Restructuring a CA IDMS Database

Laura Rochon
Hera Evolution LLC

Hera
Evolution

IUA/CA IDMS™ Technical Conference   May 7-11, 2018

Abstract

- You need to change a CA IDMS record ? This session will present considerations for restructuring a CA IDMS database record. It focuses on the RESTRUCTURE and RESTRUCTURE CONNECT utility statements, and the schema compare utility IDMSRSTC.   We will see when and how to use these utilities and we will go thru an example.

2

Laura Rochon

- Laura has worked with CA IDMS for 30 years, including close to 7 years with Cullinet and CA. Laura is a frequent presenter at CA World and User Conferences in both North America and Europe. As a technical and application DBA, Laura has supported multiple clients in North America, by teaching classes, performing database and system reviews, installation and maintenance, and just normal DBA work. She presently works for Hera Evolution Inc, a leader in CA IDMS Support.

Agenda

1	INTRODUCTION
2	STEPS FOR A RESTRUCTURE
3	IDMSRSTC UTILITY
4	IDMSRSTTT TABLE
5	RESTRUCTURE SEGMENT & RESTRUCTURE CONNECT UTILITIES
6	INTERESTING CASES

4

Introduction

- The restructure utility is used when a logical change to the database is needed

- Types of changes:
  - Add or delete element in a record
  - Change the length or position or usage mode of element
  - Change the format of a record (compressed to fixed or vice-versa)
  - Add or delete sets
  - Add or delete prior or owner pointers for existing sets
  - Add or remove record compression
  - Change the control length of compressed records

Changes NOT performed by RESTRUCTURE

- Changes to a CALC key (unload/reload)
  - Note: a restructure might be necessary

- Change Duplicates Option of CALC or Sort key (user program)

- Change location mode of a record (unload/reload)

- Change a record's area (unload/reload)

- Change a record's id (user program)

How a restructure works

- The RESTRUCTURE SEGMENT utility
  - performs an area sweep
  - modifies each record identified in the base restructuring table
- If not enough room on page, SR2/SR3 will be created
- If new prior pointers are being added to an existing set, a SPILL file is created
- The RESTRUCTURE CONNECT utility performs an area sweep to connect the new pointers. It will refer to the SPILL file and fill in new pointers in existing set

Steps for a restructure

- Create a base restructure table
  - By coding RSTT macros
  or
  - By running IDMSRSTC utility to generate RSTT macros
  - Assemble and link-edit base restructure table
- Put areas offline
- Backup areas

Steps for a restructure (cont'd)

- Print SPACE to verify if any logically delete records

- Physically erase logically delete records in present

- RESTRUCTURE

- If adding new pointers to existing set, run RESTRUCTURE CONNECT

- PRINT SPACE to determine if many SR2/SR3 are created

- Validate

- Reflect changes in schema and subschemas

- Backup

IDMSRSTC utility

- IDMSRSTC runs in batch only

- IDMSRSTC runs local mode or under CV

- Compares new version to old version of schema and generates RSTT macro statements

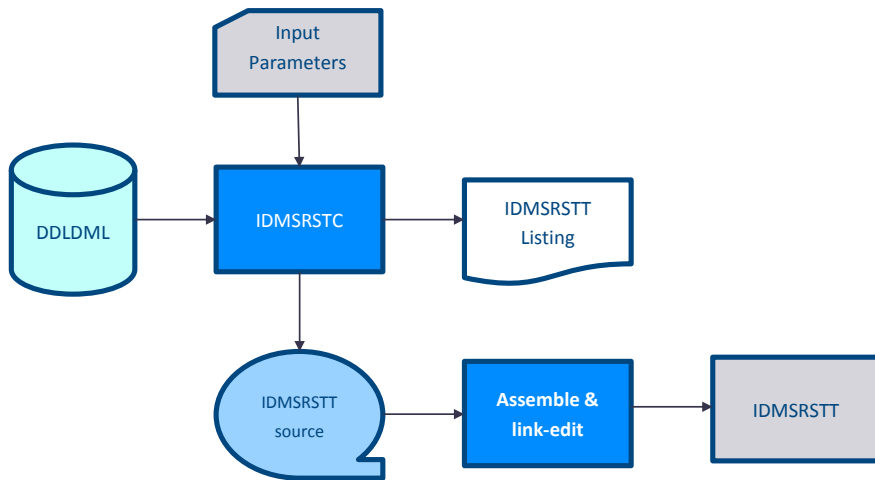- RSTT macros need to be assembled and link-edited

VERY IMPORTANT: generated RSTT macros should be reviewed for accuracy!

IDMSRSTC utility

```
             ┌──────────────┐
             │    Input     │
             │  Parameters  │
             └──────────────┘
                    │
                    ▼
  ┌────────┐   ┌──────────┐      ┌──────────────┐
  │ DDLDML │──▶│ IDMSRSTC │─────▶│   IDMSRSTT   │
  └────────┘   └──────────┘      │    Listing   │
                    │            └──────────────┘
                    ▼
            ┌──────────────┐   ┌──────────────┐   ┌──────────────┐
            │   IDMSRSTT   │──▶│  Assemble &  │──▶│   IDMSRSTT   │
            │    source    │   │   link-edit  │   │              │
            └──────────────┘   └──────────────┘   └──────────────┘
```

IDMSRSTC syntax

SIGNON USER=userid PASSWORD=password .

OLD SCHEMA old-schema-name
NEW SCHEMA new-schema-name   .

SIGNOFF/BYE/LOGOFF .

IDMSRSTC example

```
1IDMSRSTC 18.5                CA, INC.              DATE     TIME     PAGE
 CAGJI5               RESTRUCTURE SCHEMA COMPARE ACTIVITY LIST    03/17/18   17475353   0001
0
 000001   OLD SCHEMA IS EMPSCHM V 100
 000002   NEW SCHEMA IS LXRSCHEM V 1
 000003   .
  IDMSRSTT BUFSIZE=(1100,1100)   OLD NON-HOSP-CLAIM        00000001
 *                               NEW NON-HOSP-CLAIM        00000002
 *IDMSRSTT RECNAME=COVERAGE                                00000003
 000003*+ I DC601235  RECORD UNCHANGED IN OLD AND NEW SCHEMAS              WORD
 *IDMSRSTT RECNAME=DENTAL-CLAIM                            00000004
 000003*+ I DC601235  RECORD UNCHANGED IN OLD AND NEW SCHEMAS              WORD
 *IDMSRSTT RECNAME=DEPARTMENT                              00000005
 000003*+ I DC601235  RECORD UNCHANGED IN OLD AND NEW SCHEMAS              WORD

 …
```

IDMSRSTT macros

IDMSRSTT BUFSIZE	1 per SRTT table
IDMSRSTT RECNAME	1 per record restructured
IDMSRSTT SETPTR	ALL or 1 per pointer in prefix
IDMSRSTT FIELD	ALL or 1 per field or continuous of fields to be modified
IDMSRSTT END	1 per SRTT table
END	1 per SRTT table

IDMSRSTT example

```
IDMSRSTT RECNAME=EMPLOYEE                                              00000006
IDMSRSTT SETPTR=ALL                                                    00000007
IDMSRSTT FIELD=(1,1,4)          COPY EMP-ID-0415                       00000008
IDMSRSTT FIELD=(5,5,10)         COPY EMP-FIRST-NAME-0415               00000009
IDMSRSTT FIELD=(5CL1' ',15,5,NEW)                                      00000010
*               INIT EMP-FIRST-NAME-0415                               00000011
IDMSRSTT FIELD=(15,20,15)       COPY EMP-LAST-NAME-0415                00000012
IDMSRSTT FIELD=(5CL1' ',35,5,NEW,CTRL)                                 00000013
*               INIT EMP-LAST-NAME-0415                                00000014
IDMSRSTT FIELD=(30,40,20)       COPY EMP-STREET-0415                   00000015
IDMSRSTT FIELD=(50,60,15)       COPY EMP-CITY-0415                     00000016
IDMSRSTT FIELD=(65,75,2)        COPY EMP-STATE-0415                    00000017
IDMSRSTT FIELD=(67,77,5)        COPY EMP-ZIP-FIRST-FIVE-0415           00000018
IDMSRSTT FIELD=(72,82,4)        COPY EMP-ZIP-LAST-FOUR-0415            00000019
IDMSRSTT FIELD=(76,86,10)       COPY EMP-PHONE-0415                    00000020
IDMSRSTT FIELD=(86,96,2)        COPY STATUS-0415                       00000021
IDMSRSTT FIELD=(88,98,9)        COPY SS-NUMBER-0415                    00000022
IDMSRSTT FIELD=(97,107,4)       COPY START-YEAR-0415                   00000023
IDMSRSTT FIELD=(101,111,2)      COPY START-MONTH-0415                  00000024
IDMSRSTT FIELD=(103,113,2)      COPY START-DAY-0415                    00000025
IDMSRSTT FIELD=(105,115,4)      COPY TERMINATION-YEAR-0415             00000026
IDMSRSTT FIELD=(109,119,2)      COPY TERMINATION-MONTH-0415            00000027
IDMSRSTT FIELD=(111,121,2)      COPY TERMINATION-DAY-0415              00000028
IDMSRSTT FIELD=(113,123,4)      COPY BIRTH-YEAR-0415                   00000029
IDMSRSTT FIELD=(117,127,2)      COPY BIRTH-MONTH-0415                  00000030
IDMSRSTT FIELD=(119,129,2)      COPY BIRTH-DAY-0415                    00000031
```

(old position, new position, length)

IDMSRSTT example (cont'd)

```
IDMSRSTT RECNAME=EMPLOYEE                                              00000006
IDMSRSTT SETPTR=ALL                                                    00000007
IDMSRSTT FIELD=(1,1,4)          COPY EMP-ID-0415                       00000008
IDMSRSTT FIELD=(5,5,10)         COPY EMP-FIRST-NAME-0415               00000009
IDMSRSTT FIELD=(5CL1' ',15,5,NEW)                                      00000010
*               INIT EMP-FIRST-NAME-0415                               00000011
IDMSRSTT FIELD=(15,20,15)       COPY EMP-LAST-NAME-0415                00000012
IDMSRSTT FIELD=(5CL1' ',35,5,NEW,CTRL)                                 00000013
*               INIT EMP-LAST-NAME-0415                                00000014
IDMSRSTT FIELD=(30,40,91)       COPY remaining fields                  00000015
```

IDMSRSTT example (cont'd)

```
IDMSRSTT RECNAME=OFFICE                                                    00000038
IDMSRSTT SETPTR=(1,1)           COPY OWNER NEXT OFFICE-EMPLOYEE            00000039
IDMSRSTT SETPTR=(2,2)           COPY OWNER PRIOR OFFICE-EMPLOYEE           00000040
IDMSRSTT SETPTR=(,3)            ADD MEMBER INDEX OFFICE-STATE-NDX          00000041
IDMSRSTT FIELD=(1,1,3)          COPY OFFICE-CODE-0450                      00000042
IDMSRSTT FIELD=(4,4,20)         COPY OFFICE-STREET-0450                    00000043
IDMSRSTT FIELD=(24,24,15)       COPY OFFICE-CITY-0450                      00000044
IDMSRSTT FIELD=(39,39,2,,CTRL)  COPY OFFICE-STATE-0450                     00000045
IDMSRSTT FIELD=(41,41,5)        COPY OFFICE-ZIP-FIRST-FIVE-0450            00000046
IDMSRSTT FIELD=(46,46,4)        COPY OFFICE-ZIP-LAST-FOUR-0450             00000047
IDMSRSTT FIELD=(50,50,7)        COPY OFFICE-PHONE-0450                     00000048
IDMSRSTT FIELD=(57,57,7)        COPY OFFICE-PHONE-0450                     00000049
IDMSRSTT FIELD=(64,64,7)        COPY OFFICE-PHONE-0450                     00000050
IDMSRSTT FIELD=(71,71,3)        COPY OFFICE-AREA-CODE-0450                 00000051
IDMSRSTT FIELD=(74,74,3)        COPY SPEED-DIAL-0450                       00000052
```

IDMSRSTT example (cont'd)

```
IDMSRSTT RECNAME=OFFICE                                                    00000038
IDMSRSTT SETPTR=(1,1)           COPY OWNER NEXT OFFICE-EMPLOYEE            00000039
IDMSRSTT SETPTR=(2,2)           COPY OWNER PRIOR OFFICE-EMPLOYEE           00000040
IDMSRSTT SETPTR=(,3)            ADD MEMBER INDEX OFFICE-STATE-NDX          00000041
IDMSRSTT FIELD=ALL              copy all data fields
```

Prereqs for a RESTRUCTURE

- Batch utility in local mode
- Cannot be run against native VSAM file
- Must remove logically deleted record if changes to prefix

Logically deleted records

- PRINT SPACE report:

```
AREA        EMPDEMO.EMP-DEMO-AREA              Distribution of USED Space Report
            Maximum                                        Percent of
Record Type Length     Occurrences   Total Space Used      Total Used
   SR1002      40           3               120              12.09
**LD1002      24           3                72               7.25
Space Inv.   480           1               480              48.38
Overhead      32          10               320              32.25
        *** logically deleted records FOUND ***
```

- Deleting logically deleted records:
  - Thru IDMSBCF:
  - CLEANUP SEGMENT segment-name USING ssc-name ;

RESTRUCTURE

```
Input              RESTRUCTURE        IDMSRSTT
Parameters                            Listing


IDMSRSTT           SPILL              Database
                   file


Input              RESTRUCTURE        IDMSRSTT
Parameters         CONNECT            Listing
```
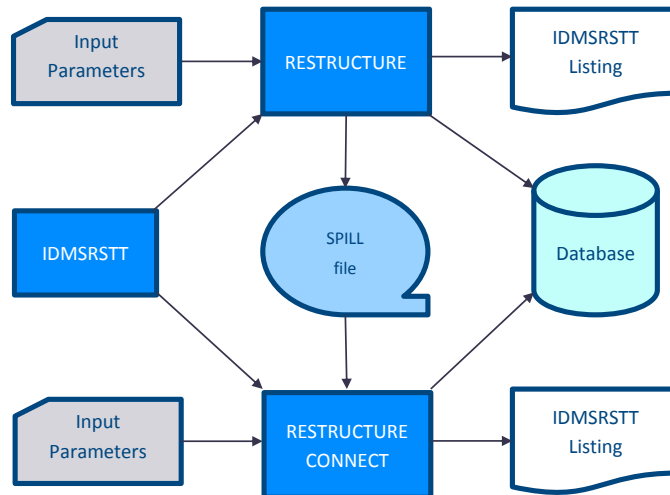
RESTRUCTURE syntax

RESTRUCTURE ⟶ SEGMENT segment-name

⟶ DBNAME dbname

   USING subschema-name

   CONTINUE YES/NO

   RSTTMOD IDMSRSTT/rstt-module-name

   AREA area-name1, area-name2, …

   ;

RESTRUCTURE example

```
RESTRUCTURE SEGMENT EMPDEMO USING EMPSS01
    RSTTMOD RSTTLX2 ;
0UT000038 Starting Restructure of area EMPDEMO.EMP-DEMO-REGION
UT000041 Completed processing of area EMPDEMO.EMP-DEMO-REGION,  Pages read=50  Records read=346
UT011023  Record name EMPLOYEE  Record id 415  Found=56  Changed=56  Ldel=0
0UT000038 Starting Restructure of area EMPDEMO.ORG-DEMO-REGION
UT000041 Completed processing of area EMPDEMO.ORG-DEMO-REGION,  Pages read=25  Records read=195
UT011023  Record name OFFICE  Record id 450  Found=5  Changed=5  Ldel=0
 Status = 0      SQLSTATE = 00000
```

RESTRUCTURE CONNECT

- The RESTRUCTURE CONNECT utility connects new prior and owner pointers in existing sets

- The utility is run after the RESTRUCTURE SEGMENT

- The utility uses the RSTT table and the information in the SPILL file generated by RESTRUCTURE

RESTRUCTURE CONNECT syntax

RESTRUCTURE CONNECT ⟶ SEGMENT segment-name
⟶ DBNAME dbname

   USING subschema-name

   CONTINUE YES/NO

   RSTTMOD IDMSRSTT/rstt-modname

   AREA area-name1 area-name2

   ;

Interesting cases

- Adding a new set between existing records
  - Must develop program to connect the records
  - If set is MA, implementation is 2 phased:
    - Define new set as OM, OA or MM
    - Run restructure
    - Run one-time-shot program to connect ALL member records
    - Change set definition to MA

Interesting cases (cont'd)

- Changing a set from chained to index
  - Write a program to sweep the area, and walk the set, and call IDMSTBLU to build the index structure
  - Restructure the database (if a pointer needs to be removed)
  - Run a MAINTAIN INDEX from SORT3 using the data from the first step

Interesting cases (cont'd)

- Changing a set from index to chained
  - Write a program to sweep the area, and walk the set, and call IDMSTBLU to delete the index structure and also to produce a workfile (with dbkeys of owner/member records)
  - Run a MAINTAIN INDEX from SORT3 using the data from the previous step
  - Restructure the database to remove index pointer, and add chained pointers
  - Sort workfile from 1st step by owner dbkey
  - Write a one-time-shot program to read the sorted workedfile, access the owner record, access the member record and issue a CONNECT record to set

Interesting cases (cont'd)

- Modifying a calc key
  - Restructure the CALC record
    - If adding an element to the calc key, initialize it thru restructure or user-written program
    - If the control length of the record has changed and the record is compressed or variable-length, it is important to indicate CTRL in RSTT
  - Unload/Reload the database
    - If there are VIA records around the CALC record, you will need to run a 2nd unload/reload to recluster the VIA record

Summary

- RSTT macros can be coded manually or generated by IDMSRSTC utility

- RESTRUCTURE restructures records according to the RSTT table

- RESTRUCTURE CONNECT will connect new prior and owner pointers to existing sets

- Important to backup data before and after a restructure

Questions & Answers

Please Complete a Session Evaluation Form

- The number for this session is **D11**

- After completing your session evaluation form, place it in the envelope at the front of the room