

Implementing IBM InfoSphere Change Data Capture for DB2 z/OS V6.5

Configuration

Performance tuning

Migration



Jason Arnold



International Technical Support Organization

**Implementing IBM InfoSphere Change Data Capture
for DB2 z/OS V6.5**

July 2011

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (July 2011)

This edition applies to Version 6, Release 5, Modification 1 of InfoSphere Change Data Capture for z/OS.

This document created or updated on July 26, 2011.

© Copyright International Business Machines Corporation 2011. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
Preface	vii
The author who wrote this paper	vii
Now you can become a published author, too!	vii
Comments welcome	viii
Stay connected to IBM Redbooks	viii
Chapter 1. Introduction to Change Data Capture	1
1.1 Capture	2
1.2 Transformations	2
1.3 Apply	2
1.3.1 Access Server	3
1.3.2 Management Console	4
Chapter 2. Configuring InfoSphere Change Data Capture for z/OS	5
2.1 InfoSphere Change Data Capture target libraries	6
2.2 Before configuring InfoSphere CDC on z/OS	6
2.2.1 Network	6
2.2.2 Security	6
2.2.3 Documentation	7
2.3 Create the agent-specific libraries	7
2.4 Define the product administration log	7
2.5 Define the VSAM metadata cluster	9
2.6 DB2 binds for the metadata table creation utility	10
2.7 SCHCDATA	11
2.7.1 CHCCFGxx	12
2.7.2 CHCCMMxx	12
2.7.3 CHCDBMxx	12
2.7.4 CHCLDRxx	13
2.7.5 CHCUCSF1	13
2.8 DB2 metadata tables	13
2.9 Bind DB2 plans	15
2.10 The started task	15
2.11 Access Server and Management Console	16
2.12 Access Server definitions	17
2.12.1 CDC agents	17
2.12.2 CDC user IDs	18
2.12.3 Agent/user ID mappings	20
2.13 Verifying the InfoSphere Change Data Capture installation	20
2.13.1 Creating the subscription	21
2.13.2 Creating the table mapping	21
2.13.3 Testing refresh	24
2.13.4 Testing continuous mirroring	25
2.14 Configuring notification settings	25
Chapter 3. Tuning InfoSphere Change Data Capture for z/OS	27
3.1 CHCCFGxx	28

3.1.1 CHCCFGP1	29
3.2 CHCCMMxx	29
3.2.1 CHCCMMP1	29
3.3 CHCDBMxx	30
3.3.1 CHCDBMP1	35
3.4 CHCLDRxx	35
3.4.1 CHCLDRP1	38
3.5 Latency and throughput monitoring	38
3.6 Monitoring of InfoSphere Change Data Capture CPU usage	41
3.6.1 InfoSphere Change Data Capture tasks	41
3.6.2 Using the DB2 LOAD utility for refresh	43
3.6.3 The effect of the COMMITFREQ parameter	44
Chapter 4. Upgrading from CDC V6.2	45
4.1 DB2 binds for the metadata table creation utility	46
4.2 DB2 metadata tables	47
4.3 Bind DB2 plans	49
4.4 New parameter	49
Related publications	51
IBM Redbooks	51
Other publications	51
Online resources	51
Help from IBM	51

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	LiveAudit™	System z®
DataStage®	POWER®	Tivoli®
DB2®	Redbooks®	WebSphere®
IBM®	Redpaper™	z/OS®
InfoSphere™	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

IBM® InfoSphere™ Change Data Capture for z/OS® uses log-based change data capture technology to provide low impact capture and rapid delivery of changes to and from DB2® z/OS in heterogeneous environments without impacting source systems. Customers get the up-to-date information they need to make actionable, trusted business decisions while optimizing MIPS costs. Change Data Capture can also be used to synchronize data in real time between multiple data environments to support active data warehousing, live reporting, operational business intelligence, application consolidations and migrations, master data management, and to deliver data to SOA environments.

This IBM Redpaper™ document describes InfoSphere Change Data Capture, how to install and configure it, and how to migrate to the latest release.

The author who wrote this paper

This paper was produced by a technical specialist working at the International Technical Support Organization, Poughkeepsie Center.

Jason Arnold is an IBM IT Specialist for Information Integration and Data Warehousing on System z® in the United States. He has nine years of experience with System z, including five years of experience with data warehousing and business intelligence solutions. His areas of expertise include integration of legacy data into System z data warehouses and data synchronization. Jason holds Bachelor of Science degrees in Computer Science and Mathematics from Northern Illinois University.

Thanks to the following people for their contributions to this project:

Mike Ebbers
International Technical Support Organization, Poughkeepsie Center

Dino Tonelli
IBM Poughkeepsie

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Introduction to Change Data Capture

InfoSphere Change Data Capture is an enterprise data synchronization and change capture solution. InfoSphere Change Data Capture can be used to capture changes in near real time as data is inserted, updated, and deleted in tables in a source database. These captured changes are then propagated to tables in target databases or used to drive other processes via messaging. The source and target databases could potentially be databases from different vendors running on different platforms.

The flow of data through Change Data Capture can be divided into three parts:

- ▶ Capture
- ▶ Transformations
- ▶ Apply

1.1 Capture

Changes are captured by asynchronously reading from the database recovery logs. Depending on the source database, InfoSphere Change Data Capture might read the log files directly or it might use an API to read data from the log. InfoSphere Change Data Capture has native log-based capture capabilities for the following databases:

- ▶ DB2 z/OS
- ▶ DB2 for IBM i
- ▶ DB2 Linux, UNIX, Windows
- ▶ Oracle
- ▶ SQL Server
- ▶ Sybase

Upon reading data from the database recovery logs, InfoSphere Change Data Capture filters data based on the table where the change occurred. Only data pertaining to tables of interest is retained for further processing. InfoSphere Change Data Capture then stages the changes in a holding area until a commit of these changes occurs in the source database. If a rollback occurs instead, InfoSphere Change Data Capture discards the associated changes.

1.2 Transformations

In many cases, the data models of the source and target will not be the same. InfoSphere Change Data Capture can apply transformations to the data while it is in flight between source and target. Commonly used transformations include:

- ▶ Adding other information that can be obtained from the database recovery logs
- ▶ Concatenation and other string functions
- ▶ Data type conversion
- ▶ Arithmetic
- ▶ Joining to lookup data in secondary tables
- ▶ If/then/else logic

For more complex transformations, a product such as InfoSphere DataStage® can be used in conjunction with InfoSphere Change Data Capture.

1.3 Apply

After transformations, the changes are applied to the target database. Changes are applied by executing SQL statements against the appropriate target tables. InfoSphere Change Data Capture has native apply capabilities for the following databases:

- ▶ DB2 z/OS
- ▶ DB2 on IBM i
- ▶ DB2 Linux, UNIX, Windows
- ▶ Oracle
- ▶ SQL Server

- ▶ Sybase
- ▶ Teradata

InfoSphere Change Data Capture also provides some level of customization around the SQL statements that are executed as part of apply processing. For example, it is possible to:

- ▶ Ignore deletes that occur at the source
- ▶ Convert source updates and deletes into target inserts (an audit or change history table)
- ▶ Convert source inserts into target updates (multiple source records merge to one target record)

InfoSphere Change Data Capture also supports some non-database targets, including:

- ▶ Flat files
- ▶ InfoSphere DataStage
- ▶ WebSphere® MQ
- ▶ JMS messaging

1.3.1 Access Server

Given that InfoSphere Change Data Capture is an enterprise-wide data synchronization solution, it might be in use on many physical machines across the enterprise. Some databases might be a source only, some databases might be a target only, and some databases might act as both a source and a target.

The administration server component of InfoSphere Change Data Capture is known as *Access Server*. Access Server maintains metadata describing how InfoSphere Change Data Capture is deployed across the enterprise. Only one Access Server is required to be installed in the enterprise, although multiple Access Servers can be used to provide a higher level of availability. Access Server maintains information about:

- ▶ Every CDC agent
- ▶ Every CDC user ID
- ▶ Mappings between CDC agents and CDC user IDs

A *CDC agent* is an instance of InfoSphere Change Data Capture that is installed and configured against a specific database instance. Every database that will act as either a source or a target for InfoSphere Change Data Capture needs an associated CDC agent. A single CDC agent is capable of acting as both a source and a target. During data synchronization, the source agent and the target agent communicate directly via TCP/IP.

A *CDC user ID* is a login name and password that is defined for the purposes of interacting with CDC across the enterprise. CDC user IDs map to operating system user IDs on the machines with databases that will be used as sources and targets for InfoSphere Change Data Capture. Various levels of CDC privileges are associated with each CDC user ID.

For each CDC agent that a CDC user ID needs access to, an agent/user ID mapping must be created. The mapping specifies that a particular CDC user ID is allowed to interact with the particular CDC agent. Additionally, it allows you to specify what user ID and password should be used to communicate with the database that the agent is installed against.

1.3.2 Management Console

Management Console is the end user client interface for InfoSphere Change Data Capture. Management Console is used to do the following:

- ▶ Create, modify, or delete CDC agent definitions
- ▶ Create, modify, or delete CDC user ID definitions
- ▶ Create, modify, or delete agent/user ID mappings
- ▶ Map source tables to target tables
- ▶ Start and stop data synchronization
- ▶ Monitor of InfoSphere Change Data Capture operations

To use Management Console, a user connects to an Access Server with a CDC user ID. The relationships between Management Console, Access Server, and CDC agents are summarized in Figure 1-1.

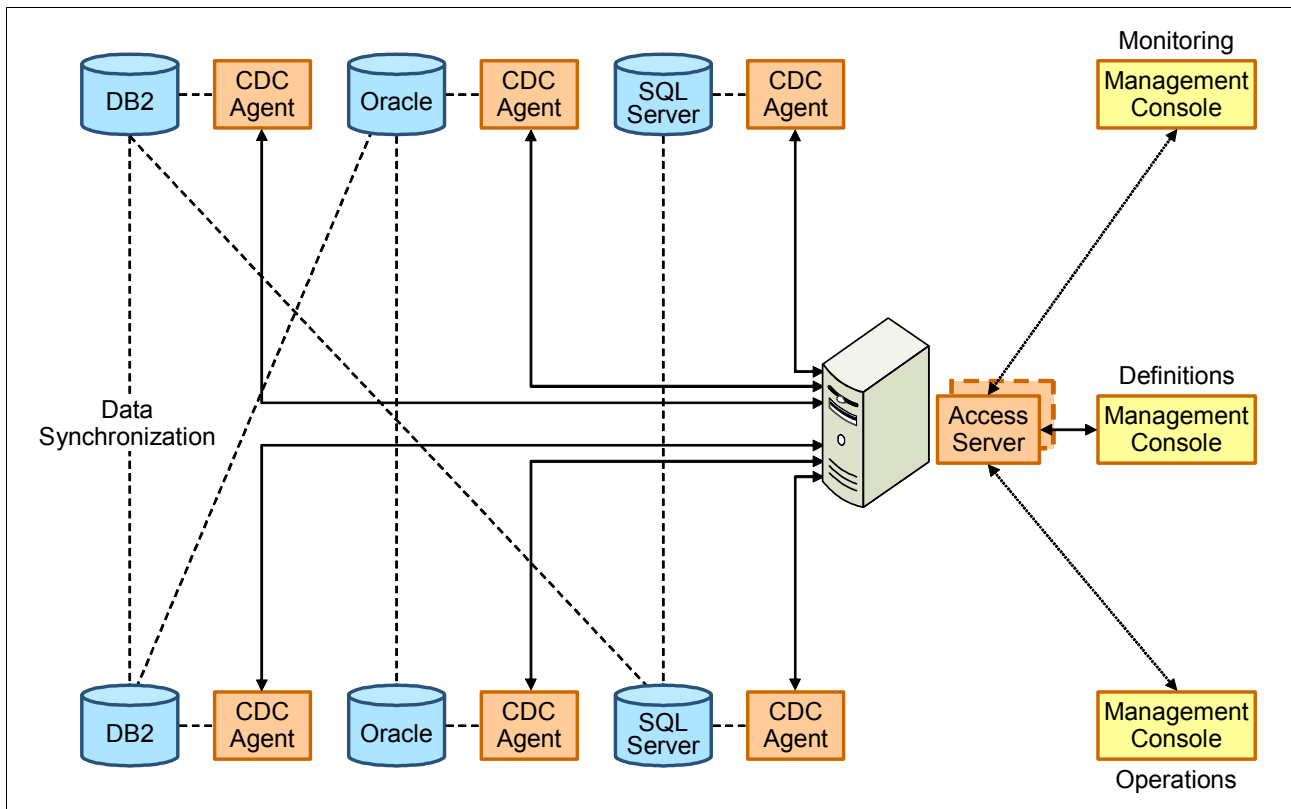


Figure 1-1 InfoSphere Change Data Capture enterprise architecture



Configuring InfoSphere Change Data Capture for z/OS

In this chapter, we explain how to configure a CDC agent for DB2 z/OS. This discussion assumes that the product has already been SMP/E installed. Refer to the program directory, GI11-7880, for more information about the SMP/E installation process. Our initial configuration was created for product verification. In the last part of this chapter, we discuss performance tuning of product parameters.

If you are doing an upgrade from V6.2 to V6.5, see Chapter 4, “Upgrading from CDC V6.2” on page 45.

2.1 InfoSphere Change Data Capture target libraries

In our installation, the target libraries were all under a high level qualifier of ICDC.ZDW. The SMP/E installation gave us the following target libraries with content as noted:

ICDC.ZDW.SCHCASM	A sample assembler user exit.
ICDC.ZDW.SCHCC	Several sample C user exits.
ICDC.ZDW.SCHCCNTL	Sample JCL.
ICDC.ZDW.SCHCCOB	Several sample COBOL user exits.
ICDC.ZDW.SCHCDATA	CDC agent configuration settings.
ICDC.ZDW.SCHCDBRM	DBRMs used by the CDC agent for DB2 z/OS.
ICDC.ZDW.SCHCH	C header files that are used when building C user exits.
ICDC.ZDW.SCHCLOAD	The load modules for the CDC agent for DB2 z/OS. This library must be APF authorized.
ICDC.ZDW.SCHCMAC	Assembler macros that are used when building assembler user exits.
ICDC.ZDW.SHCNOTC	License information.
ICDC.ZDW.SCHCTTL	Data that can be loaded into sample tables to test the product installation.

2.2 Before configuring InfoSphere CDC on z/OS

There are several things you can do to streamline and speed up the configuration process. These pre-configuration steps are broken down into three categories: networking, security, and understanding the documentation.

2.2.1 Network

You need to dedicate a TCP/IP port for InfoSphere CDC. This numeric port is defined on the z/OS LPAR on which InfoSphere CDC will run. Later, you will specify this port number in the SCHCDATA(CHCMMXX) dataset.

Make sure that all involved platforms can communicate with each other, taking into account firewalls and other network security. This includes the network connections:

- ▶ Between all sources and targets
- ▶ Between the Access Server and all source and targets
- ▶ Between Management Console and the Access Server

2.2.2 Security

Create an ID under which the CDC started task will run that has the following defined:

- ▶ An OMVS segment

- ▶ SYSCTRL in DB2

Note: This is a requirement for the started task ID, not for any user IDs. The ID for the task does not need any logon capability. SYSCTRL will be used mainly for access to the catalog, trace, and monitor authority.

- ▶ Access to all the tables on which you will be defining replication
- ▶ Alter access to all the Change Data Capture data sets

The ID of the started task must have the same name as the schema of the DB2 metadata tables. Create an ID that allows you to create the same name for the schema of the DB2 metadata tables.

APF authorization

You need to APF authorize the SCHCLOAD data set.

2.2.3 Documentation

Always read the comments in the JCL for each configuration batch job. They provide important guidance to ensure that your configuration proceeds smoothly.

For detailed documentation on installing, configuring, and using CDC, visit the InfoSphere Change Data Capture Information Center at:

<http://publib.boulder.ibm.com/infocenter/cdc/v6r5m1/index.jsp>

2.3 Create the agent-specific libraries

Make copies of SCHCCNTL and SCHCDATA. Create a copy of SCHCCNTL for each CDC agent that will be configured. One CDC agent is needed for each DB2 z/OS subsystem that will act as either a source or target for data synchronization. A single agent is capable of acting as both a source and a target. In the case of a DB2 z/OS data sharing group, only one CDC agent is needed for the whole group.

The SCHCDATA library can hold multiple configurations, so it is possible to use one copy to hold configurations for multiple CDC agents. In our environment, we created a separate copy of SCHCDATA for each CDC agent. Additionally, we allocated a new load library to hold CDC user exits.

This gave us three new libraries:

- ▶ ICDC.ZDW.DDW1.SCHCCNTL
- ▶ ICDC.ZDW.DDW1.SCHCDATA
- ▶ ICDC.ZDW.SHCEXIT

2.4 Define the product administration log

The product administration log (PAL) is used to hold CDC agent log data. This log data is viewable through the Management Console event log viewer. To define the PAL VSAM cluster, edit and run member CHCDFPAL in the SCHCCNTL sample library, as shown in Example 2-1 on page 8. You can customize the DSN and VOLSER for the PAL cluster.

Example 2-1 Defining the product administration log

```
//CHCDFPAL JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=X,REGION=OM,
//          TIME=300,NOTIFY=&SYSUID
//*
//VSAMDATA EXEC PGM=IEBDG
//SYSIN DD *
          DSD      OUTPUT=(VSAMINIT)
          FD       NAME=KEYS,STARTLOC=1,LENGTH=32,FILL=X'FF'
          CREATE   NAME=(KEYS),QUANTITY=1
/*
//SYSPRINT DD SYSOUT=*
//VSAMINIT DD DSNAME=&&INITDATA,DISP=(,PASS),
//          UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1)),
//          DCB=(RECFM=F,LRECL=36)
/*
//DEFLDPAL EXEC PGM=IDCAMS,
//          COND=(0,NE)
//SYSIN DD *
DELETE /* SO THAT THIS JOB CAN BE RERUN WITHOUT CHANGES */ -
      ICDC.ZDW.DDW1.PAL -
      CLUSTER -
      PURGE
SET MAXCC = 0
SET LASTCC = 0

DEFINE -
  CLUSTER -
    (NAME(ICDC.ZDW.DDW1.PAL) -
     VOL(DWCDC1) -
     SHAREOPTIONS(2 3) -
     TRK(4 4) -
     RECSZ(36 1024) -
     INDEXED -
     NOREUSE -
     KEYS(32 0)) -
  DATA -
    (NAME(ICDC.ZDW.DDW1.PAL.DATA)) -
  INDEX -
    (NAME(ICDC.ZDW.DDW1.PAL.INDEX))

  REPRO -
    INFILE(CHCPALDT) -
    OUTDATASET(ICDC.ZDW.DDW1.PAL)
/*
//SYSPRINT DD SYSOUT=*
//CHCPALDT DD DSNAME=&&INITDATA,DISP=(OLD,DELETE)
/*
```

2.5 Define the VSAM metadata cluster

Metadata describing the data synchronization definitions, the mappings from source tables to target tables, and alerts and alarms settings is stored in a metadata repository at the CDC agent level. Most metadata is stored in DB2 tables, but the alerts and alarms settings are stored in a VSAM cluster. Member CHCDFMTD in the SCHCCNTL library is used to define this VSAM metadata cluster. You can customize the DSN and VOLSER for the metadata cluster, as shown in Example 2-2.

Example 2-2 Defining the metadata cluster

```
//CHCDFMTD JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=X,REGION=0M,
//          TIME=300,NOTIFY=&SYSUID
//*
//FRMTDATA EXEC PGM=IEBDG
//SYSIN DD *
        DSD      INPUT=(NLDMTDTA),OUTPUT=(VSAMINIT)
        FD       NAME=KEYS,INPUT=NLDMTDTA,STARTLOC=17,LENGTH=64,      X
                FROMLOC=1
        FD       NAME=DATA,INPUT=NLDMTDTA,STARTLOC=145,LENGTH=8,      X
                FROMLOC=65
        FD       NAME=FLTRFL,STARTLOC=65,LENGTH=1332
        FD       NAME=CHNLFL,STARTLOC=153,LENGTH=505
        FD       NAME=CTRLFL,STARTLOC=1,LENGTH=16
        FD       NAME=HIKYHI,STARTLOC=17,LENGTH=127,FILL=X'FF'
        FD       NAME=HIKYL1,STARTLOC=144,LENGTH=1,FILL=X'FE'
        FD       NAME=HIKYL2,STARTLOC=144,LENGTH=1,FILL=X'FF'
        FD       NAME=DATAFL,STARTLOC=145,LENGTH=1900
        CREATE   INPUT=NLDMTDTA,NAME=(KEYS,FLTRFL),QUANTITY=1
        CREATE   INPUT=NLDMTDTA,NAME=(KEYS,DATA,CHNLFL),QUANTITY=105
        CREATE   INPUT=NLDMTDTA,NAME=(KEYS,FLTRFL),QUANTITY=1
        CREATE   INPUT=NLDMTDTA,NAME=(KEYS,DATA,CHNLFL),QUANTITY=105
        CREATE   NAME=(CTRLFL,HIKYHI,HIKYL1,DATAFL),QUANTITY=1
        CREATE   NAME=(CTRLFL,HIKYHI,HIKYL2,DATAFL),QUANTITY=1
        END
/*
//SYSPRINT DD SYSOUT=*
//NLDMTDTA DD DSNAME=ICDC.ZDW.SCHCDATA(CHCMTDIN),DISP=SHR
//VSAMINIT DD DSNAME=&&INITDATA,DISP=(,PASS),
//          UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1)),
//          DCB=(RECFM=VB,BLKSIZE=4096,LRECL=2048)
/*
//BLDMTDTA EXEC PGM=IDCAMS
//SYSIN DD *
        DELETE /* SO THAT THIS JOB CAN BE RERUN WITHOUT CHANGES */ -
                ICDC.ZDW.DDW1.META -
                CLUSTER -
                PURGE
        SET MAXCC = 0
        SET LASTCC = 0

        DEFINE -
                CLUSTER -
                (NAME(ICDC.ZDW.DDW1.META) -
```

```

VOL(DWCDC1) -
SHAREOPTIONS(2 3) -
CYL(20 20) -
RECSZ(512 2048) -
CISZ(4096) -
INDEXED -
NOREUSE -
KEYS(128 16)) -
DATA -
(NAME(ICDC.ZDW.DDW1.META.DATA)) -
INDEX -
(NAME(ICDC.ZDW.DDW1.META.INDEX))

REPRO -
INFILE(VSAMINIT) -
OUTDATASET(ICDC.ZDW.DDW1.META)
/*
//SYSPRINT DD SYSOUT=*
//VSAMINIT DD DSNNAME=&&INITDATA,DISP=(OLD,DELETE)
//*
//

```

2.6 DB2 binds for the metadata table creation utility

Prior to creating the DB2 metadata tables, you need to create a z/OS user ID that the CDC agent started task will run under. Pick a name for this user ID at this point because the DB2 metadata tables must have a schema (owner) name that matches the user ID that the started task will run under. When defining this user ID, make sure to define an OMVS segment.

This user ID has to have SYSCTRL authority in DB2. You can grant this privilege using CHCGRNTA in SCHCCNTL, or via any other method of your choosing.

The DB2 metadata tables are created by a utility program. The utility program takes various input values as parameters and creates the DB2 metadata tables as appropriate. Two jobs must be run to create the metadata tables. The first job binds the plans used by the utility program; the second job runs the utility program.

Sample job CHCMDMBD in SCHCCNTL is used to do the binds for the utility program. Follow the instructions contained within the job. Keep the following things in mind:

- ▶ The parameter <PlanSuffix> must be replaced with a two character suffix. This two character suffix becomes a setting in the SCHCDATA configuration.
- ▶ The parameter <CHCUserID> must be replaced with the z/OS user ID that was created earlier in this process.

Example 2-3 DB2 binds for the metadata creation utility program

```

//CHCMDMBD JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=&SYSUID
//DB2BIND EXEC PGM=IKJEFT01,
//          DYNAMNBR=20
//STEPLIB DD DSN=DB21010.ZDW.SDSNLOAD,DISP=SHR          <==
//SYSTSIN DD *
          DSN SYSTEM(DDW1)                               /* <== */
          BIND -

```

```

PACKAGE(CHCPK101) /* <== */ -
MEMBER(CHCMDMMN) -
ACTION(REPLACE) -
CURRENTDATA(NO) -
ENCODING(UNICODE) -
OWNER(CHCO1) /* <== */ -
QUALIFIER(CHCO1) /* <== */
BIND -
PACKAGE(CHCPK101) /* <== */ -
MEMBER(CHCMDMDB) -
ACTION(REPLACE) -
CURRENTDATA(NO) -
ENCODING(UNICODE) -
OWNER(CHCO1) /* <== */ -
QUALIFIER(CHCO1) /* <== */
BIND -
PACKAGE(CHCPK201) /* <== */ -
MEMBER(CHCDB2DL) -
ACTION(REPLACE) -
CURRENTDATA(NO) -
ENCODING(UNICODE) -
OWNER(CHCO1) /* <== */ -
QUALIFIER(CHCO1) /* <== */
BIND -
PLAN(CHCMDM01) /* <== */ -
PKLIST(CHCPK101.CHCMDMMN -
        CHCPK101.CHCMDMDB -
        CHCPK201.CHCDB2DL -
        ) -
ACTION(REPLACE) RETAIN -
CACHESIZE(256) -
ISOLATION(CS) -
OWNER(CHCO1) /* <== */ -
RELEASE(COMMIT)
END
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DBRMLIB DD DSNAME=ICDC.ZDW.SHCDBRM,DISP=SHR <==
//SYSUDUMP DD SYSOUT=*
/*
//

```

2.7 SCHCDATA

The SCHCDATA library contains configuration settings for the CDC agents. A single SCHCDATA library can contain multiple sets of configurations. Each configuration member has a two character suffix. The JCL for the CDC agent started task contains a parameter that specifies the two character suffix that should be used when reading SCHCDATA members. For the initial configuration, we made a copy of all of the members, using suffix 'F1', indicating a functional test configuration. Our initial configuration is described in the following sections.

2.7.1 CHCCFGxx

The CHCCFGxx configuration members specify general configuration settings. The CHCCFGxx settings must start with the CONFIG keyword. Our initial CHCCFGxx configuration was:

```
CONFIG REPSTATSINTERVAL=5
```

The default value for REPSTATSINTERVAL is 0. By setting REPSTATSINTERVAL to 5, messages will be generated in the JES job log every 5 minutes, indicating the position of the DB2 log reader process. This parameter can be helpful during initial installation verification for CDC agents that will act as sources because it allows you to track the progress of the CDC agent as it reads through the DB2 log. Here is an example of a message generated by enabling REPSTATSINTERVAL:

```
Latest scraped data was written at 2009-07-09-17.15.52, sent 1 changes, current  
Log Position is X'0000001485813133'
```

2.7.2 CHCCMMxx

The CHCCMMxx configuration members specify configuration settings related to TCP/IP communications. The CHCCMMxx settings must start with the TCP/IP keyword. Our initial CHCCMMxx configuration was:

```
TCP/IP SERVICENAME=11001
```

SERVICENAME is a required parameter that specifies either a symbolic service name that has been defined to the communications server or a numeric port number. This defines the TCP/IP port that the CDC agent will listen on.

2.7.3 CHCDBMxx

The CHCDBMxx configuration members specify configuration settings related to DB2. The CHCDBMxx settings must start with the DB2 keyword. Our initial CHCDBMxx configuration is shown in Example 2-4.

Example 2-4 CHCDBM initial configuration

```
DB2 SSID=DDW1,  
    PLANSUFFIX=01,  
    SQLWHEREROWSEL=YES,  
    LOGPOLLINTERVAL=(5,ALWAYS)
```

SSID is a required parameter that specifies the name of the DB2 subsystem or data sharing group to which the CDC agent is to attach. If InfoSphere Change Data Capture is being installed against a DB2 data sharing group, this should be set to the group name so that the CDC agent can attach to any of the group members.

PLANSUFFIX is an optional parameter. If not specified, PLANSUFFIX defaults to the xx suffix of the CHCDBMxx member in use.

SQLWHEREROWSEL is an optional parameter. If set to YES, InfoSphere Change Data Capture can use WHERE clauses to limit the data that is SELECTed from source tables during initial loads or refreshes of target tables. The default value for this parameter is NO.

LOGPOLLINTERVAL is a very significant parameter, which we explain in detail in the section about tuning InfoSphere Change Data Capture. The recommendation for initial configuration is to set LOGPOLLINTERVAL to (5,ALWAYS) in non-data-sharing or 5 in data-sharing environments for agents that will act as sources.

2.7.4 CHCLDRxx

The CHCLDRxx configuration members specify configuration settings related to using the DB2 LOAD utility for refresh operation. By default, this feature is disabled, and it is not necessary to edit the contents of this configuration member during initial configuration. We discuss the contents of this member as part of the discussion about tuning InfoSphere Change Data Capture.

2.7.5 CHCUCSF1

In the past CHCUCSxx was used for Unicode Conversion Services configuration. In InfoSphere Change Data Capture V6.5, Unicode Conversion Services configuration is automatic. This configuration member does not need to be edited.

2.8 DB2 metadata tables

Sample job CHCMDMUT in SCHCCNTL is used to run the utility program to create the DB2 metadata tables. This utility program creates one database and two tablespaces. Follow the instructions that are contained within the job, and also make the following configuration changes:

- ▶ If you are going to use an existing STOGROUP, you still need to specify a catalog name parameter <CHCCatlg>. The value you specify will not get used.
- ▶ If you are going to use an existing STOGROUP, you do not have to change the <CHCVOLSER> parameter.
- ▶ The parameter <CHCUserID> must be replaced with the z/OS user ID the CDC started task will run under.
- ▶ Edit the DSNs on the STEPLIB and CHCCNTRL DDs to point to the correct SCHCLOAD and SCHCDATA libraries.
- ▶ Edit the two SYSMDUMP DDs to use a DSN that is specific to this CDC instance.

Example 2-5 DB2 metadata table creation

```
//CHCMDMUT JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=&SYSUID
//DELETE EXEC PGM=IEFBR14
//SYSMDUMP DD DSNAME=ICDC.ZDW.DDW1.SYSMDUMP,DISP=(MOD,DELETE),
//          UNIT=SYSALLDA,
//          SPACE=(CYL,(1))
//*
//IEFPROC EXEC PGM=CHCMIT,
//          PARM=('CHCMDMUT',
//              '/TRAP(OFF),TER(UAIMM)',
//              '/CONFIG=F1'),
//          REGION=0M
```

```

//STEPLIB DD DSNAME=ICDC.ZDW.SCHCLOAD,DISP=SHR
//          DD DSNAME=DB21010.ZDW.SDSNEXIT,DISP=SHR
//          DD DSNAME=DB21010.ZDW.SDSNLOAD,DISP=SHR
//CHCCTRL DD DDNAME=IEFRDER
//IEFRDER DD DSNAME=ICDC.ZDW.SCHCDATA,DISP=SHR
//SYSMDUMP DD DSNAME=ICDC.ZDW.DDW1.SYSMDUMP,DISP=(MOD,CATLG),
//          UNIT=SYSALLDA,
//          SPACE=(CYL,(100,25))
//ABNLIGNR DD DUMMY
//SYSIN DD *
* Lines with an '*' in column 1 are treated as comments.
*
* Subsystem or data sharing group name of the
* DB2 system which holds (or will hold) the meta-data.
SUBSYS=DDW1
* The plan suffix used when the plan CHCMDMxx was bound.
PLANSFX=01
* The name of the DB2 storage group which should be used
* to create any meta-data objects. This storage group
* will be created if it does not exist.
STOGROUP=SYSDEFLT
* The list of volume names to be used to create the
* storage group if it does not exist. Must be specified
* if the storage group does not exist.
VOLUMES=<CHCVOLSER>
* The name of the catalog to use in creating the
* storage group if it does not exist. Must be specified
* if the storage group does not exist.
CATLG=CATALOG
* The name of the database which holds the meta-data.
* This database will be created if it does not exist.
DBNAME=CDCDB
* The name of the table space which will contain the majority
* of the meta-data tables. This tablespace should be a Unicode
* table space with a page size of 4K. This table space will
* be created if it does not exist.
TBSP01=CDCTS1
* The name of the table space which will contain meta-data tables
* which require a 32K page size. This tablespace should be a
* Unicode table space with a page size of 32K. This table space
* will be created if it does not exist.
TBSP02=CDCTS2
* The user name which owns the meta-data.
OWNER=CHC01
* The SQLID this job should operate under. By default, this
* is the user ID this job is executing under.
SQLID=RES01
* When set to Y, this option will cause all existing
* meta-data tables to be dropped before this utility
* starts its regular processing.
DROPTBL=N
* When set to Y, this option will cause all DB2 system
* catalog tables indexes to be created. The storage group
* used for the indexes will be the storage group specified

```



```
CRSYSIND=N
/*
//SYSPRINT DD SYSOUT=*
//*
//
```

2.9 Bind DB2 plans

Sample job CHCBNDPL can be used to bind the plans needed by the DB2 z/OS CDC agent. When editing this job, keep the following considerations in mind:

- ▶ The parameter <PlanSuffix> must be replaced with a two character suffix. This two character suffix should match the setting of PLANSUFFIX in CHCDBMxx.
- ▶ When doing an initial configuration, you will not have any user exit DBRMs that must be included in any of the plans.
- ▶ The parameter <CHCUserID> must be replaced with z/OS user ID that the CDC started task will run under.
- ▶ If you get a return code of 4 because of missing EXECUTE privileges, run the following SQL statements against DB2, replacing xx with the plan suffix, and then rerun CHCBNDPL:

```
GRANT ALL ON PACKAGE CHCPK1xx.* TO chcuserid;
GRANT ALL ON PACKAGE CHCPK2xx.* TO chcuserid;
```

2.10 The started task

A sample started task is provided in member CHCPROC in the SCHCCNTL library. Keep the following considerations in mind when modifying CHCPROC:

- ▶ Replace <ConfigSuffix> with the xx suffix that matches the SCHCDATA configuration you want the started task to use.
- ▶ Edit the STEPLIB DD to point to the correct SCHCLOAD library.
- ▶ The IEFORDER DD should point to the SCHCDATA library containing the appropriate configuration members.
- ▶ The data set referred to by the SYSDUMP DD in the DELETE and IEFPROC job steps cannot be shared between multiple InfoSphere Change Data Capture started tasks.
- ▶ The CHCMTDTA DD must point to the VSAM metadata cluster that was defined.
- ▶ The CHCPALOG DD must point to the PAL cluster that was defined.
- ▶ For the initial configuration, the CHCCACHE and CHCCHCTL DDs can be set to DUMMY. Just comment out the CHCCACHE and CHCCHCTL DDs provided in case the caching feature must be turned on at a later time.
- ▶ The CHCUXLIB DD should point to the user exit library that was allocated.
- ▶ Verify that the SYSTCPD DD is correct.

Our initial started task is shown in Example 2-6 on page 16.

Example 2-6 Initial InfoSphere Change Data Capture started task

```
//CHCPROC1 PROC CONFIG=P1,
//          COMM=COMM,
//          DBMS=DBMS
//*
//DELETE   EXEC PGM=IEFBR14
//SYSMDUMP DD DSN=ICDC.ZDW.DDW1.SYSMDUMP,DISP=(MOD,DELETE),
//          UNIT=SYSALLDA,
//          SPACE=(CYL,(1))
//*
//IEFPROC  EXEC PGM=CHCMIT,
//          PARM=(' /ENVAR(_ICONV_UCS2_PREFIX=CEE)',
//          '&COMM,&DBMS,CONFIG=&CONFIG'),
//          REGION=0M,TIME=1440
//STEPLIB  DD DSN=ICDC.ZDW.SCHCLOAD,DISP=SHR
//          DD DSN=DB21010.ZDW.SDSNEXIT,DISP=SHR
//          DD DSN=DB21010.ZDW.SDSNLOAD,DISP=SHR
//CHCCNTRL DD DDNAME=IEFRDER
//IEFRDER  DD DSN=ICDC.ZDW.DDW1.SCHCDATA,DISP=SHR
//CHCPRI   DD SYSOUT=*
//CHCAUDIT DD SYSOUT=*
//CHCREPRT DD SYSOUT=*
//CHCMTDTA DD DSN=ICDC.ZDW.DDW1.META,DISP=SHR
//CHCPALOG DD DSN=ICDC.ZDW.DDW1.PAL,DISP=SHR
//*CHCCACHE DD DSN=<CACHE.QUALIFIER>.CHCCACHE,DISP=SHR,
//*          AMP=('BUFND=256,ACCBIAS=SW')
//CHCCACHE DD DUMMY
//*CHCCHCTL DD DSN=<CACHE.QUALIFIER>.CHCCHCTL,DISP=SHR
//CHCCHCTL DD DUMMY
//CHCUXLIB DD DSN=ICDC.ZDW.SCHCEXIT,DISP=SHR
//SYSTCPD  DD DSN=TCPIP.TCPDATA(TCPDATA),DISP=SHR
//UTPRINT  DD DUMMY
//* SEE THE NOTE ABOVE ABOUT THE DISPOSITION CODED FOR THIS DATA SET *
//SYSMDUMP DD DSN=ICDC.ZDW.DDW1.SYSMDUMP,DISP=(MOD,CATLG), *NOTE*
//          UNIT=SYSALLDA,
//          SPACE=(CYL,(400,100))
//ABNLIGNR DD DUMMY
//*
```

After the necessary definitions are in place to ensure that the CDC started task will run under the appropriate user ID, it can be started.

2.11 Access Server and Management Console

You are now ready to install the Access Server. The Access Server supports the following 32-bit Windows operating systems:

- ▶ Windows Vista
- ▶ Windows 7
- ▶ Windows Server 2003
- ▶ Windows XP Professional

The Access Server supports the following UNIX and Linux operating systems:

- ▶ HP-UX Version 11.23 or higher (PA-RISC)
- ▶ AIX® Version 5.3 or higher (POWER®)
- ▶ Sun Solaris Version 2.9 or higher (SPARC)
- ▶ Red Hat AS 5 or higher (x86)
- ▶ SLES 10 or higher (x86)

The Access Server requires two consecutive ports. By default, Access Server uses ports 10101 and 10102. When connecting to the Access Server with the Management Console, the user always specifies the lower numbered port of the pair.

During the Access Server installation, also take note of the password that you give to the Admin user ID. When first connecting to Access Server with the Management Console client, you must log on with this user ID and password.

Before running installation verification tests, the Management Console must be installed. The Management Console supports the following 32-bit Windows operating systems:

- ▶ Windows Vista
- ▶ Windows 7
- ▶ Windows Server 2003
- ▶ Windows XP Professional

For more information about installing the Access Server and Management Console refer to the Access Server and Management Console Installation Guide.

2.12 Access Server definitions

The Management Console client is used to define the CDC agents, CDC user IDs, and agent/user ID mappings to Access Server. When connecting to the Access Server for the first time, use the Admin user ID and password that were specified during installation of the Access Server. The server name refers to the machine on which Access Server is running. The port refers to the lower numbered of the two consecutive ports that were assigned to Access Server. The default value for this port is 10101.

2.12.1 CDC agents

All Access Server definitions are created and managed on the Access Manager tab. To define a CDC agent, right-click in the Datastore Management area and select **New Datastore**. For our initial setup we only had one CDC agent because the same DB2 z/OS subsystem was being used as both the source and the target. Our CDC agent was defined as shown in Figure 2-1 on page 18.

New Datastore

Identification

Name: DSNT

Description: DSNT on demomvs.demopkg.ibm.com

Host Name: demomvs.demopkg.ibm.com

Port: 11001

Ping

Properties

Datastore Type: Dual

Platform Type: z/OS

Database Type: DB2 for z/OS

Datastore Version: V6R5M0B10285T0

Connection Parameters...

Multiuser Configuration

You can allow multiple users to configure subscriptions in a datastore. Each unique user must explicitly lock a subscription to edit it.

Require subscriptions to be locked before editing

OK Cancel

Figure 2-1 Defining a CDC agent

Make the appropriate entries in the Identification section. The port number should match the port specified in the CHCCMMxx configuration member. After filling in the name, description, server, and port fields, click **Ping** to test the connection. If the connection is successful, the Properties fields will automatically be populated.

2.12.2 CDC user IDs

Define a CDC user ID, again starting from the Access Manager tab. Right-click in the User Management area, and select **New User**. Provide a user name and password. This password is used only for this CDC user to connect to the Access Server via the Management Console. Figure 2-2 on page 19 is an example of creating a CDC user ID.

Figure 2-2 Creating a CDC user ID

CDC user roles

There are several roles that can be assigned to the user ID. The role controls the authorities associated with the user ID. Roles and their authorities are defined in this section.

System Administrator

Users assigned to this role can perform all available operations in Management Console. Only super users should be assigned to this role because they will have complete access to the monitoring and configuration perspectives in Management Console, including the system parameters.

Administrator

Users assigned to this role can perform all available operations in Management Console but cannot modify InfoSphere CDC system parameters. The ability to change system parameters through Management Console is applicable only to non-DB2 z/OS CDC agents.

Operator

Users assigned to this role have access to only the monitoring perspective in Management Console. Operators have permissions to do the following:

- ▶ Start, stop, and monitor replication, but *not* configure replication.
- ▶ View the tables selected for refresh and start a refresh, but *not* select or remove tables for refresh.
- ▶ Access to the topology view.
- ▶ View notifications for subscriptions and datastores.

Monitor

Users assigned to this role only have access to the monitoring perspective in Management Console and have permissions to do the following:

- ▶ View the event log, statistics, and table mappings.

- ▶ View replication state and status and latency threshold information.
- ▶ Access to the topology view.

Users with the Monitor role do not have the ability to start and stop replication, configure replication, refresh tables, or view notifications for subscriptions and datastores.

In addition to these roles, the “Enable user account and datastore administration” check box is used to give a CDC user ID the authority to create, edit, and delete Access Server definitions, including CDC agents, CDC user IDs, and agent/user ID mappings.

2.12.3 Agent/user ID mappings

Before the new CDC user ID is able to work with the CDC agent, an agent/user ID mapping must be created. The agent/user ID mapping gives the CDC user ID access to the CDC agent, and also specifies the local OS user ID and password that should be used when the CDC user ID in question is communicating with the CDC agent.

To create this mapping, left-click the CDC user ID in the User Management area. Right-click in the Connection Management area and select **Assign Datastore**. Select the appropriate CDC agent. We created the mapping shown in Figure 2-3.

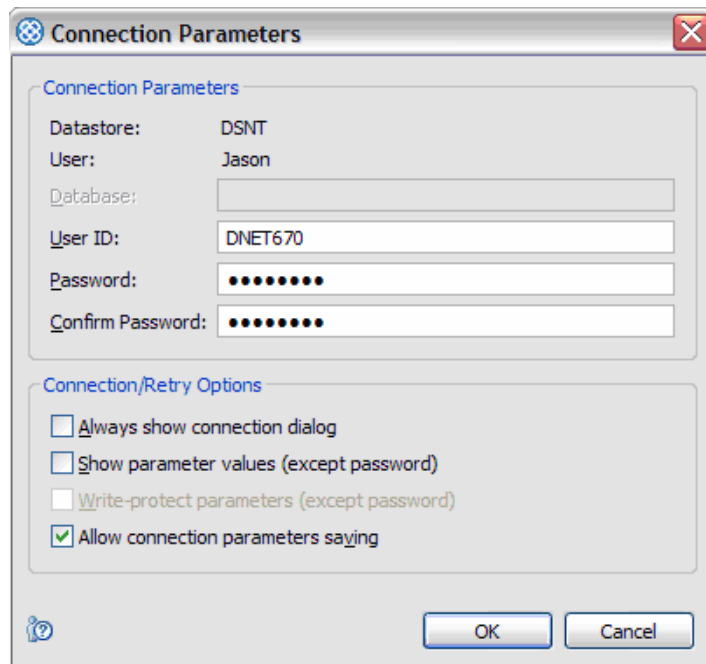


Figure 2-3 Creating an agent/user ID mapping

2.13 Verifying the InfoSphere Change Data Capture installation

This section describes the simple test we performed using the DB2 sample tables to verify that all of the steps up to this point had been done successfully. This type of test can be done with any table in your environment; we suggest that you do this initial test with a table with a very small number of rows so that data synchronization can be visually verified.

2.13.1 Creating the subscription

The Admin user ID was not given access to the CDC agent, so you must log out and log back in with the new CDC user ID before creating the subscription. Do this by selecting **File** → **Access Server** → **Disconnect** from the menu bar, then selecting the **Connect** option to reconnect as a different CDC user.

Create a new subscription. All data synchronization definitions must belong to a subscription. A subscription corresponds to a thread at the CDC agent level. A subscription is also associated with a specific source agent and a specific target agent. Additionally, a single table can be used as a source only one time within any subscription.

To test the functionality of the CDC agent, create a new subscription using the same CDC agent as both the source and target. Click the Configuration tab and the Subscriptions sub-tab. Right-click in the Subscriptions area and select **New Subscription**. The resulting window is shown in Figure 2-4. Provide a name and description for the new subscription and select the source and target from the drop-down lists. These lists are populated based on the datastores that were defined previously.

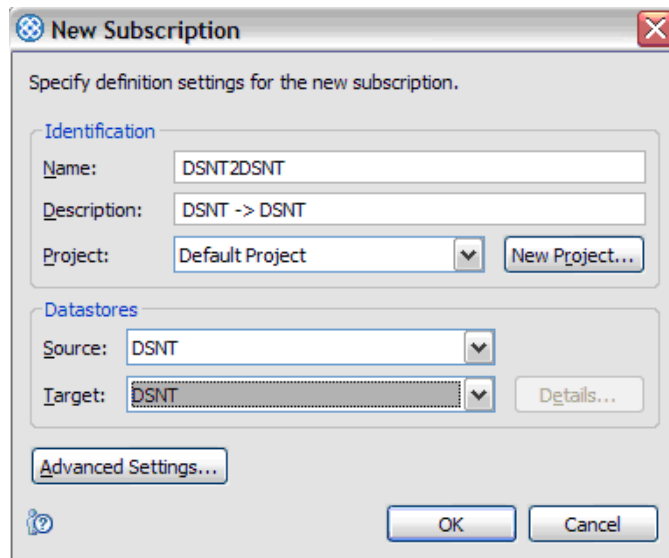


Figure 2-4 Creating a new subscription

2.13.2 Creating the table mapping

The next step is to create the table mapping. Management Console can create the target table for you, but it requires that the target tablespace already exists. For our purposes, we manually created the target table before creating the table mapping in Management Console. Initially, we did not have any data in the target table.

With the new subscription highlighted, right-click in the Table Mappings area and select **Map Tables**. On the next panel (Figure 2-5 on page 22), select “One table mapping of any type” in the Custom mapping area and select the type “Standard.” A standard table mapping means that an insert at the source generates a corresponding insert at the target. An update at the source generates an update at the target. A delete at the source generates a delete at the target. The Automatic mapping types (the first two radio buttons) are used for creating a large number of table mappings, whereas the Custom mapping type option is used to create a single table mapping.

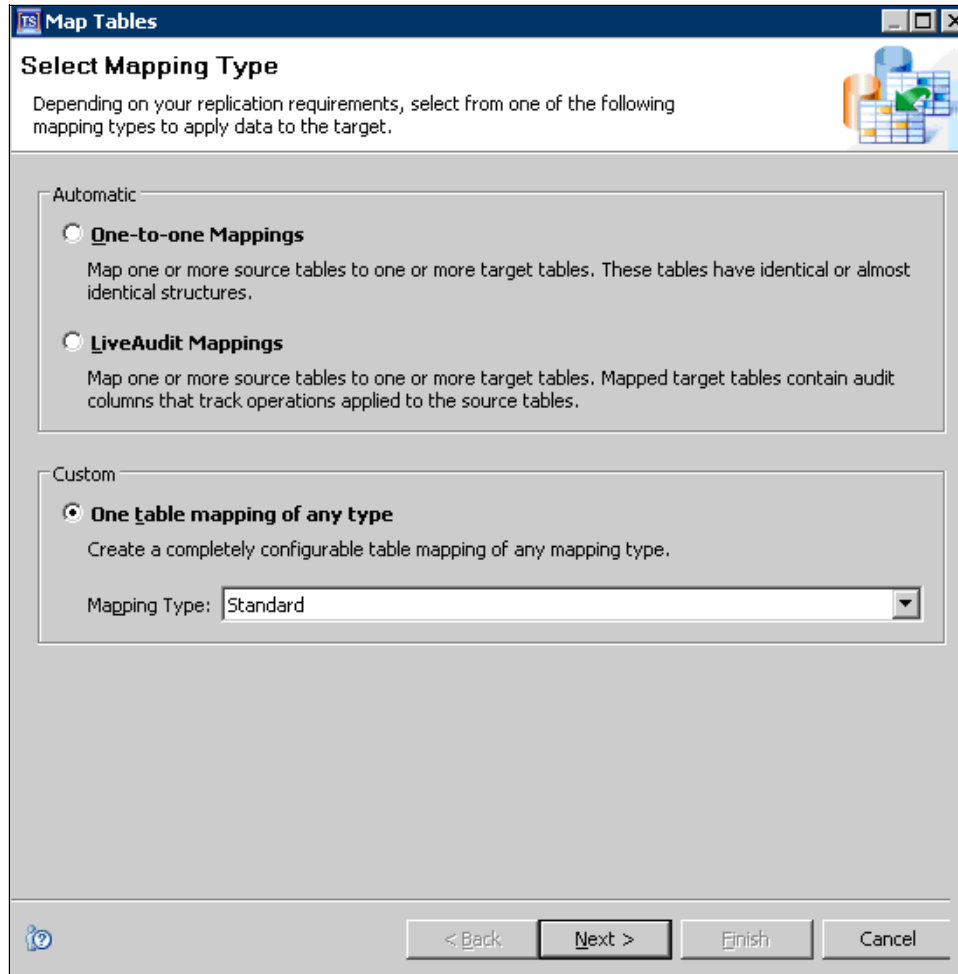


Figure 2-5 Creating a new table mapping

Click **Next**. The next panel (Figure 2-6 on page 23) prompts you to select the source table for data synchronization. The first level in the list represents database names. Expanding a database provides a list of schemas. Expanding a schema provides a list of tables.

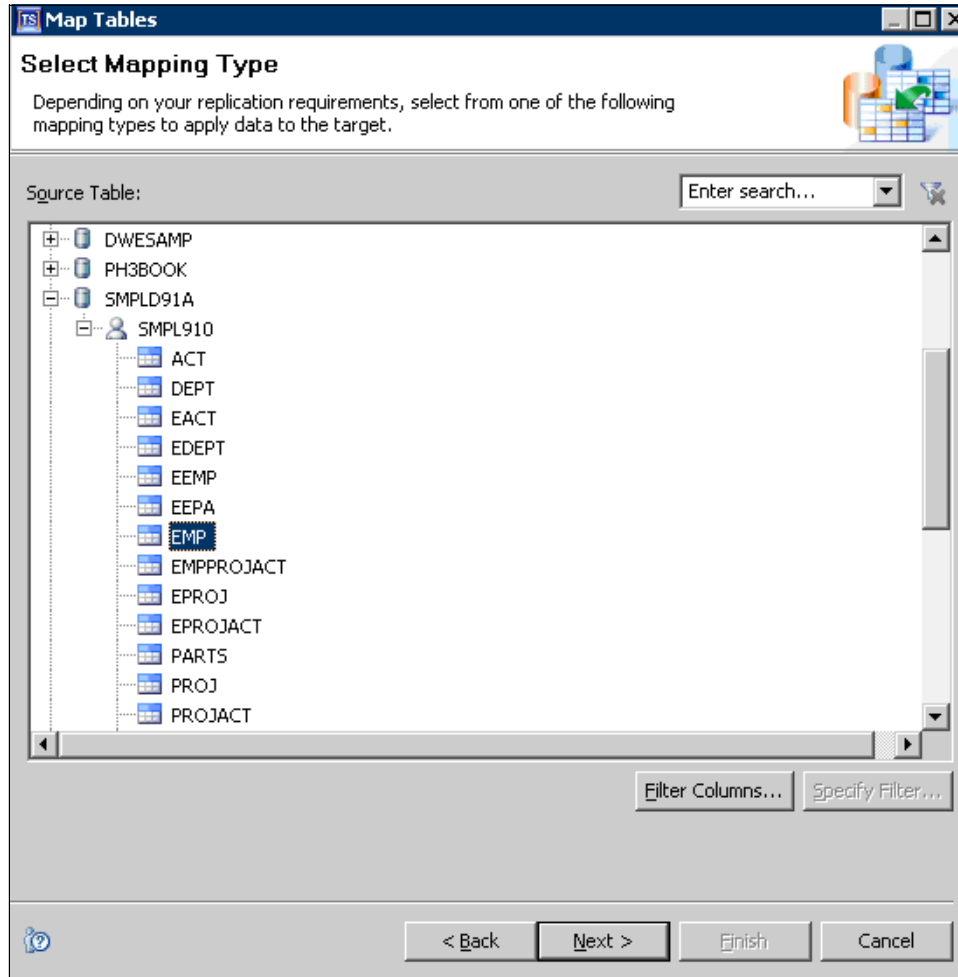


Figure 2-6 Selecting the source table

The next panel prompts you to select the target table in the same manner. If the target table does not already exist, click **Create Table** to create it.

Set the replication method on the next panel, shown in Figure 2-7 on page 24. **Refresh** means that this table mapping can only be used to do initial loads and full refreshes. It cannot be used to do log-based change capture or near-real-time synchronization. **Mirror** means that the table mapping can be used to do both initial loads and near-real-time synchronization.

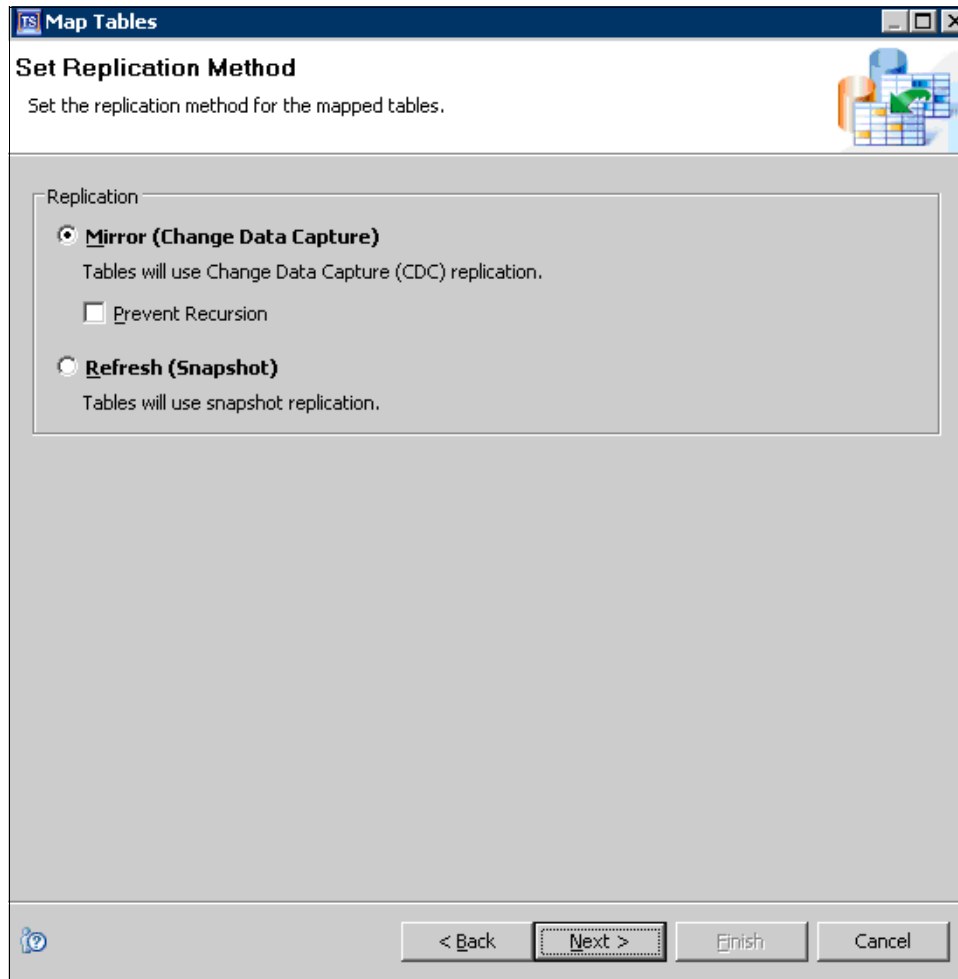


Figure 2-7 Set the replication method

Click **Finish** on the next panel to display the column mapping details. If the source and target tables are identical, InfoSphere Change Data Capture will automatically determine the appropriate column mappings.

2.13.3 Testing refresh

Under the Table Mappings tab, the new table mapping will show a status of “Refresh.” The first test is to attempt an initial load of the target table. By default, InfoSphere Change Data Capture will perform a refresh by reading all of the data from the source table and inserting data into the target table via SQL statements. (As part of tuning, we show how the DB2 LOAD utility can be invoked as an alternative to SQL statements to load data into the target table; see 3.4, “CHCLDRxx” on page 35.)

To test refresh of the subscription, right-click the subscription under the Subscriptions tab, and select **Start Mirroring (Continuous)**. Click the Monitoring tab, and watch the status of the subscription. During refresh, the state will show as “Refresh before mirror.” When the refresh is complete, the state will change to “Mirror continuous.” If the subscription instead changes to an error status, right-click the subscription and select **Show Event Log** to view related error messages.

After the subscription has changed state to Mirror continuous, the target table should look exactly the same as the source table. This can be verified with SPUFI or other query tools.

2.13.4 Testing continuous mirroring

After the refresh completes, InfoSphere Change Data Capture immediately puts the subscription into continuous mirroring. At this point, committed changes to the data in the source table are propagated to the target table in near real time. This can be verified with SPUFI or other query tools.

2.14 Configuring notification settings

InfoSphere Change Data Capture divides all messages into 45 categories based on severity of the message, whether the message is related to source or target, and the aspect of the product that the message is related to. By default, the messages in all of these categories are written to the CHCPRINT DD in the CDC agent started task. InfoSphere Change Data Capture provides the ability for messages to be sent to the syslog or to user exits. It also provides the ability to stop messages from being sent to CHCPRINT. All of this is configured at the message category level.

To change notification settings for a subscription, return to the Configuration tab. Right-click the subscription and select **Notifications**. Select a message category and uncheck the default check box to configure destinations for messages in that category.

Figure 2-8 on page 26 shows that we sent all fatal and error messages to the syslog.

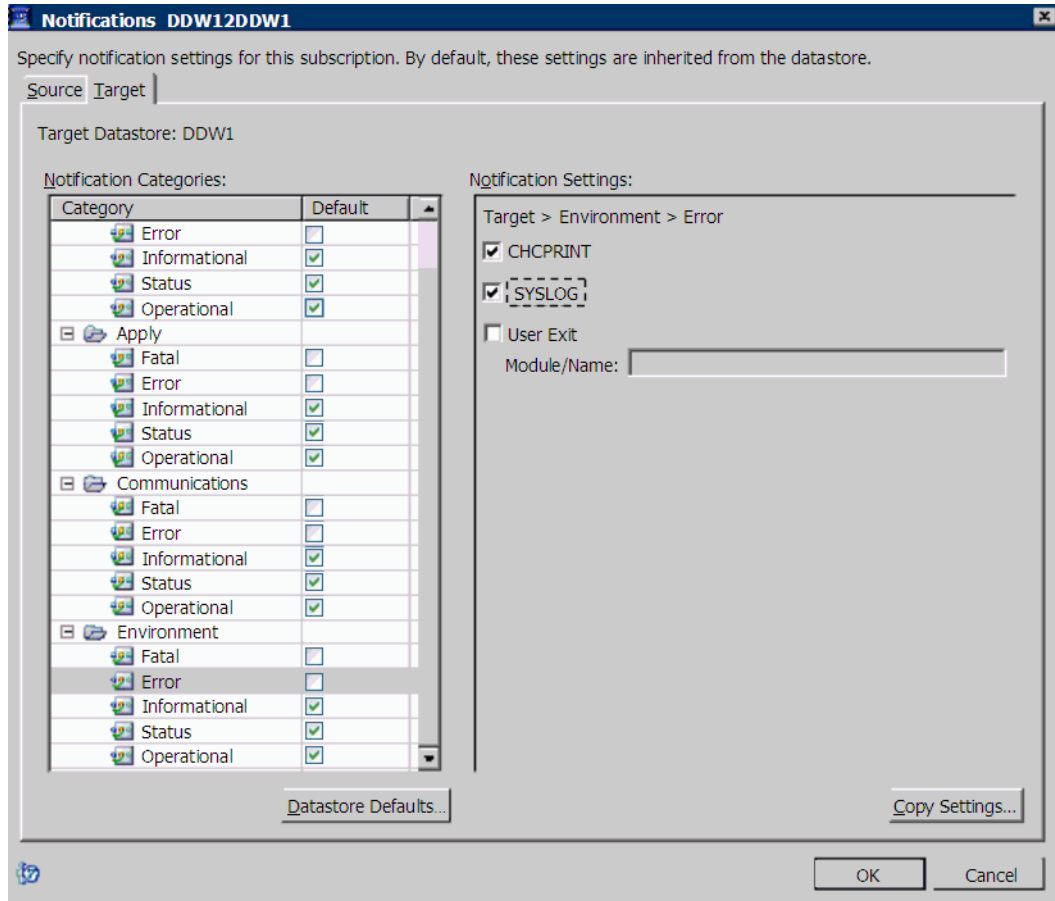


Figure 2-8 Configuring notification settings



Tuning InfoSphere Change Data Capture for z/OS

In the previous chapter, we explained how to create an initial CDC agent configuration. We explained the various SCHCDATA members that control product functionality, and we showed an initial functional test configuration.

In this chapter, we take a closer look at key SCHCDATA parameters. We discuss the impact of these parameters on data latency, throughput, and CPU usage. Finally, we discuss performance monitoring for InfoSphere Change Data Capture. Parameters that already have been set in the initial configuration and do not impact performance or availability are not discussed in this chapter.

For more information about all of the configuration parameters that are allowed in SCHCDATA members, refer to the IBM InfoSphere Change Data Capture Version 6.5 User Documentation.

3.1 CHCCFGxx

InfoSphere Change Data Capture reads general product configuration control statements when it is being initialized. These statements are read from the CHCCFGxx configuration control data set member. The CHCCFGxx member is the first member of the SCHCDATA library that InfoSphere Change Data Capture reads during initialization.

AUTORESTARTINTERVAL

AUTORESTARTINTERVAL works in conjunction with persistent subscriptions. By default, subscriptions do not automatically recover from temporary error conditions, such as a communications failure. Marking a subscription as persistent allows it to automatically recover from these types of conditions. AUTORESTARTINTERVAL controls the interval, in minutes, after which persistent subscriptions will attempt to recover.

The default value is zero, which means that even if a subscription is marked as persistent, no automatic recovery will be attempted. Intervals in the range of 1 - 60 are valid. If you plan to use persistent subscriptions, set AUTORESTARTINTERVAL to a value that is low enough to re-establish the connection in a timely manner, but not flood the system with failed attempts.

HEARTBEATTIMEOUT

HEARTBEATTIMEOUT controls the timeout interval, in minutes, for heartbeat messages between source and target CDC agents. If either CDC agent involved in an active subscription does not receive a heartbeat message within the HEARTBEATTIMEOUT interval, it will be considered a communications failure and the subscription will be stopped. If the subscription is defined as persistent, it will attempt to automatically recover from the communications failure at an interval controlled by AUTORESTARTINTERVAL.

The default value is 15, indicating a timeout interval of 15 minutes. Intervals in the range of 3 - 999 minutes are valid. A value of zero will disable the heartbeat feature. The recommendation is to use a common value for HEARTBEATTIMEOUT across all CDC agents.

PALCLEANUPTIME

PALCLEANUPTIME specifies the time of day that product administration log (PAL) data will be cleaned up. The PAL VSAM cluster holds the data which is viewed through the Management Console event log viewer. The cleanup occurs on a daily basis. This parameter works in conjunction with PALRETPD.

If this parameter is not explicitly specified, no daily PAL cleanup will be performed. Values in the range 00:00 - 23:59 are valid. The recommendation is to provide a value for this parameter.

PALRETPD

PALRETPD is a number that controls what PAL data should be deleted when a daily PAL cleanup occurs. Messages older than the number of days specified on the PALRETPD parameter will be deleted.

This parameter defaults to a value of 14, indicating that data older than 14 days should be deleted. Values in the range 1 - 9999 are valid.

REPSTATSINTERVAL

In the initial configuration, we recommended that REPSTATSINTERVAL be set to a value of 5 for CDC agents that will act as sources. Setting REPSTATSINTERVAL to 5 caused the

CDC agent to generate messages every five minutes indicating the log scraper task's position in the DB2 log. Values in the range 5 - 1440 are valid.

Although minimal, REPSTATSINTERVAL does cause some additional overhead. Setting REPSTATSINTERVAL to zero will disable messages reporting the position of the log scraper task in the DB2 log. REPSTATSINTERVAL has no effect on data synchronization. It strictly controls the frequency of the messages indicating the log scraper's position.

3.1.1 CHCCFGP1

We created a new set of SCHCDATA configuration members for our tuned configuration. We used a suffix of P1 for this new configuration. To activate this configuration, the CDC agent started task was stopped and restarted using CONFIG=P1.

Our CHCCFGP1 configuration member is shown in Example 3-1.

Example 3-1 CHCCFG final configuration

```
CONFIG REPSTATSINTERVAL=0,  
AUTORESTARTINTERVAL=1,  
HEARTBEATTIMEOUT=3,  
PALCLEANUPTIME=00:00
```

3.2 CHCCMMxx

InfoSphere Change Data Capture reads communications configuration control statements when communications are being initialized. These statements are read from the CHCCMMxx configuration control data set member.

CTRLQUEUE SIZE

CTRLQUEUE SIZE controls the size of the queue for the control communications link. The control communications link is used for message transfers that coordinate the replication of data between CDC agents.

The default value for CTRLQUEUE SIZE is 64 K. Valid values are in the range 64 K - 1 M. The default value is sufficient except in cases where there are a large number of subscriptions between the same pair of CDC agents.

DATAQUEUE SIZE

DATAQUEUE SIZE controls the size of the queue for the data communications link. The data communications link is used for the transmission of replicated data between CDC agents.

The default value for DATAQUEUE SIZE is 1 M. Valid values are in the range 64 K - 1024 M. The appropriate settings for CTRLQUEUE SIZE and DATAQUEUE SIZE depend on the data being replicated. DATAQUEUE SIZE can have a significant impact on virtual storage allocations because each subscription will allocate two data queues. Start with the default value and adjust upwards until latency goals are met or further adjustments no longer improve latency.

3.2.1 CHCCMMP1

Our final CHCCMMxx configuration is shown in Example 3-2 on page 30.

Example 3-2 CHCCMM final configuration

```
TCP/IP SERVICENAME=11001,  
DATAQUEUESIZE=8M,  
CTRLQUEUESIZE=64K
```

3.3 CHCDBMxx

InfoSphere Change Data Capture reads DBMS configuration control statements when DBMS repositories are being initialized. These statements are read from the CHCDBMxx configuration control data set member.

Many of the CHCDBMxx parameters are related to tuning of the log read process. For the parameters to make sense, you need to understand how InfoSphere Change Data Capture reads information from the DB2 log.

During initialization, InfoSphere Change Data Capture issues an IFI command to activate a DB2 MONITOR trace (IFCID 126) using an available OPx buffer as the target. As new data is written to the DB2 log, more trace data gets written to the OPx buffer. When a certain amount of data is available in the OPx buffer, DB2 notifies the InfoSphere Change Data Capture started task via the POST service. The amount of data required to be written to the OPx buffer before InfoSphere Change Data Capture is notified is controlled by the BUFSIZE and BUFTHRESHOLD parameters.

When InfoSphere Change Data Capture receives such a notification, it reads the new DB2 log records one at a time via IFI. When it hits the current end of the DB2 log, it waits for the next notification from DB2 that the OPx buffer threshold has been reached.

By default, one instance of the log read process will occur for every CDC subscription that specifies the DB2 z/OS subsystem or data sharing group as the source. If you are going to create multiple subscriptions that have the same DB2 z/OS subsystem or data sharing group as the source, the recommendation is that you enable caching of the data that comes from the log read process. With caching turned on, there is one log read process, which fills the cache. From there, each subscription locates the data it needs from the cache instead of issuing new IFI commands. Caching is controlled by the CACHEBLOCKISZE, CACHELEVEL1RESERVED, and CACHELEVEL1SIZE parameters.

Once the subscriptions have the data they need from the DB2 log, they communicate with the appropriate target CDC agent. The target CDC agent, in turn, applies the changes to the target tables by serially issuing the appropriate SQL statements. When the target CDC agent issues a commit, it stores the last source DBMS log position (RBA or LRSN) associated with work that has been committed at the target. This log position is known as a *bookmark*, and is used as a restart point for the source log read process if InfoSphere Change Data Capture is stopped or an error occurs.

BUFSIZE

BUFSIZE controls the size of the DB2 OPx buffer that is used to hold data from the MONITOR trace that InfoSphere Change Data Capture starts.

This parameter works in conjunction with BUFTHRESHOLD to control asynchronous notification that new DB2 log data is available. The default setting is 132, which indicates a buffer size of 132 kilobytes. Valid values are in the range 8 - 1024.

BUFTHRESHOLD

BUFTHRESHOLD controls the percentage of the OPx buffer that must be full before InfoSphere Change Data Capture is notified that there is new DB2 log data to read.

The default is 1, which indicates 1%. When used in conjunction with the default BUFSIZE value, this indicates that 1.32 kilobytes of MONITOR trace data must be written to the OPx buffer before InfoSphere Change Data Capture will wake up and start issuing new IFI reads. Valid values are in the range 1 - 99.

It is important to note that when there is continuous DB2 activity, BUFSIZE and BUFTHRESHOLD have no impact. Once InfoSphere Change Data Capture starts issuing IFI reads, it will continue to do so until it hits the current end of log. After hitting end of log, BUFSIZE and BUFTHRESHOLD control the amount of new log data that must exist before InfoSphere Change Data Capture will start issuing new IFI reads.

In an environment without constant DB2 activity, BUFSIZE and BUFTHRESHOLD have a significant impact on CPU usage. When InfoSphere Change Data Capture is woken up more frequently to issue new IFI reads, more CPU will be used. So, although higher settings for BUFSIZE and BUFTHRESHOLD generally lead to lower CPU usage in such situations, they result in higher latency because InfoSphere Change Data Capture will not start issuing new IFI reads until more MONITOR trace data has been written.

When adjusting BUFSIZE and BUFTHRESHOLD, the system log should be monitored for the following message:

```
DSNW133I + DSNWVOPX - TRACE DATA LOST, <OPx> NOT ACCESSIBLE RC=08
```

It is important to understand that this message does not indicate that DB2 log data has been lost. Even when this occurs, InfoSphere Change Data Capture will successfully replicate all changed data. However, this message does indicate that InfoSphere Change Data Capture is unable to clear data out of the OPx buffer at the rate that new data is coming in. If this occurs, there are three potential solutions:

- ▶ Increase BUFSIZE
- ▶ Decrease BUFTHRESHOLD
- ▶ Increase the priority of the InfoSphere Change Data Capture started task

CACHEBLOCKSIZE

CACHEBLOCKSIZE, CACHELEVEL1RESERVED, and CACHELEVEL1SIZE must all be set before caching will be enabled. Additionally, the level 2 cache must be created.

CACHEBLOCKSIZE controls the size of the blocks of data that are moved between the level 1 and level 2 caches. The level 1 cache is an in-memory cache. The level 2 cache is a VSAM data set. The size of the level 1 cache must be a multiple of the block size. The size of the level 1 and level 2 caches is significant. If the data that a subscription is looking for is found in the level 1 cache, no disk I/O is required. If the data is instead found in the level 2 cache, the data will be retrieved from the level 2 VSAM cluster. If the data is not in either the level 1 or level 2 caches, a new synchronous log read process will be started to issue IFI reads against DB2. This synchronous log read process will continue until the data that the subscription needs is located in one of the caches.

CACHEBLOCKSIZE is specified as an integer value and indicates the block size in megabytes. A block size of one or two megabytes is sufficient in most cases.

CACHELEVEL1RESERVED

CACHELEVEL1RESERVED specifies the size, in megabytes, of the portion of the level 1 cache which is reserved for holding recent DB2 log data. Typically, 80% of the level 1 cache should be reserved. Additionally, the following restrictions must be met:

- ▶ The reserved portion of the level 1 cache must be at least 5 megabytes.
- ▶ The reserved portion of the level 1 cache must be an integer multiple of the cache block size (CACHEBLOCKSIZE).
- ▶ There must be at least one block in the level 1 cache that is not reserved. See CACHELEVEL1SIZE.

CACHELEVEL1SIZE

CACHELEVEL1SIZE controls the size, in megabytes, of the level 1 cache. CACHELEVEL1SIZE must be a multiple of CACHEBLOCKSIZE. Additionally, it must be at least one block larger than CACHELEVEL1RESERVED. The level 1 cache must meet the following criteria:

- ▶ Be at least 10 megabytes.
- ▶ Consist of at least 2 blocks. See CACHEBLOCKSIZE.
- ▶ Be an exact multiple of the block size.

The level 1 cache is an in-memory cache. This storage will be allocated by the InfoSphere Change Data Capture started task. It should be sized to minimize paging.

Once CACHEBLOCKSIZE, CACHELEVEL1RESERVED, and CACHELEVEL1SIZE are set, the level 2 cache must be defined and allocated to the InfoSphere Change Data Capture started task.

To define the level 2 cache, edit and submit job CHCCRCCH in the SCHCCNTL library. Note that the level 2 cache must be at least one block larger than the level 1 cache. Our level 2 cache was defined as shown in Example 3-3.

Example 3-3 Defining the level 2 cache

```
//CHCCRCCH JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//PURGE EXEC PGM=IEFBR14
//CHCCHCTL DD DSN=ICDC.ZDW.DDW1.L2CACHE.CHCCHCTL,
//          DISP=(MOD,DELETE),SPACE=(1024,1),
//          DCB=(LRECL=1024,RECFM=FB)
//*
//CREATE EXEC PGM=IEFBR14,COND=(12,LT)
//CHCCHCTL DD DSN=ICDC.ZDW.DDW1.L2CACHE.CHCCHCTL,DISP=(NEW,CATLG),
//          SPACE=(1024,1),DCB=(LRECL=1024,RECFM=FB)
//*
//DEFMTDTA EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE -
  ICDC.ZDW.DDW1.L2CACHE.CHCCACHE -
  CLUSTER -
  PURGE

SET MAXCC = 0
SET LASTCC = 0
```

```

DEFINE -
  CLUSTER -
    (NAME(ICDC.ZDW.DDW1.L2CACHE.CHCCACHE) -
     VOL(DWDBS1) -
     RECORDSIZE(32760 32760) -
     NUMBERED -
     SHAREOPTIONS(2))-
  DATA -
    (NAME(ICDC.ZDW.DDW1.L2CACHE.CHCCACHE.DA) -
     BUFFERSPACE(8388608) -
     CYLINDERS(250))
/*
//

```

To allocate the level 2 cache to the InfoSphere Change Data Capture started task, we added the DDs shown in Example 3-4.

Example 3-4 DDs required for caching

```

//CHCCACHE DD DSNAME=ICDC.ZDW.DDW1.L2CACHE.CHCCACHE,DISP=SHR,
//          AMP=('BUFND=256,ACCBIAS=SW')
//CHCCHCTL DD DSNAME=ICDC.ZDW.DDW1.L2CACHE.CHCCHCTL,DISP=SHR

```

COMMITFREQ

COMMITFREQ controls the frequency at which the target CDC agent commits changes to the target tables. This parameter is specified as a triplet of values. The first value is the number of changes that can be applied to the target tables before a commit is generated. The second value is the number of seconds that are allowed since the previous commit before another commit is generated. The third parameter is the number of commits that are allowed at the source before a new target commit is generated.

The target CDC agent will issue a commit upon receipt of the next notification of a source commit after any one of those three conditions is satisfied. Transactional integrity is always maintained. The three conditions are tracked at the subscription level. In other words, the number of changes applied, the number of seconds since the previous commit, and the number of source commits are all tracked separately for each subscription. A commit will be executed for the changes for a single subscription when one of the conditions is met for that subscription.

Decreasing the frequency at which the target CDC agent issues commits can significantly increase throughput and decrease CPU usage, at the risk of increasing the likelihood of locking conflicts and increasing replication latency. It is important to understand that InfoSphere Change Data Capture will never apply changes that have not been committed at the source. The target CDC agent only sees changes which have already been committed at the source.

The third parameter is of particular importance when the DB2 changes being captured consist of small units of work, like online activity. Likewise, large DB2 units of work where only a small subset of the data is being replicated become small units of work when applied by the target CDC agent. The third parameter can be used to group these small units of work into a larger unit of work, thus greatly increasing throughput.

The default value for the COMMITFREQ parameter is (200,30,1).

LOGPOLLINTERVAL

LOGPOLLINTERVAL is used to force InfoSphere Change Data Capture to wake up and issue a new IFI read even if the OPx buffer threshold, as controlled by BUFSIZE and BUFTHRESHOLD, has not yet been reached. If new DB2 log data is not yet available, the IFI read will block until new data becomes available.

There are two distinct uses for LOGPOLLINTERVAL. The first usage is to account for non-local DB2 activity in a data sharing group. In a data sharing group, only activity on the local DB2 log causes data to be written to the MONITOR trace, which in turn causes InfoSphere Change Data Capture to wake up and issue new IFI reads. If the local DB2 member becomes dormant, LOGPOLLINTERVAL can be used as a way to assure that InfoSphere Change Data Capture continues to process changes occurring on the other members.

The other usage of LOGPOLLINTERVAL is applicable in both data sharing and non data sharing. Consider the situation where new trace data has been written to the OPx buffer, but the threshold has not yet been reached. If, at this time, DB2 activity ceases for a period of time, the last DB2 changes will not replicate until more DB2 activity occurs. LOGPOLLINTERVAL can be used to force a new IFI read even though the OPx buffer threshold has not been met. This is generally used to ensure that SLAs are met, even in the event of an extended period of low DB2 activity.

The default value for LOGPOLLINTERVAL is 5, indicating that if the source DB2 subsystem is part of a data sharing group, InfoSphere Change Data Capture should issue an IFI read if it has not issued one for the last 5 seconds. If the source DB2 subsystem is stand-alone, IFI reads will never be issued unless the OPx buffer threshold is reached. To force the LOGPOLLINTERVAL parameter to take effect in non-data sharing, the ALWAYS statement must be used. For example, to force InfoSphere Change Data Capture to issue an IFI read after 5 seconds in non data sharing, you would specify LOGPOLLINTERVAL=(5,ALWAYS).

Setting LOGPOLLINTERVAL to small values can cause increased CPU usage due to the fact that InfoSphere Change Data Capture will wake up to process small amounts of new DB2 log data. Just like with BUFSIZE and BUFTHRESHOLD, LOGPOLLINTERVAL has no effect when there is constant DB2 activity. Typically, LOGPOLLINTERVAL should be set to a value just less than the desired SLA. For example, if the SLA is that data latency will be less than 60 seconds, LOGPOLLINTERVAL should be set to a value just less than 60. This way, the SLA should be met even in the event that DB2 activity stops on the subsystem or local member.

MAXINSTGLOBSIZE

MAXINSTGLOBSIZE specifies the maximum amount of data for a LOB or XML column that InfoSphere CDC for z/OS will hold in storage. For an XML column or a LOB column with a defined size greater than this value, InfoSphere CDC for z/OS will use a LOB locator and not bring more than this amount of data into storage. In this case, only one buffer of the specified size will be required per subscription. For LOB columns with a defined size less than or equal to this value, InfoSphere CDC for z/OS will not use a locator and use one buffer per LOB column. This provides better performance at the cost of more storage.

The default value is 100 kilobytes. Valid values are in the range 32 Kb - 128 Mb.

RECOVERYRETRYLIMIT

RECOVERYRETRYLIMIT controls the number of times InfoSphere Change Data Capture will attempt to re-apply changes to the target tables when a deadlock or timeout occurs. If the retry limit is exceeded, the subscription will be stopped and put into an error state, unless this behavior is overridden with the ENDONERROR CHCDBMxx parameter. The

recommendation is that you set RECOVERYRETRYLIMIT to an appropriate value and do not override the ENDONERROR default.

The default value for RECOVERYRETRYLIMIT is 0. Valid values are in the range 0 - 65535.

RETRYCACHESIZE

RETRYCACHESIZE controls the amount of storage used by each subscription to hold changes which InfoSphere Change Data Capture will re-attempt to apply to the target database due to a deadlock or timeout condition. This parameter only takes effect if RECOVERYRETRYLIMIT is non-zero. This storage is acquired above the bar.

The default value for RETRYCACHESIZE is 1 Gb. Valid values are in the range 1 Mb - 9999 Gb.

USELOADER

USELOADER controls whether or not InfoSphere Change Data Capture will ever use the DB2 LOAD utility to handle a table refresh. By default, USELOADER is set to NO, indicating that all table refreshes will be processed by issuing SQL statements against the target table. To enable use of the DB2 LOAD utility, set USELOADER=YES, and configure the CHCLDRxx configuration member.

3.3.1 CHCDBMP1

For our final CHCDBMxx configuration we enabled caching, set LOGPOLLINTERVAL to meet our target SLAs, enabled use of the DB2 LOAD utility, and decreased COMMIT frequency via the COMMITFREQ parameter. Our final configuration is shown in Example 3-5.

Example 3-5 CHCDBM final configuration

```
DB2 SSID=DDW1,  
PLANSUFFIX=01,  
SQLWHERESESEL=YES,  
LOGPOLLINTERVAL=(60,ALWAYS),  
BUFSIZE=1024,  
RECOVERYRETRYLIMIT=3,  
RETRYCACHESIZE=64M,  
USELOADER=YES,  
CACHELEVEL1RESERVED=30,  
CACHEBLOCKSIZE=2,  
CACHELEVEL1SIZE=38,  
COMMITFREQ=(5000,30,5000)
```

3.4 CHCLDRxx

InfoSphere Change Data Capture reads DB2 LOAD utility configuration control statements before starting to refresh tables. These statements are read from the CHCLDRxx configuration control data set member. The CHCLDRxx configuration member will only be used if USELOADER=YES in the CHCDBMxx configuration member, or if DB2 LOAD utility support has been enabled for a particular subscription via the CHCMTAUT utility.

Setting USELOADER=YES in CHCDBMxx allows table refreshes to use the DB2 LOAD utility instead of refreshing the target table via SQL statements. However, the refresh of the target table will revert to using SQL statements if any of the following conditions are true:

- ▶ There is more than one table in the target table space.
- ▶ InfoSphere Change Data Capture is being used to capture changes on the target table.
- ▶ The total length of all of the columns in the target table exceeds 32,760 bytes minus the number of columns in the table.
- ▶ A user exit has been enabled on this table mapping and the clear table operation has been disabled.
- ▶ The target table has a ROWID column.
- ▶ The table mapping is a LiveAudit™ table mapping.

The parameters in the CHCLDRxx configuration member correspond directly to parameters on the LOAD DATA REPLACE utility control statement that InfoSphere Change Data Capture will build to refresh the target table. For more information, refer to the DB2 z/OS Utility Guide and Reference.

DISCARDS

The DISCARDS parameter corresponds to the DISCARDS keyword in the LOAD DATA control statement. This parameter is optional. If not specified, a value of 0 will be used, indicating that the LOAD should continue regardless of the number of input records which fail to LOAD. Specifying any other integer value will set a maximum for the number of input records which can be discarded before the LOAD abnormally terminates.

DSNDEVT

The DSNDEVT parameter specifies the device type to use when allocating DB2 LOAD utility work data sets. Generic or actual device names can be specified. This parameter is optional. If not specified, SYSDA will be used.

DSNHLQ

The DSNHLQ parameter specifies the high level qualifier used when allocating the DB2 LOAD utility work data sets. The qualifier cannot exceed 11 characters in length. InfoSphere Change Data Capture builds data set names by concatenating the keyword setting, the name of the replication agent, the current date and time, and the default DD name of the data set specified in the DB2 LOAD utility documentation. InfoSphere Change Data Capture requires sufficient authority to allocate, write to, read from, and delete these data sets. This parameter is optional. If not specified, the authorization ID of the InfoSphere Change Data Capture started task will be used.

ENFORCE

The ENFORCE parameter corresponds to the ENFORCE keyword in the LOAD DATA control statement. This parameter is optional. If not specified, CONSTRAINTS will be used, indicating that DB2 will perform constraint checking when doing the LOAD. If ENFORCE=NO, no constraint checking will occur during the LOAD, and the target table will be placed in check pending status following the LOAD if any referential or check constraints exist on the table.

KEEPDICTIONARY

The KEEPDICTIONARY parameter corresponds to the KEEPDICTIONARY keyword in the LOAD DATA control statement. This parameter is optional. If not specified, NO will be used, indicating that DB2 is to build a new compression dictionary if the target table space is compressed. Using KEEPDICTIONARY=YES will force DB2 to reuse the existing compression dictionary.

LOG

The LOG parameter corresponds to the LOG keyword in the LOAD DATA control statement. This keyword is optional. If not specified, YES will be used, indicating that DB2 is to log records that are inserted into the target table as part of the LOAD. Using LOG=NO will cause DB2 to skip logging during the LOAD process. If LOG=NO, upon completion of the LOAD, the target table will be put into copy pending status unless the table is defined as NOT LOGGED (keyword introduced in DB2 9).

PREFORMAT

The PREFORMAT parameter corresponds to the PREFORMAT keyword in the LOAD DATA control statement. This keyword is optional. If not specified, NO will be used, indicating that DB2 is not to preformat remaining pages upon completion of the LOAD. Set PREFORMAT=YES, if you want DB2 to do this preformatting.

REUSE

The REUSE parameter corresponds to the REUSE keyword in the LOAD DATA control statement. This keyword is optional. If not specified, NO will be used, indicating that DB2 should delete and reallocate the underlying VSAM data sets. Using REUSE=YES, will force DB2 to reuse the existing VSAM datasets.

SORTDEVT

The SORTDEVT parameter corresponds to the SORTDEVT keyword in the LOAD DATA control statement. This keyword is optional. If not specified, no SORTDEVT keyword will be included in the LOAD DATA statement. Set SORTDEVT to the name of the device that the DB2 LOAD utility should use when dynamically allocating temporary data sets for DFSORT.

SORTKEYS

The SORTKEYS parameter corresponds to the SORTKEYS keyword in the LOAD DATA control statement. This parameter is optional. If not specified, a value of 0 will be used. SORTKEYS should be set to an integer value that estimates the number of index keys that will need to be sorted during the LOAD. Refer to the DB2 z/OS Utility Guide and Reference for more information about setting SORTKEYS.

SORTNUM

The SORTNUM parameter corresponds to the SORTNUM keyword in the LOAD DATA control statement. This parameter is optional. If not specified, a value of 0 will be used. SORTNUM should be set to an integer value which specified how many data sets DFSORT should dynamically allocate. Refer to the DB2 z/OS Utility Guide and Reference for more information about setting SORTNUM.

SORTOUTSIZE

The SORTOUTSIZE parameter is required and should be set to a pair of values indicating the primary and secondary allocations, in cylinders, for the temporary work file for sort output. For example, SORTOUTSIZE=(100,10).

SYSDISCSIZE

The SYSDISCSIZE parameter is required and should be set to a pair of values indicating the primary and secondary allocations, in cylinders, for the discard data set. The discard data set will hold records that failed to LOAD properly.

SYSERRSIZE

The SYSERRSIZE parameter is required and should be set to a pair of values indicating the primary and secondary allocations, in cylinders, for the error data set. The error data set will hold information about errors that were encountered during the LOAD.

SYSMAPSIZE

The SYSMAPSIZE parameter is required and should be set to a pair of values indicating the primary and secondary allocations, in cylinders, for the SYSMAP data set. The SYSMAP data set is used as a work data set during error processing.

SYSUT1SIZE

The SYSUT1SIZE parameter is required and should be set to a pair of values indicating the primary and secondary allocations, in cylinders, for the sort input work file.

3.4.1 CHCLDRP1

Our final CHCLDRxx configuration is shown in Example 3-6.

Example 3-6 CHCLDR final configuration

```
DB2LOADER LOG=YES,  
    PREFORMAT=NO,  
    KEEPDICTIONARY=NO,  
    REUSE=NO,  
    DSNDEVT=3390,  
    DSNHLQ=ICDC.DDW1,  
    SYSUT1SIZE=(700,70),  
    SORTKEYS=6000000,  
    SORTOUTSIZE=(700,70),  
    SYSERRSIZE=(10,1),  
    SYSMAPSIZE=(100,10),  
    SYSDISCSIZE=(10,1)
```

3.5 Latency and throughput monitoring

Tuning InfoSphere Change Data Capture is an iterative process. The appropriate settings for the various configuration parameters depend on the data, the volume of changes, the transformations, and the SLAs that must be met. Fortunately, Management Console provides latency and throughput monitoring, which can help you determine how much tuning is required and help you assess the impact of parameter changes.

Before you can view latency and throughput information for a subscription, you must enable collection of statistics for that subscription. To enable collection of statistics, click the Monitoring tab in Management Console. Right-click the subscription of interest, and select **Statistics** → **Collect Statistics**. See Figure 3-1 on page 39.

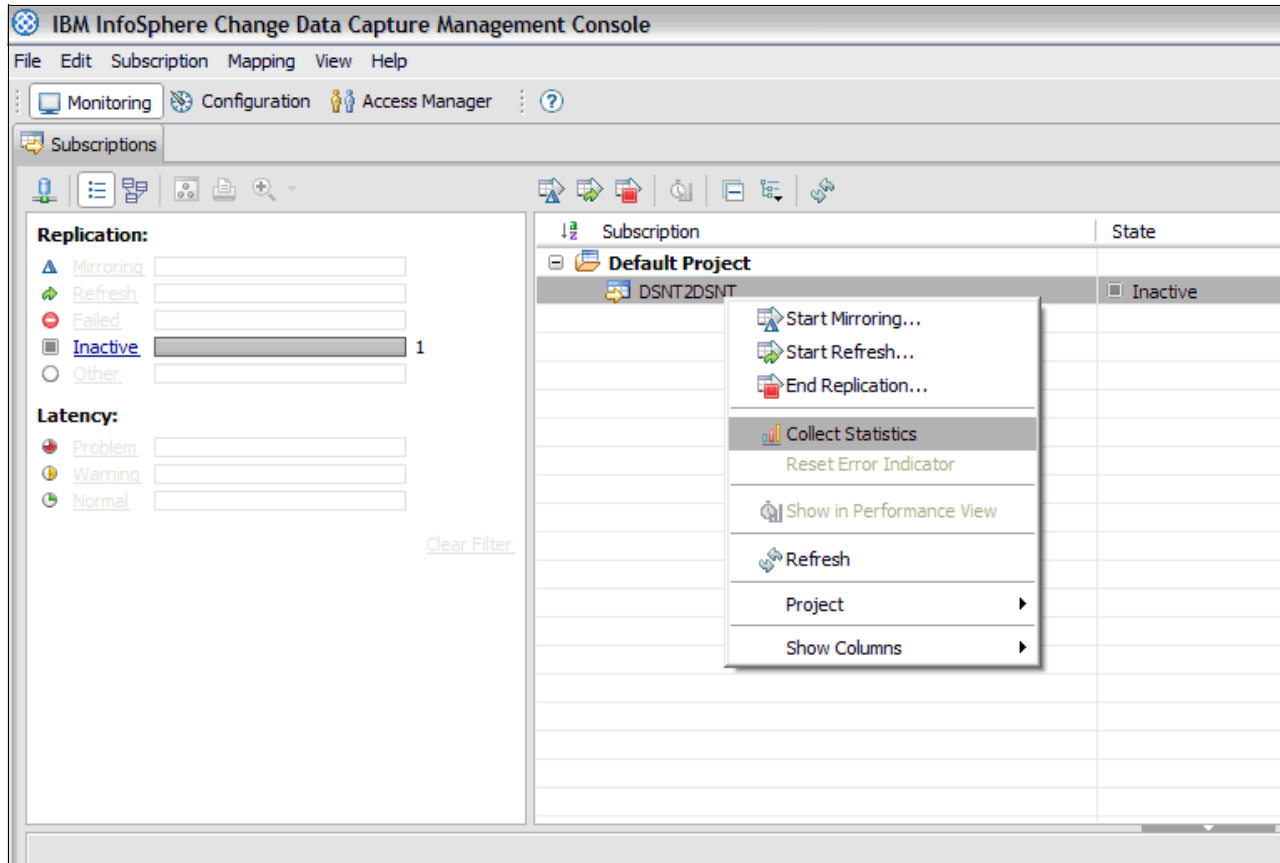


Figure 3-1 Enabling statistics collection

Now that statistics collection is enabled, click the activity or latency monitoring buttons in the lower right corner of the window, as shown in Figure 3-2.

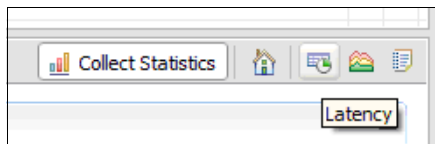


Figure 3-2 Monitoring button

In the statistics windows, you are shown a tabular view on the left side, which also tracks maximum, minimum, and average values. On the right side is a graph that periodically refreshes.

With statistics collection enabled and the statistics window open, we ran test workloads against the DB2 z/OS source tables and watched InfoSphere Change Data Capture performance via Management Console.

Prior to tuning, InfoSphere Change Data Capture was unable to keep up with the DB2 z/OS workload. This was evident from the fact that the latency, which measures the time between the source commit and the target commit, continued to grow, as shown in Figure 3-3 on page 40.

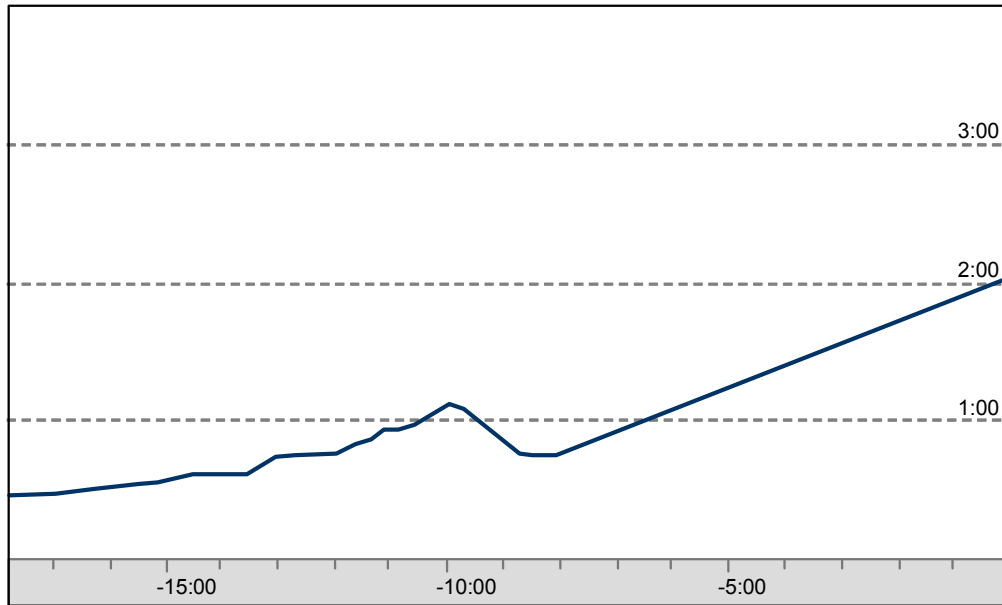


Figure 3-3 Growing latency indicates that tuning is required

Upon switching to the throughput graph (Figure 3-4), we saw that our maximum throughput was less than 500 kilobytes per second. This was an indication that we had a lot of room to grow via tuning. Throughputs in the range of 18 megabytes per second per subscription have been achieved during InfoSphere Change Data Capture for z/OS benchmarks. For more information, see the IBM InfoSphere Change Data Capture Version 6.2 (DB2 for z/OS) Performance Evaluation and Analysis whitepaper.

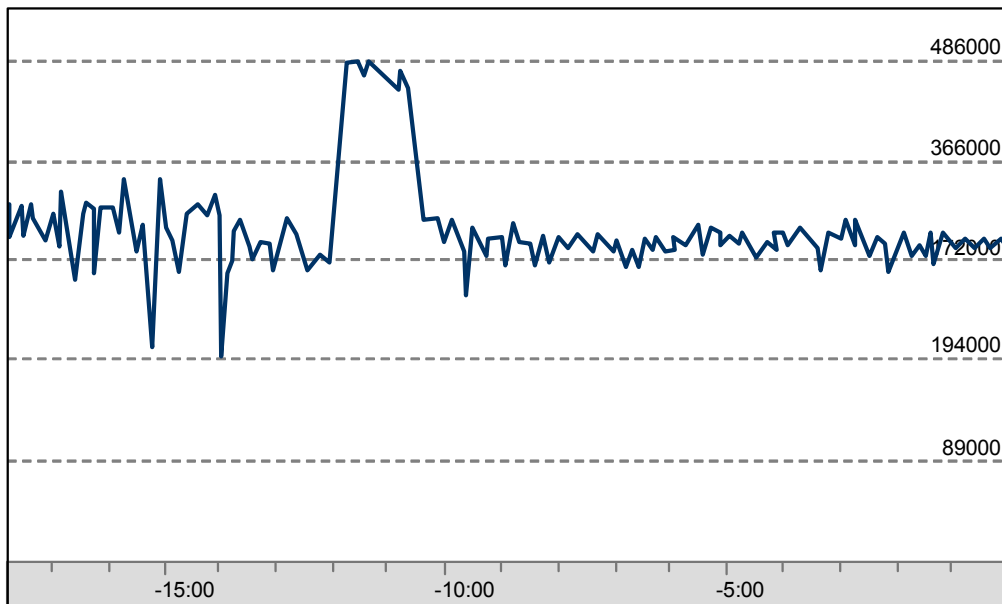


Figure 3-4 Initial throughput before tuning

After tuning, we were able to sustain sub-second latency with a throughput of 4 megabytes per second per subscription.

3.6 Monitoring of InfoSphere Change Data Capture CPU usage

In addition to tuning InfoSphere Change Data Capture to meet latency and throughput requirements, it is important to understand how various configuration parameters impact CPU utilization. In this section, we provide an explanation of the key tasks that run as part of the InfoSphere Change Data Capture address space. We also take a look at how several configuration parameters can impact CPU utilization.

For the purpose of this discussion, we monitored CPU usage of the various InfoSphere Change Data Capture tasks with IBM Tivoli® Omegamon XE for DB2 Performance Expert on z/OS.

3.6.1 InfoSphere Change Data Capture tasks

In most cases, it is sufficient to monitor InfoSphere Change Data Capture at the plan level because most of the plans directly correspond to tasks that the CDC agent is performing. In certain cases, it becomes necessary to look at individual DBRMs. In general, we refer to the various InfoSphere Change Data Capture tasks by plan name and by an associated three letter acronym. Keep in mind that each plan name has a two-character suffix that was specified when running the bind job, CHCBNDPL. When we are referring to an individual DBRM, we will explicitly state this.

CHCDALxx (DAL)

DAL stands for DBMS Asynchronous Log. The DAL task is responsible for setting up and taking down the MONITOR trace, as well as clearing data from the OPx buffer upon notification from DB2. Every InfoSphere Change Data Capture started task will have a DAL task, even if the associated CDC agent is never used as a source.

DCW

The DCW task is not associated with any DB2 plans or DBRMs. The sole responsibility of the DCW task is to move data between the level 1 and level 2 caches when caching is enabled.

CHCDLPxx (DLP)

DLP stands for DBMS Log Profile. The DLP task gathers information from the DB2 log that is not directly related to changes in the data of the source tables. For example, the DLP task looks for REORGs on the source tables.

CHCDLRxx (DLR)

DLR stands for DBMS Log data cache Reader. The DLR task is responsible for reading data from the DB2 log and populating the level 1 cache. One DLR task can be shared over multiple subscriptions.

CHCDLSDB (DLS)

CHCDLSDB is a DBRM associated with the DLS task. The DLS task is responsible for processing DB2 log data for a single subscription. The DLS task receives data from the corresponding DSL task. It will group changes by unit of work and stage them in hiperspace waiting for a COMMIT or a ROLLBACK. If a ROLLBACK is detected, the changes associated with that unit of work are discarded. If a COMMIT is detected, the changes associated with that unit of work are released to be sent to the target CDC agent. By default, the hiperspace staging area for a single DLS task, and therefore for a single subscription, is allowed to grow up to 2 Gb in size. The staging area can be restricted to a smaller maximum size via the MAXSUBSCRSTAGESIZE parameter in CHCDBMxx. The DLS DBRM is part of the SDT plan.

CHCDSCxx (DSC)

DSC stands for DBMS Supervision and Control. The DSC task is responsible for startup and shutdown of subscriptions. It is also responsible for transition of subscription state, for example, from refresh to mirroring.

CHCDSLxx (DSL)

DSL stands for DBMS synchronous log. The DSL task is responsible for reading DB2 log data for a single subscription. If caching is enabled, the DSL task will attempt to find the necessary data in the level 1 or level 2 cache. If caching is not enabled, or the necessary data is not in either cache, the DSL task will obtain the necessary data by issuing an IFI read. The DSL task continues to read log data from the cache or via IFI reads until it hits the current end of log. Upon hitting end of log, it suspends until awoken by the DAL task.

CHCDTCxx (DTC)

DTC stands for DBMS Table Change. The DTC task is responsible for applying changes to the target tables for a single subscription. Data transformations that take place at the target occur as part of the DTC task.

CHCDTRDB and CHCDTRD8 (DTR)

These two DBRMs are associated with the DTR task. DTR stands for DBMS Table Refresh. The DTR task is responsible for reading data from a source table as part of a refresh. The DTR DBRMs are part of the SDT plan.

CHCPAAxx (PAA)

PAA stands for Product Administration Agent. The PAA task is responsible for supporting Management Console. For example, the PAA task is used to provide Event Log information to the user via Management Console.

CHCSCTxx (SCT)

SCT stands for Source Control. The SCT task coordinates communication of the control information for a single subscription at the source side.

CHCSDTxx (SDT)

SDT stands for Source Data. The SDT task coordinates communication of the data information for a single subscription at the source side. Data transformations that take place at the source occur as part of the SDT task.

CHCTCTxx (TCT)

TCT stands for Target Control. The TCT task coordinates communication of the control information for a single subscription at the target side.

Figure 3-5 on page 43 shows the relationship between the DAL, DSL, DLS, and DTC tasks. Note that DI (Data In) and DO (Data Out) are TCP/IP tasks that are not associated with a DB2 plan. This diagram assumes that caching is not enabled.

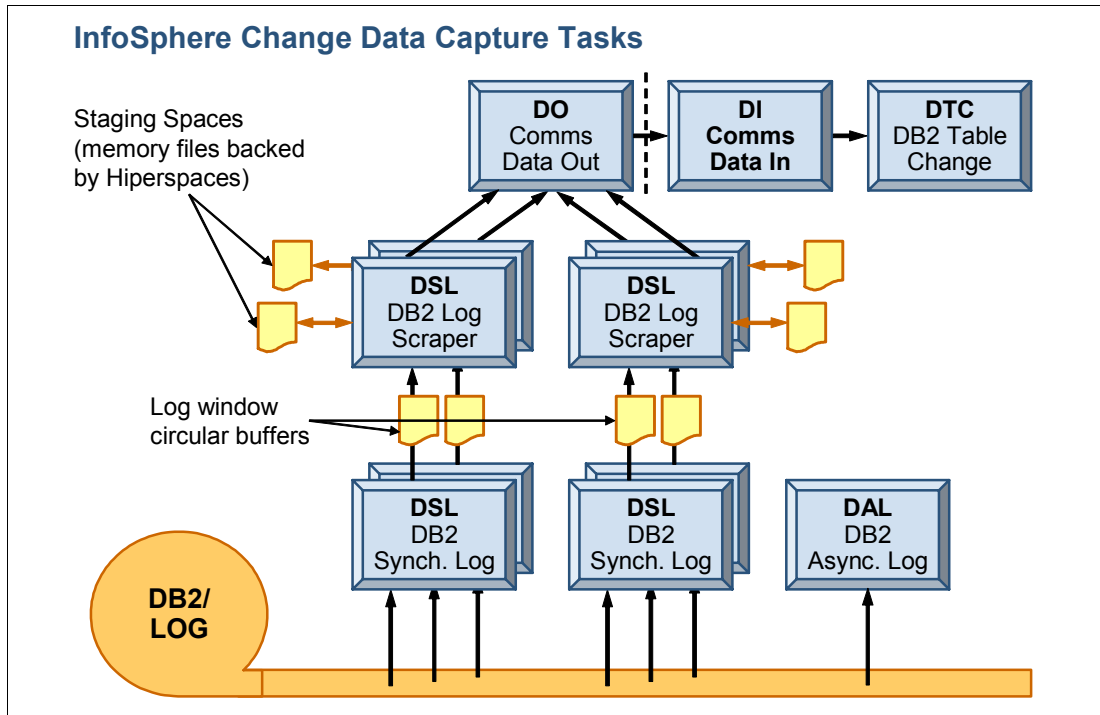


Figure 3-5 Task relationships with caching disabled

If caching is enabled, the DSL tasks instead attempt to read from the level 1 and level 2 caches. Only if the data is not found in the caches, would the DSL task issue an IFI read. Figure 3-6 shows the relationship between the DLR, DCW, and DSL tasks when caching is enabled.

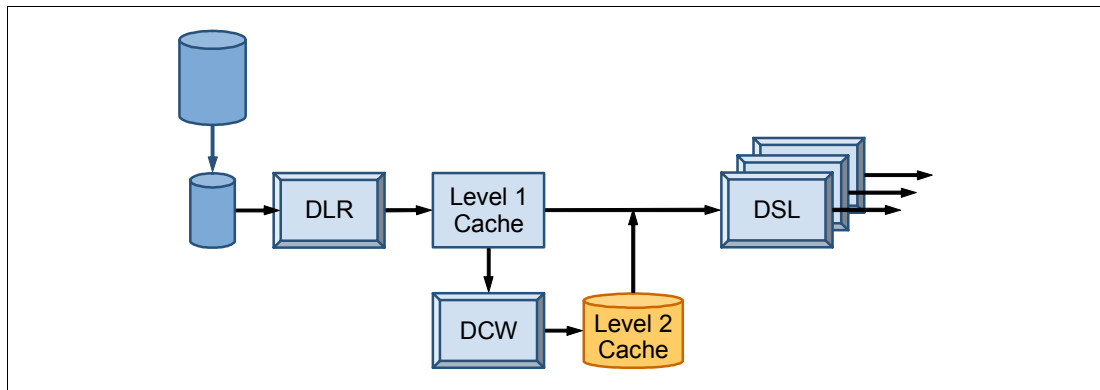


Figure 3-6 Task relationships with caching enabled

3.6.2 Using the DB2 LOAD utility for refresh

Typically, a refresh or initial load of the target tables is performed before continuous synchronization of the data begins. Ideally, this is the only time that a full refresh takes place. However, it is important to have a tuned refresh in the event that a full refresh is needed for recovery purposes. Any time InfoSphere Change Data Capture is not running and changes are occurring in the DB2 data, it will be falling behind the end of the DB2 log. When InfoSphere Change Data Capture comes back up, it will pick up where it left off in the DB2 log, and try to get caught up to the end of the log. If the DB2 workload is heavy enough and the current latency of the data is high enough, it might be decided that a full refresh of the

target tables is the more appropriate recovery method. In these cases, the elapsed time of the refresh is the largest factor in the overall recovery time.

Table 3-1 shows a comparison of USELOADER=NO, where the refresh is done with SQL statements, and USELOADER=YES, where the refresh is done with the DB2 LOAD utility. In this case, we were refreshing a little over six million rows. Although there was not a significant CPU difference between the two methods, the elapsed time difference is substantial. Typically in a recovery scenario, the elapsed time required to complete recovery and bring data latency back within the SLAs is critical.

Additionally, some of the work done by the DB2 LOAD utility is eligible for zIIP offload. In this case, the target table only had one index. Higher percentages of zIIP offload should be possible for tables with multiple indexes.

The total CPU times in the table were calculated by summing the CPU times from the DTR, SDT, and DTC tasks, as well as CPU used by the LOAD utility in the case of USELOADER=YES.

Table 3-1 Refresh time for 6,344,819 rows

	USELOADER=NO	USELOADER=YES
Total CPU time	3:57.196194	3:51.838860
zIIP-eligible CPU time	0.000000	1.113035
Total elapsed time	8:09.810448	4:39.977580

3.6.3 The effect of the COMMITFREQ parameter

The COMMITFREQ parameter is important, and challenging to tune correctly. It has an impact on many aspects of InfoSphere Change Data Capture performance. As we discussed in the section on CHCDBMxx, COMMITFREQ can greatly increase throughput. However, there is a trade-off that takes place because COMMITFREQ can also cause data latency to increase, and can increase the likelihood of deadlocks or timeouts. Additionally, the COMMITFREQ parameter can have a large impact on the amount of CPU that is used by the apply process, namely the DTC task.

Table 3-2 shows the amount of CPU, per change, that was used by the DTC task under two different COMMITFREQ settings. Data is given for two different subscriptions that were running concurrently.

Table 3-2 Effect of changing COMMITFREQ settings

CPU microseconds per change used by the DTC task	COMMITFREQ=(200,30,1)	COMMITFREQ=(5000,30,5000)
Subscription 1	234.768992	71.469031
Subscription 2	239.482282	72.047571



Upgrading from CDC V6.2

If you are upgrading from V6.2 to V6.5 of InfoSphere Change Data Capture for z/OS, you do not need to run all of the configuration steps. The following items from your existing 6.2 environment can be used as is for the new 6.5 environment:

- ▶ The PAL VSAM cluster
- ▶ The metadata VSAM cluster
- ▶ The SCHCDATA library
- ▶ The level 2 log cache

The only change required in the started task JCL is to point to the new 6.5 load library.

This chapter described the configuration steps that do need to be run when performing an upgrade. The required steps have to do with upgrading the DB2 metadata tables.

4.1 DB2 binds for the metadata table creation utility

In V6.5 the DB2 metadata tables are created by a utility program. The utility program takes various input values as parameters and creates the DB2 metadata tables as appropriate.

Two jobs must be run to upgrade the metadata tables. The first job, shown in Example 4-1, binds the plans used by the utility program. The second job runs the utility program. Sample job CHCMDMBD in SCHCCNTL is used to do the binds for the utility program. Follow the instructions that are contained within the job. Keep the following things in mind:

- ▶ The parameter <PlanSuffix> needs to be replaced with a two character suffix. This two character suffix should match the setting of PLANSUFFIX in CHCDBMxx.
- ▶ The parameter <CHCUserID> needs to be replaced with the z/OS user ID that the started task runs under.
- ▶ Replace <CHC.HCHC650> with the high level qualifier for the new V6.5 SCHCDBRM library.

Example 4-1 DB2 binds for the metadata creation utility program

```
//CHCMDMBD JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=&SYSUID
//DB2BIND EXEC PGM=IKJEFT01,
//          DYNAMNBR=20
//STEPLIB DD DSN=DB21010.ZDW.SDSNLOAD,DISP=SHR          <==
//SYSTSIN DD *
DSN SYSTEM(DDW1)                                       /* <== */
BIND -
  PACKAGE(CHCPK101)                                     /* <== */ -
  MEMBER(CHCMDMMN) -
  ACTION(REPLACE) -
  CURRENTDATA(NO) -
  ENCODING(UNICODE) -
  OWNER(CHC01)                                         /* <== */ -
  QUALIFIER(CHC01)                                     /* <== */
BIND -
  PACKAGE(CHCPK101)                                     /* <== */ -
  MEMBER(CHCMDMDB) -
  ACTION(REPLACE) -
  CURRENTDATA(NO) -
  ENCODING(UNICODE) -
  OWNER(CHC01)                                         /* <== */ -
  QUALIFIER(CHC01)                                     /* <== */
BIND -
  PACKAGE(CHCPK201)                                     /* <== */ -
  MEMBER(CHCDB2DL) -
  ACTION(REPLACE) -
  CURRENTDATA(NO) -
  ENCODING(UNICODE) -
  OWNER(CHC01)                                         /* <== */ -
  QUALIFIER(CHC01)                                     /* <== */
BIND -
  PLAN(CHCMDM01)                                       /* <== */ -
  PKLIST(CHCPK101.CHCMDMMN -
         CHCPK101.CHCMDMDB -
         CHCPK201.CHCDB2DL -
        ) -
```



```

ACTION(REPLACE) RETAIN -
CACHESIZE(256) -
ISOLATION(CS) -
OWNER(CHCO1) /* <== */ -
RELEASE(COMMIT)
END
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DBRMLIB DD DSN=ICDC.ZDW.SCHCDBRM,DISP=SHR <==
//SYSUDUMP DD SYSOUT=*
//*
//

```

4.2 DB2 metadata tables

Sample job CHCMDMUT in SCHCCNTL is used to run the utility program to upgrade the DB2 metadata tables. Follow the instructions that are contained within the job, which is shown in Example 4-2. There are several important considerations for this job:

- ▶ Make sure that you replace <CHCDBName> with the name of the existing database containing the CDC metadata tables. This is how the job determines that it is doing an upgrade.
- ▶ Set <CHCGroup> to the storage group that is being used by the existing CDC metadata database.
- ▶ You do not need to replace <CHCVOLSER>.
- ▶ You still need to specify a catalog name parameter <CHCCatlg>. The value you specify will not get used.
- ▶ The parameter <CHCUserID> needs to be replaced with z/OS user ID the CDC started task will run under.
- ▶ You will need to edit the DSNs on the STEPLIB and CHCCNTRL DDs to point to the correct SCHCLOAD and SCHCDATA libraries. Since the old SCHCDATA library can be used as is, this is the V6.2 SCHCDATA, or a copy of it. The SCHCLOAD DSN must point to the new V6.5 load library.
- ▶ The two SYSDUMP DDs should be edited to use a DSN that is specific to this CDC instance.

Example 4-2 DB2 metadata table creation

```

//CHCMDMUT JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=&SYSUID
//DELETE EXEC PGM=IEFBR14
//SYSDUMP DD DSN=ICDC.ZDW.DDW1.SYSDUMP,DISP=(MOD,DELETE),
// UNIT=SYSALLDA,
// SPACE=(CYL,(1))
//*
//IEFPROC EXEC PGM=CHCMIT,
// PARM=('CHCMDMUT',
// '/TRAP(OFF),TER(UAIMM)',
// '/CONFIG=F1'),
// REGION=OM

```

```

//STEPLIB DD DSN=ICDC.ZDW.SCHCLOAD,DISP=SHR
//          DD DSN=ICDC.ZDW.SDSNEXIT,DISP=SHR
//          DD DSN=ICDC.ZDW.SDSNLOAD,DISP=SHR
//CHCCTRL DD DDNAME=IEFRDER
//IEFRDER DD DSN=ICDC.ZDW.SCHCDATA,DISP=SHR
//SYSMDUMP DD DSN=ICDC.ZDW.DDW1.SYSMDUMP,DISP=(MOD,CATLG),
//          UNIT=SYSALLDA,
//          SPACE=(CYL,(100,25))
//ABNLIGNR DD DUMMY
//SYSIN DD *
* Lines with an '*' in column 1 are treated as comments.
*
* Subsystem or data sharing group name of the
* DB2 system which holds (or will hold) the meta-data.
SUBSYS=DDW1
* The plan suffix used when the plan CHCMDMxx was bound.
PLANSFX=01
* The name of the DB2 storage group which should be used
* to create any meta-data objects. This storage group
* will be created if it does not exist.
STOGROUP=SYSDEFLT
* The list of volume names to be used to create the
* storage group if it does not exist. Must be specified
* if the storage group does not exist.
VOLUMES=<CHCVOLSER>
* The name of the catalog to use in creating the
* storage group if it does not exist. Must be specified
* if the storage group does not exist.
CATLG=CATALOG
* The name of the database which holds the meta-data.
* This database will be created if it does not exist.
DBNAME=CDCDB
* The name of the table space which will contain the majority
* of the meta-data tables. This tablespace should be a Unicode
* table space with a page size of 4K. This table space will
* be created if it does not exist.
TBSP01=CDCTS1
* The name of the table space which will contain meta-data tables
* which require a 32K page size. This tablespace should be a
* Unicode table space with a page size of 32K. This table space
* will be created if it does not exist.
TBSP02=CDCTS2
* The user name which owns the meta-data.
OWNER=CHC01
* The SQLID this job should operate under. By default, this
* is the user ID this job is executing under.
SQLID=RES01
* When set to Y, this option will cause all existing
* meta-data tables to be dropped before this utility
* starts its regular processing.

```

```
DROPTBL=N
* When set to Y, this option will cause all DB2 system
* catalog tables indexes to be created. The storage group
* used for the indexes will be the storage group specified
CRSYSIND=N
/*
//SYSPRINT DD SYSOUT=*
/*
//
```

4.3 Bind DB2 plans

Sample job CHCBNDPL can be used to bind the plans needed by the DB2 z/OS CDC agent. When editing this job, keep the following items in mind:

- ▶ The parameter <PlanSuffix> must be replaced with a two character suffix. This two character suffix should match the setting of PLANSUFFIX in CHCDBMxx.
- ▶ The parameter <CHCUserID> must be replaced with the z/OS user ID that the CDC started task will run under.
- ▶ The parameter <CHC.HCHC650> must be replaced with the high level qualifier for the new V6.5 SCHCDBRM library.

4.4 New parameter

There is one new parameter in V6.5 that you should consider adding to your CHCDBMxx configuration member. See “MAXINSTGLOBSIZE” on page 34.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publication provides additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Co-locating Transactional and Data Warehouse Workloads on System z*, SG24-7726

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- ▶ Program Directory for IBM InfoSphere Change Data Capture for z/OS V6.5, GI11-7880
- ▶ InfoSphere Change Data Capture for z/OS V6.5 User Documentation
- ▶ InfoSphere Change Data Capture V6.5 Management Console Administration Guide
- ▶ InfoSphere Change Data Capture V6.5 Access Server and Management Console Installation Guide

Online resources

These websites are also relevant as further information sources:

- ▶ installing, configuring, and using CDC, please visit the InfoSphere Change Data Capture Information Center

<http://publib.boulder.ibm.com/infocenter/cdc/v6r5m1/index.jsp/>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Implementing IBM InfoSphere Change Data Capture for DB2 z/OS V6.5



Configuration

Performance tuning

Migration

IBM InfoSphere Change Data Capture for z/OS uses log-based change data capture technology to provide low impact capture and rapid delivery of changes to and from DB2 z/OS in heterogeneous environments without impacting source systems. Customers get the up-to-date information they need to make actionable, trusted business decisions while optimizing MIPS costs. Change Data Capture can also be used to synchronize data in real time between multiple data environments to support active data warehousing, live reporting, operational business intelligence, application consolidations and migrations, master data management, and to deliver data to SOA environments.

This IBM Redpaper document describes InfoSphere Change Data Capture, how to install and configure it, and how to migrate to the latest release.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks