\

RACFICE2

Mark Nelson
IBM® Corporation
2455 South Road
Poughkeepsie, NY 12603
markan@us.ibm.com

11 April 2010

Table of Contents

Overview

With the introduction of the RACF Database Unload Utility (IRRDBU00) and the RACF SMF Unload Utility (IRRADU00), installations using RACF have the ability to do unprecedented data analysis and reporting. Tools such as relational data managers, such as DB2® can perform very complicated analysis on very large amounts of data,.Clients have had great success linking IRRDBU00 and IRRADU00 data with existing corporate data repositories.

There are times, however, when the power of tools such as DB2 isn't needed. Consider the creation of a simple report based on a few of the unloaded record types. In cases such as these, simpler tools, such as a sorting utility such as IBM's DFSORT® can be used.

In 1994, we made the RACFICE tool available on the "Downloads" section of the RACF web page (http:\\www.ibm.com/racf). This tool proved popular and with OS/390 Version 2 Release 8, the vast majority of its contents were shipped in 'SYS1.SAMPLIB(IRRICE)'.

Since then, there have been several major enhancements to DFSORT which have added new features that provide powerful tools for performing new types of analysis on IRRDBU00 and IRRADU00 output. The RACFICE2 package describes these features and how they can be used with IRRDBU00 and IRRADU00.

Disclaimer

This program contains code made available by IBM Corporation on an "AS-IS" basis. Any one receiving this program is considered to be licensed under IBM copyrights to use the IBM-provided code in any way he or she deems fit, including copying it and redistributing it, except that it may be neither sold nor incorporated within a product that is sold. No license under any IBM patents or patent applications is to be implied from this copyright license.

The software is provided "as-is", and IBM disclaims all warranties, express or implied, including but not limited to implied warranties of merchantability or fitness for a particular purpose.

Please note that the IBM Support Center does not provide support for any of these programs. Please direct all questions and reports of problems to RACF-L.

Subscribing to RACF-L

RACF-L is an unmoderated discussion list for RACF questions, etc. that is accessible via the Internet. Directions for subscribing to RACF-L can be found in the front of all of the RACF manuals as well as at http://www.ibm.com/systems/z/os/zos/features/racf/links/racf-l.html.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

Change Activity

Date	Description
15 April	Initial version

RACFICE2 Contents

The RACFICE2 package contains:

1. ICETOOL jobs which create these reports:
    - Finding all of the digital certificates which expire in the next 30 days. This is a demonstration of the DFSORT “rolling date” selection criteria.
    - Listing all of the users which are members of a UNIVERSAL group, which demonstrates the power of the DFSORT SPLICE facility.
    - Decoding the contents of the RACF options that are stored in the ICB in the RACF database, which demonstrates the flexibility in DFSORT's record reformatting.
2. DFSORT Symbols for all of the IRRDBU00 and IRRADU00 fields (as of z/OS Release 11).

The RACFICE2 package consists of these files:

File Name	Description
racfice2.pdf	Documentation for the package (this file).
racfice2.xmit	● Sample DFSORT jobs ● DFSORT symbols for the fields in the IRRDBU00 and IRRADU00 records

Table 1: Contents of the RACFICE2 Package

Once unloaded the racfice2.xmit file contains these members:

Member Name	Description
$$COPY$$	Copyright notice
ADUSYMBL	DFSORT symbols for IRRADU00 output
CERTEXP	Find certificates in the RACF data base which expire in the next 30 days.
DBUSYMBL	DFSORT symbols for IRRDBU00 output
ICB	Listing of the RACF data base ICB
UNIVGRP	Lists all of the users connected to a universal group
UACCIDSP	Finds all of the general resource profiles which have an ID(*) value that is less than the UACC.

Table 2: Contents of the racfice2.xmit Data Set

Installing RACFICE2

Installing RACFICE consists of three distinct steps:

1. Getting the RACFICE2 code from ftp://public.dhe.ibm.com/eserver/zseries/zos/racf/racfice2
2. Moving the RACFICE2 code to your MVS or OS/390 system, and
3. Installing the RACFICE2 code and preparing it for use.

Downloading the RACFICE2 Code

There are two ways of downloading the RACFICE2 code The first is to use a web browser to download the RACFICE2 files.To do this,

1. Open the RACF home page at http://www-03.ibm.com/systems/z/os/zos/features/racf/
2. Click on the “Resources” tab at the top of the page.
3. Click on the “Downloads” link.
4. Click on the “RACFICE2” link.
5. Click on the file that you wish to download.

The second method is to download the code using FTP from the IBM FTP web site at public.dhe.ibm.com/eserver/zseries/zos/racf/racfice2/. When downloading using FTP, be sure to set the FTP mode to “binary” for the racfice2.xmit file to ensure that character conversion is not performed.

Note: When the file is installed on your MVS system the target data set must have an LRECL of 80. You can do this from within FTP using the

- SITE LRECL 80,
- LOCSITE LRECL 80, or
- LITERAL SITE LRECL 80 command, depending upon your FTP client. :

Installing the RACFICE Code

Now that the RACFICE code has been received, you must unpack it. This is done using the RECEIVE command:

```
RECEIVE INDATASET('USER01.RACFICE2.XMIT')
```

Additional Information

You can find complete details about DFSORT and ICETOOL in the *DFSORT Application Programming Guide* (SC33-4035). You can find articles, news, tips, techniques, examples, and even an *ICETOOL Mini-User Guide* on the DFSORT homepage at http://www.ibm.com/storage/dfsort/, and you can download DFSORT/ICETOOL papers and examples by FTP using links from the home page.

The JCL to run the RACF Database Unload Utility is documented in the *RACF Security Administrators Guide* and the JCL to run the RACF SMF Unload Utility is documented in the *RACF Auditor's Guide.* . The record mappings for both of these utilities is documented in *RACF Macros and Interfaces.*

An Important Note on Column Numbers

Both IRRADU00 and IRRDBU00 create records that are variable length. Variable length records have a four byte record descriptor word (RDW) describing the length of the record. DFSORT considers the RDW to be a part of the selectable record columns. This means that you must add 4 to any of the field positions identified for the IRRADU00 and IRRDBU00 records described in RACF Macros and Interfaces. For example, if the field for the user ID associated with an event is documented as in *RACF Macros and Interfaces* as beginning at column 59, we add 4 to this position to get 63, the value that we use in both the DFSORT control statement for record selection and the ICETOOL keywords that refer to the field.

BYTIME: Counting the RACF Events by Date and Hour

Sometimes it is useful to see the distribution of events across the day to identify those times when there are flurries of activity. This can be done using the DFSORT OCCURS operator to count the number of records which have a common value. The trick with county the number of events in an hour is to examine only the hour portion of the TIME_WRITTEN field. This field has the format “hh:mm:ss”. Counting only the unique “hh” values, which is done in the DFSORT code below, yields a report with events broken down by date and hour.

```
 OCCURS  FROM(IRRADU00) LIST(PRINT) -
         PAGE -
         TITLE('RACF Activity by Hour') -
         DATE(YMD/) -
         TIME(12:)  -
         BLANK -
         ON(32,10,CH)                HEADER('Date') -
         ON(23,2,CH)                 HEADER('Hour') -
         ON(5,8,CH)                  HEADER('Event') -
         ON(VALCNT)                  HEADER('RACF Events Logged')
```

Table 3: Using DFSORT OCCURS to Summarize Events by the Date and Hour.

```
 - 1 -        RACF Activity by Hour        10/04/14        03:38:21 pm

Date         Hour   Event      RACF Events Logged
----------   ----   --------   ------------------
2009-05-07   09     GENERAL                     2
2009-05-07   10     GENERAL                     3
2009-05-07   10     JOBINIT                   265
2009-05-07   11     JOBINIT                     2
2009-05-07   12     ACCESS                    882
2009-05-07   12     ALTUSER                     1
2009-05-07   12     GENERAL                     1
2009-05-07   12     JOBINIT                     6
2009-05-07   13     ACCESS                  66819
2009-05-07   13     JOBINIT                   102
2009-05-07   13     PERMIT                      5
2009-05-07   13     RALTER                      3
2009-05-07   13     RDEFINE                     1
2009-05-07   13     SETROPTS                    1
2009-05-07   14     ACCESS                   4034
```

Table 4: Using DFSORT OCCURS to Summarize Events by the Hour.

CERTEXP: Find the Digital Certificates Which Expire in the Next “n” Days

DFSORT provides facilities for processing “date” information which is very useful when processing the output of IRRDBU00 and IRRADU00. One of the most useful is DFSORT's support for “relative dates”, which is a date which is plus or minus a specified number of days from the current date. We can use this relative date support to find all of the digital certificates stored in the RACF data base which expire within the next “n” days.

This report is created by selecting the “General Resource Certificate Data” records, which are type 0560. At offset 282 the field “GRCERT_END_DATE” defines the date from which the certificate is no longer valid. We are looking for the 0560 records which match this criteria:

```
286,10,CH,LT,DATE1(-)+30
```

This instructs DFSORT to select those records which have a date value starting in column 286 and is ten character long, which is less than today's date plus 30 days. “DATE1(-)” describes the format of the date in the IRRDBU00 record to DFSORT as yyyy-mm-dd.

The output of this report is shown in Table 5.

```
- 1 -        Expired or Nearly Expired Certificates        08/12/30        12:21:49 am

Name                                                           Start Date   End Date
------------------------------------------------------------   ----------   ----------
0D8B4FEEAAD2185BF4756A9D29E17FFB.OU=Class¢1¢Public¢Primary¢C   1998-05-12   2008-05-12
```

Table 5: Report: Soon-to-Expire Certificates

ICB: Formatting the ICB

Up to this point, we've focused on using DFSORT to select records of interest from a large set. There are occasions when we don't need to DFSORT to extract records, but rather to reformat records. The ICB “report” does just that. This report extracts the Inventory Control Block (ICB), which is the first record in the RACF database, and re-formats it into an easy-to-ready report, a sample of which is shown in Table 6 .

```
ICB information                   Page:001

Number of BAM Blocks=00000001
RBA of highest CIB=00000001A000
RBA of first block in sequence set=00000000E000
RBA of first BAM block =00000000C000

ICBFLAGS=X<30>
  Dataset Extended=NO
  Resident data blocks=YES
  Data set format=NEW
  HPCS Write in Progress=NO

Number of templates=05
BAM high water mark=00000000C000
ICBSTAT=X<3D>
  Bypass RACINIT stats=NO
  Bypass Data set statistics=YES
  No TAPEVOL statistics=YES
  No DASD volume statistics=YES
  No terminal statistics=YES
  No ADSP protection=NO
  EGN=YES

ICBSTAT1=X<30>
  Tape Volume protection=NO
  DASD Volume protection=NO
  Generic profiles for the dataset class=YES
  Generic commands for the dataset class=YES
  Input data set name used for logging=NO
  JES-XMBALLRACF=NO
  ...
  ...
```

Table 6: Formatting the ICB Using DFSORT

Let's take a look at how this report was created. The first part of this DFSORT control statement which created this report is shown in Table 7.

```
//S1        EXEC   PGM=ICEMAN
//SYSOUT    DD SYSOUT=*
//SORTIN    DD DISP=SHR,DSN=...     RACF Database
//SORTOUT   DD SYSOUT=*
//SYSIN DD  *
   OPTION COPY
   ALTSEQ CODE=(005C)
   OUTFIL STARTREC=1,ENDREC=1,LINES=255,
     HEADER2=(3/,C'ICB information',20X,C'Page:',PAGE=(EDIT=(TTT)),3/),
     BUILD=(C'Number of BAM Blocks=',5,4,HEX,/,
            C'RBA of highest CIB=',9,6,HEX,/,
            C'RBA of first block in sequence set=',15,6,HEX,/,
            C'RBA of first BAM block =',21,6,HEX,2/,

            C'ICBFLAGS=X<',27,1,HEX,C'>',/,
            C'  Dataset Extended=',
               27,1,CHANGE=(3,B'1.......',C'YES',B'0.......',C'NO'),/,
...
...
```

Table 7: DFSORT Job to Format the ICB (Fragment 1)

Let's examine these statements.

```
OPTION COPY
```

... instructs DFSORT that record in this report are going to be processed in the order in which they are in the input data set, which is the RACF database. In a few moments, we'll tell DFSORT that we are only going to process one record.

```
ALTSEQ CODE=(005C)
```

... instructs DFSORT to substitute the EBCDIC X'5C' (period) character for a hexadecimal zero (X'00'). We'll see how this is used later on.

```
OUTFIL STARTREC=1,ENDREC=1,LINES=255,
```

... starts the core of the DFSORT job. STARTREC=1,ENDREC=1 tell DFSORT that we are processing exactly one record. LINES=255 defines how many lines per page in the report.

```
HEADER2=(3/,C'ICB information',20X,C'Page:',PAGE=(EDIT=(TTT)),3/),
```

... defines the header for the report: Skip three lines, put out the string “ICB information”, skip 20 characters, put out the page number as “Page:nnn”, and then skip three lines. Since we are formatting only one record (the ICB), the header appears only once.

```
BUILD=(C'Number of BAM Blocks=',5,4,HEX,/,
```

... puts into the report the string “Number of BAM Block=” followed by the four bytes extracted from the byte 5 of the ICB, presented in hexadecimal format. We repeat this processing for all of the fields in the ICB. These fields are documented in “RACF: Diagnosis”.

One important note: The RACF database is a fixed-length record data set, so we don't have to the length of the record descriptor word (RDW) to each of the DFSORT column values. However, the ICB is documented in “RACF: Diagnosis” as starting at offset 0, which corresponds to DFSORT column 1.

The vast majority of the report is created by putting out a line of descriptive text (such as “Number of BAM blocks =”) followed the value that is extracted from the record. This value is converted depending upon the type of information that is in the field. “HEX” presents the value as a hexadecimal value. “BI,TO=ZD” instructs DFSORT to treat the value as a binary value and to convert it to zoned decimal for output. For example, if the field contains a X'0F', then the value “15” is printed.

DFSORT supports the output of a string value based upon the setting of a single bit within a value. This very useful reporting tool is used extensively in this report. For example, ICBFLAGS is a byte which contains several flags. The code fragment is

```
C'ICBFLAGS=X<',27,1,HEX,C'>',/,
C'  Dataset Extended=',
   27,1,CHANGE=(3,B'1.......',C'YES',B'0.......',C'NO'),/,
```

... examines the ICBFLAGS and if the first bit is ON prints out “Dataset Extended=YES' and “Dataset Extended=NO” if this bit is off. This translation is performed by the “CHANGE=” value. This statement looks at character 27 for 1 byte then reserves 3 bytes in the output and puts a “YES” in the field if the first mapping (the one with the first bit set to 1) is matched, and a NO the second mapping (the one with the first bit set to 0) is matched.

UNIVGRP: Finding all of the members of a UNIVERSAL Group

Creating group profiles is a very popular way to simplify security administration and start down the path of creating a “role-based” access control environment. z/OS V1R2 introduced the UNIVERSAL group. Universal groups allow can have an unlimited number of user connect to them, as long as the user does have an authority higher than “USE” in the group.

Each user profile contains a list of all of the groups to which the user is connected. Each non-UNIVERSAL group profile contains a list of all of the users connected to the group. UNIVERSAL group profiles only contain a list of those users who have an authority in the group greater than USE.

The downside of this implementation is that a LISTGRP command for a universal group shows only those users who have greater than USE authority.

We can use the the output of the RACF Data Base Unload Utility (IRRDBU00) and IBM's DFSORT utility to find all of the members of a universal group. We do this by finding all of the groups which are universal groups and find all of the user-group connection records which reference those groups. The two IRRDBU00 records that we process for the “Users Connected to UNIVERSAL Groups” report are listed in Table 8.

Record Type	**Name**	**Fields Used**
0100	GROUP_BASIC_DATA	Group name, owner, universal flag
0205	USER_GROUP_CONN	Group name, attributes

Table 8: IRRDBU00 Record Types to Find Users Connected to UNIVERSAL Group

The process to create this report is to:

1. Extract the GROUP BASIC DATA (0100) records and place the data into the same columns as the USER GROUP CONNECTION record, with the "UNIVERSAL" flag appended at the end.

2. Extract all of the USER_CONNECT_DATA records and pad them with blanks to be the same size as the records created in step 1.

3. Combine the records from step 1 and step 2 and sort them based on the group name and the record type. For each group, this places the record which contains the "is this a UNIVERSAL" group flag record ahead of the records which define the members of the group.

4. Using the DFSORT ICETOOL SPLICE command, merge the records, so that the "is this a   UNIVERSAL" group flag record is placed on each of the group member records.

5. Select all of the records which have a "YES" for the "is this a UNIVERSAL group" flag.
6. Create a report for the records selected abo.ve

See member UNIVGRP for the details. The output of this report is shown in Table 9.

```
- 1 -         Users Connected to a UNIVERSAL Group        10/04/14        02:21:19 pm

UNIVGRP1

User       Created      Owner      Spec?   Oper?   Audit?
--------   ----------   --------   -----   -----   ------
USER001    2010-04-14   MARKN      NO      NO      NO
USER002    2010-04-14   MARKN      YES     NO      NO
USER003    2010-04-14   MARKN      NO      YES     NO
USER004    2010-04-14   MARKN      NO      NO      YES
USER005    2010-04-14   MARKN      YES     NO      YES
USER006    2010-04-14   MARKN      NO      NO      NO
USER007    2010-04-14   MARKN      NO      NO      NO
USER008    2010-04-14   MARKN      NO      NO      NO

- 2 -         Users Connected to a UNIVERSAL Group        10/04/14        02:21:19 pm

UNIVGRP2

User       Created      Owner      Spec?   Oper?   Audit?
--------   ----------   --------   -----   -----   ------
USER011    2010-04-14   MARKN      NO      NO      NO
USER012    2010-04-14   MARKN      YES     NO      NO
USER013    2010-04-14   MARKN      NO      YES     NO
USER014    2010-04-14   MARKN      NO      NO      YES
USER015    2010-04-14   MARKN      YES     NO      YES
USER016    2010-04-14   MARKN      NO      NO      NO
USER017    2010-04-14   MARKN      NO      NO      NO
USER018    2010-04-14   MARKN      NO      NO      NO
```

Table 9: The UNIVERSAL Group Member Report

ADUSYMBL/DBUSYMBL: DFSORT ICETOOL Symbols

Members ADUSYMBL and DBUSYMBL contain all the DFSORT symbols all of the fields in the IRRDBU00 and IRRADU00 output as of z/OS V1R11. When the original RACFICE package was created in 1994, the DFSORT symbols were created by manually editing the IRRDBU00 and IRRADU00 DB2 Load Utility Statements. This was a tedious and error prone process.

For RACFICE2, a new technique was considered. Rather than manually processing the DB2 Load Utility Statements, writing a small application, possibly in REXX, was considered. Prior to writing the code, we talked to Frank Yaeger to see if this could be done using just DFSORT. It turns out that it could!

Table 10 is a short DFSORT program which processing the IRRDBU00 and IRRADU00 DB2 Load Utility Statements and creates the DFSORT symbol statements. These load utility statements contain information about the tables, databases, and table spaces into which the output of IRRDBU00 and IRRADU00 are placed. A typical load utility statement looks like this:

```
INIT_EVENT_TYPE       POSITION(1:8)          CHAR(8),
```

What we want to create is a DFSORT Symbol statement which looks like this:

```
INIT_EVENT_TYPE,5,8,CH
```

Our job is complicated since the DB2 Load Utility statement contains the starting position of the field and the ending position, however, the length could be on a separate record. Since we have the starting and ending position, we could calculate the length of the field. We also need to ignore extraneous DB2 Load Utility statements.

Our DFSORT code does this by:

- Finding all of the records which have the string “POSITION(“ and are do not have the string “AUTHID_NAME”. This eliminates all of the records which have information that we don't need, such as database information, table space information, and the like. It also eliminates the table of auth Ids (which has the field “AUTHID_NAME”) as this is not actually a table created by IRRDBU00.
- Parses the selected fields into three parts:
  1. The first part starts at the first non-blank and ends at the first blank for a maximum of 20 characters. This is the field name
  2. The second part starts rignt after the first “(“ and continues to the next “:”. This is the starting column.
  3. The third starts right after the “:” from above and extends to the next “)”. This is the ending column
- Creates a DFSORT symbol statement by putting the field name, the starting column, and the ending column minus the starting column.
- Squeezes out all of the extraneous blanks and shifts the record to the left.

RACFICE2 contains the result of this DFSORT example in members ADUSYMBL and DBUSYMBL. The DFSORT code which create the example is not included.

```
//S1        EXEC  PGM=ICETOOL,PARM='MSGPRT=ALL'
//RACDBULD  DD  DISP=SHR,DSN=SYS1.SAMPLIB(RACDBULD)
//SORTOUT   DD  SYSOUT=*
//TOOLMSG   DD SYSOUT=*                                                 00010000
//PRINT     DD SYSOUT=*                                                 00020000
//DFSMSG    DD SYSOUT=*                                                 00021000
//DBUSYMCL  DD DISP=SHR,DSN=MARKN.RACFICE2.CNTL(DBUSYMCL)
//DBUSYMHD  DD DISP=SHR,DSN=MARKN.RACFICE2.CNTL(DBUSYMHD)
//TOOLIN    DD *                                                        00024000
  SORT FROM(RACDBULD) TO(DBUSYMCL) USING(CTL1)
/*
//CTL1CNTL DD *
   OPTION COPY
  INCLUDE COND=(1,60,SS,EQ,C'POSITION(',AND,
                1,60,SS,NE,C'AUTHID_NAME')
   INREC IFOUTLEN=80,
     IFTHEN=(WHEN=INIT,
       PARSE=(%01=(STARTAT=NONBLANK,ENDBEFR=C' ',FIXLEN=20),
              %02=(STARTAFT=C'(',ENDBEFR=C':',FIXLEN=4),
              %03=(ENDBEFR=C')',FIXLEN=4)),
       BUILD=(%01,X,%02,UFF,ADD,+4,EDIT=(IIIT),X,
         %03,UFF,SUB,%02,UFF,ADD,+1,EDIT=(IIIT),X,C'CH')),
     IFTHEN=(WHEN=INIT,BUILD=(1,33,SQZ=(SHIFT=LEFT,MID=C',')))
/*
```

Table 10: DFSORT Job to Create DFSORT Symbol Statements from DB2 Load Utility Statements