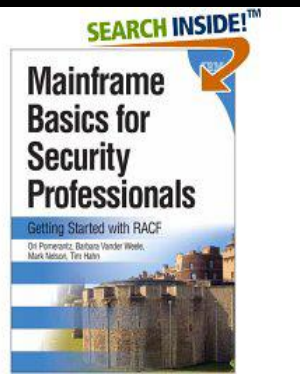


The Joy of JOINKEYS

RACF® Users Group of New England
Chicago RACF Users Group
Kentucky/Ohio/Indiana RACF Users Group
12 May 2020

Mark Nelson, CISSP®, CSSLP®
markan@us.ibm.com



DFSORT, ICETOOL, and JOINKEYS

- **DFSORT is IBM's high-performance sort, merge, copy, analysis and reporting product**
 - **“It’s way more than just sorting!”**
- **ICETOOL leverages the capabilities of DFSORT to perform multiple operations on one or more data sets in a single job step.**
- **JOINKEYS allows you to "join" two files by one or more keys**



**Check out “ Getting Started with DFSORT” for tutorials and examples:
[https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4SC236880/\\$file/iceg200_v2r4.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4SC236880/$file/iceg200_v2r4.pdf)**

JOINKEYS

- **JOINKEYS** allows you easily to create joined records in a variety of ways including inner join, full outer join, left outer join, right outer join, and unpaired combinations.
- **The data for the JOINKEYS is in two input DD names**
 - The two input DD names can be of different types (fixed, variable, VSAM, and so on)
 - The keys (common fields) can be in different locations in the record
 - The two DD names can point to the same data set

JOINKEYS...

- **There are three control statements for a JOINKEYS operation:**
 - **JOINKEYS:** You must specify **two** JOINKEYS statements, one for each input file, specifying:
 - The DD name of the file
 - The length and sequence of the keys in the file
 - Indicate whether the file is already sorted by those keys,
 - **JOIN** (optional): Defines the type of join. Defaults to “inner”
 - **REFORMAT** (optional for JOIN ONLY) : Defines the fields that you want in the joined records. You can also request an indicator of where the key was found ('B' for both files, '1' for file 1 only or '2' for file 2 only) and a fill character for missing bytes.

The Types of Joins

- **Consider two tables:**
 - One which contains baseball players names and a team ID
 - One which contains the team ID and the name of the team

Player	Team ID
Kranepool, Ed	NYM
Berra, Yogi	NYN
Gaedel, Eddie	SLB

Team ID	Name
NYM	NY Mets
NYN	NY Yankees
SFG	SF Giants

The Types of Joins...

Player	Team ID
Kranepool, Ed	NYM
Berra, Yogi	NYN
Gaedel, Eddie	SLB

Team ID	Name
NYM	NY Mets
NYN	NY Yankees
SFG	SF Giants

- **JOINKEYS** allows you to create these **JOINS**
 - **Inner join:** (Default) Only the paired records from (Kranepool, Berra)
 - **Left outer join:** The player records (Kranepool, Berra, Gaedel)
 - NAME will be blank for Gaedel
 - **Right outer join:** The team ID records
 - Player will be blank for SF Giants
 - **Full outer join:** All records
 - Player will be blank for SF Giants
 - NAME will be blank for Gaedel
 - **Unpaired players** (Gaedel)
 - **Unpaired teams** (SFG)
 - **All unpaired** (Gaedel, SFG)

Sample JOINKEYS Job

```
//STEP0100 EXEC PGM=SORT
//SYSOUT DD SYSOUT=*
//INA DD *
-----1-----2-----3-----4
KRANEPOOL, ED NYM
BERRA, YOGI NYY
GAEDEL, EDDIE SLB
//INB DD *
NYM NY METS
NYY NY YANKEES
SFG SF GIANTS
//SORTOUT DD SYSOUT=*
//SYSIN DD *
OPTION COPY
JOINKEYS F1=INA, FIELDS=(31,3,A)
JOINKEYS F2=INB, FIELDS=(01,3,A)
REFORMAT FIELDS=(F1:1,35,F2:5,15)
//*
```

The output would be:

```
-----1-----2-----3-----4-----5
KRANEPOOL, ED NYM NY METS
BERRA, YOGI NYY NY YANKEES
```

Using JOINKEYS With IRRDDBU00

Certificate DN Information in IRRDBU00

RACF stores digital certificates in the RACF data base as profiles in the DIGTCERT class

- Profile name is a “munged” version of the certificate issuer and serial #

The issuer's distinguished name (IDN) and subject's distinguished name (SDN) are stored as opaque data within the DIGTCERT profile

- Prior to z/OS® V2.1, to get the IDN and SDN, the certificate had to be extracted and decoded

Starting with z/OS V2.1, the RACF Database Unload Utility now unloads the issuers distinguished name and the subjects distinguished

- Unloaded into the 1560 record
- “Additional information” for the 0560 record
- The profile name and class link the 1560 record to the other 05xx records

How can these records be joined using DFSORT? JOINKEYS!

The 1560 Record

- **The 1560 contains:**

- The issuer's distinguished name
- The subject's distinguished name
- The hashing algorithm used for the signing the certificate

The “1560” record links to the “0560” record using the profile name

- **DFSORT's JOINKEY operator can be used when processing IRRDBU00 output**

- **The Mapping of the 1560 Record is:**

Field Name	Type	Position		Comments
		Start	End	
CERTN_RECORD_TYPE	Int	1	4	Record type of the certificate information record (1560).
CERTN_NAME	Char	6	251	General resource name as taken from the profile name.
CERTN_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
CERTN_ISSUER_DN	Char	262	1285	Issuer's distinguished name. (1024 characters)
CERTN_SUBJECT_DN	Char	1287	2310	Subject's distinguished name. (1024 characters)
CERTN_SIG_ALG	Char	2312	2327	Certificate signature algorithm. Valid values are md2RSA, md5RSA, sha1RSA, sha1DSA, sha256RSA, sha224RSA, sha384RSA, sha512RSA, sha1ECDSA, sha256ECDSA, sha224ECDSA, sha384ECDSA, sha512ECDSA, and UNKNOWN.

The 0560 Record

- The Mapping of the 0560 Record is:

Field Name	Type	Position		Comments
		Start	End	
GRCERT_RECORD_TYPE	Int	1	4	Record type of the Certificate Data Record (0560)
GRCERT_NAME	Char	6	251	General resource name as taken from the profile name.
GRCERT_CLASS_NAME	Char	253	260	Name of the class to which the profile belongs.
GRCERT_START_DATE	Date	262	271	The date from which this certificate is valid.
GRCERT_START_TIME	Time	273	280	The time from which this certificate is valid.
GRCERT_END_DATE	Date	282	291	The date from which this certificate is no longer valid
GRCERT_END_TIME	Time	293	300	The time from which this certificate is no longer valid.
GRCERT_KEY_TYPE	Char	302	309	The type of key associated with the certificate.
GRCERT_KEY_SIZE	Int	311	320	The size of private key associated with the certificate.
GRCERT_LAST_SERIAL	Char	322	337	The hexadecimal representation of the low-order eight-bytes of the serial number last signed with this key.
GRCERT_RING_SEQN	Int	339	348	A sequence number for certificates within the ring.

JOINKEYS to Join 1560 and 0560 Records

```
//MARKNSRT  JOB CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,MSGLEVEL=1
//DS$STAND  EXEC PGM=SORT
//SYSOUT    DD SYSOUT=*
//SORTJNF1  DD DISP=SHR,DSN=MARKN.TEST.IRRDBU00
//SORTJNF2  DD DISP=SHR,DSN=MARKN.TEST.IRRDBU00
//SORTOUT   DD SYSOUT=*
//*-----
/* Remember: The IRRDBU00 Output is VB! Add +4 to all of the starting
/* positions documented in RACF Macros and Interfaces (SA22-7682)
/*-----
//SYSIN     DD *
JOINKEYS  FILE=F1,FIELDS=(10,246,A,257,8,A)
JOINKEYS  FILE=F2,FIELDS=(10,246,A,257,8,A)

REFORMAT  FIELDS=(F1:266,20,286,20,
                  F2:2316,16,266,1025,1291,1025)

OPTION COPY
```

JOINKEYS to Join 1560 and 0560 Records....

```
OUTFILE HEADER2=(50:'Certificates in the RACF Data Base',
/,
58:'Prepared on ',DATE,/,/,
63:'at ',TIME,/,/,
105:'Page: ',PAGE=(EDIT=(TTT) ),3/,

01:'Subject DN',
76:'Start',
96:'End',/,
01:'Issuer DN',
76:'Date',
87:'Time',
96:'Date',
107:'Time',
116:'Key Type',/,
01:74'-',
76:10'-',
87:08'-',
96:10'-',
107:08'-',
116:15'-'),
BUILD=(01:57,74,/,01:1082,74,76:1,20,21,20,41,16,/)
/*
```

JOINKEYS to Join 1560 and 0560 Records...

```
//JNF1CNTL DD *  
    OPTION VLSCMP  
        INCLUDE COND=(5,4,CH,EQ,C'0560')  
/*  
//JNF2CNTL DD *  
    OPTION VLSCMP  
        INCLUDE COND=(5,4,CH,EQ,C'1560')  
/*
```

JOINKEYS to Join 1560 and 0560 Records...

Certificates in the RACF Data Base
Prepared on 04/11/20
at 14:14:46

Page:001

Subject DN Issuer DN	Start Date	Time	End Date	Time	Key Type
OU=Class 1 Public Primary Certification Authority.O=VeriSign, Inc..C=US CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.OU=www.	1998-05-12	00:00:00	2008-05-12	23:59:59	md2RSA
personal-basic@thawte.com.CN=Thawte Personal Basic CA.OU=Certification Ser personal-basic@thawte.com.CN=Thawte Personal Basic CA.OU=Certification Ser	1996-01-01	00:00:00	2020-12-31	23:59:59	md5RSA
personal-freemail@thawte.com.CN=Thawte Personal Freemail CA.OU=Certificati personal-freemail@thawte.com.CN=Thawte Personal Freemail CA.OU=Certificati	1996-01-01	00:00:00	2020-12-31	23:59:59	md5RSA
personal-premium@thawte.com.CN=Thawte Personal Premium CA.OU=Certification personal-premium@thawte.com.CN=Thawte Personal Premium CA.OU=Certification	1996-01-01	00:00:00	2020-12-31	23:59:59	md5RSA
CN=BAD LABEL.T=UNIT TESTING.OU=RACF_CERTIFICATE_EXPIRATION.O=IBM.L=POUGHKE CN=BAD LABEL.T=UNIT TESTING.OU=RACF_CERTIFICATE_EXPIRATION.O=IBM.L=POUGHKE	2009-01-01	04:00:00	2010-01-02	03:59:59	sha1RSA
CN=DAVE FRISHBERG.T=SENIOR SOFTWARE ENGINEER.OU=SYSTEMS AND TECHNOLOGY GRO CN=DAVE FRISHBERG.T=SENIOR SOFTWARE ENGINEER.OU=SYSTEMS AND TECHNOLOGY GRO	2011-11-10	04:00:00	2012-11-11	03:59:59	sha1RSA
CN=MARK NELSON.T=SENIOR SOFTWARE ENGINEER.OU=SYSTEMS AND TECHNOLOGY GROUP. CN=MARK NELSON.T=SENIOR SOFTWARE ENGINEER.OU=SYSTEMS AND TECHNOLOGY GROUP.	2011-11-04	04:00:00	2012-11-05	03:59:59	sha1RSA
CN=STG Code Signing CA.OU=IBM Code Signing.O=IBM Corporation.C=US CN=STG Code Signing CA.OU=IBM Code Signing.O=IBM Corporation.C=US	2008-07-01	04:00:00	2028-07-01	03:59:59	sha1RSA
CN=Test Certificate.T=RACF_CERTIFICATE_EXPIRATION.OU=Unit Test.OU=CertLabe CN=Test Certificate.T=RACF_CERTIFICATE_EXPIRATION.OU=Unit Test.OU=CertLabe	2012-01-24	05:00:00	2012-01-25	04:59:59	sha1RSA
CN=Test Certificate.T=RACF_CERTIFICATE_EXPIRATION.OU=Unit Test.OU=CertLabe CN=Test Certificate.T=RACF_CERTIFICATE_EXPIRATION.OU=Unit Test.OU=CertLabe	2012-01-24	05:00:00	2012-01-25	04:59:59	sha1RSA
CN=Test Certificate.T=RACF_CERTIFICATE_EXPIRATION.OU=Unit Test.OU=CertLabe CN=Test Certificate.T=RACF_CERTIFICATE_EXPIRATION.OU=Unit Test.OU=CertLabe	2012-01-24	05:00:00	2012-01-25	04:59:59	sha1RSA

Solving a Real-World Problem with JOINKEYS

Background

- **Imagine a RACF database with a group (we'll call it "BIGGROUP") into which almost every user is placed**
- **Imagine how easy it would be to merely put BIGGROUP on access lists to "get things to work"**
- **Imagine an auditor finding that profiles which had BIGGROUP on the access list were flagged as violating the installations "need to know" policy**
- **What would you do?**

Background...

- The installation decided to segment BIGGROUP into a set of role-groups (we'll call them SMLGRP01, SMLGRP02, SMLGRP03.... etc.)
- **Considerations: The client had:**
 - Only three months to get this done
 - A major application and an unmovable project deadline that depended on the BIGGROUP entries
- **Question: How would the client:**
 - Find all of the references to BIGGROUP
 - Notify the profile owners that they needed to move from BIGGROUP to one or more SMLGRPxx access list entries?
 - Provide a backout plan in to ensure that there were application outages caused by this migration?

Approach

- 1. Create a list/report which identified every BIGGROUP access list entry, showing the:**
 - Profile name and class
 - Access level
 - Profile owner
- 2. Survey all of the application owners and users and create and populate the SMLGRPs**
- 3. Work with the application owners and profile owners to add the SMLGRPs to the access list**
 - BIGGROUP would remain on the access list
- 4. De-populate BIGGROUP**
 - In the event of a production problem, users could be re-connected to BIGGROUP on an emergency basis.

How to Analyze the RACF Data Base?

- **We used JOINKEYS against the IRRDBU00 output to create the reports needed to:**
 - Find all of the BIGGROUP references and the associated profile owners and volsers
 - Map the contents of the BIGGROUP and the SMLGRPs to identify:
 - » All of the users in BIGGROUP who were not in any SMLGRP
 - > These were the users who would no longer have access once BIGGROUP was “drained”
 - » The set of SMLGRP members who were not in BIGGROUP
 - > These were the users who may have gotten more authority than they had before

JOINKEYS Joining Access List Entry to Profile Owner

```
//GROUPREF EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//PRINT DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//DBU1 DD DISP=SHR,DSN=USER01.IRRDBU00
//DBU2 DD DISP=SHR,DSN=USER01.IRRDBU00
//TEMP0001 DD UNIT=SYSALLDA,SPACE=(TRK,(10,10,0))
//TOOLIN DD *
COPY JKFROM TO(TEMP0001) USING(JOIN)

DISPLAY FROM(TEMP0001) LIST(PRINT) -
PAGE -
TITLE('Data Set Profiles with References to BIGGROUP') -
DATE(YMD/) -
TIME(12:) -
BLANK -
ON(01,44,CH) HEADER('Data Set Name') -
ON(46,06,CH) HEADER('VOLSER') -
ON(53,08,CH) HEADER('Owner')

//JOINCNTL DD *
OPTION VLSCMP
JOINKEYS F1=DBU1,FIELDS=(10,44,A,55,6,A),
INCLUDE=(5,4,CH,EQ,C'0400')
JOINKEYS F2=DBU2,FIELDS=(10,44,A,55,6,A),
INCLUDE=(5,4,CH,EQ,C'0404',AND,62,8,CH,EQ,C'BIGGROUP')
REFORMAT FIELDS=(F1:10,45,55,7,78,9,
F2:71,9)
/*
```

JOINKEYS Joining Access List Entry to Profile Owner...

Data Set Profiles with References to BIGGROUP

13/03/19

11:03:37 pm

Data Set Name

VOLSER

Owner

SYS1.MACS

PPP

SYS1.TOOL*

PPP

SYS1.TOOL.TSCENV

MVSSPT

SYS1.TOOL.TSCUSER

MVSSPT

The Joy of JOINKEYS

**RACF® Users Group of New England
Chicago RACF Users Group
Kentucky/Ohio/Indiana RACF Users Group
12 May 2020**

**Mark Nelson, CISSP®, CSSLP®
markan@us.ibm.com**

