Recover CCE Data with Manual Synchronization Methods

Contents

Introduction Prerequisites Requirements Components Used Problem Statement - Logger/HDS Database Option 1: ICMDBA Manual Synchronization (Logger A to Logger B as an example) Export the Configuration Data Synchronization the Configuration Data Option 2: Manual Synchronization - SQL Management Studio (HDS 1 to HDS 2 as an example) Export the HDS Database Restore the HDS Database Backup

Introduction

This document describes how to recover data from Cisco Contact Center Enterprise (CCE) Historical Data Server (HDS) or Logger.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Unified Contact Center Enterprise (UCCE)
- Package Contact Center Enterprise (PCCE)

Components Used

The information in this document is based on these software versions:

• Unified Contact Center Enterprise (UCCE/PCCE) 12.5

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Problem Statement - Logger/HDS Database

If services are stopped on a Logger/HDS, such as due to a network or service outage, data could

be differed or missed in the reports. The logger that has the data gap needs to be re-synced to its peer and then the HDS is reconfigured to pull back the data from the logger.

Option 1: ICMDBA Manual Synchronization (Logger A to Logger B as an example)

You must perform these procedures to synchronize the Logger configuration data between Logger A and Logger B :

- 1. Export the configuration data on Logger A and Logger B.
- 2. Synchronize the configuration data in the database between Logger A and Logger B.

Export the Configuration Data

Complete these steps to export the configuration data on both Logger A and Logger B:

Step 1. Stop the Logger service.

Step 2. Select Start > Run, type ICMDBA, and click OK to start ICMDBA.

Step 3. Select the database as shown by arrow A in the image to specify the server, instance, and node.



Step 4. Select Data > Export from the menu bar, as shown by arrow B in the image.

Export data from - citi_sideA	×
V Lockard Changes	Export
Truncate Config Message Log	Cancel
Data type:	Help
CONFIGURATION -	
Destination Path:	
C:\icm\bin	Browse

Step 5. Select **CONFIGURATION** from the Data type dropdown list.

Step 6. Use the Browse button to indicate the data destination path.

Step 7. Check Lockout Changes to ensure that changes are not made to the database during the export

operation.

Step 8. Click Export.

Export	x
	^
< >	~
Start Cancel	

Step 9. Click **Start** to export the configuration data. When the process is complete, a message is displayed

which indicates the action was successful. Click **OK** and then click **Close** to exit. Step 10. Start the Logger service.

Synchronization the Configuration Data

This procedure copies valid configuration data from the Logger to the paired Logger. Complete the steps in one of these sections to synchronize the Logger database.

Copy the Configuration Data from Logger A to Logger B.

Copy the Configuration Data from Logger B to Logger A.

Complete these steps if the configuration data in Logger A is valid and you want to copy the data to Logger B:

Step 1. Stop the Logger service on Logger B.

Step 2. Start ICMDBA on Logger A and complete these steps to copy the configuration data from Logger A to Logger B, where Logger A is the source and Logger B is the target:

a. Select the database as shown by arrow A in the image to specify the server, instance, and node.

b. Select Data > Synchronize from the menu, as shown by arrow C in the image. The Synchronize window is displayed:

	x
Instance: citi	Synchronize
C Lockout Changes	Cancel
Truncate Config Message Log	Help
Source	
Server: Server:	
12UCCE-RA Add 12UCCE-RB	Add
Database: Database:	
citi_sideA 💌 citi_sideB 💌	

c. Check Lockout Changes to ensure that changes are not made to the database during the synchronization operation.

d. Select the server name and database for both source and target from the dropdown lists. Click **Add** and enter the server name in the Add Server box to select a server that is not in the dropdown list.

e. Click Synchronize.

The Synchronize box is displayed.

f. If the services are started, you get the pop up as shown in the image. To stop the service, click **Yes** and proceed further. This stops the service on the targeted side.

ICMDBA		x
?	You may not proceed with this operation while "Cisco ICM citi LoggerB" is running on the server "12UCCE-RB". Would you like to stop it?	
	Yes No	

Synchronize	x
	^
	~
Start Cancel	

- g. Click **Start** to synchronize the data. A confirmation window is displayed:



h. Verify the database name whose configuration data is to be replaced and the server name on which the database resides. If both are correct, click **Yes** to continue. When the process is complete, a message is displayed which indicates that the action was successful. Click **OK** and then click **Close** to exit.

Step 3. Start the logger service on Logger B.

Option 2: Manual Synchronization - SQL Management Studio (HDS 1 to HDS 2 as an example)

You must perform these procedures to synchronize the HDS database data between HDS A and HDS B :

1. Export the HDS data backup on HDS 1 and HDS 1.

2. Restore the HDS backup to the corrupted database. (HDS 2 is corrupted here, so restore the HDS 1 backup to the HDS 2.)

Export the HDS Database

Complete these steps to export the data on both HDS 1 and HDS 2:

- Step 1. Stop the Distributor service.
- Step 2. Open SQL Management Studio on the HDS server.
- Step 3. Select the database as shown in the image to specify the server, instance, and node.



Step 4. Right click on the database -> Task -> Backup and save it as ".bak" file

SQLQuery1.sql - 12UCCE-AW.citi_hds (CHASE\administrat	Detach	n) Qui	ick l
File Edit View Project Tools Window Help	Take Offline		
🖸 - O 🏠 - 🖆 - 🚔 💾 🔐 📠 New Query 📠 🕻	Bring Online	_	
8 ♥ Ŷ♥ citi_hds • ▷ Execute ■ √	Stretch	► 2 =	
Object Explorer 👻 म	Encrypt Columns		
Connect 👻 🍟 🗏 🝸 🖸 🚸	Data Discovery and Classification		
🗉 🐻 12UCCE-AW (SQL Server 14.0.1000.169 - CHASE\administ	Vulaerakiita Assessment		
🖃 📁 Databases	Vulnerability Assessment	•	
🗉 🧰 System Databases	Shrink	•	
🛞 🧰 Datab New Database		_	
E citi_av New Query	Back Up		
E Citi_h Script Database as	Restore	•	
🗉 🗐 Security	Mirror		
Server Ot Tasks			
Replicatio Policies	Launch Database Mirroring Monitor		
🗉 📁 PolyBase	Ship Transaction Logs		
Always O Facets	Generate Scripte		
🗉 📶 Managen 🛛 Start PowerShell	Generate Scripts	one	UOL
E Integratic Azure Data Studio	Generate In-Memory OLTP Migration Checklists		

🗧 Back Up Database - citi_hds	;				-	•	x
Select a page ≱ General	🗊 Script 👻 😮 Help						
Media Options Backup Options	Source						
	Database:	citi_hds					~
	Recovery model:	SIMPLE					
	Backup type:	Full					¥
	Copy-only backup						
	Backup component:						
Connection	Database						
Server: 12UCCE-AW	O Files and filegroups:						
Connection: CHASE\administrator	Destination						
Vew connection properties	Back up to:	Disk					~
	C\Pmoram Files\Microsoft SQL Serve	w\MSSOL12 MSSO	LSERVER\MSSOL\B	ackun ⁱ oti bols bak			_
						Add	
Progress						Remov	
C Ready					ľ	Conten	
.415-					[
					ОК	Can	cel

Restore the HDS Database Backup

Complete these steps to export the data on to HDS 2 and restore it.

Step 1. Stop the Distributor service.

Step 2. Take the backup file for HDS 1 and transfer/FTP it to HDS 2 server.

Step 3. Open SQL Management Studio on the HDS 2 server.

Step 4. Select the database as shown in the image to specify the server, instance, and node.



Step 5. Restore the database. Use the SSMS and select the back up file transferred from HDS 1 through FTP/transfer.

Right click on the database > Task > Restore > Database

Choose **From device** button and select the **<database>.bak** Click **Restore** and check box. Go to **Options** tab, click **Overwrite** the current database. Click **OK**.

🕕 Ready		
Select a page	🗊 Script 🔹 😮 Help	
General		
V Files	Source	
	O Database:	cti_hds V
	Device:	C/HDS BACKUP/HDSA.bak
	Database	
	Database.	00_100
	Destination	
	Database:	cti_hds v
	Restore to:	The last backup taken (Thursday, August 18, 2022 9:16:31 AM) Timeline
Connection	Restore plan	
12UCCE-AW ICHASE\administrator]	Backup sets to restore:	
	Restore Name	Component Type Server Database Position First LSN
	citi_hds-Full Dat	atabase Backup Database Full 12UCCE-AW ctl_hds 1 8600000209120007
Vew connection properties		
Progress		
Done	<	III >
\checkmark		Verfy Backup Media
		OK Cancel Help
P. Darton Database with the		
Kestore Database - citi_nos		
Ready		
Select a page	🗐 Script 🔹 😮 Help	
Files		
🖋 Options	Restore options	
	 Overwrite the existing 	g database (WITH REPLACE)
	Preserve the replication	ion settings (WITH KEEP, REPLICATION)
	Restrict access to the	e restored database (WITH RESTRICTED_USER)
	Restrict access to the Recovery state:	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY
	Restrict access to the Recovery state: Standby file:	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY <td< td=""></td<>
	Restrict access to the Recovery state: Standby file: Leave the database ready	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL by to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.
	Restrict access to the Recovery state: Standby file: Leave the database ready	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL dy to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.
Connection	Restrict access to the Recovery state: Standby file: Leave the database ready Tail-Log backup	erestored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL dy to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.
Connection VI 12UCCE-AW [CHASE\administrator]	Restrict access to the Recovery state: Standby file: Leave the database ready Tail-Log backup Tail-Log backup	erestored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY • C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ty to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore
Connection ¥₩ 12UCCE-AW [CHASE\administrator]	Restrict access to the Recovery state: Standby file: Leave the database ready Tail-Log backup Tail-Log backup Cave tail-log backup to Leave source to Cave tail-log backup to Cave tail-l	te restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ty to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state COVERY
Connection 12UCCE-AW [CHASE\administrator]	Restrict access to the Recovery state: Standby file: Leave the database read Tail-Log backup Take tail-log backup Leave source d (WITH NOREC	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL dy to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state
Connection VI 12UCCE-AW [CHASE\administrator] View connection properties	Restrict access to the Recovery state: Standby file: Leave the database read Tail-Log backup Tail-Log backup Leave source d (WITH NOREC Backup file:	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ty to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state COVERY) C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL
Connection VI 12UCCE-AW [CHASE\administrator] View connection properties Progress		e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL dy to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state COVERY) C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL times to do direction database.
Connection	Restrict access to the Recovery state: Standby file: Leave the database read Tail-Log backup Take tail-log backup I Leave source o (WITH NORECO Backup file: Server connections Close existing connect	erestored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL dy to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state COVERY) C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ctions to destination database
Connection	Restrict access to the Recovery state: Standby file: Leave the database read; Tail-Log backup Tail-Log backup Take tail-log backup I Leave source of (WITH NOREC) Backup file: Server connections Close existing connect	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL dy to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state COVERY) C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ctions to destination database
Connection View connection properties Progress Done Done	Restrict access to the Recovery state: Standby file: Leave the database read Tail-Log backup Tail-Log backup Tail-Log backup Case essure of Recover connections Cose existing connect	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ty to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state COVERY) C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ctions to destination database
Connection	Restrict access to the Recovery state: Standby file: Leave the database read Tail-Log backup Tail-Log backup Tail-Log backup Leave source o (WITH NORECO Backup file: Server connections Close existing connect Prompt	e restored database (WITH RESTRICTED_USER) RESTORE WITH RECOVERY C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL dy to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. before restore database in the restoring state COVERY) C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL ctions to destination database

This restore shows no progress for the backup check for corruption, but it starts to import once the verification is done. Checks for corruption can take a while. It depends on the speed of the server and size of the database.

Step 6. After successful import, run the queries as follows against HDS 2 database.

Truncate table Recovery

Truncate table Logger_Admin (only if you take backup and restore Logger Databases, not applicable for HDS)

This truncates table "Recovery" (that is delete all rows of table' Recovery') in the database. After it's truncated and services started, the recovery process tries to get the historical data (rows) from the other HDS since the backup was taken.

Step 7. Start Distributor services. Allow enough time for data to get replicated (Logger -> HDS).

Step 8. At the end of this exercise, verify BOTH min DateTime and max DateTime match between Logger A and Logger B and between HDS1 and HDS2.

select max(RecoveryKey) from Termination_Call_Detail

select max(DateTime) from Termination_Call_Detail

select max(DateTime) from Termination_Call_Detail

Step 9. Also, you can use ICMDBA to check the Space used summary for both Logger and HDS database with min DateTime and max DateTime.