# Security Analysis Using the RACF Database Unload Utility (IRRDBU00), RACF SMF Data Unload Utility (IRRADU00), and DFSORT's ICETOOL Utility

8 September, 1998

Mark A. Nelson

IBM Corporation
RACF Development
Mail Station P385 Building 706 Department BWVA
Poughkeepsie, NY 12601

Internet: markn@vnet.ibm.com
IBMMAIL: USIBMV4B@IBMMAIL

# Abstract

Effective security management requires flexible analysis and reporting tools. The RACF[1] product has introduced two tools that assist security administrators and auditors; the RACF Database Unload Utility (IRRDBU00), introduced in RACF 1.9.0, and the RACF SMF Data Unload Utility (IRRADU00), introduced in RACF 2.1.0. This paper describes a technique for creating reports using the output of these utilities as input to the ICETOOL utility provided with DFSORT[2] . Many examples are included.

## Version History

| **Date** | **Description** |
|---|---|
| **22 September, 1994** | The original version, which included samples that used IRRADU00 and IRRDBU00 output. |
| **31 July, 1996** | Version 2, which added a mapping of the RACF Report Writer functions to IRRADU00 output, RACF Remote Sharing samples, and a RACF/VM sample. The tool has been re-packaged into a single partitioned data set (PDS) for the sample code and separate files for the LIST3270 and PostScript (PS) documentation. |
| **15 September, 1997** | Version 3, which changed the download location to `ftp.s390.ibm.com` and offered a download via http from the RACF home page at `http://www.s390.ibm.com/racf.` |
| **8 September, 1998** | Version 4, which incorporated these changes:<br>• DFSORT symbol statements for IRRADU00 and IRRDBU00,<br>• Use of // SET to set the default names of IRRDBU00 and IRRADU00 data sets,<br>• Demonstration of new techniques for using DFSORT's ICETOOL utility with IRRDBU00 output,<br>• Documentation in Adobe PDF format, and<br>• Several new reports. |

Changes introduced with this version are marked with a “|.”

[1] RACF is a trademark of the International Business Machines Corporation.

[2] DFSORT is a trademark of the International Business Machines Corporation.

# Contents

# Figures

# Overview

As a security administrator or auditor, you are asked to oversee your corporate data assets and safeguard them from peril or theft.  You have voluminous data before you: procedures, policies, rules for data and system access, audit trails, etc.  Your mission is to find places where interlopers could gain access (“exposures”) and analyze patterns of events to disclose data meddling.  You have a challenging assignment!

You understand the need for system integrity and iron-clad access control.  That's why your “bet-your-business” applications are executed in an MVS[3] environment with RACF as your security manager.

In recent years, RACF has provided two new utilities to assist you in your security administration and auditing missions; the RACF Database Unload Utility (IRRDBU00), introduced in RACF 1.9.0, and the RACF SMF Data Unload Utility (IRRADU00), introduced in RACF 2.1.0.  These utilities share a common philosophy: Take security relevant information such as access rules, user and group definition data, and the audit trail and translate it into a format that you can easily browse, load into the relational database management system of your choice, or process using code that you write.

One method of processing the output of IRRDBU00 and IRRADU00 is with IBM[4]'s DFSORT product.  DFSORT provides a fast and efficient means of data sorting and record selection.  In addition, DFSORT includes a simple yet powerful reporting mechanism called ICETOOL.  DFSORT and ICETOOL assist you in your security administration and auditing tasks through their powerful record selection and reporting capabilities.

[3] MVS is a trademark of the International Business Machines Corporation.

[4] IBM is a registered trademark of the International Business Machines Corporation.

# The RACFICE Tool

This paper along with the sample materials listed below are available on the System/390[5] File Transfer Protocol (FTP) server, in the directory /u/ftp/os390/racf/racfice. You may download these files either by anonymous FTP from ftp.s390.ibm.com, or by following the links from the RACF home page at http://www.s390.ibm.com/racf.

The files that are there are:

| File Name | Description |
|---|---|
| **racfice.xmit** | RACFICE code samples in TSO TRANSMIT format. |
| **racfice.iebupdte.txt** | RACFICE code samples in IEBUPDTE format. |
| **racfice.pdf** | Documentation in Adobe PDF format. |
| **racfice.sampadu.xmit** | Sample IRRADU00 data in TSO TRANSMIT format. |
| **racfice.sampdbu.xmit** | Sample IRRDBU00 data in TSO TRANSMIT format. |

**Note:** These samples are provided for tutorial purposes only. This code has not been submitted to formal IBM testing. This source is distributed on an "as-is" basis, without any warranties either expressed or implied.

The file racfice.xmit is a sequential file in TSO TRANSMIT form which, when unloaded, creates a partitioned data set with the commands required to create sample reports and a JCL procedure which can be used to easily execute ICETOOL.

The members in the RACFICE partitioned data set are:

**Member(s)** **Description**

**$$CNTL$$** JCL to invoke the RACFICE proc for each of the sample reports. This JCL uses the MVS JCLLIB statement to use the member RACFICE as a private proc.

**RACFICE** A JCL procedure that invokes DFSORT's ICETOOL utility, using the output of IRRDBU00 and IRRADU00 as data input.

**xxxx,xxxxCNTL** Each RACFICE report consists of two members in the RACFICE data set. For example, the report that lists all of the users who are RACF SPECIAL, is named SPEC. The two members of the RACFICE data set that define this report are:

| | |
|---|---|
| **SPEC** | ICETOOL definitions of the report format |
| **SPECCNTL** | DFSORT control statements that select the RACF SPECIAL users from the IRRDBU00 output. |

A list of reports contained in the RACFICE package may be found in "Using ICETOOL with the Output of IRRDBU00" on page 13 and "Using ICETOOL with the Output of IRRADU00" on page 15.

**$xxxxxxx** Examples of advanced ICETOOL techniques.

**ADUSYMBL** DFSORT symbol statements for IRRADU00 data.

**DBUSYMBL** DFSORT symbol statements for IRRDBU00 data.

The file racfice.iebupdte.txt is a sequential file in IEBUPDTE form which, when processed by the IEBUPDTE utility, creates a partitioned data set with the commands required to create sample reports and a JCL procedure which can be used to easily execute ICETOOL. The members that are created are the same as are created when using the TSO TRANSMIT format data.

[5] System/390 is a trademark of the International Business Machines Corporation.

RACFICE and ICETOOL are discussed in the article *RACF and DFSORT Security Analysis Tools*, written by Mark Nelson and Frank Yaeger, which appeared in the October, 1996 edition of *Technical Support* Magazine. You can retrieve a PDF format version of this article by visiting the National Association of Systems Programmer's website at http://www.nascom.com and selecting “Technical Support Magazine,” then “1996 Index of Articles” then “Articles,” then “Security,” then *RACF and DFSORT Security Analysis Tools*.

RACFICE requires DFSORT Release 13.

# Installing RACFICE

Installing RACFICE consists of three distinct steps:

1. Downloading the RACFICE code from the `ftp.s390.ibm.com`,
2. Moving the RACFICE code to your MVS or OS/390 system, and
3. Installing the RACFICE code and preparing it for use.

## Downloading the RACFICE Code

There are two ways of downloading the RACFICE code:

1. Use a web browser to download the RACFICE files.

   You can download the RACFICE files using practically any web browser.  To do so:

   a. Open the RACF home page URL, which is `http://www.s390.ibm.com/racf.`,
   b. Scroll down to the section title “Sample Materials,” and click on the RACFICE link, and then
   c. Download each file by clicking each file name.

2. Extract the RACF files from `ftp.s390.ibm.com` by using your file transfer protocol (FTP) program.  You may FTP from either your MVS system or your workstation. Note that if you FTP from your MVS system you can eliminate the next step, described in “Moving the RACFICE Code to Your MVS or OS/390 System.”

   When FTPing the RACFICE files, be sure to set your FTP mode to BINARY for the `racfice.xmit`, `racfice.pdf`, `racfice.sampadu.xmit`, and `racfice.sampdbu.xmit` files.  You must set your FTP mode to ASCII for the `racfice.iebupdte.txt` file.

## Moving the RACFICE Code to Your MVS or OS/390 System

In this step, you move the RACFUCE files to your MVS or OS/390 system.  If you performed the FTP directly from your MVS or OS/390 system, then you may skip this step.

Note that you do not have to download the `racfice.pdf` file. This documentation file is intended for use on your workstation.

You have two choices in moving your data to your MVS or OS/390 system:

1. Use your workstation's terminal emulator or data transfer utility, or
2. Use FTP to transfer the files.

Note that you may perform the FTP from either the MVS or OS/390 system or from your workstation.

No matter what method you use to download the data, you must:

1. Load the `racfice.xmit`, `racfice.sampdbu.xmit`, and `racfice.sampadu.xmit` files in binary.  For FTP, this is done with the FTP BINARY command. Your terminal emulator or data transfer utility may have a command or pull-down menu to set the data transfer mode to BINARY.

2. Load the `racfice.iebupdte.xmit` file in ASCII.  For FTP, this is done with the FTP ASCII command. Your terminal emulator or data transfer utility may have a command or pull-down menu to set the data transfer mode to ASCII or TEXT.

3. Set the characteristics of the output datasets (on MVS or OS/390) to a logical record length (LRECL) of 80 and a record format (RECFM) of fixed blocked (FB).  For FTP, this is normally done with the “SITE LRECL 80,” “LOCSITE LRECL 80,”  “LITERAL SITE LRECL 80,” or “QUOTE SITE LRECL 80” FTP commands.

4. Place the files onto your MVS system with the FTP GET (if your are running FTP from your MVS or OS/390 system) command or the FTP PUT (if you are running FTP from your workstation) command.

## Installing the RACFICE Code

Now that the RACFICE code has been received, you must unpack it. The steps to do this are:

1. Unpack the RACFICE code samples and test data from the RACFICE XMIT data set. This is done by using the TSO RECEIVE command, specifying INDATASET. For example, if you placed the RACFICE XMIT file in the data set `USER01.RACFICE.XMIT`, and the sample data in `USER01.RACFICE.SAMPDBU` and `USER01.RACFICE.SAMPADU`, the commands are:

   ```
   RECEIVE INDATASET('USER01.RACFICE.XMIT')
   RECEIVE INDATASET('USER01.RACFICE.SAMPDBU')
   RECEIVE INDATASET('USER01.RACFICE.SAMPADU')
   ```

   RECEIVE prompts you for the data set name into which it places the RACFICE tool.

2. Update the member `RACFICE` with the default data set names for your IRRADU00 and IRRDBU00 output.

   **Note:** These are defaults only. You can easily override these names when you invoke the RACFICE proc. You may want to run the RACFICE reports using the sample data that is shipped in the `racfice.sampadu.xmit` and `racfice.sampdbu.xmit` files.

3. Update the job card in member `$$CNTL$$` to conform to local requirements. Find the `JCLLIB=` statement and be sure that it points to the data set that you created in 1.

You may then submit `$$CNTL$$` to verify that RACFICE has been installed correctly. You may select what reports are created by commenting out the RACFICE invocations for the reports which you do not want.

# Where Can You Get More Information?

Questions and comments on this package may be directed to the RACF customer forum (MVSRACF CFORUM) on TalkLink[6] , to the RACF-L discussion group, or to the author directly at any of the addresses at the front of this paper.

## How to Get to the RACF-L Discussion Group

RACF-L is an unmoderated discussion list for RACF questions, etc. that is accessible via the Internet.

You need to know two addresses to use RACF-L. For administrative messages (subscribe, help, etc.) you would send a message to `listserv@uga.cc.uga.edu` with your “command” in the body of the message. First you should subscribe so you receive future postings. To do that, the body of your administrative message would contain `subscribe racf-l your_first_name your_last_name`.

Once you've subscribed, you can find out other capabilities of the list by sending an administrative message with the body containing the command `help`.

To actually post to the list, you would send your message, with an appropriate subject line, to `racf-l@uga.cc.uga.edu`.

[6] TalkLink is a trademark of the International Business Machines Corporation.

# Using DFSORT's ICETOOL Utility

This section contains an overview of ICETOOL. This paper is not intended to be a comprehensive description of ICETOOL and, in fact, only discusses a few of the twelve available ICETOOL operators (COPY, COUNT, DEFAULTS, DISPLAY, MODE, OCCURS, RANGE, SELECT, SORT, STATS, UNIQUE and VERIFY). You can find complete details about DFSORT and ICETOOL in *DFSORT Application Programming Guide (SC33-4035).* You can find articles, news, tips, techniques, examples, and even an ICETOOL Mini-User Guide on the DFSORT home page at URL `http://www.ibm.com/storage/dfsort/` and you can download DFSORT/ICETOOL papers and examples by FTP using links from the home page.

You can find more information on RACF at the RACF home page at URL `http://www.s390.ibm.com/racf/`

You can think of ICETOOL as a “wrap-around” application to the DFSORT product, with ICETOOL providing report writer-like functions. ICETOOL creates the report headers, page numbers, date and time stamps, page separation, and summary information. DFSORT provides the record selection and record ordering. The best way to describe ICETOOL is with examples. Figure 1 on page 9 shows the job control language (JCL) and control statements required to find all of the IRRADU00 records that are applicable to a specific user ID. Let's take these statements apart one at a time, starting with the JCL.

## JCL to Invoke ICETOOL

This section describes the JCL statements required to invoke ICETOOL. The JCL statements required are:

| **Statement** | **Use** |
|---|---|
| **JOB** | Initiates the job. |
| **EXEC** | Specifies the program name (PGM=ICETOOL). |
| **TOOLMSG DD** | Message data set for ICETOOL-created messages |
| **PRINT DD** | Data set for the report or reports that are being produced. The name of this DD is determined by the `LIST` keyword in the `DISPLAY` operator. |
| **DFSMSG DD** | Data set for the messages produced by DFSORT as it selects records. After your reports are developed and debugged, you may want to allocate this DD to DUMMY to reduce the amount of data that is created. |
| **TOOLIN DD** | Control statements for ICETOOL |
| **INDD DD** | The output from IRRADU00 that is the input for this report. The name of this DD is determined by the `FROM` keyword in the `SORT` statement. |
| **TEMP0001** | A temporary work data set. The name of this DD is determined by the `TO` keyword in the `SORT` statement. |

# ICETOOL Control Statements

The previous section described the JCL statements required to invoke ICETOOL. This section describes the control statements required to select records, create the desired report headings, and print the report. In the example in Figure 1 on page 9, we are creating a report that lists all of the events associated with a specific user ID, in this case IBMUSER.

**Note:** IBM recommends that the IBMUSER user ID be revoked immediately after RACF installation and not used as a user ID. The query and report shown in Figure 1 on page 9 and Figure 2 on page 10 are examples only.

To create this report, our ICETOOL job consists of two parts:

1. Select the records and copy them to a temporary file using the `COPY` operator. We are copying the records from the data set allocated to `INDD` to the data set allocated to `TEMP0001`. The input and output DD names are identified using the `FROM` and `TO` keywords. ICETOOL's COPY operator uses the DFSORT control statements that are allocated to the `SELUCNTL` DD statement. **The DFSORT control statements select the desired records.** The DD name used for the DFSORT control statements is identified using the `USING` keyword. This keyword identifies the first part of the DD name. The complete DD name is created by appending the characters “`CNTL`” to the `USING` value. Note that the `USING` value must be exactly 4 characters.

   In this example, the `INCLUDE` statement selects all of the records which have the string “`ACCESS  `” in columns 8 to 15 and the string “`IBMUSER `” in columns 63 to 70.

2. Select the fields for listings and create the report headings using the `DISPLAY` operator.

   The `DISPLAY` operator is where we define the report headings and identify the fields that we want in the report. In our example, we read the data in from the temporary data set identified in the `FROM` keyword. Our report is directed to the DD statement identified in the `LIST` keyword. The `PAGE`, `TITLE`, `DATE`, and `TIME` keywords identify the information that we want placed in the page title line.

   The selection of the fields for inclusion in the report is done using the `ON` keyword. Each `ON` keyword identifies one field for the report by the starting position, length, and data type for that field. Since all IRRADU00 and IRRDBU00 data is character data, the data type “`CH`” is usually used.

   The `HEADER` keyword is used to assign a column heading to the field selected in the `ON` keyword.

   The `BLANK` keyword instructs ICETOOL to adjust the width for each column automatically according to the size of the ON field and its HEADER.

```
//JOBNAME   JOB Job Card......
//SELECT    EXEC PGM=ICETOOL
//TOOLMSG   DD SYSOUT=*
//PRINT     DD SYSOUT=*
//DFSMSG    DD DUMMY
//TOOLIN    DD *
***********************************************************************
* Find all of the records which are applicable to a specific          *
* user ID.                                                            *
*                                                                     *
* The DFSORT "INCLUDE" statement is used to select the user ID.       *
* The DFSORT control statements are pointed to by the ICETOOL         *
* "USING" keyword.                                                    *
***********************************************************************
 COPY    FROM(INDD) TO(TEMP0001) USING(SELU)
 DISPLAY FROM(TEMP0001) LIST(PRINT) -
         PAGE -
         TITLE('Events Associated with a Specific User')-
         DATE(YMD/) -
         TIME(12:)  -
         BLANK -
         ON(63,8,CH)  HEADER('User ID') -
         ON(5,8,CH)   HEADER('Event') -
         ON(14,8,CH)  HEADER('Qualifier') -
         ON(23,8,CH)  HEADER('Time') -
         ON(32,10,CH) HEADER('Date') -
         ON(43,4,CH)  HEADER('System') -
         ON(175,8,CH) HEADER('Terminal') -
         ON(184,08,CH) HEADER('Jobname')
//INDD      DD DISP=SHR,DSN=MARKN.TYPE80.IRRADU00
//TEMP0001  DD DISP=(NEW,DELETE,DELETE),SPACE=(CYL,(20,5,0))
//SELUCNTL  DD *
 INCLUDE COND=(5,8,CH,EQ,C'ACCESS',AND,
               63,8,CH,EQ,C'IBMUSER')
 OPTION   VLSHRT
/*
```

Figure 1. Finding all IRRADU00 Records Associated with a Specific User ID. These are the JCL and control statements required to find all of the RACF SMF Data Unload records associated with a specific user ID.

## 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 (SC23-3732)*. In our example, the field for the user ID associated with an event is defined in *RACF Macros and Interfaces* as beginning at record position 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 `ON` keywords to select the fields for the report.

```
- 1 -        Events Associated with a Specific User        94/08/30        05:14:49 pm

User ID    Event      Qualifier   Time       Date         System    Terminal   Jobname
--------   --------   ---------   --------   ----------   ------    --------   --------
IBMUSER    ACCESS     SUCCESS     15:49:59   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:27   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:27   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:27   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:27   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:28   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:28   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:28   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:28   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:28   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:28   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:28   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:50:29   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:52:53   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:52:55   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     15:52:58   1994-02-08   IM13                 IBMUSERM
IBMUSER    ACCESS     SUCCESS     15:52:58   1994-02-08   IM13                 IBMUSERM
IBMUSER    ACCESS     SUCCESS     15:53:28   1994-02-08   IM13                 IBMUSERM
IBMUSER    ACCESS     SUCCESS     16:28:55   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:28:55   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:28:55   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:28:55   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:29:02   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:29:02   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:29:02   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:29:02   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:30:08   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:42:09   1994-02-08   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     16:42:11   1994-02-08   IM13                 IBMUSERM
IBMUSER    ACCESS     SUCCESS     16:42:12   1994-02-08   IM13                 IBMUSERM
IBMUSER    ACCESS     SUCCESS     16:42:32   1994-02-08   IM13                 IBMUSERM
IBMUSER    ACCESS     SUCCESS     17:10:21   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:11:01   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:11:01   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:11:01   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:11:02   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:42:00   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:42:01   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:42:01   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:42:36   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:42:36   1994-02-10   IM13      LOCALC10   IBMUSER
IBMUSER    ACCESS     SUCCESS     17:43:11   1994-02-10   IM13      LOCALC10   IBMUSER
```

Figure 2. Report for all IRRADU00 Records Associated with a Specific User ID. This report was created by the JCL and control statements in Figure 1.

# The RACFICE PROC

The RACFICE member contains a JCL proc which simplifies the JCL required to execute RACFICE reports. This proc contains JCL symbolic variables that represent the input to the RACFICE tool. These variables are:

**DBUDATA** Output of IRRDBU00 that is being used as input to the RACFICE tool.

**ADUDATA** Output of IRRADU00 that is being used as input to the RACFICE tool.

**REPORT** The name of the report that is being generated. See “Using ICETOOL with the Output of IRRDBU00” on page 13 and “Using ICETOOL with the Output of IRRADU00” on page 15 for lists of the reports that are shipped with RACFICE. “Creating Your Own Reports” on page 17 describes how you can create your own.

You don't need to specify each of these variables each time you execute the RACFICE proc. For example, if the default IRRDBU00 and IRRADU00 data sets have been specified in the RACFICE proc in step 2 on page 5, then you create the report that lists all of the audit records for a specific user (the report that is shown in Figure 2 on page 10) with the JCL:

```
//jobname  JOB  Job card...
//stepname EXEC RACFICE,REPORT=SELU
```

If the default IRRDBU00 or IRRADU00 data sets are not correct, you can override them. For example, if your IRRDBU00 output is in the data set USER01.TEST.IRRDBU00, and your IRRADU00 output is in the data set USER01.TEST.IRRADU00, then code:

```
//jobname  JOB  Job card...
//         SET  ADUDATA=USER01.TEST.IRRADU00
//         SET  DBUDATA=USER01.TEST.IRRDBU00
//stepname EXEC RACFICE,REPORT=SELU
```

Be sure to specify on the JCLLIB data set the data set which contains the RACFICE PROC.

## Can Records from RACF/VM Be Processed by IRRADU00?

Yes they can! The format of the SMF records created by RACF/MVS and RACF/VM is identical. With RACF/VM 1.9.2, you must move them to your RACF/MVS system for processing. This requires the use of the SMFCONV utility on RACF/VM. See the RACF Auditor's Guide for details on the SMFCONV utility.

RACF/VM 1.10 supports the RACF SMF Unload Utility from VM, eliminating the need for moving data to MVS. See the announcement letter for RACF/VM, 296-109, for more details. You can find RACF announcement letters on the World Wide Web from the RACF home page at ttp://www.s390.ibm.com/racf/.

# Using ICETOOL with the Output of IRRDBU00

The `racfice.xmit` file contains several sample reports based on the output of IRRDBU00. These reports are:

**Name** **Description**

**ASOC** Users who have explicit RACF Remote Sharing Facility (RRSF) associations defined.
**Value:** Identifies users who can direct commands.

**BGGR** Discrete general resource profiles with generic characters.
**Value:** Finds profiles which aren't protecting what you think they are protecting.

**CCON** Count of user's connections, flagging those users with more than “x” connections.
**Value:** Helps find a performance bottleneck caused by excessive group connections.

**CGEN** Count of general resource profiles.
**Value:** Identifies basic characteristics of your RACF database.

**CUGD** Count of user, group, and data set profiles.
**Value:** Identifies basic characteristics of your RACF database.

**CONN** User IDs With group privileges above use.
**Value:** Identifies users with additional privileges.

**IDSC** Dataset conditional access list entries with an ID(*) entry of other than NONE.
**Value:** Identifies dataset profiles that allow any RACF-authenticated user to access data.

**IDSS** Dataset standard access list entries with an ID(*) entry of other than NONE.
**Value:** Identifies dataset profiles that allow any RACF-authenticated user to access data.

**IGRC** General resource conditional access list entries with an ID(*) entry of other than NONE.
**Value:** Identifies general resource profiles that allow any RACF-authenticated user to access data.

**IGRS** General resource standard access list entries with an ID(*) entry of other than NONE.
**Value:** Identifies general resource profiles that allow any RACF-authenticated user to access data.

**OMVS** User IDs Which Have a UNIX System Services (OMVS) segment defined.
**Value:** Identifies users who can use OS/390's UNIX Systems Services with a non-default UID.

**SUPU** UNIX[7] System Services “super users” (UID of Zero)
**Value:** Identifies users who have extraordinary privileges within the OS/390 UNIX System Services environment.

**UADS** Dataset profiles with UACCs other than NONE.
**Value:** Identifies dataset profiles that allow any user to access data.

**UAGR** General resource profiles with UACCs other than NONE.
**Value:** Identifies general resource profiles that allow any user to access data.

**UGLB** User IDs with extraordinary global authorities.
**Value:** Identifies users with extraordinary RACF authority.

**UGRP** User IDs With extraordinary RACF group authorities.
**Value:** Identifies users with extraordinary RACF authority.

**UIDS** UNIX System Services UIDs which are used more than once.
**Value:** Identifies UNIX System Services users who are sharing authority characteristics.

**URVK** User IDs which are currently revoked.
**Value:** Identifies users who have had a revocation performed.

In addition, these RACFICE contains these reports which demonstrate advanced ICETOOL techniques:

**Name** **Description**

**$CFQG** A count of the number of fully-qualified generic profiles that are defined for each high-level qualifier (HLQ).
**Value:** Identifies users who defining excessive fully qualified generic profiles.

**$CHLQ** A count of the number of generic profiles that are defined for each high-level qualifier (HLQ).
**Value:** Identifies a potential performance bottleneck.

**$ULAST90** Identifies the user profiles which have been created within the past 90 days.
**Value:** Shows recent administrative activity.

[7] UNIX is a registered trademark in the United States and/or other countries, licensed exclusively through X/Open Company Limited.

# Using ICETOOL with the Output of IRRADU00

The `racfice.xmit` file contains several sample reports based on the output of IRRADU00. These reports are:

**Name** **Description**

**ACD$** Users who are using automatic command direction.
**Value:** Identifies users who are using the RACF Remote Sharing Facility.

**ECD$** Users who are directing commands explicitly.
**Value:** Identifies users who are using the RACF Remote Sharing Facility.

**LOGB** Users who log on with LOGON BY.
**Value:** Identifies users who are logging on as another user.

**LOGF** All users with excessive incorrect passwords.
**Value:** Identifies users who exceeding a “bad password” threshold. This threshold is independent of the SETROPTS PASSWORD(REVOKE(nn)) value.

**OPER** Accesses allowed because the user has OPERATIONS.
**Value:** Control of users with the OPERATIONS attribute.

**PWD$** Users who are using password synchronization.
**Value:** Identifies users who are using the RACF Remote Sharing Facility.

**RACL** RACLINK audit records.
**Value:** Identifies users who are using the RACF Remote Sharing Facility.

**SELU** All audit records for a specific user.
**Value:** Reporting on all audited events for a user.

**SPEC** Accesses allowed because the user has SPECIAL.
**Value:** Control of users with the SPECIAL attribute.

**PWD$** Users who are using password synchronization
**Value:** Identifies users who are using the RACF Remote Sharing Facility.

**TRMF** Excessive incorrect passwords from terminals
**Value:** Identifies intruders who are attempting to guess password but are moving from one ID to another to avoid the revocation of user IDs.

**VIOL** Access Violations.
**Value:** Identifies failed events.

**WARN** Accesses Allowed Due to WARNING Mode Profiles.
**Value:** Identifies events which were allowed but which you may want to prevent in the future or explicitly allow by changing a profile's access list.

# Creating Your Own Reports

You can create your own reports using the RACFICE proc. To do so:

1. Identify the specific information that you want selected. Create the DFSORT selection criteria and place it in the RACFICE data set with a unique one to four character report identifier appended with the characters "CNTL". If there is an existing RACFICE report that has similar selection criteria, use it as a model.
2. Define the report format using ICETOOL command syntax. Place the ICETOOL commands into the RACFICE data set under the one to four character report identifier that you have selected.
3. Update your report JCL to invoke the RACFICE PROC with the REPORT keyword set to the one to four character report identifier.

# Advanced Techniques: Using DFSORT Symbols

On 8 September, 1998, IBM annouced release 14 of its sorting and reporting product, DFSORT. Along with many new features and enhancements, this release contains a new symbols processing feature of particular interest to users of DFSORT's ICETOOL reporting utility.

With DFSORT Release 14, you can define symbols to replace fields and constants in DFSORT and ICETOOL control statements. Symbols make your control statements easier to read, understand and maintain.

To increase your productivity, the RACF team has already created the DFSORT symbol names for all of the IRRADU00 and IRRDBU00 records, using the same names RACF uses in the DB2 tables, as documented in "OS/390 Security Server (RACF) Macros and Interfaces". This allows you to create symbol data sets for these records and use them with DFSORT and ICETOOL.

For example, before release 14, if you wanted to select all of the IRRADU00 ACCESS records associated with the user ID USER01, you would code the following DFSORT statement:

```
  INCLUDE COND=(5,8,CH,EQ,C'ACCESS',AND,63,8,CH,EQ,C'USER01')
```

Note that you had to know that IRRADU00 placed the "event code" in column 5 for 8 characters, and the user ID in column 63 for 8 characters.

With release 14, you could specify the supplied ADUSYMBL data set in your DFSORT run and use this DFSORT statement:

```
  INCLUDE COND=(INIT_EVENT_TYPE,EQ,C'ACCESS',AND,
                INIT_EVT_USER_ID,EQ,C'USER01')
```

Once you make the supplied symbol data sets available, you no longer have to figure out the positions, lengths or formats of fields, or worry about offsets vs positions or whether to add 4 for the record descriptor word (RDW). Just use the names for the fields you want.

For details on DFSORT Release 14 and its symbols processing feature, visit the DFSORT Web page at: http://www.ibm.com/storage/dfsort/

Version 4 of RACFICE provides the DFSORT symbol names you'll need in members ADUSYMBL, for IRRADU00 records, and DBUSYMBL, for IRRDBU00 records.

Note that the reports in RACFICE do not use DFSORT symbols to avoid the requirement that DFSORT Release 14 be available. However, the IRRADU00 and IRRDBU00 symbol definitions provided can be used with DFSORT Release 14 for your own reports.

# Advanced Techniques: Using OUTFIL

DFSORT Release 13 introduced the powerful OUTFIL operator, which allows you to create one or more output data sets from a single pass over your input data.

You can use OUTFIL to subset your IRRDBU00 or IRRADU00 data. For example, if you want to separate our your user basic data (0200) and group basic data (0100) for your help desk, OUTFIL can easily be used.

OUTFIL has other uses as well. For example, would you like to count all of the generic data set profiles under each high-level qualifier (HLQ)? You can do just that using OUTFIL and DFSORT's "string search" (SS) operator.

There are three steps to get this report:

1. Select all of the data set basic data records (columns 5[8] to 8 contain "0400") which represent data set profiles which are generic (columns 62 to 65 contain "YES ").

   We do this with DFSORT statements that look like:

   ```
   SORT    FIELDS=(10,44,CH,A)
   INCLUDE COND=(5,4,CH,EQ,C'0400',AND,62,4,CH,EQ,C'YES ')
   ```

2. Using the "." in the data set name as a delimiter, place each of the HLQs into one of eight output data sets, depending upon the length of the HLQ. For example, the HLQs "A," "B," and "C" all go to one dataset, while "ABC," "DEF," and "GHI." go to another.

   We do this with DFSORT OUTFIL statements that look like:

   ```
   OUTFIL  INCLUDE=(11,1,CH,EQ,C'.',OR,11,1,CH,EQ,C' '),
           OUTREC=(1,4,5,4,X,10,1,7X),FNAMES=DS$1
   OUTFIL  INCLUDE=(12,1,CH,EQ,C'.'),
           OUTREC=(1,4,5,4,X,10,2,6X),FNAMES=DS$2
   OUTFIL  INCLUDE=(13,1,CH,EQ,C'.',AND,10,3,SS,NE,C'.'),
           OUTREC=(1,4,5,4,X,10,3,5X),FNAMES=DS$3
   OUTFIL  INCLUDE=(14,1,CH,EQ,C'.',AND,10,4,SS,NE,C'.'),
           OUTREC=(1,4,5,4,X,10,4,4X),FNAMES=DS$4
   OUTFIL  INCLUDE=(15,1,CH,EQ,C'.',AND,10,5,SS,NE,C'.'),
           OUTREC=(1,4,5,4,X,10,5,3X),FNAMES=DS$5
   OUTFIL  INCLUDE=(16,1,CH,EQ,C'.',AND,10,6,SS,NE,C'.'),
           OUTREC=(1,4,5,4,X,10,6,2X),FNAMES=DS$6
   OUTFIL  INCLUDE=((17,1,CH,EQ,C'.',OR,17,1,CH,EQ,C' '),
           AND,10,7,SS,NE,C'.'),
           OUTREC=(1,4,5,4,X,10,7,1X),FNAMES=DS$7
   OUTFIL  SAVE,
           OUTREC=(1,4,5,4,X,10,8),FNAMES=DS$8
   OPTION   VLSHRT
   ```

   Each of the OUTFIL statements is looking for a "." separator in a specific column. For example, OUTFIL finds the "." in column 11, we know that this is a single character HLQ, and we send it to the dataset allocated to DDNAME DS$1.

   We do a similar test for bytes 2 through 9 of the data set name. Note that we don't test the first byte (byte 10) of the data set name, since you can't have a dataset profile that has no HLQ.

[8] Remember that these column numbers include the record descriptor word. See "An Important Note on Column Numbers" on page 9 for details.

Note that starting with our test for the “.” in the fourth character of the data set name (column 13) we've added an additional test. The reason for this additional test is that from the fourth byte of the data set name to the end it is possible to find a “.” that isn't a delimiter for the HLQ. How is that possible? Consider the data set name “A.B.C” The “.” that separates qualifier “B” from qualifier “C” occurs in the fourth byte of the data set name. However, it does not delineate the **HLQ**. To ensure that we aren't fooled by multiple qualifiers in the first eight bytes of the data set name, we add a second test to the OUTFIL statement stating that the record should be written to DDNAME DS$3 only if the fourth byte contains the “.” **and none of the preceding bytes contained a period (“.”). We do the latter test using DFSORT's “string search” operator, SS.** We add a similar test to all of the subsequent OUTFIL statements.

3. Using all of the data sets created in 2 on page 19, we use ICETOOL to count the records which represent generic data sets. 2 on page 19 placed the HLQ as the first 8 bytes of the output record, so all we need to do is count based on those bytes. Since we are looking only for those IDs which have an excessive number of generic profiles under a specific HLQ, we filter out those IDs which have less than our threshold. We set the threshold using ICETOOL's HIGHER(nnn) operator, which in this example is 200.

   We do this with DFSORT ICETOOL statements that look like:

```
OCCURS  FROM(DBUDATA) LIST(PRINT) -
        PAGE -
        TITLE('HLQs With Excessive Generic Profiles')-
        DATE(YMD/) -
        TIME(12:)  -
        BLANK -
        ON(10,8,CH)   HEADER('HLQ') -
        ON(VALCNT)    HEADER('Count') -
        HIGHER(200)
```

This report may be found in member $CHLQ in `racfice.xmit` and `racfice.iebupdte.txt.` A sample output from this report is shown in Figure 3.

```
- 1 -        HLQs With Excessive Generic Profiles        98/09/02         10:56:00 pm

HLQ                 Count
--------   ---------------
FRED                  299
```

Figure 3. Sample Output From $CHLQ Report

A similar report in member $CFQG counts all of the **fully-qualified** data set profiles for each HLQ. This report is useful in ensuring that your user community isn't creating large numbers of fully-qualified data set profiles.

# Advanced Techniques: Date-Sensitive Processing

One common reporting requirement is the creation of reports which contain data for a prior time period. This "rolling window" can be several days, weeks, or months in length. An example of this type of report is a report which shows all of the user profiles created within the past 90 days. Can ICETOOL be used to create such a report? By itself, no. However, using TSO's REXX programming interface, we can dynamically create DFSORT and ICETOOL statements to perform the data extraction and report.

Member $ULAST90 in `racfice.xmit` and `racfice.iebupdte.txt` contains the JCL to create the DFSORT and ICETOOL control statements to list all of the user profiles that were created in the past 90 days. This sample can consists of three distinct parts:

1. Copying the REXX exec which creates the DFSORT and ICETOOL control statements to a temporary data set, which is shown in Figure 4 and Figure 5 on page 22,
2. Executing the EXEC to create the DFSORT and ICETOOL control statements, which is shown in Figure 6 on page 22,
3. Executing ICETOOL to create the report, which is shown in Figure 7 on page 23.

```
//MARKNICE JOB MSGLEVEL=(0,0),CLASS=5,NOTIFY=&SYSUID,MSGCLASS=H
/*JOBPARM  S=ANY,LINES=99
//         SET DBUDATA=MARKN.TEST.IRRDBU00
//*******************************************************************
//MAKEEXEC EXEC PGM=ICEGENER
//SYSUT1   DD   DATA,DLM=ZZ
 /* REXX ************************************************************/
 /**       REXX EXEC to find all user profiles that were           **/
 /**       created in the last <user_age> days                     **/
 /*******************************************************************/


 /*******************************************************************/
 /**  Set the variable user_age below to the number of days        **/
 /**  for this "rolling window". That is, find all of the          **/
 /**  user profiles that were created in the past <user_age>       **/
 /**  days.                                                        **/
 /*******************************************************************/
 user_age              = 90

 /*******************************************************************/
 /**  Get the current date in REXX base format and standard format.**/
 /**  Convert the standard format into IRRADU00 and IRRDBU00       **/
 /**  output format (YYYY-MM-DD).                                  **/
 /*******************************************************************/
 current_days=DATE('B')
 current_date=DATE('S',current_days,'B')
 current_date=INSERT("-",current_date,4)
 current_date=INSERT("-",current_date,7)


 /*******************************************************************/
 /**  Calculate the selection criteria date by subtracting         **/
 /**  <user_age> from the current date. Convert this to IRRADU00   **/
 /**  and IRRDBU00 output (YYYY-MM-DD).                            **/
 /*******************************************************************/
 user_days             = current_days - user_age

 user_date             = DATE('S',user_days,'B')
 user_date=INSERT("-",user_date,4)
 user_date=INSERT("-",user_date,7)

 Say "Today is "  current_date
 Say "Processing user profiles defined after "  user_date
```

Figure 4. Copying the Sample REXX Code to a Temporary Dataset (Part 1)

```
 /*******************************************************************/
 /**  Create DFSORT control statements to select all of the        **/
 /**  "User Basic Data" records (0200) where the creation date     **/
 /**  is greater or equal to the date that we calculated above     **/
 /*******************************************************************/
 sortcntl.1=" INCLUDE COND=(5,4,CH,EQ,C'0200',AND,"
 sortcntl.2="        19,10,CH,GE,C'"||user_date||"')"
 sortcntl.3=" OPTION VLSHRT"

 /*******************************************************************/
 /**  Write the DFSORT control statements to a dataset.            **/
 /*******************************************************************/
 execio 3 diskw SORTCNTL "(ST"  sortcntl.

 /*******************************************************************/
 /**  Create DFSORT ICETOOL control statements to create a         **/
 /**  report showing the creation date, user ID, creator,          **/
 /**  and characteristics of the user ID.                          **/
 /*******************************************************************/
 icecntl.1= " COPY    FROM(DBUDATA) TO(TEMP0001) USING(SORT)         "
 icecntl.2= " DISPLAY FROM(TEMP0001) LIST(PRINT) -                   "
 icecntl.3= "         PAGE -                                         "
 icecntl.4= "         TITLE('User Defined Within the past "||user_age||,
                      " days')-                                      "
 icecntl.5= "         DATE(YMD/) -                                   "
 icecntl.6= "         TIME(12:)  -                                   "
 icecntl.7= "         BLANK -                                        "
 icecntl.8= "         ON(19,10,CH)  HEADER('Date') -                 "
 icecntl.9= "         ON(10,8,CH)   HEADER('User ID') -              "
 icecntl.10="         ON(30,8,CH)   HEADER('Owner') -                "
 icecntl.11="         ON(44,4,CH)   HEADER('Special') -              "
 icecntl.12="         ON(49,4,CH)   HEADER('Operations') -           "
 icecntl.13="         ON(385,4,CH)  HEADER('Auditor')  -             "
 icecntl.14="         ON(118,10,CH) HEADER('Last Date') -            "
 icecntl.15="         ON(109,8,CH)  HEADER('Last Time')              "

 /*******************************************************************/
 /**  Write the DFSORT ICETOOL control statements to a dataset.    **/
 /*******************************************************************/
 execio 15 diskw ICECNTL "(ST"   icecntl.


 /*******************************************************************/
 /**  END!                                                         **/
 /*******************************************************************/
ZZ
//SYSUT2   DD   DISP=(NEW,PASS,DELETE),DSN=&&CLIST(REXXICE),
//         SPACE=(TRK,(10,2,5)),
//         UNIT=SYSALLDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   DUMMY
```

Figure 5. Copying the Sample REXX Code to a Temporary Dataset (Part 2)

```
//*******************************************************************
//EXECREXX EXEC PGM=IKJEFT01,PARM='%REXXICE'
//SYSTSPRT DD SYSOUT=*
//SYSPROC  DD DISP=(OLD,DELETE,DELETE),DSN=&&CLIST
//SYSTSIN  DD DUMMY
//SORTCNTL DD DISP=(NEW,PASS,DELETE),DSN=&&SORTCNTL,
//         SPACE=(TRK,(1,0,0)),
//         UNIT=SYSALLDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//ICECNTL  DD DISP=(NEW,PASS,DELETE),DSN=&&ICECNTL,
//         SPACE=(TRK,(1,0,0)),
//         UNIT=SYSALLDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
```

Figure 6. Executing the Sample REXX Code to Create DFSORT and ICETOOL control statements

```
//*******************************************************************
//RACFICE   EXEC PGM=ICETOOL
//TOOLMSG   DD SYSOUT=*
//PRINT     DD SYSOUT=*
//DFSMSG    DD SYSOUT=*
//DBUDATA   DD DISP=SHR,DSN=&DBUDATA
//TEMP0001  DD DISP=(NEW,DELETE,DELETE),SPACE=(CYL,(20,5,0))
//SORTCNTL  DD DISP=SHR,DSN=&SORTCNTL
//TOOLIN    DD DISP=SHR,DSN=&ICECNTL
```

Figure 7. Executing ICETOOL to Create the Time-Sensitive Report

Sample output from this report is shown in Figure 8.

```
- 1 -        User Defined Within the past 90 days        98/09/02        10:59:20 pm

Date         User ID    Owner      Special   Operations   Auditor   Last Date    Last Time
----------   --------   --------   -------   ----------   -------   ----------   ---------
1998-09-02   BARNEY     MARKN      NO        NO           NO
1998-09-02   BIGGUY     MARKN      NO        NO           NO
1998-09-02   BMOC       MARKN      NO        NO           NO
1998-09-02   CONNAUTH   MARKN      NO        NO           NO
1998-09-02   CREAAUTH   MARKN      NO        NO           NO
1998-09-02   DSUACC     MARKN      NO        NO           NO
1998-09-02   FRED       MARKN      NO        NO           NO
1998-09-02   GLBAUDIT   MARKN      NO        NO           NO
1998-09-02   GLBOPER    MARKN      NO        YES          NO
1998-09-02   GLBSPEC    MARKN      YES       NO           NO
1998-09-02   HYDE       MARKN      NO        NO           NO
1998-09-02   IDSC       MARKN      NO        NO           NO
1998-09-02   IDSS       MARKN      NO        NO           NO
```

Figure 8. Sample Output From $CHLQ Report

# Appendix A. Can DFSORT's ICETOOL Utility Replace the RACF Report Writer?

For many reporting needs, ICETOOL can replace the RACF Report Writer. With the introduction of the RACF SMF Data Unload Utility in 1994, IBM stabalized the RACF Report Writer at the RACF 1.9.2 level of function. This does not mean that the Report Writer is no longer supported; It means that there are no future enhancements planned for the Report Writer and that requirements against the Report Writer are no longer being accepted.

The rationale for this decision is that there are many products for an installation to use to create reports and analyze data, and that requiring the use of a proprietary report generator for the analysis of security data didn't make as much sense as formatting the security data into a format that could then be used by the report generation tool of the customer's choice. The added benefit of this decision is that many of these report generation tools, such as DFSORT's ICETOOL, have better selection and analysis capabilities than the RACF Report Writer, and are actively enhanced over time to provide even more capabilities of which you can take advantage.

Figure 9 and Figure 10 on page 25 show mappings of frequently-used RACF Report Writer statements and the equivalent location of that information in the IRRADU00 records.

| **RACF Report Writer Statement** | **IRRADU00 Equivalent** | **Position in IRRADU00 Records** |
|---|---|---|
| DATE | IFASMFDP DATE control statement<br>DATE_WRITTEN field on each record | Not applicable<br>28 |
| TIME | IFASMFDP TIME control statement<br>TIME_WRITTEN field on each record | Not applicable<br>19 |
| VIOLATIONS | VIOLATION field on each record | 44 |
| SUCCESSES | Event Qualifier on each record not equal to “SUCCESS” | 10 |
| WARNINGS | USER_WARNING field on each record | 54 |
| USER/NOUSER | EVT_USER_ID field on each record | 59 |
| JOB/NOJOB | JOB_NAME field on each record | 180 |
| OWNER/NOOWNER | OWN_ID field on the record | Depends on the record type |
| GROUP/NOGROUP | EVT_GRP_ID field on each record | 68 |
| STEP | There is no IRRADU00 equivalent | Not applicable |
| STATUS | Select only the RACINIT, SETROPTS, and RVARY command records | Not applicable |
| SYSID | EVT_SYSTEM_SMFID field on each record | 39 |
| TERMINAL | TERM | 171 |

Figure 9. RACF Report Writer Equivalents in IRRADU00 Output (Part 1). This is part 1 of the table which defines the most frequently-used RACF Report Writer functions and identifies the equivalent in the records created by IRRADU00.

| **RACF Report Writer Statement** | **IRRADU00 Equivalent** | **Position in IRRADU00 Records** |
|---|---|---|
| AUTHORITY | | |
| NORMAL | AUTH_NORMAL | 77 |
| SPECIAL | AUTH_SPECIAL | 82 |
| OPERATIONS | AUTH_OPER | 87 |
| AUDITOR | AUTH_AUDIT | 92 |
| EXIT | AUTH_EXIT | 97 |
| FAILSOFT | AUTH_FAILSFT | 102 |
| BYPASSED | AUTH_BYPASS | 107 |
| TRUSTED | AUTH_TRUSTED | 112 |
| REASON | | |
| CLASS | LOG_CLASS | 117 |
| USER | LOG_USER | 122 |
| SPECIAL | LOG_SPECIAL | 127 |
| RESOURCE | LOG_ACCESS | 132 |
| COMMAND | LOG_ALWAYS | 142 |
| CMDVIOL | LOG_CMDVIOL | 147 |
| AUDITOR | LOG_GLOBAL | 152 |
| SECAUDIT | LOG_LEVEL | 157 |
| VMAUDIT | LOG_VMEVENT | 223 |
| SECLABELAUDIT | LOG_SECL | 233 |
| LOGOPTIONS | LOG_LOGOPT | 228 |
| COMPATMODE | LOG_COMPATM | 238 |
| APPLAUDIT | LOG_APPLAUD | 243 |

Figure 10. RACF Report Writer Equivalents in IRRADU00 Output (Part 2). This is part 2 of the table which defines the most frequently-used RACF Report Writer functions and identifies the equivalent in the records created by IRRADU00.

## Figures

| id | File | Page | References |
|---|---|---|---|
| EXAMP1 | RACFICE VERSION4 | 9 | 1<br>7, 8, 8, 10 |
| RESULT1 | RACFICE VERSION4 | 10 | 2<br>8, 11 |
| CHLQRPT | RACFICE VERSION4 | 20 | 3<br>20 |
| REXX1A | RACFICE VERSION4 | 21 | 4<br>21 |
| REXX1B | RACFICE VERSION4 | 22 | 5<br>21 |
| REXXIC2 | RACFICE VERSION4 | 22 | 6<br>21 |
| REXXIC3 | RACFICE VERSION4 | 23 | 7<br>21 |
| ULSTRPT | RACFICE VERSION4 | 23 | 8<br>23 |
| RWQUIV1 | RACFICE VERSION4 | 24 | 9<br>24 |
| RWQUIV2 | RACFICE VERSION4 | 25 | 10<br>24 |

## Headings

| id | File | Page | References |
|---|---|---|---|
| 2MVS | RACFICE VERSION4 | 4 | Moving the RACFICE Code to Your MVS or OS/390 System<br>4 |
| COLNOTE | RACFICE VERSION4 | 9 | An Important Note on Column Numbers<br>19 |
| DBUREPS | RACFICE VERSION4 | 13 | Using ICETOOL with the Output of IRRDBU00<br>2, 11 |
| ADUREPS | RACFICE VERSION4 | 15 | Using ICETOOL with the Output of IRRADU00<br>2, 11 |
| OWNRPTS | RACFICE VERSION4 | 17 | Creating Your Own Reports<br>11 |
| SYMBLS | RACFICE VERSION4 | 18 | Advanced Techniques: Using DFSORT Symbols |
| OUTFIL | RACFICE VERSION4 | 19 | Advanced Techniques: Using OUTFIL |
| DATEP | RACFICE VERSION4 | 21 | Advanced Techniques: Date-Sensitive Processing |

## List Items

| id | File | Page | References |
|---|---|---|---|
| UNPACK | RACFICE VERSION4 | 5 | 1<br>5 |
| UPDMEM | RACFICE VERSION4 | 5 | 2<br>11 |
| UPDJOBC | RACFICE VERSION4 | 5 | 3 |
| OUTFIL | RACFICE VERSION4 | 19 | 2<br>20, 20 |

## Footnotes

| id | File | Page | References |
|---|---|---|---|
| UNIX | RACFICE VERSION4 | 14 | 7<br>13 |

## Revisions

| id | File | Page | References |
|---|---|---|---|
| V2 | RACFICE VERSION4 | i | ii, ii, 2, 2, 2, 2, 2, 2, 3, 5, 5, 5, 7, 7, 11, 11, 11, 11, 12, 13, 14, 15, 16, 17, 24, 25 |
| V3 | RACFICE VERSION4 | i | ii, ii, 2, 2, 2, 2, 2, 3, 3 |
| V4 | RACFICE VERSION4 | i | ii, ii, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 7, 7, 11, 11, 11, 11, 11, 11, 13, 13, 13, 13, 13, 13, 13, 14, 17, 18, 18, 20, 20, 23 |

## Processing Options

Runtime values:

| | |
|---|---|
| Document fileid | RACFICE VERSION4 |
| Document type | USERDOC |
| Document style | DEFAULT |
| Profile | EDFPRF40 |
| Service Level | 0032 |
| SCRIPT/VS Release | 4.0.0 |
| Date | 99.01.25 |
| Time | 23:01:40 |
| Device | PSA |
| Number of Passes | 2 |
| Index | YES |
| SYSVAR A | YES |
| SYSVAR B | MIN2 |
| SYSVAR D | NO |
| SYSVAR G | INLINE |
| SYSVAR H | N |
| SYSVAR N | NONE |
| SYSVAR S | 1 |
| SYSVAR T | RIGHT |
| SYSVAR X | YES |

Formatting values used:

| | |
|---|---|
| Annotation | YES |
| Cross reference listing | YES |
| Cross reference head prefix only | NO |
| Dialog | LABEL |

| | |
|---|---|
| Duplex | NO |
| DVCF conditions file | (none) |
| DVCF value 1 | (none) |
| DVCF value 2 | (none) |
| DVCF value 3 | (none) |
| DVCF value 4 | (none) |
| DVCF value 5 | (none) |
| DVCF value 6 | (none) |
| DVCF value 7 | (none) |
| DVCF value 8 | (none) |
| DVCF value 9 | (none) |
| Explode | NO |
| Figure list on new page | YES |
| Figure/table number separation | NO |
| Folio-by-chapter | NO |
| Head 0 body text | (none) |
| Head 1 body text | (none) |
| Head 1 appendix text | Appendix |
| Hyphenation | YES |
| Justification | NO |
| Language | ENGL |
| Keyboard | 395 |
| Layout | 1 |
| Leader dots | YES |
| Master index | (none) |
| Partial TOC (maximum level) | 4 |
| Partial TOC (new page after) | INLINE |
| Print example id's | NO |
| Print cross reference page numbers | YES |
| Process value | (none) |
| Punctuation move characters | ., |
| Read cross-reference file | (none) |
| Running heading/footing rule | NONE |
| Show index entries | NO |
| Table of Contents (maximum level) | (none) |
| Table list on new page | YES |
| Title page (draft) alignment | RIGHT |
| Write cross-reference file | (none) |