover -unplanned -app MySql -s W2K3-SRC –t W2K3-TGT -builtIn -tag LATEST **-tagtype USERDEFINED"**

8 Limitations

- Works only on 32 bit platform
- Supports only a single instance (default instance, no multiple instances of
- MySql Server)
- Does not support MySQL Cluster environment.

Hitachi Data Systems

Corporate Headquarters
750 Central Expressway
Santa Clara, California 95050-2627
U.S.A.
Phone: 1 408 970 1000
www.hds.com
info@hds.com

Asia Pacific and Americas
750 Central Expressway
Santa Clara, California 95050-2627
U.S.A.
Phone: 1 408 970 1000
info@hds.com

Europe Headquarters
Sefton Park
Stoke Poges
Buckinghamshire SL2 4HD
United Kingdom
Phone: + 44 (0)1753 618000
info.eu@hds.com

Hitachi Data Systems

MK-98DF8220-00