

IBM Z Performance and Capacity Analytics

Version 3.1

Implementation Guide

and

Best Practices

Last updated: 18 July 2023

<https://www.ibm.com/support/pages/node/7007309>

Contents

About this document	1
Introduction to IBM Z Performance and Capacity Analytics.....	1
Skill requirements	1
System requirements.....	2
Installing base features and applying PTFs	2
Installing SMP/E base features	2
Applying latest SMP/E PTFs	3
Creating the Db2 database for IZPCA	3
Implementing RACF security.....	5
Creating RACF groups	5
Connecting IZPCA users to the DRLUSER group.....	5
Creating a DRLPROF data set	5
Creating an IZPCA startup CLIST.....	5
Creating the IZPCA system tables	6
Setting up QMF	7
Installation verification	7
Determining Db2 table partitioning.....	7
Installing required components.....	9
Customizing lookup tables	10
Batch Collect vs Continuous Collect.....	10
Setting up a batch collect.....	11
Continuous Collect configuration options	12
Standalone (Single system)	12
Defining a DASD-only log stream	12
Setting up the SMF Extractor	13
Customizing the Continuous Collector startup	15
Hub and Spokes (multi-system)	16
Setting up the DataMover on the Hub	16
Setting up the spoke systems	19
PROC and JCL to run the DataMover(s)	21
Setting up the Continuous Collector	22
Maintenance recommendations.....	23
Improving COLLECT performance	23

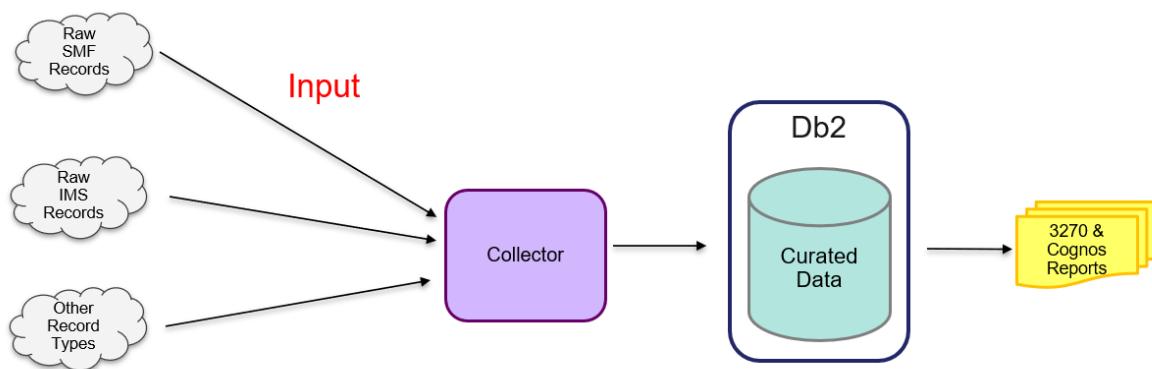
Summary of implementation recommendations	24
Cognos implementation.....	25
Server machine specifications	25
Overview of Cognos Installation	25
IBM ShopZ - Downloading Cognos.....	26
ShopZ Ordering IZPCA.....	26
ShopZ Locating the Cognos Media.....	28
Media Files – additional considerations	28
Installing Cognos in Windows	28
Post Cognos installation tasks.....	28
Define the Data Server Connection	29
Installing the IZPCA Report Package	29
Verifying Cognos installation for IZPCA reporting	29
Appendix A. Sample JCL to create a new CSI	30
Appendix B. Sample JCL to create your <i>user.DRLFPROF</i> data set.....	31
Appendix C. Sample SQL statements for Partitioning.....	36
Appendix D Sample Batch Collect JCL.....	38
Appendix E. Sample Continuous Collect JCL	39
Appendix F. Continuous Collector operator commands.....	40
Stop the Continuous Collector	40
Modify the Continuous Collector.....	40
Interval Message	40
Commit Frequency.....	40
Full Statistics Frequency.....	40
Pause and Resume the Continuous Collector	40
Commit Time messages	40
ReOpenLogstream function	41
REFRESH the Continuous Collector definitions.....	41
Debugging	41
Appendix G. SMF Extractor operator commands	42
Appendix H. Collecting and processing CICS Monitor records	43
Populate the IZPCA CICS_DICTIONARY table	43
Collecting CICS Monitor records (SMF110).....	43
Appendix I. DataMover.sh	44
Appendix J. Glossary	45

About this document

This document (also known as *The IZPCA Installation Cookbook*) is intended for new installations of IBM Z Performance and Capacity Analytics 3.1.0. It contains step-by-step instructions and recommended practices to guide you through the implementation so you can start using the product and explore its benefits as quickly as possible.

Introduction to IBM Z Performance and Capacity Analytics

IBM Z Performance and Capacity Analytics (IZPCA) collects records from various sources, curates the data and stores the results in a Db2 data base table. The stored information can then be reported on at hourly, daily, weekly, or monthly intervals.



Skill requirements

The following table lists the tasks and skills required to install and customize IZPCA. The tasks are listed in the order they are to be completed.

Task	Required skill and authority
Installing the base feature	z/OS Systems Programmer
Applying PTFs	z/OS Systems Programmer; IZPCA Administrator with SYSADM authority
Creating the Db2 database for IZPCA	Db2 Administrator; z/OS Systems Programmer
Implementing RACF security	RACF Security Administrator; IZPCA Administrator
Creating a DRLPROF data set	IZPCA Administrator; IZPCA User
Creating an IZPCA startup CLIST	z/OS Systems Programmer
Creating the IZPCA system tables	IZPCA Administrator with SYSADM authority
Setting up QMF	IZPCA Administrator with SYSADM authority
Installation verification	IZPCA Administrator with SYSADM authority
Determining Db2 table partitioning	Db2 Administrator; IZPCA Administrator with SYSADM authority; Capacity Planner
Installing required components	IZPCA Administrator with SYSADM authority

Task	Required skill and authority
Setting up a batch collect	z/OS Systems Programmer; IZPCA Administrator with SYSADM authority
Setting up the Hub and Spokes (multi-system)	z/OS Systems Programmer; UNIX knowledge
Setting up continuous collect	z/OS Systems Programmer; IZPCA Administrator with SYSADM authority
Cognos implementation	IZPCA Administrator with SYSADM authority; Cognos Administrator

System requirements

- z/OS 2.2.0 or later
- Db2 for z/OS 11.0.0 or later
- 64bit JAVA to run the DataMovers on each Spoke and Hub

Installing base features and applying PTFs

Skill required: z/OS Systems Programmer.

Installing SMP/E base features

The full installation instructions are in the [Program Directory for IBM Z Performance and Capacity Analytics 3.1.0](#). A summary is documented in this section.

1. Copy the sample members DRLJZAAL, DRLJZACC, DRLJZAPP, DRLJZDDD, DRLJZFSZ, DRLJZMKD, DRLJZRCV, and DRLJZSMK, from the IBM.HLI0C01.F6 product RELFILE to a work data set.
2. To install the product, tailor the JCL members then run the jobs in the following order:
 - a. DRLJZAAL - Allocate local data sets and SMP/E Target and Distribution Libraries.
 - b. DRLJZFSZ – Allocate the zFS data set, allocate the directory structure, and mount the zFS.
 - c. DRLJZSMK - Invoke the supplied DRLJZMKD EXEC to allocate file system mount point for the zFS data set.
 - d. If you are NOT planning to use your global CSI, create and prime a CSI data set for IZPCA. See [Appendix A](#) for sample JCL to create a new CSI if required.
 - e. DRLJZDDD – Create the DDDEF entries.
 - f. DRLJZRCV – Perform the SMP/E Receive.
 - g. DRLJZAPP – Perform the SMP/E Apply.
 - h. DRLJZACC – Perform the SMP/E Accept.

Applying latest SMP/E PTFs

Skill required: z/OS Systems Programmer, IZPCA Administrator with SYSADM authority.

IZPCA adopts a continuous development process and as such regularly releases new APAR/PTFs that fall into two categories:

- Maintenance APARs that address general user and solution issues.
- New Feature APARs that introduce additional features and capabilities.

To maintain compatibility and keep up to date with new features, maintenance must be applied on a regular basis.

All IZPCA PTF/APAR information for both maintenance and new features can be found here:

<https://www.ibm.com/support/pages/node/6187269>

Creating the Db2 database for IZPCA

Skill required: Db2 Administrator.

Before you begin

Before you start installing Db2, your Db2 Administrator must create a new dedicated Db2 subsystem for use with IZPCA only.

About this task

BEST PRACTICE:

Install IZPCA in its own dedicated Db2 subsystem.

This allows customization of that Db2 subsystem to suit the IZPCA processing requirements and avoids any possibility of IZPCA causing issues with any other application systems.

The global Db2 ZPARMs affect all applications executing in the same Db2 subsystem. Having IZPCA in its own dedicated Db2 subsystem allows you to modify global Db2 ZPARMs to better align with IZPCA requirements.

To modify the global Db2 ZPARM settings for IZPCA, observe the following recommendations:

1. LIKE_BLANK_IN SIGNIFICANT

This parameter MUST be set to NO.

If this is set to yes, the collector will not work when using the 'INCLUDE' clause.

2. NUMLKTS and NUMLKUS

When performing batch collect with very large buffers, Db2 holds many simultaneous locks. This can result in an abend when the maximum number of locks is reached.

Two Db2 ZPARMs contribute to this situation:

❖ NUMLKTS

The number of locks that can be held on a table/partition before escalation to a table/partition lock occurs.

Reduce the NUMLKTS parameter to 1000 so IZPCA makes use of Db2 lock escalation and prevents lock abends. After 1000 pages are locked, the lock escalates to a partition level lock and releases the 1000 page locks.

❖ **NUMLKUS**

The number of locks a user may hold at any given time.

Increase the NUMLKUS parameter to 500,000 as the collect user may be required to maintain locks on hundreds of tables at the same time. For example, 1000 pages locked for 500 tables = maximum 500,000 locks held.

3. URLGWTH

The number of uncommitted records held before a warning is issued.

Increase the URLGWTH value from default 10 to 100 (meaning 100,000) to prevent excessive warning messages in the Db2 Master.

4. REORG_DROP_PBG_PARTS

Set to DISABLE. If set to ENABLE, it can cause issues with the purge process when using the REORG/DISCARD option.

IZPCA uses a Db2 LOB (introduced by APAR PH46966). The LOB index buffer pool is implicitly set by the TSBPLOB subsystem parameter. The default is BP0. If you set it to any other value, for example BP3, then you must grant use of that buffer pool to the IZPCA user or to public.

Example command:

```
GRANT USE OF BUFFERPOOL BP3 TO PUBLIC
```

Procedure

1. Edit the *h/q.SDRLCNTL(DRLJDBIN)* member and make the following changes:

- a. Scroll to the bottom, uncomment the DD statement for the appropriate version of SDRLCNTL(DRLJDCV*) for your level of Db2, and modify the content of the DRLJDCV* member to suit your environment. By uncommenting the DD statement, this member is now included in the SYSIN concatenation for DRLJDBIN and used to create the IZPCA VIEWs on IBM Db2 version-specific system tables.
- b. Tailor DRLJBIN according to the notes in the member.
- c. Modify the data set name in the LIB parameter of the RUN statement to suit your environment.
- d. Remove the QUERYACCELERATION(NONE) parameter from the BIND PACKAGE.
Note: Do not forget to remove the continuation dash on the previous line.

2. DB2ADM authority is required to run this job.

3. Submit the job to complete the following steps:

- Run the Db2 BIND for IZPCA.
- Create the storage groups only if special storage groups are required for IZPCA.
- Create a Db2 database.
- Create IZPCA VIEWs on IBM Db2 version-specific system tables.

Implementing RACF security

Skill required: RACF Security Administrator, IZPCA Administrator.

Creating RACF groups

Create three RACF groups DRL, DRLSYS, and DRLUSER and connect the IZPCA Administrator to all three.

```
ADDGROUP DRL
ADDGROUP DRLSYS
ADDGROUP DRLUSER
CONNECT (ADMIN1,ADMIN2) GROUP (DRL)
CONNECT (ADMIN1,ADMIN2) GROUP (DRLSYS)
CONNECT (ADMIN1,ADMIN2) GROUP (DRLUSER)
```

Connecting IZPCA users to the DRLUSER group

Connect the IZPCA users to the DRLUSER group only.

```
CONNECT (USER1,USER2) GROUP (DRLUSER)
```

Creating a DRLPROF data set

Skill required: IZPCA Administrator; IZPCA User.

About this task

Each user of the IZPCA dialog needs to create a *user.DRLPROF* data set that is used to set up their IZPCA ISPF environment defaults. The *user.DRLPROF* is read by the DRLEINIT startup REXX exec invoked by the startup CLIST described in "[Creating an IZPCA startup CLIST](#)".

Procedure

1. Create a *user.DRLPROF* data set with attributes DSORG=PS, RECFM=FB, and LRECL=80.
2. Copy the sample from *hlq.SDRLCNTL(DRLPROF)* to your *user.DRLPROF* data set.
3. Tailor *user.DRLPROF* to suit your environment. Refer to the sample in [Appendix B](#) for instructions and information on the parameter settings.
4. To use the IZPCA ISPF dialog for different IZPCA instances, update the DRLPROF data set each time you switch between instances and restart the IZPCA ISPF dialog with the RESET option. The sample CLIST includes RESET on the %DRLEINIT command.

Creating an IZPCA startup CLIST

Skill required: z/OS Systems Programmer.

Tailor the ISPF startup procedure:

- Customize the PROC in the following example to suit your system.
- Store it in a procedure library (PROC library) on your system.
- Add the proc to the RACF class STARTED. If not using RACF, add the started task to your security system.

```

PROC 0
/*********************************************************************
/* CLIST TO ALLOCATE IZPCA DATA SETS
/*********************************************************************
CONTROL NOFLUSH NOMSG NOLIST NOCONLIST
SET &IZPCA31 = IZPCA.V3R1

ISPEXEC CONTROL ERRORS RETURN

ALTLIB ACTIVATE APPLICATION(CLIST) +
DATASET('&IZPCA31..SDRLEXEC')
ISPEXEC LIBDEF ISPLLIB DATASET ID('&IZPCA31..SDRLLOAD')
ISPEXEC LIBDEF ISPMLIB DATASET ID('&IZPCA31..SDRIMENU')
ISPEXEC LIBDEF ISPPLIB DATASET ID('&IZPCA31..SDRLPENU')
ISPEXEC LIBDEF ISPSLIB DATASET ID('&IZPCA31..SDRLSKEL')
ISPEXEC LIBDEF ISPTLIB DATASET ID('&IZPCA31..SDRLTENU')

ISPEXEC SELECT CMD(%DRLEINIT RESET) NEWAPPL(DRL) PASSLIB

ALTLIB DEACTIVATE APPLICATION(CLIST)
ISPEXEC LIBDEF ISPLLIB
ISPEXEC LIBDEF ISPMLIB
ISPEXEC LIBDEF ISPPLIB
ISPEXEC LIBDEF ISPSLIB
ISPEXEC LIBDEF ISPTLIB
EXIT

```

Creating the IZPCA system tables

Skill required: IZPCA Administrator with SYSADM authority.

Before you begin

Ensure the following libraries are allocated at login:

ISPLLIB: *h/q.SDSNLOAD*

About this task

Start IZPCA using the startup CLIST created in the previous step. From the ISPF command line, enter the command:

TSO STARTPCA

Use the IZPCA ISPF dialog to create the system tables as follows.

Procedure

1. From the IBM Z Performance and Capacity Analytics Administration menu, select option 1 **System** and press Enter.
2. From the System pop-up menu, select option 2 **System tables** and press Enter.
3. On the System Tables pop-up window, press F5 Create. Your ISPF session will be locked for a few minutes while the system tables are created.

4. Wait until your ISPF session is freed and the following confirmation message is displayed:
DRLA077 System tables are created successfully.

Setting up QMF

Skill required: IZPCA Administrator with SYSADM authority.

This step is optional and dependent on having the QMF software.

If you are planning to use QMF with the ISPF reports included with IZPCA, then Import the QMF initialization query using the IZPCA panels as follows:

1. From the IBM Z Performance and Capacity Analytics Administration panel, select option 1 **System** and press Enter.
2. From the System pop-up menu, select option 3 **Import QMF initialization query** and press Enter.

Installation verification

Skill required: IZPCA Administrator with SYSADM authority.

When the setup is complete, perform verification testing by installing and collecting to the Sample component. To do this, follow the instructions in "[Testing the installation of the base](#)" in the *Administration Guide and Reference*.

Determining Db2 table partitioning

Skill required: Db2 Administrator; IZPCA Administrator with SYSADM authority; Capacity Planner.

Before you begin

Before you install any components, consider if you wish to run multiple collectors in parallel, collecting SMF records to a single Db2 database, to improve throughput.

If you are intending to collect data from multiple sysplexes or collect high volumes of SMF data, you need to prepare the database for partitioning. This enables multiple IZPCA collectors to run against the database in parallel.

About this task

If you are intending to implement this architecture, follow the procedure below to enable partitioning on MVS System ID by RANGE.

By default, table spaces are created with partitioning by GROWTH. If you do not enable partitioning on MVS System ID by RANGE, your Collectors will cause locks on the database while they run in parallel.

Here is an example of how to split the SMF record processing between 3 collectors, running in parallel, and using partitioning by RANGE:

*SMF log 1 contains MVS System IDs AAAA and DDDD are processed by collector 1
SMF log 2 contains MVS System IDs BBBB and EEEE are processed by collector 2
SMF log 3 contains MVS System IDs CCCC and FFFF are processed by collector 3*

Db2 partitions are set up as follows:

<i>Partition</i>	<i>LIMITKEY (MVS_SYSTEM_ID)</i>	
1	AAAA	Note:
2	BBBB	Collector 1 has exclusive use of partitions 1 and 4
3	CCCC	Collector 2 has exclusive use of partitions 2 and 5
4	DDDD	Collector 3 has exclusive use of partitions 3 and 6.
5	EEEE	
6	FFFF	
7	<i>Maxvalue</i>	

By partitioning this way and ensuring the collectors match the Db2 partitions. This enables IZPCA to use a feature in Db2 known as *lock escalation* when committing data to Db2 tables. Lock escalation occurs when a process reaches the maximum number of locks allowed on a table space (LOCKMAX or NUMLKTS parameter).

When this occurs, Db2 takes a single lock on the entire partition and releases the other (page) locks. Using lock escalation allows IZPCA to maintain concurrency without the overhead of excessive locking. Without Db2 lock escalation, the thread may hold more locks than the system allows (set by NUMLKUS parameter).

Partitioning of Db2 tables during installation for IZPCA components is controlled by the system tables GENERATE_KEYS and GENERATE_PROFILES. These tables are created during the create system tables process but require modification to suit your environment.

The SDRLDEFS member DRLTKEYS contains the default definitions installed during the create system tables process. As supplied:

- The GENERATE_PROFILES table contains all the PROFILEs used in the SDRLDEFS data set for Db2 table space, table, and index creation. The default table space type is GROWTH.
- The GENERATE_KEYS table contains 4 default keys per PROFILE as a sample for partitioning by RANGE.

Procedure

1. Refer to [Appendix C](#) as a model for specifying the SQL statements to change the GENERATE_PROFILES and GENERATE_KEYS system tables for RANGE partitioning.
2. Note that the profiles you need to change to RANGE in the GENERATE_PROFILES table are the following two **only**:
 - SMF
 - SMF SYSTEM_ID
3. Create a member containing the following SQL statements:
 - The SQL delete statements to remove the default profiles SMF and SMF SYSTEM_ID that were installed during system table creation.
 - The new SQL statements you require for RANGE partitioning for the profiles SMF and SMF SYSTEM_ID.

4. Modify the JCL supplied in member *h/q.SDRLCNTL(DRLJGENA)* to suit your environment, replace the DRLIN content with your SQL statements, then submit the job to action the SQL statements.

Installing required components

Skill required: IZPCA Administrator with SYSADM authority.

Before you begin

Before you start installing components, carefully consider which components you need to perform performance management and capacity planning. Each installed component creates additional overhead and uses additional MIPS. Only install the components that are required to meet your needs.

To evaluate your needs, examine the predefined reports for each component and subcomponent. In this way, you can limit the data that needs to be collected and match the requirements to the business needs.

For a full list of the IZPCA reports, refer to:

[IBM Z Performance and Capacity Analytics V3.1.0 reports list](#)

About this task

Do not install components that do not meet your needs.

The following components meet the needs of most businesses, and would be a good starting point:

- Key Performance Metrics - z/OS
Requires SMF records 30, 70, 71, 72, 73, 74, 99, 113
- Key Performance Metrics - DB2
Requires SMF records 100, 101, 102
- Capacity Planning - z/OS
Requires SMF records 30, 70, 71, 72, 73, 89, 113
- Capacity Planning - DB2
Requires SMF record 101

For more information, refer to [Introduction to the Key Performance Metrics components](#) in the [Administration Guide and Reference](#) and the [Capacity Planning Guide and Reference](#).

Procedure

1. From the IBM Z Performance and Capacity Analytics Administration menu, select option 2 **Components** and press Enter.
2. From the Components menu, type / to select the component you wish to install, then press F6 Install.
3. If a Component Parts submenu is displayed, type / against one or more of the component parts you wish to install, then press Enter.
4. On the Installation Options pop-up menu, select either of the following options:
 1. Select option 2 **Batch** and press Enter. This generates a JCL member, which you can then submit for batch execution and use SDSF to monitor the progress of the job.
 2. Select option 1 **Online** and press Enter. This installs the component immediately but locks your ISPF session during the installation process.

Customizing lookup tables

About this task

Many of the components use lookup tables that may need to be customized in order to perform aggregation and calculations to suit your requirements.

Review the supplied content of the lookup tables to determine if they suit your requirements. If required, customize your lookup tables before you start collecting any data with the collector.

The spreadsheet [IZPCA-LookupTables-XREF](#) contains a cross reference of components to lookup tables to enable you to determine which lookup tables need to be reviewed. To help with customization, the spreadsheet also contains details of some of the lookup tables.

Procedure

To modify a lookup table:

1. Ensure that the user ID of the person updating the lookup tables has SYSADM authority.
2. Determine the name of the lookup table from the [IZPCA-LookupTables-XREF](#) that you wish to modify.
3. From the IBM Z Performance and Capacity Analytics Administration menu, select option 4 **Tables**.
4. From the list of tables, locate the lookup table, select it with / and move your cursor to Edit in the menu bar at the top of the panel and press Enter.
5. From the Edit dropdown menu, select option 3 **ISPF editor**.
6. If the **Temporary update of the SQLMAX value** panel is displayed, update the value as instructed and press Enter.
7. A formatted edit of the lookup table is displayed.
8. Update the table as required keeping to the format as initially displayed.
You may change existing lines or add additional ones.
9. Press PF3 to save the changes.

Batch Collect vs Continuous Collect

You have the option of running the Collector in Batch mode, or in Continuous mode.

Batch mode collects data from SMF, IMS, and other sources. These input sources are assigned to the DD name DRLLOG. The data is processed by the collector and stored in Db2 tables. At completion statistics about the collect are written to the JOB log.

Continuous mode only collects SMF data, and the input is from a log stream created by the SMF Extractor. The data is processed continuously and written to Db2 regularly based on the COMMIT AFTER value supplied in the COLLECT statement. The frequency of the statistics output is controlled by the collect parameter FULL STATISTICS AFTER on the COLLECT statement.

Once you have your Batch mode set up and you have verified the installation, you may wish to implement Continuous mode.

Setting up a batch collect

Skill required: z/OS Systems Programmer; IZPCA Administrator.

A batch collect reads SMF or non-SMF records from a sequential data set.

Sample JCL can be found here: *hlq.SDRLCNTL(DRLJCOLL)* and in [Appendix D](#).

If you plan to run in zIIP mode, you must add the SDRLLOAD data set to your list of APF authorized data sets. For example:

```
SETPROG APF,ADD,DSNAME= izpca-hlq.SDRLLOAD,SMS
```

In the DRLJCOLL JCL, DD name DRLNDICT is only required if collecting CICS Monitor data. If you do not collect these records, assign DRLNDICT to DUMMY. See [Appendix H](#) for more information on processing CICS Monitor records.

BEST PRACTICE:

The collector stores records in an internal buffer and updates that buffer with subsequent matching records then writes these to Db2 at Commit time. After a commit, if there are additional records that match a committed record, the record is read back into the buffer, incremented with the matching records, and updated in Db2 at commit.

It is beneficial to make the buffer as large as possible, thus reducing the amount of work done in Db2 and the number of Db2 log entries generated.

- Small batch collect buffers result in waits for I/O, which contribute to increased CPU usage. Set a buffer size of 1300M, if available. In SDSF, the VMAP command shows the size of the EXTENDED-PRIVATE area. Take this value and reduce it by 100M to use as the upper maximum for the buffer size parameter.

The following example is a partial list from the VMAP command. The value for the EXTENDED-PRIVATE area is 1652M, therefore set the buffer size on the COLLECT statement to $1652 - 100 = 1552$ M.

SDSF VIRTUAL STORAGE MAP XYZ1			XYZ1	LINE 1-24
COMMAND INPUT ==>				SCR
PREFIX=*	DEST=(ALL)	OWNER=XYZUSR*	SYSNAME=	
NP	NAME	Start-Address	End-Address	Size
	HV-SHARED	00000200_00000000	0001FFFF_FFFFFFFF	510T
	HV-COMMON	000001EF_80000000	000001FF_FFFFFFFF	66G
	HV-PRIVATE	00000050_00000000	000001EF_7FFFFFFF	1662G
	LOCALSYSAREA	00000010_00000000	0000004F_FFFFFFFF	256G
	RESERVED	00000000_80000000	0000000F_FFFFFFFF	62G
	EXTENDED-PRIVATE	00000000_18C00000	00000000_7FFFFFFF	1652M
	EXTENDED-CSA	00000000_09192000	00000000_18BFFFFFF	250M
	EXTENDED-MLPA	00000000_0895B000	00000000_09191FFF	8412K
	EXTENDED-FLPA	00000000_00000000	00000000_00000000	0

- Running the collector in zIIP mode:
 - Using large collect buffers will reduce General CP usage and provide increased offload to zIIP processors, reducing the consumption of MIPS/MSUs. Locating on a machine that has additional capacity and zIIP engines will be beneficial.

- ❖ The zIIP process is interrupted during the commit phase, hence more zIIP percentage is achieved with a larger buffer. A smaller buffer size means dropping out of zIIP to process the Db2 updates.
- ❖ While running in zIIP mode reduces General CP usage, the elapse times for batch collects may increase.

Continuous Collect configuration options

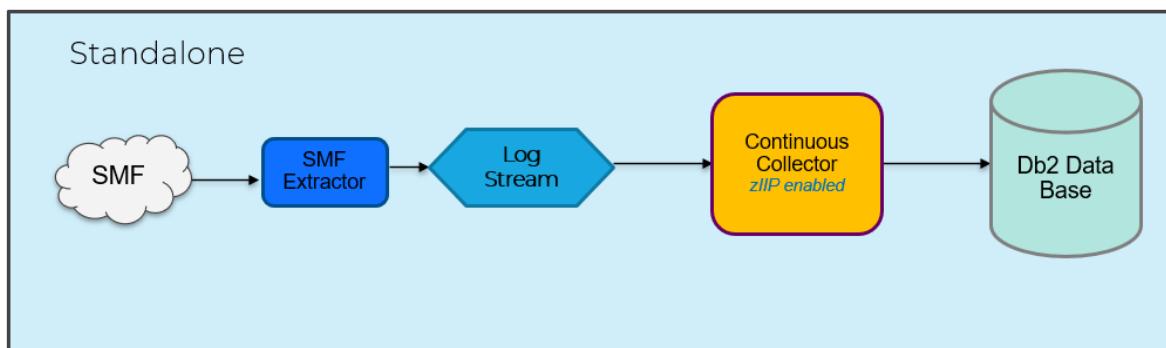
Skill required: z/OS Systems Programmer; IZPCA Administrator with SYSADM authority; Capacity Planner.

There are several ways the Continuous Collector can be configured.

The simplest configuration is standalone. An SMF Extractor writing to a DASD log stream where the Continuous Collector of IZPCA (CC) reads that log stream data and then updates the IZPCA Db2 database.

The first task is to configure and implement a log stream and an SMF Extractor.

Standalone (Single system)



Defining a DASD-only log stream

Customize the JCL before running the job by following the instructions in the comments.

Note: RETPD must be greater than zero.

```

//CRLOGRDS JOB (IZPCA)
//*****
//** DEFINE IZPCA LOG STREAM ON DASD
//*****
//** Change the following:
//**      <LOG_STRM> - Log stream name. Name must be unique by
//**      SYSID. Recommended name is <SYSID>.DRL.LOGSTRM
//*****
//IXCMIAPU EXEC PGM=IXCMIAPU
//STEPLIB DD DISP=SHR, DSN=SYS1.MIGLIB
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE (LOGR)
DEFINE LOGSTREAM NAME(log.stream.name)
DESCRIPTION (IZPCA_LOGSTREAM)
DASDONLY(YES)
MAXBUFSIZE(65532)
LS_SIZE(000000)
LS_ALLOCHEAD(3)
STG_SIZE(5000)

```

```

AUTODELETE(YES)
RETPD(1)
HLQ(DRL)
HIGHOFFLOAD(40)
LOWOFFLOAD(0)
DIAG(YES)
/*

```

BEST PRACTICE:

Ensure the log streams have adequate storage and are defined as DASD log streams, this is mandatory. These may require some fine tuning. Do not use coupling facility log streams.

AUTODELETE(YES) and RETPD (1) are required. These parameters allow the data to be kept for up to one day and will be deleted once processed. The value of RETPD must be a minimum of 1 but may be higher depending on retention requirements and available storage.

DASD log streams are preferred for several reasons:

- They are isolated to a single system.
- Coupling Facility (CF) log streams take up valuable space in the CF and XCF bandwidth.
- CF log streams require disk bandwidth to offload from the CF if required
- CF log streams introduce additional complexity in the case of failure.

CF log streams are not recommended. However, if CF log streams are set up, they should be in a separate CF structure and not share the same structure as the SYSLOG.

Setting up the SMF Extractor

Set up the SMF Extractor to intercept the SMF records and write them to the log stream.

DRLJSMFO - SMF Extractor control file

Sample member DRLJSMFO is provided in the *hlq.SDRLCNTL* library, and shown here. It contains parameters and instructions for configuring the SMF Extractor control file.

Before the first startup of the SMF Extractor, copy this member to your *user.LOCAL.CNTL* data set and rename as required, then customize the parameters by following the instructions in the comments.

Additional control statements are described in Chapter 4 [Installation Optional Extensions](#) of the IZPCA Administration Guide and Reference.

```

*****
* NAME: DRLJSMFO
*
* FUNCTION: SMF EXTRACTOR CONTROL FILE.
*
* COPY THIS MEMBER TO A DATA SET AS SMFPxxxx WHERE xxxx IS THE
* 4 CHARACTER SMF ID OF THE SYSTEM WHERE THE MEMBER WILL BE USED.
*
* THESE PARAMETERS MUST BE CUSTOMIZED TO MATCH YOUR INSTALLATION
* REQUIREMENTS PRIOR TO THE FIRST STARTUP OF THE SMF EXTRACTOR.
* * SET OUTLGR TO THE NAME OF THE OUTPUT LOG STREAM
* SET SMFREC TO A COMMA SEPARATED LIST OF SMF RECORDS TO BE RECORDED
*=====
OUTLGR=log.stream.name
SMFREC=30,70,71,72,73,74,89,99,100,101,102,113

```

DRLJSMFX - SMF Extractor startup procedure

The library *hlq.SDRLEXTR* containing the SMF Extractor modules must be added to your list of APF authorized data sets before attempting to start the SMF Extractor. For example:

```
SETPROG APF,ADD,DSNAME= hlq.SDRLEXTR,SMS
```

Sample member DRLJSMFX is provided in the *hlq.SDRLCNTL* library, and shown here. It contains the JCL for the SMF Extractor startup procedure.

Before the first startup of the SMF Extractor:

- Customize the parameters by following the instructions in the comments.
- Store the started task proc in a procedure library (PROC library) on your system.
- Add the started task to the RACF class STARTED. If not using RACF, add the started task to your security system.

```
//*****
//** NAME: DRLJSMFX
//*
//** FUNCTION: SMF EXTRACTOR STARTED TASK PROC
//*
//** The data set pointed to by the SMFEXPARM DD contains the SMF
//** Extractor parameters for each system in the sysplex.
//*
//** The parameter for a given system should be in member SMFPxxxx
//** where xxxx is the systems SMF ID.
//*
//*****=====
//SMFEXTP PROC SMFID=SYSA
//CKKXMAI EXEC PGM=CKKXMAI,REGION=0M,
// PARM='TRACE(N),MSGlvl(9),SDUMP(Y),MAXSD(2),MAXQD(4000)'
//*
//STEPLIB DD DISP=SHR,DSN=hlq.SDRLEXTR
//SYSPRINT DD SYSOUT=*
//SMFEXPARM DD DISP=SHR,DSN=user.LOCAL.CNTL(SMFP&SMFID.) ← SMF Extractor
//                                         control file
//SYSUDUMP DD SYSOUT=*
//SMFPRNT DD SYSOUT=*
//*
```

BEST PRACTICE:

The SMF Extractor must run at a high Dispatching Priority (DP). It must be as close to the SMF DP as possible. The recommended DP is SYSSTC (x'FE') since SMF usually runs at SYSTEM (x'FF') DP.

The following values should be used in the SMF Extractor JCL:

- REGION=0M (on either the JOB or EXEC statement)
- MAXQD can be updated from the default of 4000 to a maximum of 9999, if necessary, when the SMF Extractor runs out of queue space. The value should be increased incrementally to balance functionality with available resources. A value of 4000 is usually enough for a very busy LPAR.

On the SMFREC parm in the SMF Extractor control file, only select the SMF record types that are required for the IZPCA components you have installed. You can do this by selecting option 2 Components from the IBM Z Performance and Capacity Analytics Administration menu, type / to select a component, press Enter, then look for the RECORD object types. Selecting every record type introduces unnecessary performance overhead.

If you are experiencing problems with the SMF Extractor, you may wish to turn on diagnostics by doing the following:

specify TRACE(Y) and SDUMP(Y) on the EXEC card.

```
CKKXMAI EXEC PGM=CKKXMAI,REGION=0M,
PARM='TRACE(Y),MSGlvl(9),SDUMP(Y),MAXSD(2),MAXQD(4000)'
```

IZPCA does not use the original SMF logs created daily through your normal SMF processing (MAN logs, GDGs, and so on). Those are retained as per your normal operational procedures.

Customizing the Continuous Collector startup

If you plan to run in zIIP mode, you must add the SDRLOAD data set to your list of APF authorized data sets. For example:

```
SETPROG APF,ADD,DSNAME= h/q.SDRLOAD,SMS
```

A sample member DRLJCCOL is provided in the *h/q.SDRLCNTL* library. It contains the JCL for a started task to collect SMF data continuously. Customize the sample JCL by following the instructions in the comments. The sample JCL member is included in [Appendix E](#).

For the Continuous Collector, consider the values for the following 4 options on the COLLECT statement:

- COMMIT AFTER n MINUTES
The Continuous Collector commits to Db2 at regular intervals. The interval is set by this option of the COLLECT statement. The possible values are 1, 2, 5, 10, 15, 20, 30 and 60.
- FULL STATISTICS AFTER n COMMITS
By default, the collect statistics are written to the job log after each Db2 commit. Producing a statistics report after every commit in a continuous environment will have an impact on the volume of spool used. To limit the volume of output you may choose to write the collect statistics less frequently. This option of the COLLECT statement enables you to only write statistics after a specified number of commits have taken place. Alternatively, DRLOUT may be assigned to DASD.
- BUFFER SIZE nnn M
Small collect buffers result in waits for I/O, which contribute to increased CPU usage. Specify a buffer size of 1300M if available. In SDSF, the command VMAP shows the size of the EXTENDED-PRIVATE area. Reduce this by 100M and this is the upper maximum for buffer size.
- ZIIP
Much of the processing for IZPCA can be offloaded to a zIIP processor by adding the ZIIP option to the COLLECT statement. This reduces costs and reduces the load on your CP processors.

BEST PRACTICE:

The collector stores records in an internal buffer and updates that buffer with subsequent matching records then writes these to Db2 at Commit time. After a commit if there are additional records that match a committed record, the record is read back into the buffer, incremented with the matching records, and updated in Db2 at commit.

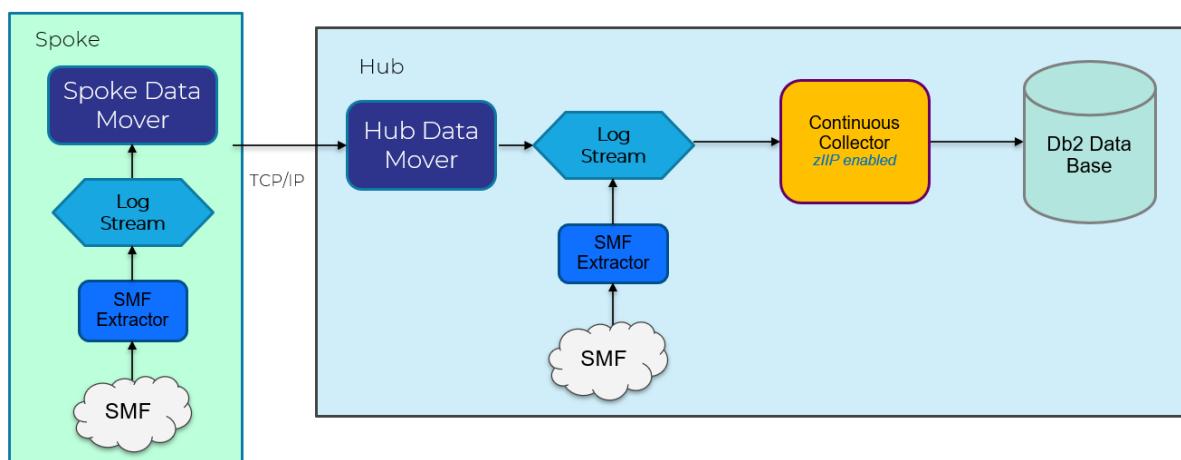
It is beneficial to make the buffer as large as possible, thus reducing the amount of work done in Db2 and the number of Db2 log entries generated.

- Running the collector in zIIP mode:
 - ❖ Using large collect buffers will reduce General CP usage and provide increased offload to zIIP processors, reducing the consumption of MIPS/MSUs. Locating on a machine that has additional capacity and zIIP engines will be beneficial.
 - ❖ The zIIP process is interrupted during the commit phase, hence more zIIP percentage is achieved with a larger buffer. A smaller buffer size means dropping out of zIIP to process the Db2 updates.
 - ❖ While running in zIIP mode reduces General CP usage the elapse times for batch collects may increase.

Hub and Spokes (multi-system)

Skill required: z/OS Systems Programmer; UNIX knowledge.

When multiple z/OS systems send data to a single IZPCA system for collection to a Db2 Database, this is referred to as a Hub and Spoke configuration. SMF Extractor data is sent between z/OS systems using DataMovers (DM). In this configuration, each remote system (Spoke) requires an SMF Extractor which writes SMF data to a DASD log stream which is forwarded to the HUB using a DataMover (referred to as the *data sender*). The Hub DataMover (referred to as the *data receiver*) writes the data from the Spoke system(s) to a local DASD log stream which the Continuous Collector reads and commits updates the IZPCA Db2 database.



Setting up the DataMover on the Hub

Before you begin

Before setting up the Spoke systems:

Ensure you have completed the [standalone \(Single system\)](#) configuration.

Note: Both the Hub SMF Extractor and Spoke DataMover write to the same log stream on the Hub.

About this task

Set up a DataMover on the Hub to receive the Spoke data and store it in a log stream that will be read by the Continuous Collector.

Procedure

Follow these steps to set up the DataMover on the Hub system.

1. On the hub system, in OMVS, locate the source directory created when IBM Z Performance and Capacity Analytics 3.1.0 was installed.

The standard installed location is /usr/lpp/IBM/IZPCA/v3r1m0/IBM/. If not in that location, issue the command **df | grep IZPCA** to find the directory.

The output from the command will be similar to this example, noting that the high-level qualifier for the data sets (IZPCA.V3R1M0) will be different for your installation:

```
/u/izpca/REG/V310/usr/lpp/IBM/IZPCA/v3r1m0 (IZPCA.V310.ZFS) 32094/50400  
4294967271 Available
```

If the source directory does not exist, it needs to be mounted.

The TSO command to mount the filesystem read-only is:

```
MOUNT FILESYSTEM(prefix.zfs) TYPE (ZFS) MODE(READ) MOUNTPOINT(local_mountpoint)
```

2. Create an OMVS directory for the DataMover based on your installation standards.

For example, create the OMVS directory /var/IZPCA using the mkdir command:

```
mkdir /var/IZPCA
```

3. Install the DataMover. To do this, extract the DataMover using the command:

```
tar -xvof /usr/lpp/IBM/IZPCA/v3r1m0/IBM/DRLPJDM -C /var/IZPCA/
```

where

- /usr/lpp/IBM/IZPCA/v3r1m0/IBM/DRLPJDM is the source directory/file for IBM Z Performance and Capacity Analytics
- /var/IZPCA is where you want the contents of the tar file to reside, and which is your destination directory for the DataMover

This creates the following directories:

```
/var/IZPCA/DataMover  
/var/IZPCA/DataMover/config  
/var/IZPCA/DataMover/java  
/var/IZPCA/DataMover/mappings
```

4. To run the DataMover, a user needs write permission on the DataMover directory. Change permissions using either of the following methods:
 - Use the change mode of access command:
chmod 770 /var/IZPCA/DataMover
 - Give the user running the DataMover ownership of the directory /var/IZPCA/DataMover
5. Tailor the DataMover.sh file in your /var/IZPCA/DataMover directory to your system requirements. A sample is supplied in [Appendix I](#) with the parts that require change highlighted in bold.
6. Create a DataMover Hub properties configuration file for the Hub DataMover in the DataMover config directory /var/IZPCA/DataMover/config. This is an example of a Hub DataMover configuration file:

```
# File name: hub.properties
#
# 5698-AS3 Copyright Teracloud ApS 2018, 2023
#
# 21st Century Software DataMover (Hub)
#
# You need to modify:
#
#   The port the input TCPIP stage will listen on
#   The name of the Log stream in the output stage
#
routes = 1
route.1.name = Hub
#
# Listen for connections from the Spokes and receive their data.
#
# If you want to use SSL you need to perform the certificate
# exchange before you change the value to YES
#
input.1.type = TCPIP
input.1.port = 54020
input.1.stats_interval = 15
input.1.use_ssl = no
#
# For output, we write the data into the log stream
#
outputs.1 = 1
output.1.1.type = LOGSTREAM
output.1.1.logstream = IFASMF.DASD.LS01
output.1.1.sync = YES
#
```

BEST PRACTICE when applying maintenance:

After maintenance, when extracting the DataMover with the tar command, extract to a different directory to avoid overwriting any changes you may have made. Once extracted, copy the file you need to the original directory. For example, the DataMover.jar file.

Setting up the spoke systems

About this task

Set up an SMF Extractor and DataMover on each Spoke to collect the SMF data and transmit it to the Hub.

Procedure

1. Copy required software to each of the Spoke locations:

- The SMF Extractor load library *hlq.SDRLEXTR*.

Use TSO XMIT to transfer this data set to the target Spoke system, then TSO RECEIVE to receive the data set on the target system.

- The DataMover and configuration file in the z/OS UNIX directory
/u/izpca/REG/V310/usr/lpp/IBM/IZPCA/v3r1m0/IBM/DRLPJDM.

Create a directory on the target Spoke system to contain the file DRLPJDM.

For example:

```
cd usr           swap to user directory
mkdir izpcawork  create an IZPCA work directory
chmod 770 izpcawork  change the permissions on the work directory
```

Copy the file to this new Spoke directory using SCP as follows:

```
ssh <userid>@<hub node>
scp <source file> <userid>@<spoke node>:<target file>
```

For example:

```
scp /u/izpca/REG/V310/usr/lpp/IBM/IZPCA/v3r1m0/IBM/DRLPJDM
      userid@node:/usr/izpcawork/DRLPJDM
```

Alternatively, you can use Filezilla if you have this product.

2. Extract the DataMover following these steps:

- a. Create a target directory on each Spoke system where you will run the DataMover from.

For example:

```
mkdir /var/IZPCA
```

- b. Change the permissions on the work directory to allow write access:

```
chmod 770 /var/IZPCA
```

where /var/IZPCA is where you want the contents of the tar file to reside, and which is your destination directory for the DataMover.

- c. Run the extract for the DataMover directories using the command:

```
tar -xvf /usr/izpcawork/DRLPJDM -C /var/IZPCA/
```

3. Set up a DASD log stream on each Spoke system. Refer to the standalone system process for details.

4. Set up an SMF Extractor on each of the Spoke systems selecting the record types that are required. Refer to the standalone system process for details.

5. Set up a DataMover on each Spoke system.

If in a sysplex, create separate DataMover directories for the Spokes. Do not use the same DataMover directory for multiple Spokes as this will adversely impact the buffering.

This DataMover will read and transmit the log stream data to the Hub.

Tailor the DataMover.sh file in your /var/IZPCA/DataMover directory to your system requirements. A sample is supplied in [Appendix I](#) with the parts that require change highlighted in bold.

Create a DataMover Spoke properties file in the DataMover config directory. This is an example of a Spoke DataMover configuration file:

```
# File name: spoke.properties
#
# 5698-AS3 Copyright Teracloud ApS 2018, 2023
#
# 21st Century Software DataMover (Spoke)
#
# You need to modify:
#   The name of the Log stream in the INPUT stage
#   The hostname in the output TCPIP stage
#   The port in the output TCPIP stage
#
routes = 1
route.1.name = Spoke
#
# Our input arrives in a log stream, put there by the SMF Extractor
# This will take data from the log stream and
# then erase it once sent.
#
input.1.type = LOGSTREAM
input.1.logstream = IFASMF.DASD.LS02
input.1.block = 30
input.1.wipe = YES
input.1.checkpoint = NO
input.1.strip_header = NO
input.1.check_LGRH = YES
input.1.sourcename = LOGSTREAM
input.1.sourcetype = RAWSMF
#
# For output, we send the data to the hub
#
# If you want to use SSL you need to perform the
# certificate exchange before you
# change the value to YES
#
# The directory is used to buffer packets for later transmission
# if the link drops out. If you decide to use a path rather than
# buffer, use an absolute path (output.1.1.directory = /var/IZPCA).
# Ensure you have a unique directory for each Spoke.
#
```

```

outputs.1 = 1
output.1.1.type = TCPIP
output.1.1.host = HUB.com
output.1.1.port = 54020
output.1.1.use_ssl = NO
output.1.1.stats_interval = 15
output.1.1.directory = buffer
output.1.1.use_buffer = NO
#

```

PROC and JCL to run the DataMover(s)

Add a DataMover procedure to your PROCLIB on each system where a DataMover will be implemented:

- Customize the procedure to suit your system requirements.
- Store the started task proc in a procedure library (PROC library) on your system.
- Add the started task to the RACF class STARTED. If not using RACF, add the started task to your security system

The following example is supplied in *hlq.SDRLCNTL(DRLJDMP)*.

```

//** IBM Z Performance and Capacity Analytics DataMover
//** 5698-AS3 (C) Copyright Teracloud S.A. 2018.
//*
//DRLJDMP PROC CONFIG='config'
//*
// * Set to the directory you installed the DataMover into
// *
// SET WDIR='/var/IZPCA/DataMover'
//*
//RUNDM EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
// PARM=('PGM /bin/sh &WDIR./DataMover.sh &CONFIG')
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//*
//      PEND

```

To invoke the DataMover using the DataMover procedure, use the JCL example:

```

//DATAMVR JOB (999,XXX), 'DM', CLASS=B, MSGLEVEL=(1,1),
//           MSGCLASS=X, REGION=0M, NOTIFY=&SYSUID
//PROCLIB JCLLIB ORDER=your.PROCLIB
//-----
// * Change the CONFIG parm to your Hub or Spoke
// * configuration file.
//-----
//RUN EXEC PROC=DataMover,CONFIG='hub'
/*

```

Setting up the Continuous Collector

Once the SMF Extractors and DataMovers are set up, the Continuous Collector can be set up on the Hub system.

A sample member DRLJCCOL is provided in the SDRLCNTL library. It contains the JCL for a started task to collect SMF data continuously. Customize the sample JCL by following the instructions in the comments. The sample JCL member is included in [Appendix E](#).

For the Continuous Collector, consider the values for the following 4 options on the COLLECT statement:

- COMMIT AFTER n MINUTES

The Continuous Collector commits to Db2 at regular intervals. The interval is set by this option of the COLLECT statement. The possible values are 1, 2, 5, 10, 15, 20, 30 and 60.

- FULL STATISTICS AFTER n COMMITS

By default, the collect statistics are written to the job log after each Db2 commit. Producing a statistics report after every commit in a continuous environment will have an impact on the volume of spool used. To limit the volume of output you may choose to write the collect statistics less frequently. This option of the COLLECT statement enables you to only write statistics after a specified number of commits have taken place. Alternatively, DRLOUT may be assigned to DASD.

- BUFFER SIZE nnn M

Small collect buffers result in waits for I/O, which contribute to increased CPU usage. Specify a buffer size of 1300M if available. In SDSF, the VMAP command shows the size of the EXTENDED-PRIVATE area. Reduce this by 100M and this is the upper maximum for buffer size.

- ZIIP

Much of the processing for IZPCA can be offloaded to zIIP processor by adding the ZIIP option to the COLLECT statement. This will reduce costs and reduce the load on your CP processors.

BEST PRACTICE:

The collector stores records in an internal buffer and updates that buffer with subsequent matching records then writes these to Db2 at Commit time. After a commit, if there are additional records that match a committed record, the record is read back into the buffer, incremented with the matching records, and updated in Db2 at commit.

It is beneficial to make the buffer as large as possible, thus reducing the amount of work done in Db2 and the number of Db2 log entries generated.

- Small batch collect buffers result in waits for I/O, which contribute to increased CPU usage. Specify a buffer size of 1300M if available. In SDSF, the VMAP command shows the size of the EXTENDED-PRIVATE area. Reduce this by 100M and this is the upper maximum for buffer size.
- Running the collector in zIIP mode:

- ❖ Using large collect buffers reduces General CP usage and provides increased offload to zIIP processors, reducing the consumption of MIPS/MSU. Locating on a machine that has additional capacity and zIIP engines will be beneficial.
- ❖ The zIIP process is interrupted during the commit phase, hence more zIIP percentage is achieved with a larger buffer. A smaller buffer size means dropping out of zIIP to process the Db2 updates.
- ❖ While running in zIIP mode reduces General CP usage, the elapse times for batch collects may increase.

Maintenance recommendations

Maintaining the IBM Z Performance and Capacity Analytics database includes the following tasks:

- Purging unneeded data
- Reorganizing the database
- Updating Db2 statistics using the RUNSTATS and STOSPACE utilities
- Backing up data
- Updating views on the Db2 catalog
- Protecting the integrity of data by controlling the access.

Regular maintenance tasks are as follows:

Execute the RUNSTATS utility periodically while the database is growing to:

- Provide the Db2 optimizer with information. (After the database stabilizes, RUNSTATS does not make a significant contribution to the Db2 optimizer.)
- Provide table size statistics for IBM Z Performance and Capacity Analytics.

Improving COLLECT performance

The following practices and tuning tasks can be applied to the COLLECT process to improve performance:

1. Optimize the collect buffer size.

Optimizing the size of the collect buffer has the greatest impact on performance.

- a. Reduce the number of times IBM Z Performance and Capacity Analytics stops reading a log data set to write data to the database by increasing the buffer size.

Message DRL0313I shows the number of database updates because of a full buffer. Look for cases where the number of updates could be reduced by increasing the size of the buffer.

For more information, refer to the [COLLECT statement](#) in the *Language Guide and Reference*.

- b. Do not use the COMMIT AFTER *nn* records operand on the COLLECT statement. This will not give the most efficient use of the buffers.

- 2. Reduce the amount of data committed to the database.
 - a. Only install the components you need.
 - b. Remove unnecessary tables using the INCLUDE/EXCLUDE clauses of the COLLECT statement.
 - c. Examine collect messages to determine the most active tables.
 - d. Concentrate on tables with a lot of buffer and database inserts and updates shown in DRL0326I messages.

Note: With Db2 multiple insert functionality, when data is collected to data tables, the insert statements are issued in bulk. Multiple rows are inserted with a single Db2 multiple insert statement. This results in significant performance improvements. However, this performance improvement decreases as the number of columns inserted increases.

Summary of implementation recommendations

The following table brings together the recommendations that should be reviewed and where appropriate, adopted during implementations.

Topic	Description	Action	Comments
IZPCA	Dedicated Db2 Subsystem	Install on dedicated Db2	Avoid running other business application in this Db2 subsystem. This can cause significant issues with setting the Db2 ZPARMs to suitable values for IZPCA.
	Maintenance	Apply latest maintenance	This should be completed on a regular basis to ensure forward compatibility of new features. Ensure you comply with all HOLD actions.
Continuous Collector	Match SMF Interval or set to 15 mins.	Set COMMIT time to 15 mins	If you set the Commit time too low it may introduce serious performance issues.
Batch Collector	Small batch collect buffers results in waits for I/O which contribute to increased CPU usage	Set buffer size to 1300 M	Subject to storage availability. Non-Production install was limited by available memory and ran with a much smaller buffer size.
	Use zIIP with large collect buffers.	Use zIIP command in Batch collector	Requires zIIP engine availability. Non-Production install did have zIIP collect enabled.
Db2 Management	Manage the database size; discard data no longer relevant	Review settings for DRLPURGECOND	Update table to reflect Data Governance and Compliance.
	Run Purge or Discard and Reorg maintenance jobs regularly	Add DRLJPURG to your batch scheduler	Old data must be discard to maintain a reasonable amount of data. Ensure Db2 Active logs are very large as the reorg process deletes millions of rows and can use a lot of db2 log space while running.
	RUNSTATS and STOSTATS	Run RUNSTATS and STOSTATS regularly	Optimizes Db2 and updates Db2 statistics.
Db2 Params	Make use of Db2 lock escalation	Set NUMLKTS parameter to 1000	Non-Production install may have this parameter set very low (eg 10) to prevent abends however this will cause a huge number of messages relating to lock escalation in the DB2 master log.

		Set NUMLKUS parameter to 500,000	Non-Production install may have this parameter set low (eg 10,000) resulting in abends due to maximum locks per user being breached.
		Set URLGWTH value to 100(k)	Reduces messages.
	Purging data.	Set REORG_DROP_PBG_PARTS=DISABLE	Needed to avoid issues with reorg.
HUB DataMover	Provides statistics every 5 mins	input.1.stats_interval = 15 minutes	Helps with identifying issues with Continuous Collection.
Spoke DataMover	Use log streams as buffers	output.1.1.directory = buffer output.1.1.use_buffer = NO	Avoids using z/OS UNIX. Leaves data in the log stream rather than copying it into the file system until the file is full..
SMF Extractor	Include trace options	TRACE(N) and SDUMP(N) on the EXEC card. CKKXMAI EXEC PGM=CKKXMAI,REGION=0M, PARM='TRACE(Y),MSGlvl(9),SDUMP(Y), MAXSD(2),MAXQD(4000)'	Allows you to capture additional information when required.

Cognos implementation

Skill required: IZPCA Administrator with SYSADM authority; Cognos Administrator.

The following details provide a quick guideline to assist with the implementation of Cognos 11.1.7. For a more advanced installation, go to the following link:

[How to Install Cognos Analytics 11.1.x with Distributed server installation \(ibm.com\)](#)

Cognos can be installed in either Windows or Linux. However, note that the Cognos Framework Manager is only supported in Windows. The component is used to manage the publication of Cognos reports and is only required periodically, such as during maintenance to Cognos. It can be installed elsewhere if the network access and credentials are in place. If it is to be installed on the same machine as the Cognos Server, then this would need to be Windows.

Server machine specifications

The following specifications are recommended requirements. They should be reviewed post-implementation and tuned appropriately.

- A minimum of 10 GB of free space is required to install the software and 4 GB of free space on the drive that contains the temporary directory used by IBM Cognos Analytics components.

Ensure that you have sufficient disk space for future requirements, as by default each user can use 500 MB of disk space for uploaded files.

- A minimum requirement of 32 GB for RAM is recommended.
- For each deployment, a sizing exercise is highly recommended.

Overview of Cognos Installation

For a Windows installation, make sure to install on C: drive. There is a known issue with the Cognos Easy Install option in Windows. If you change the installation drive from C: to another drive, then the installation fails when trying to install the local Informix database.

The user installing Cognos must have the relevant Windows Administration privileges. Also, note that all relevant network and port access credentials need to be granted prior to initiating the installation process.

IBM ShopZ - Downloading Cognos

For details of minimum server and software license requirements see:

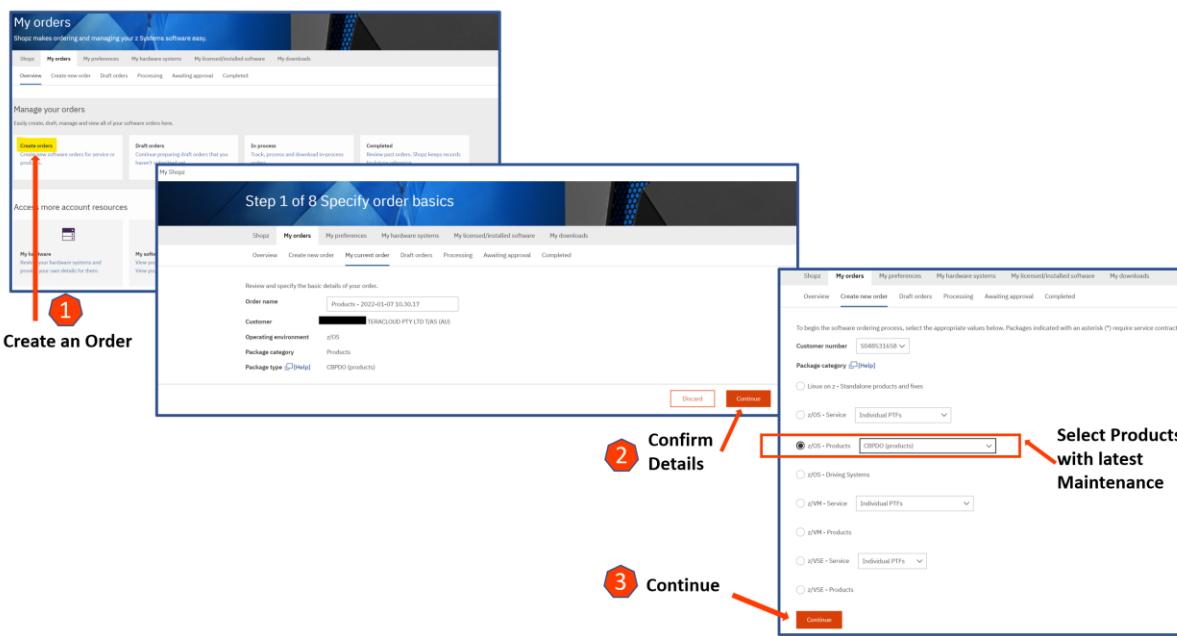
[IBM Cognos Analytics on Premises 11.1.x: Supported Software Environments](#)

The Cognos installable files are delivered to customers via the IBM ShopZ Orders for the IZPCA product. When a customer orders IZPCA they get a limited use license of Cognos. The install files are available in ShopZ. This can be confusing as it is often the case that distributed teams would be designated as the owners for Cognos as it is a Windows/Linux solution. Distributed teams would not be familiar with ShopZ as this is commonly administered by the Mainframe team.

To download the install files, the ShopZ Administrator needs to locate the media in the ShopZ order under the Section entitled “Additional Licensed Publications”.

The following diagrams depict how to order IZPCA and once the order delivery email is received, then how to locate the install media.

ShopZ Ordering IZPCA



Create an Order

Step 1 of 8 Specify order basics

2 Confirm Details

3 Continue

Select Products with latest Maintenance

Step 2 of 8 Select hardware systems

Hardware systems for customer number 5048531658

4 Confirm CPU Model

Step 3 of 8 Report installed software

5 Continue

Step 4 of 8 Shop for products

6 Select PRODUCT DESCRIPTION and search for "Capacity Analytics"

7 Select "Z Perf Capacity Analytics" and Continue

Step 5 of 8 Specify order contents

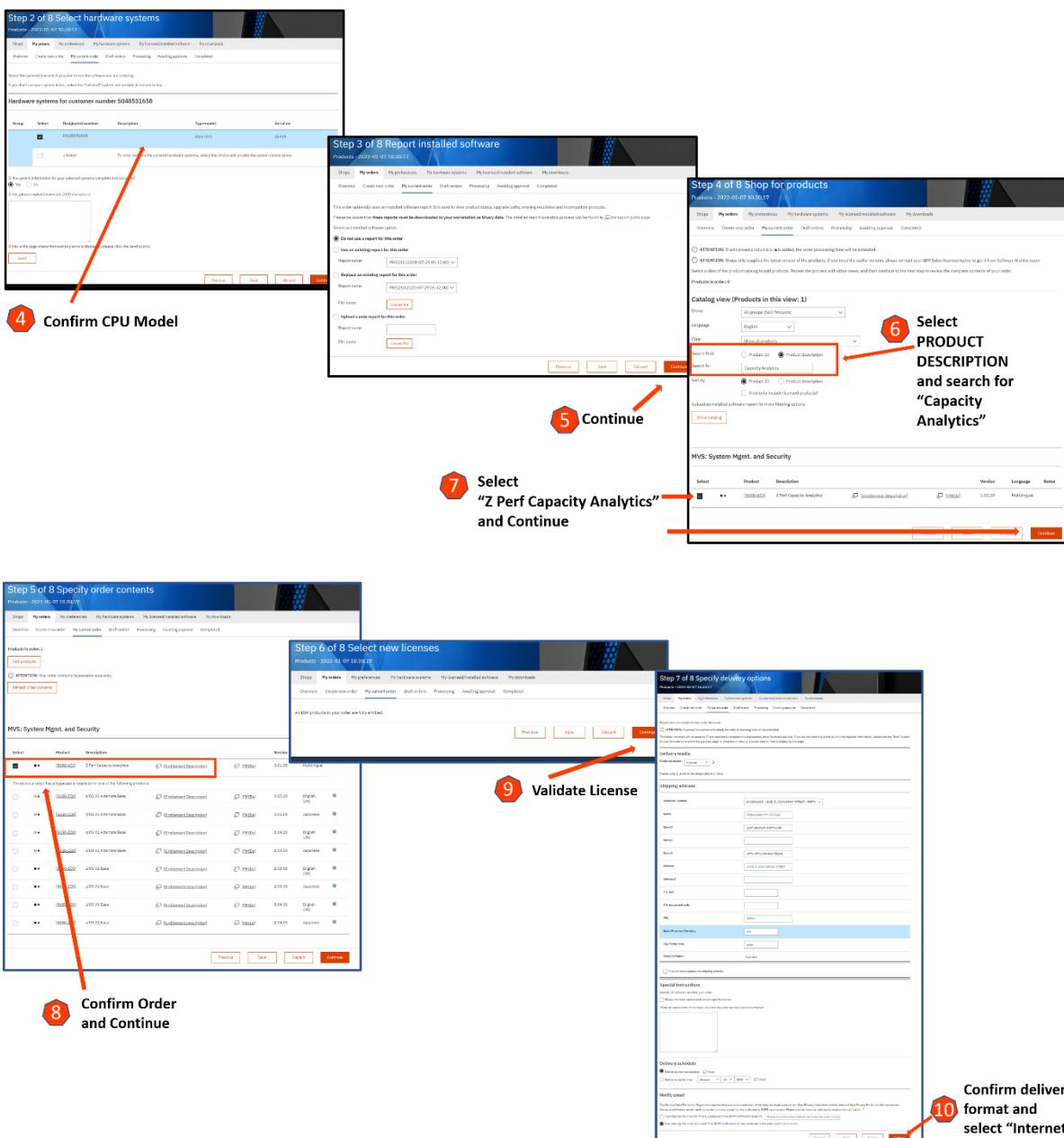
8 Confirm Order and Continue

Step 6 of 8 Select new licenses

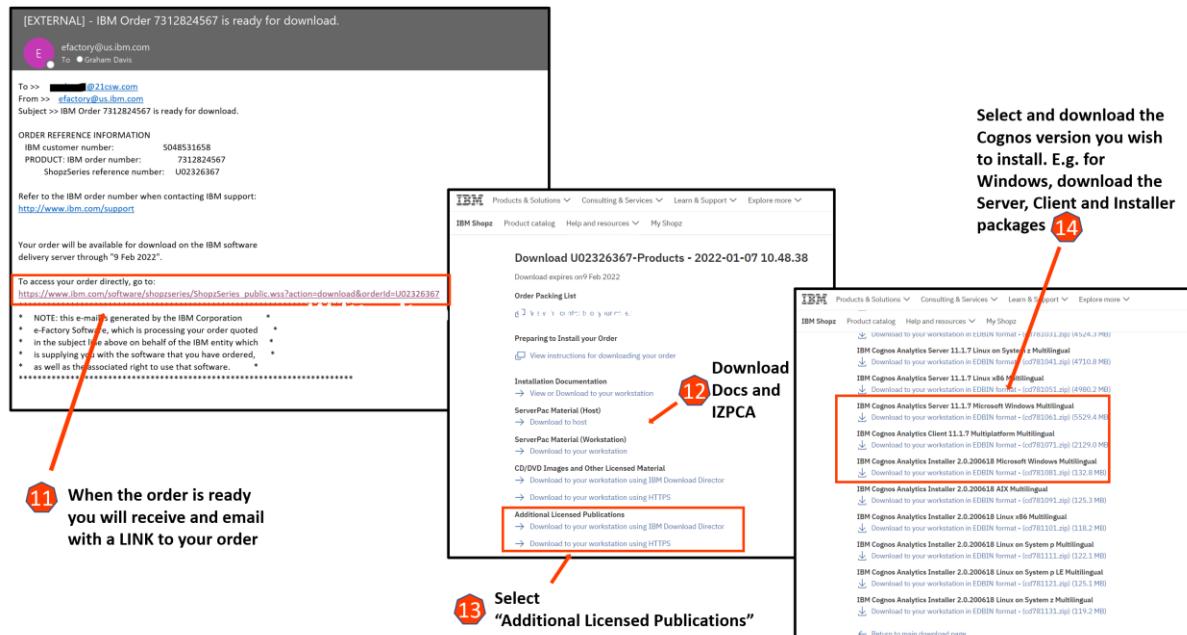
9 Validate License

Step 7 of 8 Specify delivery options

10 Confirm delivery format and select "Internet"



ShopZ Locating the Cognos Media



The screenshot illustrates the process of locating Cognos media files. It shows three main steps:

- 11** When the order is ready you will receive and email with a LINK to your order (red arrow pointing from the email screenshot to the download link in the Order Packing List).
- 12** Select Docs and IZPCA (red arrow pointing from the "Additional Licensed Publications" section to the "Download" button).
- 13** Select "Additional Licensed Publications" (red arrow pointing from the "Additional Licensed Publications" section to the "Download" button).
- 14** Select and download the Cognos version you wish to install. E.g. for Windows, download the Server, Client and Installer packages (red arrow pointing from the "Download" button to the list of available downloads).

Select and download the Cognos version you wish to install. E.g. for Windows, download the Server, Client and Installer packages **14**

Media Files – additional considerations

There are three files:

IBM Cognos Analytics Server 11.1.7 - cd781061.zip

IBM Cognos Analytics Client 11.1.7 - cd781071.zip

IBM Cognos Analytics Installer 11.1.7 - cd781081.zip

Copy these files to your Cognos Server machine. The Server and Client file MUST remain zipped. Unzip the Install file and execute it. You will be guided through the installation.

Installing Cognos in Windows

The following IBM links guide you through installing Cognos and Framework Manager.

Install Cognos using the Easy Install option:

<https://www.ibm.com/support/pages/how-install-cognos-analytics-111x>

Install and configure the Framework Manager:

[IBM Cognos Framework Manager - IBM Documentation](#)

Post Cognos installation tasks

Download the JDBC drivers from the Db2 for z/OS files location.

These two files should be downloaded in binary to the Cognos Driver folder. For example:
C:\Program Files\ibm\cognos\analytics\drivers

- db2jcc4.jar
- db2jcc_license_cisuz.jar

You can find these files in z/OS UNIX under /usr/lpp/db2/db2c10/jdbc/classes or similar depending on your mount point. Look at the mount table in ish to figure out the location.

After these drivers have been placed in the Cognos Drivers folder, the server will require a restart to take effect.

Define the Data Server Connection

Before you can run any reports, you need to connect Cognos to the IZPCA database on z/OS.

You will need the following from the Db2 Master address space. You may need to contact your Mainframe Db2 Administrator and Systems Programmer to get this information:

- IP address of z/OS
- Port number for DDF
- Location of DDF (DDF is case sensitive and must be in uppercase)
- A userid with password that does not expire
- This userid to be connected to RACF groups DRL and DRLSYS if you used the default DRLDB data base, otherwise connect to the schemas that you used

In the browser, go back to the Manage options and select Data Server Connections.

Create a new connection called **IZDS** (do not change the name, because if you do, you will need to customize other things as well).

Select type *IBM Db2*

Select **Use the following signon** as Authentication Method and specify the userid and password for the IZPCA Db2 access.

Click Edit on the **Connection Details** and specify the JDBC URL in the following format:

jdbc:db2://10.3.20.82:4740/MOPDBC0

and change the ip address, the port, and the location (uppercase) to suit your installation.

Click **Close** and **Save**.

Click **Test** and it should result in success.

Installing the IZPCA Report Package

Follow the instructions detailed in [Implementing Cognos reporting](#) in the *IBM Z Performance and Capacity Analytics Guide to Reporting*.

Verifying Cognos installation for IZPCA reporting

To verify the Cognos installation for IZPCA reporting, follow these steps:

1. Ensure that the IZPCA component is installed, and data has been collected into the Db2 tables to support the reports.
2. Logon to Cognos at <http://your-ip-address:9300/bi>
3. Select Team Content.
4. Select IBM Z Performance and Capacity Analytics.
5. You will see subsequent folders and reports. Run a selection of these reports as required.

Appendix A. Sample JCL to create a new CSI

```
//DEFCSI1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
  DEFINE CLUSTER(NAME(IZPCA.V310.CSI)
                 FREESPACE(20 5)           -
                 KEYS(24 0)              -
                 RECORDSIZE(24 143)       -
                 SHAREOPTIONS(2,3)        -
               DATA(NAME(IZPCA.V310.CSI.DATA)  -
                     CONTROLINTERVALSIZE(8192) -
                     CYLINDER(20,20))        -
               INDEX(NAME(IZPCA.V310.CSI.INDEX) -
                     CONTROLINTERVALSIZE(4096) -
                     TRACK(15 1))
/*
//PRIMCSI2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SMPCSI   DD DSN=IZPCA.V310.CSI,DISP=SHR
//ZPOOL    DD DSN=SYS1.MACLIB(GIMZPOOL),DISP=SHR
//SYSIN   DD *
  REPRO OUTFILE(SMPCSI)  INFILE(ZPOOL)
/*
```

Appendix B. Sample JCL to create your user.DRLFPROF data set

```
*****
/*
/* 5698-AS3 */
/* Copyright Teracloud ApS 2018, 2022 */
/* and IBM Corp. 1993, 2015 */
*/
*****
```

/* This data set contains all the customization parameters that */
/* must be modified if the user does not use the default values. */
/* The alphanumeric parameters must be specified between " " */
/* characters. */

```
*****
drl_grant      = "YES"           /* Allow/Disallow GRANTS @05A */  

drlshwid       = "NO"            /* Show ID on top level menus */  

drltenu      = "DRLvrm.SDRLTENU" /* TDS tables library @D3c */  

qmfds_tlib     = ""              /* QMF tables library if PTR542*/  

bkmgr_tlib     = ""              /* BookManager tables library */  

                                /* PTR542*/  

modtenu        = ""              /* User tables library PTR542*/  

def_db2subs   = "DSN"          /* DB2 subsystem */  

def_db2plan   = "DRLPLAN"      /* DB2 plan name @01A */  

def_dbname    = "DRLDB"         /* DB2 database name */  

def_storgrp   = "DRLSG"         /* Storage group name */  

def_syspref   = "DRLSYS"        /* System table prefix */  

def_othtbpx   = "DRL"           /* Other tables prefix */  

def_drlshwid  = "YES"          /* Show TDS system info @01A */  

def_tsbpool   = "BP0"          /* Bufferpool for data */  

def_ixbpool   = "BP0"          /* Bufferpool for indexes */  

def_lobbpool  = "BP32K"         /* Bufferpool for blob @06A */  

def_iduser1   = "DRLUSER"       /* Default user group id */  

def_idsqlusr  = "DRLUSER"       /* SQL id in QMF */  

def_rptdialg  = 1              /* reporting dialog mode */  

                                /* 1 = enduser (default) */  

                                /* 2 = administrator */  

def_qmflng   = "PROMPTED"     /* QMF language (SQL/PROMPTED) */  

def_qmfprt   = "Q"             /* QMF SYSOUT class */  

def_drlprt   = "A"             /* Print SYSOUT class for batch*/  

                                /* tabular reports w/o QMF */  

def_pagelen  = 60             /* Lines/page when printing */  

                                /* tabular reports w/o QMF */  

def_drlmax   = 5000           /* SQLMAX value (max no of rows*/  

                                /* retr. by PR/DB2* NTERrface */  

                                /* for ISPF/PDF* function) */  

def_printer   = ""              /* Default graph. printer */  

def_db2dspfx  = "DB2.VA10"     /* DB2 dataset prefix @D4C */  

def_db2dssfx  = ""              /* DB2 dataset suffix */  

def_qmfdsdspfx = "DB2.VA10"     /* QMF dataset prefix @D4C */  

def_dsnpref   = "DRLvrm"        /* TDS dataset prefix @D3c */  

def_dsnlocdn  = "DRL.LOCAL.DEFS" /* Loc def.for DB2V4 */  

def_usrlocdn  = "DRL.LOCAL.USER.DEFS" /* User Defs */  

def_modform   = "DRL.LOCAL.ADMCFORM" /*Local GDDM forms */  

def_dsnreprt  = "DRL.LOCAL.REPORTS" /*Local rep.dataset */  

def_dsnchrt   = "DRL.LOCAL.CHARTS" /*Local charts ds */  

def_drlmsgs   = "DRL.LOCAL.MESSAGES" /*Dataset for msgs */  

-----*/  

/* The following variables are set each time a user starts the */  

/* dialog */  

-----*/
```

```
-----*/
```

```

/* BookManager datasets */
/*-----*/
bkmgr_mlib = ""          /* BookManager message library */
bkmgr_plib = ""          /* BookManager panel library */
bkmgr_tlib = ""          /* BookManager tables library */
bookshlf_ds = ""         /* Bookshelf dataset */
/*-----*/
dsnsufx = "SDRLDEFS"    /* Definition dataset suffix */
execsfx = "SDRLEXEC"    /* Exec dataset suffix */
loadsfx = "SDRLLOAD"    /* Load dataset suffix */
skelsfx = "SDRLSKEL"    /* Skeleton dataset suffix */
/*-----*/
/* Japanese langauge is no available with           8@04C */
/* IBM Z Performance and Capacity Analytics        */
/* The variables jpnlsfx and jpn_qmf_sfx are kept for */
/* compatability with DRLFPROF from earlier        */
/* products and versions but are set to English   */
/* NOTE:                                            */
/* def_diallang will default to 1 in the DRLEINII EXEC */
/* */
/*-----*/
englsfx = "ENU"          /* English default libr suffix */
jpnlsfx = "JPN"          /* Jap. compatibilty only @04C */
def_diallang = 1          /* Def. dialog lang is English */
/*-----*/
repsufx = "SDRLR"        /* Report dataset suffix */
plibsfx = "SDRLP"        /* Panel library suffix */
messsfx = "SDRLM"        /* Message library suffix */
formsfx = "SDRLF"        /* FORM library suffix */
tabsfx = "SDRLT"         /* TABLES library suf */
/*-----*/
eng_qmf_sfx = "E"        /* English default libr suffix */
jpn_qmf_sfx = "K"        /* Jap. compatibilty only @04C */
/*-----*/
/* Change the following QMF language_dependent identifiers to */
/* reflect values in use at your site. P.R. uses these values to */
/* dynamically allocate QMF and make it available for P.R. users. */
/* Note: You can use QMF library suffixes if your site follows the */
/* recommended naming standard for QMF data sets. */
/* Otherwise, use fully-qualified names to identify QMF data sets. */
/* Change <qmfdef> variable to "DATASET" to use fully-qualified */
/* data set names; retain the value "SUFFIX" to use the standard */
/* QMF naming convention. */
/* Note: P.R. adds a 1-char language ID to each value below if you */
/*       use "SUFFIX" but does not if you use "DATASET". */
/*-----*/
qmfdef = "SUFFIX"        /* QMF library suffixes are used */
                         /* Change this var. to "DATASET" */
                         /* If you define fully qualified */
                         /* dataset names below */
qmfclib = "SDSQCLT"      /* QMF CLIST library suffix */
                         /* or fully qualified dataset name */
                         /* of QMF CLIST library */
qmfclibe = "SDSQCLT"     /* ENG. QMF CLIST library suffix */
                         /* or fully qualified dataset name */
                         /* of QMF CLIST library */
qmfelib = "SDSQEXC"      /* QMF EXEC library suffix */

```

```

                /* or fully qualified dataset name */
                /* of QMF EXEC library */
                /* ENG. QMF EXEC library suffix */
                /* or fully qualified dataset name */
                /* of QMF EXEC library */
                /* QMF Panel library suffix */
                /* or fully qualified dataset name */
                /* of QMF Panel library */
                /* QMF Message library suffix */
                /* or fully qualified dataset name */
                /* of QMF Message library */
                /* QMF Skeleton library suffix */
                /* or fully qualified dataset name */
                /* of QMF Skeleton library */
                /* ADMGGMAP library suffix */
                /* or fully qualified dataset name */
                /* of ADMGGMAP library */
                /* QMF Help Panel library suffix */
                /* or fully qualified VSAM dataset */
                /* name of QMF DSQPNLx libr */
                /* DDname of QMF DSQPNLx library */
                /* QMF Tables library suf.
                -----
                /* Change the following QMF identifiers to reflect values in use at */
                /* your site. P.R. uses these values to dynamically allocate QMF */
                /* and make it available for P.R. users. */
                /* Note: You can use QMF library suffixes if your site follows the */
                /* recommended naming standard for QMF data sets. Otherwise use */
                /* fully-qualified names to identify QMF data sets. */
                /* Change <qmfdef> variable to "DATASET" to use fully-qualified */
                /* data set names; retain the value "SUFFIX" to use the standard */
                /* QMF naming convention.
                -----
                qmfload      = "SDSQLOAD"          /* QMF Load library suffix or */
                                         /* fully qualified dataset name */
                                         /* of QMF Load library */
                qmfchart     = "SDSQCHRT"         /* ADMCFORM library suffix */
                                         /* or fully qualified dataset name */
                                         /* of ADMCFORM library */
                qmfdsdum    = "DUMMY"            /* QMF DSQUDUMP,default=DUMMY */
                                         /* or fully qualified dataset name */
                                         /* to be allocated to DSQUDUMP */
                qmfdebug     = "DUMMY"            /* QMF DSQDEBUG,default=DUMMY */
                                         /* or fully qualified dataset name */
                                         /* to be allocated to DSQDEBUG */
                dsunit       = "SYSDA"             /* disk unit */

                -----
                /* Change the following DB2 identifiers to reflect values in use at */
                /* your site. P.R. uses these values to dynamically allocate DB2 */
                /* and make it available for P.R. users. */
                /* Note: You can use DB2 library suffixes if your site follows the */
                /* recommended naming standard for DB2 data sets. Otherwise use */
                /* fully-qualified names to identify DB2 data sets. */
                /* Change <db2def> variable to "DATASET" to use fully-qualified */
                /* data set names; retain the value "SUFFIX" to use the standard */
                /* DB2 naming convention.
                -----
                db2exit      = "USER.SDSNEXIT"    /* The fully qualified data set */
                                         /* name of your customized */
                                         /* SDSNEXIT data set.           @07A */

                db2ver       = "10"                /* DB2 version no                 @D5C */

```

```

db2rel      = "1"                      /* DB2 release no          */
db2def      = "SUFFIX"                  /* DB2 library suffixes are used   */
                                         /* Change this var. to "DATASET" */
                                         /* If you define fully qualified */
                                         /* dataset names below          */

db2l1lib    = "RUNLIB.LOAD"             /* DB2 Runlib Load library name */
                                         /* or fully qualified dataset name */
                                         /* of DB2 Runlib Load library   */
                                         /* DB2 Load library suffix     */
                                         /* or fully qualified dataset name */
                                         /* of DB2 Load library         */

db2load     = "SDSNLOAD"                /* DB2 Clist library suffix   */
                                         /* or fully qualified dataset name */
                                         /* of DB2 CLIST library       */

db2clst     = "SDSNCLST"                /* DB2 Message library suffix */
                                         /* or fully qualified dataset name */
                                         /* of DB2 Message library PQ72364 */

db2mllib    = "SDSNSPFM"                /* DB2 Panel library suffix   */
                                         /* or fully qualified dataset name */
                                         /* of DB2 Panel library PQ72364 */

db2plib     = "SDSNSPFP"                /* DB2 Panel lib sufz        PQ72364 */
                                         /* English lib sufz           */
                                         /* Japanese lib sufz          */

db2plib2   = "SDSNPFP"                 /* DB2 Panel lib sufz        PQ72364 */
                                         /* English lib sufz           */
                                         /* Japanese lib sufz          */

/*-----*/
gddmload   = "GDDM.SADMMOD"            /* GDDM load library          */
admsymb1   = "GDDM.SADMSYM"             /* GDDM symbol library        */
admdefs    = ""                        /* GDDM nickname library     */

/*-----*/
/* Specify the full name of the GDDM Master Print Queue Dsn in */
/* the following variable. If supplied, it overrides any value */
/* specified for TSOPRNT in the GDDM external defaults file. */
/* The value (if supplied) causes an ADMPRNTQ DD statement to */
/* be added to the batch jcl for graphic reports. */
/*-----*/
admprntq  = ""                        /* GDDM Mstr Prt QDS          */
                                         /* 13@D2d*/
/*-----*/
/* Change the following Information/Management identifiers */
/* to reflect values in use at your site. */
/* P.R. uses these values when generating problem records */
/* in the Info/Man database. */
/*-----*/
def_gesessn = 'BLGSES00'                /* Session member          */
def_geprivcl = 'MASTER'                 /* Privilege Class          */

/*-----*/
/* Change the following variables if you plan to use */
/* the optional JAVA GUI Viewer, or if you don't use QMF or GDDM.*/
/* With VIEWER="YES" you will be able to run your own reports */
/* any time in the future on the JAVA GUI Viewer. */
/* Java Viewer code is downloadable from an IBM WEB site, refer */
/* to the official product documentation. PTR627*/
/* Reports, when VIEWER="NO" & QMFUSE="YES" occupy less DB2 space*/
/*-----*/
viewer    = "NO"                      /* "YES" = VIEWER used PTR627*/
qmfuse   = "YES"                     /* "YES" = QMF used,          */
                                         /* "NO" = QMF is not used    */

```

```

gddmuse = "YES"          /* "YES" = GDDM used,           */
/* "NO" = GDDM is not used   */
/* Note: GDDM must be used if */
/* QMF is used.             */
qmfprint = "YES"         /* "YES" = dsqprint dataset   */
/* "NO" = dsqprint SYSOUT    */
/*-----*/
/* When generating tabular reports without QMF, P.R. uses '.' */
/* (period) as decimal separator and ',' (comma) as thousands */
/* separator. You can exchange the decimal and thousands sep- */
/* arators by specifying decsep="COMMA". Any other value of   */
/* decsep causes P.R. to use period as a decimal separator.   */
/*-----*/
decsep = "PERIOD"       /* Decimal separator            */
/* "PERIOD" or "COMMA"        */
/*-----*/
/*-----*/
/* subhdrv - Substitute Header Variable (in Online Reports) */
/*-----*/
/* 'Y'- replace variables without a value in the report header */
/* with a customized text string (F11 on Data Selection        */
/* panel, or if not defined there, from message DRLA170)      */
/* However, this action increases report creation time.       */
/* If a variable value is left empty and if it also appears */
/* in the report header, the time to generate the report     */
/* will increase in QMF-mode (qmfuse="YES").                  */
/* 'N'- the variable in the header is replaced with the column */
/* name from the WHERE clause of the query.                   */
/*-----*/
/* Note: If this variable is omitted, or set to anything else */
/* than 'Y', 'N' will be used as default. This variable       */
/* is only used in QMF-mode.                                 */
/*-----*/
subhdrv = 'Y'            /* 'Y' or 'N'                  */
/*-----*/
/*-----7@D0a*/
/* Job statement information (required for batch jobs).      */
/*-----*/
/* Note: Do not use JCL comments in these JCL statements.    @P1a*/
/*-----*/
def_jclsta1 = ""          /*Job statement line 1        */
def_jclsta2 = ""          /*Job statement line 2        */
def_jclsta3 = ""          /*Job statement line 3        */
def_jclsta4 = ""          /*Job statement line 4        */
/*-----7@02A*/
/* IBM DB2 Analytics Accelerator default values to use      */
/* DB2 with an Accelerator exploiting IDAA_ONLY loads.      */
/*-----*/
def_useaot = "NO"          /* No Accelerator-only-tables */
def_accelerator = "idaaname" /* Name if using AOTs        */
def_timeint = "T"          /* Time int for tables       */

```

Appendix C. Sample SQL statements for Partitioning

The following sample SQL statements enable you to update the GENERATE_PROFILES and GENERATE_KEYS system tables with your Db2 partitioning requirements. A JCL example is provided in *h/q.SDRLCNTL(DRLJGENA)*. Replace the DRLIN input with your new statements.

```

SQL DELETE FROM &SYSPREFIX.GENERATE_PROFILES
    WHERE PROFILE      = 'SMF'
        AND COMPONENT_ID = '%'
        AND SUBCOMPONENT_ID = '%';

SQL DELETE FROM &SYSPREFIX.GENERATE_PROFILES
    WHERE PROFILE      = 'SMF SYSTEM_ID'
        AND COMPONENT_ID = '%'
        AND SUBCOMPONENT_ID = '%';

SQL COMMIT;

SQL INSERT INTO &SYSPREFIX.GENERATE_PROFILES
    (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID,
     TABLESPACE_NAME, TABLESPACE_TYPE,
     MAXPARTS, NUMPARTS, SEGSIZE,
     TBSPACE1, TBSPACE2, INDEX1, INDEX2)
VALUES ('SMF', '%', '%', '%', 'RANGE', 64, 8, 4,
'PCTFREE 0 USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1 DSSIZE 16G',
'BUFFERPOOL &TSBUFFERPOOL LOCKSIZE ANY LOCKMAX 1000 COMPRESS YES',
'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1 CLUSTER',
'BUFFERPOOL &IXBUFFERPOOL COMPRESS NO');

SQL INSERT INTO &SYSPREFIX.GENERATE_PROFILES
    (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID,
     TABLESPACE_NAME, TABLESPACE_TYPE,
     MAXPARTS, NUMPARTS, SEGSIZE,
     TBSPACE1, TBSPACE2, INDEX1, INDEX2)
VALUES ('SMF SYSTEM_ID', '%', '%', '%', 'RANGE', 64, 8, 4,
'PCTFREE 0 USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1 DSSIZE 32G',
'BUFFERPOOL &TSBUFFERPOOL LOCKSIZE ANY LOCKMAX 1000 COMPRESS YES',
'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1 CLUSTER',
'BUFFERPOOL &IXBUFFERPOOL COMPRESS NO');

SQL COMMIT;

SQL DELETE FROM &SYSPREFIX.GENERATE_KEYS
    WHERE PROFILE      = 'SMF'
        AND COMPONENT_ID = '%'
        AND SUBCOMPONENT_ID = '%';

SQL DELETE FROM &SYSPREFIX.GENERATE_KEYS
    WHERE PROFILE      = 'SMF SYSTEM_ID'
        AND COMPONENT_ID = '%'
        AND SUBCOMPONENT_ID = '%';

SQL COMMIT;

/* The PART_NUM value (fifth item in the value list) determines */
/* how many partitions are created. */
/* The PARTITION_KEY values (eighth item in the value list) must */
/* be in alphabetic/numerical ascending order. */
/* The highest partition number must always be MAXVALUE. */

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
    (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
     PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES ('SMF', '%', '%', '%', 1,
'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
'MVS_SYSTEM_ID',

```

```

      '''FSYS''',
      'Y');

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
  (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
   PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES('SMF', '%', '%', '%', 2,
      'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
      'MVS_SYSTEM_ID',
      '''LSYS''',
      'Y');

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
  (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
   PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES('SMF', '%', '%', '%', 3,
      'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
      'MVS_SYSTEM_ID',
      '''TSYS''',
      'Y');

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
  (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
   PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES('SMF', '%', '%', '%', 4,
      'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
      'MVS_SYSTEM_ID',
      'MAXVALUE',
      'Y');

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
  (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
   PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES('SMF SYSTEM_ID', '%', '%', '%', 1,
      'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
      'SYSTEM_ID',
      '''FSYS''',
      'Y');

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
  (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
   PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES('SMF SYSTEM_ID', '%', '%', '%', 2,
      'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
      'SYSTEM_ID',
      '''LSYS''',
      'Y');

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
  (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
   PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES('SMF SYSTEM_ID', '%', '%', '%', 3,
      'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
      'SYSTEM_ID',
      '''TSYS''',
      'Y');

SQL INSERT INTO &SYSPREFIX.GENERATE_KEYS
  (PROFILE, COMPONENT_ID, SUBCOMPONENT_ID, TABLESPACE_NAME, PART_NUM,
   PARTITION_OPTIONS, RANGE_COLUMNS, PARTITION_KEY, INCLUSIVE)
VALUES('SMF SYSTEMID', '%', '%', '%', 4,
      'USING STOGROUP &STOGROUP PRIQTY -1 SECQTY -1',
      'SYSTEM_ID',
      'MAXVALUE',
      'Y');

```

```
'Y');
SQL COMMIT;
```

Appendix D Sample Batch Collect JCL

```
//COLLECT0  JOB (TDS),'',CLASS=A,MSGCLASS=X,REGION=0M,NOTIFY=&SYSUID
//LOG1      EXEC PGM=DRLPLC,PARM=( ,
//                  'SYSTEM=DED1',
//                  'PLAN=DRLPLAN',
//                  'SYSPREFIX=DRLSYSxx',
//                  '&PREFIX=DRLxx',
//                  '&DATABASE=DRLDBxx',
//                  '&STOGROUP=SYSDEFLT')
//STEPLIB   DD  DISP=SHR,DSN=IZPCA.V310.SDRLOAD
//          DD  DISP=SHR,DSN=DB2.VDB10.SDSNLOAD
//DRLNDICT  DD  DUMMY,
//                  DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//DRLLOG    DD  DISP=SHR,DSN=smf_log_data
//DRLOUT    DD  SYSOUT=*
//DRLDUMP   DD  SYSOUT=*,DCB=(RECFM=VB,LRECL=32756)
//DRLIN     DD  *
      SET USERS='DRLUSRxx';
COLLECT SMF
  COMMIT AFTER BUFFER FULL
  BUFFER SIZE 1300M
  ON OVERFLOW CONTINUE
  ZIIP;
/*

```

Appendix E. Sample Continuous Collect JCL

```

//DRLJJCOL JOB (ACCT#),'CONTINUOUS COLLECT'
//*****
//** NAME: DRLJCCOL
//*
//** FUNCTION: Sample procedure for started task to collect
//** SMF data continuously.
//*
//*****
//** Customization:
//** a. Change VER to match your IZPCA version.
//** b. Change IZPCALD to match the HLQ you used for IZPCA.
//** c. Change DB2LOD to match your Db2 HLQ and version.
//** d. Change the IZPCA parameters to match your system
//** SYSPREFIX - default DRLSYS
//** SYSTEM - default DSN
//** &PREFIX - default DRL
//** &USERS - default DRLUSER
//** &STOGROUP - default DRLSG
//** &DATABASE - default DRLDB
//** e. Change logstream-name to a valid log stream name.
//*
//*****
// SET VER=310
// SET IZPCALD=IZPCA.V&VER.
// SET DB2LOD=DB2.VD10
//*
//DRLSMFCC EXEC PGM=DRLPLC,REGION=0M,
// PARM=(SYSPREFIX=DRLSYS,SYSTEM=DSN,&STOGROUP=DRLSG,'
// '&PREFIX=DRL,&DATABASE=DRLDB,&USERS=DRLUSER')
//STEPLIB DD DISP=SHR,DSN=&IZPCALD..SDRLLOAD
// DD DISP=SHR,DSN=&DB2LOD..SDSNLOAD
//DRLIN DD *
SET USERS='DRLUSER';
COLLECT SMF CONTINUOUSLY FROM logstream-name
COMMIT AFTER 15 MINUTES
FULL STATISTICS AFTER 4 COMMITS
BUFFER SIZE 1300 M
ON OVERFLOW CONTINUE
ZIIP;
/*
//DRLOUT DD SYSOUT=*
//DRLDUMP DD SYSOUT=*

```

Appendix F. Continuous Collector operator commands

Stop the Continuous Collector

/P jobname

/STOP jobname

Modify the Continuous Collector

/MODIFY or /F

Interval Message

Turn the commit heartbeat message DRL0384I ON or OFF, the message is ON by default when the COMMIT phrase FULL STATISTICS AFTER is used, and that time period is greater than the COMMIT AFTER timer period.

F jobname, INTERVAL MESSAGE ON | OFF

F jobname, IM ON | OFF short form

Commit Frequency

Change the frequency at which the Collector commits the data collected to the Db2 tables.

F jobname, COMMIT AFTER BUFFER FULL

F jobname, CA BUFFER FULL

F jobname, COMMIT AFTER count RECORDS | MINUTES | SECONDS

15, 30, or 60 SECONDS

1, 2, 5, 10, 15, 20, 30, 60 MINUTES

F jobname, CA count R | M | S

Full Statistics Frequency

Enable/disable messages at the start and end of database updates and database commits. This function is OFF initially but may be turned on to give an indication of how long the Db2 update and commit are taking. It is not advisable to leave this turned on for a long period of time as performance may be impacted and additional output is written to the DRLOUT data set.

F jobname, FULL STATISTICS AFTER count

F jobname, FSA count short form

Pause and Resume the Continuous Collector

Pause the Continuous Collector until a RESUME command is received. While in the paused state all MODIFY or STOP commands are still accepted, but no data is read from the input log stream. You can use the pause process to avoid contention with other activities in the IZDS Db2 subsystem.

F jobname, PAUSE

F jobname, RESUME

Commit Time messages

Enable/disable messages at the start and end of database updates and database commits. This function is OFF initially but may be turned on to give an indication of how long the Db2 update and commit are taking. It is not advisable to leave this turned on for a long period of time as performance may be impacted and additional output is written to the DRLOUT data set.

F jobname, COMMIT TIME ON | OFF
F jobname, CT ON | OFF short form

ReOpenLogstream function

Closing and reopening the log stream will free any data set extents that have been processed and not freed automatically. The FREE NOW command immediately closes the log stream and reopens it. The FREE n command closes and reopens the log stream immediately, and then the close and reopen process will continue to occur every n hours.

F jobname, LOGSTREAM FREE ON

Stop the log stream free process.

F jobname, LOGSTREAM FREE OFF

Close and reopen the log stream immediately. Once only.

F jobname, LOGSTREAM FREE NOW

Set the log stream free process to happen every 1, 2, 3, 4, 5, or 6 hours.

F jobname, LOGSTREAM FREE n

REFRESH the Continuous Collector definitions

The collector is quiesced and restarted at the time specified, this refreshes all the definitions from SDRLDEFS.

Refresh the collector at hhmm every day until turned off.

F jobname,REFRESH AT hhmm

Stop the refresh process.

F jobname, REFRESH OFF

Start the refresh process at the last “AT time” or “0200” if a time has never been set. Repeat every date at the same time until turned off.

F jobname,REFRESH ON

Refresh the collector immediately.

F jobname, REFRESH NOW

Debugging

Display some additional debug information.

F jobname, SNAP ON | OFF

Appendix G. SMF Extractor operator commands

Start the SMF Extractor.

S *smfext*,SYSTEM=*sysname*

Terminate the SMF Extractor.

F *smfext*,STOP

Change a Startup Option

F *smfext*,C option=*newvalue*

Change a startup option to a new value while the SMF Extractor is running. Any SMF Extractor startup option can be modified.

Request statistics.

F *smfext*,STAT COUNTS / PARMS / DEBUG / ALL

- COUNTS displays number of SMF records read from the SMF VSAM files and the number written to
- the extract files. The counts are cumulative.
- PARMS displays current values of startup options.
- DEBUG displays additional status debug information.
- ALL lists the COUNTS, PARMS and DEBUG information.

Display SMF Extractor Version/Release/Modification level.

F *smfext*,VER

Request Status.

F *smfext*,STATUS|STAT PROCESS

The PROCESS parameter displays differs depending on the z/OS system version.

Pre z/OS V2.3 only displays SMF record types 0 - 255. z/OS V2.3 or higher displays SMF record types 0 - 2047.

F *smfext*,STATUS|STAT PROCESS(ZERO)

The PROCESS(ZERO) parameter only displays SMF record types that have a zero count.

F *smfext*,STATUS|STAT PROCESS(NZERO)

The PROCESS(NZERO) parameter only displays SMF record types that have a non-zero count.

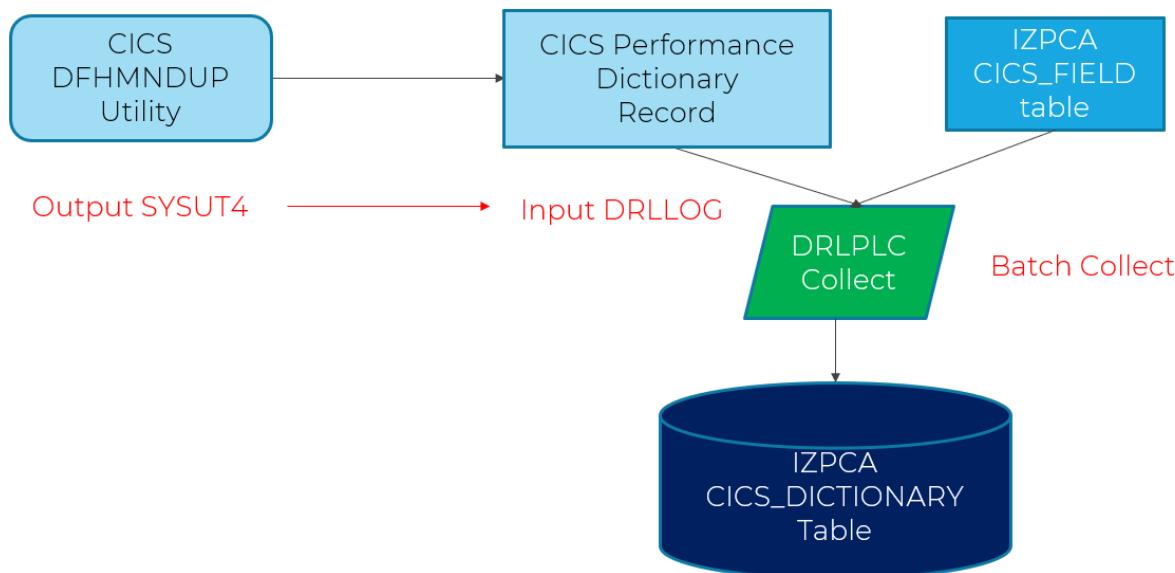
Appendix H. Collecting and processing CICS Monitor records

Populate the IZPCA CICS DICTIONARY table

The CICS utility DFHMNDUP is used to create a CICS Performance Dictionary Record for a CICS system. This record is read in batch by the IZPCA collector (as DRLLOG input) and in conjunction with the IZPCA CICS_FIELD table create the IZPCA CICS_DICTIONARY entries for the CICS system.

This process needs to be completed for each CICS system that will supply monitor records to the IZPCA collector.

If you have defined any user fields to CICS then these must be added to the IZPCA CICS_FIELD table prior to attempting to build the CICS_DICTIONARY entries.



Collecting CICS Monitor records (SMF110)

Once the CICS_DICTIONARY records have been created the CICS Monitor record may be collected in batch or with the Continuous Collector. IZPCA used the CICS_DICTIONARY records to interpret the data in the CICS Monitor records.

Any SMF records not matching CICS_DICTIONARY entries will be written to the DRLNDICT data set. These records may be processed once the CICS_DICTIONARY is updated with the details of the missing CICS system.

Appendix I. DataMover.sh

```

#!/bin/sh
#
# DataMover
#
# IBM Z Performance and Capacity Analytics 3.1
# LICENSED MATERIALS - Property of Teracloud S.A.
# 5698-AS3 (C) Copyright Teracloud S.A. 2018, 2023
#
# Shell script to run the DataMover
#
# Configuration area - tailor to suit your installation
#
# Runtime directory. Other paths are relative to it.
#
# Probably better to use different directories if you are running
# multiple DataMovers on the same system.
#
# Mac: /Users/drl/Documents/IZPCA/DataMover
# Linux: /Users/drl/DataMover
rundir="/u/drl/datamover"
#
# logging.properties controls which messages get sent where
#
# logfile="logging.properties"
#
# The main executable
#
# jarfile="java/DataMover.jar"
#
# The config file that tells it what to do.
#
# config="config/$1.properties"
#
# -----
# Environment area - Where's Java?
# Need when running as a batch job/started task
#
# Mac: /Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home
# Linux: /usr/java/jdk1.8.0_20/bin
# zOS: /apc/java800/64bit/usr/lpp/java/J8.0_64
export JAVA_HOME=/apc/java800/64bit/usr/lpp/java/J8.0_64
export PATH=$PATH:$JAVA_HOME/bin
#
# -----
# Execution area - DO NOT change anything below this point
# REMOVED for this example only

```

Appendix J. Glossary

DataMover. A program supplied with IZPCA for manipulating SMF data and log streams.

Hub. The z/OS system where IZPCA collects the SMF data into a Db2 database.

SMF Extractor. A program supplied with IZPCA to extract SMF records using the SMF exits and storing them on a log stream.

Spoke. A remote z/OS system that collects SMF data and forwards it to a Hub z/OS system for processing by IZPCA.