

# **E-SRF**

**EKC Security  
Reporting Facility**

**Release 2.1**

**Grouping Facility**



E-SRF V2R1 – **GENERAL AVAILABILITY**, Revised May 19, 2005  
EKC Inc.

The Resource Grouping Facility™ is a proprietary product developed and maintained by

EKC Inc.  
10400 West Higgins Road  
Rosemont, Illinois 60018  
U.S.A.

(847) 296-8010

Technical Support:  
(847) 296-8035

EKC, Inc. provides only software program products which fully comply with, and maintain MVS integrity.

The vendor hereby warrants that:

- 1) E-SRF™ ("Software") performs only those functions which are described in the published specifications;
- 2) there are no methods for gaining access to the Software or other computer resources or data of Licensee (such as a master access key, ID, password, or trap door) other than set forth in the published specifications;
- 3) the Software does not introduce any MVS integrity exposures. The program code, with the exception of one utility, runs totally in non-authorized, problem state. The one utility, EKCRXCAT, requires APF-authorization to read the MVS System Catalogs.
- 4) the software shall be year 2000 compliant, and shall function correctly in the next century according to published specifications as long as regular software maintenance is applied.

Copyright © EKC Inc. USA 1996, 1997, 1998, 1999  
All Rights Reserved

Reproduction of this manual without written permission of EKC Inc. is strictly prohibited.

**Version 2, Release 1 May 2005**

All product names referenced herein are trademarks of their respective companies.

*Printed in USA*

# Contents

<b>RESOURCE GROUPING FACILITY.....</b>	<b>1</b>
INTRODUCTION .....	1
PURPOSE.....	1
RESOURCE GROUPING IN E-SRF .....	2
<b>SETTING UP THE GROUPING RULES.....</b>	<b>3</b>
LOCATION AND FORMAT OF GROUPING RULE DATA SETS .....	3
RULE SYNTAX.....	4
<i>Rule Set Format</i> .....	4
<i>Control Statement</i> .....	4
<i>Rule Entries</i> .....	7
<i>Rule Comments</i> .....	9
<i>Rule Continuation</i> .....	9
PATTERN MASKING.....	10
<i>Use of the asterisk</i> .....	10
<i>Use of the dash</i> .....	10
USE OF THE NEXT: FACILITY TO CONTINUE RESOURCE RULE SETS .....	12
MASKING EXAMPLES .....	13
<b>GROUPING DATA SETS .....</b>	<b>14</b>
EXAMPLES .....	14
<b>GROUPING RESOURCES .....</b>	<b>15</b>
EXAMPLES .....	15
<b>GROUPING SOURCES .....</b>	<b>16</b>
EXAMPLES .....	16
<b>GROUPING USERS .....</b>	<b>17</b>
EXAMPLES (ACF2) .....	17
EXAMPLES (RACF) .....	17
<b>PROCESSING FACILITIES .....</b>	<b>19</b>
OVERVIEW .....	19
TSO COMMANDS & BATCH PROGRAMS.....	20
<i><b>SRFCOMP</b></i> - Compile a set of Grouping Rules from TSO .....	21
<i><b>SRFDCMP</b></i> - Decompile a set of Grouping Rules from TSO .....	22
<i><b>Test Facility</b></i> - Interactively testing a single rule set or a full table of rules .....	23
<i><b>SRETEST</b></i> - Test a set of Grouping Rules from TSO .....	24
<i><b>SREBCMP</b></i> Compile a Partitioned or Sequential Data set of rule sets in batch .....	25
<i><b>EKCRLGRP</b></i> - Produce a cross-reference listing of groups and masks in batch .....	26
<b>GROUPING RULE PROCESSING FOR E-SRF REPORTS.....</b>	<b>29</b>
INCLUDE/EXCLUDE PROCESSING.....	30
ACCESS ANALYSIS REPORTS.....	30
<b>APPENDIX A - SCENARIO .....</b>	<b>31</b>
SCENARIO SETUP.....	31
GROUP LIKE RESOURCES .....	32
ESTABLISH GROUPING RULES .....	33
<i>Helpful Hints</i> .....	35
COMPILE GROUPING RULES .....	36

*This Page Intentionally Left Blank*

# E-SRF Publications

The following is a list of publications supplied with E-SRF:

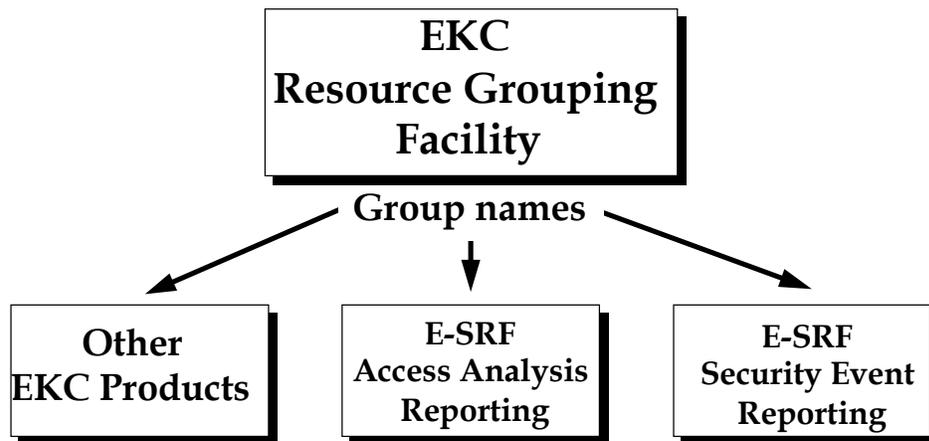
<b>Name</b>	<b>Contents</b>
<i>Installation Guide</i>	E-SRF installation including: installation and maintenance steps, startup and shutdown considerations, and backup and recovery procedures.
<i>Change Summary Guide</i>	Contains all new features and system function changes.
<i>General Overview</i>	An overview of E-SRF and its components.
<i>User Guide</i>	A “How To” guide for user’s of E-SRF.
<i>Access Analysis Reports Guide for ACF2</i>	Brief overview of Access Analysis reports, explanation of the four DataOwner and LogonidOwner reports, command syntax, utilities necessary for creating input to reports, and JCL.
<i>Event Reporting Facility - Command Reference</i>	Explains the Event Reporting Facility command processor, command syntax, and JCL.
<i>Event Reporting Facility - Data Dictionary Reference</i>	Explains the structure of the E-SRF Masterfile.
<i>Event Reporting Facility - Report Overlays Guide</i>	An overview of the report overlays provided with the Event Reporting Facility.
<i>Event Reporting Facility - Messages Reference</i>	Lists Event Reporting Facility messages and codes.
<i>Resource Grouping Facility Guide</i>	Brief overview of the Resource Grouping Facility, its relationship to E-SRF, language command syntax, TSO commands and JCL.

*This Page Intentionally Left Blank*

# Resource Grouping Facility

## *Introduction*

The Resource Grouping Facility is an EKC utility, independent of a particular product and, therefore, may be used by any EKC product. It is designed to associate data sets and resources with a group name.



## *Purpose*

Data sets and resources seldom have common names across the entire environment. Even if your organization follows naming conventions, the number of acquisitions and mergers taking place these days makes it difficult to keep up with those naming conventions as the acquired company's information assets are integrated into your data center. Yet, there are instances where it would be helpful to group information. If this grouping cannot be accomplished using those naming conventions, the EKC Resource Grouping Facility can help.

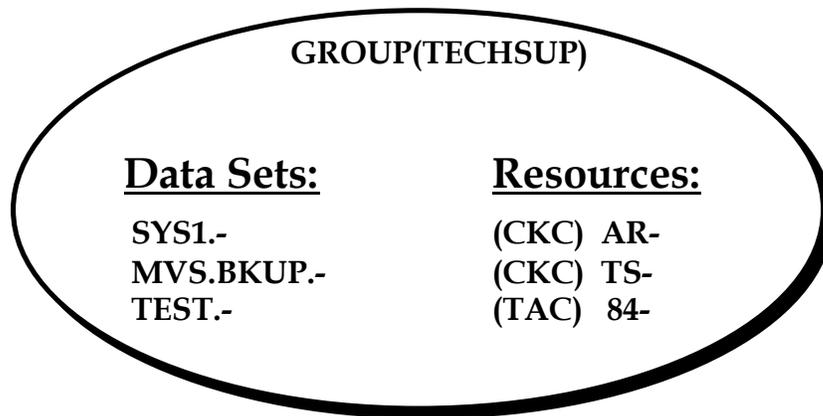
The Resource Grouping Facility is used to identify which data sets and resources belong to which group. The data set or resource names do not need to be a standard naming convention to be included in the same group.

The group definitions in the Resource Grouping Facility are called Grouping Rules. Each rule represents a set of data sets using High Level Indices, a set of resources by class or type, a set of sources by VTAM LUs, or a set of users by specific user identification information. Many entries in the Resource Grouping Rules may be masks. Pattern masking is described later in this document.

## **Resource Grouping in E-SRF**

One EKC product using the Resource Grouping Facility is E-SRF, the Security Reporting Facility.

E-SRF reports provide the owner (or manager) of a set of data sets or resources information about the security-related events that have occurred. E-SRF needs a mechanism to define which data sets and resources belong to each group (and which groups are owned by each owner). Data sets and resources may be combined into the same groups.



# Setting Up the Grouping Rules

The installation of the Resource Grouping Facility has been incorporated into the E-SRF Installation Procedures outlined in the *E-SRF Installation Guide*.

## **Location and Format of Grouping Rule Data Sets**

Source for the Grouping Rule Sets reside in a Variable Blocked (VB) Partitioned Data Set (PDS). This allows for the easy creation and editing of the rule sets via available utilities such as TSO-ISPF or TSO-Edit. Each rule set is stored as its own member in this PDS. Only one rule set may be part of a member. Any member name may be chosen, although we recommend it be similar to the actual rule content to avoid confusion. The Resource Grouping Compiler compiles the source for the Rule Sets into Rule Object Records. Rule Object Records are stored in a sequential Variable Blocked data set.

The specifications for the Grouping Rules Partitioned Data Set and Rule Object Sequential File are:

### **Source PDS:**

RECFM=VB,LRECL=255,BLKSIZE=xxxx (4096 is recommended, but anything will work)

### **Object Sequential File:**

RECFM=VBS,LRECL=32752,BLKSIZE=xxxx (4096 is recommended, but anything will work)

This LRECL is so large that the data set must be allocated with a batch job.

```
//ALLOC EXEC PGM=IEFBR14
//DD1 DD DSN=esrf.objfile, DISP=(NEW,CATLG),UNIT=SYSDA
// DCB=(LRECL=32752,BLKSIZE=4096,RECFM=VBS)
```

## Rule Syntax

Rule Sets are composed of a series of rule lines that describe the grouping rules for a series of either data sets (based on high-level qualifier) or resources (such as all the transactions for a particular CICS Region).

## Rule Set Format

The rule set consists of one control statement and several lines that supply a masking pattern and a group name to which all data sets or resources matching that pattern should be assigned.

Examples of Grouping rule sets:

```
$INDEX(SYS1) DEFGROUP(SYSALL) OR $CLASS(CKC) DEFGROUP(FINANCE)
BROADCAST GROUP(MESSAGES)  AP- GROUP(ACCTSPAY)
MAN*  GROUP(AUDIT)          AR- GROUP(ACCTSREC)
PARMLIB GROUP(SYSTEMS)      PA- GROUP(PAYROLL)
PROCLIB GROUP(SYSTEMS)
```

The sections that follow provide details on writing grouping rules.

## Control Statement

Only one control statement is allowed in a rule set. A statement is continued by placing a dash (-) as the last character of the input line. The total length of all input lines together, excluding trailing blanks, is limited to 512 characters, however, the maximum length of any one operand is 240 characters.

**Syntax:**    \$INDEX(high-level-index-mask-list) **OR**  
              \$CLASS(resource-type-mask-list)  
              DEFGROUP(default-groupname)  
              [PREFIX(data set high-level-indices)]NOPREFIX]  
              [SYSID( \_|system-identifier-mask-list)]  
              [NAME(reference-name)]

\$INDEX or \$CLASS is required. DEFGROUP is required. The rest of the parameters are optional.

## Operands:

### **\$INDEX**

Used for data set rule sets and specifies the high level index, the high level index mask, or the list of high level index masks to which this data set rule set applies. The Resource Grouping implementation limits the size of each entry to eight characters. *IBM data set implementation has always limited this to eight characters.*

Examples of \$INDEX keyword specifications are:

```
$INDEX(COR,CTG,CPR)
$INDEX(SYS*)
```

The \$INDEX( ) control statement must contain the full value for the high level indices you are selecting. For example, \$INDEX(ABC) refers only to one high level index that is exactly ABC. \$INDEX(ABC\*\*\*) is a mask for all data set high level indices from 4 to 6 characters that begin with ABC. The \$INDEX and \$CLASS keywords are mutually exclusive – only one of them can be used. \$INDEX relates to datasets while \$CLASS relates to other resources such as transactions, etc.

### **\$CLASS**

Used for resource rule sets and specifies the class name, the class name mask, or the list of class name masks to which this resource rule set applies. The Resource Grouping implementation limits the size of each entry to eight characters. *IBM SAF implementation limits the class name to eight characters.* The \$INDEX and \$CLASS keywords are mutually exclusive – only one of them can be used. \$INDEX relates to datasets while \$CLASS relates to other resources such as transactions, etc.

Examples of \$CLASS keyword specifications are:

```
$CLASS(OTRAN)
$CLASS(TCICSTRN)
$CLASS(CK*)
```

### **DEFGROUP**

**(this keyword is required)** Used to specify the default group (GROUP keyword in each Rule Line) to which a rule entry should apply. This operand is for ease of input of the rule entries and to control which group should be assigned in the case a particular data set name or resource name does not match any pattern specified in the rule set. The group name may be up to sixteen characters in length. Note that E-SRF supports the use of masking in its selection and reporting based upon the Group names. Thus the group names should be structured in a manner that will take advantage of this feature. The specified Groupname may not contain an asterisk, nor end in a dash.

**PREFIX**

**Data Set Rules:** this specification overrides the value of the \$INDEX keyword and specifies the high level indices to be prefixed to all data set masks unless they are enclosed in quotes. If only one index is specified, the PREFIX defaults to that index or mask. If more than one index mask is specified in the \$INDEX keyword, the PREFIX keyword will be defaulted to \*\*\*\*\*, as in:

PREFIX(\*\*\*\*\*)

This will allow rules to be specified based upon the second level index without having to place them all in quotation marks.

**Resource Rules:** the PREFIX keyword also specifies the high level indices to be prefixed to resource rules, however, there is no default value assigned. The default value for resource rules is NOPREFIX.

**Both Data Set and Resource Rules:** the prefix value may be up to 40 characters in length and, when it is prefixed to the data set or resource mask, a period is appended to the end of it, before the mask.

**NOPREFIX**

Used for data set rules, this specifies that no prefix is to be added to data set or resource masks.

**SYSID**

Used to specify the system identifier, the system identifier mask, or the list of system identifier masks to which this rule set applies. Default is all system identifiers.

**NAME**

This is the name that the rule set is to be referenced by and stored under. Defaults to the first \$INDEX value in the case of data set rules and to the first \$CLASS value in the case of resource rules. When a rule set is decompiled into a Partitioned Data set, this name is used as the partitioned data set member name. In this case characters used in the masking that are invalid (such as \* or -) will be changed so the member name is valid. The name may be up to eight characters in length.

## Rule Entries

Only one logical input line is allowed for each rule entry. However, input lines may be continued by placing a dash (-) as the last character of an input line.

**Note:** A rule set does not necessarily have to contain any rule set entries. Since the resource class or the high level index of a data set name is defined along with a default group, if all of the event entries for that class or index are to be grouped together, the only statement required is the \$INDEX or \$CLASS control statement.

The following example shows a rule set with no entries. All SYS1 data sets would be grouped into SYSTEMS because of the default group specification.

```
$INDEX(SYS1)  DEFGROUP(SYSTEMS)
```

**Syntax:** data set-mask|resource-mask  
[SYSID(-|system-identifier-mask)]  
[CLASS(-|resource-type-mask)]  
[VOL(-|volume-mask)]  
[RECTYPE(ALL|VIOLATION|LOG|SPECIAL)]  
[INCLUDE|EXCLUDE]  
[GROUP(default-groupname|groupname)]

### Operands:

**dataset-mask** (required for data set rule sets) specifies the name of the data set or the data set mask to which this rule entry applies. If the data set name or data set mask contains blanks, it must be enclosed in quotation marks. Normally the value of the PREFIX field will be appended in front of the data set-mask specified as a high level index, but if the data set name or data set mask is enclosed in quotation marks, the PREFIX value will not be appended to the data set name or mask. The data set-mask, including the attached PREFIX value, is limited to a maximum of 256 characters.

**resource-mask** (required for resource rule sets) specifies the name of the resource or the resource mask to which this rule entry applies. If the resource name or the resource mask contains blanks, it must be enclosed in quotation marks. The resource mask, including the attached PREFIX value, is limited to a maximum of 256 characters.

**SYSID** specifies the system identifier to which this rule entry applies. Note that this selection of a system identifier must be within the subset of those selected for the rule set as a whole as defined by the control statement **SYSID** parameter. If this parameter is omitted, the rule entry will apply to all system identifiers included in the **SYSID** parameter of the control statement.

**CLASS** **(resource rules only)** Specifies the resource class to which this rule entry applies. Note that this selection of the resource class must be within the subset of those selected for the rule set as a whole as defined by the control statement **CLASS** parameter. If this parameter is omitted, the rule entry will apply to all resource classes included in the **CLASS** parameter of the control statement.

**VOLUME** **(data set rules only)** Specifies the volume mask used on which the reported data set resides. Default is any volume.

**RECTYPE** specifies the specific event record type to which this rule entry applies. Different rule entries may apply to different record types. One installation may wish for particular system identifiers to exclude security events that are loggings and keep those referring to violations or special loggings. If this parameter is omitted, the rule entry will apply to all record types. Multiple entries may be made separated by commas.

**ALL** All event records will be selected.

**VIOLATION** Only event records associated with a data set or resource violation will be selected.

**LOG** Only event records associated with allow-but-log requests will be selected.

**SPECIAL** Only event records associated with special privilege logging events will be selected.

**INCLUDE|EXCLUDE** specifies whether the data set or resource event described by this rule entry should be included or excluded as EKC products, such as E-SRF, process the Resident Security System journal records.

## **GROUP**

specifies the group name to which the data set or resource event described by this rule entry belongs. This is an optional parameter and if it is omitted, the default groupname specified in the control statement **DEFGROUP** parameter will be used. The group name may be up to sixteen characters in length. Note that the Resource Grouping Facility supports the use of masking in its selection and reporting based upon the Group names. Thus the group names should be structured in a manner that will take advantage of this feature. The specified Groupname may not contain an asterisk, nor end in a dash.

## **Rule Comments**

Rule line comments can be used following a semi-colon in the rule line and are kept with the rule set. Rule line comments are optionally displayed on product logs, such as the E-SRF Reporting Facility database update log, for records that are EXCLUDEd. For more information on EXCLUDEd records, see the section, “*INCLUDE/EXCLUDE Processing*”, later in this document.

Comment lines with a semi-colon in column 1 (one) (those not associated with a specific rule line) are kept as a “comment-block” with the rule set.

Upon decompilation, the comments associated with each rule line are appended to the specific rule line. Note that rule lines may be re-ordered since they will be decompiled into the order that they will be searched via the Resource Grouping Rule Interpreter.

The comment lines in the comment-block are reproduced by the compiler as the first lines in the decompiled output.

For more information about rule compile/decompile processing, see the section, “*Processing Facilities*”, later in this document.

## **Rule Continuation**

Continuation is indicated by a dash “-” as the last character of a rule line or control entry in a rule set. Note that dash, “-”, is also used as a pattern masking control character so care should be taken to place some keyword, such as GROUP, after its use in this manner, so it can be easily distinguished.

## **Pattern Masking**

The Resource Grouping Facility supports pattern masking. This allows for a single definition to refer to multiple resource or data set names. The key characters used to specify these masks are the asterisk (“\*”) and the dash (“-”).

Resource and data set names are made up of segments. These segments are separated by periods. For data set names, the segment length may be up to eight characters and the entire data set name itself is limited to 256 characters, including periods. For resource names, the segment length may be up to 256 characters, but, regardless of how many segments, the total resource name may only be 256 characters. Resource Grouping pattern masking works identically for both data set and resource names.

For the purposes of this description, the actual data set or resource name is referred to as the “target” name and the pattern mask is referred to as the “mask.” The target segment is a portion of the data set or resource name and the mask segment is part of the pattern mask.

### **Use of the asterisk**

The asterisk is used in a mask to specify that any character, but only a single character or blank, in the same position of the target segment is allowed and should be treated as a match. Thus:

ab*	matches	abc
	and	ab
	but does not match	abcd
	or	a

### **Use of the dash**

The dash actually has three possible uses. They are:

Use of the dash imbedded within a mask segment. In this case, the dash is treated just like any other literal character and will match a target segment only if the dash also appears in the same position. In this case it is not being used as a masking character at all. For example:

a-b	only matches	a-b
-----	--------------	-----

Use of the dash as the last character of a mask segment. In this case, the dash is used to indicate that the entire segment should be padded out to its maximum length with asterisks. For example:

ab-	matches	ab
	and	abcdef
	and	abcdefghijklmnopqrstuvwxy*

but it does not match a

Remember, there is a maximum segment length of eight characters for data set masks, but no such limit is imposed for resource masks.

Use of the dash as the only character of a mask segment. In this case, the dash is used to specify that any number of target segments may appear in its place. This is often used to use a mask for a data set name to group all those data sets beginning with a high level index or two as in:

PROD.ACCTG.-	matches	PROD.ACCTG.VER1.LOAD
	and	PROD.ACCTG.DATA
	and	PROD.ACCTG

or to combine all data sets of a certain type, as specified by convention as being the last segment of a data set name:

PROD.ACCTG.-.LIB	matches	PROD.ACCTG.SOURCE.LIB
	and	PROD.ACCTG.MACROS.LIB
	and	PROD.ACCTG.LIB

## **Use of the NEXT: Facility to continue Resource Rule Sets**

The NEXT: Facility allows the splitting or continuation of resource rule sets. An example of the need for this might be the USER Class which defines the grouping parameters for each user (See Page 17). The E-SRF Grouping Rule sets are limited to about 700 rule lines.

In either the DEFGROUP or the GROUP keywords, the construct NEXT:newclass<sup>1</sup> can be specified. If this Group is chosen, then the grouping resource interpretation will begin anew with the specified class. An example of this would be:

```
$CLASS(USER) DEFGROUP(NEXT:USER1); used as a continuation
-.UID1 GROUP(ABC)
-.UID2 GROUP(ABD)
...
-.UID700 GROUP(XYZ)
$CLASS(USER1) DEFGROUP(UNKNOWN)
-.UID701 GROUP(QRT)
etc.
```

Another option is to split the Grouping Rules by some major criteria, as in:

```
$CLASS(USER) DEFGROUP(UNKNOWN.USER)
-.C- GROUP(NEXT:USERCHI) ; CHICAGO USER
-.N- GROUP(NEXT:USERNYC); NEW YORK CITY USER
...
$CLASS(USERCHI) DEFGROUP(UNK.CHIUSER)
...
$CLASS(USERNYC) DEFGROUP(UNK.NYCUSER)
...
```

---

<sup>1</sup> Note: There is no space between "NEXT:" and the class of the continued grouping rule set.

## Masking Examples

The following are examples of grouping rule sets. After each description is the rule set that implements.

1. Group together all SYS1 data sets in the group SYSTEM and exclude normal logging updates to the new system residence volumes (those beginning with the text NEWR).

```
$INDEX(sys1) DEFGROUP(system)
- VOL(newr-) RECTYPE(log) EXCLUDE; development sysres volumes
```

2. Split the accounting data sets into two different groups for receivables and payables.

```
$INDEX(acct) DEFGROUP(accounts)
pay-- GROUP(acctspay)
rcv-- GROUP(acctsrcv)
```

3. Split the CICS production transactions into different groups.

```
$CLASS(ckc) DEFGROUP(cicsprod)
ap- GROUP(acctspay) ; accounts payable transactions
ar- GROUP(acctsrcv) ; accounts receivable transactions
acfm GROUP(security) ; acf2 security command
pay- GROUP(payroll) ; payroll transactions
```

4. Exclude the CICS test region transaction loggings. Put the rest in a group called CICSTEST.

```
$CLASS(ckt) DEFGROUP(cicstest)
- RECTYPE(log) EXCLUDE; test region transactions
```

5. Define the DFSMS functions into one Storage Administration Group

```
$CLASS(fac) DEFGROUP(facilities)
stgadmin GROUP(stgadmin)
```

6. Select groupname by second level index for a series of high-level indices:

```
$INDEX(abc,qrt,yr*) DEFGROUP(abc)
usa.- GROUP(grp.usa)
can.- GROUP(grp.canada)
mex.- GROUP(grp.mexico)
```

## Grouping Data Sets

Data set grouping rules are written by high level qualifier. Each rule will use a single high level index, a masked high level index, or a series of indices or masked indices. You determine how specific the rule sets must be based on the groupings you want to obtain.

- All data sets of the same high level index are put in the same Grouping Rule.
- Each individual data set will only be put in one group. EKC products look for the most specific listing. So you can have a general, masked listing such as DATA.- in one group, but put a special data set DATA.MONTHLY in a different group.
- While each data set can only belong in one group, a group can contain many data sets and resources. You can use the same group name any number of times in Grouping Rules.

### Examples

```
$INDEX(SYS1,SYS2) DEFGROUP(UNKNOWN)
  ACF2.-      GROUP(Security)
  BROADCAST   GROUP(Telecomm)
  LINKLIB     GROUP(Systems)
  MAN*        GROUP(Audit)
  PARMLIB     GROUP(Systems)
  PROCLIB     GROUP(Systems)
  UADS        GROUP(Security)
```

This rule shows the different system data sets that begin with a high level index of SYS1 or SYS2. The default group of UNKNOWN is one approach to identifying which data sets are not defined yet. If you are grouping all data sets, the UNKNOWN designation will indicate to you that a data set is not defined in your grouping rules, because it was assigned to the default group. This example shows both the '\*' and '-' as masking characters.

This example shows five different groups: AUDIT, SECURITY, SYSTEMS, TELECOMM and UNNKNOWN. It is important to know all group names you have used so that they can be defined to the E-SRF Masterfile. You cannot automatically distribute E-SRF Event Reports without the groups defined.

```
$INDEX(ACCT*) DEFGROUP(Finance)
  DATA.-     GROUP(Acctrec)
  MASTER      GROUP(Audit)
  MONTHLY.DATA GROUP(Genlegr)
  PAY.MASTER.DATA GROUP(Payroll)
```

This rule example applies to all data sets that begin with a five-character high level index of ACCT and one character.

This grouping rule uses a default of a more general group: FINANCE. Notice that the group AUDIT is used in both examples. You can use the same group any number of times to create the group definitions you want.

## Grouping Resources

Resources are different from data sets because there are usually no qualifiers in the resource name. In fact, there is no convention for naming resources. CICS transactions are generally four alpha-numeric characters, DB2 tables can be quite long and may be divided into names, separated by periods. Because the names can vary greatly, resource grouping rules do not use the name in a \$INDEX control statement. The \$CLASS control statement is used. It describes the kind of resource you are grouping with the resource class or resource type. The rule lines then contain the resource name or mask and the group designation.

### Examples

```
$CLASS(CKP,CKT) DEFGROUP(UNKNOWN); ACF2 EXAMPLE or
$CLASS(TCICSTRN) DEFGROUP(UNKNOWN); RACF EXAMPLE
  ACFM GROUP(Security)
  A*** GROUP(Accting)
  CEMT GROUP(Systems)
  E-   GROUP(Telecomm)
  MAIL GROUP(Telecomm)
  PR01 GROUP(Payroll)
  PR46 GROUP(Production)
  1234 GROUP(Payroll)
```

This rule shows either the ACF2 Resource Type Codes or the RACF Resource Class used as the \$CLASS value. Then each CICS transaction is listed either specifically or represented by a mask. If your site uses naming conventions for CICS transactions, this process becomes much easier and the rule will be much smaller because of the use of masking. If you are not using naming conventions, you may have to list each CICS transaction individually and the rule could be quite large.

```
$CLASS(PGM) DEFGROUP(SYSTEMS); ACF2 EXAMPLE or
$CLASS(PROGRAM) DEFGROUP(SYSTEMS); RACF EXAMPLE
  HRINFO GROUP(Finance)
  IMASPZAP GROUP(Systems)
  PAYROLL GROUP(Payroll)
```

This rule example shows the ACF2 Type Code or the RACF Resource Class for programs. Then each program that is to be grouped is listed.

This process continues for each type of resource you want to group. You do not have to group all resources or all data sets before these grouping rules can be used. Only group those things you want to distribute reports for or those things you want to display the group for.

**NOTE:** To find out all the CICS transactions you have in your environment, ask your systems programmer for a listing of all active transactions in your CICS regions.

## Grouping Sources

Sources are grouped by the VTAM LU name associated with a physical source. Because of the explosion of dial-in capabilities and LU pooling, these grouping rules will not be useful to most users. If you want to create Grouping rules for sources, there are several things to think about:

- The entire LU pool should be assigned to the same group.
- You may want to group sources that are used for dial-in, but not those within your facilities.

**NOTE:** This is a special type of grouping rule used only by the Event Reporting of E-SRF. A special command must be issued to E-SRF to begin using these rules: `SET SOURCE(LUS)`. See the *E-SRF Command Reference* for more information.

### Examples

```
$CLASS(LUS) DEFGROUP(UNKNOWN)
    LU42-      GROUP(Sales)
    LU426129   GROUP(Executive)
    LU6714-    GROUP(Accting)
    LU671495   GROUP(Personnel)
    LU99-      GROUP(Dial-in)
```

The \$CLASS value of the above rule must match the SET command issued to the E-SRF Masterfile (LUS). In this example, some specific LUs are identified because they are used for a particularly sensitive area, such as EXECUTIVE offices or PERSONNEL. Pools of LUs are identified for specific areas, and the dial-in ports are grouped in DIAL-IN.

## Grouping Users

This is the most difficult Grouping Rule structure to describe because all RSS (resident security system) structures are different, therefore, user identification information will be different as well. A \$CLASS control statement is followed by rule lines containing a user identification information string. This string will be different for every RSS. For example, in ACF2 it will be the Imageid followed by the 24-character UID String. In RACF, it will be the Imageid followed by a combination of owner id, default connect group, and userid. See the *E-SRF Data Dictionary Reference* for more information.

**NOTE:** This is a special type of grouping rule used only by the Event Reporting of E-SRF. A special command must be issued to E-SRF to begin using these rules: SET USERID(*USERS*). See the *E-SRF Command Reference* for more information.

For ACF2, the E-SRF Event Reporting concatenates the Imageid before the UID string so that the same UID may be grouped into different groups depending on the Imageid. Thus, the format of the resource name presented to the grouping system is: Imageid.UID. Also, for ACF2 users, ACF2 defaults to expanding the UID string with asterisks. The E-SRF Grouping Rules do not, so, if you want to just specify the first part of a UID string, then you must specify the rule line as: -.CGXYZ- .

For RACF, the E-SRF Event Reporting concatenates the Imageid before the User's Default Group before the Userid, so the format is: Imageid.OWNERID||DFT-GROUP||USERID. Note that the Ownerid, the Default Group, and the Userid are all padded out to eight characters with blanks.

If there is only one RSS image in use, that is, the same UID string is used on all processors, the Imageid can be ignored. To do this, specify PREFIX(\*-) in the control cards.

### Examples (ACF2)

```
$CLASS (USERS)  DEFGROUP (UNKNOWN)
    CHICAGO.ACHPAYCLK-    GROUP (Payroll)
    NEWYORK.BNYSYSPRG-   GROUP (Systems)
    NEWYORK.BNYTELCLK-   GROUP (Telecomm)
```

Or, to default the Imageid to all Imageid's, use:

```
$CLASS (USERS)  DEFGROUP (UNKNOWN)  PREFIX (*-)
    ACHPAYCLK-  GROUP (Payroll)
    BNYSYSPRG-  GROUP (Systems)
    BNYTELCLK-  GROUP (Telecomm)
```

This example shows the ACF2 UID String combinations with their designated groups.

### Examples (RACF)

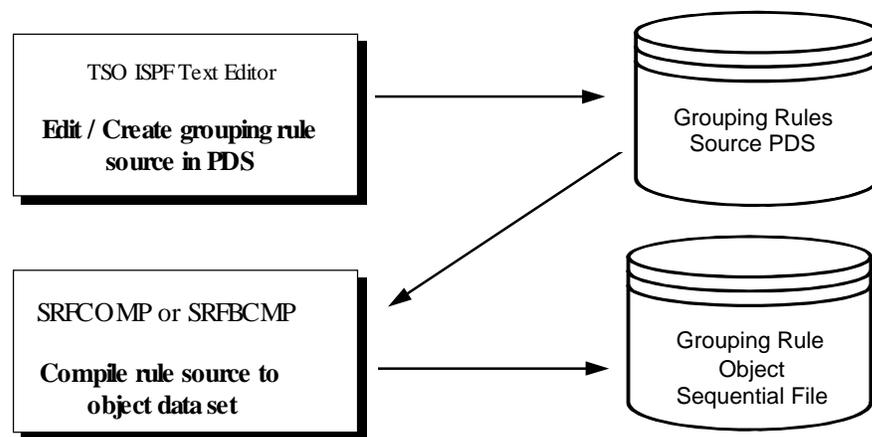
```
$CLASS (USERS)  DEFGROUP (UNKNOWN)
    CHICAGO.PAYADMINPAYROLL-  GROUP (PAYROLL)
    NEWYORK.SYSADMINSYSTEMS-  GROUP (SYSTEMS)
    NEWYORK.TELADMINTELECOMM-  GROUP (TELECOMM)
```

*This page intentionally left blank*

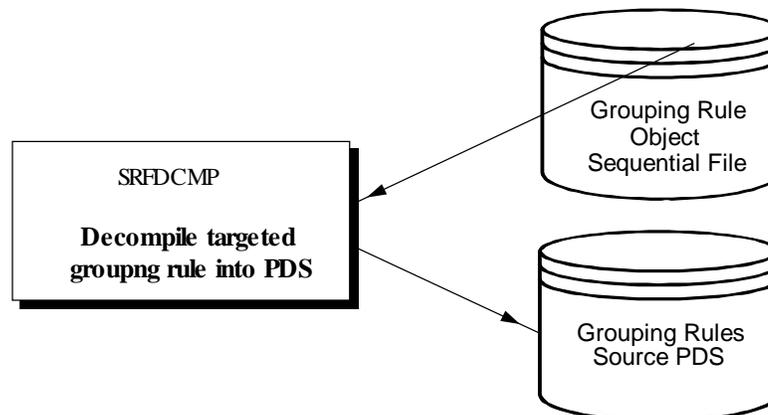
# Processing Facilities

## Overview

Grouping Rule Sets reside in a variable blocked partitioned data set with no sequence numbers. Once you have written the grouping rules you want, the rules must be compiled similar to a programmer compiling a program. The Resource Grouping Facility compiler routines take the partitioned data set members and compile them into Rule Object Records, the format necessary for other products to use grouping rules. The Rule Object Records reside in a sequential Variable Blocked spanned data set.



If you need to recover your source PDS from the Rule Object Records, the process can be reversed using SRFDMP.



## ***TSO Commands & Batch Programs***

Three TSO Commands and two Batch Programs as well as a test facility to interactively determine the results of selected input, are available. In addition, a report shows a listing of which data set and resource masks are included in which groups.

These commands and programs are detailed on the following pages:

**SRFCOMP**      **Compile a set of Grouping Rules** - this TSO command takes a Partitioned Data set of Rule Sets and compiles them into Rule Object Records.

**SRFDCMP**      **Decompile a set of Grouping Rules** - this TSO command takes the Rule Object Records and converts them back to a Partitioned Data set of Rule Sets.

**SRFTEST**      **Test a set of Grouping Rules** - this TSO command takes the Rule Object Records and tests the results of the Grouping Facility Interpretation of these rules given a set of input parameters.

**SRFBCMP**      **Compile an entire Partitioned Data set of rule sets** - this batch program takes an entire Partitioned Data set of Rule Sets and compiles them into Rule Object Records. This program will also accept a sequential data set with rule sets and compile them all into Rule Object Records.

**EKCRLGRP**      **Produce a cross-reference listing of groups and masks** - this batch report program takes the Rule Object Records and produces a report by Group as to which rule lines from which rule sets generate entries for that Group.

## **SRFCOMP - Compile a set of Grouping Rules from TSO**

**Function:** This command is used to compile a grouping rules data set into rule object records. This command can be used to compile individual rules, when changes are made, or the entire rules PDS.

**Syntax:** SRFCOMP SOURCE(input-pds) OBJECT(output-dsn)

LIST/NOLIST COMMENT/NOCOMMENT  
TEST/NOTEST

### **Operands:**

#### **SOURCE**

**(this keyword is required)** The input-pds defines a variable blocked partitioned data set (without sequence numbers) where each member contains an input rule set. If a member is specified, only that member will be compiled and the data set specified in the OBJECT parameter will be opened for MOD (an Operating System term indicating data is to be added at the end of the data set), allowing this rule set to be added to the existing rule object records. If no member is specified, all the members in the partitioned data set will be compiled. The data set specified in the OBJECT parameter will be opened as OLD if it exists, and NEW if it does not, which means that the results of the compilation will replace all of the Rule Object Records in the data set.

#### **OBJECT**

The output-dsn defines a sequential variable blocked output data set which the compiled rules in the form of rule object records will be stored. If this operand is omitted, the rules will be compiled but no storage will take place. If the output-dsn does not exist, the SRFCOMP TSO command will allocate it.

#### **LIST/NOLIST**

The Grouping Rule Compiler will list the input rule sets being compiled to the TSO terminal.

#### **COMMENT/NOCOMMENT**

The comments in an input rule set are, by default, stored in the rule object record and are available during interpretation and for decompilation. The NOCOMMENT operand will instruct the Rule Compiler to discard the comments. Once comments are discarded, they cannot be regenerated from a decompile operation.

**TEST/NOTEST**

The SRFDCMP TSO Command will stop after each rule set is compiled and allow for interactive testing. See the section, “*Test Facilities*”, described later in this document.

**SRFDCMP - Decompile a set of Grouping Rules from TSO**

**Function:** Once the grouping rules have been compiled, you may want to decompile them back into the grouping rules PDS to make modifications. This can also be used in cases where you have lost the grouping rules PDS and need to recreate it.

**Syntax:** SRFDCMP OBJECT(input-dsn) SOURCE(output-pds)  
LIST/NOLIST TEST/NOTEST

**Operands:**

**OBJECT**

**(this keyword is required)** Defines the data set containing the Rule Object Records to be decompiled. Decomilation will re-create the source input from the Rule Object format and this command will list the source on the TSO terminal and optionally, store it into a Partitioned Data Set. If a single rule set is specified within parenthesis, only that rule object record will be decompiled.

**SOURCE**

Defines the output partitioned data set into which the Rule Sets will be stored. If the data set does not exist, it will be created. If it is omitted, the SRFDCMP will just list the source statements and optionally test the rule set.

**LIST/NOLIST**

Indicates whether the input statements will be listed or not.

**TEST/NOTEST**

The SRFDCMP TSO Command will stop after each rule set is compiled and allow for interactive testing. See the section, “*Test Facilities*”, described later in this document.

## **Test Facility - Interactively testing a single rule set or a full table of rules**

The test facility can be used to make sure that the results you expect from your grouping rules are in fact the results that would be processed in the Resource Grouping Facility. The test facility will tell you which group a data set or resource would be associated with.

If the interactive test facilities are entered via the **SRFTEST** command, the specified rule set will be located and then used for testing. If the test facilities are entered via the **SRFCOMP** or **SRFDCMP** commands, the specific rule set currently being processed will be interactively tested.

Input is accepted, the rule set is interpreted, and the results are displayed. The process continues interactively until a null input line is entered. A null input line is one without anything specified on it. The input is in the following format:

**DSNAME(test-dsname) SYSID(system-id) VOLUME(volser) VIO/LOG/SPEC**

**or**

**RESOURCE(resource-name) SYSID(system-id) CLASS(class) VIO/LOG/SPEC**

The testing system keeps using the old values for parameters until they are changed allowing the user to interactively change one parameter at a time. If a **DSNAME** is specified, the system assumes that it will be testing data set rules. To change from testing data set rules to resource rules, enter **DSNAME()** which will reset the **DSNAME** parameter and cause the interactive system to begin testing resource rules.

<b>DSNAME</b>	specifies the fully qualified data set name
<b>RESOURCE</b>	specifies the fully qualified resource name
<b>SYSID</b>	specifies the eight character system identifier
<b>VOLUME</b>	specifies the six character volume serial number
<b>CLASS</b>	specifies the eight character resource class
<b>VIOLATION</b>	specifies that the request is for a violation event
<b>LOGGING</b>	specifies that the request is for an allow-but-log event
<b>SPECIAL</b>	specifies that the request is for a “special” logging event in which the authority of the user overrode the Resident Security System rules.

## **SRFTEST - Test a set of Grouping Rules from TSO**

**Function:** The test facility can be used to determine if a grouping rule does, in fact, group the appropriate data sets and resources together.

**Syntax**     SRFTEST OBJECT(input-dsn) TEST/NOTEST

### **Operands:**

**OBJECT**                    **(this keyword is required)** Specifies the input Rule Object Record data set. All rules object records will be read into storage and a table of index masks and sysid masks will be built for the data set rule sets and a table of class masks and sysid masks will be built for the resource rule sets. If a rule object record appears more than once in the data set (based on the Reference Name parameter of the Rule Set) , the latest one will be used.

**TEST/NOTEST**            After the rule sets are read into storage and the tables built, the rules are available for interactive testing. See the section, "*Test Facilities*", described on the previous page in this document.

## **SRFBCMP      Compile a Partitioned or Sequential Data set of rule sets in batch**

**Function:** This facility can be used to compile rule sets into Rule Object Records. Use either the RULEPDS DD card or the RULES DD card. The RULEPDS refers to an entire variable blocked partitioned data set while the RULES refers to a variable blocked sequential data set. All of the members of the Partitioned Data Set will be compiled and the Rule Object output dataset will be completely re-written. When using sequential input, the batch program determines the separation between two rule sets as a \$ in column 1. Therefore, it is recommended that all rule lines are specified beginning in column 2 so that a mask beginning with a \$ will not be interpreted as the beginning of the next rule set.

If a sequential data set is developed for the rule sets, a partitioned data set may be created by decompiling the Rule Object Records into a partitioned data set using the TSO command, SRFDCMP.

The sequential data set input is useful when an installation is developing multiple rule sets programmatically. Both the RULES and RULEPDS data sets will be processed by SRFBCOMP in a single execution.

**NOTE: The Rule Object Record dataset is rewritten in its entirety with each execution of SRFBCMP.**

### **The JCL for SRFBCMP is:**

```
//COMPILE EXEC PGM=SRFBCMP,PARM='parms',REGION=2048K
//SYSPRINT DD SYSOUT=A
//RULEPDS DD DISP=SHR,DSN=source partitioned data set
//RULES DD DISP=SHR,DSN=source sequential data set
//RULEOBJ DD DISP=OLD,DSN=rule object record output data set
```

### **Parameters are:**

#### **LIST/NOLIST**

If **NOLIST** is specified, the compiler output will not be listed unless there was an error, in which case the entire rule set will be listed. If there is no errors, the compiler will only produce a single line that indicates the rule set was compiled successfully. The **NOLIST** parameter also implies a **CONTINUOUS** output request (see below).

#### **COMMENT/NOCOMMENT**

The comments in an input rule set are, by default, stored in the rule object record and are available during interpretation and for decompilation. The **NOCOMMENT** operand will instruct the Rule Compiler to discard the comments.

#### **BREAK/CONTINUOUS**

Each Rule Set will begin a new output page if **BREAK** (the default) is specified. If **CONTINUOUS** is specified, page breaks will not be forced and the output will be continuous.

## **EKRLGRP - Cross Reference Utility for Grouping Rules**

**Function:** This program will produce a cross-reference listing of groups and the masking lines that will make up the group in addition to optionally producing an export data set and a command data set. The rule set name, the mask, and other parameters are listed for each rule line which will cause data sets and resources to be placed in the group. Optionally, this utility will produce a command output dataset with one command for each group defined or an export dataset with one entry per rule line combination.

### **The JCL for EKRLGRP is:**

```
//XREF EXEC PGM=EKRLGRP
//SYSPRINT DD SYSOUT=A
//GRPRULES DD DISP=SHR,DSN=rule object record data set
//CMDOUT DD command output dataset (optional)
//EXPORT DD export output dataset (optional)
```

**Keyword Parameters are (parameters are not required, but the following may be specified):**

### **Selection control keyword:**

**GROUP** - a Group name mask that will limit the output. Only records whose groupnames match the mask will be written to the output datasets or printed in the output.

### **Output formatting keywords:**

**TITLE** - a title of up to 64 characters that will be placed at the top of each page beginning with the second page.

**LINES** - the number of lines per page of output. The default is 55.

**COMMENTS** - the Grouping Rule Comments are to be printed. Comments are always inserted into the EXPORT output. On the printed output, they are placed on the line immediately below the Rule Line Mask.

**FIELDS**(field1, field2,...,fieldn) - the names of the fields and the order of appearance that will be displayed on the output. The **MASK** field must be specified last if it is specified. See the field names below for more information. This keyword selection does not affect the EXPORT output. The default statement is:

**FIELDS(GNAME,RNAME,RTYPE,SYSID,CLASS,RECORDS,MASK)**

**SORTBY**(field1,field2,...,fieldn) - the order of output for both the printed output and the EXPORT output. Records are sorted in ascending order controlled by the fields specified. See the field names below for more information. The default statement is:

**SORTBY(GNAME,RTYPE,SYSID,CLASS,MASK)**

**BREAKAT**(field) - the name of the field that will be used to determine when an additional blank line is to be inserted into the output. See the field names below for more information. Or use **BREAKAT()** to indicate no additional blank lines are to be inserted. The default statement is:

**BREAKAT(GNAME)**

**Field Name Definitions:**

**GNAME** - the Groupname

**RNAME** - the Rule Set Name

**RTYPE** - the Rule Set Type --- **DATASET** or **RESOURCE**

**SYSID** - the SYSID mask

**CLASS** or **INDEX** - these fieldnames are interchangeable and specify either the CLASSname or the High Level Index of the dataset mask.

**RECORDS** - the record types

**VOLUME** - the volume serial number (for dataset grouping rules only)

**MASK** - the dataset or resource mask

**Command Output controls:**

**CMDOUT** - specifies that command output is to be written to the optional CMDOUT DDNAME.  
The format is:

**prefix GROUP(groupname) suffix**

**CMDPFX**(prefix) - defines the prefix string to be placed prior to the GROUP keyword of each line.

**CMDSFY**(suffix) - defines the suffix string to be placed after the GROUP keyword on each line.

**Export Output controls:**

**EXPORT(DIF|RECORD)** - specifies that a data output file is to be written to the optional EXPORT DDNAME.

**DIF** - a comma delimited data interchange format dataset is created which is appropriate for downloading to a PC based application. **DIF** is default.

**RECORD** - a variable record format, fixed column dataset is created which is appropriate for input to a mainframe application program.

**EXPORT output formats:**

**Data Interchange Format (DIF)** - this is a comma delimited output with all of the fields. The field order is:

**GROUP\_NAME**

**RULE\_SET\_NAME**

**RULE\_TYPE**

**SYSID**

**CLASS/INDEX**

**REC\_TYPES**

**VOLSER**

**RULE\_MASK**

**RULE\_COMMENT**

**Record Format:**

<b>COLUMNS</b>	<b>Description</b>
<b>0-15</b>	<b>Group name</b>
<b>17-24</b>	<b>Rule Set Name</b>
<b>26-33</b>	<b>Rule Set Type --- DATASET or RESOURCE</b>
<b>35-42</b>	<b>Sysid mask</b>
<b>44-51</b>	<b>Class or Index</b>
<b>53-60</b>	<b>Record Types</b>
<b>62-69</b>	<b>Volume Serial Number Mask</b>
<b>71-144</b>	<b>Rule line mask</b>
<b>146-210</b>	<b>Rule line comment (NOTE: this field will be blank truncated)</b>

**Output:**

```

EKCRLEGRP-06.07.99      EKCRLE GROUPING FACILITY: GROUPNAME/MASK REPORTING UTILITY
MON, JUNE  7, 1999  11:41                                     PAGE.....1
PROCESSING JOURNAL:

EKCRLEGRP-001 INPUT CONTROL CARDS:

      TITLE(TEST OF REVISED GROUP UTILITY)
      SORTBY(RTYPE,INDEX,MASK)
      FIELDS(GNAME,RNAME,RTYPE,INDEX,RECORDS,MASK)
      BREAKAT(INDEX)
      COMMENTS

EKCRLEGRP-102 GROUPING RULE OBJECT DATASET INFORMATION:

      NAME: BARRY.ESRF.RULEOBJ
      HIGHEST COMPILE DATE STAMP: 07-JUN-99 @ 11:28
      NUMBER OF RULE RECORDS READ: 21
      NUMBER OF DUPLICATES SKIPPED: 1
---
      EKCRLE GROUPING FACILITY: GROUPNAME/MASK REPORTING UTILITY
MON, JUNE  7, 1999  11:41      TEST OF REVISED GROUP UTILITY      PAGE.....2
PROCESSING CLASS/INDEX: BARRY

GROUP NAME          RULE SET  RULE SET  CLASS/   RECORD   DATASET/RESOURCE MASK
                   NAME      TYPE      INDEX    TYPES    COMMENTS

BARRY               BARRY    DATASET  BARRY    ALL      *** DEFAULT GROUPNAME ***
                   BARRY    DATASET  BARRY    ALL      -->BARRY'S DATASETS
                   BARRY    DATASET  BARRY    ALL      BARRY.ESRF-. -
                   BARRY    DATASET  BARRY    ALL      -->ESRF SOURCE/OBJECT, ETC.

ESRF                ESRF     DATASET  ESRF     ALL      *** DEFAULT GROUPNAME ***
                   ESRF     DATASET  ESRF     ALL      -->ESRF DATASETS

SYS1TEST           SYS1TEST DATASET  SYS1     ALL      *** DEFAULT GROUPNAME ***

SYSTEM             SYS*     DATASET  SYS*     ALL      *** DEFAULT GROUPNAME ***

TOMC               TOMC     DATASET  TOMC     ALL      *** DEFAULT GROUPNAME ***
                   TOMC     DATASET  TOMC     ALL      -->TOM'S DATASETS
                   TOMC     DATASET  TOMC     ALL      TOMC.ESRF-. -
                   TOMC     DATASET  TOMC     ALL      -->ESRF SOURCE & OBJECT

OPENMVS           ESRFOMVS RESOURCE ESRFOMVS ALL      *** DEFAULT GROUPNAME ***
BARRY             ESRFOMVS RESOURCE ESRFOMVS ALL      HOME/BARRY/-
                   ESRFOMVS RESOURCE ESRFOMVS ALL      -->BARRY'S STUFF

EB                 ESRFOMVS RESOURCE ESRFOMVS ALL      HOME/EB/-
                   ESRFOMVS RESOURCE ESRFOMVS ALL      -->EB'S STUFF

```

# Grouping Rule Processing for E-SRF Reports

EKC products such as the E-SRF Event Reporting Facility use the grouping rules to determine which security events to process for a report. When you run an E-SRF report, it will initialize the rule sets for processing using the following steps, repeating the process for both data set and resource rule sets:

1. All of the rule sets are read into storage from the Grouping Rules Object Record data set.
2. The number of indices and classes are counted.
3. Storage for the index and class tables is obtained.
4. A class/index table is created specifying the class/index and rule set address, then the information is sorted from most specific to least specific.

\$INDEX Rule Entries	Rule Set Address
ABC	
BACKUP	
SYS1	
SYS*	
TEST	

\$CLASS Rule Entries	Rule Set Address
AB-	
APY	
CKC	
CKT	
TAC	

The resulting table is used to identify to which group each data set or resource belongs. When E-SRF needs the group name, it presents the data set or resource name to the Resource Grouping Facility, which then scans the index table or the class table comparing the masks in the table with the E-SRF data set or resource. If the Resource Grouping Facility finds a match, it uses the address to find the rule set and read the entries to determine the appropriate group name.

If no match is made in the class/index table, the GROUP will be set to either the high level index or the class name. This enables the easy implementation of E-SRF because all data sets with the same high level index will be, by default, grouped together. All resources will be grouped together by type (or CLASS). Thus, until you are ready to separate data sets within the same group into different groups or group together different high level qualifiers and resource classes, no rules need be written.

Rather than using the high level index or the class name as the group, the installation may want to group together the events for which the rules do not specify a particular group, the following default rules can be used:

\$INDEX(-) DEFGROUP(unknown)

\$CLASS(-) DEFGROUP(unknown)

## ***INCLUDE/EXCLUDE Processing***

In the E-SRF usage of the Resource Grouping Facility, records may be excluded from use in E-SRF reports. You may want to exclude records that are not necessary for the particular group. For example, you can create a group with only violation records by excluding the loggings. The Resource Grouping Facility processes any exclude statements that exist.

When, the Resource Grouping Facility reviews the rules, it identifies any INCLUDE or EXCLUDE definition, and a GROUP definition if the identified security event was included.

If the security event is EXCLUDEd, E-SRF does not continue processing the event record. If the event is INCLUDEd, E-SRF places the GROUP name into the event record.

## ***Access Analysis Reports***

Operation during the E-SRF Access Analysis Reports is similar to that processed during the Event Reporting Facility processing, except that the type of event record is not applicable. The Access Analysis Reports always use the Group value specified for a Violation Record in order to determine the Group name.

Grouping of users in the LogonidOwner Reports are done by masking (using masking control characters specified above where applicable) fields in the Logonid records. No grouping rules are necessary.

## Appendix A - Scenario

This section provides examples of establishing grouping rules for data sets and resources.

Company	→	IMA Corporation
Resources	→	Data Sets for Payroll, Human Resource and Accounting applications
	→	CICS and IMS transactions

### Scenario Setup

The following Resource and Data Set examples use ACF2 masking syntax when referencing multiple transactions or data set names with one entry. Masking is a common ACF2 practice, used for convenience when referencing more than one item. The masking characters of “ - “ a dash and “ \* ” an asterisk, are used to denote a wild characters. This dash means that any resource or data set name matching the pattern up to the dash will meet the criteria. For example, 'PAY-' means that anything with **PAY** as the first 3 characters will match this pattern: '**PAY\***' means that anything with **PAY** and one additional character will match this pattern. For more information on masking, see the section on *Pattern Masking* in this guide.

The chart below presents the non-data set resources of IMA Corporation used in the grouping rule example.

IMA Corporation		RESOURCES
Application Name	Transactions	
CICS - Payroll	PAYR, PAYS, PAYT, PAYU, PAYV, PAYX, PAYY, PAYZ	
CICS - Human Resources	all transactions that start with letter <b>H</b> :	H-
IMS - Accounting	all transactions that start with letter <b>A</b> :	A-

The next chart presents the data sets of IMA Corporation used in the grouping rule example.

IMA CORPORATION		DATA SETS
Application Name	Data Set Names	
Payroll	PAY.- PAYROLL.- CHI.PAY.-	LA.PAY.- HQ.SYS.PAY.- PAYROLL.MASTER
Human Resources	HRSYS.- HR.EMP.-	HR.-.BKUP.-
Accounting	ACCT.- AP.-	ACCTPAY.- GLAP.-
Systems	SYS*	

## Group Like Resources

Start the grouping process by determining which data sets and resources belong together as like resources. The goal of grouping is to group like resources together for the purpose of reporting by ownership and by group.

The groups you define should include functionally similar data set and resources that may be used by the same organization department. In our example of IMA Corporation, we have set up five groups. Each established group has a name of sixteen characters or less.

Functional Area	Group Name	Resources	Data Sets
Accounting	ACCTING	All CICS transactions that start with <b>P</b> and <b>H</b> (except those specified below in this table) All IMS transactions that start with <b>A</b>	All data sets with high level index of: ACCT HRSYS
Payroll	PAYROLL	CICS transactions: PAYR, PAYS	All data sets with high level index of: PAY CHI.PAY LA.PAY HQ.SYS.PAY
Accounts Payable	ACCTPAY	IMS transactions: ACCT, ACC2 ACC4, MACP	All data sets with high level index of: AP ACCTPAY GL.AP
HR Select/View Functions	HRVIEW	CICS transactions: PAYU, PAYX PAYY	All data sets with high level index of: PAYROLL
HR Update Functions	HRUPDATE	CICS transactions: PAYT, PAYV PAYZ HR12, H114 HF32, HR60	All data sets with high level index of: HR.EMP HR.-.BKUP Specific data set: 'PAYROLL.MASTER'
Systems	SYSTEM		System data sets that have high level index of: SYS*

## Establish Grouping Rules

The following grouping rules would be established for each high level index or resource class. This approach is a resource to group relationship. Rules are written at the resource level, with the group referenced in the rule itself. There may be one group in a rule or many groups in a rule.

This example also uses ACF2 syntax for resource class, i.e., ITR for IMS transactions; CKC for CICS transactions.

```
$CLASS(CKC) DEFGROUP(FINANCE)
H114 GROUP(HRUPDATE)
HF30 GROUP(HRUPDATE)
HR- GROUP(HRUPDATE)
H- GROUP(ACCTING)
PAYR GROUP(PAYROLL)
PAYS GROUP(PAYROLL)
PAYT GROUP(HRUPDATE)
PAYU GROUP(HRVIEW)
PAYV GROUP(HRUPDATE)
PAYX GROUP(HRVIEW)
PAYY GROUP(HRVIEW)
PAYZ GROUP(HRUPDATE)
P- GROUP(ACCTING)
TIME EXCLUDE; Time remaining -
until end of day. Throw this -
txn away for E-SRF event -
reporting
```

CICS transactions - class of CKC

One ruleset - multiple transactions, multiple groups

DEFGROUP → FINANCE

All transactions that do not match the rule line entries will be assigned this default group.

Individual rule line entries will assign transactions matching the names to designated group names.

TIME transaction has been excluded from grouping rule. This is for E-SRF Masterfile processing Events for TIME will not be recorded in the Masterfile.

Comment in rule line after exclude entry.

```
$CLASS(ITR) DEFGROUP(FINANCE)
ACCT GROUP(ACCTPAY)
ACC2 GROUP(ACCTPAY)
ACC4 GROUP(ACCTPAY)
A- GROUP(ACCTING)
MACP GROUP(ACCTPAY)
```

IMS transactions - class of ITR

One ruleset - multiple transactions, multiple groups

DEFGROUP → FINANCE

All transactions that do not match the rule line entries will be assigned this default group.

Individual rule line entries will assign transactions matching the names to designated group names.

Grouping rules for data sets use the high level index (HLI) or masking of the high level index.

\$INDEX(ACCT,HRSYS) DEFGROUP(ACCTING)

Data Set grouping rule for 2 HLI: ACCT, HRSYS

All data sets matching these HLI will be assigned ACCTING

\$INDEX(CHI) DEFGROUP(FINANCE)  
PAY.- GROUP(PAYROLL)

Data Set grouping rule for 1 HLI: CHI

All data sets matching CHI.PAY.- will be assigned  
PAYROLL group.

All other CHI data sets will be assigned FINANCE group.

\$INDEX(HQ) DEFGROUP(FINANCE)  
SYS.PAY.-GROUP(PAYROLL)

Data Set grouping rule for 1 HLI: HQ

All data sets matching HQ.SYS.PAY.- will be assigned  
PAYROLL group.

All other HQ data sets will be assigned FINANCE group.

\$INDEX(AP, ACCTPAY) DEFGROUP(ACCTPAY)

Data Set grouping rule for 2 HLI: AP, ACCTPAY

All data sets matching the HLIs will be assigned ACCTPAY  
group.

\$INDEX(PAYROLL) DEFGROUP(FINANCE)  
MASTER GROUP(HRUPDATE)  
- GROUP(HRVIEW)

Data Set grouping rule for 1 HLI: PAYROLL

The specific data set 'PAYROLL.MASTER' will be  
assigned HRUPDATE group.

All other PAYROLL data sets will be assigned HRVIEW  
group.

\$INDEX(PAY) DEFGROUP(PAYROLL)

Data Set grouping rule for 1 HLI: PAY

All data sets matching PAY HLI will be assigned PAYROLL  
group.

```
$INDEX(LA) DEFGROUP(FINANCE)
PAY.-      GROUP(PAYROLL)
```

Data Set grouping rule for 1HLI: LA

All data sets matching LA.PAY.- will be assigned PAYROLL group.

All other LA data sets will be assigned FINANCE group.

```
$INDEX(GL) DEFGROUP(FINANCE)
AP.-      GROUP(ACCTPAY)
```

Data Set grouping rule for 1 HLI: GL

All data sets matching GL.AP.- will be assigned ACCTPAY group.

All other GL data sets will be assigned FINANCE group.

```
$INDEX(HR) DEFGROUP(FINANCE)
EMP.-      GROUP(HRUPDATE)
-.BKUP.-   GROUP(HRUPDATE)
```

Data Set grouping rule for 1 HLI: HR

All data sets matching HR.EMP.- will be assigned HRUPDATE group.

All data sets matching HR.-BKUP.- will be assigned HRUPDATE group.

assigned FINANCE  
All other HR data sets will be group.

```
$INDEX(SYS*) DEFGROUP(SYSTEMS)
```

Data Set grouping rule for 1HLI: SYS\*

All data sets matching SYS\* will be assigned SYSTEMS group.

This covers all HLIs matching SYS1 through SYS9 and SYSA through SYSZ.

## Helpful Hints

Some of the grouping rules could have been established several different ways. The DEFGROUP is always there to allow any new or unknown data set or resource automatic assignment. Some of the grouping rules could have used DEFGROUP versus specifying a separate entry. However, you must consider any self-documenting procedures that can be helpful later. You never know when new resources can impact the established rulesets without your knowledge.

Comments can also be established within each ruleset. They will be retained for future reference.

Example of rules that could vary:

```
$INDEX(HR) DEFGROUP(FINANCE)
EMP.- GROUP(HRUPDATE)
-.BKUP.- GROUP(HRUPDATE)
```

```
$INDEX(HR) DEFGROUP(HRDFault)
EMP.- GROUP(HRUPDATE); human resources
-.BKUP.- GROUP(HRUPDATE); backup data set
- GROUP(FINANCE); all other HR dsns
```

This example brings to your attention the masking used on the last entry. It is more obvious that any *other* data sets with a high-level qualifier of HR will be assigned the FINANCE group. Comments have also been used in the second example.

### Compile Grouping Rules

The grouping rules above must be compiled in order for E-SRF, or other EKC products to use them. The SRFCOMP command or SRFCOMP batch program can be used to compile the rules into a sequential data set, known as the Rule Object Record.

```
EDIT ---- ESRF.JCL.LIB(COMPILE) - 01.01 ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==>  CSR
***** ***** TOP OF DATA *****
000001 //DBHCEKCR JOB (T,DGT,DGR3000,4,N-28), 'EKC CONSULT',
000002 //                                CLASS=A,MSGCLASS=L,NOTIFY=DBHC
000003 //*
000004 //*****
000005 //*
000006 //*                                COMPILER GROUPING RULES                                *
000007 //*
000008 //*****
000009 //*
000010 //COMPILE EXEC PGM=SRFCOMP,REGION=0M
000011 //*
000012 //RULEPDS DD DSN=ESRF.RULES.SOURCE,DISP=SHR
000013 //RULEOBJ DD DSN=ESRF.RULES.OBJECT,DISP=OLD
000014 //SYSPRINT DD SYSOUT=*
000015 //
```

When you have compiled the rules, execute EKCRLGRP to review a cross-reference report displaying the compiled rules and their associated groups.