MVS/ESA and RACF Version 1 Release 9 Security Implementation Guide

Document Number GG24-3585-00

August 10, 1995

IBM International Technical Support Center
Poughkeepsie, New York, USA

Take Note!

Before using this information, and the product it supports, there is some general information you should be aware of. This information is in a section called Special Notices, which immediately follows the Table of Contents.

First Edition (March 1991)

This edition applies to MVS/SP Version 3 Release 1.3 and RACF Version 1 Release 9.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM World Trade Corporation,
International Technical Support Center,
Department H52, Building 930,
P.O. Box 950,
Poughkeepsie, NY 12602 USA

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Abstract

This document provides installation information on enhancements included in the MVS/SP Version 3 Release 1.3 and RACF Version 1 Release 9 products, with specific attention to implementing the security enhancements. It contains recomendations and examples on how to use and implement the functions as well as a short description of how they work.

LSYS (328 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help the customer understand the new functions of RACF 1.9 and how they relate to the added support in MVS/SP 3.1.3 and its JES2 and JES3 subsystems. The information in this publication is not intended as the specification of the programming interfaces that are provided by RACF 1.9 for use by customers in writing programs that request or receive its services. See the Publications section of the IBM Programming Announcement for RACF 1.9.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this document is not intended to state or imply that only IBM′s program product may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service. Evaluation and verification of operation in conjunction with other products, except those designated by IBM, is the user′s responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, New York 10577, USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed on an ′As Is′ basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

You can reproduce a page in this document as a transparency, if that page has the copyright notice. The copyright notice must appear on each page being reproduced.

Trademarks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM*	**ES/3090***	**MVS/DFP***	**NetView***
CICS/ESA*	**Hiperbatch***	**MVS/ESA***	**VM***
DFSMS*	**Hiperspace***	**MVS/SP***	**VM/XA***
ESA*	**MVS***	**MVS/XA***	**VTAM***

Preface

In October of 1989, IBM announced several enhancements to large systems security. To accomplish the new level of security, many products were enhanced. This publicatation addresses these enhancements, how they can be installed in a customer environment, and provides some recommendations and guidelines. The information in this publication is divided into 14 chapters, as follows:

- Chapter 1 is an overview of the security enhancements. It discusses how these enhancements can be used to implement security in an MVS/ESA environment.
- Chapter 2 introduces the concept of security classification and SECLABELs.
- Chapter 3 discusses the details of SECLABELs and how they can be used to enhance security in a commercial environment.
- Chapter 4 is a collection of the enhancements that were made for the RACF product.
- Chapter 5 discusses the new role of the System Authorization Facility (SAF).
- Chapter 6 contains information about the security enhancements in a JES environment. It addresses how to implement security in both JES2 and JES3.
- Chapter 7 discusses JES spool security enhancements.
- Chapter 8 contains information about the security in an NJE environment.
- Chapter 9 contains information about the security in RJE and RJP environments.
- Chapter 10 describes the implementation of security for operator consoles and commands.
- Chapter 11 discusses the Print Services Facility (PSF/MVS) security enhancements.
- Chapter 12 contains the new security scheme for SDSF.
- Chapter 13 describes the security enhancements made for MVS/SP BCP, MVS/DFP, ACF/VTAM, and TSO/E.
- Chapter 14 contains information about a B1 secure facility.

This document presents a commercial view of the new security and audit functions for organizations not implementing full B1 security. While the new security functions include those that are required for B1 and others, this document analyzes both categories for their benefits in the formal commercial environment. Installations planning the implementation of B1 level security must follow the strict guidelines provided in *MVS/ESA Planning: B1 Security*.

Related Publications

The following IBM publications provide more detailed information that may assist the security administrator or the system programmer in fully understanding particular subject areas:

MVS/ESA

- *MVS/ESA Conversion Notebook Volume 2*, GC28-1568
- *MVS/ESA Operations: Systems Commands*, GC28-1826
- *MVS/ESA Planning: B1 Security*, GC28-1800
- *MVS/ESA Planning: Security*, GC28-1604 (for MVS/SP Version 4)
- *MVS/ESA Planning: Security*, GC28-1801 (for MVS/SP Version 3)
- *MVS/ESA SPL: Application Guide*, GC28-1852

- *MVS/ESA SPL: Initialization and Tuning*, GC28-1828
- *MVS/ESA SPL: Installation Exits*, GC28-1836
- *MVS/ESA SPL: System Management Facilities*, GC28-1819
- *MVS/ESA VSAM Administration Guide*, SC26-4518
- *MVS/ESA VSAM Administration: Macro Instruction Reference*, SC26-4517

JES2

- *MVS/ESA SPL: JES2 Customization*, LY28-1010
- *MVS/ESA SPL: JES2 Initialization and Tuning*, SC28-1038

JES3

- *MVS/ESA JES3 Conversion Notebook*, GC23-0014
- *MVS/ESA JES3 Customization*, LY28-1013
- *MVS/ESA Operations: JES3 Commands*, SC23-0074
- *MVS/ESA SPL: JES3 Initialization and Tuning*, SC23-0073

RACF

- *Resource Access Control Facility (RACF) Auditor's Guide,* SC28-1342
- *Resource Access Control Facility (RACF) Command Language Reference,* SC28-0733
- *Resource Access Control Facility (RACF) Data Areas* LY28-1830
- *Resource Access Control Facility (RACF) Diagnosis Guide* LY28-1016
- *Resource Access Control Facility (RACF) General Information,* GC28-0722
- *Resource Access Control Facility (RACF) General User's Guide,* GC28-1341
- *Resource Access Control Facility (RACF) Macros and Interfaces,* SC28-1345
- *Resource Access Control Facility (RACF) Master Index,* GC28-1035
- *Resource Access Control Facility (RACF) Messages and Codes,* SC38-1014
- *Resource Access Control Facility (RACF) Program Directory for MVS Systems,* GC28-1054
- *Resource Access Control Facility (RACF) Program Directory for VM Systems,* GC28-1034
- *Resource Access Control Facility (RACF) Security Administrators Guide,* SC28-1340
- *Systems Programming Library: RACF,* SC28-1343

TSO/E

- *TSO/E Version 2 Command Reference*, SC28-1881
- *TSO/E Version 2 Customization*, SC28-1872
- *TSO/E Version 2 Primer*, GC28-1879
- *TSO/E Version 2 User's Guide*, SC28-1880

PSF/MVS

- *About Type: IBM's Guide for Type Users*, G544-3122
- *About Type: IBM's Technical reference for 240-Digitized Type*, S544-3516
- *Advanced Function Printing Diagnosis Guide*, LH40-0201
- *Advanced Function Printing Printer Information Manual*, G544-3290
- *Advanced Function Printing Printer Summary*, G544-3135
- *Advanced Function Printing Software General Information*, G544-3415
- *Font Data Stream Reference*, S544-3289
- *Print Data Stream Reference*, S544-3202
- *Print Services Facility Diagnosis Guide*, LH40-0202
- *Print Services Facility Security Guide*, S544-3291
- *Print Services Facility/MVS Application Programming Guide*, S544-3084
- *Print Services Facility/MVS Licensed Program Specifications*, GH35-0055
- *Print Services Facility/MVS Messages and Codes*, S544-3431
- *Print Services Facility/MVS System Programming Guide*, SH35-0091
- *Print Services Facility/MVS Update Guide*, S544-3133

Acknowledgements

The authors of this publication are:

Shuichi Aoki	ITSC Poughkeepsie
Bonnie Barthel	ITSC Poughkeepsie
Ulrich Boche	IBM Germany
Luiz Fadel	ITSC Poughkeepsie
Joseph Goyanko	IBM Philippines
Willie Kooistra	IBM New Zealand
Kurt Meiser	Coopers & Lybrand
Ron Northrup	ISM South Africa
Antti Numminen	IBM Finland
Roberto Oso	IBM Brazil
David Pasch	IBM Australia
Bernard Race	IBM France
Paul Rogers	ITSC Poughkeepsie
Andre van Wyk	ISM South Africa
Najamuddin Zabidi	IBM Malaysia

Also contributing to this document are:

Michael Cox	WSC Gaithersburg
George Dawson	IBM Australia
Horst Martin Dzatkowski	IBM Germany
Martin Ferrier	IBM UK
John Hutchinson	WSC Gaithersburg
Takayuki Kaiso	IBM Japan
Tapio Koskinen	IBM Finland
Mike Marsh	IBM UK
Hans Hoy Nielsen	IBM Denmark
Michel Perraud	IBM France
Sirkka Siltanen	IBM Australia

The authors are indebted to the many individuals from DSD who took time to review this presentation and to provide technical corrections and suggested improvements.

Comments by readers of this document will be greatly appreciated. Please feel free to return your Reader's Comment Form to the ITSC, Poughkeepsie.

Abbreviations

In this document, the following abbreviations are used:

ACF/VTAM* or VTAM* ACF/VTAM* Version 3 Release 3, program number 5685-085.

DFSMS* The Data Facility Storage Management Subsystem, consisting of the following products at these minimum release levels:

- MVS/DFP* Version 3 Release 1.1
- DFHSM Version 2.4, program number 5665-329
- DFDSS Version 2.4, program number 5665-327
- RACF Version 1.8.1, program number 5740-XXH
- DFSORT Version 1 Release 10, program number 5740-SM1

MVS or MVS/SP 3.1.3 MVS/SP* Version 3 Release 1.3, program numbers 5685-001 (JES2) or 5685-002 (JES3).

MVS/DFP* MVS/DFP* Version 3 Release 1.1, program number 5665-XA3.

MVS/ESA* or MVS/SP* Version 3 Program numbers 5685-001 (JES2) or 5685-002 (JES3), and MVS/Data Facility Product (MVS/DFP*) Version 3, program number 5665-XA3.

MVS/XA* MVS/SP-JES2 Version 2 Release 2.0, program number 5740-XC6, or MVS/SP-JES3 Version 2 Release 2.1, program number 5665-291, unless otherwise stated.

PSF or PSF/MVS PSF/MVS Version 1 Release 3, program number 5665-275.

RACF 1.9 RACF Version 1 Release 9, program number 5740-XXH.

RMF Resource Measurement Facility Version 4 for MVS/ESA*, program number 5685-029.

TSO or TSO/E TSO/E Version 2 Release 1.1, program number 5685-025.

Chapter 1. Implementing Security in an MVS/ESA Environment

IBM offers a variety of new and enhanced system security and integrity controls for its MVS/ESA operating system. These controls function in combination with RACF 1.9 and certain related subsystems and components. Although some of the new functions are available for older software versions and releases and combinations of products at different levels, the full set of functions is provided when the software components are at the following minimum required levels:

- MVS/SP 3.1.3
- MVS/SP-JES2 3.1.3 or MVS/SP-JES3 3.1.3
- MVS/DFP 3.1.1
- ACF/VTAM 3.3
- TSO/E 2.1.1
- RACF 1.9
- PSF/MVS 1.3

Appendix A, “Minimum Software Requirements for New Functions” on page 263 describes the minimum set of software required to implement each of the new functions. This chapter provides an overview of the new and enhanced functions available with the full set of related products. These enhancement can be categorized as follows:

- Extensions of the security environment
- Enhanced control over jobs
- Enhanced control over resources
- Refined and extended RACF controls
- Enhanced networking controls

1.1.1 Security Environment Extensions

The MVS/SP 3.1.3 security enhancements provide more complete security for the computing environment in the following areas:

- Console operator logon
- Operator command security
- Security environment for all work

1.1.1.1 Console Operator Logon

Access to operator consoles can be controlled by defining console operators as RACF users. A new RACF class, CONSOLE, is used to define profiles for operator consoles and construct access lists of users or groups authorized to use them. The resource name is the MCS console-ID. When setting up console control, one of the following system-wide options must be selected for logon:

- Optional - an operator can log on; if an operator is not logged on, the existing MCS console authorization is in effect.
- Automatic - the console is defined as a RACF user and automatically logged on with that userid during system startup.
- Required - an operator must log on with a valid userid and password before commands are accepted.

Access to all consoles should be controlled through RACF using the required option. Individual accountability can be obtained by using this option with operator command control.

1.1.1.2 Operator Command Security

Operator command authorization and auditing can now be controlled through RACF based on user or group authorization. This new control includes MVS, JES2, or JES3 commands and is used regardless of the source of the command, such as operator console, batch job stream, internal reader, SVC 34, or SDSF. SMF records written by RACF contain the full command string for auditing purposes. A new RACF class, OPERCMDS, is used to define command profiles. Profile name prefixes identify the commands types: MVS commands have the prefix MVS; JES commands have the prefix JES2 or JES3. The existing MCS authority levels are carried over to RACF through access level requirements. Master console authority is given with CONTROL access, the ALL authority is given with UPDATE access, and the INFO authority is given with READ access in the access list. RACF control over operator commands should allow all users to enter display commands and restrict other commands to individually authorized groups or users.

RACF control of operator commands is a significant improvement for system integrity and security and should be used by all MVS installations. Where various independent controls had to be used in the past, the use of operator command control provides one central point of control - RACF.

1.1.1.3 Security Environment for All Work

The protection of jobs has become more comprehensive. Jobs are now protected throughout their entire lifetime from *first card in* to *last line out*. Two elements are used to achieve this level of protection:

- Security tokens are control blocks attached to a job for its entire lifetime; for its input, execution, and output phases. Before MVS/ESA 3.1.3, RACF protection was available only during the execution phase. A security token contains information such as the execution userid, the submittor ID, the security label, and the port of entry. Security tokens are built and maintained by the MVS system authorization facility (SAF) and exist as long as MVS/ESA 3.1.3 is installed. The set of products listed earlier implement security tokens; downlevel products may provide only a subset of the new functions because they do not use security tokens.
- Printer output can be produced with automatic page overlays based on a job's SECLABEL. This function requires printers with PSF capabilities and the definition of two areas on a printed page, the user printable area (UPA) and the area where security overlays can be printed. Output labeling is automatic when the above conditions are met and the SECLABEL class is active. In addition, a new RACF class, PSFMPL, can be used to authorize a user to suppress output labeling and, for example, use the space outside the UPA. Output labeling is desirable in any installation. However, be aware that this feature may require applications to be modified because of the smaller area where data may be printed.

1.1.2 Enhanced Job Control

The MVS/ESA security enhancements provide additional and improved controls over jobs in these areas:

- Job submission controls
- Surrogate user support
- Output and cancel controls

1.1.2.1 Job Submission Controls

Job submission controls include:

- Early job verification as a standard function that cannot be turned off - SETROPTS NOEARLYVERIFY is not effective. The verification has been enhanced to include SECLABEL checking at the time a job is submitted. Early verification has always been a useful option and it is now a standard feature.
- Userid propagation now includes propagation of the user's SECLABEL across NJE nodes. Propagation is a standard feature and can be turned off only for certain userids, as before in RACF 1.8.1, through the PROPCNTL class. Profiles in this class are used to suppress propagation selectively for environments such as multi-user address spaces where submitted jobs are not generally allowed to inherit the address space userid and authority. Propagation is a very useful function that supports security and usability in an MVS system with RACF.
- Job submission and cancellation can be controlled by jobname through RACF. Similar controls were previously available only through exits such as the TSO submit and output/cancel exits. JESJOBS is a new RACF class in which jobname profiles can be defined and user or groups put on the access list. Profiles have a qualifier of SUBMIT or CANCEL to distinguish between these controls. Jobname control can be implemented for all jobnames or selectively where only certain jobnames are excluded from public use and users can freely select jobnames other than the restricted ones. Selective jobname control is usually easy to implement through a few profiles in JESJOBS. However, it may not cover existing TSO exit controls such as restricting a user to his userid plus an additional character. To replace existing exit functions with RACF controls often requires implementation of comprehensive jobname control. This may require profiles for each user in JESJOBS and corresponding maintenance as user definitions change. Where possible, equivalent entries in the global access checking table for JESJOBS containing the &RACUID variable may be a good solution.

  Jobname control is a RACF feature that needs careful evaluation prior to its implementation.
- RACF 1.9 can now control the way jobs are entered into the system. Users and groups can be given or denied access to JES devices, RJE/RJP readers, NJE readers, and the internal reader. In addition to checking admission to the system at submit time, these ports of entry (POE) are recorded in the security token of a job and can be used for conditional access checking when the job is executing. POE profiles can be defined in the new JESINPUT class, and users and groups can be put on the access lists. A limitation may exist because a single POE profile (INTRDR) controls all jobs coming from the JES internal reader. Therefore, a distinction between TSO submitted jobs and jobs sent through SYSOUT INTRDR is not possible. Job entry control is a very useful feature, particularly in conjunction with conditional access checking.

1.1.2.2 Surrogate User Support

This new RACF 1.9 function authorizes a surrogate user to submit a job for a principal user without knowing the principal' password or sharing his privileges and authorities. This approach provides a controlled way of delegating the submission of jobs while maintaining individual accountability. The security token of a job contains the IDs of both the submittor and the principal user. Similar support was previously available only through a router exit. Profiles for principal userids can be defined in the new SURROGAT class, and surrogate userids or groups are added to the access list.

While surrogate user support can be used for real users, it can also be used for automated job submission. It allows a user to set up production IDs with fine and granular access authority for which jobs are submitted in a secure way through a central job scheduler package. This function is a very important security enhancement and should therefore be used by all MVS/RACF installations. Current users of the router exit are encouraged to switch to the new support.

1.1.2.3 Output and Cancel Controls

SYSIN and SYSOUT data sets can now be protected through RACF; similar controls previously required JES or TSO exits. JES uses a new data set name structure for SYSIN and SYSOUT that allows the user to specify the lowest level qualifier in his JCL and to write access rules for his spool files. Profiles for SYSIN and SYSOUT data sets are defined in the new JESSPOOL class. Browsing requires only READ access authority; any other operation (such as requeuing, printing and deleting) requires ALTER access. Output processing using TSO OUTPUT, external writers, TSO TRANSMIT, BDT, or SDSF requests are all controlled by RACF. A new sample exit, IKJEFF53, that is sensitive to the JESSPOOL class is available. This new RACF control can and should replace functions currently implemented in exits. It allows granular control of output by the end user if the GENERICOWNER concept described in 4.8, “GENERICOWNER” on page 43 is implemented for the JESSPOOL class.

RACF 1.9 can be used to control the destination of output and can prevent output from printing on printers other than those required by the particular application or security classification. A new RACF class, WRITER, is used to define profiles for output devices and to allow user or group access. SECLABELs are checked following reversed MAC logic; to avoid exposures, the SECLABEL of the output device must dominate the label of the job output. Selective RACF 1.9 destination control is a good security enhancement in any MVS installation.

1.1.3 Enhanced Control over Resources

The MVS/ESA security enhancements provide enhanced controls over the following types of resources:

- Global controls
- Data set protection
- General resources

1.1.3.1 Global Controls

Two control enhancements fall into this category:

- Conditional access is the most specific way to specify access authority in RACF. It supports the concept of application integrity. Previous RACF releases supported WHEN(PROGRAM). RACF 1.9 now supports the following new conditions:
  - WHEN(TERMINAL) - RACF accepts the request only if the user is logged on to a specific terminal. A typical application would be to limit the submission of certain jobs or commands to specific terminals.
  - WHEN(JESINPUT) - the request is granted only if the the job has entered the system through a specific JES input device. This function can, for example, be used to prevent the submission of production jobs from RJE stations.
  - WHEN(CONSOLE) - requests are restricted to a certain system console. As an example, cancelling TSO users is valid only if the command is entered at a specific console.

  Conditional access controls should be considered by all MVS installations for resources with very restricted access.
- The universal access (UACC) entry on an access list determines the access authority of users not specifically authorized through user or group entries. RACF 1.9 now allows users to make a distinction between RACF defined and undefined users. ID(*) can be used to set the access authority for all defined users not specifically authorized. When ID(*) is used, the UACC value applies only to undefined users. For example, ID(*) ACCESS(READ) in combination with UACC(NONE) could be used to exclude all undefined users from access while granting registered users read access.

Except for an initial migration period, undefined users should not exist in most well managed RACF installations. Few resources should have ID(*) ACCESS(READ) or UACC(READ) authority; EXECUTE or NONE is, in most cases, the appropriate levels of public access. The new control provides benefits only where these fundamental rules cannot be implemented.

1.1.3.2 Data Set Protection

The following enhancements have been made to data set protection:

- A new RACF option, CATDSNS, can be used to require that all permanent data sets be cataloged. Besides a convenient way of enforcing a policy, this option may also be used for the benefit of the new DSNS option of the LISTDSD command. This option lists the names of all data sets protected by a generic profile. Since it uses the catalog to create the list, the list is more dependable if CATDSNS is active. With this option active, RACF does not grant access to uncataloged permanent data sets even if MAC and DAC checking would allow access. Data sets must be cataloged through the master catalog; JOBCAT and STEPCAT environments are not supported.

  Exceptions to these rules are necessary. For example, the maintenance of more than one system makes it necessary to have uncataloged system data sets accessible to the system support group. Such exceptions can be implemented through:

  - Discrete data set profiles
  - Fully qualified generic data set profiles
  - ICHUNCAT profiles in the FACILITY class to allow access to an uncataloged data set
  - ICHUCAT profiles in the FACILITY class to allow the use of JOBCAT or STEPCAT

  CATDSNS should be considered by all MVS installations. This option should be implemented carefully using the WARNING option and exceptions should generally be defined in the FACILITY class.

- Temporary data sets are protected by MVS during the lifetime of a job and are automatically deleted at job completion. Traditionally, RACF has not addressed these data sets except for the ERASE ALL option. Potential exposures existed in shared DASD environments (access from a different MVS) and when temporary data sets were not deleted due to system crashes. A new class, TEMPDSN, which does not contain any profiles, enables this function. Orphan temporary data sets can then be deleted by users with the OPERATIONS attribute and no user can access them. This is a reasonable and useful option for all MVS installations to select.

1.1.3.3 General Resources

A potential problem with generic profiles is that, prior to RACF 1.9, users with class authority (CLAUTH) to a resource class could define or change any profiles in that class. For this reason, CLAUTH could in many cases not be granted to users with a need to establish profiles in their area and these profiles had to be established through security administration. For general resource classes, an installation can now establish more granular control over the creation of profiles through SETROPTS GENERICOWNER. More users can be given class authorization and be allowed to define their own profiles without interfering with other profiles. When GENERICOWNER is active, users are authorized to create profiles only under an existing profile they own. To use this function effectively, the system administrator creates one high-level profile that he owns and, for each user with CLAUTH, one more specific profiles owned by the user. New resource classes such as JESSPOOL and JESJOBS can be used very efficiently with the GENERICOWNER concept. This control should be used by all MVS installations.

1.1.4 Refined and Extended RACF Controls

A number of existing controls were refined and some new controls were created, including:

- Resource name enhancements
- Device allocation control
- Destination control
- Auditing enhancements

1.1.4.1 Resource Name Enhancements

Resource naming has been enhanced by:

- Generic naming enhancements in the following areas:
  - Generic names in the DATASET class can now contain ** in the middle of the profile instead of only at the end. The ** matches zero or more qualifiers. The default is NOEGN.
  - Enhanced generic naming is now always in effect for general resource profiles.

  SETROPTS EGN should be specified on all systems and never deactivated.
- RACF variables can be defined as group profiles in the RACFVARS class. These variables can be used as name qualifiers of general resource profiles to protect all members of the RACVARS profile at once. This is a convenient way to protect resources with identical protection requirements and unlike names with one resource profile. Traditionally, generic profiles could be used to protect multiple resources with similar names with one profile. Also, group profiles could be used in specific classes (that were accessed through RACLIST and FRACHECK) for resources with unlike names. This new function extends the group concept to all general classes. These variables cannot be used in the DATASET class. The new facility is very useful in reducing RACF administration effort.

1.1.4.2 Device Allocation Control

The MVS/ESA security enhancements provide RACF control over the allocation of unit-record, teleprocessing or communications, and graphic devices. Profiles in the new DEVICES class permit users or groups to such devices. The profile names contain the device name and address. This is an important control to consider.

1.1.4.3 Destination Control

RACF 1.9 and JES 3.1.3 provides control over output destinations based on both users and data. Controls specify which users or groups can process the output and which data classifications can be printed on which JES devices. These JES devices include local printers, RJE printers, and NJE transmitters. Profiles in the WRITER class authorize users through access lists. SECLABELs can be used to verify that the device classification meets the requirements of the output. SECLABEL checking is reversed to ensure that the classification of the device meets or exceeds the data classification. This is an important function for installations using security classification.

1.1.4.4 Auditing Enhancements

The RACF auditing functions have been enhanced as follows:

- Logging is now possible in the PROGRAM class.
- SETROPTS LOGOPTIONS is a new global option to override or enhance other logging options by resource class. LOGOPTIONS is also used to request logging in classes without profiles.

- SETROPTS SECLABELAUDIT is a new option that causes logging for all resources protected by a security label. SETROPTS SECLEVELAUDIT still provides logging for the security levels defined in resource profiles.

The new logging options should be evaluated carefully and used as required by the installation's security policy. The RACF Report Writer has been enhanced to support the new logging options.

1.1.5 Enhanced Networking Controls

Network controls have been enhanced in the following areas:

- VTAM controls
- RJE/RJP security
- NJE security

1.1.5.1 VTAM Controls

There are two new VTAM controls:

- VTAM application authorization verifies that non-APF authorized programs that open a VTAM application ACB are authorized through a profile in the new VTAMAPPL class.
- VTAM LU6.2 session verification uses session keys that are stored and managed in APPCLU class profiles to authenticate LU6.2 nodes.

Both controls aid in achieving system integrity.

1.1.5.2 RJE/RJP Security

The authentication of RJE or RJP stations can now be performed through RACF. Traditionally this process involved password comparisons in JES. RACF authentication requires two elements:

- The remote terminal name must be defined in the FACILITY class. This triggers password checking in RACF instead of JES.
- The remote terminal name must be defined as a RACF user.

This approach eliminates some of the potential exposures that previously existed and should be used in all RJE/RJP installations

1.1.5.3 NJE Security

NJE security through RACF 1.9 is a major enhancement of the security in networked host environments. It consists of several elements:

- NJE nodes are defined to RACF in the new NODES resource class. These definitions identify the nodes and establish their properties for the local system. The most important node property is the level of trust, which determines the general treatment of jobs and output received from a node:
  - trusted - job are accepted - no password re-verification
  - semi-trusted - jobs are accepted - password required
  - untrusted - jobs or SYSOUT not accepted.

  In addition, local nodes can be defined in a &RACLNDE profile; jobs are accepted and passwords are optional here. Other more specific controls defined in NODE profiles are userids, group names, and SECLABELs.

- Jobs submitted from other nodes are first evaluated against the general level of trust in the node and then against the node specifications of acceptable userids, group names, and SECLABELs attached to the job.
- Likewise, SYSOUT coming from other than an untrusted node is evaluated against node specific user, group, and SECLABEL rules for the acceptance of output.
- Userids and SECLABELS are now propagated across the network. In addition, userids, group names, and SECLABELS can be translated at the receiving node, contingent on the level of trust in the sending node:
  - Incoming jobs are translated if the node is at least trusted.
  - Incoming SYSOUT is translated if the node is at least semi-trusted.

The new NJE support is a very powerful and important security enhancement that should be considered by all NJE installations. However, it is much more complex than this simple overview suggests and must therefore be carefully be analyzed and implemented.

Chapter 2. SECLABELs

Discretionary access control (DAC) is a method of restricting access to resources based upon the identity of a user or the groups to which a user belongs. It is called discretionary because the owner of the data uses his judgement in passing access to the data on to another user based on that user's need to know. A user is permitted access to a resource at a specific access level. For instance, a user may have update access to some resources, but only read access to others. Discretionary access is hierarchical; update access also allows a user read access.

Mandatory access control (MAC) is a method of restricting access to objects, based on the sensitivity of the information that the object contains and the authorization of the user to access information at that level of sensitivity. This control is called mandatory because it is outside the control of the owner of the data. Usually, it is under the control of the system or security administrator. MAC is a means of putting a *brick wall* around resources with the confidence that, while discretion may still take place within the wall, a user may not pass access outside the wall. This concept establishes a set of rules or security policy to prevent users from accessing information outside their own authorization and preventing them from declassifying the information.

Mandatory access control is used in combination with discretionary access control; MAC provides central enforcement of an organization's security policy, above and beyond DAC. DAC is based on access rules typically defined by resource owners (at their discretion); MAC establishes boundaries for the validity of DAC rules. RACF verifies an access request against MAC requirements first, and if successful, then evaluates the request against DAC access rules. If MAC requirements are not met, RACF fails the access request even if DAC authorization is sufficient. Figure 1 illustrates both DAC and MAC from the user's perspective.

Mandatory access control was first introduced in RACF 1.7 in the form of security levels and categories; RACF 1.9 provides the new SECLABEL support as an extension of the original concept.

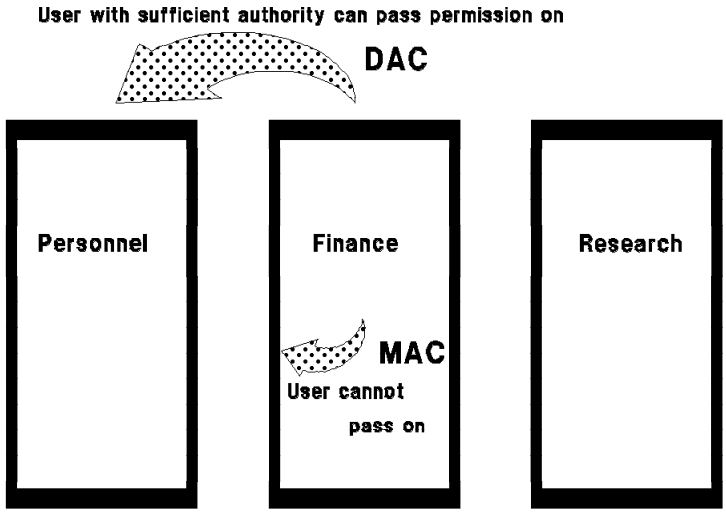

Figure 1. DAC and MAC User Views

2.1 Security Levels

Security levels can be used to label resources based on their sensitivity. Levels are usually defined in an organization's security policy as part of an information classification scheme. They have names that are meaningful for the organization and can be used to label documents and other forms of information. Information classification provides, for each level, guidance for the correct identification and labeling of information assets as well as handling and protection requirements.

Security levels can be defined to RACF and assigned to information assets stored and maintained on MVS systems such as data sets, terminals, and programs. Users can then be authorized to access levels; this authorization is commonly called user clearance.

To illustrate classification in security levels, Figure 2 shows the complete set of information assets subdivided horizontally into segments representing the different levels of sensitivity. In this example, the security levels Restricted, Confidential, Internal, and Unrestricted are hierarchical classification elements from the highest to lowest level. A user with a clearance of Restricted has access to all levels; a user with a clearance of Internal can access Internal and Unrestricted, but not Confidential or Restricted.

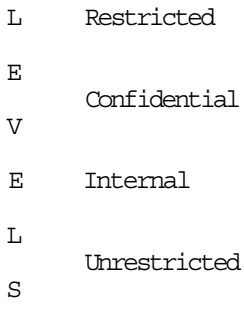

Figure 2. Security Levels

2.2 Security Categories

Categories can be used to label resources with regard to independent compartments to which they belong. They are defined with meaningful names in an organization's security policy. Categories provide the controls required by service bureaus or management information system (MIS) departments with similar orientation to segregate their customers from each other and from system resources. Even organizations without a defined classification scheme usually have natural categories.

Security categories can be defined to RACF and assigned to information assets stored and maintained on MVS systems independent of security levels. To illustrate classification by categories, Figure 3 shows the complete set of information assets subdivided vertically into compartments representing different business areas and applications.

In this example, the categories are Personnel, Finance, Research, and Other. Unlike hierarchical security levels, categories are independent classification elements; the category Personnel is an entity completely independent and separate from Research. A user needs separate clearance to each category to be allowed access. Categories are the *brick walls* that represent MAC.

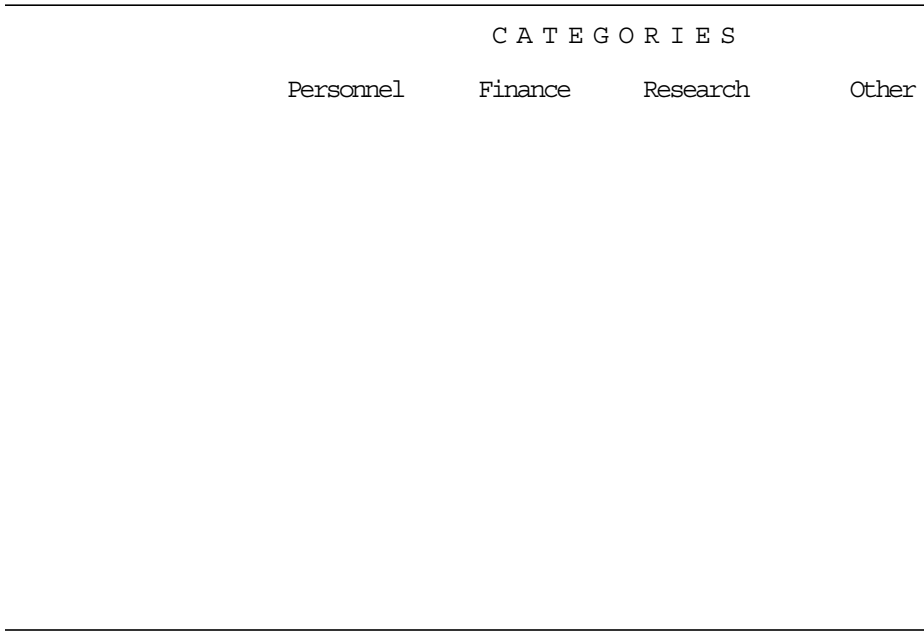

Figure 3. Security Categories

2.3 Security Levels and Categories

Information assets are classified in two ways; they exist in intersections of level and categories. To access a classified object, users must have clearance to the specific category and to a level equal to or higher than the object's level. A classification scheme using both security levels and categories, prior to RACF 1.9 can be depicted as shown in Figure 4. In this example, object X has a security level of Confidential and the category Research. A user allowed to access object X must be defined with a category of Research and a security level of Confidential or Restricted. Although a combination of security level and categories is the typical implementation, RACF does allow the use of only levels or only categories; they are independent entities.

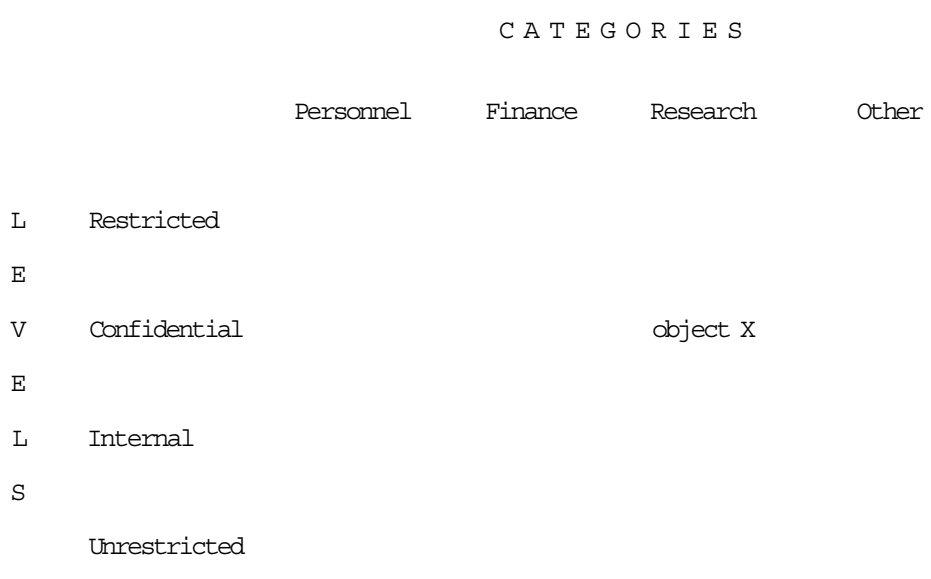

Figure 4. Security Levels and Categories

2.4 Security Labels

Security labels, or SECLABELs, offer more flexibility and choices than the older concept, but can increase the complexity of a MAC implementation. A SECLABEL is an externalized, named label that represents a combination of one security level and zero or more categories. A SECLABEL is not required to have any categories, but it must have a security level. When associated with resources, users, and jobs, security labels provide the following enhancements over security levels and security categories:

- Users can log on with different security labels at different times, but with the same userid. Without security labels, a user always has the same default security level and categories.
- Output printed for a user or job by the Print Services Facility (PSF/MVS) can have an overlay related to the security label of the user or job printed on every page.
- It is easier to maintain the security label of users and data. Changing the definition of a security label affects all users and resources that have that security label; the change does not have to be made to each profile as required with security levels and categories.

Although SECLABELs are mandatory in a B1 environment, they can also be very useful in non-B1 security implementations.

In Figure 5, two sets of SECLABELs with different characteristics are shown:

- Object labels, designed primarily to classify data, represent a combination of a security level and one category. They are shown in parentheses and are named according to the level and category they represent, such as PR (Personnel/Restricted), RC (Research/Confidential), FI (Finance/Internal), and OU (Other/Unrestricted). Where appropriate, users can also be given access to these labels.

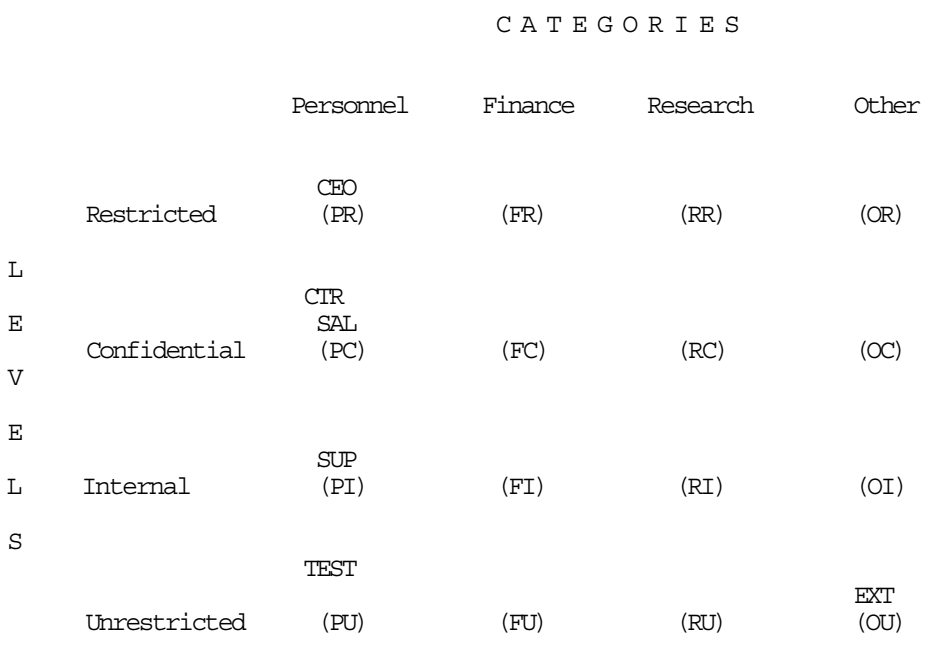

Figure 5. Security Labels

- Subject labels, designed primarily to authorize access, represent a combination of one security level and several categories. These labels have names related to job functions, such as CEO (Chief Officers), CTR (Comptroller), SAL (Salary - management access to confidential data in Personnel and Finance), SUP (Supervisor access to internal data across all categories), TEST (access to all unrestricted data across categories), and EXT (access for external users) and are shown in boxes that can span more than one category.

2.4.1 Dominance Concept

The dominance concept says that SECLABEL X *dominates* SECLABEL Y if (1) the level of SECLABEL X is greater than or equal to the level of SECLABEL Y, AND (2) category set X includes all categories in category set Y. Note that category set X may have more categories than Y, but it must have all categories of Y. Figure 6 shows the dominance relationship among all the SECLABELs defined in Figure 5 on page 12. For example, the CEO SECLABEL dominates the SAL SECLABEL because the Restricted level is higher than the Confidential level and the CEO SECLABEL has access to both the Personnel and Finance categories of the SAL SECLABEL.

MAC checking uses the dominance concept in evaluating access requests. Typically, the user′s SECLABEL must dominate the resource′s SECLABEL; the security level represented by the user′s SECLABEL must be equal to or higher than the security level represented by the resource′s SECLABEL and the security categories represented by the user′s SECLABEL must contain all the security categories represented by the resource′s SECLABEL. For some resources, however, the opposite is true; the resource′s SECLABEL must dominate the user′s SECLABEL. This reverse dominance prevents a user with a high clearance level from sending data to an output device with a lower level. This condition can be defined for resource classes through the CDT attribute RVRSMAC. Reverse MAC checking means that the SECLABEL processing is the reverse of what is typical; the resource′s SECLABEL must dominate the user′s SECLABEL. Standard IBM classes with this property are TERMINAL, CONSOLE, and WRITER.

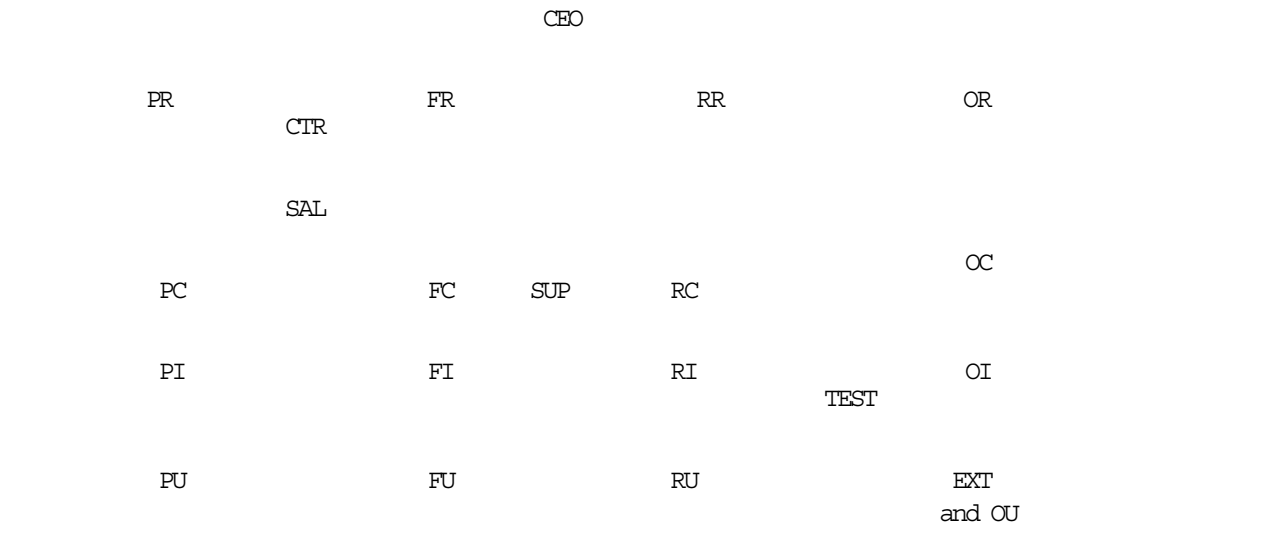

Figure 6. SECLABEL Dominance

2.4.2 System-assigned SECLABELs

At IPL time, RACF dynamically creates system SECLABELs that cannot be defined, modified, or deleted by any user, but can be assigned to users and resources. The following SECLABELs are defined:

- SYSHIGH is created at the highest security level defined to RACF and includes all defined categories. Resources with SYSHIGH can be accessed only by users with SYSHIGH; users with SYSHIGH have access to all other SECLABELs. If a higher level or additional categories are defined, SYSHIGH automatically assumes the new values when the SECLABEL class is refreshed.
- SYSLOW is created at the lowest security level defined to RACF and includes no categories. Resources with SYSLOW can be accessed by users with any valid SECLABEL; users with SYSLOW can access only resources that are SYSLOW. If a lower level is defined, SYSLOW automatically assumes the new value when the SECLABEL class is refreshed.
- SYSNONE is assigned to resources that have no security-relevant data. MAC verification considers SYSNONE to be equivalent to any user's SECLABEL, automatically allowing access. Assigning SYSNONE to a user has the same effect as assigning SYSLOW to the user.

These labels are used by system processes. Specific IBM guidance should be followed when assigning these labels. Figure 7 shows the relationship between the system-assigned SECLABELs and all the SECLABELs defined in Figure 5 on page 12.

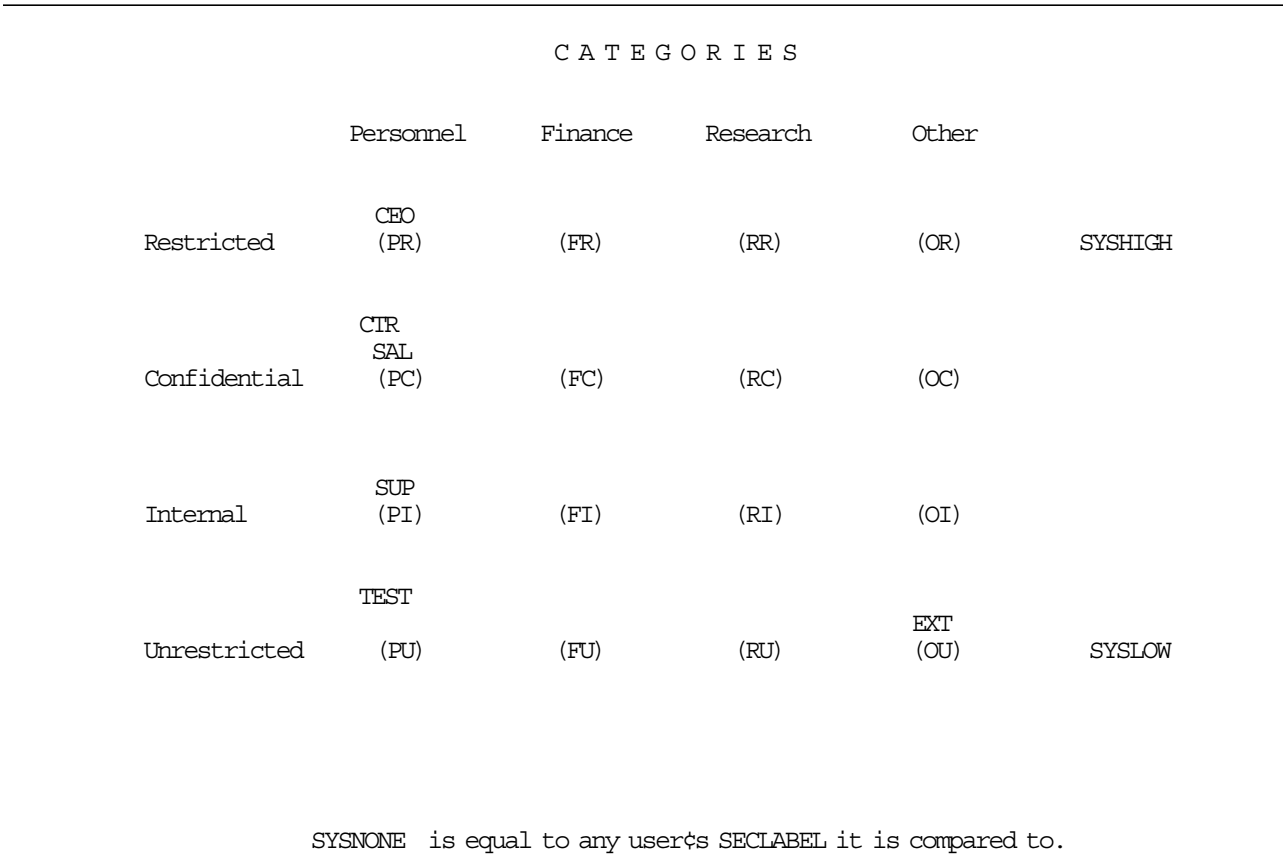

Figure 7. System SECLABELs

To determine the level and categories of SYSHIGH and the level of SYSLOW, use the RLIST commands:

```
RLIST SECDATA LEVEL

RLIST SECDATA CATEGORY
```

By definition, SYSHIGH is at the highest level defined with all defined categories; SYSLOW is at the lowest level defined with no categories. Figure 8 shows an example for the SECLABELs defined in Figure 5 on page 12.

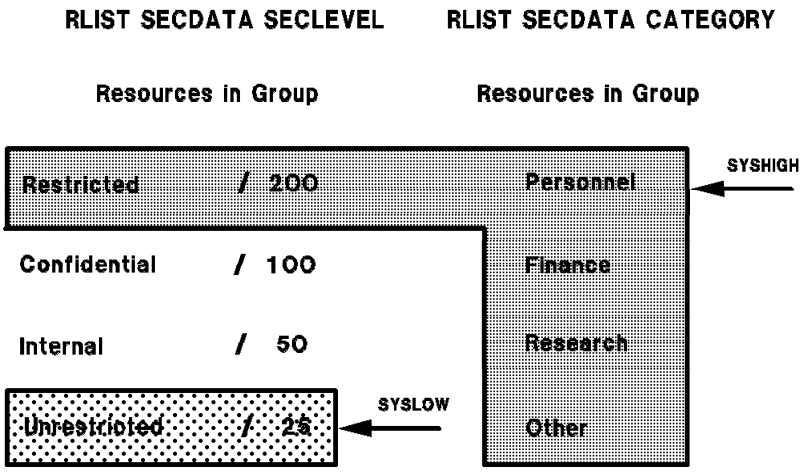

Figure 8. Listing for SYSHIGH and SYSLOW

2.4.3 RACSLUNK SECLABEL

RACSLUNK is a special SECLABEL that JES assigns to SYSOUT that is received from another node if the SECLABEL class is active on the receiving node and the NJE header contains an unknown or blank SECLABEL. If the SECLABEL class is inactive on the receiving node and if RACF is at the Release 1.9 level, the received SYSOUT retains its original value; if RACF is downlevel, the SECLABEL is set to blanks.

Since the RACSLUNK SECLABEL is not predefined, no one is able to access the SYSOUT. This situation can be handled in one of two ways:

- Define RACSLUNK to RACF as a valid SECLABEL and authorize users to access it.
- Use the NODES class translation facility to translate it to a known SECLABEL.

For additional information on the translation facility see Chapter 8, “NJE Security Control” on page 117.

2.4.4 Comparison of Levels and Categories

Table 1 compares the use of SECLABELs to LEVELs and CATEGORYs.

.

Table 1. SECLABELs Compared to SECLEVELs and CATEGORYs

Function	SECLABELS	SECLEVEL and CATEGORYs
Implementation	External; can be coded in JCL or entered on a TSO logon screen.	Internal to RACF.
Definition	Must have a level; categories are optional.	SECLEVELs and CATEGORYs are independent; either one or both can be defined.
Modification	Definition in the SECLABEL profile.	Definition in each resource profile.
Assignment	A user is assigned a default SECLABEL, but can be permitted to more than one SECLABEL by userid or group id.	Only one level and one set of zero or more categories can be assigned to a user.
Security Options	Allows many security options using RVRSMAC, MLACTIVE, and MLS.	Security options are fixed; the user's level and categories must dominate the resource's level and categories.
Protection of System Resources	Special system SECLABELs are used that change with new definitions.	Installation defined levels and categories are used that must be changed manually with new definitions.
Propagation	To batch job submissions, created SYSOUT data sets, across nodes.	None.
PSF Support	Overlay printed on every page.	None.

Chapter 3. Implementing SECLABELs

Starting with RACF 1.9 it is possible to define security labels, or SECLABELs. Figure 5 on page 12 shows the relationship among security level, security category, and SECLABEL. In summary:

- A **security level** is an installation-defined name that corresponds to a numerical security classification. The security level is hierarchical; the higher the number, the higher the security level. Security levels are a set of values having a fixed order, such as Restricted, Confidential, Internal, and Unrestricted.
- A **category** is an installation-defined name corresponding to departments or areas with similar security requirements. The categories are non-hierarchical; each category is selected independent of the others; such as Personnel, Finance, Research, and Other.
- A **SECLABEL** associates a security level with zero or more categories. Since the SECLABEL is composed of both hierarchical and non-hierarchical elements, the term *dominance* is used to describe the relationship between two SECLABELs.

3.1 SECLABEL Checking

SECLABEL checking occurs only if the SECLABEL class is activated with the following command:

```
SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)
```

Since the SECLABEL class is defined in the class descriptor table (CDT) with the RACLREQ=YES option (RACLIST required) and DFTRETC=8 (default return code), it must be brought into storage using RACLIST. Otherwise, RACF does not find the SECLABEL profile in storage, does not do I/O to the RACF database looking for a profile, and returns an access request denied condition (return code 8) to the caller which fails the RACF authorization. Because the SECLABEL profiles are kept in storage, the following command must be used to refresh the in-storage profiles whenever a SECLABEL definition changes:

```
SETROPTS RACLIST(SECLABEL) REFRESH
```

Otherwise, the old definition continues to be used.

To understand how a SECLABEL is checked, it is important to understand the concept of *dominance* as defined in Figure 6 on page 13. The following access authorizations bypass SECLABEL checking:

- If a started task is defined as trusted or privileged in the started procedure table (SPT), it is allowed to access any resource without SECLABEL checking.
- If a program is defined with NOPASS in the programming properties table (PPT), it is allowed to access any data set without SECLABEL checking.
- If permitted by the global access checking table, the user is allowed to access the resource without SECLABEL checking.

If the SECLABEL check passes, RACF proceeds with authorization checking just as though the SECLABEL class was not active; otherwise, the request is denied.

Note: For a complete list of authorization checks, refer to the *RACF Security Administrators Guide*.

When the SECLABEL class is active, the SECLABEL information in a resource profile is used for authorization checking and any existing SECLEVEL and CATEGORY information in that profile is ignored; however, the information is still maintained in the profile. When the SECLABEL class is not active, RACF continues to use the SECLEVEL and CATEGORY information as it is currently implemented.

A user with a SECLABEL defined can access any resource protected by a profile without a SECLABEL if he has discretionary access to the resource. A user without a SECLABEL defined cannot access any resource protected by a profile with a SECLABEL. Before a resource is protected by a SECLABEL, all users who need to access this resource should be assigned an appropriate SECLABEL; otherwise, they cannot access the resource. The following special considerations apply to SECLABEL processing:

- **Reverse MAC checking.** Several IBM classes are defined in the CDT with an attribute of reverse MAC checking (RVRSMAC=YES). For these classes, the resource's SECLABEL must dominate the user's SECLABEL. The IBM defined classes with this attribute are:

  ```
  - CONSOLE      - TERMINAL      - WRITER
  ```

  Reverse checking prevents the declassification of information. For example, a terminal in an unsecured place, such as the lobby, should be protected with a low SECLABEL. A user with a higher SECLABEL and, therefore, able to display more classified information, is not allowed to use this terminal. Otherwise, he could display classified information that could be viewed by an unauthorized person in the lobby.

  The WRITER class indicates who can print a SYSOUT data set on a specific printer. The following definition is an example:

  ```
  RDEFINE  WRITER  JES2.LOCAL.PRTxx  SECLABEL(SAL)  UACC(READ)
  ```

  With this definition, only SYSOUT data sets with a SECLABEL dominated by the SECLABEL SAL are allowed to print on PRTxx. Reverse checking is not affected by the setting of MLS.

- **TSO messages.** The LISTBC command uses only MAC checking, not DAC checking, in the DIRAUTH class when the class is active and appropriate parameters are set in the IKJTSOxx member of the SYS1.PARMLIB data set. The DAC is checked on the receiving side, and a user is not allowed to receive a message if the receiving user's SECLABEL does not dominate the message's SECLABEL. Refer to 13.7, "TSO Message Control" on page 249 for more information.

- **TSO TRANSMIT and RECEIVE commands.** The receiving user's SECLABEL must dominate the message's SECLABEL. For more information, see 7.4.4, "TSO TRANSMIT/RECEIVE Commands" on page 112.

- **Job submission.** A user is allowed to submit a job at a SECLABEL only if the submittor's SECLABEL dominates the job's SECLABEL or the job's SECLABEL dominates the submittor's SECLABEL. This restriction prevents the user from crossing category boundaries when submitting a job. If MLS is in effect, the job's SECLABEL must dominate the submittor's SECLABEL whether it is a TSO user submission or a surrogate submission. Refer to 6.9, "SURROGAT Class" on page 91 for more information.

- **Compatibility mode.** Compatibility mode provides SECLABEL compatibility between RACF 1.9 and down-level products that use old macro formats where a SECLABEL cannot be specified. Compatibility mode allows security label authorization checking to pass in situations that would normally fail, allowing down-level products to be compatible with SECLABELs until they can be enhanced to use SECLABELs. When compatibility mode is active, RACF checks all SECLABELs the user is permitted to. If any one dominates the resource SECLABEL, access is allowed. This checking is independent of the requestor's current SECLABEL. COMPATMODE is ignored for jobs that were verified with RACF 1.9 keywords; that is, it cannot be used to bypass SECLABEL checking for those products that do support SECLABELs. NOCOMPATMODE is in effect at RACF installation. Compatibility mode can be activated with the following command:

  ```
  SETROPTS  COMPATMODE | NOCOMPATMODE
  ```

  For example, DFHSM issues a RACINIT on behalf of the user each time a request to recall a data set is made. Since this macro does not allow a SECLABEL to be specified, an ACEE with the user's default SECLABEL is created. The RACHECK issued to verify whether the user is authorized to recall the data set may fail if the ACEE SECLABEL does not dominate the data set SECLABEL,

even though the user's current logged-on SECLABEL does. COMPATMODE allows the recall; normal authorization checking, including a SECLABEL check, is done at OPEN time.

- **MLACTIVE, MLS, or both active.** For more information on how MLS and MLACTIVE affect SECLABEL checking, refer to 3.6.2, "MLACTIVE Active" on page 25 and 3.6.3, "MLS Active" on page 27.

3.2 Defining a SECLABEL

Before defining the SECLABEL, the levels and categories that comprise the SECLABEL must first be defined in the SECDATA class. The SECDATA class does not have to be active if SECLABELs, not levels and categories, are being used. To define a SECLABEL's security level (level_name) use the following command:

```
RDEFINE SECDATA SECLEVEL ADDMEM(level_name/level_number)
```

The level_number associated with the level_name is what makes the security level hierarchical. If the SECLABEL is to have categories (category_name), they must also be defined using:

```
RDEFINE SECDATA CATEGORY ADDMEM(category_name)
```

Then the SECLABEL can be defined as follows:

```
RDEFINE SECLABEL label_name SECLEVEL(level_name) ADDCATEGORY(category_name)
```

SECLABEL definitions obey the following rules:

- The SECLABEL name must have from one to eight alphanumeric characters. The first character must be alphabetic.
- The SECLEVEL name must be specified. Only one name is allowed.
- The CATEGORY names are optional.

Note: With previous SECLEVEL and CATEGORY protection, it was possible to protect a resource with, or assign a user to, a CATEGORY without a SECLEVEL. With SECLABEL protection, a SECLEVEL is required; CATAGORYs are still optional. It is possible, however, to define only one SECLEVEL and assign it to all security labels.

Only a system SPECIAL user can define, delete, or alter a SECLABEL definition. To avoid security exposures, this ability can be further restricted using the MLSTABLE and MLQUIET SETROPTS options. For more information, refer to 3.5, "Controlling SECLABELs" on page 24.

3.3 Changing a SECLABEL

The following commands can be used to change a SECLABEL definition:

- To add or delete a CATEGORY from a SECLABEL:

  ```
  RALT SECLABEL label_name DELCATEGORY(category_name)

  RALT SECLABEL label_name ADDCATEGORY(category_name)
  ```

  After a CATEGORY is deleted from the SECLABEL definition, if it is not being used by any other SECLABEL, it should be deleted with the following command:

  ```
  RDEL SECDATA CATEGORY DELMEM(category_name)
  ```

Do not delete a CATEGORY until it has been deleted from all SECLABELs. When a category is deleted, the category is removed from SYSHIGH so that SYSHIGH no longer dominates any SECLABEL containing that category. The results are unpredictable.

- To change the SECLEVEL associated with a SECLABEL:
  - Define the new SECLEVEL:

    ```
    RALT SECDATA SECLEVEL ADDMEM(new_level_name/new_level_number)
    ```

  - Change each SECLABEL that uses the old_level_name:

    ```
    RALT SECLABEL label_name SECLEVEL(new_level_name)
    ```

  - Delete the old SECLEVEL:

    ```
    RALT SECDATA SECLEVEL DELMEM(old_level_name)
    ```

  - Refresh the SECLABEL class; otherwise the old definition is still used:

    ```
    SETROPTS RACLIST(SECLABEL) REFRESH
    ```

- To delete a SECLABEL:

  ```
  RDEL SECLABEL label_name
  ```

  Do not delete a SECLABEL profile without deleting or replacing its use in any profile, otherwise:
  - Users with this SECLABEL cannot access the system.
  - Resources protected with this SECLABEL cannot be accessed.
  - SYSOUTs protected with this SECLABEL cannot be printed or deleted; these should be printed or deleted before the SECLABEL is deleted.

The following commands can be used to determine what profiles are protected with a specific SECLABEL:

```
SEARCH CLASS(class_name) NOMASK SECLABEL(label_name)
```

For example, to change SECLABEL xx to yy or to delete SECLABEL xx from all user profiles, the following SEARCH and resulting CLIST commands can be executed:

```
SEARCH CLASS(USER) NOMASK SECLABEL(xx) CLIST(¢ALU ¢,¢ SECLABEL(yy)¢)

SEARCH CLASS(USER) NOMASK SECLABEL(xx) CLIST(¢ALU ¢,¢ NOSECLABEL¢)
```

3.4 Assigning SECLABELs

Once SECLABELs are defined, they can be assigned to users and resources. Jobs and TSO sessions execute with a SECLABEL that is the user's default SECLABEL, is assigned by the user, or has been propagated.

A user can assign a SECLABEL to a user or a resource profile if the user is authorized to define profiles in that class, or is authorized to change the specific profile and is also permitted to the SECLABEL. A user with the RACF SPECIAL attribute is allowed to assign any SECLABEL; however, SPECIAL users can log on only with SECLABELs they are authorized for or with SYSHIGH. The ability to assign a SECLABEL depends also on the setting of SECLABELCONTROL, MLSTABLE, and MLQUIET. For more information, refer to 3.5, "Controlling SECLABELs" on page 24.

3.4.1 User SECLABELs

If the SECLABEL class is active, users should be assigned default SECLABELs because users who do not specify a SECLABEL explicitly run without a SECLABEL and are not allowed to access any resources protected by a SECLABEL. Each user can be assigned two default SECLABELs with either of the following commands:

```
ADDUSER  userid...  SECLABEL(label_name)  TSO(SECLABEL(label_name))

ALTUSER  userid...  SECLABEL(label_name)  TSO(SECLABEL(label_name))
```

The userid or the group to which the user is currently connected must be authorized to use the SECLABEL. The following command can be used to authorize a user to a SECLABEL:

```
PERMIT  label_name  CLASS(SECLABEL)  ACCESS(READ)  ID(user or group)
```

Even though a user may be permitted to more than one SECLABEL, most users should be permitted to only one SECLABEL; the one defined in the base segment of the user profile. If a user is permitted to more than one SECLABEL, it is sometimes necessary to find out which SECLABEL is currently being used. One of the following methods can be used:

- The SECLABEL field of the STATUS SDSF panel shows the current job's SECLABEL or, for output, the SECLABEL of the user when the output was produced.
- RACF provides a new built-in REXX RACVAR function to run on MVS. The RACVAR function has the following format:

  ```
  X = RACVAR(¢argument¢)
  ```

  The argument is one of the following: USERID, GROUPID, SECLABEL, or ACEESTAT. If SECLABEL is used as an argument, this function returns the current SECLABEL in X. For additional information refer to 13.8, "TSO RACVAR Function" on page 253.

The predefined SECLABELs SYSHIGH, SYSLOW, and SYSNONE can be assigned to users. As with any other SECLABEL, the user must be permitted to use it. A user with the system SPECIAL attribute is allowed to use SYSHIGH without permission. The assignment of the SYSNONE SECLABEL to a user has the same effect as assigning SYSLOW.

3.4.2 Resource SECLABELs

A SECLABEL may be assigned to a profile with any of the following commands:

```
RDEFINE   class_name    profile_name   SECLABEL(label_name)  ...
RALT      class_name    profile_name   SECLABEL(label_name)  ...

ADDSD     profile_name  SECLABEL(label_name)  ...
ALTDSD    profile_name  SECLABEL(label_name)  ...
```

Only one SECLABEL can be defined in a resource profile, and it must be a validly defined SECLABEL or a predefined SECLABEL: SYSHIGH, SYSLOW, or SYSNONE. If a SECLABEL of SYSNONE is assigned to a resource, it is treated as being equal to any SECLABEL it is compared to; it automatically passes all SECLABEL checking.

3.4.3 Job or Session SECLABELs

Even though a user can be permitted to more than one SECLABEL, a job or session can have only one SECLABEL. If the SECLABEL class is active, the SECLABEL is determined as follows:

- For a TSO session, if the SECLABEL class is active, the TSO screen in Figure 9 is displayed with the new SECLABEL field. If there is a default SECLABEL in the TSO segment of a user profile, it is displayed in the SECLABEL field; otherwise, the field is displayed as blanks. The user can log on with the displayed SECLABEL or can enter in the SECLABEL field any SECLABEL that he is permitted to with at least READ access. The session is established with the displayed SECLABEL and that SECLABEL is saved in the user's TSO segment as the default SECLABEL to be displayed on the user's next logon screen. If no TSO segment is defined for the userid in the RACF profile, the information is not saved in the SYS1.UADS data set and, therefore, is not displayed on subsequent TSO logon panels.

  If no SECLABEL is displayed, none is entered by the user, or the displayed SECLABEL is blanked by the user, and if a default SECLABEL is defined in the base segment, the default SECLABEL is not displayed, not saved in the user's TSO segment, but is used to establish the session. If neither of the defaults is defined, and MLACTIVE is not active, the session is established without a SECLABEL; if MLACTIVE is active, a message is displayed that a default SECLABEL could not be obtained and the session cannot be established.

  If the user's base segment contains a default, the user cannot establish a TSO session without a SECLABEL; even if the user blanks out the TSO SECLABEL field, the SECLABEL in the base segment is used to establish the session. If the user's TSO segment contains a default and his base segment does not, and if MLACTIVE is not active, the user can decide to log on without a SECLABEL by blanking the SECLABEL field.

  To change the SECLABEL currently assigned to a TSO session, it is necessary to log off and log on again, specifying the new SECLABEL on the logon screen.

- For a batch job, a SECLABEL can be specified on the job card. In order for it to be used, the submittor must be permitted to the SECLABEL with at least READ access and either his current SECLABEL must dominate the SECLABEL of the job or the job's SECLABEL must dominate his. This is to prevent data from crossing category boundaries. If MLS is in effect, the job's SECLABEL must dominate the submittor's SECLABEL. The following example shows the JES JCL that may be used to change the SECLABEL under which a job runs:

  ```
  //USER12A1 JOB (acc-info),BRUCE,CLASS=A,MSGCLASS=X,SECLABEL=SAL,NOTIFY=USER12
  //STEP0A  EXEC PGM=.....
  //FTSIN   DD   DSN=.....
      .
      .
  ```

  USER12 has logged on with the SECLABEL of CTR. One of this user's responsibilities is to run a job for a group of users with a SECLABEL of SAL. For these users to be able to view the output of this job, USER12A1, USER12 must specify the parameter SECLABEL=SAL on the job card. Refer to Figure 5 on page 12 for the SECLABEL structure used here.

- When a user submits a job from a TSO session with neither a USER= or SECLABEL= on the job card, both can be propagated from his current TSO session. If the session was established without a SECLABEL, MLACTIVE cannot be in effect. The job is executed with the default SECLABEL from the user's base segment if one was defined; otherwise, the job executes without a SECLABEL.

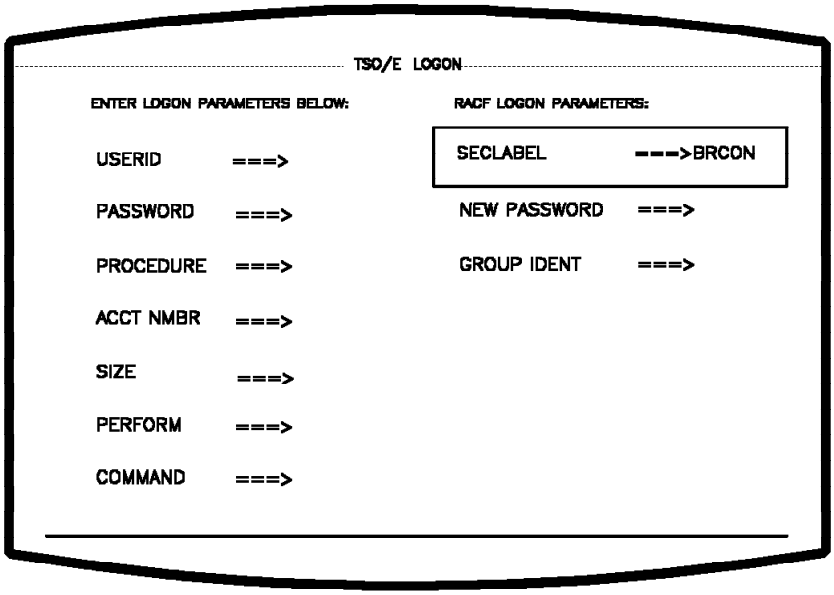

Figure 9. TSO Logon Screen

- When a user submits a job from his TSO session with a USER= on the job card, it is a SURROGAT submission. A SECLABEL can be specified on the job card; if not, it is taken from the base segment of the USER= that is coded on the job card. In order for any SECLABEL to be used, both the submittor and the USER= must be permitted to the SECLABEL of the job with at least READ access and either the submittor's SECLABEL must dominate the SECLABEL of the job or the job's SECLABEL must dominate his. This is to prevent data from crossing category boundaries. If MLS is in effect, the job's SECLABEL must dominate the submittor's SECLABEL. If there is no default SECLABEL in the user's base segment and MLACTIVE is not in effect, the job executes without a SECLABEL. If MLACTIVE is in effect, the job fails with insufficient security label authority and the output is labelled SYSLOW.
- When a job comes from an external card reader or RJE workstation, there must be a USER= on the job card. A SECLABEL can be specified on the job card, but if one is not specified, the SECLABEL is taken from the base segment of the USER= that is coded on the job card. If there is no default SECLABEL in the user's base segment and MLACTIVE is not in effect, the job executes without a SECLABEL. If MLACTIVE is in effect, the job fails with insufficient security label authority and the output is labelled SYSLOW.

3.4.4 SECLABEL Propagation and Translation

With SAF and JES 3.1.3, the userid and the SECLABEL are propagated when a job is submitted through the internal reader. The propagation follows the same rules that existed in the previous RACF and JES releases except that the userid and the SECLABEL may also be propagated for jobs submitted from another node. The receiving node propagates the submittor's userid and SECLABEL (from the submitting node) to the job. The receiving node may also translate the userid and the SECLABEL to a userid and a SECLABEL known to this system. In any case, the translated or propagated userid must be permitted to the translated or propagated SECLABEL at the receiving node. The SECLABEL is not translated or propagated when explicitly coded in the job card. The propagation and translation of a userid or a SECLABEL occurs only for nodes defined as trusted. For more information, refer to Chapter 8, "NJE Security Control" on page 117.

3.5 Controlling SECLABELs

Once an installation has protected all resources and has activated the SECLABEL class, it may wish to further limit who can define, change, or assign SECLABELs. Three SETROPTS options are available:

- When **SECLABELCONTROL** is in effect:
  - Only a system SPECIAL user is allowed to assign, change, or delete a SECLABEL field in a resource or user profile, or define or change the definition of a SECLEVEL, CATEGORY, or SECLABEL profile.
  - A group-SPECIAL user can add or modify the SECLABEL in a user profile within his scope of control. Also, group-SPECIAL users must be permitted to the SECLABEL profiles with at least READ access authority.
  - Users without the SPECIAL attribute cannot specify the SECLABEL operand.

  With NOSECLABELCONTROL, any user that has ALTER access to the profile and READ access to the SECLABEL being assigned can specify SECLABEL on RACF commands.
- With **MLSTABLE** active, no one is allowed to change the SECLABEL assigned to any resource or the definition of the SECLABEL itself, unless the system is in tranquil state. The tranquil state is achieved with MLQUIET and means that a SECLABEL can be changed only when it is not being used. Otherwise, there is a potential security exposure.
- With **MLQUIET** active, the system is in a tranquil state. In this state, only started procedures, console operators, or users with the system SPECIAL attribute are allowed to log on, start new jobs, or access resources.

  When this option is active, it prevents any action that requires the use of RACINIT, RACHECK, or RACDEF macros, unless the action is invoked by a system SPECIAL user, a started procedure, or a console operator. However, it does not prevent access to a resource to which access was already granted by the RACHECK macro.

  If MLQUIET is issued while jobs are running or while users are logged on, the failure of subsequent RACHECK, RACINIT or RACDEF calls may cause them to abend. MLQUIET should be issued only after all users, other than the security administrator or the console operators, have logged off and all jobs have completed. Since neither ADDUSER nor DELUSER commands can alter the SECLABEL of a resource that is being used, these commands can be executed regardless of the current setting of the MLQUIET option. Similarly, the ALTUSER is executed because it can change only the user's default SECLABEL and not the one currently being used.

3.6 Multi-Level Security

A multi-level security system contains information with different sensitivities and permits simultaneous access to that information by users with different security clearances, but prevents users from obtaining access to information for which they lack authorization. Before the implementation of multi-level security, installations supported different security clearances with physical isolation to prevent unauthorized access to sensitive data. Figure 5 on page 12 illustrates how SECLABELs can be designed and defined; however, it shows neither the effect of the dominant nature of security labels for access authorization nor the different levels of MAC checking that RACF 1.9 provides:

- SECLABEL class active
- MLACTIVE active
- MLS active

These also represent typical migration steps to achieve a desired level of MAC protection.

3.6.1 SECLABEL Class Active

When the SECLABEL class is active and security levels, categories, and labels are defined, the most basic level of MAC checking is in effect. An installation can selectively assign SECLABELs to resources that are considered sensitive and to the users that must have access to those resources. Resources that are not assigned SECLABELs can be accessed by any user through the normal DAC rules; users that are not assigned a SECLABEL can access only those resources that do not have SECLABELs through the normal DAC rules.

Figure 10 shows that MAC covers only part of the existing resources. Users with SECLABEL SAL pass MAC checking for resources protected by SECLABELs reflecting categories Personnel and Finance at the levels Confidential and Internal, such as PC or FI. After passing the SECLABEL check, users are still subject to access rules based on DAC permission to access the resource.

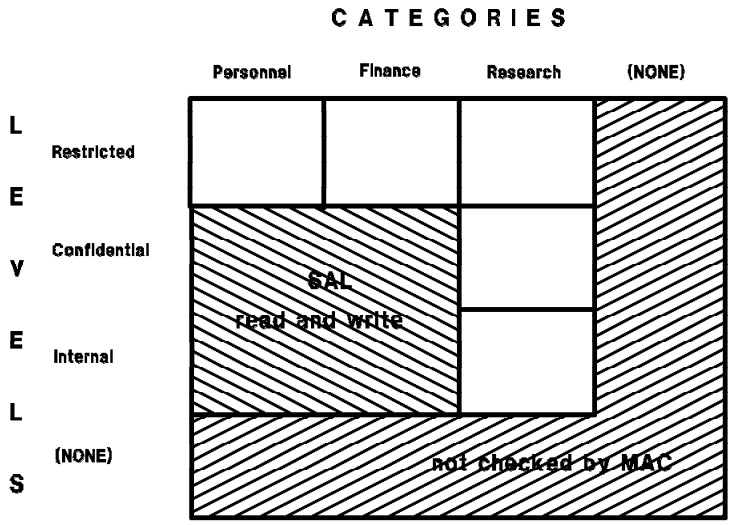

Figure 10. SECLABEL Class Active

Activating the SECLABEL class is a good first step toward implementing MAC security. However, similar to DAC protection without PROTECTALL, it has the potential disadvantage that unclassified resources may exist which, by policy, should be classified. In a B1 environment, the SECLABEL class must be active.

3.6.2 MLACTIVE Active

When the SECLABEL class and MLACTIVE are both active, the next level of MAC checking is in effect. All users, all jobs entering the system, all data sets, and all profiles defined in each active class with the SLBLREQ=YES option in the class descriptor table are required to have a SECLABEL. Any resource in a class that requires SECLABELs that is protected by a profile without a SECLABEL cannot be accessed. Resources that are not RACF protected can still be accessed. A side-effect of MLACTIVE is that it requires all users to be defined to RACF; since all users entering the system are required to have a SECLABEL, and users are required to be permitted to any SECLABEL they use, all users must be RACF defined. The following IBM defined classes have the SLBLREQ=YES attribute:

- DEVICES - TAPEVOL - TERMINAL - WRITER

MLACTIVE affects these classes only if the class and the SECLABEL class are active. MLACTIVE is activated with the following command:

```
SETROPTS MLACTIVE (WARNING | FAILURES) | NOMLACTIVE
```

Figure 11 is an example of access for a resource class that is defined with SLBLREQ. The security level Unrestricted and the category Other is used to label resources that were not covered by MAC in the previous example. Users with SECLABEL SAL pass MAC checking for resources protected by SECLABELs reflecting categories Personnel and Finance at the levels Confidential, Internal, and Unrestricted. After passing the SECLABEL check, users are still subject to access rules based on DAC permission to access the resource. Before activating MLACTIVE with the FAILURES option, ensure that:

- All users are defined to RACF with a SECLABEL.
- All resources in the classes listed above are protected with a profile and a SECLABEL.

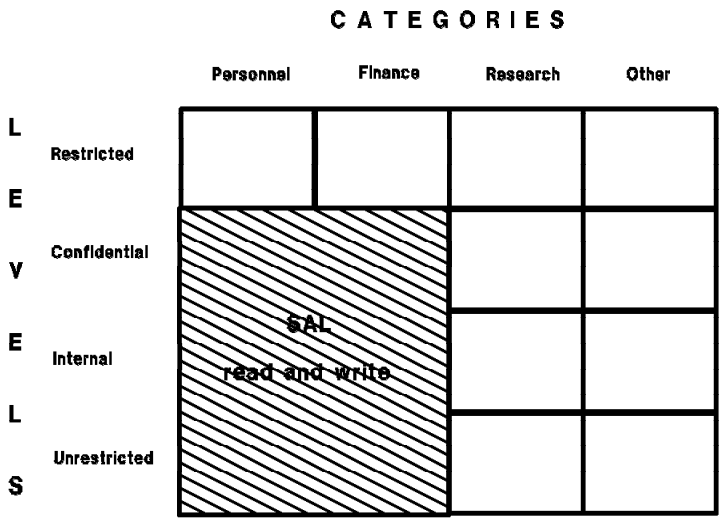

Figure 11. MLACTIVE Active

When MLACTIVE is in FAILURES mode and the SECLABEL class is active, the following applies to profiles in the active classes listed above:

- A trusted or privileged started procedure can always access the resource.
- A program defined with NOPASS in the PPT can access any data set.
- Access is allowed if the global access checking table grants the access.
- Users without a SECLABEL are not allowed to access the system. Only a system SPECIAL user can access the system without a SECLABEL, but is not allowed to access any resource protected by a SECLABEL.
- No user can access, rename, or delete data sets that are either unprotected by a RACF profile or are protected by a RACF profile without a SECLABEL.
- Temporary data sets must be protected with the activation of the TEMPDSN class; otherwise, the SCRATCH function for temporary data sets fails.
- When the WRITER class is active, a profile with a SECLABEL must exist for each printer, NJE node, and RJE device; otherwise, the system cannot print SYSOUT or send SYSOUT to another node or RJE device.
- If a terminal is protected by a profile without a SECLABEL, only a system SPECIAL user can access it. If a terminal is not protected, anyone can enter the system through this terminal.

- Access to protected tape volumes or unit record devices without a SECLABEL is denied. If they are not protected, access is granted.

When MLACTIVE is activated in WARNING mode, the system sends a warning message for any condition previously described, but allows access. However, MLACTIVE should be activated in WARNING mode only when most of the resources protected by those classes are correctly protected. If not, console message buffers may be easily filled with warning messages.

MLACTIVE in WARNING or FAILURES mode also has the following effects on the resource classes listed above:

- A SECLABEL profile field can only be changed, never nullified.
- Any new profile defined must have a SECLABEL. If a SECLABEL is not assigned, the system assigns, as the default, the current SECLABEL of the user who is defining the profile and no warning message is issued.

MLACTIVE is a means to enforce security policy in commercial environments. It ensures that unclassified data cannot be accessed. However, it does not address concerns of exposures through inadvertent declassification. Such concerns are more typical in military and similar environments. In a B1 environment, MLACTIVE must be active.

3.6.3 MLS Active

The maximum level of MAC checking is in effect when the SECLABEL class is active, MLACTIVE is active, and MLS is active. MLS active does not require MLACTIVE to be active, but does require that the SECLABEL class be active. When the MLS option is active, the confinement or **-property* is enforced for all resources in all classes if the profile has a SECLABEL and the requested access is higher than READ. MLS can be activated with the following command:

```
SETROPTS MLS (WARNING|FAILURES) | NOMLS
```

The MAC rule is that a user can access, for read or write, only those resources that are dominated by the user. The *-property adds the rule that a user can have write access to only those resources that dominate the user. This property prevents the declassification of resources by enforcing the following access rules:

- **Read Only.** A user can read a resource if the user's SECLABEL dominates the resource's SECLABEL.
- **Write only.** A user can write to a resource if the resource's SECLABEL dominates the user's SECLABEL.
- **Read/Write.** A user can read and write to a resource only if the SECLABEL of the user and the SECLABEL of the resource are equal.

The terms READ and WRITE may be somewhat misleading since they seem to suggest data set operations to many people. It should be noted that the MVS architecture does not support write-only operations (UPDATE access always includes READ) and that therefore the above mentioned write-up concept is not applicable to MVS data sets. Write-up applies to and makes sense for resources such as messages transmitted to other users and output routed to printers.

With SETROPTS MLS, access granted by SECLABELs changes dramatically. Figure 12 on page 28 shows that users with the same SECLABEL of SAL experience different access capabilities, as follows:

- They pass MAC checking for READ and WRITE requests for resources protected by SECLABELs reflecting categories Personnel and Finance at the Confidential level, as before.

- For SECLABELs reflecting levels Internal and Unrestricted within the same categories, only READ requests are permissible since WRITE operations would potentially declassify information.
- For resources protected by SECLABELs reflecting the Restricted level within the same categories, WRITE requests are now allowed. READ from a higher classification would represent a declassification.

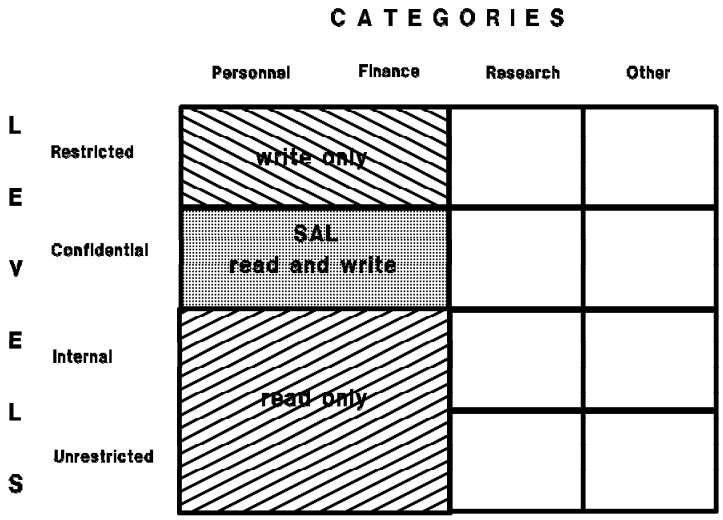

Figure 12. MLS Active

In WARNING mode, a message is issued for any unauthorized access, but the access is granted. MLS checking does not apply under the following conditions:

- The SECLABEL class is not active.
- The resource class is not active.
- When a started procedure is defined as trusted or privileged. For data set checking, if the program is defined with NOPASS in the PPT.
- Access to the resource is granted by a profile in the global access checking table.
- The resource is not protected by a RACF profile.

3.6.3.1 MLS Effects on the DATASET Class

When MLS is active, a user can:

- Read a data set if the user′s SECLABEL dominates the data set′s SECLABEL.
- Write to a data set if the data set′s SECLABEL is equal to the user′s SECLABEL. When MVS opens a data set, it is opened for update, never for write-only.

The rationale is to prevent declassification of information by preventing a user from writing to a less classified data set. Data sets that must be updated by every user and contain no relevant security information, such as catalogs, should be assigned a SECLABEL of SYSNONE. A data set SECLABEL of SYSNONE has the same effect as being equal to the user′s SECLABEL; therefore allowing the update of the data set (assuming DAC is passed). Another suggestion is to include these data sets in the global access checking table. For a list of recommended SECLABELs for system data sets, refer to the *RACF Security Administrators Guide*.

MLS is a means of implementing a military security policy and therefore is a requirement in a B1 environment. These additional controls may not be required for the implementation of many commercial security policies; they may not even be compatible with them. Typical commercial installations may reject the write-up concept for integrity reasons, and the concept of read-only access to lower classifications may raise usability and administration concerns. Refer to *MVS/ESA Planning: B1 Security* for more information.

3.6.3.2 MLS Effects on Job Submissions

With NOMLS, the submittor must be permitted to the job′s SECLABEL and either the submittor′s SECLABEL has to dominate the job′s or the job′s SECLABEL has to dominate the submittor′s. This is to prevent the crossing of category boundaries. However, when MLS is active, the submittor must still be permitted to the job′s SECLABEL, but the SECLABEL of the job must dominate the submittor′s SECLABEL. This is to avoid the following sequence of events:

- USERA submits a job on behalf of USERB.
- USERA has a higher SECLABEL than USERB.
- USERA includes SYSIN information at USERA′s SECLABEL.
- The job is then executed by USERB.
- USERB is allowed to see the SYSOUT which may contain information from the SYSIN data set and is, therefore, outside the scope of USERB′s SECLABEL.

This applies to the user′s own jobs as well as any SURROGAT submissions. A SECLABEL of SYSNONE assigned to a user has the effect of SYSLOW and, therefore, does not equate the submittor′s SECLABEL to the job owner′s SECLABEL.

3.6.3.3 MLS Effects on the TAPEVOL Class

When MLS is active, a user can write to a tape only if the SECLABEL is dominated by the SECLABEL of the tape volume profile. In addition, RACF ensures that all data written to the tape is protected by the SECLABEL specified in the tape volume profile.

If NOMLS, it is the user′s responsibility to ensure that each data set has the same security label as the one specified in the tape volume profile.

3.6.3.4 MLS Effects on the JESSPOOL Class

When a user allocates a SYSOUT data set, it is protected with the user token. The SECLABEL of the user and the SECLABEL of the SYSOUT data set are the same and the user is always able to write to the SYSOUT data set.

However, another task executing in the same address space with a different SECLABEL can attempt to write to the same SYSOUT data set. With NOMLS, the write is allowed if the task′s SECLABEL dominates the SYSOUT data set′s SECLABEL. With MLS active, the write is allowed only if the task′s SECLABEL and the shared SYSOUT data set′s SECLABEL are equal.

3.6.4 Security Classification Summary

MAC through SECLABELs is a powerful and important control that complements the existing DAC concept in RACF. It is recommended that SECLABELs be used as an initial step toward implementing MLACTIVE, an excellent way to enforce commercial security policies. Installations that have to comply with DoD B1 requirements must also implement MLS. Table 2 shows the effect on the system of each combination of SECLABEL class, MLS, MLACTIVE, and MLQUIET.

Table 2. Summary of Multi-level Security Options				
SECLABEL	**MLS**	**MLACTIVE**	**MLQUIET**	**Effect**
Active	Off	Off	Off	Use SECLABEL class
Inactive	Off	Off	Off	Use SECDATA class if active
Active	On	Off	Off	Enforce *-property (No writedown)
Inactive	On	Off	Off	NO effect when SECLABEL inactive
Active	Off	On	Off	Requires security labels for secure resources and users
Inactive	Off	On	Off	No effect when SECLABEL inactive
Active	On	On	Off	Enforce *-property and all resources labeled
Inactive	On	On	Off	No effect when SECLABEL inactive
Either	Either	Either	On	Fail except for TCB

3.7 Automatic Data Set Protection and Modeling Options

This section describes how new the RACF functions and features relate to automatic data set protection (ADSP) and modeling. To put this approach in perspective, it contrasts these different concepts to achieve automatic and complete data set protection in RACF:

- ADSP in combination with modeling, the traditional approach based on discrete RACF profiles.
- Generic profiles (on systems with Always Call), the contemporary approach.

Theoretically, there is no difference in the level of protection that can be achieved using either ADSP with discrete profiles or NOADSP and PROTECTALL with generic profiles.

3.7.1 Automatic Data Set Protection

Early RACF releases used only discrete profiles to protect individual resources; that is, 100 data sets are protected by 100 discrete RACF profiles. For the DATASET class, a discrete profile is checked only in conjunction with a corresponding RACF indicator; namely a VTOC bit indicating that a particular data set is protected by RACF.

ADSP is a facility to automate RACF protection by automatically creating a discrete RACF profile when a new data set is allocated by a user with the ADSP attribute, provided the system-wide SETROPTS ADSP option is active. This approach works well for most user data sets because they are often used exclusively by that individual user, who can access it based on naming conventions (userid is equal to the data set HLQ). For group data sets, however, usability problems may exist. The creating user is put on the access list of the automatically created profile with ALTER authority, and if the GRPACC attribute is also in effect, the group name is placed on the access list with UPDATE authority. In many cases, the access list does not reflect the authorization desired or needed.

3.7.2 Modeling

The RACF modeling facility solves the above problem. It allows model profiles to be defined for user or group data sets, which are used as a base for new discrete or generic profiles. This approach results in profiles with desired access lists and other characteristics. Modeling is active when the SETROPTS MODEL option is selected and model profiles have been defined for the user or group.

3.7.3 Generic Profiles

RACF 1.5 introduced generic profiles. Generic profiles protect more than one resource (based on similarities in the resource name) with like protection requirements. For generic profile checking to be effective, it has to be invoked for all authorization decisions. This means, for the DATASET class, that it must be called independent of the presence of a RACF indicator. The MVS/DFP function providing this service is Always Call, a standard feature of current MVS/DFP versions and releases. It cannot be turned off. Generic profiles typically contain generic characters (%, *, or **) in the resource name. If several profiles cover a resource, the most specific one is used to determine access rights. Because there can be significantly fewer generic profiles than discrete profiles, there are several advantages to using generics:

- Generic profiles make RACF administration much easier and RACF more usable. When adding or changing information in profiles, generics minimize the number of profiles to be changed. There are also fewer problems synchronizing profiles and resources after restore operations.
- The smaller number of generic profiles needed allows smaller RACF data sets and more efficient resident options. Performance can be dramatically better when using generic profiles.
- For the system auditor, an advantage of discrete profiles was that a list of all profiles produced the names of all protected resources, while a list of generic profiles did not. In both cases, unprotected data sets might exist and be accessible unless PROTECTALL was in effect. The CATDSN option introduced in RACF 1.9 provides the ability to list the data sets protected by a generic profile as well as generate type 83 SMF records containing the list of data set names affected by a SECLABEL change, giving the advantage now to generics.

Installations implementing B1 level security should follow the guidelines provided in *MVS/ESA Planning: B1 Security*; however, these guidelines do not address RACF profile types. It is therefore the installation's choice whether to use discrete profiles, generic profiles, models, or any combination thereof.

3.7.4 ADSP and Modeling with SECLABELs

SECLABEL information is never copied from the model profile. If MLACTIVE is not in effect, and a SECLABEL is not specified, the profile does not have a SECLABEL. However, if MLACTIVE is in effect and a SECLABEL is not specified, the current SECLABEL with which the job or session is executing becomes the SECLABEL of the new discrete or generic data set profile. This may or may not be what is needed to protect the new data set.

3.8 Recommendations

The use of ADSP is not recommended. PROTECTALL should be used with generic profiles whenever possible. ADSP is active by default and, unless required, it should be deactivated. Discrete profiles, used as the exception, can be created using RACF commands. Generic profile checking is inactive by default. The following command deactivates ADSP and activates PROTECTALL and both generic profile checking and generic command processing for all defined classes:

```
SETROPTS NOADSP PROTECTALL(WARNING) GENCMD (*) GENERIC(*)
```

Note: PROTECTALL should be activated initially in WARNING mode, but ensure that a plan is in place and a deadline established for implementing it in FAILURES mode.

When implementing SECLABELs, consider the following recommendations:

- Security labels provide advantages over security levels and categories. They are easier to maintain, protect data without a resource profile (spool data, PSF/MVS overlays), and a user can log on with different security labels. Before activating the SECLABEL class, carefully read the sections discussing security label considerations in the *RACF Security Administrators Guide*.
- When the SECLABEL class is active, define for each user in the environment a default SECLABEL in the base segment of the user's profile. This action avoids problems if a user enters the system without a SECLABEL, and is not able to access resources protected by a SECLABEL.
- Permit most users to have only one SECLABEL; exceptions are system SPECIAL and group-SPECIAL users.
- When protecting a resource with a SECLABEL, assign an appropriate SECLABEL to all users who have to access the resource.
- When the SECLABEL class is active, any changes to a SECLABEL definition should be made in a very well controlled environment with no jobs running and no users logged on except the security administrator. This means that both SECLABELCONTROL and MLSTABLE should be active.
- When adding a SECLEVEL, be aware that the value of the SYSHIGH or SYSLOW SECLABELs could be affected.

If the SECLABEL class is active, RACF uses the SECLABEL in the profile for all authorization checking, ignoring any SECLEVEL and CATEGORY data in the profile. However, the SECLEVEL data in the profile is still used for:

- Erase-on-scratch when requested for a specific level.
- SECLEVELAUDIT

Chapter 4. RACF 1.9 Enhancements

MVS/SP Version 3 Release 1.3, together with RACF Version 1 Release 9, enhances enterprise security over a broader variety of functions and features than previous releases. The enhancements described in this chapter are those made to the RACF product and RACF functions.

4.1 New IBM-Defined RACF Classes

A significant number of new IBM-defined RACF classes have been added in RACF 1.9 and existing resource classes provide additional functions. Table 3 lists the new functions and the classes that are used to implement the functions.

Table 3. New RACF Classes

Function	**Classes Used to Implement**
JES/Control over Jobs	JESINPUT, JESJOBS, JESSPOOL, SDSF, SURROGAT, WRITER
VTAM Authorization	APPCLU, DIRAUTH, VTAMAPPL
Operator Command Control	CONSOLE, OPERCMDS
Mandatory Access Control (MAC)	PSFMPL, SECLABEL
NJE Control over Jobs and Output	NODES, WRITER
Device Allocation Control	DEVICES
Hiperbatch Control	DLFCLASS
RACF Administration Enhancement	RACFVARS
Temporary Data Set Control	TEMPDSN
TSO Message Control	SMESSAGE, DIRAUTH
Netview/Access	NVASAPDT

4.2 New Uses of the FACILITY Class

The FACILITY class now provides these additional functions:

- Control of RJE/NJE sign ons
- CATDSNS exceptions
- LLA control
- Batch LSR control

4.3 New RACF Profile Segments

RACF is more than a security product, it is a repository for information used by other products. Table 4 lists profile segments that are already in use and new segments introduced with RACF 1.9.

Table 4. New RACF Profile Segments

RACF Release	Segment Name	Profile Name	Supported Product	Segment Contents
1.8.0	TSO	USER	TSO	TSO default information for the logon panel.
1.8.1	DFP	USER or GROUP	DFSMS	Data application, data class, management class, and storage class defaults for the ACS routines.
1.8.1	DFP	DATASET	DFSMS	RESOWNER for data sets protected by this profile.
1.9	SESSION	APPCLU	VTAM	Session key, key life, session entity name, and invalid attempt counts for LU 6.2 sessions.
1.9	DLFDATA	DLFCLASS	DLF	RETAIN indicator and authorized jobnames for Hiperbatch objects.
1.9	CICS	USER	CICS	Operator ID, operator classes, terminal timeout value, operator priority, XRF resignon option.

4.4 GENERIC Operand on RLIST

The RLIST command now allows the GENERIC operand to be specified as follows:

```
RLIST ... GENERIC | NOGENERIC
```

Its function is similar to the GENERIC operand on the LISTDSD command, but improved. If the user specifies * as the resource name (requesting a list of all defined profiles) and specifies:

- GENERIC, all generic profiles are listed; discrete profiles are ignored.
- NOGENERIC, all discrete profiles are listed; generic profiles are ignored.
- Neither, all profiles are listed.

If the user specifies a particular resource or profile name, and specifies:

- GENERIC, the best-fit generic profile is listed; discrete profiles are ignored.
- NOGENERIC, the discrete profile is listed; generic profiles are ignored.
- Neither, the discrete profile is listed if one exists; otherwise, the best-fit generic profile is listed. This differs from the LISTDSD command, which defaults to discrete if neither GENERIC nor NOGENERIC is requested.

4.5 Trusted Programs and Procedures

MVS has always provided facilities to assign properties to programs or address spaces that enable them to bypass security checking and security logging. These facilities are used by IBM and are also available for installation use. Typical reasons for assigning such properties are startup considerations (a component might be coming up prior to the complete initialization of RACF), performance considerations (avoid checking of frequent system actions), or ease of use (save the trouble of defining proper access rules). Trusted procedures, a new concept introduced with RACF 1.9, allows RACF checking to be bypassed while still offering the option of logging such events.

4.5.1 Programming Properties Table

The programming properties table (PPT) is an MVS control block that assigns special properties to APF authorized programs. Three PPT properties are of concern for security and auditing:

- The bypass password protection (NOPASS) indicator, which bypasses RACF checking for data sets.
- The data set integrity bypass indicator (DSI), which allows a program to allocate a data set even if it is held by another address space with DISP=OLD.
- The assignment of a system storage protection key (KEY(0-7)).

While all these properties are critical, this discussion focusses on the PPT NOPASS indicator because it is the one that RACF 1.9 addresses with the creation of trusted procedures. This bit is propagated to the JFCB NOPASS indicator, which is tested by OPEN. If present, OPEN bypasses SAF calls and grants access to any data set in the system.

The IBM default PPT contains a number of programs with the NOPASS option. An installation can change the IBM defaults and also add local entries to the PPT through SCHEDxx members in SYS1.PARMLIB. Good control over the PPT requires that write access to SYS1.PARMLIB and all APF libraries are tightly controlled.

4.5.2 Started Procedures Table

The started procedures table (SPT) is a RACF control table that assigns a userid and a group to a started procedure that can be used to determine RACF access authorities. This table is necessary because a started procedure does not have a job card where the user and group would normally be specified. The SPT can also assign the started procedure security bypass privileges. The PRIVILEGED indicator, available in earlier RACF releases, allows RACF access without logging for almost all RACHECKs. PRIVILEGED provides more privileges than the PPT NOPASS indicator since it grants access in all classes; NOPASS grants access to only data sets. RACF 1.9 introduces the SPT TRUSTED attribute that grants the same access as the PRIVILEGED attribute, but provides the ability to log the access. The logging is actually controlled by the SETROPTS LOGOPTIONS rather than by individual RACF profiles. Refer to 4.14.3, “LOGOPTIONS” on page 57 for more information. The PRIVILEGED attribute is still supported by RACF 1.9 and if a started procedure has both the PRIVILEGED and the TRUSTED attributes, the PRIVILEGED attribute overrides and disallows logging.

The IBM default SPT does not contain any entries. It must be set up by the installation by coding an Assembler language module and link-editing it as ICHRIN03 in SYS1.LPALIB or the equivalent. Good control over the SPT attributes requires that write access to all procedure libraries and all load libraries is well controlled.

4.5.3 TRUSTED Option of the RACROUTE Macro

Any address spaces that are verified with **RACROUTE REQUEST=VERIFYX,TRUSTED=YES** are also considered trusted procedures; they pass almost all RACHECKs and, depending on the SETROPTS LOGOPTIONS, can be audited.

RACROUTE VERIFY requests require APF authorization and are typically coded in resource managers and other system programs. To avoid integrity exposures, such programs must be written following the MVS integrity coding guidelines.

4.5.4 Trusted Procedures Summary

Table 5 summarizes the characteristics of the various techniques of bypassing RACF checking. Use of PPT NOPASS and SPT PRIVILEGED are not allowed in a B1 environment because they violate the requirement that all accesses to resources have to be able to be audited; only the trusted attribute allows auditing.

Table 5. Techniques for Bypassing RACF

Control Mechanism	Object	Classes Bypassed	Condition	Checked By	Changed By	Logging Possible	B1
PPT NOPASS	Program name	DATASET	APF authorized	MVS JSCBPASS	PARMLIB UPDATE	No	No
SPT Privileged	Procedure name	All	Started procedure	RACF	SPT assembly	No	No
SPT Trusted	Procedure name	All	Started procedure	RACF	SPT assembly	Yes	Yes
VERIFYX TRUSTED =YES	User name	All	Caller APF authorized	RACF	Resource manager code	Yes	Yes

4.5.5 Recommendations

The following recommendations are intended for a non-B1 security environment. They are designed for a well controlled commercial computing environment where no legal or contractual need for B1 level security exists.

- Bypassing security is not a good idea from a control point of view; however, IBM-supplied settings are probably required for MVS to start and execute properly. Installation defined bypass privileges should not be used without a thorough investigation of alternative solutions. The PPT NOPASS and the SPT PRIVILEGED attributes should be replaced with the new SPT TRUSTED attribute to allow possible logging with the LOGOPTIONS options. Sample PPT and SPT entries in the RACINSTL member of SYS1.SAMPLIB describe how to convert NOPASS to TRUSTED. Table 6 can also be used to aid in this conversion.
- For performance reasons, trusted procedures should not be audited under normal circumstances.
- The use of the RACF authorization table ICHAUTAB is not recommended.

Table 6. Converting PPT NOPASS to SPT TRUSTED		
PPT Entry to Override with SCHEDxx PASS	**Function**	**SPT Entry to Define as TRUSTED**
HASJES20	JES2	JES2 procname
IATINTK	JES3	JES3 procname
IATINTKF	JES3 FSS	JES3CI procname
n/a	PSF	APSWPROC procname
IEEMB860	Master	n/a
n/a	Mount command	IEEVMPCR
ISTINM01	VTAM	VTAM procname
CSVLLCRE	LLA	LLA procname
COFMINIT	VLF	VLF procname
IFASMF	SMF	SMF
IGDSSI01	Catalog services	n/a
IEAVTDSV	Dump Services	DUMPSRV

4.6 Class Descriptor Table Enhancements

The RACF class descriptor table (CDT) describes attributes of RACF general resource classes. RACF is, with the exception of user, group, and data set profiles, strictly table-driven. This technique allows new resource classes to be added as needed to support more resource managers and new applications, and also enables a RACF installation to define its own resource classes. RACF 1.9 does not remove the existing limitation on the number of resource classes an installation can add; the maximum remains 128.

4.6.1 Defining New Installation Classes

To define new installation classes to RACF, the following steps should be taken:

1. Define the class to RACF with an entry in the CDT. To achieve separation of IBM entries from user entries, the CDT is actually split into two parts: the IBM table (ICHRRCDX) and the installation table (ICHRRCDE). Use the ICHERCDE macro for each new class to create ICHRRCDE. The last entry in the table must be an ICHERCDE macro with no operands. In the linkage editor step, use the ORDER statement to ensure that the ICHERCDE macro with no parameters is last in the table; otherwise, any classes defined after the ICHERCDE with no operands are not known to RACF.
2. Define the class to the system authorization facility (SAF) with an entry in the SAF router table. Similar to the CDT, the router table also consists of two parts: the IBM table (ICHRFR0X) and the installation table (ICHRFR01). Use the ICHRFRTB macro for each new class to create ICHRFR01.
3. Activate authorization checking for the class using the SETROPTS command.

4.6.2 New CDT Parameters

Several additional parameters have been added to the CDT. In addition, the maximum length of resource names has been increased from 39 to 246 characters if the restructured database (RDB) format is used. The new parameters are:

- DFTRETC - PROFDEF - RACLREQ - RVRSMAC - SLBLREQ

DFTRETC = 0 | 4 | 8: Specifies the return code that RACF passes to the resource manager when no profile is found. If not specified, a return code of 4 is passed to the resource manager, which allows the resource manager to decide whether the access should be allowed or denied if there is no profile protecting the resource. The SMESSAGE class has a default return code of 0, which indicates that access should be allowed if no profile is found. The following IBM defined classes have the default return code of 8, which means that access should be denied if no profile is found:

- CONSOLE - JESINPUT - JESJOBS - JESSPOOL
- PSFMPL - SECLABEL - WRITER

By using the default return code of 8, an installation can implement PROTECTALL for a general resource class similar to the PROTECTALL available for the DATASET class. Whenever a class whose default return code is 8 is activated, ensure that at least one profile is defined or access to all resources in that class is denied. For example, if the WRITER class is activated prior to defining any profiles, no SYSOUT can be printed and no jobs or SYSOUT can be sent to a remote node. This new CDT option makes the use of SETROPTS CLASSACT(*) a very dangerous command.

In order to avoid most problems, define the most generic profile possible (for example, **) allowing anyone to access the resource before activating the class. Then define more specific profiles.

Installations implementing B1 level security should follow the guidelines provided in *MVS/ESA Planning: B1 Security*. The default return code for all installation-defined resource classes should be set to 8 (alternatively, default generic profiles with SECLABEL of SYSHIGH must be defined).

PROFDEF=YES | NO: Indicates whether profiles can be created in this class. The default is YES. The following IBM defined classes specify PROFDEF=NO:

- DIRAUTH - TEMPDSN

These classes are used for function activation and logging control. Since no profiles exist, the only way to log accesses for these classes is with the new SETROPTS LOGOPTIONS control. Refer to 4.14.3, “LOGOPTIONS” on page 57 for more information.

RACLREQ=YES | NO: Specifies whether profiles in this class have to be RACLISTed. If a profile is not found in storage and YES is specified in the class descriptor entry for the class, RACF does not issue an I/O request to the RACF data base to retrieve the profile. The return code passed back to the resource manager is the DFTRETC. The default is NO. If YES is specified, RACLIST=ALLOWED must also be specified. The following IBM defined classes specify RACLREQ=YES:

- DEVICES - NODES - OPERCMDS - PROPCNTL (old)
- PSFMPL - RACFVARS - SECLABEL - VTAMAPPL

Note: RACLREQ=NO does not mean that this class cannot be RACLISTed. Whether a class can be RACLISTed or not is determined by the RACLIST=ALLOWED | NOTALLOWED option.

Starting with RACF 1.9 and MVS/ESA, all RACLISTed profiles are kept in a data space. In previous releases they were in CSA.

RVRSMAC=YES | NO: Indicates whether MAC checking is reversed. For most classes, the user's SECLABEL must dominate the resource's SECLABEL in order for access to be allowed. For the classes that specify RVRSMAC = YES, the resource's SECLABEL must dominate the user's SECLABEL in order for access to be allowed. RVRSMAC has meaning only when the SECLABEL class is active. The default is NO. The following IBM defined classes specify RVRSMAC = YES:

- CONSOLE - TERMINAL - WRITER

Reverse checking prevents the declassification of information. For example, a terminal in an unsecured place, such as the lobby, should be protected with a low SECLABEL. A user with a higher SECLABEL and, therefore, able to display more classified information, is not allowed to use this terminal. Otherwise, he could display classified information that could be viewed by an unauthorized person in the lobby.

The WRITER class authorizes a user to print a SYSOUT data set on a specific printer. For example:

```
RDEFINE  WRITER  JES2.LOCAL.PRTxx  SECLABEL(SAL)  UACC(READ)
```

With this definition, only SYSOUT data sets with a SECLABEL dominated by the SAL SECLABEL are allowed to print on PRTxx.

SLBLREQ=YES | NO: Indicates whether profiles in this class must have a SECLABEL when the SECLABEL class is active and the MLACTIVE option is active. For classes that specify YES, only those resources protected by profiles with a valid SECLABEL can be accessed. The default is NO; however, even when NO is specified (SECLABELs are not required), if the SECLABEL class is active and the profile has a SECLABEL, normal SECLABEL checking occurs. The following IBM defined classes specify SLBLREQ=YES:

- DEVICES - TERMINAL - TAPEVOL - WRITER

When MLACTIVE is active, data sets are also required to have SECLABELs, even though there is no CDT entry for the DATASET class that specifies SLBLREQ=YES. Refer to 3.6.2, "MLACTIVE Active" on page 25 for more information.

For a list of the new RACF 1.9 classes, their profile names, and their new CDT parameter settings, see Appendix B, "RACF Resource Classes" on page 265.

4.6.3 Recommendations

The following recommendations are intended for a non-B1 security environment. They are designed for a well controlled commercial computing environment where no legal or contractual need for B1 level security exists. It is important, however, that the activation of RACF resource classes be carefully planned and implemented individually. A global activation of all classes at once can most certainly lead to disastrous results.

- The major new control enhancement is the ability to set the default return code for a profile-not-found condition. Earlier RACF releases passed a return code of 4 back to the caller, and the resource manager decided to grant or deny access. This led to inconsistencies (IMS grants access, CICS does not) which can now be eliminated through the setting of DFTRETC in the CDT definition. DFTRETC=8 should be used as a standard for all installation defined classes. A similar effect can be achieved using default generic profiles with UACC=NONE, but the CDT solution is less prone to errors and omissions and therefore more secure. Where standard IBM classes (such as TIMS) do not use this parameter, installation defined classes should be considered, provided the maximum number of classes has not yet been reached.
- Requirements for SECLABELs should be set for installation defined classes where appropriate.
- The CDT POSIT flag for audit should be turned on for installation defined classes, but statistics should not be collected. For generic profiles, the statistics option is ignored, but for discrete

profiles this option could cause a severe performance impact without providing much useful information.

- RACLIST should be allowed for all installation-defined classes. Also consider requiring RACLIST or permitting GENLIST if the number of profiles in a class, their use, or maintenance characteristics justify it.
- The RACLREQ option should be used only with DEFTRETC=8 to avoid security exposures in case the RACLIST was not actually issued.

4.7 Resource Name Enhancements

The following resource name enhancements have been made to RACF 1.9:

- A new RACF class, RACFVARS, has been added to ease the definition of general resource profiles. Qualifiers in a profile name can be a variable defined in the RACFVARS class.
- Enhanced generic naming (EGN) for data sets, introduced in RACF 1.8.1, has been further enhanced in RACF 1.9.
- EGN has been extended to general resources.

4.7.1 RACFVARS Class

With RACF 1.9, qualifiers of general resource profile names can be variables; variables cannot be used in data set profile names. Profile names containing variables are considered generic profiles. Several values can be assigned to each variable through a profile in the RACFVARS class. This new facility allows one general resource profile to protect many resources with unlike names, reducing the number of profiles that are needed. Generic profiles should still be used for similar resource names; resource group classes should still be used when available (GTERMINL, GDASDVOL).

Note: A resource group class is effective only when RACLIST processing is active for the corresponding resource member class.

Variable names must begin with an ampersand (&), can be up to eight characters long, and cannot contain any generic characters or a period (.). Names starting with &RAC are reserved for RACF use and should not be used. The resource names assigned to the variable are added by the ADDMEM operand to the RACFVARS profile. They can be up to 39 characters long and cannot contain generic characters. All the resources must be in the same class and come from a class that accepts generic profile names.

RACLIST processing helps ensure high performance when accessing RACF profiles by loading the profiles into storage. RACLIST is required for the RACFVARS class because the class descriptor table specifies RACLREQ=YES for RACFVARS. The following command is used to activate and RACLIST the RACFVARS class:

```
SETROPTS  CLASSACT(RACFVARS)  RACLIST(RACFVARS)
```

Any time a change is made to a RACFVARS profile, the in-storage profiles have to be refreshed for the RACFVARS class before the changes take effect, for example:

```
SETROPTS  RACLIST(RACFVARS)  REFRESH
```

The following examples show the use of RACFVARS:

- **Example 1.** This example creates a profile in the TAPEVOL class to protect several tape volumes. This approach can be used if the tape volume naming standards do not allow the creation of a generic profile to protect them. There is no resource group class for TAPEVOL. The real

advantage of this approach is that only one access list is changed when maintaining authorization to these tape volumes.

A profile in the RACFVARS class is used to assign several values to TAPEVOL profiles, which consist of only one qualifier. First, define a profile in RACFVARS and add values to it as members to that profile. The profile name must begin with an ampersand. The following profile covers tape volumes T11111 and T22222:

```
RDEFINE RACFVARS &TAP35 UACC(NONE) ADDMEM(T11111 T22222)
```

Then define the actual protecting profile in class TAPEVOL and permit groups or users to it:

```
RDEFINE  TAPEVOL  &TAP35  UACC(NONE)

PERMIT   &TAP35  CLASS(TAPEVOL)  ID(GSMVS)  ACCESS(ALTER)
```

Activate the TAPEVOL and RACFVARS classes and enable RACLIST processing for the RACFVARS class:

```
SETROPTS CLASSACT(TAPEVOL RACFVARS) RACLIST(RACFVARS)
```

If tape volume T33333 becomes one of the tapes to be protected by &TAP35, use the following commands:

```
RALTER   RACFVARS  &TAP35  ADDMEM(T33333)

SETROPTS RACLIST(RACFVARS)  REFRESH
```

- **Example 2.** This example shows how to use a RACFVARS profile to assign several values to a qualifier for a class whose profile names consist of several qualifiers.

  This example creates a profile in the JESJOBS class that includes all the differently named restricted jobnames that have to be protected from submission. First, define a profile in the RACFVARS class with the protected jobnames:

  ```
  RDEFINE  RACFVARS  &PROTJOB  UACC(NONE)  ADDMEM(SALARY PAYROLLPERSONAL)
  ```

  Then create the actual JESJOBS profile that protects the jobnames:

  ```
  RDEFINE JESJOBS SUBMIT.*.&PROTJOB.* UACC(NONE)
  ```

  Now authorize a user or group to submit those protected jobnames with just one definition:

  ```
  PERMIT SUBMIT.*.&PROTJOB.* CLASS(JESJOBS) ID(GSMVS) ACCESS(READ)
  ```

  Activate the RACFVARS and JESJOBS classes and enable RACLIST processing for the RACFVARS class:

  ```
  SETROPTS CLASSACT(JESJOBS RACFVARS) RACLIST(RACFVARS)
  ```

For more information about the JESJOBS class, see 6.8, “JESJOBS Class” on page 86. For information on how to use RACFVARS with the NODES class, see 8.1.3, “NJE Levels of Trust” on page 119.

4.7.2 EGN for the DATASET Class

As before, NOEGN is in effect at RACF installation. EGN is activated with the following command:

```
SETROPTS EGN
```

One of the best reasons to implement EGN is to have compatible naming standards between the storage products and RACF. Complete syntax rules for data set profile names with EGN and NOEGN can be found in *RACF Command Language Reference*. Basically, when RACF 1.9 EGN is in effect, the characters that can be used in a generic data set profile name are:

% Specified in any qualifier except the first, matches a single character in a name. For example, A.B% equates to A.BA, A.BB, A.BC, and so on.

***** Specified as any qualifier except the first to match one qualifier, or as the last character in any qualifier except the first, to match zero or more characters until the end of the qualifier. For example, A.B.* equates to A.B.C, A.B.D, A.B.ABC, and so on, and A.B* equates to A.B, A.BA, A.BB, A.BCD, and so on.

****** Specified once as either the middle or ending qualifier to match zero or more qualifiers in a name. For example A.**.B equates to A.B, A.C.B, A.C.DEF.B, and so on.

A data set profile name can now include ** in the middle of the profile, representing zero or more qualifiers, making the following definitions valid:

```
ADDSD   ¢SYS1.**.LIST¢
ADDSD   ¢SYS1.**.LIST*¢
ADDSD   ¢SYS1.**.%LIST¢
ADDSD   ¢SYS1.*.**.%LIST¢
```

Profiles such as SYS1.* that have a single asterisk as one of the qualifiers and were created when NOEGN was in effect are translated to SYS1.*.** automatically when EGN is activated. The reason for this is that SYS1.* has a different meaning when EGN is activated, as a single asterisk covers only one qualifier.

Still, a generic character cannot be specified as the first qualifier of a data set profile. A profile can only have one ** and a qualifier cannot begin with a generic character other than (%). The following definitions are therefore invalid:

```
ADDSD   ¢*.**.LIST¢
ADDSD   ¢**.LIST¢
ADDSD   ¢SYS1.**.LIST.**¢
ADDSD   ¢SYS1.*LIST¢
ADDSD   ¢SYS1.**LIST¢
```

RACF still searches the profiles starting with the most specific match. To find the order of the search, see *RACF Security Administrators Guide*. As before, the LISTDSD command with the GENERIC operand can be used to find out the matching RACF generic profile for a data set, as follows:

```
LISTDSD   DA(¢SYS1.LINKLIB¢)   GENERIC
```

Note: Activate EGN as soon as possible and do not deactivate it, because profiles created when EGN is active can be incorrectly interpreted if NOEGN is in effect. More information on this can be found in the *RACF Command Language Reference*, in the chapter on profile naming considerations. If the RACF database is shared between multiple systems, EGN must be activated at the same time in all systems. It is not possible to run one system with EGN active and another one with EGN inactive.

4.7.3 EGN for General Resource Classes

Enhanced generic naming is always in effect for general resource profiles. Complete syntax rules for general resource profile names with EGN and NOEGN can be found in *RACF Command Language Reference*. Basically, with RACF 1.9, generic characters can be used anywhere in a general resource profile. An ** is a valid profile name in some resource classes. Unlike data set profile names with EGN active, an * at the end of a general resource profile name matches one or more qualifiers. The entry in the class descriptor table (CDT) for the class determines what restrictions apply:

- If FIRST=ANY is specified, the first qualifier can have generic characters.
- If FIRST=ANY and OTHER=ANY, then generic characters can be specified in any qualifier; otherwise the specified restrictions apply.

For a more complete description of the class descriptor table see *RACF Macros and Interfaces.* The following profile definitions are valid:

```
RDEFINE   JESINPUT *              UACC(READ)
RDEFINE   JESJOBS **.PAYROLL%.*   UACC(NONE)
RDEFINE   JESSPOOL *.USER35.**    UACC(READ)
```

Note: With RACF 1.9, the %* combination requires special attention. Profiles containing %* cannot be defined and any existing profiles containing %* should be deleted before creating any new profiles with a middle or beginning * or **. For details, refer to *RACF Security Administrators Guide*.

4.8 GENERICOWNER

Before RACF 1.9, a user with class authorization to a general resource class could define resource profiles in that class without restrictions, including a more specific profile to allow access when access was intentionally denied. Starting with RACF 1.9, the creation of more specific profiles in a general resource class can be restricted by using GENERICOWNER.

Note: GENERICOWNER does not apply to DATASET class profiles.

When the GENERICOWNER option is in effect, and a generic profile in a general resource class already exists to protect the resource, a more specific profile to protect the same resource can be created only by a user who has class authorization in that class and is one of the following:

- The owner of the next-less-specific profile.
- A user with the RACF SPECIAL attribute.
- A user with the group-SPECIAL attribute in a group that owns the next-less-specific profile.
- A user with the group-SPECIAL attribute in a group that owns the user that owns the next-less-specific profile.

In order for this support to take effect for a newly created generic profile, a generic refresh is needed. The GENERICOWNER option is activated using the following command:

```
SETROPTS GENERICOWNER
```

NOGENERICOWNER is in effect at RACF installation. The GENERICOWNER option can be used when there are subsets of profiles in a general resource class that logically belong to different users, as shown in the following examples:

- **Example 1:** This example shows the recommended way to share a user's output among several users. If the JESSPOOL class is active and the GENERICOWNER option is not in effect, users with class authorization in the JESSPOOL class can create profiles in it without any restrictions. With the GENERICOWNER option in effect, the following definitions restrict the user's authorization to a subset of profiles in the JESSPOOL class. It is also a mechanism for delegating administrative work to the user owning the output. In this example, USER34 is given authority to control access to his own output files in the JESSPOOL class in the C6JES2 node. The format of the JESSPOOL class profile is:

  ```
  nodename.userid.jobname.jobid.dsnumber.dsname
  ```

  The security administrator should own the least specific generic profile:

  ```
  RDEFINE   JESSPOOL   C6JES2.**   UACC(NONE)   OWNER(SECADM)
  ```

  To give USER34 the ability to control his own spool files, the security administrator defines the following profile:

  ```
  RDEFINE   JESSPOOL   C6JES2.USER34.**    OWNER(USER34)   UACC(NONE)
  ```

Then USER34 is given the class authorization to JESSPOOL, which allows him to define more specific profiles in that class:

```
ALTUSER USER34 CLAUTH(JESSPOOL)
```

With the GENERICOWNER option in effect, USER34 is able to define and change profiles only for his own output in the C6JES2 node. Otherwise, he would be able to define profiles for other users' output files. USER34 can now protect all his output files that have a jobname beginning with MYJOB. Then he can permit other users to access output files that are covered by this profile:

```
RDEFINE   JESSPOOL   C6JES2.USER34.MYJOB*.**    UACC(NONE)

PERMIT    C6JES2.USER34.MYJOB*.**  CLASS(JESSPOOL)   ID(USER33)   ACCESS(READ)
```

If USER34 tries to define a profile for USER33's output files, he gets an authorization failed message. The command and resulting message is as follows:

```
RDEFINE   JESSPOOL   C6JES2.USER33.**   UACC(READ)

NOT AUTHORIZED TO DEFINE C6JES2.USER33.**
```

For more information about the JESSPOOL class see Chapter 7, "SYSIN / SYSOUT - JES Spool" on page 95.

- **Example 2:** In this example, a service bureau installation wants to authorize its customers to specific jobnames for accounting purposes. Every customer also has to have one RACF administrator (group-SPECIAL) who can maintain the definitions for the different customer departments, but is not authorized to specify definitions to the other customer's jobnames.

  Figure 13 shows two customers defined to RACF as groups GFIN and GPERS. Both have subgroups that serve as default groups for the users of different departments. USER00 and USER30 are connected to GFIN and GPERS, respectively, as RACF group-SPECIAL users. The systems programming group, which takes care of the system maintenance, is defined as GSYS, and has three subgroups. The groups HLQPERS and HLQFIN are data control groups for customers data sets. In this example, a group is owned by its superior group, which is a general recommendation to implement the ownership of a group structure in RACF.

  The requirement is that customer GFIN must use jobnames starting with letter A, and GPERS jobnames starting with B. To implement this scenario, use profiles in the JESJOBS class and activate the GENERICOWNER option.

  First, define one generic profile in the JESJOBS class and permit systems programming groups to it. This allows them to submit jobnames that are not specifically restricted by other definitions in the JESJOBS class:

  ```
  RDEFINE   JESJOBS   SUBMIT.*.**   UACC(NONE)

  PERMIT    JESJOBS   SUBMIT.*.**   ID(GSMVS GSNET GSDB GSYS)   ACCESS(READ)
  ```

  Then define profiles for jobnames starting with A and B in the JESJOBS class and assign the RACF group-SPECIAL users of both customers to become the owners of their corresponding profile:

  ```
  RDEFINE   JESJOBS   SUBMIT.*.A*.*   UACC(NONE)   OWNER(USER00)

  RDEFINE   JESJOBS   SUBMIT.*.B*.*   UACC(NONE)   OWNER(USER30)
  ```

  Give class authorization to JESJOBS to both group-SPECIAL users and activate the GENERICOWNER option:

  ```
  ALTUSER   (USER00 USER30)   CLAUTH(JESJOBS)

  SETROPTS   GENERICOWNER
  ```

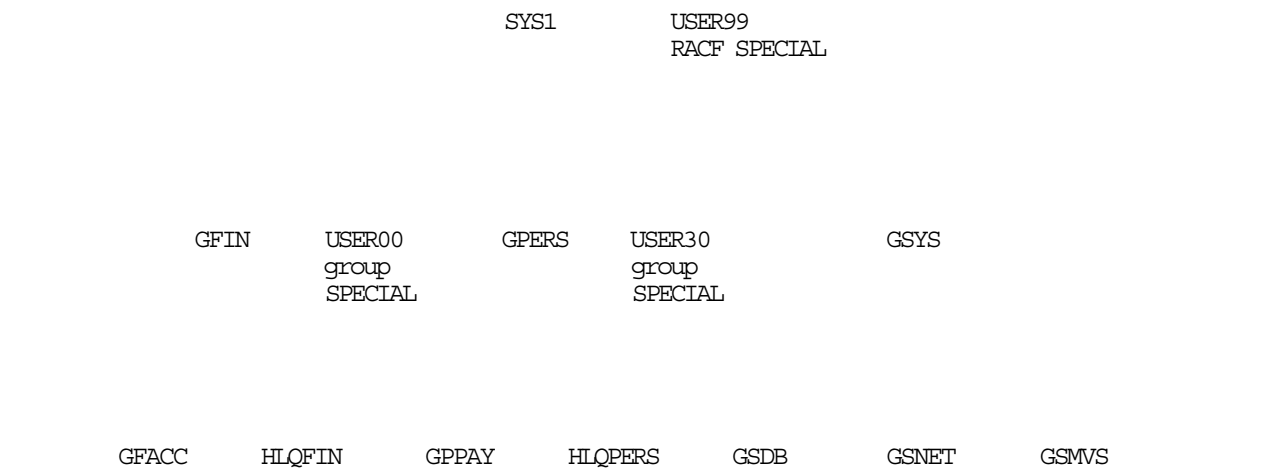

Figure 13. Sample Group Structure

Now both RACF group-SPECIAL users can define more specific profiles in the JESJOBS class and authorize groups to use them. For example, USER30 can define only jobnames that start with B. Because the GENERICOWNER option has been activated, he cannot define jobnames that start with any other letter. The following definitions by USER30 force groups GPHIRE and GPPAY to use more specific jobnames:

```
RDEFINE   JESJOBS   SUBMIT.*.B22222%%.*   UACC(NONE)
RDEFINE   JESJOBS   SUBMIT.*.B33333%%.*   UACC(NONE)

PERMIT   SUBMIT.*.B22222%%.*   CLASS(JESJOBS)   ID(GPHIRE)   ACCESS(READ)
PERMIT   SUBMIT.*.B33333%%.*   CLASS(JESJOBS)   ID(GPPAY)   ACCESS(READ)
```

Similar definitions can be made by USER00 for the other customer. The jobnames that do not start with the letters A or B can be submitted by the system maintenance groups of the service bureau itself. They can also be restricted to specific jobnames.

For more information about the JESJOBS class, see 6.8, “JESJOBS Class” on page 86.

4.9 CATDSNS

The CATDSNS option requires non-temporary tape or DASD data sets to be cataloged through the master catalog in order for RACHECK to allow access. CATDSNS can be activated in either WARNING or FAILURES mode using the SETROPTS command:

```
SETROPTS CATDSNS ( WARNING | FAILURES )
```

NOCATDSNS is in effect at RACF installation. When CATDSNS(FAILURES) is in effect, the following exceptions allow access to an uncataloged data set:

- If the uncataloged data set is protected by a discrete profile with the volume serial, a user can access the data set with the level specified in the access list of that profile. If an installation has identically named uncataloged data sets on different volumes, discrete profiles allow access to be given separately to these data sets:

```
ADDSD  data_set_name VOLUME(volid) UNIT(unit) UACC(NONE)

PERMIT data_set_name ID(USER34) ACCESS(UPDATE)
```

- If the data set is protected by a fully qualified generic profile, the user can access the data set with the level specified in the profile:

```
ADDSD  data_set_name GENERIC UACC(NONE) ...fully qualified data_set_name

PERMIT data_set_name ID(USER34) ACCESS(UPDATE)
```

- If the uncataloged data set is protected by a profile in the FACILITY class and the user has at least READ access to the FACILITY class profile, the user can access the data set with the level specified in the DATASET class profile. The format of the FACILITY class profile is:

```
ICHUNCAT.data_set_name
```

  An uncataloged data set is accessible to a user with at least READ access to the profile:

```
RDEFINE FACILITY ICHUNCAT.data_set_name UACC(NONE)

PERMIT  ICHUNCAT.data_set_name CLASS(FACILITY) ID(USER34) ACCESS(READ)
```

  The level at which the user has access to the uncataloged data set is still determined from the actual profile for that data set in the DATASET class thus, READ authority to the profile in the FACILITY class only gives authority to access the uncataloged data set; authorization to the FACILITY class profile alone is not sufficient to process the data set.

  With this approach, it is very simple to control access to all uncataloged data sets with just one profile ICHUNCAT.** in the FACILITY class.

- If the user is RACF SPECIAL, the user can access any uncataloged data set, but a WARNING message is issued.

Example: An installation has two SYS1.LPALIB data sets, one for the production system and one for the test system, as shown in Figure 14. The SYS1.LPALIB data set for the backup system is uncataloged and only the system programmers are allowed to update that data set. The following definition protects all system data sets:

```
ADDSD ¢SYS1.**¢ UACC(NONE)
```

In addition to this profile, there can be more specific profiles beginning with SYS1 with specific authorization requirements (for example SYS1.BRODCAST UACC(UPDATE)) or they can be specified in the global access checking table. Authorization to read system data sets is given to system programmers connected to the group TRAINEE. Experienced system programmers are connected to the group SYSPROG, which is given ALTER authorization to system data sets.

```
PERMIT ¢SYS1.**¢ ID(TRAINEE) ACCESS(READ)

PERMIT ¢SYS1.**¢ ID(SYSPROG) ACCESS(ALTER)
```

The CATDSNS option is activated and a profile is created in the FACILITY class to authorize the system programmer groups to access uncataloged system data sets:

```
SETROPTS CATDSNS(FAILURES)

RDEFINE  FACILITY  ICHUNCAT.SYS1.**  UACC(NONE)

PERMIT   ICHUNCAT.SYS1.** CLASS(FACILITY)  ID(TRAINEE SYSPROG)  ACCESS(READ)
```

The ability to access uncataloged system data sets is provided for these groups through the profile in the FACILITY class and the level of authorization through the profile in the DATASET class. Additionally, any user with the SPECIAL attribute can access the uncataloged system data sets.

Note: Access to an uncataloged data set is not granted by an entry in the global access checking table. The user must have access to the data set in one of the above mentioned profiles.

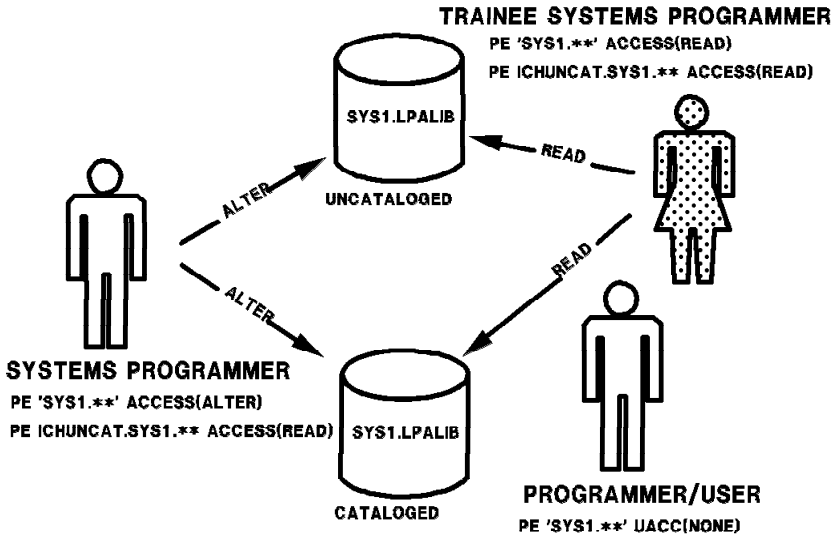

Figure 14. CATDSNS Example

When CATDSNS(FAILURES) is active, if a user uncatalogs a data set, he cannot access it or delete it until profiles have been defined to allow him to access the uncataloged data set. If CATDSNS is activated in WARNING mode, access to uncataloged data sets is allowed but a not-cataloged message is issued every time a user accesses an uncataloged data set without proper authorization. If the data set is protected by a generic profile and a user is not authorized to it, the normal authorization failed message is issued instead.

CATDSNS covers only those data sets that are cataloged in the master catalog or have an alias pointing to a user catalog in the master catalog. The use of JOBCATs and STEPCATs is not allowed. If access is through a JOBCAT or STEPCAT, RACF access requests fail unless the user has at least READ access to the ICHUCAT profile in the FACILITY class.

4.9.1 Effect on Type 83 SMF Record

A new SMF type 83 record contains a list of the data sets that can be identified through the master catalog that are affected by a change, addition, or deletion of the SECLABEL information in a RACF profile. If the CATDSNS option is in effect, the output of this list, like that of the one produced with the LISTDSD DSNS command, can be considered more complete. For an example of the type 83 SMF record in a Report Writer listing, see 4.14.1, “SMF Enhancements” on page 55.

4.9.2 Effect on the LISTDSD Command

The LISTDSD command with the new DSNS operand can be used to list all data sets that are protected by a specific generic profile, as long as the data set can be found through the master catalog. If the CATDSNS option is in effect, the output of this list can be considered more complete. Figure 15 on page 48 shows the output of the following command:

```
LISTDSD DA(¢SYSPROG.*.**¢) DSNS
```

The following command, with the NORACF operand, produces only the section titled *DATA SETS AFFECTED BY PROFILE CHANGE*:

```
LISTDSD DA(¢SYSPROG.*.**¢) DSNS NORACF
```

```
INFORMATION FOR DATASET SYSPROG.*.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    SYSPROG         NONE          NO      NO

                    .
                    .
                    .


DATA SETS AFFECTED BY PROFILE CHANGE
------------------------------------
SYSPROG.EXEC.RACF.CLIST
SYSPROG.ISPF.ISPPROF
SYSPROG.ISPF.ISPTLIB
SYSPROG.JCL.CNTL
SYSPROG.JCL.JOB
SYSPROG.RACF.OUTLIST
SYSPROG.SPFLOG1.LIST
SYSPROG.SPFTEMP0.CNTL
SYSPROG.SPF146.OUTLIST
SYSPROG.SPF151.OUTLIST
SYSPROG.SPF152.OUTLIST
```

Figure 15. LISTDSD Output With DSNS Operand

4.10 Group Tree in Storage

The group tree in storage (GTS) facility improves the performance of group authority processing for RACF commands. This performance improvement can be gained without restructuring the RACF database or making any changes to RACF. GTS improves performance for the use of group attributes (group-SPECIAL, group-OPERATIONS, group-AUDITOR) when the user has authority over a wide or deep group tree. Although the term 'group tree in storage' is used, RACF does not maintain an actual group tree in storage.

Previously, both the RACF command processors and the RACF SVC routines built a group-authority tree from top to bottom. The tree contained a complete list of groups in which the user had group attributes. This tree was built by reading all group records for groups and subgroups for which the user had authority. Building this tree could be very time consuming, and could also require large amounts of I/O. In addition, the tree was deleted by the command processors at command processing termination, causing it to be rebuilt for each RACF command issued. RACF 1.9 improves the performance of group authority checking in several areas:

- RACF builds a subset of the group authority tree from bottom up only as far as needed to obtain the requested authority.
- RACF command processors use enhanced SVC routines to perform group authority checking, instead of building their own group tree and deleting it when processing completes.
- If RACF 1.9 is running with MVS/SP 3.1.0 or higher, RACF retrieves the group information objects needed to build the tree from the virtual lookaside facility (VLF) data space where they were created when initially read from the RACF database.

To activate VLF for the GTS function, define the VLF class name IRRGTS and the major name GTS in the PARMLIB member COFVLFxx. If IRRGTS is not found in PARMLIB, VLF fails the RACF request to

define VLF objects and VLF is not used by the GTS function. This action allows an installation to choose whether to use VLF for the GTS function. The following is an example of COFVLFxx:

```
SYS1.PARMLIB(COFVLFxx): CLASS NAME(IRRGTS)
                        EMAJ(GTS)
```

If an installation is a shared RACF environment, VLF should be used only when all systems sharing the RACF database are at RACF 1.9. Information about group changes are stored in the RACF database, and would not be reflected correctly in the VLF data spaces of other systems unless they are also at RACF 1.9.

4.11 RACF Utilities and the RDB

The new restructured database (RDB) eliminates some constraints of the old format. The new format provides space for the enhanced functions provided in RACF 1.9 and provides a base for future growth. Changes to the format include:

- The blocksize of the RACF RDB is increased from 1024 to 4096 bytes. If the system is storage constrained, be aware that, for the restructured database, the block size is 4 KB (K=1024) rather than 1 KB, and thus an equal number of data blocks requires four times the ECSA storage than the non-restructured database.
- Longer profile names can be defined. The old format allows profile names to be 39 characters long; the new format allows profile names to be 246 characters long. Some new profile definitions with RACF 1.9 can easily exceed 39 characters.
- The index structure has been enhanced so that each segment has its own index entry, allowing it to be read without having to read the base segment. Null segments do not take up space in the RDB.
- There are no connect profiles in the RDB. The conversion program merges most information from the connect profiles into the user profile; the information not merged is no longer supported. For a list of these connect profile fields, see *Systems Programming Library: RACF.*

Another important aspect of the new restructured database is that it would be the base for future RACF enhancements, if any. RACF 1.9 is the last release of RACF to allow the unrestructured format of the data base.

4.11.1 New Utilities

Six new utilities (IRR...) support the restructured RACF database (RDB). Five of the new utilities perform, for the most part, the same functions as the existing utilities (ICH...). The sixth new utility, IRRDSC00, converts the database from the old structure to the new RACF database structure. The new utilities and their functions are:

IRRMIN00 Initialization / upgrade

Formats a non-VSAM DASD data set for use as a restructured RACF database for RACF 1.9 or later. Use this utility during the initial installation of RACF 1.9 (PARM=NEW) or to update a restructured RACF database (PARM=UPDATE).

IRRUT100 Cross reference

Lists all the occurrences of a userid or group name that exist in the restructured RACF database.

IRRUT200 Verification

Identifies inconsistencies in the internal organization of a restructured RACF database and provides information about the size and organization of a restructured RACF database. Use this utility to create an exact copy of a restructured RACF database. IRRUT200 performs more extensive verification than ICHUT200 and gives error messages and return codes that ICHUT200 does not.

IRRUT300 Block update

Modifies the records in a restructured RACF database. Use this utility to correct any inconsistencies that the restructured RACF database verification utility (IRRUT200) identifies.

IRRUT400 Data set split / merge / extend

Splits a single, existing restructured RACF database into several parts, re-combines or redistributes the physical data on those parts, and copies a restructured RACF database into a larger database. IRRUT400 has a new UNLOCKINPUT keyword not available with ICHUT400 to unlock all databases that have been previously locked by the LOCKINPUT keyword.

IRRDSC00 Convert from old to new database structure (new)

Converts the old RACF database structure, with up to 255 parts, into the new RACF database structure with the same number of parts. The output database specified cannot have a disposition of new, and it must be pre-formatted by IRRMIN00.

4.11.2 Allocating a New RACF Database

To allocate and initialize the RACF 1.9 database, run the IRRMIN00 or ICHMIN00 utility. IRRMIN00 is used to allocate a database using the new format and ICHMIN00 is used for a database in the old format. The JCL to execute either utility is as follows:

```
//S1 EXEC PGM=yyyyyyyy,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSTEMP DD DISP=SHR,DSN=SYS1.MODGEN(xxxxxxxx)
//SYSRACF DD DSN=SYS1.TEST,DISP=(,CATLG),
//  VOL=SER=RACCAT,UNIT=3380,SPACE=(CYL,(20),,CONTIG),
//  DCB=DSORG=PSU
```

Where:

yyyyyyyy Utility name. ICHMIN00 for the old format; IRRMIN00 for the new format.

xxxxxxxx Template name. ICHTEMP0 for the old format; IRRTEMP1 for the new format.

Note: The RACF 1.9 template data resides in SYS1.MODGEN; not in SYS1.MACLIB where it resided in previous releases of RACF.

4.11.3 Migrating from RACF 1.8.1

The migration from RACF 1.8.1 can be done in three steps. First, ensure that the RACF database is adequately backed up. Then, run the template utility ICHMIN00 on the existing database. Finally, run the restructure utility IRRDSC00. This final step is optional because the existing system does function using the old structured database. When the database that is being updating is the active database, ICHMIN00 obtains an exclusive reserve (enqueue) on it. If any of the copies of the database blocks are in main storage, they are rendered invalid and not referenced. RACF restores the database blocks as they are needed. To update the templates, use the RACF 1.9 utility ICHMIN00 with PARM=UPDATE as shown in the following JCL:

```
//S1 EXEC  PGM=ICHMIN00,PARM=UPDATE
//SYSPRINT DD SYSOUT=*
//SYSTEMP  DD DISP=SHR,DSN=SYS1.MODGEN(ICHTEMP0)
//SYSRACF  DD DSN=SYS1.RACF,DISP=SHR
```

Migration testing was done using RACF 1.8.1. If converting from an earlier release, consult the *RACF Program Directory for MVS Systems* for more information.

4.11.4 Converting a RACF Database to RDB Format

The procedure discussed below creates a new RACF database in the RDB format while the system is active. This procedure ensures that after the conversion to the RDB format, the primary database is the active database and the system does not lose usability of the in-store buffers. It would be advisable to have little or no activity on the system, as specifying LOCKINPUT causes the database to refuse inputs until it is unlocked. If the system is going to be re-IPLed, the procedure is much simpler: create the new data sets, copy the databases, change ICHRDSNT to point to the new data sets, and re-IPL the system.

To convert a RACF database to RDB format, assume that the system is IPLed with a primary non-RDB data set SYS1.RACF and a backup non-RDB data set called SYS1.RACF.BACKUP. Then, proceed as follows:

1. Run the data set convert utility with the parameter LOCKINPUT, to read the primary data set and construct a new RDB data set:

   ```
   SYS1.RACF  --IRRDSC00--> SYS1.RACF.RDB
   ```

2. Use the RVARY INACT command to inactivate the backup data set:

   ```
   SYS1.RACF.BACKUP
   ```

3. Use ICHUT200 or another utility to make a backup copy of the backup data set. This is to ensure that there is a backup copy of the backup data set although it may never be used. If the conversion is successful, this data set can be deleted:

   ```
   SYS1.RACF.BACKUP  --ICHUT200--> SYS1.RACF.BACKUP.NEW
   ```

4. Use ICHUT200 or another utility to copy the primary data set onto the backup data set. Both data sets are now identical and locked:

   ```
   SYS1.RACF  --ICHUT200--> SYS1.RACF.BACKUP
   ```

5. Use the RVARY ACTIVE command to activate the backup data set (both are active and locked).

6. Use the RVARY SWITCH command to switch the active data set. The SYS1.RACF.BACKUP is now the locked primary data set and SYS1.RACF is now an inactive locked backup data set with buffers:

```
SYS1.RACF.BACKUP  (Active, locked, primary)

SYS1.RACF  (Inactive, locked)
```

7. Rename the current inactive locked backup data set, SYS1.RACF, to a different name:

```
SYS1.RACF  ---> SYS1.RACF.OLD
```

8. Rename the output of IRRDSC00, SYS1.RACF.RDB, to SYS1.RACF. (Now the current backup data set is in RDB format, and has the same name, SYS1.RACF, as the primary data set had at the beginning of this process):

```
SYS1.RACF.RDB  ---> SYS1.RACF
```

9. Use the RVARY ACTIVE command to activate the backup data set SYS1.RACF (both are now active).

10. Use the RVARY SWITCH command to switch the SYS1.RACF data set with the SYS1.RACF.BACKUP data set. SYS1.RACF is now the RDB primary data set with buffers, and SYS1.RACF.BACKUP as the non-RDB inactive locked backup data set:

```
SYS1.RACF  (Active)

SYS1.RACF.BACKUP  (Inactive)
```

11. Run IRRDSC00 against the backup data set, SYS1.RACF.BACKUP, with UNLOCKINPUT:

```
--IRRDSC00--> SYS1.RACF.BACKUP
```

12. Use the RVARY ACTIVE command to activate the backup data set, SYS1.RACF.BACKUP:

```
SYS1.RACF  (Active)

SYS1.RACF.BACKUP  (Active)
```

The following JCL can be used to convert a non-RDB data set to an RDB:

```
//CONVERT  EXEC PGM=IRRDSC00,PARM=LOCKINPUT
//SYSPRINT DD   SYSOUT=*
//INDD1    DD   DSN=SYS1.RACF,DISP=OLD
//OUTDD1   DD   DSN=SYS1.RACF.RDB,DISP=OLD
```

Where:

PARM=LOCKINPUT Prevents updates while the job is running.

INDD(n) Old format database to be converted.

OUDD(n) Newly created restructured database.

Note: Rerun this job with PARM=UNLOCKINPUT option to unlock the database.

4.12 ID(*) in the Access List

With RACF 1.9, ID(*) can be specified in an access list to represent all RACF-defined users. This control enables an installation to give different access authorizations to RACF-defined users and users not defined to RACF. UACC applies to any users not defined to RACF and any RACF-defined users that are not on the access list; ID(*) is similar to having all RACF-defined users on the access list with the specified access. The following command allows access by all users:

```
RDEFINE TERMINAL terminal UACC(READ)
```

Use the following commands to disallow access by users not defined to RACF, but allow any RACF-defined user READ access:

```
RDEFINE TERMINAL terminal UACC(NONE)

PERMIT  terminal CLASS(TERMINAL) ID(*) ACCESS(READ)
```

4.13 New Forms of Conditional Access

Conditional access to a resource can be given to a user or group with the WHEN parameter on the RACF PERMIT command. Prior to RACF 1.9, access to data sets could be granted based on the controlled program used to access the data set by specifying WHEN(PROGRAM(program_name). RACF 1.9 allows the specification of several new conditional accesses that can require a user to access general resources from particular RACF-defined devices such as WHEN(TERMINAL(terminal_id)), WHEN(JESINPUT(device_name)), and WHEN(CONSOLE(console_id)). The following comments apply to all forms of conditional access:

- If the SECLABEL class is active, SECLABEL checking applies to accesses granted by the conditional criteria.
- On the PERMIT command, ID(*) can be specified to allow access by all RACF defined userids.
- If more than one condition is used in a conditional access list, any *one* of the conditions allows the requested access.
- Program_name, terminal_id, device_name, and console_id cannot contain generic characters and must be protected by either a discrete or generic profile in the respective class; otherwise, the WHEN condition is ignored.
- The WHEN(PROGRAM) access is applicable to only data set profiles; the other WHEN conditional accesses are applicable to both data set and general resource profiles.

4.13.1 WHEN(PROGRAM) Access

This facility is not new with RACF 1.9. RACF 1.9 does, however, improve its usability. It is now possible to audit attempted accesses to controlled programs and to specify a NOTIFY userid for profiles in the PROGRAM class. Note that the PROGRAM class does not have to be active for WHEN(PROGRAM) to be effective, but the SETROPTS WHEN(PROGRAM) option must be active. For details, refer to the discussion on program control in *RACF Security Administrators Guide*.

4.13.2 WHEN(TERMINAL) Access

This facility allows access to be given only when the user is logged on to a specific terminal. The following example shows the use of WHEN(TERMINAL) with the JESJOBS class:

- Define a profile in the JESJOBS class so that no user can submit a job with jobname DSMON:

  ```
  RDEFINE  JESJOBS SUBMIT.*.DSMON.*  UACC(NONE)
  ```

- Permit USER82 to submit jobs with jobname DSMON:

  ```
  PERMIT  SUBMIT.*.DSMON.*  CLASS(JESJOBS)  ID(USER82)  ACC(READ)
  ```

- Permit USER70 to submit jobs with jobname DSMON only from terminal C6LOC5E1:

  ```
  PERMIT  SUBMIT.*.DSMON.*  CLASS(JESJOBS)  ID(USER70)  ACC(READ)  WHEN(TERMINAL(C6LOC5E1))
  ```

Note that the WHEN(TERMINAL) condition is effective only if the TERMINAL class is active. Before activating the TERMINAL class in order to use the WHEN(TERMINAL) conditional access, become

acquainted with the section on protecting terminals on MVS in the *RACF Security Administrators Guide*. It is important that terminals be defined and users be permitted access before the SETROPTS TERMINAL(NONE) option is activated or no user can logon to the system.

4.13.3 WHEN(JESINPUT) Access

This facility allows access to be given only when the user enters the system through the specific JES input device. The following example shows the use of WHEN(JESINPUT) with the JESJOBS class:

- Define a profile in the JESJOBS class so that no user can submit a job with jobname DSMON:

```
RDEFINE JESJOBS SUBMIT.*.DSMON.* UACC(NONE)
```

- Permit any user defined in group GSMVS to submit jobname DSMON from system C2JES2:

```
PERMIT SUBMIT.*.DSMON.* CLASS(JESJOBS) ID(GSMVS) ACC(READ) WHEN(JESINPUT(C2JES2))
```

Note that the WHEN(JESINPUT) condition is effective only if the JESINPUT class is active. For more details on the JESINPUT CLASS see Chapter 6, “Implementing Security on Job Entry Subsystems” on page 71.

4.13.4 WHEN(CONSOLE) Access

This facility allows access to be restricted to requests that come from a specified system console. Although not the sole use, the most obvious use of WHEN(CONSOLE) is to restrict the use of certain operator commands to certain consoles. The following example shows the use of WHEN(CONSOLE) with the OPERCMDS class:

- Define a profile in the OPERCMDS class so that no user can cancel TSO users:

```
RDEFINE OPERCMDS MVS.CANCEL.TSU.* UACC(NONE) AUDIT(ALL)
```

- Permit the group GSMVS to cancel TSO users from CONSOLE 07:

```
PERMIT MVS.CANCEL.TSU.* CLASS(OPERCMDS) ACC(UPDATE) ID(GSMVS) WHEN(CONSOLE(07))
```

Note that the WHEN(CONSOLE) condition is effective only if the CONSOLE class is active. For more details on the CONSOLE class see Chapter 10, “Console and Command Security” on page 163.

4.13.5 Recommendations

For WHEN(PROGRAM), if the SECLABEL class is active, the SECLABEL of the user must dominate the SECLABEL of the data set containing the load module. Unless a load module library contains classified programs, it should have a SECLABEL of SYSNONE. This label allows program access by users with any SECLABEL, even if the MLACTIVE and MLS options are enabled. For additional information see 3.6.3, “MLS Active” on page 27 and the *RACF Security Administrators Guide*.

For WHEN(TERMINAL) and WHEN(CONSOLE), since the TERMINAL and CONSOLE classes specify RVRSMAC=YES in the CDT, their SECLABELs must dominate the user's SECLABEL. The SECLABEL of a terminal or console should be assigned based on its physical level of protection. A terminal located in a hotel lobby should probably be assigned a SECLABEL of SYSLOW, whereas a console located in a computer room could probably be assigned a SECLABEL of SYSHIGH.

4.14 Auditing Enhancements

Auditing within RACF is controlled by the system auditor, who sets the required auditing options based on the installation's needs. Figure 16 shows an overview of the auditing enhancements that include:

- RACF audit options are now usable for PROGRAM class profiles.
- RACF global options include auditing by security label or resource class.
- RACF RACROUTE changes provide the ability to include a character string in an SMF record and a call for auditing without authorization checking.
- SMF records have been enhanced; a new SMF record type is added.
- RACF Report Writer has been enhanced.

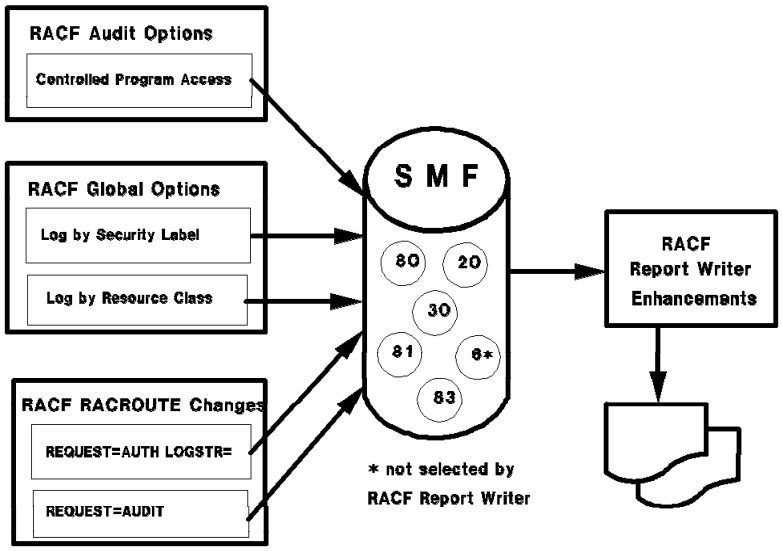

Figure 16. Enhanced Auditing in RACF 1.9

4.14.1 SMF Enhancements

In order to support the new security environment, more Type 80 records are being generated for many new RACF interfaces, such as:

- Console operator command security (OPERCMDS)
- Device allocation security (DEVICES)
- JES spool data set security (JESSPOOL,WRITER)
- DFP-managed temporary data set security (TEMPDSN)
- VTAM connections security checking (VTAMAPPL)
- SEND and LISTBC security (SMESSAGE,DIRAUTH)
- Job entry security (JESJOBS,PROPCNTL)
- TSO OUTPUT and CANCEL security (JESJOBS,JESSPOOL)
- PSF/MVS page overlay security (PSFMPL)

There are also several new fields in the type 80 records. For example, the SECLABEL of the user is included in the base segment and several new relocate sections are defined for the user's and the resource's security tokens. New event codes and qualifiers are introduced. For example, it is now possible to audit an undefined user's logon attempts with a new type 80 record (event code 1, qualifier 9). This means that RACINIT is called even when a user who is not defined to RACF attempts to log on to the system.

The new type 83 SMF record is a RACF processing record for auditing data sets that are affected by a RACF command that caused the SECLABEL of a data set to be changed. The SECLABEL of a data set can be changed explicitly by using the ALTDSD command with SECLABEL operand against the profile that protects the data set. However, a SECLABEL change of a data set can also be caused by adding a new, more specific profile with a different SECLABEL that protects the data set. This applies also to a deletion of a data set profile, when there is an existing profile with a different SECLABEL that protects the data set. The list of data sets produced by SMF record type 83 is exactly the same as the list produced by the LISTDSD DSNS command. If CATDSNS is active, the list of data set names can be considered more complete.

Note: SMF type 83 records are produced only when the MLACTIVE option is active and the SECLABEL of a generic profile is changed.

Figure 17 shows how the deletion of a more specific profile produces a change in SECLABEL for some data sets. The first record is logged because SYSPROG, a RACF SPECIAL user, deleted a data set profile. The second record is from the SMF type 83 record, which shows the data sets whose SECLABELs are changed due to the DELDSD command. For a full description of RACF SMF record types, see *Systems Programming Library: RACF* and *RACF Auditors Guide.*

```
90.205 12:58:15 SMF6   SYSPROG   SYS1    SCGSA006   0 15  0  JOBID=(SYSPROG 90.205 12:27:14),USERDATA=(),OWNER=SYSPROG
                       TESTUSER                             AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS)
                                                            SESSION=TSO LOGON,TERMINAL=SCGSA006
                                                              DELDSD SYSPROG.SMF.**
                                                              NEW SECLABEL=SE,OLD SECLABEL=SCPE,LINK=3
90.205 12:58:15 SMF6   SYSPROG   SYS1    SCGSA006   0 15  3  JOBID=(SYSPROG 90.205 12:27:14),USERDATA=(),OWNER=SYSPROG
                       TESTUSER                             AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS)
                                                            SESSION=TSO LOGON,TERMINAL=SCGSA006
                                                            LINK=3,DATA SETS AFFECTED:
                                                            SYSPRPG.SMF.SAVE1,SYSPROG.SMF.SAVE2,SYSPROG.SMF.SAVE3,
                                                            SYSPROG.SMF.SAVE4
```

Figure 17. Report Writer Listing for an SMF Type 83 Record

4.14.2 SECLABELAUDIT

With the AUDITOR attribute, a user can request that access attempts to resources be audited by security label. This means that accesses can be logged for any resource with a specific SECLABEL, whether the resource profiles have any auditing requirements or not. The level of logging is determined from either the SECLABEL profile or the resource profile. If either of the profiles request auditing, the access is audited. This option can be used, for example, to ensure that all confidential data has at least the required level of logging specified so that each resource profile does not have to be examined for proper auditing options. NOSECLABELAUDIT is in effect at RACF installation. To activate this option, specify:

```
SETROPTS SECLABELAUDIT
```

Figure 18 assumes that the following RACF definitions have been made:

```
RDEFINE SECLABEL  SYSHIGH  AUDIT(ALL)

ADDSD   ¢USER13.*.**¢   AUDIT(FAILURES(READ))

ADDSD   ¢USER22.*.**¢   AUDIT(FAILURES(READ))
```

If the SECLABELAUDIT option is not in effect, only the failed accesses to data sets covered by the profiles in the example are logged. If the SECLABELAUDIT option is activated, the information shown in Figure 18 is logged to the SMF data set when user SYSPROG accesses any data set that is covered by a profile with a SECLABEL of SYSHIGH.

```
                                                          E
                                                          V  Q
                                                          E  U
                                *JOB/USER  *STEP/     --TERMINAL--  N  A
 DATE     TIME     SYSID        NAME      GROUP         ID     LVL  T  L
90.061 18:54:49 SMF6   SYSPROG   SYS1      C6LOC5E4      0    2  0  JOBID=(SYSPROG 90.061 18:42:41),USERDATA=(),OWNER=RADMIN
                       TEST USER                                    AUTH=(OPERATIONS),REASON=(SECLABELAUDIT)
                                                                    USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                    DATASET=USER13.DATA,GENPROF=USER13.**,VOLUME=RACTS1,LEVEL=00,INTENT=
                                                                    READ,ALLOWED=ALTER,RESOURCE SECLABEL=SYSHIGH
90.061 18:54:51 SMF6   SYSPROG   SYS1      C6LOC5E4      0    2  0  JOBID=(SYSPROG 90.061 18:42:41),USERDATA=(),OWNER=RADMIN
                       TEST USER                                    AUTH=(OPERATIONS),REASON=(SECLABELAUDIT)
                                                                    USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                    DATASET=USER13.DATA,GENPROF=USER13.**,VOLUME=RACTS1,LEVEL=00,INTENT=
                                                                    READ,ALLOWED=ALTER,RESOURCE SECLABEL=SYSHIGH
90.061 18:54:54 SMF6   SYSPROG   SYS1      C6LOC5E4      0    2  0  JOBID=(SYSPROG 90.061 18:42:41),USERDATA=(),OWNER=RADMIN
                       TEST USER                                    AUTH=(OPERATIONS),REASON=(SECLABELAUDIT)
                                                                    USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                    DATASET=USER13.DATA,GENPROF=USER13.**,VOLUME=RACTS1,LEVEL=00,INTENT=
                                                                    UPDATE,ALLOWED=ALTER,RESOURCE SECLABEL=SYSHIGH
90.061 18:55:11 SMF6   SYSPROG   SYS1      C6LOC5E4      0    2  0  JOBID=(SYSPROG 90.061 18:42:41),USERDATA=(),OWNER=USER22
                       TEST USER                                    AUTH=(NORMAL),REASON=(SECLABELAUDIT)
                                                                    USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                    DATASET=USER22.JCL.CNTL,GENPROF=USER22.*.**,VOLUME=RACTS1,LEVEL=00,
                                                                    INTENT=READ,ALLOWED=UPDATE,RESOURCE SECLABEL=SYSHIGH
90.061 18:55:11 SMF6   SYSPROG   SYS1      C6LOC5E4      0    2  0  JOBID=(SYSPROG 90.061 18:42:41),USERDATA=(),OWNER=USER22
                       TEST USER                                    AUTH=(NORMAL),REASON=(SECLABELAUDIT)
                                                                    USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                    DATASET=USER22.JCL.CNTL,GENPROF=USER22.*.**,VOLUME=RACTS1,LEVEL=00,
                                                                    INTENT=READ,ALLOWED=UPDATE,RESOURCE SECLABEL=SYSHIGH
90.061 18:56:11 SMF6   SYSPROG   SYS1      C6LOC5E4      0    2  1  JOBID=(SYSPROG 90.061 18:42:41),USERDATA=(),OWNER=USER22
                       TEST USER                                    AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING,SECLABELAUDIT)
                                                                    USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                    DATASET=USER22.JCL.CNTL,GENPROF=USER22.*.**,VOLUME=RACTS1,LEVEL=00,
                                                                    INTENT=ALTER,ALLOWED=UPDATE,RESOURCE SECLABEL=SYSHIGH
```

Figure 18. Report Writer Listing when SECLABELAUDIT is Used

4.14.3 LOGOPTIONS

With the AUDITOR attribute, a user can request that access attempts to resources be audited by resource class. This means that all RACHECKs against an active class can be audited, even those made by trusted procedures. Before RACF 1.9, it was possible to obtain logging for a resource class only by manipulating each profile in the class. The following command is used to activate resource class logging:

```
SETROPTS LOGOPTIONS(option(class_name))
```

The class_name can be DATASET or any class defined in the class descriptor table. Resources are not required to have profiles for an access to be audited. In fact, LOGOPTIONS is the only way to audit accesses to resources in classes that are not allowed to have profiles (PROFDEF=NO); DIRAUTH and TEMPDSN. It is also the only way to audit accesses to resources by trusted procedures. With LOGOPTIONS, auditing is requested for each class with one of the following options:

ALWAYS All attempts to access resources protected by the class are audited, despite audit options in the resource profile.

NEVER No attempts to access resources protected by the class are audited, despite audit options in the resource profile. All auditing is suppressed.

SUCCESSES All successful attempts to access resources protected by the class are audited as well as any auditing requested in the resource profile.

FAILURES All failed attempts to access resources protected by the class are audited as well as any auditing requested in the resource profile.

DEFAULT Auditing is controlled by the profile protecting the resource, if a profile exists. The default for all classes can be specified with DEFAULT(*).

LOGOPTIONS(DEFAULT(*)) is in effect at RACF installation. Even if LOGOPTIONS requests logging, the event is not logged if either of the following occurs:

- The option is SUCCESSES, FAILURES, or DEFAULT and the program issues a RACHECK or RACDEF request that specifies no logging.
- RACF grants access to a resource because of an entry in the global access checking table.

For example, the following command requests that all access attempts to the DEVICES class be audited:

```
SETROPTS  LOGOPTIONS(ALWAYS(DEVICES))
```

With this command, auditing is done every time a user allocates a unit record, teleprocessing, or graphic device, whether the device is protected by a profile or not, and whether the profile specifies auditing or not. To reset logging to be controlled by profiles, use the following command:

```
SETROPTS  LOGOPTIONS(DEFAULT(DEVICES))
```

Figure 19 shows the effect that LOGOPTIONS has when used with the OPERCMDS class. The OPERCMDS class is active, there are no profiles defined in that class, and LOGOPTIONS(ALWAYS(OPERCMDS)) is in effect. The sequence of events is as follows:

1. TSO user SYSPROG issues DISPLAY ACTIVE (D A,L) command through the TSO OPER facility. The event is logged with REASON(LOGOPTIONS).
2. A profile for the DISPLAY command is defined in the class OPERCMDS with AUDIT(FAILURES(READ)).
3. User SYSPROG is denied access to the DISPLAY command.
4. The class has to be refreshed, because RACLIST processing has been activated for the class.
5. SETROPTS LOGOPTIONS(DEFAULT(OPERCMDS)) is defined to determine the logging options from the profiles in the OPERCMDS class.
6. User SYSPROG issues the display command from SDSF, the command is not executed, and the event is logged because of the profile.
7. The universal access to the profile is changed to READ.
8. User SYSPROG is given READ access authority in the profile.
9. The OPERCMDS class is refreshed.
10. User SYSPROG issues the DISPLAY command, which is not logged because the profile has AUDIT(FAILURES) and LOGOPTIONS(DEFAULT(OPERCMDS)) is in effect.

11. Logging options are changed to be controlled from the LOGOPTIONS operand and all accesses are to be logged.

12. USER82, who is a console operator logged on to the console, issues the DISPLAY command and the event is logged with REASON(LOGOPTIONS).

```
90.061 18:02:16                          RACF REPORT - LISTING OF PROCESS RECORDS                                  PAGE  18
                                                          E
                                                          V  Q
                                                          E  U
                          *JOB/USER  *STEP/     --TERMINAL--  N  A
 DATE     TIME    SYSID    NAME     GROUP        ID     LVL  T  L
90.061 17:45:19 SMF6   SYSPROG   SYS1      C6LOC5E4    0  7  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=SYSPROG
Step 1       TEST USER                                AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                              LOGSTR=¢D A,L¢
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                              OPERCMDS=MVS.DISPLAY.*,LEVEL=00
90.061 17:45:19 SMF6   SYSPROG   SYS1      C6LOC5E4    0 21  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=SYSPROG
Step 2       TEST USER                                AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                RDEFINE OPERCMDS MVS.DISPLAY.* OWNER(SYSPROG) UACC(NONE) LEVEL(00)
                                                                AUDIT(FAILURES(READ)) NONOTIFY
90.061 17:45:19 SMF6   SYSPROG   SYS1      C6LOC5E4    0 19  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=SYSPROG
Step 3       TEST USER                                AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                PERMIT MVS.DISPLAY.* CLASS(OPERCMDS) ID(SYSPROG) ACCESS(NONE)
90.061 17:45:27 SMF6   SYSPROG   SYS1      C6LOC5E4    0 24  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=
Step 4       TEST USER                                AUTH=(SPECIAL,OPERATIONS),REASON=(COMMAND)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                SETROPTS REFRESH RACLIST(OPERCMDS)
90.061 17:45:55 SMF6   SYSPROG   SYS1      C6LOC5E4    0 24  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=
Step 5       TEST USER                                AUTH=(AUDITOR),REASON=(COMMAND)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                SETROPTS LOGOPTIONS(DFLT(OPERCMDS))
90.061 17:46:31 SMF6   SYSPROG   SYS1      C6LOC5E4    0  2  1  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=SYSPROG
Step 6       TEST USER                                AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                              LOGSTR=¢D A,L¢
                                                              USER SECLABEL=SYSHIGH
                                                              OPERCMDS=MVS.DISPLAY.JOB,GENPROF=MVS.DISPLAY.*,LEVEL=00,INTENT=READ,
                                                              ALLOWED=NONE
90.061 17:48:27 SMF6   SYSPROG   SYS1      C6LOC5E4    0 20  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=SYSPROG
Step 7       TEST USER                                AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                RALTER OPERCMDS MVS.DISPLAY.* UACC(READ)
90.061 17:48:27 SMF6   SYSPROG   SYS1      C6LOC5E4    0 19  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=SYSPROG
Step 8       TEST USER                                AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                PERMIT MVS.DISPLAY.* CLASS(OPERCMDS) ID(SYSPROG) ACCESS(READ)
90.061 17:48:34 SMF6   SYSPROG   SYS1      C6LOC5E4    0 24  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=
Step 9       TEST USER                                AUTH=(SPECIAL,OPERATIONS),REASON=(COMMAND)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                SETROPTS REFRESH RACLIST(OPERCMDS)

Step 10  DISPLAY COMMAND IS NOT LOGGED

90.061 17:49:52 SMF6   SYSPROG   SYS1      C6LOC5E4    0 24  0  JOBID=(SYSPROG 90.061 13:04:50),USERDATA=(),OWNER=
Step 11      TEST USER                                AUTH=(AUDITOR),REASON=(COMMAND)
                                                              USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=C6LOC5E4
                                                                SETROPTS LOGOPTIONS(ALWS(OPERCMDS))
90.061 17:50:25 SMF6   USER82    GSMVS               0  2  0  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=SYSPROG
Step 12      USER OF GSMVS GROUP                      AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                              LOGSTR=¢D A,L¢
                                                              USER SECLABEL=SESY
                                                              OPERCMDS=MVS.DISPLAY.JOB,GENPROF=MVS.DISPLAY.*,LEVEL=00,INTENT=READ,
                                                              ALLOWED=READ
```

Figure 19. Report Writer Listing when LOGOPTIONS Operand is Used

In steps 6 and 12 of the preceding example, the LOGSTR field in the report is a result of a new keyword on the RACROUTE macro. See 4.14.5, “New Audit Controls with RACROUTE Macro” on page 60 for details. For more information about the OPERCMDS class see 10.6, “Command Security in a JES2 Environment” on page 185.

4.14.4 Auditing Controlled Programs

With RACF 1.9, it is possible to audit attempted accesses to controlled programs by specifying the AUDIT operand on the RDEFINE or RALTER commands for controlled programs in the PROGRAM class. NOTIFY userid can also be specified for profiles defined in the PROGRAM class. The following example shows how to request auditing from controlled programs:

Load module SPCHECK in the load library USER22.TEST.LOAD is defined as a controlled program and auditing is requested for failed accesses to this program as follows:

```
SETROPTS  WHEN(PROGRAM)

RDEFINE   PROGRAM  SPCHECK  ADDMEM(¢USER22.TEST.LOAD¢/RACTS1/PADCHK)
          UACC(READ)  AUDIT(FAILURES(READ))  NOTIFY(RADMIN)
```

Figure 20 shows the auditing information produced after user USER22 tried to execute the program SPCHECK from USER22.TEST.LOAD.

```
90.206 14:36:06                          RACF REPORT - LISTING OF PROCESS RECORDS                          PAGE  25
                                                          E
                                                          V  Q
                                                          E  U
                          *JOB/USER  *STEP/     --TERMINAL--  N  A
 DATE     TIME    SYSID    NAME      GROUP        ID     LVL  T  L
90.206 14:30:03 SMF6    USER22     SYS1                  0   2  1  JOBID=(SYSPROG4 90.206 14:30:02),USERDATA=(),OWNER=SYSPROG
                        TESTUSER                                   AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                                   SESSION=INTERNAL READER BATCH JOB,JESINPUT=INTRDR,EXENODE=C5JES2,
                                                                   SUBMITTING USER=USER22,SUBMITTING NODE=C5JES2,SUBMITTING GROUP=SYS1
                                                                   PROGRAM=SPCHECK,LEVEL=00,INTENT=READ,ALLOWED=NONE
```

Figure 20. Report Writer Listing for Program Access Attempt

4.14.5 New Audit Controls with RACROUTE Macro

Two new RACROUTE operands for auditing are included in RACF 1.9:

- The LOGSTR operand is used to pass up to 255 bytes of text to RACF to be included in the type 80 SMF record. It can be used for AUTH, AUDIT, VERIFY, or VERIFYX requests. The Report Writer displays this additional data immediately below the area where current process records are displayed. In addition, the LOGSTR parameter data is available to the RACHECK pre- and post-processing exits. It is used, for example, by MVS in the OPERCMDS class to record the actual text of the MVS command issued. It is also used by JES in the JESSPOOL class to distinguish between SYSIN and SYSOUT data sets being created, deleted, or accessed.
- RACROUTE REQUEST=AUDIT is used to request auditing of an access to a resource without requesting an authorization check. It is used, for example, by VTAM to audit partner LU 6.2 verification; VTAM itself does the authorization checking, then requests that RACF audit the access. Refer to 13.3.2, “LU 6.2 Partner Verification Control” on page 231 for more information. A side effect of this operand is that a message is issued to the network security administrator. This operand is restricted to use by authorized, supervisor state, or system key callers and executes in key 0.

Other new RACROUTE macro parameters are described in 13.9, “New RACROUTE Macro Parameters” on page 254.

4.14.6 Report Writer Enhancements

The RACF Report Writer enhancements enable the security auditor to generate audit reports that include:

- SECLABELs
- Security tokens
- New reasons for logging

4.14.7 Auditing Summary

RACF 1.9 does not affect the following events, which are *always* logged:

- Every use of RVARY or SETROPTS commands.
- Every RACINIT failure.
- Every time the console operator grants access to a resource as part of failsoft processing.

Nor does RACF 1.9 affect the fact that the LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH commands are *never* logged. It does, however, complicate the decision process for those accesses that may or may not be logged. Figure 21 on page 62 can be used to determine whether an access attempt is to be logged.

4.14.8 Recommendations

Set LOGOPTIONS(ALWAYS) with care to avoid excessive amounts of logging that can impact the performance of the system. This applies specifically to an environment where system tasks are run with the TRUSTED attribute from the started procedures table, which means that they are to be audited. LOGOPTIONS is the only way to obtain auditing of trusted procedures.

LOGOPTIONS is also the only way to obtain auditing for classes for which profiles cannot be defined, such as TEMPDSN. When the TEMPDSN class is active, only the owning job step is authorized to access a temporary data set created in a job step. By setting LOGOPTIONS(FAILURES(TEMPDSN)) all failed accesses against temporary data sets can be audited. For additional information, see 13.4, “DFP-Managed Temporary Data Set Control” on page 235.

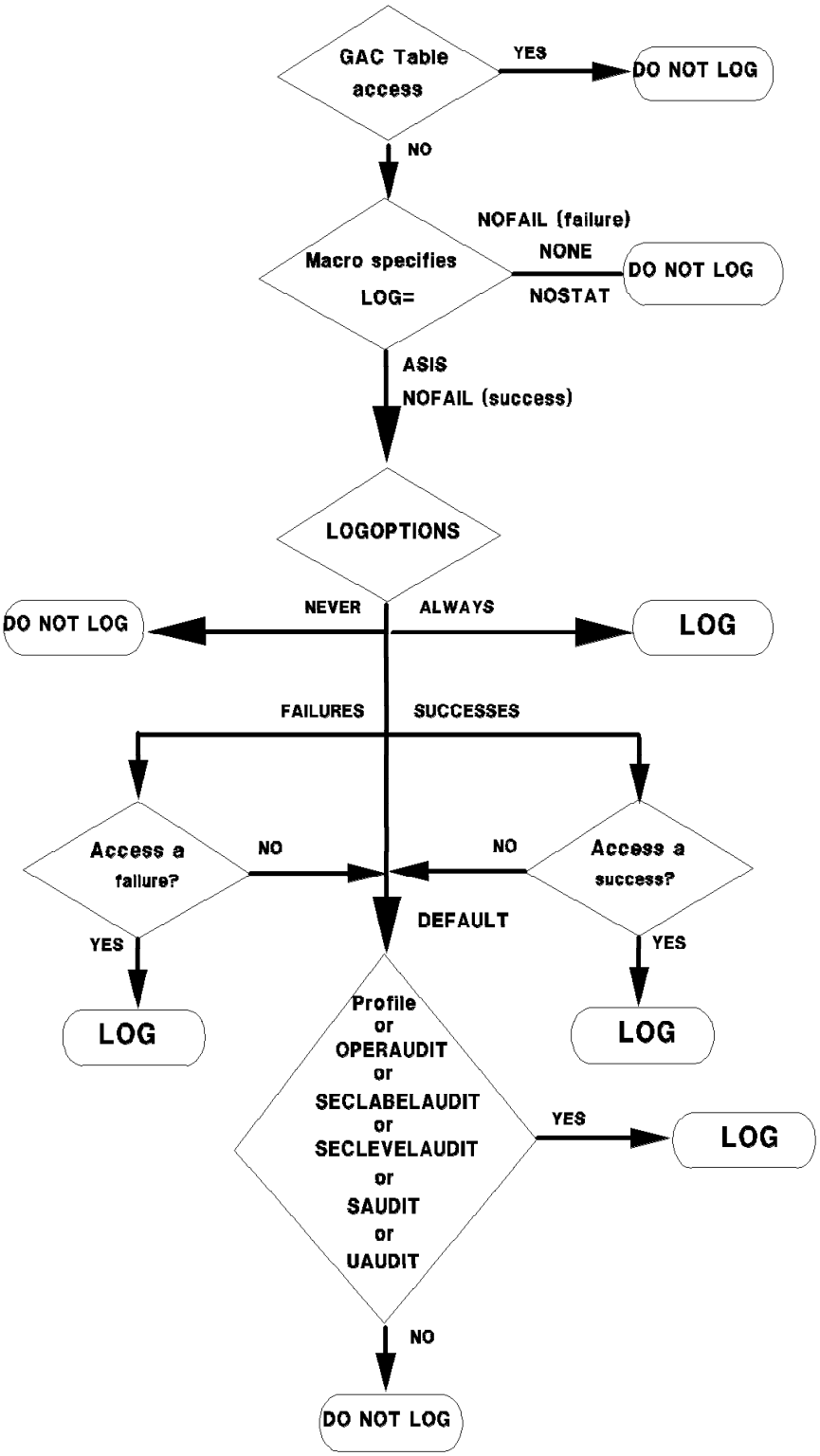

Figure 21. To Log or Not to Log?

Chapter 5. System Authorization Facility Interface

The system authorization facility (SAF) is part of the MVS/ESA operating system. SAF is present on an MVS system even when RACF is not. Before MVS 3.1.3, this was not significant, because SAF's only function was as a router between MVS and RACF; without RACF, SAF had no real function. However, with MVS 3.1.3, SAF's traditional role as a router has significantly changed. Now SAF establishes default security, provides security functions when RACF is not active, and performs propagation and token services.

Resource managers are responsible for calling SAF to determine whether to allow a user access to the system or to a resource. The resource manager is responsible for enforcing the decision made by SAF or RACF. Resource managers include:

- DADSM for data set access authority
- DFHSM for data set allocation authority
- CICS for CICS sign-on and transaction authorization
- JES for user identification and verification

Figure 22 illustrates the new SAF. Based on the original user's request, the resource manager formulates a request and passes it to SAF. Depending on the nature of the request, SAF may respond directly or may pass the request to RACF. In either case, the user receives a response from the resource manager after the resource manager considers the response from SAF.

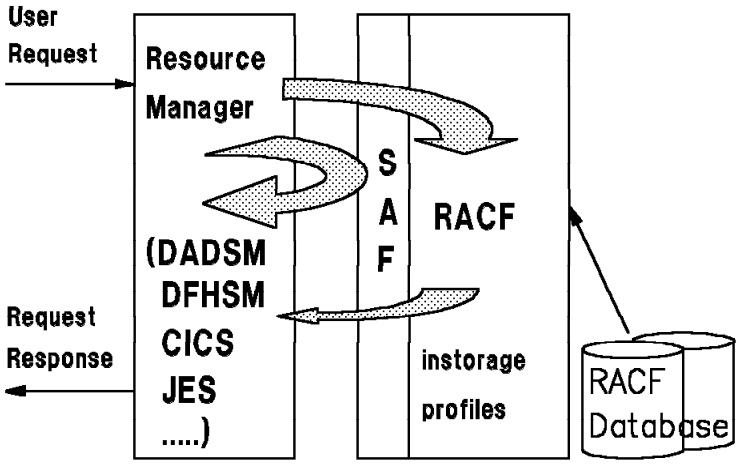

Figure 22. System Authorization Facility Overview

Access to the RACF database may not be required if the response can be formulated from profiles that are in storage. The new SAF functions include:

- Providing a security environment for all work. In order to provide this environment for system address spaces, SAF is initialized earlier.
- Building security tokens when RACF is not active or is downlevel.
- Supporting security token creation and maintenance. When a request does not require RACF action, SAF handles the request without invoking RACF.
- Propagating non-NJE userid and SECLABEL information. SAF now ensures that security related information for the unit of work is correctly propagated. RACF performs NJE propagation.

SAF also performs early verification and authorization of security related information such as userid, password, and SECLABEL. The advantage of this is that SAF and RACF perform propagation and early verification on a single pass, as opposed to a separate pass for each. The support for this resides in SAF because it must be available even when RACF is not.

5.1 New Security Environment

SAF provides a security environment for all work. Before MVS 3.1.3, the security environment described in Figure 23 was created by RACF when a unit of work was executed. This environment is represented by the RACF accessor environment element (ACEE) that describes the user and his authorities for the unit of work and was deleted when execution completed. This security environment is in effect only while the unit of work is in execution; it cannot protect the work before execution or after execution completes. It also cannot protect any output of the work that cannot be protected by a RACF profile; for example, SYSOUT data sets.

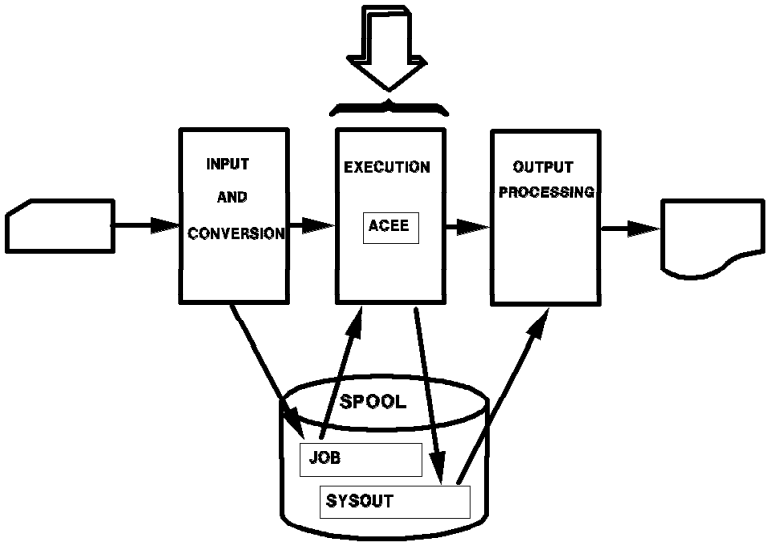

Figure 23. Pre-3.1.3 Security Environment

With MVS 3.1.3, the security environment described above is unchanged and still exists during execution. Additionally, SAF establishes the security environment described in Figure 24 on page 65 when a unit of work initially enters the system and is maintained by SAF until the last output created by the work is purged from the system. This environment is represented by the security token that, like the ACEE, describes the user and his authorities for the unit of work. When a unit of work enters the system, it can enter with a security token, or a security token can be assigned based on its origin. The security token is maintained through input and conversion, execution, and output processing. Any output created by the work is given the security token, which can be used to identify printed output or can be passed along with the data to another system.

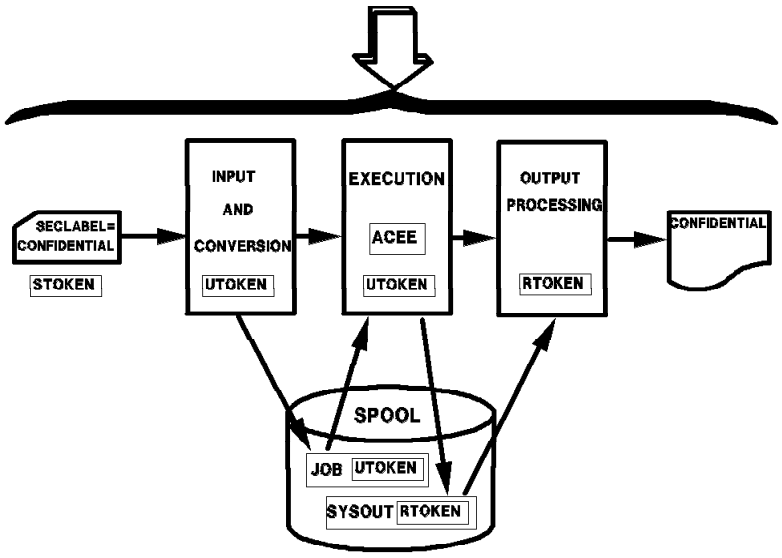

Figure 24. 3.1.3 Security Environment

5.1.1 SAF Early Initialization

For SAF to provide a security environment for address spaces prior to RACF initialization, it is necessary for SAF to be initialized earlier in NIP processing. As shown in Figure 25, SAF initialization is now a Resource Initialization Module (RIM) that is invoked right after IEAVNP05, the NIP module that builds LPA. Because many of the services that may be used by the SAF user exit are not available this early in NIP processing, a new user exit, ICHRTX01, is loaded and used as the SAF user exit from this point until the Master Scheduler initialization replaces its address with the address of the original SAF user exit, ICHRTX00. Because of the limited services available to the programmer and the risks involved in coding an exit that runs at this point in NIP processing, most installations should probably not implement an ICHRTX01 exit. With this implementation, existing ICHRTX00 exit routines begin getting control at the same point in Master Scheduler Initialization that they did with the old SAF.

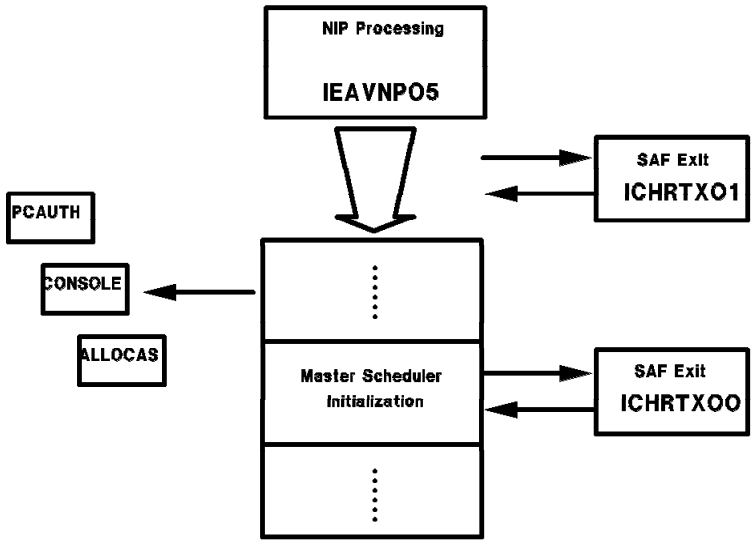

Figure 25. SAF Early Initialization

5.2 Security Tokens

A security token is a packet of security information. It is 80 decimal bytes long, 10 of which are currently reserved for expansion. The security token can be in either internal or external format and SAF provides the TOKENMAP service to convert from one to the other. The internal format is used by RACF and is encoded to prevent its use by application programs; the external format is provided for use by applications and can be mapped using the ICHRUTKN macro. Two formats are provided so that changes to the content or format of the internal security token do not affect any application code that uses the external format. The TOKENMAP conversion service can create any RACF version of the external security token from the current version of the internal security token to ensure compatibility. Figure 26 shows the contents of the external format security token. See 8.1.4, “Tokens in an NJE Environment” on page 120 for additional information.

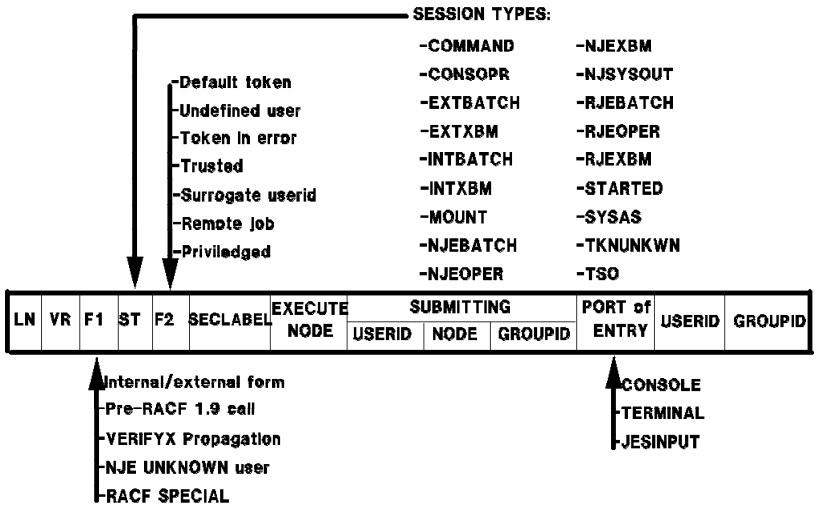

Figure 26. Security Token External Format

The RACF security token can be described as follows:

- Length - 80 (decimal).
- RACF version indicator.
- Flags to indicate the format of the token, its type, if the user is RACF special, if it is *trusted* or *privileged*, or whether it represents a pre-RACF 1.9 call that is used for compatibility mode.
- Session type to indicate if the token was created for a command, a system address space, an internal or external batch job, NJE or RJE, or for a started task.
- SECLABEL of whatever object the token protects.
- Node where the unit of work executed.
- Submitting node, userid, and groupid.
- Port of Entry where the unit of work originated. CONSOLE, TERMINAL, and JESINPUT are the new forms of conditional access.
- Userid that owns the token.
- Groupid is the current group for the userid, not necessarily the user’s default group.

5.2.1 Token Types

Although the fields within all tokens are identical, there are several types of tokens, each named for the way it is created and used, as follows:

UTOKEN A security token representing a user is called a user token or UTOKEN. It contains sufficient security information to identify the user and the security label of the job or session. It is a job token when a job attempts to access a resource. There are two cases where a UTOKEN is carried with a job before execution:

- If a job is re-routed to a different execution node
- If a job is off-loaded before execution and then re-loaded

A UTOKEN is fixed for the life of the job or session, from JES input services until the job or session is purged from the JES queues after all printing is complete. A job or session cannot change either the security levels or categories associated with the token. If this were not the case, and a job or session could change its label, a security violation may occur if the new label allows an access that the old label would not have allowed. MVS functions that use a UTOKEN include:

- **Job submission:** JES calls SAF to verify the user and obtain a UTOKEN that JES saves in a JES control block for the life of the job or session. This UTOKEN is used when JES creates SYSIN data sets on behalf of the job or session.
- **Job submission from a job or session:** When the JES internal reader is used to submit a job, JES passes the submittor′s UTOKEN to SAF along with information from the job card. SAF can then prevent a user from submitting a job with a lower security label, which could declassify information.
- **Job execution:** JES passes the UTOKEN to the initiator, which calls SAF to create the security environment for the execution of the job, to create the ACEE.
- **SYSOUT processing:** JES makes the UTOKEN for a job available to PSF/MVS. PSF/MVS uses the token to verify a user′s authorization to suppress labeling.
- **Command processing:** The UTOKEN of the console operator or TSO session is passed to SAF to verify the user′s authority to issue the command.

RTOKEN A security token representing a resource is called a resource token or RTOKEN. It is identical to the UTOKEN of the user that created the resource. The RTOKEN is used for the following types of resources:

- SYSIN data sets
- SYSOUT data sets
- Messages passed across address spaces
- Messages transmitted through the TSO mail log facility
- JESNEWS data sets

In each case, SAF returns the RTOKEN to the resource manager when the resource is created. The resource manager passes the RTOKEN with the resource name on all subsequent calls to RACF for access control.

STOKEN A security token representing the submittor of a job is called a submittor token or STOKEN. This type of token is usually very short lived when attached to a a batch job submitted by a TSO user, before it is transformed into a user token. It is also transmitted with an NJE job to the execution node.

Default A default token is a skeleton security token created by SAF when there is no security product or when the security product is not active. If the security product is not active, SAF returns a default token when invoked with REQUEST=VERIFYX. A bit in the token indicates that this is a default token. In cases where a default token is returned and a valid SECLABEL was not specified on the call, and no default could be determined from the RACF profiles, a default SECLABEL of SYSHIGH is returned if the resulting default token is trusted; SYSLOW is returned if the token is not trusted. If a token is a default token and not trusted, RACINITs and RACHECKs against this token fail and the failure is audited.

When SAF is invoked with REQUEST=VERIFY and SESSION=SYSAS, SAF returns a default ACEE. Currently, address spaces created prior to RACF initialization do not have an ACEE; when RACF becomes active and a RACHECK or other RACF function is invoked, RACF issues an abend with an appropriate completion code since there is no ACEE. With MVS 3.1.3, SAF ensures that every address space has an ACEE so that when (if) the security product becomes active, RACHECK processing can be performed as usual. When the security product is not active, a default token is returned on a RACROUTE REQUEST=VERIFYX. A default ACEE, built by SAF, is returned on a RACROUTE REQUEST=VERIFY, and:

- The default token is flagged as belonging to a system address space.
- An additional call must be made to SAF to delete the default ACEE.
- The userid for the master address space is **+MASTER+**; for other address spaces it is **+address space name**, for example:

  ```
  +ALLOCAS
  +CONSOLE
  +SMF
  ```

Unknown user The unknown user token is a special security token used at store-and-forward nodes, or created when the user is undefined on the SYSOUT destination node. In order to create a security environment for NJE, SAF must build an unknown user token.

An unknown user token is built implicitly by RACROUTE REQUEST=VERIFYX if the user is not a RACF defined user, or explicitly if SESSION=TKNUNKWN is coded on the VERIFYX request. If a SECLABEL is not provided on the VERIFYX request, the default SECLABEL is SYSHIGH if the resulting token is trusted and SYSLOW if it is not trusted. When JES specifies SESSION=TKNUNKWN, it specifies SYSHIGH as the SECLABEL. If RACF is installed, the defaults for both local and NJE unknown users can be changed by the security administrator. The system defaults are as follows:

- ++++++++ for local jobs. It can be changed using:

  ```
  SETROPTS   JES(UNDEFINEDUSER(userid))
  ```

- ???????? for NJE. It can be changed using:

  ```
  SETROPTS   JES(NJEUSRID(userid))
  ```

Defaults appear only in external form and cannot be authorized for anything. They are treated as undefined users on RACHECK and allowed access based on the UACC in the profile. The defaults are used to build the data set names for JES spool data sets that can be used on the PERMIT command to allow access.

Note: The unknown user token is not the same as the default token.

Error A error token is a token that is returned when the security product cannot validate the user. The job is flushed and RACF messages are written to JESMSGLG. An error token is created to allow the JESMSGLG to be viewed and purged; neither of which is possible if there is no security token.

Figure 27 shows the major changes that were made to SAF to provide security tokens. SAF creates and maintains tokens in order to maintain a security environment. To take advantage of this new security environment, the following requests are made with the RACROUTE macro:

VERIFYX The VERIFYX request creates a UTOKEN when work first enters the system. With MVS 3.1.3, VERIFYX replaces JES EARLYVERIFY processing. The input passed to SAF on this call includes:

- Userid
- Group
- Execution node
- Security label
- Address to return the created UTOKEN
- Trusted user indicator

This call verifies the userid and password to determine whether the user is known to RACF. Once the user is verified, a UTOKEN is built and returned to the caller. This UTOKEN exists for the life of the session. If a security product is not active, SAF builds a default UTOKEN to satisfy the request.

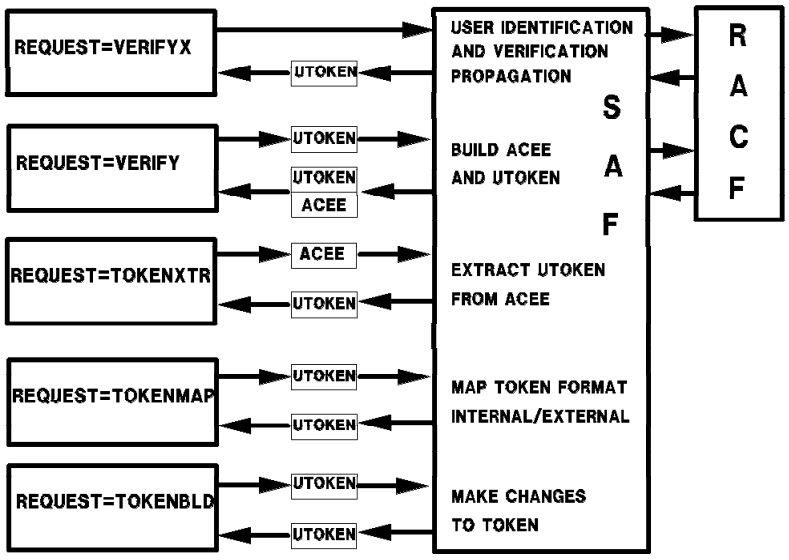

Figure 27. SAF Token Support

VERIFY The VERIFY request, an existing RACROUTE call, accepts as input a UTOKEN and new keywords, and returns both an ACEE and UTOKEN to the caller. If a security product is not active, SAF builds a default ACEE and UTOKEN to satisfy the request.

TOKENXTR The TOKENXTR request extracts a UTOKEN, in internal format, from the task or address space ACEE. Any information not available in the ACEE is returned as blanks. This UTOKEN can be used as input to subsequent RACROUTE macro calls. SAF performs this function without RACF action.

TOKENBLD The TOKENBLD request modifies an existing token. The modification does not change the existing token; a new token is built from fields in the existing token and the passed parameters. The following order of priority exists when placing fields in the new token:

- Keywords specified on the request
- Fields within the STOKEN keyword
- Fields within the existing token

SAF performs this function without RACF action.

TOKENMAP The TOKENMAP request converts an internal- or external-format token to a specific version of an internal or external format token. Internal format refers to the format received from a RACROUTE REQUEST=VERIFYX or RACROUTE REQUEST=TOKENXTR. External format refers to that which is mapped by the ICHRUTKN macro. RACROUTE REQUEST=TOKENMAP is the only interface to obtain an external-format token.

The primary purpose of this function is to allow a caller to reference individual fields within the token, regardless of the current version of the token being used by SAF. The caller provides SAF with the length and version of the token to be converted and SAF maps it correctly.

5.2.2 SAF Propagation

SAF performs non-NJE propagation and early verification. All jobs are verified at input time and early verification is no longer an option. RACF performs propagation for NJE.

A new SAF function handles security information propagation and verification in a manner consistent with current IEFCMAUT processing. The user's security information is extracted from the current ACEE by JES using a RACROUTE REQUEST=TOKENXTR. The information is placed in an STOKEN that is passed into a RACROUTE REQUEST=VERIFYX request. The SAF VERIFYX function actually performs the propagation, which was previously done by IEFCMAUT and the scheduler (with RACINIT at execution time). In processing errors, SAF builds and return messages corresponding to those formerly issued by IEFCMAUT and the scheduler.

If RACF is down-level, SAF still performs propagation. SAF changes VERIFYX requests to VERIFY requests so that they can be satisfied as usual by the down-level RACF product.

Chapter 6. Implementing Security on Job Entry Subsystems

This chapter and the following five chapters discuss the implementation of security in JES2 Release 3.1.3, JES3 Release 3.1.3, and RACF Release 1.9. JES Release 3.1.3 refers to both JES2 and JES3. With the enhancements made to JES 3.1.3, you can control and audit resources managed by JES. The resources that you can protect include:

- Jobs entering the system by input source.
- Jobs entering the system by specific job name.
- Jobs entering the system through surrogate users.
- Spool data sets, including SYSIN and SYSOUT.
- JES commands entered by operators.
- Access to consoles, both local and remote.
- Access to output devices, including printers, punches, nodes, remote workstations, and spool offload functions.
- NJE security - controlling jobs and SYSOUT between JES2, JES3, VM/RSCS, and VSE/POWER.
- Guaranteed print labeling to PSF/MVS controlled printers.

RACF 1.9 has been enhanced to provide new resource classes for the protection of these resources. All these new classes are discussed in detail in the remainder of this document.

Note: Once you decide to protect a resource, you **must** define your jesname in the RACF Started Procedures Table with the trusted attribute. See 4.5.2, “Started Procedures Table” on page 35 for additional information.

6.1 JES Release 3.1.3 Changes

There are several changes in both JESes that affect security. Many are discussed in detail in this chapter.

6.1.1 Job Message Data Set Name Changes

The message data sets printed for every job have new names in this release. These new names are shown in Table 7:

Table 7. JES Message Data Set Name Changes

JES2 Old Name	JES3 Old Name	JES2/JES3 New Name
$JES2LOG	JESMSG	JESMSGLG (note)
$JCLIMG	JESJCL	JESJCL
$SYSMSGS	SYSMSG	JESYSMSG

Note: Used for security related messages when calls are made to SAF/RACF.

6.1.2 JES2 General Purpose Subtasks

Many security calls cause an MVS WAIT to occur. Therefore, a structural change to JES2 provides general-purpose subtasks to perform these calls. These subtasks are created at JES2 initialization. A default of ten subtasks are created. To specify a number other than the default, you must change the following initialization statement and hot-start JES2:

```
SUBTDEF   GSUBNUM=n
```

n specifies a number from 1 to 50.

Note: JES3 already has general subtasks and requires no additional definitions.

6.2 RACF Resource Classes Used by JES

JES uses the RACF resource classes listed below. To activate RACF protection, the resource class must be made active and profiles defined to grant users access to the resource:

- **JESINPUT** - JESINPUT in RACF permits conditional access support for commands or jobs that are entered into the system from a JES input device. After creating at least one profile to allow access, activate the JESINPUT class:

  ```
  SETROPTS CLASSACT(JESINPUT)
  ```

- **JESJOBS** - JESJOBS in RACF permits an installation to control who can submit jobs by job name. For example, you can permit certain job names to be entered only by certain userids or groups. This class is also used by TSO to cancel jobs by job name. After creating at least one profile to allow access, activate the JESJOBS class:

  ```
  SETROPTS CLASSACT(JESJOBS)
  ```

- **SURROGAT** - SURROGAT in RACF allows an installation to establish surrogate userids. User1 can submit jobs on behalf of another user, user2, without having to specify user2′s password. Jobs submitted in this way by user1 execute with the authority of user2. To activate the SURROGAT class, specify:

  ```
  SETROPTS CLASSACT(SURROGAT)
  ```

- **NODES** - NODES in RACF controls whether jobs and SYSOUT can enter the system from other nodes. Also, RACF can control whether the jobs entering the system from other nodes need userid and password verification checking. The NODES class is also used to translate userids, groups, and SECLABELs. To activate the NODES class, specify:

  ```
  SETROPTS CLASSACT(NODES)   RACLIST(NODES)
  ```

- **WRITER** - WRITER in RACF controls where output can be sent. You can restrict or authorize the use of writers for local printers and punches, RJE devices, and NJE transmissions. For NJE, RACF verifies the security of jobs and SYSOUT transmissions to ensure that the user is authorized to send data to another node in a network. After creating at least one profile to allow access, activate the WRITER class:

  ```
  SETROPTS CLASSACT(WRITER)
  ```

- **JESSPOOL** - JESSPOOL in RACF permits an installation to protect the data that resides on the JES spool. It prevents unauthorized users from reading, modifying, printing, deleting, or copying a job′s data. After creating at least one profile to allow access, activate the JESSPOOL class:

  ```
  SETROPTS CLASSACT(JESSPOOL)
  ```

- **OPERCMDS** - OPERCMDS in RACF permits an installation to authorize a userid to a command or group of commands by creating a RACF user profile for the console and placing the console′s or

operator's userid in the access list of the OPERCMDS profile. Profiles can be defined in the OPERCMDS class to authorize all operator commands. To activate the OPERCMDS class, specify:

```
SETROPTS CLASSACT(OPERCMDS)  RACLIST(OPERCMDS)
```

- **CONSOLE** - CONSOLE in RACF permits an installation to specify which consoles operators can use to enter certain commands. After creating at least one profile to allow access, activate the CONSOLE class:

  ```
  SETROPTS CLASSACT(CONSOLE)
  ```

- **FACILITY** - The FACILITY class is used by JES for remote workstations to force the user to enter a password to be checked by RACF. This class is also used for operator command checking for NJE and RJE/RJP. To activate the FACILITY class, specify:

  ```
  SETROPTS CLASSACT(FACILITY)
  ```

6.3 JES Exits for SAF Calls

JES calls SAF/RACF for all security decisions and auditing. If no decision is made, any existing default checking is used. Access to the SAF/RACF services is provided in each JES by a macro service. The $SEAS macro is used by JES2 and the IATXSEC macro is used by JES3. This single macro calls a new JES user exit before the SAF/RACF calls, calls SAF by issuing a RACROUTE, and calls a new JES user exit following the SAF RACHECK call to RACF.

These macros can be used in any JES exit to call SAF or RACF. The macro calls are shown in Figure 28.

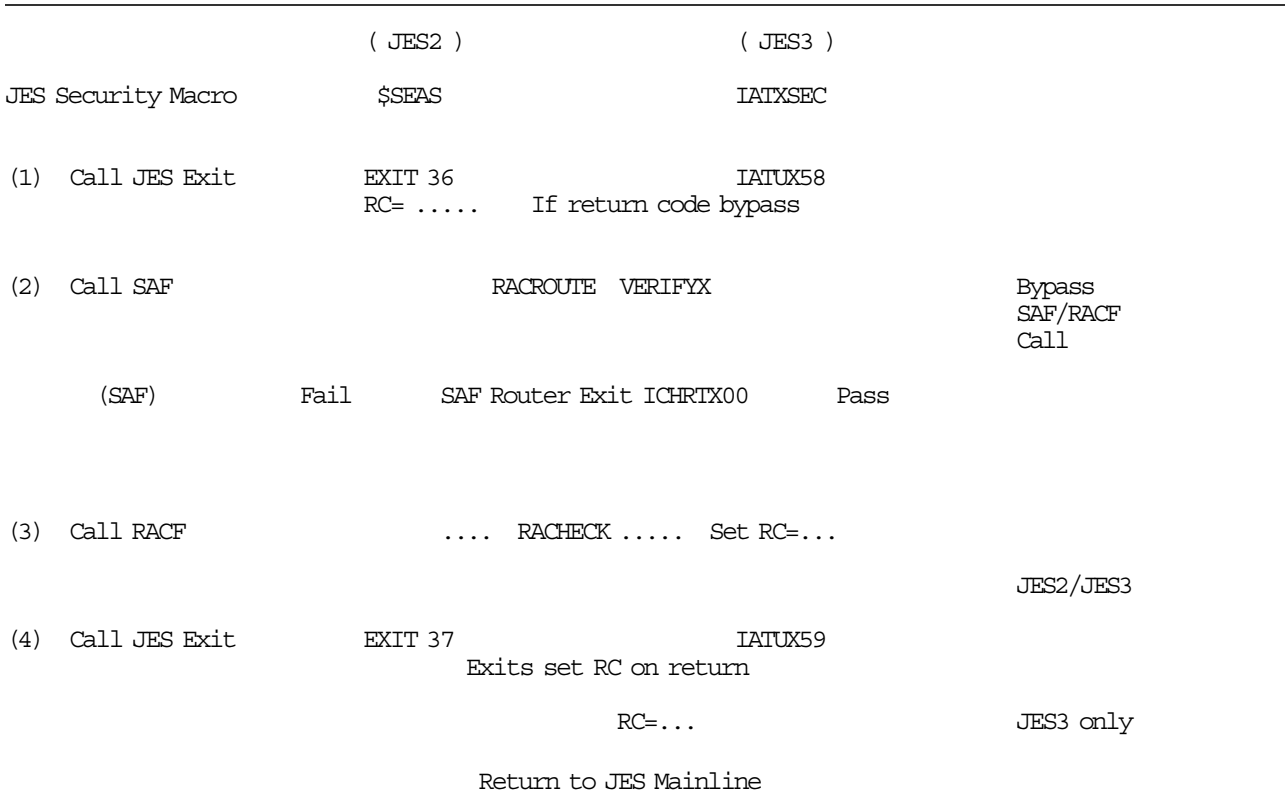

Figure 28. JES Security Exits

The pre-SAF exits are Exit 36 for JES2 and IATUX58 for JES3. The post-SAF exits are Exit 37 for JES2 and IATUX59 for JES3.

Note: You can change the information that JES passes to SAF in the pre-SAF exit. You can also change the results of the authorization check in the exits.

The macros work as follows (The numbers refer to Figure 28):

1. Call JES Exit

   JES2 User Exit 36 - This exit is taken prior to the SAF call. Within the exit, the installation can:

   - Bypass the SAF call and perform its own security checking. However, if the SAF call is bypassed, the installation must provide the expected information, which includes the security token.
   - Provide additional security checking.
   - Issue installation specified return and reason codes.
   - Disable the SAF security checking.

   JES3 User Exit IATUX58 - This exit is taken prior to the SAF call. Within the exit, the installation can:

   - Bypass the SAF call and perform its own security checking. However, if the SAF call is bypassed, the installation must provide the expected information, which includes the security token.
   - Allow security processing that does not involve SAF or RACF.
   - Reject installation security requests and not call SAF.
   - Allow security processing through RACF and IATUX59.
   - Allow security processing through RACF and bypass IATUX59.
   - Bypass all future calls to IATUX58.

2. Call SAF

   JES calls SAF for all security decisions and auditing through these new macros. All calls to SAF are executed under a general subtask. The subtask processing is initiated when JES issues a RACROUTE request.

   After processing the return code passed from the user exit (if processing is to continue), JES requests SAF services by using the RACROUTE macro. At JES input service, this request is a RACROUTE VERIFYX. This request is issued by JES for security authorization checking for job entry.

   Note: This request is not issued for started tasks, TSO logons, and operator mount requests.

   SAF - SAF either rejects or accepts the job:

   - If the job is rejected, an error token is returned. RACF messages are written to the job's JESMSGLG data set and the job is flagged to be flushed from the system. The JESMSGLG data set is the only output printed.
   - If the job is accepted, SAF propagates the current RACF userid, and, if used, the security label (SECLABEL) if this security information is not specified on the job statement. SAF also returns a UTOKEN.

3. Call RACF

SAF issues a RACHECK macro to pass the supplied security information specified by JES to RACF. RACF validates the security information and returns control to JES with a return code and default security information.

4. Call JES Exit

   JES2 User Exit 37 - This exit is taken following the SAF/RACF call. With this exit the installation can:

   - Examine and modify the return codes from SAF.
   - Request operator confirmation of a request.
   - Prohibit access or reduce the level of access.

   JES3 User Exit IATUX59 - This exit is taken following the SAF/RACF call. With this exit the installation can:

   - Examine and modify the return codes from SAF.
   - Allow security processing that does not involve SAF or RACF.
   - The installation can reject requests and disregard RACF security processing.
   - Accept RACF decisions.
   - Bypass all future calls to IATUX59.

6.3.1 JES2 User Considerations

If propagation functions are provided in your current JES2 system in Exits 2, 3, 4, or 20, this code must be changed to provide information in the SAFINFO parameter list passed in Exit 36. The $SAFINFO macro describes the parameters that are used to build the RACROUTE VERIFYX parameters.

If propagation is currently provided in JES2 exits for jobs that are submitted through RJE, local card readers, or NJE, you can change the session type in Exit 36 to provide propagation.

Note: For a coding example, see routine PROPRMT in Exit HASX036A in SYS1.SAMPLIB.

6.3.2 JES3 User Considerations

User exit IATUX58 can be used to modify security checks or make security decisions. The mapping macro, IATYSSX, can be used to change the parameters passed on the RACROUTE VERIFYX request.

6.4 RACF BATCHALLRACF Option

The BATCHALLRACF option specifies that RACF is to test all batch jobs for the presence of a userid and password in the job statement, or in SAF propagated identification information. SAF propagates the userid if it is missing from the job statement. RACF validates the users security information for all batch jobs submitted by the internal reader function. If this validation fails, JES does not allow the job to enter the system.

6.4.1 Operation

As an example, consider a user who is defined in SYS1.UADS but not to RACF, and RACF is initialized with the BATCHALLRACF option active. If this user submits a job, there is no RACF definition of the user and JES fails the job at input time based on the return code from RACF.

USERX is defined in SYS1.UADS using the TSO ACCOUNT command and the BATCHALLRACF option is activated:

```
SETROPTS  JES(BATCHALLRACF)
```

If USERX submits a job, the following notification is sent to the TSO session:

```
(JES2)  16.54.09 JOB02704 $HASP165 USERXA  ENDED AT C2JES2 - JCL ERROR

(JES3)  IAT6108   JOB  USERXA  (JOB02704)  FAILED BY SECURITY CHECK
```

The following message is placed into the job's JESMSGLG data set:

```
(JES2)  IRR008I JOB FAILED.  USER PARAMETER REQUIRED ON JOB STATEMENT.

(JES3)  ICH408I USER(DUMMY   ) GROUP(        ) NAME(???                  )
        LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
        IRR012I VERIFICATION FAILED.  USER PROFILE NOT FOUND.
```

Note: If the NOBATCHALLRACF option is active, the job submitted in this example is allowed to execute.

6.4.2 Considerations

The BATCHALLRACF option is used to force all batch users to be defined to RACF. This eliminates defining users in SYS1.UADS and is the recommended method for providing security in a well planned installation.

Specifying NOBATCHALLRACF allows jobs to enter the system and execute without RACF authorization. However, a userid specified on the job card is validated. If a userid is not specified, the job receives an unknown user token and the undefined user userid. Propagation can take place for jobs if the NOBATCHALLRACF option is in effect. An undefined user can access only unprotected resources or resources at the universal access level. For more information on undefined users, see section 5.2, “Security Tokens” on page 66.

Note: NOBATCHALLRACF is in effect during RACF initialization.

6.5 JES Job Validation

With JES Release 3.1.3, jobs are verified at job input time. The EARLYVERIFY function is a standard feature, whereas in previous releases of JES, it was an optional feature. The NOEARLYVERIFY command has no effect in JES Release 3.1.3.

Job validation occurs during the input phase of processing and consists of the following checks when an applicable resource class is activated as shown in Figure 29:

- USERID - GROUP - SECLABEL - PASSWORD (From Job card)
- Propagation of user's security to job
- Port of Entry (POE) - JESINPUT check

- Job name control - JESJOBS check
- Surrogate user support - SURROGAT check

A **UTOKEN** is returned after job validation.

When RACF is active, RACF ensures that the job password, userid, groupid, and security label are valid before allowing the job to be processed.

With the capability to create RACF profile definitions for job submission controls, code previously put into exits can be removed. Three new resource classes, JESINPUT, JESJOBS, and SURROGAT, have been added for job validation. These resource classes have control over the input source or Port of Entry (POE), job names, and submission of jobs on behalf of other users. You can protect the sources of job entry, such as internal readers, device readers, RJP, and network nodes. Authorizing network jobs and SYSOUT(NJE) is discussed in Chapter 8, “NJE Security Control” on page 117.

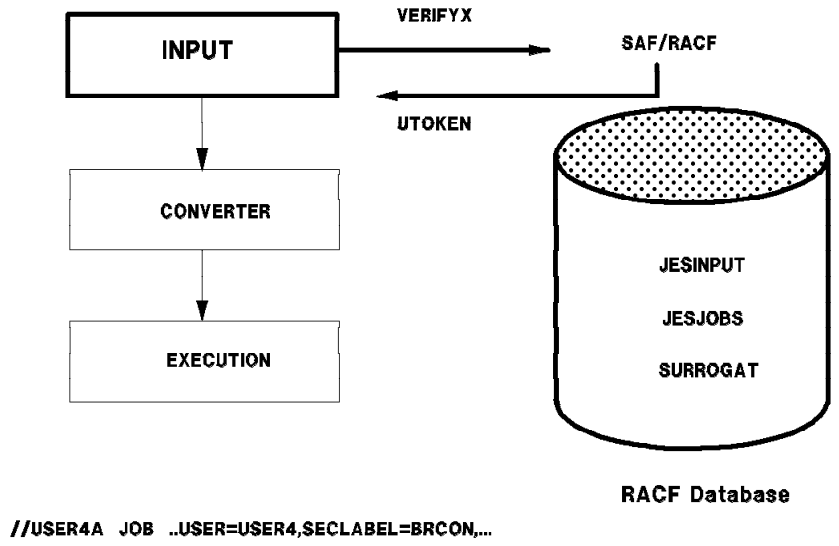

Figure 29. Job Validation at Input Processing

6.6 Propagation with SAF

SAF propagates the current RACF userid from each (already validated) RACF user who submits a batch job either by using the INTRDR or the TSO SUBMIT command. Thus, jobs executed within the same JES complex from which they are submitted, are automatically identified with the user and the user's RACF profile. The other user information that can be propagated is the SECLABEL.

Note: With JES 3.1.3 installed, propagation is now done by SAF rather than by IEFCMAUT.

If the SECLABEL class is active, SAF propagates the SECLABEL when the job is submitted through an internal reader or TSO, as described in this section. For more information on SAF in the context of propagation, see 5.2.2, “SAF Propagation” on page 70.

In order to determine what information is propagated in the following examples, the RACF LOGOPTIONS command is turned on to allow all SMF information to be recorded:

```
SETROPTS LOGOPTIONS(ALWAYS(JESINPUT JESSPOOL))
```

The RACF Report Writer program is used to interpret the records.

Example 1: No SECLABEL specified, propagated from user profile.

A TSO user is defined as USER4 with authorization to submit jobs. USER4 can submit jobs starting with any job name. USER4 has multiple SECLABELs defined and the default SECLABEL is NUNC. The SECLABELs to be used are NUNC and NCON. The 'USER=' statement is omitted from the job card and is propagated from the user's RACF profile. The job card is specified as follows:

```
//PAYAAID JOB (999,POK),¢USER4¢,MSGCLASS=T,CLASS=A
```

The Report Writer output shows the job passing through the INTRDR and ending with output on the spool:

```
90.053 16:05:02 SMF2   USER4     P0112              0  2  0  JOBID=(JES2 90.053 08:20:17),USERDATA=(),OWNER=P0112AW
                       ROBYN                                  AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                              USER SECLABEL=NUNC,SESSION=INTERNAL READER BATCH JOB,JESINPUT=INTRDR,
    ...Pay starting...                                        EXENODE=C2JES2,SUBMITTING USER=USER4,SUBMITTING NODE=C2JES2,
                                                              SUBMITTING GROUP=P0112
                                                              JESINPUT=INTRDR,LEVEL=00,INTENT=READ,ALLOWED=READ
90.053 16:05:02 SMF2   USER4     P0112              0  2  0  JOBID=(PAYAAID 90.053 16:05:01),USERDATA=(),OWNER=
                       ROBYN                                  AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                              LOGSTR=¢SYSOUT CREATE¢
    ...Pay writing spool...                                   USER SECLABEL=NUNC
                                                              JESSPOOL=C2JES2.USER4.PAYAAID.JOB04372.D0000101.?,LEVEL=00,INTENT=
                                                              UPDATE,ALLOWED=ALTER,RESOURCE SECLABEL=NUNC
```

Example 2: No SECLABEL specified, propagated from user profile.

In this example, USER4 submits a job that in turn submits a job through the internal reader. Propagation is maintained throughout the submission process.

Note: The only difference between Examples 1 and 2 is the method of job submission. JES makes no distinction for the POE. The POE is the INTRDR in both cases.

The following JCL was used:

```
//SUBJOB JOB (999,POK),¢TESTUSER¢,MSGCLASS=X,
//DUMMY  EXEC PGM=IEBGENER
//SYSPRINT  DD  SYSOUT=*
//SYSUT2    DD  SYSOUT=(A,INTRDR)
//SYSUT1    DD  DSN=USER4.JCL.CNTL(JOB6),DISP=SHR
//SYSIN     DD  DUMMY
//PAYAAID JOB (999,POK),¢USER4¢,MSGCLASS=T,CLASS=A
.
.
```

```
90.053 16:05:53 SMF2   USER4     P0112              0  2  0  JOBID=(JES2 90.053 08:20:17),USERDATA=(),OWNER=P0112AW
                       ROBYN                                  AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                              USER SECLABEL=NUNC,SESSION=INTERNAL READER BATCH JOB,JESINPUT=INTRDR,
..Subjob starting..                                       EXENODE=C2JES2,SUBMITTING USER=USER4,SUBMITTING NODE=C2JES2,
                                                              SUBMITTING GROUP=P0112
                                                              JESINPUT=INTRDR,LEVEL=00,INTENT=READ,ALLOWED=READ
90.053 16:05:55 SMF2   USER4     P0112              0  2  0  JOBID=(JES2 90.053 08:20:17),USERDATA=(),OWNER=P0112AW
                       ROBYN                                  AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                              USER SECLABEL=NUNC,SESSION=INTERNAL READER BATCH JOB,JESINPUT=INTRDR,
..Pay starting..                                          EXENODE=C2JES2,SUBMITTING USER=USER4,SUBMITTING NODE=C2JES2,
                                                              SUBMITTING GROUP=P0112
                                                              JESINPUT=INTRDR,LEVEL=00,INTENT=READ,ALLOWED=READ
90.053 16:05:56 SMF2   USER4     P0112              0  2  0  JOBID=(PAYAAID 90.053 16:05:54),USERDATA=(),OWNER=
                       ROBYN                                  AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                              LOGSTR=¢SYSOUT CREATE¢
    ...Pay writing spool...                                   USER SECLABEL=NUNC
                                                              JESSPOOL=C2JES2.USER4.PAYAAID.JOB04375.D0000101.?,LEVEL=00,INTENT=
                                                              UPDATE,ALLOWED=ALTER,RESOURCE SECLABEL=NUNC
```

Example 3: A SECLABEL is specified, but only the userid is propagated.

In this example, USER4 submits with a SECLABEL=NCON specified in the job card. USER4 has been permitted to use the SECLABEL NCON. All the user information is propagated. The following JCL was used:

```
//PAYAAID JOB (999,POK),¢USER4¢,MSGCLASS=T,CLASS=A,
//         SECLABEL=NCON
```

```
90.053 16:05:27 SMF2   USER4     P0112          0  2  0  JOBID=(JES2 90.053 08:20:17),USERDATA=(),OWNER=P0112AW
                       ROBYN                              AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                          USER SECLABEL=NCON,SESSION=INTERNAL READER BATCH JOB,JESINPUT=INTRDR,
    ...Pay starting...                                    EXENODE=C2JES2,SUBMITTING USER=USER4,SUBMITTING NODE=C2JES2,
                                                          SUBMITTING GROUP=P0112
                                                          JESINPUT=INTRDR,LEVEL=00,INTENT=READ,ALLOWED=READ
90.053 16:05:28 SMF2   USER4     P0112          0  2  0  JOBID=(PAYAAID 90.053 16:05:27),USERDATA=(),OWNER=
                       ROBYN                              AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                          LOGSTR=¢SYSOUT CREATE¢
    ...Pay writing spool...                               USER SECLABEL=NCON
                                                          JESSPOOL=C2JES2.USER4.PAYAAID.JOB04373.D0000101.?,LEVEL=00,INTENT=
                                                          UPDATE,ALLOWED=ALTER,RESOURCE SECLABEL=NCON
```

6.7 JESINPUT Class

With the RACF JESINPUT class, you can protect all input sources to the system, including internal readers, device readers, nodes, and RJE/RJP workstations. JES Input Service performs early validation of a job. The submittor's UTOKEN is used by JES input service. Figure 30 shows the input sources for jobs entering an MVS/JES system.

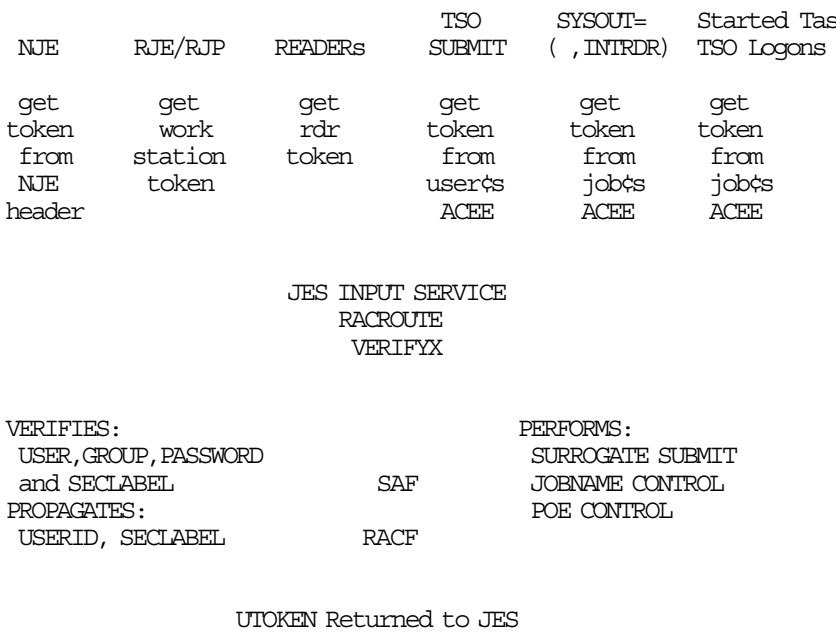

Figure 30. JES Input Sources

For example, you might want to prevent certain users from submitting jobs through particular RJE/RJP workstations or particular nodes. To do this, you have to activate the JESINPUT class and define profiles to restrict access for those users. The input source is controlled by a Port of Entry (POE) name in the RACF profile definition. The format of the JESINPUT class profile name is:

poe-name

6.7.1 JES Device POE Names

The Port of Entry (POE) controls a user's authority to submit batch jobs through specific JES input devices. The valid POE names are shown in Table 8.

Note that there is no distinction made between INTRDR submits by an executing job and a TSO submit command. The INTRDR poe name is used in both cases.

Table 8. JES POE Names

Device	JES2 POE Name	JES3 POE Name
JES reader	RDRnn	Jname of reader
Disk reader	n/a	JES3DRDS
RJE/RJP reader	RnnnnRDm (Note)	Workstation name
NJE reader (BSC)	Adjacent nodename	Adjacent nodename
NJE reader (SNA)	Adjacent nodename	NJERDR
Dump job	n/a	DUMPJOB
Spool offload	OFFn.JR	n/a
Internal reader	INTRDR	INTRDR
TSO SUBMIT	INTRDR	INTRDR

Note: For JES2 reader names where the nnn is less than 999, a period is required, for example; Rnnn.RDm.

6.7.2 JESINPUT Profile Definitions

This section includes examples of the use of RACF profiles to control users submitting jobs to the system. The profile name for the JESINPUT definitions is the POE name. These names are shown in Table 8. The JESINPUT class is defined by:

```
RDEFINE  JESINPUT  poe-name   UACC(....)
PERMIT   poe-name  CLASS(JESINPUT)  ID(user or group)  ACCESS(....)
```

Only the following access authorities are meaningful:

NONE Specifies that the input device can be used only by those users explicitly defined in the access list.

READ The minimum authority required for access to the devices.

6.7.3 JESINPUT Class Considerations

If your installation allows all jobs to enter the system regardless of the input source, then you do not have to activate the JESINPUT class.

If you wish only to audit all jobs entering the system, without restricting jobs entering from any source, specify:

```
SETROPTS   CLASSACT(JESINPUT)
SETROPTS   LOGOPTIONS(ALWAYS(JESINPUT))
```

Note: Auditing requires the activation of the JESINPUT class. Therefore, a profile must exist in order to allow jobs to be submitted:

```
RDEFINE  JESINPUT  *  UACC(READ)
```

If your installation wishes to restrict jobs that enter the system based on the input source or a specific user or group of users, then you must define profiles to restrict submission. These considerations are discussed in the next sections on the JESINPUT class.

Note that there are two approaches when defining profiles:

- Restict or deny all users from the resource by:

  ```
  RDEFINE   JESINPUT   poe-name   UACC(NONE)
  ```

  Then permit users access to the resource:

  ```
  PERMIT   poe-name   CL(JESINPUT)   ID(user or group)   ACC(READ)
  ```

- Allow all users access to the resource by:

  ```
  RDEFINE   JESINPUT   poe-name   UACC(READ)
  ```

  Then deny a user or group access to the resource:

  ```
  PERMIT   poe-name   CL(JESINPUT)   ID(user or group)   ACC(NONE)
  ```

The decision for the JESINPUT class depends on how many profiles have to be created to allow or deny access. If you want all users to be able to submit jobs with the INTRDR poe-name, then you probably do not want to use the first approach, above. You would need a PERMIT for every user or group. Choose the option that is easiest.

6.7.4 JESINPUT Class for Internal Readers

The internal reader POE name is INTRDR. The input source control is for jobs submitted by the TSO SUBMIT command under ISPF or by an executing job that submits an input stream through a DD statement in its JCL.

In the following example, the RDEFINE allows all users to use the internal reader. With RACF, a PERMIT command is allowed only if a profile exists for the profile name. Jobs submitted for userid, USER4, are restricted from the INTRDR input source by the PERMIT statement:

```
RDEFINE   JESINPUT   INTRDR   UACC(READ)
PERMIT   INTRDR   CLASS(JESINPUT)   ID(USER4)   ACCESS(NONE)
```

If a job for USER4 is submitted through the INTRDR, the submittor receives the following notification on the TSO screen:

```
        //USER4ID JOB (999,POK),¢USER4¢,USER=USER4,NOTIFY=USER4

(JES2)  16.54.09 JOB02704 $HASP165 USER4ID  ENDED AT C2JES2 - JCL ERROR

(JES3)  IAT6108   JOB  USER4ID  (JOB02704)  FAILED BY SECURITY CHECK
```

The following message is placed into the job's **JESMSGLG** data set in JES2 and JES3 systems:

```
ICH408I USER(USER4   ) GROUP(USER    ) NAME(                    )
  LOGON/JOB INITIATION - NOT AUTHORIZED TO TERMINAL/CONSOLE
```

6.7.5 JESINPUT Class for Local Readers

Access to the physical readers can be protected by coding the JESINPUT class for local readers, for example, RDR1.

In the following example, RDR1 is controlled by not allowing the submission of jobs for all users defined to RACF. To allow access for the submission of jobs through RDR1, use a PERMIT command with an ACCESS of READ. The PERMIT statements allow OPER1 and the group TESTGRP to submit jobs using RDR1.

```
RDEFINE   JESINPUT   RDR1   UACC(NONE)
PERMIT   RDR1   CLASS(JESINPUT)   ID(OPER1)   ACCESS(READ)
PERMIT   RDR1   CLASS(JESINPUT)   ID(TESTGRP)   ACCESS(READ)
```

The ID field can specify a group name you want to authorize or restrict from the input source.

A TSO user defined as OPER2, submits a job with the following job card:

```
//OPER2A  JOB (999,POK),¢OPER2¢,USER=OPER2
```

If a job for OPER2 is submitted through RDR1, the submittor receives the following notification in the JESMSGLG data set in the job's output:

```
ICH408I USER(OPER2   ) GROUP(USER    ) NAME(                    )
  LOGON/JOB INITIATION - NOT AUTHORIZED TO TERMINAL/CONSOLE
```

For local readers, there is a requirement that the userid be specified on the job statement. There is no propagation of userid associated with the input source token of the operator who started the reader device. Job statements that do not have a userid specified are assigned a default userid as discussed in 5.2, "Security Tokens" on page 66.

6.7.6 JESINPUT Class for RJE/RJP Readers

Access to the RJE/RJP stations can be protected by coding the JESINPUT class for RJE/RJP stations. The RJE/RJP remote terminal name is specified as the profile name in the resource class:

- Jobs submitted through remote terminal readers require a valid userid and password on the job statement. Propagation does not occur for these types of input devices.
- In SYS1.SAMPLIB member HASX36A of a JES2 system, sample code exists to allow userid propagation. Thus, users can submit jobs without userid information on the job statement. The userid used is the work station name.

In the following example:

- The RDEFINE allows all users to submit jobs through R105.RD1**(JES2)** or POK1**(JES3)**.
- The PERMITs prohibit USER4 from submitting jobs through reader R105.RD1 or the readers at terminal POK1.
- The PERMITs for group(REMOTE1), prohibits this group of users from submitting jobs through reader R105.RD1 and the readers at terminal POK1.

JESINPUT Profile Definitions (JES2):

```
RDEFINE   JESINPUT   R105.RD1   UACC(READ)
PERMIT   R105RD1   CLASS(JESINPUT)   ID(USER4)   ACCESS(NONE)
PERMIT   R105RD1   CLASS(JESINPUT)   ID(REMOTE1)   ACCESS(NONE)
```

JESINPUT Profile Definitions (JES3):

```
RDEFINE   JESINPUT   POK1   UACC(READ)
PERMIT    POK1   CLASS(JESINPUT)   ID(USER4)   ACCESS(NONE)
PERMIT    POK1   CLASS(JESINPUT)   ID(REMOTE1)   ACCESS(NONE)
```

After USER4 had submitted a job through an RJE/RJP station, the following messages are logged in the SYSLOG:

```
(JES2)  16.54.09 JOB02706 $HASP165 USER4ID  ENDED AT C2JES2 - JCL ERROR

(JES3)  IAT6108   JOB  USER4ID  (JOB02706)  FAILED BY SECURITY CHECK
```

The following message are logged in the JESMSGLG:

```
ICH408I USER(USER4    ) GROUP(USER     ) NAME(ROBYN                )
  LOGON/JOB INITIATION - NOT AUTHORIZED TO TERMINAL/CONSOLE
```

6.7.7 JESINPUT Class for NJE Nodes

Access to the adjacent NJE nodes can be protected by coding the JESINPUT class. The adjacent NJE node name is specified as a profile name in the resource class. In Figure 31, the node names are C2JEST, C2JES2, and C5JES3.

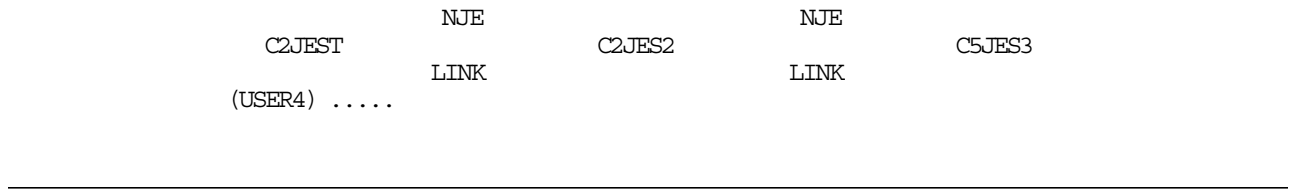

Figure 31. NJE Nodes and JESINPUT

The JES2 node C2JEST is connected to JES2 node C2JES2, which, in turn, is connected to C5JES3. A TSO user defined as USER4 is currently logged on to node C2JEST. USER4 submits a job with a /*ROUTE card to execute on node C2JES2:

```
//PAYAAID JOB (999,POK),¢USER4¢,USER=USER4,NOTIFY=USER4
/*ROUTE   XEQ C2JES2
```

In the following example:

- The RDEFINE allows all users to submit jobs from C2JEST.
- The first PERMIT prohibits jobs submitted by USER4 from entering the system at node C2JES2.
- The next PERMIT prohibits jobs submitted by group(USER) from entering the system at node C2JES2.

JESINPUT Profile Definitions (JES2) for Node C2JES2:

```
RDEFINE   JESINPUT   C2JEST   UACC(READ)
PERMIT    C2JEST   CLASS(JESINPUT)   ID(USER4)   ACCESS(NONE)
PERMIT    C2JEST   CLASS(JESINPUT)   ID(USER)   ACCESS(NONE)
```

Messages in SYSLOG:

```
      SYSLOG for C2JEST

$HASP100 PAYAAID  ON INTRDR      USER4                FROM TSU00460
USER4
$HASP520 PAYAAID  ON L6.JT1
$HASP524 L6.JT1   INACTIVE
$HASP250 PAYAAID  IS PURGED
$HASP540 PAYAAID  ON L6.SR1            10 RECORDS

     SYSLOG for C2JES2

$HASP100 PAYAAID  ON L9.JR1      USER4
SE ¢13.28.22 JOB00474 $HASP122 PAYAAID  (JOB00474 FROM C2JEST  )
RECEIVED AT C2JES2¢,LOGON,USER=(USER4)
SE ¢13.28.22 JOB00474 $HASP526 PAYAAID  TRANSMITTED FOR EXECUTION AT
C2JES2¢,LOGON,USER=(USER4)
SE ¢13.28.22 JOB00474 $HASP165 PAYAAID  (JOB00474 FROM C2JEST  ) ENDED
AT C2JES2 - JCL ERROR¢,LOGON,USER=(USER4)
ICH408I USER(USER4   ) GROUP(P0112   ) NAME(ROBYN               ) 668
  LOGON/JOB INITIATION - NOT AUTHORIZED TO TERMINAL/CONSOLE
SE ¢13.28.24 JOB00474 $HASP546 PAYAAID  SYSTEM OUTPUT RECEIVED AT
C2JEST¢,LOGON,USER=(USER4)
$HASP530 PAYAAID  ON L9.ST1            10 RECORDS
$HASP534 L9.ST1   INACTIVE
$HASP250 PAYAAID  IS PURGED
```

The following notification is sent to the user and displayed on the TSO screen:

```
$HASP122 PAYAAID (JOB00474 FROM C2JEST ) RECEIVED AT C2JES2
$HASP526 PAYAAID TRANSMITTED FOR EXECUTION AT C2JES2
$HASP165 PAYAAID (JOB00474 FROM C2JEST ) ENDED AT C2JES2 - JCL ERROR
$HASP546 PAYAAID SYSTEM OUTPUT RECEIVED AT C2JEST
```

Note: No notification other than a JCL error is sent to the user. This can be extremely misleading as there is nothing wrong with the JCL and it is actually RACF that is failing the authorization. There is also no job output information, as the job is purged after the authorization failure and before it starts.

For more information on the RACF NODES class, see Chapter 8, “NJE Security Control” on page 117.

6.7.8 SECLABELs with JESINPUT CLass

Jobs can be restricted from entering the system based on the SECLABEL of the submitted job and the SECLABEL access defined in the JESINPUT class. For a job to enter the system, the SECLABEL specified on the job card must dominate the SECLABEL defined in the POE profile. When no SECLABEL is specified on the job card, a SECLABEL is propagated from the users profile that is defined to RACF.

The SECLABELS defined for the purpose of the following examples are:

BRCON Confidential, the highest classification

NUNC Unclassified, the lowest level

- A TSO user with a userid(USER4) can submit jobs starting with any jobname.
- Grant access to the following SECLABELS for USER4 by using the PERMIT command:

```
PERMIT  BRCON  CLASS(SECLABEL)  ID(USER4)  ACCESS(READ)

PERMIT  NUNC  CLASS(SECLABEL)  ID(USER4)  ACCESS(READ)
```

- The RACF definition for the JESINPUT class and the SECLABEL are as follows:

```
RDEFINE  JESINPUT  INTRDR  UACC(READ)  SECLABEL(BRCON)
```

The INTRDR profile name of the JESINPUT class is used to illustrate the principle but it can be applied to any of the other POE names shown in Table 8 on page 80.

Example 1: The user submits a job with a SECLABEL of BRCON on the job card:

```
//PAYAAID JOB (999,POK),¢USER4¢,USER=USER4,SECLABEL=BRCON
```

The INTRDR POE is defined with the SECLABEL, BRCON.

This job completes with CC=0.

Example 2: USER4 submits a job with SECLABEL of NUNC on the job card:

```
//PAYAAID JOB (999,POK),¢USER4¢,USER=USER4,SECLABEL=NUNC
```

The SECLABEL of NUNC is lower than BRCON, the job fails, and the user is notified.

The message to the submittor's TSO screen is:

```
(JES2)   15.10.28 JOB02777 $HASP165 USER4ID  ENDED AT C2JES2 - JCL ERROR

(JES3)  IAT6108   JOB  USER4ID  (JOB02777)  FAILED BY SECURITY CHECK
```

The following notification is placed into the **JESMSGLG** data set in the output returned to the submittor.

```
ICH408I USER(USER4   ) GROUP(P0112   ) NAME(ROBYN               )
  LOGON/JOB INITIATION - NOT AUTHORIZED TO TERMINAL/CONSOLE
```

Example 3: USER4 submits a job with a SECLABEL of BRCON on the job card:

```
//PAYAAID JOB (999,POK),¢USER4¢,USER=USER4,SECLABEL=BRCON
```

The INTRDR POE is defined with a SECLABEL of NUNC:

```
RDEFINE  JESINPUT  INTRDR  UACC(READ)  SECLABEL(NUNC)
```

The SECLABEL of NUNC is lower than BRCON; therefore, the job completes with CC=0.

6.7.9 Considerations with the JESINPUT Class

The examples specify individual users, although group names can be used. The generic attribute of the SETROPTS command for the resource class must be active if generic names are used.

For a user to submit a job to another node, there may be RACF profiles in the NODES class to deny or allow submission.

For RACF to check whether a user is authorized to submit jobs through an RJE/RJP station or a local reader, the BATCHALLRACF option must be active, forcing users to specify user information on the job card.

When a SECLABEL is omitted from the job card, it is propagated from the RACF defined user profile.

When a job is transmitted through an NJE link and it cannot identify the SECLABEL, it is blanked out when it arrives at the execution node. The SYSOUT is transmitted back to the submitting node, and a SECLABEL of RACSLUNK is assigned to it. This happens when the submitting node does not have the SECLABEL class activated and the receiving node does.

6.8 JESJOBS Class

With RACF 1.9 you can use RACF to control which jobnames users can submit or cancel. In RACF, the class **JESJOBS** is used for these purposes.

If the installation has an exit that checks for job names, this exit may be removed and the new RACF control can be used instead.

6.8.1 JESJOBS Class for Job Submission

By activating the RACF resource class, JESJOBS, you can control the use of job names in the system. The IKJEFF10 exit supplied with the system is a dummy exit that allows any user to submit a job with any job name. This might not meet installation standards and this exit may be coded to at least check for the userid in the jobname. The alternative is to activate the RACF JESJOBS class and define a profile for every user. This could be cumbersome and time consuming as each user has to have a discrete profile. The next level would be to check by group, which means that any user within a group can submit a job within that group, and this approach might be more acceptable.

Note: Control of job names with the RACF resource class, JESJOBS, applies to batch jobs only. Started tasks, TSO logons, and operator mount command jobs are not checked.

Figure 32 shows user authorization for submitting a job with a job name of PAYROLL. Harry, and not Sally, is authorized through a JESJOBS profile to submit a job with the name PAYROLL. Bob, however, is allowed to submit a job only with the name PAYROLL, if the input source is a TSO userid and not a remote terminal.

The format of the JESJOBS class profile name for job submission is:

```
SUBMIT.nodename.jobname.userid
```

Note: If the JESJOBS class is activated, define the following generic profile in that class, with universal access of READ. This ensures that, if there are no other profiles in that class, any job names can be submitted:

```
RDEFINE  JESJOBS  **  UACC(READ)
```

Then you can define more specific profiles in the JESJOBS class to protect specific job names from being submitted.

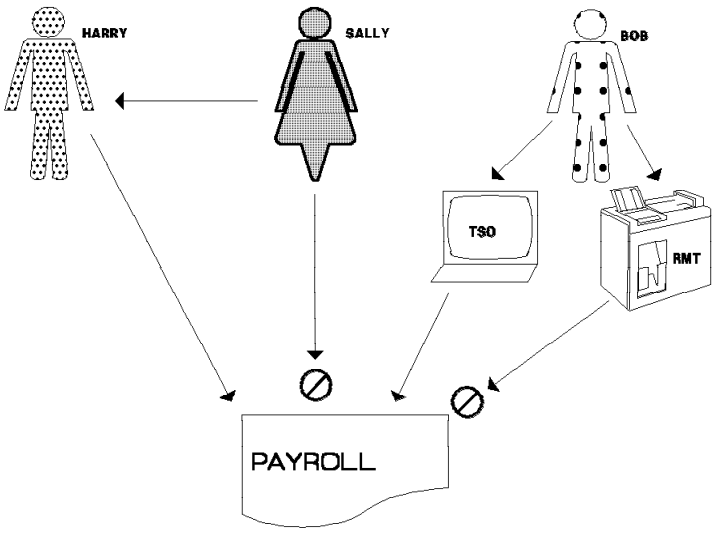

Figure 32. Job Name Control

6.8.2 Controlling Job Submission by Job Name

One way of implementing the JESJOBS class is to give only a select number of people control over the submission of restricted job names, such as operations or a production control person. It is better to specify a global authorization for users to submit all jobs and then permit only a select number of users to submit restricted jobs. Profiles have to be created to restrict jobs based on their job name.

In this example, the submission of job names starting with PAY is protected. Only USER4 is permitted to submit those jobs:

- First, to allow all job names to be submitted, specify:

  ```
  RDEFINE  JESJOBS  SUBMIT.*.**  UACC(READ)
  ```

- To not allow any users to submit any jobs starting with PAY, specify:

  ```
  RDEFINE  JESJOBS  SUBMIT.*.PAY*.*  UACC(NONE)
  ```

If a job for USER9 is submitted through the internal reader, the submittor receives the following notification on the TSO screen with the following job statement:

```
        //PAYAAID JOB (999,POK),¢USER9¢,USER=USER9
(JES2)  16.54.09 JOB02704 $HASP165 PAYAAID  ENDED AT C2JES2 - JCL ERROR

(JES3)  IAT6108  JOB  PAYAAID  (JOB02704)  FAILED BY SECURITY CHECK
```

The following message is placed into the job's **JESMSGLG** data set in JES2 and JES3 systems:

```
ICH408I USER(USER9   ) GROUP(USER    ) NAME(                    )
  SUBMITTER(USER9)
  LOGON/JOB INITIATION - NOT AUTHORIZED TO SUBMIT JOB PAYAAID
```

To allow only USER4 to submit jobs starting with the name PAY, specify:

```
PERMIT   SUBMIT.*.PAY*.*  CLASS(JESJOBS)   ID(USER4)   ACCESS(READ)
```

6.8.3 Job Submission Based on Input Source

The JESJOBS class in conjunction with the JESINPUT class can be used to further restrict users from submitting jobs. This forces users to use only pre-defined ports of entry as specified by the RACF administrator. For example, you allow users from another node to run only certain types of jobs on the current machine. The reasons for this may vary from security to performance.

Thus, authority to submit a specific job name can be based on “who” submitted the job and “where” it is submitted from.

Specific users or groups can be permitted for the submission of specific job names. In addition, control can be placed on the source of the job input through the use of the **JESINPUT** class.

In Figure 33:

- Group **PAYGROUP** is allowed to submit jobs with the name of **PAYROLL** when submitted through TSO.
- Group **PAYGROUP** cannot submit jobs through local or remote input devices.
- User **Bert** is not allowed to submit jobs with a job name of **PAYROLL**.
- Control decisions are based on who the user is and optionally on the location of the input device.

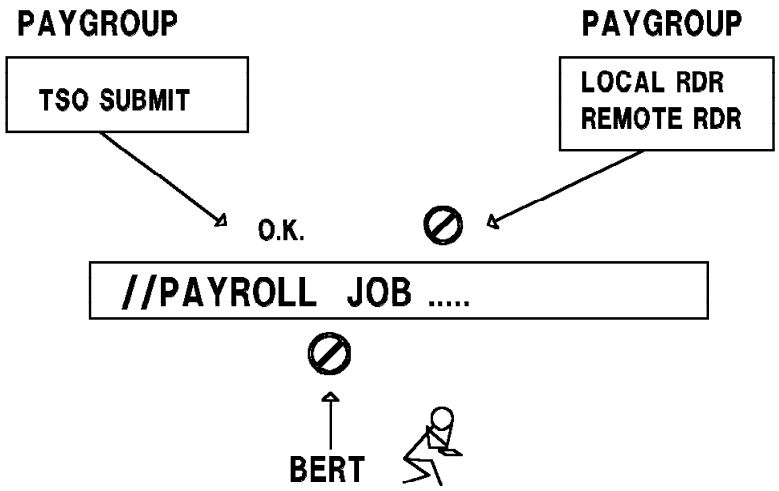

Figure 33. Controlling Job Submission by Input Source

The RACF definitions for the example shown in Figure 33 are:

```
RDEFINE  JESJOBS  *.**  UACC(READ)
RDEFINE  JESJOBS  SUBMIT.*.PAYROLL.*  UACC(NONE)
PERMIT   SUBMIT.*.PAYROLL.*  CLASS(JESJOBS)  ID(PAYGROUP)   ACC(READ)   WHEN(JESINPUT(INTRDR))
```

Through the use of the PERMIT, only users in the group PAYGROUP may submit payroll jobs from an internal reader because of the JESINPUT input device being INTRDR. That is, a job can be submitted

only by the TSO SUBMIT command or by another job with a DD statement specifying SYSOUT=(,INTRDR).

6.8.4 Job Submission in a Network

As previously stated, you may desire to restrict certain jobs based on their job name from entering the system through the NJE network.

In Figure 34, we want to restrict jobs submitted by USER4 on C2JES2 to execute on C2JEST. The only jobs allowed at C2JEST must begin with the job name of PAY.

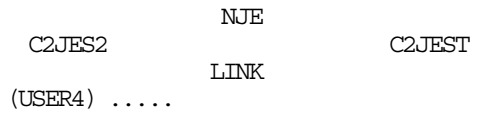

Figure 34. NJE Nodes and JESJOBS with JESINPUT

Definitions at C2JEST

- Define a profile that allows no users to submit jobs with the name PAY*:

  ```
  RDEFINE  JESJOBS  SUBMIT.*.PAY*.*  UACC(NONE)
  ```

- Define a profile that allows no input from C2JES2:

  ```
  RDEFINE  JESINPUT  C2JES2  UACC(NONE)
  ```

- Permit USER4 submitted jobs beginning with PAY* to enter the system:

  ```
  PERMIT  SUBMIT.*.PAY*.*  CLASS(JESJOBS)  ID(USER4)  ACCESS(READ)  WHEN(JESINPUT(C2JES2)
  ```

- Permit USER4 submitted jobs to enter the system if submitted from C2JES2:

  ```
  PERMIT  C2JES2  CLASS(JESINPUT)  ID(USER4)  ACCESS(READ)
  ```

In the example, USER4 can submit jobs beginning with a job name of PAY from C2JES2 to C2JEST. The RACF resource class JESJOBS is defined to prohibit any job with a prefix of PAY from starting on the system unless it enters from node C2JES2. The RACF resource class JESINPUT is defined to allow USER4 to submit any jobs from C2JES2. USER4 on C2JES2 submits a job with the following job card:

```
//PAYAAID JOB (999,POK),¢USER4¢,USER=USER4,PASSWORD=F6ASD,GROUP=P0112
```

A password is needed in the job card because node C2JES2 is not trusted.

The job is submitted on C2JES2 and routed to C2JEST where it is executed with a CC=0.

The same job, if submitted through the internal reader at C2JEST, fails with the following messages:

```
14.58.13 JOB00065 $HASP165 PAYAAID  ENDED AT C2JEST - JCL ERROR

ICH408I USER(USER4   ) GROUP(P0112   ) NAME(ROBYN               )
  LOGON/JOB INITIATION - USER IS NOT AUTHORIZED TO JOB PAYAAID
```

6.8.5 JESJOBS Class for Job Canceling

The CANCEL profile name of the RACF JESJOBS class applies only to the TSO CANCEL command. The TSO CANCEL command cancels jobs by job name. The JESJOBS class has the following profile format for job canceling:

```
CANCEL.nodename.userid.jobname
```

6.8.5.1 IKJEFF53 Exit

The IBM supplied default IKJEFF53 exit checks for job name with userid and one or more additional characters. A new IKJEFF53 exit is supplied in SYS1.SAMPLIB. For the TSO CANCEL command, it checks to see whether the JESJOBS class is active. If it is, no jobname checking is to to be done.

Note: If you activate the JESJOBS class, ensure that the TSO/E exit IKJEFF53 is replaced by the one supplied in SYS1.SAMPLIB. If the sample exit does not conform to your standards, use it as a base to build your alterations from. If the JESJOBS class is inactive, this exit functions as per the normal IBM supplied exit for the TSO CANCEL command.

However, the job name checking that is currently being done by the IBM supplied exit cannot easily be duplicated by using RACF. A profile has to be defined for each user. This is impractical for installations with large numbers of TSO users. For such installations it is more advisable to keep the old IKJEFF53 logic.

6.8.5.2 Profiles for Canceling a Job by Job Name

This example uses the previously described example on job name restrictions with the job name PAY*. It is used to illustrate who is authorized to cancel jobs beginning with the name PAY*.

Note: ALTER access is required to CANCEL a job.

The RDEFINE denies any user to cancel jobs starting with PAY:

```
RDEFINE  JESJOBS  CANCEL.*.*.PAY*  UACC(NONE)
```

Example with OPER4 attempts to cancel a job beginning with PAY, RACF does not allow the job to be cancelled and issues the following message:

```
ICH408I USER(OPER4   ) GROUP(OPER    ) NAME(OPER4               )
  CANCEL.C2JES2.USER4.PAYAAID CL(JESJOBS )
  INSUFFICIENT ACCESS AUTHORITY
  FROM CANCEL.*.*.PAY* (G)
  ACCESS INTENT(ALTER  )  ACCESS ALLOWED(NONE   )
```

To allow USER4 only to cancel jobs starting with PAY, you can specify that USER4 can cancel jobs starting with any job name by using the TSO CANCEL command, (TSO CANCEL PAYAAID(J1890)), or allow only USER4 to cancel jobs starting with PAY:

```
PERMIT  CANCEL.*.*.PAY*  CLASS(JESJOBS)  ID(USER4)  ACCESS(ALTER)
```

Note: This is contrary to the way SDSF works. If SDSF is used, SDSF translates the CANCEL command (line command) to a cancel job number ($CJXXXX), which bypasses this RACF protection. SDSF has been considerably enhanced to protect JES resources and when activated blocks the above bypass. For more information on SDSF and RACF protection, refer to Chapter 12, “SDSF Release 3” on page 205.

6.8.6 JESJOBS Considerations

The SETROPTS GENERIC(JESJOBS) must be active for generic profile names. If SETROPTS is not set, all the profiles that are created do not have the generic attributes assigned to them.

Define at least one generic profile with a universal access of READ before the resource class is activated. This ensures that anybody can at least submit a job when the JESJOBS class is activated for jobnames. After that, more restrictive profiles can be created.

IKJEFF10 has to be coded if it is to be used for jobname checking not done by RACF. The IKJEFF10 module supplied with the system does not check for job validation. If you wanted to check for correct job name validation, you had to code this exit in previous releases. Now you can let RACF do the job name validation by activating the JESJOBS class and define all the relevant profiles. The JESJOBS class for job submission is discussed in 6.8, “JESJOBS Class” on page 86.

Many installations require their users to submit jobs starting with a USERID and a suffix. The only way that this can be done is through an exit. The addition of the JESJOBS class does not provide any way to achieve this requirement with new profile definitions.

6.9 SURROGAT Class

A surrogate user is a RACF-defined user who has been authorized to submit jobs on behalf of another user (the owning user; the user specified on the job card) without having to specify the other user's password. Jobs submitted by the surrogate user execute with the identity of the owning user. After job verification is complete, a job submitted by a surrogate user runs with the owning user's authorization to resources. The output of the job is owned by the owning user.

6.9.1 Defining Surrogate Users

The SURROGAT class is used for surrogation. The format of the SURROGAT class profile name is:

```
owner-userid.SUBMIT
```

The following definitions allow USER3 to submit jobs that are owned by USER4:

First define a profile allowing USER4 as a surrogate:

```
RDEFINE  SURROGAT  USER4.SUBMIT  UACC(NONE)  OWNER(USER4)
```

Then, define USER3 to be authorized to submit jobs for USER4:

```
PERMIT  USER4.SUBMIT  CLASS(SURROGAT)  ID(USER3)  ACCESS(READ)
```

USER3 is defined as a surrogate user able to submit jobs for USER4 without having to specify a password on the job card. The job card is specified as follows:

```
//JOBXYZ  JOB (999,POK),¢USER4¢,USER=USER4
```

The job completes with RACF permitting the job to execute. If the SURROGAT class is not active, or USER3 is not permitted to the profile, the job fails with messages:

```
14.49.55 JOB03397 $HASP165 PAYAAID  ENDED AT C2JES2 - JCL ERROR

ICH408I USER(USER4   ) GROUP(P0112   ) NAME(ROBYN               )
  SUBMITTER(USER3   )
  LOGON/JOB INITIATION - SUBMITTER IS NOT AUTHORIZED BY USER
```

6.9.2 Surrogate with SECLABEL

When a surrogate user submits a job on behalf of another user, the owner's information is propagated with the job (for more information see 6.9.4, "Surrogate Propagation"). The default SECLABEL defined for the owner of the userid is associated with the submitted job. If the owner of the job does not have a SECLABEL defined in the user profile, and none is defined in the job card, the job executes without a SECLABEL. If a default SECLABEL has been defined for the ownerid, then both users have to be permitted to the same SECLABEL before the job completes successfully. If the surrogate user is not permitted to the SECLABEL to which the owning userid is defined, then RACF fails the job.

6.9.3 Surrogate Definitions with SECLABELs

A TSO user is defined as USER99 with authorization to submit jobs. USER99 can submit jobs starting with any jobname. A TSO user defined as USER4 and defined as a surrogate user is able to submit jobs for USER99 without having to specify a password on the job card. The default SECLABEL of USER99 is ICR. Both users are permitted to use a SECLABEL of ICR.

USER4 is allowed to submit jobs for USER99:

```
RDEFINE  SURROGAT  USER99.SUBMIT  UACC(NONE)  OWNER(USER99)
PERMIT  USER99.SUBMIT  CLASS(SURROGAT)  ID(USER4)  ACCESS(READ)
```

Both USER4 and USER99 are permitted to SECLABEL, ICR:

```
PERMIT  ICR  CLASS(SECLABEL)  ID(USER4)  ACCESS(READ)
PERMIT  ICR  CLASS(SECLABEL)  ID(USER99)  ACCESS(READ)
```

The job card in the following example has the 'USER=USER99' statement specified, stating that the job belongs to USER99 and must run with USER99's authority.

```
//USER4D JOB (999,POK),¢USER4¢,USER=USER99
```

The SECLABEL allocated to the job is the default SECLABEL of ICR, which has been assigned to USER99. Due to USER4 having been permitted to SECLABEL ICR, the job completes normally. If the submittor is not permitted to the same SECLABEL as the one specified in the user profile of the owner, the job fails with messages:

```
14.08.18 JOB05642 $HASP165 USER4D  ENDED AT C2JES2 - JCL ERROR

ICH408I USER(USER99  ) GROUP(USER    ) NAME(TESTUSER            )
  SUBMITTER(USER4   )
  LOGON/JOB INITIATION - SUBMITTER IS NOT AUTHORIZED TO SECURITY LABEL
```

6.9.4 Surrogate Propagation

When a surrogate user submits a job on behalf of another user, the owner's information is propagated with the job, as though the owner of the job had submitted the job. By using the SETROPTS LOGOPTIONS ALWAYS for the class JESINPUT, it is possible to see what information is propagated when the job is submitted through the internal reader. The RACF Report Writer formats the SMF data that is created using the LOGOPTIONS setting.

6.9.5 Surrogate Propagation Definitions

The following are examples of definitions used with propagation:

- SETROPTS option for determining the information that is propagated:

```
SETROPTS LOGOPTIONS(ALWAYS(JESINPUT))
```

- RACF definitions for the classes SURROGAT and SECLABEL:

```
RDEFINE   SURROGAT   USER99.SUBMIT   UACC(NONE)   OWNER(USER99)
ALTERUSER   USER99   SECLABEL(ICR)
PERMIT   USER99.SUBMIT   CLASS(SURROGAT)   ID(USER4)   ACCESS(READ)
PERMIT   ICR   CLASS(SECLABEL)   ID(USER4)   ACCESS(READ)
PERMIT   ICR   CLASS(SECLABEL)   ID(USER99)   ACCESS(READ)
```

USER4 submits the job:

```
//USER4D JOB (999,POK),¢USER4¢,USER=USER99
```

By analyzing the SMF records created by RACF using the RACF Report Writer, it is possible to see what information is propagated when a surrogate user submits a job on behalf of another user. The RACF Report Writer output is as specified in the following examples.

Example 1. No default SECLABEL in user profile:

```
90.066 12:37:29 SMF2   USER99    USER              0  2  0  JOBID=(JES2 90.065 09:27:25),USERDATA=(),OWNER=P0112AW
                       TESTUSER                             AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                            SESSION=INTERNAL READER BATCH JOB,TOKEN USER ATTRIBUTES=(
                                                            SURROGATE USERID),JESINPUT=INTRDR,EXENODE=C2JES2,SUBMITTING USER=
          ...Note Absence of SECLABEL...                    USER4,SUBMITTING NODE=C2JES2,SUBMITTING GROUP=P0112
                                                            JESINPUT=INTRDR,LEVEL=00,INTENT=READ,ALLOWED=READ
```

Example 2. Default SECLABEL defined, propagated from user profile:

```
90.066 14:10:28 SMF2   USER99    USER              0  2  0  JOBID=(JES2 90.065 09:27:25),USERDATA=(),OWNER=P0112AW
                       TESTUSER                             AUTH=(NORMAL),REASON=(LOGOPTIONS)
                                                            USER SECLABEL=ICR,SESSION=INTERNAL READER BATCH JOB,
                                                            TOKEN USER ATTRIBUTES=(SURROGATE USERID),JESINPUT=INTRDR,EXENODE=
          ...Default SECLABEL included...                   C2JES2,SUBMITTING USER=USER4,SUBMITTING NODE=C2JES2,
                                                            SUBMITTING GROUP=P0112
                                                            JESINPUT=INTRDR,LEVEL=00,INTENT=READ,ALLOWED=READ
```

6.9.6 Considerations with the SURROGAT Class

Considerations on Surrogate Job Submission:

- For the submittor to be able to access the output of a surrogated job, he must be permitted to the owning userid's output in the JESSPOOL class.
- When submitting jobs to another node through NJE, the submittor's information is passed to the execution node in the job header along with the job's owning id.
- The RACF NODES class must be checked to ensure that the submitting node and its userid is trusted. That means that the universal access of the profile to define the Node must at least be UPDATE. For more information of the NODES class see 8.2, "RACF NODES Class" on page 123.
- Surrogate processing of NJE submitted jobs is exactly the same as local processing.

Considerations on Surrogate with SECLABELs:

- If the ownerid has no default SECLABEL defined, and none is defined on the job card, the job runs without a SECLABEL.

- The default SECLABEL defined in the owning user's profile is propagated and used in the job if no SECLABEL is specified on the job card.
- If a 'SECLABEL=' statement is specified on the job card, that SECLABEL is used during execution of that job.
- Both users have to be permitted to the SECLABEL being used during the execution.

For information of the class SURROGAT when MLS is turned on see 3.6.3.2, "MLS Effects on Job Submissions" on page 29.

Considerations with Surrogate Propagation:

- In all cases, the default SECLABEL from the user profile is propagated. If a SECLABEL is specified on the job card, that SECLABEL is used instead. The submittor and owner however, both have to be permitted to the SECLABEL.
- All user information from the owner is propagated to the job.

Chapter 7. SYSIN / SYSOUT - JES Spool

The JES spool data set is a standard data set that contains SYSIN/SYSOUT data sets for jobs. RACF 1.9 can be used to provide protection to the SYSIN/SYSOUT data sets that reside on the spool, including spool files that JES appends to job output, such as JESNEWS. A new RACF resource class, **JESSPOOL**, is used for this.

7.1 JESSPOOL Profile Name

The profiles in the JESSPOOL class allow a user to grant other users access authority to some or all the spool data sets created by a specific job. The format of the JESSPOOL profile name is:

```
nodename.userid.jobname.jobid.Ddsnumber.dsname
```

Generic characters can be specified for any qualifiers in the profile name. The profiles can be up to 39 characters long. If longer profile names are needed, the RACF administrator has to restructure the RACF database using the IRRDSC00 utility. For more information on restructuring the RACF database, see 4.11.3, “Migrating from RACF 1.8.1” on page 51 or *Systems Programming Library: RACF*.

When the JESSPOOL class is active, RACF ensures that only authorized users have access to spool data sets. Authorization to these spool data sets is provided through profiles in the JESSPOOL class. If no profile exists for a data set, only the user that created the data set can access, modify, or delete it.

The RACF system administrator (SPECIAL) can activate the new JESSPOOL class with the command:

```
SETROPTS  CLASSACT(JESSPOOL)
```

To create JESSPOOL profiles, the user needs the CLAUTH authority for the JESSPOOL class. If an installation activates the GENERICOWNER option, it can then restrict each user to creating JESSPOOL profiles only for their own spool data sets. Harry would like Sally to have access to his spool data sets. The following example shows how to achieve this (Figure 35):

1. Issue the SETROPTS GENERICOWNER command:

   ```
   SETROPTS  GENERICOWNER
   ```

2. For each user who needs to create JESSPOOL profiles for his or her own spool data sets, create a JESSPOOL profile with the user's userid specified. Make the user the owner of the profile. For example, for user HARRY:

   ```
   RDEFINE  JESSPOOL  C2JES2.HARRY.**  OWNER(HARRY)  UACC(NONE)
   ```

3. Because HARRY is the owner of the previous profile, he can now use the following command to give SALLY access to his spool data sets:

   ```
   PERMIT  C2JES2.HARRY.**  CLASS(JESSPOOL)  ID(SALLY)  ACCESS(acc_auth)
   ```

   The **acc_auth** levels that can be used in the JESSPOOL profile are:

   NONE Allows no access.

   READ Allows user to browse the spool data set, but not change its attributes.

UPDATE Allows another user in the same address space to update a spool data set.

CONTROL The same as UPDATE.

ALTER Allows any operation on the spool data set, including deleting and printing. With the GENERICOWNER option inactive, this level of authority in a discrete profile allows the user to change the profile itself.

4. If HARRY needs to create more specific profiles, he requires CLAUTH authority. The security administrator can give HARRY this authorization as follows:

```
ALTUSER HARRY CLAUTH(JESSPOOL)
```

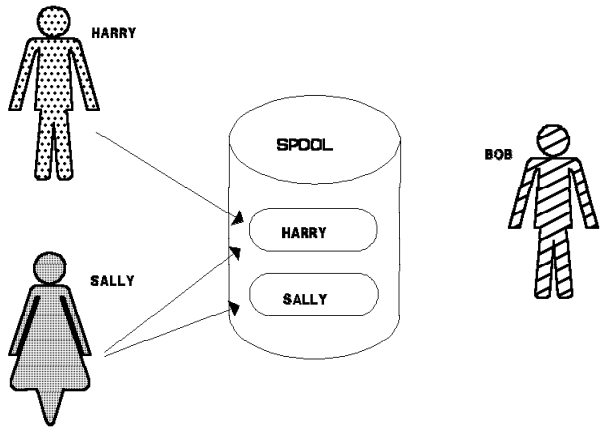

Figure 35. JES Spool Data Set Security

7.1.1 SECLABEL Considerations

For access to a spool data set when the SECLABEL class is active, the SECLABEL with which the user is currently logged on must dominate or be equal to the SECLABEL of the spool data set. For more information, see 7.4, “Process SYSOUT Requests” on page 103.

7.1.2 Auditing Considerations

For auditing purposes, RACF can optionally log the creation and each access for SYSIN and SYSOUT data sets. The selection for printing of SYSOUT data sets and the deletion of spool data sets can also be audited. The security administrator activates logging by issuing the command:

```
SETROPTS LOGOPTIONS(ALWAYS | NEVER | SUCCESSES | FAILURES(JESSPOOL))
```

Note: Do not use LOGOPTIONS(ALWAYS). It can produce many SMF records and seriously impact the performance of the system.

7.2 DSNAME for SYSIN/SYSOUT Data Sets

During job execution, JES creates data sets for a job's input (SYSIN) and output (SYSOUT). The new format of the spool data set name is:

```
userid.jobname.jobid.Ddsnumber.dsname
```

Where:

userid The userid associated with the job. This is the userid RACF uses for validation purposes when the job runs.

jobname The name that appears in the name field of the JOB statement.

jobid The job number JES assigned to the job. The jobid appears in notification messages and the JES job log of every job.

Ddsnumber The unique data-set number JES assigned to the spool data set. A "D" is always the first character of this qualifier.

dsname The name of the data set specified in the DSN= parameter of the DD statement. This name follows the naming conventions for a temporary data set, and cannot be JESYSMSG, JESJCLIN, JESJCL, or JESMSGLG.

The dsname parameter is added to the DD statement for SYSOUT and SYSIN data sets. If not specified, JES substitutes a **?** for the data set name.

The following examples show the use of the new JCL keyword:

SYSOUT Data Sets

```
1) //REPT1     DD   DSN=&PAYOUT,SYSOUT=A

2) //SYSPRINT  DD   SYSOUT=*,DSN=&IEBCOPY3

3) //SYSUDUMP  DD   SYSOUT=D
```

SYSIN Data Sets

```
4) //DASDIN1   DD   *,DSN=&PAYROLL

5) //SYSIN     DD   DATA,DSN=&CONTROLA
```

The generated spool data set names that correspond to the above examples are:

1) P0112WK.PAYROLL1.JOB00190.D0000102.PAYOUT

2) USER2.IEBCOPYA.JOB07753.D0000107.IEBCOPY3

3) USER2.APUPDATE.JOB04169.D0000108.?

4) P0112WK.PAYROLL1.JOB00190.D0000105.PAYROLL

5) DEVL1.DEVL1ACP.JOB03295.D0000110.CONTROLA

Note: The TSO ALLOCATE command does not support the DSN specification for a SYSIN/SYSOUT data set.

7.3 Special JES SYSOUT Data Sets

The JES spool contains other data sets besides the SYSIN and SYSOUT that can be protected by RACF profiles. These data sets are:

- JESNEWS
- Trace data sets (JES2 only)
- SYSLOG

7.3.1 JESNEWS Data Set

RACF provides two types of control for the JESNEWS data set: access control and update control.

Access control to receive the data set with the output is controlled with the SECLABEL class. In order for the JESNEWS data set to be printed with a user's output, a SAF call is made to verify that the user can access the JESNEWS data set. If this authorization check fails, JESNEWS is not printed.

Note: To make the JESNEWS data set accessible to all users, JESNEWS should be created by a job with a SECLABEL of SYSLOW.

Update control when creating the JESNEWS data set can be controlled with the OPERCMDS class

7.3.1.1 JES2 JESNEWS Data Set

The JESNEWS data set is an informational data set that is printed after the header page of a job's output. This data set is normally of interest to all job submitters, so it is important that all users can read this data set.

The profile name for JESNEWS in the JESSPOOL class is:

nodename.userid.$JESNEWS.STC00000.D0000000.JESNEWS

Where:

nodename The name of the node that created the JESNEWS data set.

userid The userid associated with the JES2 system.

$JESNEWS The job name given to the JESNEWS data set.

STC00000 The task number of the task that created the JESNEWS data set.

D0000000 The level of this copy of JESNEWS.

7.3.1.2 Access Control with JES2

When SECLABEL checking is inactive, all users receive the JESNEWS information in their printout as long as they have READ access to the JESNEWS profile in the JESSPOOL class. A sample entry for the profile is:

```
RDEFINE JESSPOOL C2JES2.*.$JESNEWS.*.*.JESNEWS UACC(NONE)
PERMIT C2JES2.*.$JESNEWS.*.*.JESNEWS CLASS(JESSPOOL) ID(OPER) ACCESS(READ)
PERMIT C2JES2.*.$JESNEWS.*.*.JESNEWS CLASS(JESSPOOL) ID(USER1 USER2) ACCESS(READ)
```

All operators in groups OPER, USER1, and USER2 are allowed to receive JESNEWS information on their printout. In an environment without SECLABEL checking, this is the only way to control access to the JESNEWS information.

With SECLABEL checking active, users authorized to receive the JESNEWS data set in their output are required to run their jobs with a SECLABEL that dominates or is equal to the SECLABEL of the JESNEWS data set. To keep the current JESNEWS function as it is today, specify ″SYSLOW″ in the SECLABEL field of the JESNEWS profile in the JESSPOOL class:

```
RALTER JESSPOOL C2JES2.*.$JESNEWS.*.*.JESNEWS UACC(READ) SECLABEL(SYSLOW)
```

With the above profile active in the JESSPOOL class, all users receive the JESNEWS information, except those who created job output with a blank SECLABEL field. For those users to receive the JESNEWS information, the JESNEWS data set has to be created by a user who logged on either with a SECLABEL of SYSLOW or with no SECLABEL at all. Also, the JESNEWS profile in the JESSPOOL class must have no SECLABEL defined.

7.3.1.3 Update Control with JES2

To prevent unauthorized updating of the JESNEWS data set, a profile is needed in the RACF OPERCMDS class. Any userid authorized to update the JESNEWS data set requires ALTER access to this profile. Define the JESNEWS update profile in the OPERCMDS class as follows:

```
RDEFINE OPERCMDS JES2.UPDATE.JESNEWS UACC(NONE)
PERMIT JES2.UPDATE.JESNEWS CLASS(OPERCMDS) ACCESS(ALTER) ID(userid | groupid)
```

Note: It is not sufficient to specify a SECLABEL of SYSLOW on the JESNEWS profile entry in the JESSPOOL class. The user who intends to update the JESNEWS data set **must** also be logged on with a SECLABEL of SYSLOW so that all users can receive the JESNEWS information. Remember that the RTOKEN of the JESNEWS data set is the UTOKEN of the user who updates the JESNEWS data set.

For more information on how to create this data set or delete it, see *MVS/ESA SPL: JES2 Initialization and Tuning*.

7.3.2 JES3 JESNEWS Data Set

The JESNEWS data set is an informational data set that is printed on the trailer page of a job′s output. This data set is normally of interest to all job submitters, so it is important that all users can read this data set. A UACC of READ is required so JESNEWS can print with all jobs.

The three profile names for JESNEWS in the JESSPOOL class are:

```
nodename.userid.JES3.JOB00000.D0000000.JNEWSLCL
nodename.userid.JES3.JOB00000.D0000000.JNEWSRJP
nodename.userid.JES3.JOB00000.D0000000.JNEWSTSO
```

Where:

nodename The name of the node that created the JESNEWS data set.

userid The userid associated with the JES3 system from the JES3 token.

JES3 The job name given to the JESNEWS data set from the SVTJS3NM field in the SSVT.

JOB00000 The job number of the task that created the JESNEWS data set.

D0000000 The level of this copy of JESNEWS.

JNEWSLCL The data set name of the JESNEWS printed for all local users.

JNEWSRJP The data set name of the JESNEWS printed for all remote terminal users.

JNEWSTSO The data set name of the JESNEWS printed for all TSO users.

The following profiles permit all users to have the local JESNEWS data set print with their job output:

```
RDEFINE JESSPOOL C5JES3.**.JNEWSLCL UACC(NONE)
PERMIT C5JES3.**.JNEWSLCL CLASS(JESSPOOL) ID(*) ACCESS(READ)
```

7.3.2.1 Update Control with JES3

There are two methods for creating or updating the JESNEWS data set:

- A batch job using the //*PROCESS statement
- Operator call of JESNEWS DSP

A profile is needed in the RACF OPERCMDS class to prevent unauthorized updating of the JESNEWS data set. Any userid authorized to update the JESNEWS data set requires ALTER access to this profile. An OPERCMDS RACROUTE is issued for the //*PROCESS statement as well as the operator call command. The IATXSEC macro call is used to control update of the JESNEWS data sets.

A profile for the JESNEWS data set is required and the operator or job needs to be authorized to the profile for updates. Define the profiles in the OPERCMDS class as follows:

```
RDEFINE OPERCMDS JES3.PROCESS.JESNEWS UACC(NONE)
PERMIT JES3.PROCESS.JESNEWS CLASS(OPERCMDS) ACCESS(ALTER) ID(userid | groupid)

RDEFINE OPERCMDS JES3.CALL.JESNEWS UACC(NONE)
PERMIT JES3.CALL.JESNEWS CLASS(OPERCMDS) ACCESS(ALTER) ID(userid | groupid)
PERMIT JES3.START.JESNEWS CLASS(OPERCMDS) ACCESS(ALTER) ID(userid | groupid)
PERMIT JES3.RESTART.JESNEWS CLASS(OPERCMDS) ACCESS(ALTER) ID(userid | groupid)
```

The JESNEWS data set can be created, added to, deleted, or replaced by the operator. If profiles exist in the OPERCMDS class for the JESNEWS data set access, JES3 issues an IATXSEC security call for authorization. Table 9 on page 101 indicates when this security check is made.

Table 9. JESNEWS Security Calls and Access Required			
Type Request	**Add JESNEWS**	**Delete JESNEWS**	**Replace JESNEWS**
*X	No check done	Alter access	No check done
*S	Update access	No check done	Update access
*R	Update access	No check done	Update access
*C	No check done	No check done	No check done
//*PROCESS	Update access	Alter access	Update access

The JESNEWS data set can be created, deleted, or replaced by submitting a job and using the //*PROCESS statement. A security check is made for each type of request.

7.3.2.2 SECLABEL Control with JES3

SECLABELs can be used to control access to the JESNEWS data sets. The operator or job has to have a SECLABEL of SYSLOW for updates.

With SECLABEL checking active, users authorized to receive the JESNEWS data set in their output are required to run their jobs with a SECLABEL that dominates or is equal to the SECLABEL of the JESNEWS data set. To keep the current JESNEWS function as it is today, specify ″SYSLOW″ in the SECLABEL field of the JESNEWS profile in the JESSPOOL class:

```
RALTER JESSPOOL C5JES3.**.JNEWSLCL UACC(READ) SECLABEL(SYSLOW)
```

With the above profile active in the JESSPOOL class, all users receive the JESNEWS information, except those who created job output with a blank SECLABEL field. For those users to receive the JESNEWS information, the JESNEWS data set has to be created by a user who logged on either with a SECLABEL of SYSLOW or with no SECLABEL at all. Also, the JESNEWS profile in the JESSPOOL class must have no SECLABEL defined.

7.3.3 TRACE Data Sets

TRACE data sets can contain information that can compromise an installation′s security, for example userids and passwords. These data sets can be protected in the JESSPOOL class by defining a profile for each data set and permitting access authority only to those userids that need access to it.

The profile for TRACE data sets is specified as:

```
nodeid.jesname.$TRCLOG.taskid.Ddsnumber.JESTRACE
```

Where:

nodeid Is the name of the node that created the TRACE data set.

jesname Is the name of the JES subsystem.

taskid Is the task number of the task that created the TRACE data set.

Ddsnumber Is the unique number JES assigned to the TRACE data set.

To limit access, define the following:

```
RDEFINE JESSPOOL C2JES2.JES2.*.*.*.JESTRACE UACC(NONE)
PERMIT C2JES2.JES2.*.*.*.JESTRACE CLASS(JESSPOOL) ID(SYSPRG1) ACCESS(READ)
```

In this case, only system programmer SYSPRG1 can read the trace data set.

If security label checking is active, the trace data sets should have the highest (SYSHIGH) security label in the installation. This label also ensures that the trace data set can be printed only on a device that is defined in the WRITER class with a SECLABEL of SYSHIGH.

For an operator or a system programmer to enable the trace facility in JES2, the following access rules have to be defined:

```
RDEFINE OPERCMDS JES2.DISPLAY.TRACE UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.TRACE UACC(NONE)
RDEFINE OPERCMDS JES2.START.TRACE UACC(NONE)

PERMIT JES2.DISPLAY.TRACE CL(OPERCMDS) ACCESS(READ) ID(SYSPRG1)
PERMIT JES2.STOP.TRACE CL(OPERCMDS) ACCESS(CONTROL) ID(SYSPRG1)
PERMIT JES2.START.TRACE CL(OPERCMDS) ACCESS(CONTROL) ID(SYSPRG1)
```

In this case, only system programmer SYSPRG1 can display the trace status, or start or stop a trace.

7.3.4 SYSLOG Data Set

An installation may be require that the SYSLOG data set be protected, as it contains a record of the systems daily activities. A profile name for the SYSLOG data set to define in the JESSPOOL class is:

```
NODEA.+MASTER+.SYSLOG.**
```

Where:

NODEA Nodename of local system, JES2 or JES3.

+MASTER+ Is the ownerid of the Master Scheduler address space.

SYSLOG The job name is SYSLOG.

To limit access authority, the security administrator defines a profile in the JESSPOOL class for the SYSLOG data set with the appropriate universal access, and then grants access to the userids or groupids that need a different access; for example:

```
RDEFINE JESSPOOL C2JES2.+MASTER+.SYSLOG.** UACC(NONE)

PERMIT C2JES2.+MASTER+.SYSLOG.** CLASS(JESSPOOL) ID(OPER) ACCESS(READ)
PERMIT C2JES2.+MASTER+.SYSLOG.** CLASS(JESSPOOL) ID(OPER2) ACCESS(NONE)
```

The users belonging to group OPER, except OPER2, can access the SYSLOG and browse it.

Through SDSF, the SYSLOG can be viewed by selecting the DA panel and browsing the Master Scheduler address space. The access authority to this SYSLOG selection is controlled by the above entries in the JESSPOOL class.

When OPER2 attempts to browse the Master Scheduler address space, the following messages appear in the console log:

```
ICH408I  USER(OPER2  )  GROUP(OPER   )  NAME(JAMES          )
  C2JES2.+MASTER+.SYSLOG.STC04821.D0000102.?  CL(JESSPOOL)
  INSUFFICIENT  ACCESS  AUTHORITY
  FROM  C2JES2.+MASTER+.SYSLOG.**  (G)
  ACCESS  INTENT(READ  )  ACCESS  ALLOWED(NONE  )
```

JES2 SDSF: If OPER2 selects the LOG facility in SDSF, he may still be able to view the SYSLOG, unless the security administrator also defines the following profile in the SDSF resource class:

```
RDEFINE  SDSF  ISFCMD.ODSP.SYSLOG.JES*  UACC(NONE)

PERMIT  ISFCMD.ODSP.SYSLOG.JES*  CL(SDSF)  ID(OPER)  ACCESS(READ)
PERMIT  ISFCMD.ODSP.SYSLOG.JES*  CL(SDSF)  ID(OPER2)  ACCESS(NONE)
```

Now OPER2 cannot view the SYSLOG data set at all. For more information on the SDSF class, see Chapter 12, “SDSF Release 3” on page 205.

If security label checking is active, to be able to browse the SYSLOG from the Master Scheduler address space, a user must be logged on with a SECLABEL of SYSHIGH, as this is the SECLABEL of the Master Scheduler address space. To view the SYSLOG through SDSF, a user does not have to log on with a SECLABEL of SYSHIGH.

All SYSLOG spool data sets have a SECLABEL of SYSHIGH, so to print them, select a printing device that has a SECLABEL of SYSHIGH in its WRITER class profile.

For an operator or a system programmer to be able to issue the WRITELOG command, a profile must be defined in the OPERCMDS resource class if this class is active:

```
RDEFINE  OPERCMDS  MVS.WRITELOG  UACC(NONE)
PERMIT  MVS.WRITELOG  CL(OPERCMDS)  ACCESS(READ)  ID(OPER1)
```

The operator OPER1 can now issue the WRITELOG command. The highest access necessary is READ.

7.4 Process SYSOUT Requests

JES spool data sets are accessed through the Process SYSOUT (PSO) interface. This method of accessing the spool can now be checked for access authority to the data with the JESSPOOL resource class. Authorization checks are made for a number of requestors (Figure 36):

- TSO OUTPUT commands.
- External Writers.
- TSO TRANSMIT and RECEIVE commands.
- BDT SNA NJE requests, JES3 only.
- SDSF requests.
- User written code that uses the process SYSOUT interface to obtain JES spool data sets.

Authorization is checked for OPENs, GETs, and PUTs for the data sets. No checking is done for the ALLOCATE interface request.

The JES control block, pointing to where the data sets reside, contains the RTOKEN. The control block is the PDDB in JES2, and the JDS in JES3. When a process SYSOUT request is issued for a data set using the IEFSSREQ macro, the UTOKEN of the requestor is passed to JES. JES uses the RTOKEN of

the SYSOUT data set requested and passes it to SAF/RACF with the UTOKEN of the requestor. Security authorization is checked using the JESSPOOL resource class.

Note: If the SECLABEL class is active, the requestor's SECLABEL must dominate the data set SECLABEL. These SECLABELs are in the RTOKEN and UTOKEN, respectively.

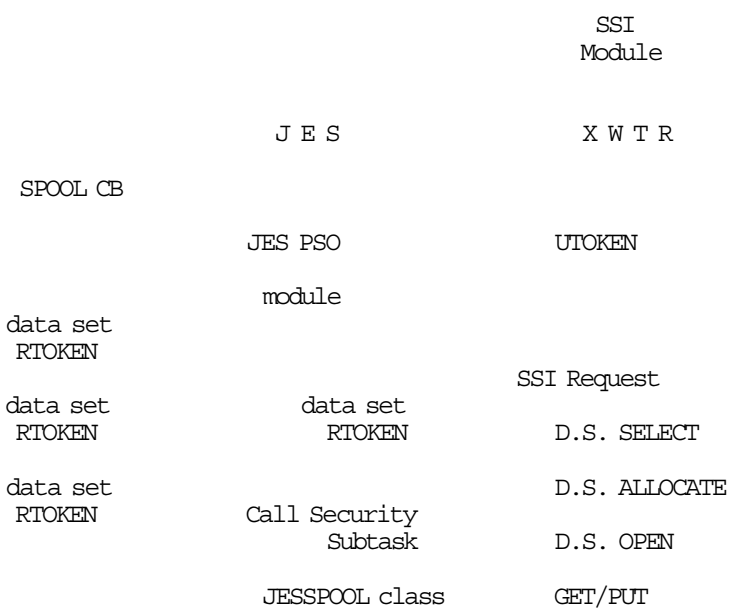

Figure 36. Process SYSOUT Interface Requests

7.4.1 PSO Interface Changes

In order to perform the proper authorization call (RACROUTE) for a TSO RECEIVE, JES must be able to distinguish between a PSO request for a userid and a PSO request for a writer name. Currently, the program name (SSSOPGMN in the IEFSSO macro) is passed without a writer name or a userid indicator. A flag byte, SSSOFLGA, has been added with the following definitions:

SSSOUSER Indicates that the program name field contains a userid. This flag is set by TSO RECEIVE processing.

SSSOWTRN Indicates that the program name field contains a writer name.

Note: JES sets the return code to the SSSOINVA value on return to the caller if both bits are set incorrectly.

7.4.1.1 Process SYSOUT GET Request

READ access is required if the userid (from UTOKEN) is not owner of the data set (from RTOKEN). A profile allowing access must exist:

```
PERMIT   nodename.**  CL(JESSPOOL)  ID(USER2)  ACC(READ)
```

7.4.1.2 Process SYSOUT PUT Request

ALTER access is required. PSO PUT occurs during UNALLOCATION even if not specifically called, for example: TSO OUTPUT jobname PRINT(*) NOKEEP).

```
PERMIT  nodename.**  CL(JESSPOOL)  ID(USER2)  ACC(ALTER)
```

Other examples of PUT requests are:

- Delete the held data set (SSSODELC)
- Change to a different class (SSSOSETC)
- Route to a different destination (SSSOROUT)
- Release the data set to the WTR queue (SSSORELC)

Note: Security calls are not made unless the attribute is actually changing. If the SAF call fails, no changes are made to the data sets attributes.

7.4.1.3 New IEFSSSO Macro Fields

The IEFSSSO macro has new fields added to support this new environment. The old data set name for SYSOUT data sets has new new fields added in the PSO extension:

SSSOWTRN Value in SSSOPGMN is a writer name (flag).

SSSOUSER Value in SSSOPGMN is a userid (flag).

SSSODDVR This version includes the 'old' DDNAME in the extension. The version value is placed into the field SSSSOVER on PSO requests.

SSSOPRCD Procedure name.

SSSOSTPD Step name.

SSSODDND DDNAME.

Note: The new data set name for SYSIN and SYSOUT data sets is returned in the field SSSODSN.

7.4.2 TSO OUTPUT Command

The TSO OUTPUT command is used to access spool data sets created by background jobs. The MSGCLASS and SYSOUT data sets should be assigned to a reserved class or explicitly held in order to be available at the terminal. The TSO OUTPUT command can be used to route a SYSOUT data set to a terminal, a sequential data set, or a member in a partitioned data set. The default options are to purge the data set and not to hold it.

Currently, held SYSOUT data sets processed through the TSO OUTPUT command are checked as follows:

- JCL/NOJCL in UADS or in the RACF TSOAUTH resource class

  The JCL attribute in UADS allows a user to use the SUBMIT, STATUS, CANCEL, and OUTPUT commands. If the UADS entries have been migrated to the RACF database, READ access to the JCL resource in the TSOAUTH resource class is needed.

- TSO CANCEL, OUTPUT, STATUS exit - IKJEFF53

The default exit supplied by IBM in SYS1.LINKLIB allows the OUTPUT command only for SYSOUTs of jobs that have job names starting with the userid plus one or more characters.

7.4.2.1 JESSPOOL Resource Class

A RACF class JESSPOOL can be defined to protect spool data sets. With the JESSPOOL class active, it is now possible to control SYSOUTs for user authorizaton not only by jobname, but also by nodeid, jobid, data set name, and userid as well. This greatly increases the flexibility in controlling OUTPUT command usage.

Access authorities allowed are READ and ALTER:

- READ allows authorized users to browse their SYSOUT data sets but not to purge them. Modifications to class and destination cannot be made, nor can the SYSOUT be deleted. If a user specifies DELETE in the TSO OUTPUT command and only READ access is allowed, RACF prevents the purge and records a violation.
- ALTER allows the user to purge the data set. The user can use any options of the OUTPUT command, for example:
  - Release the SYSOUT from the held queue.
  - Change its class or destination.
  - Delete the SYSOUT data set.

Note: An owner can access SYSOUTs without profiles in the JESSPOOL resource class.

SECLABEL Class active: With the SECLABEL class active, the user must be permitted to and logged on with the SECLABEL associated with the SYSOUT data set before being able to browse it.

7.4.2.2 Exit IKJEFF53 and JESSPOOL

With TSO/E 2.1.1, sample code for another IKJEFF53 exit is supplied in SYS1.SAMPLIB. In this sample code, for the OUTPUT command, a check is made to determine whether the JESSPOOL class is active. If it is, no jobname is restricted. If the JESSPOOL class is not active, a check is made to determine whether the JESJOBS class is active. If it is, no jobname is restricted.

Note: To allow the restriction of jobnames to be controlled by RACF, the IBM supplied exit **must** be replaced by the sample code in SYS1.SAMPLIB.

7.4.2.3 Exit IATUX30 - JES3

In JES3, this exit is entered after the IKJEFF53 exit. This exit has been changed to use field JQEOUSID instead of JQETUSID. JQEOUSID is the owning userid instead of the submitting userid.

Prior to JES3 3.1.3, JES3 considered the owner of the job to be the submittor of the job. Furthermore, the submittor of the job could transfer ownership of the job and SYSOUT by specifying //*MAIN USER= in the jobstream.

With 3.1.3, a change is made concerning ownership of jobs. The job owner, **owner userid**, is now the same as the userid under which the job executes, the **execution userid**. The job submittor, **submittor userid**, is still maintained in control blocks. This change allows the security product to control access to jobs and SYSOUT.

Execution userid: The userid that was specified by the USER= keyword on the job card. If USER= was not specified on the job card, the job owner userid is propagated from the job submitter (that is, from TSO SUBMIT). The execution userid is contained in the following control block fields:

- RQOUSID in IATYRSQ
- JCTOUSID in IATYJCT
- JQEOUSID in IATYJQE

This userid is used in all calls to the security product through the SAF interface. For installations with RACF 1.9 and desire to use its features to control jobs and SYSOUT, these are the fields to use.

Submittor userid: This is the userid who submitted the job. For example, when a TSO user submits a job, the job submitter is the TSO userid. The job submitter can also be modified by the USER= keyword on the //*MAIN statement. This keyword only affects the submitter's userid in the JES3 control blocks such as the JCT, JQE and RQ. The job submitter userid is contained in the following control block fields:

- RQTUSID in IATYRSQ
- JCTTUSID in IATYJCT
- JQETUSID in IATYJQE

This userid is not passed to the security product.

Owner userid: This is the userid that is used for jobs and SYSOUT when interfacing with JES3. Prior to JES3 3.1.3, this was the submittor userid. With native JES3 3.1.3, this is the execution userid. With APAR OY38471, the choice is left to the installation to determine which of these values should used to determine job ownership.

APAR OY38471: If an installation requires the job owner to be the submittor of the job as in releases prior to JES3 3.1.3, then APAR OY38471 should be evaluated. With APAR OY38471 installed, you can associate the job owner userid with the submittor userid on a job by job basis. The installation can change the default processing by adding code to user exit IATUX29 to turn on bit **JCTSBOWN** in byte JCTFL7 in the JCT build data area provided to the exit (and use the appropriate return code, 0 or 4). This indication is propagated to the following control block flags:

- JCTSBOWN in JCTFL7
- JQESBOWN in JQEFLG2
- RQSBOWN in RQFLG11

Additionally, the **JQE3JOWN** field contains the submittor userid instead of the execution userid. This is necessary for the proper functioning of IATUX30 in processing the commands, STATUS, CANCEL, and OUTPUT, without a jobname. This also affects the operator commands, *I U,ID= and *F U,ID=.

Note: Any installation processing performed in user exits or installation defined DSPs should be aware of the which userid is being used. This can be determined by examining the JCT, JQE, or RQ control blocks for the job as indicated above.

Example 1 - TSO user ABCDEF submits the following job

//GENER JOB MSGCLASS=A

- EXECUTION USERID - ABCDEF
- SUBMITTER USERID - ABCDEF
- OWNER USERID - ABCDEF

Since a userid was not specified on the job card, the submitter's user id (ABCDEF) is propagated to the job and used as the execution userid (JCTOUSID) and submittor's userid (JCTTUSID).

Example 2 - TSO user ABCDEF submits the following job

//GENER JOB MSGCLASS=A,USER=XYZ

- EXECUTION USERID - XYZ
- SUBMITTER USERID - ABCDEF
- OWNER USERID - XYZ (default)
- - ABCDEF (if the installation sets JCTSBOWN)

Since a userid was specified on the job card, it is used as the execution userid (JCTOUSID) and is the default job owner userid. The submitter's userid (JCTTUSID) is set to the TSO user who submitted the job. This userid can be made the job owner userid by setting JCTSBOWN.

Example 3 - TSO user ABCDEF submits the following job

//GENER JOB MSGCLASS=A

//*MAIN USER=FRED

- EXECUTION USERID - ABCDEF
- SUBMITTER USERID - FRED
- OWNER USERID - ABCDEF (default)
- - FRED (if the installation sets JCTSBOWN)

Since a userid was not specified on the job card, the submitter's userid (ABCDEF) is propagated to the job and used as the execution userid. Since USER=FRED was specified on the //*MAIN statement, it is used as the submitter's userid. The job owner userid is ABCDEF by default or FRED if JCTSBOWN is set.

Example 4 - TSO user ABCDEF submits the following job

//GENER JOB MSGCLASS=A,USER=XYZ

//*MAIN USER=FRED

- EXECUTION USERID - XYZ
- SUBMITTER USERID - FRED
- OWNER USERID - XYZ (default)
- - FRED (if the installation sets JCTSBOWN)

Since a userid was specified on the job card, it is used as the execution userid. Since USER=FRED was specified on the //*MAIN card, it is used as the submitter's userid. The job owner userid is XYZ (default) or FRED if JCTSBOWN is set. APAR OY38471 requires a rethinking of the fields added to JES3 in support of the security environment provided by RACF 1.9. Prior to this APAR, information on the job owner userid, execution userid, and the submittor userid was kept separate and distinct in the following control blocks (old name appears in parenthesis). The default JES3 processing requires the following labeling:

- JCTTUSID - Submittor userid
- JDABSUSR (JDABUSID) - Submittor userid
- JQETUSID - Submittor userid
- RQTUSID - Submittor userid

- JCTOUSID - Execution /job owner userid
- JDABOUSR (JDABRACU) - Execution/job owner userid
- JDABOGRP (JDABRACG) - Execution/job owner userid group
- JQEOUSID - Execution/job owner userid
- RQOUSID - Execution/job owner userid

If the installation chooses not to use RACF to control jobs or SYSOUT processing and use traditional JES3 concepts, the description of the information in these control blocks are:

- JCTTUSID - Submitting/job owner userid
- JDABSUSR (JDABUSID) - Submitting/job owner userid
- JQETUSID - Submitting/job owner userid
- RQTUSID - Submitting/job owner userid
- JCTOUSID - Execution userid
- JDABOUSR (JDABRACU) - execution userid
- JDABOGRP (JDABRACG) - execution userid group
- JQEOUSID - Execution userid
- RQOUSID - Execution userid

IATUX30 Considerations:

- With JES3 3.1.3, all JES3 interactions with the security product are with the execution userid as the job owner userid.
- IATUX30 is replaced by APAR OY38471. There is information in the module prologue to indicate how STATUS, CANCEL and OUTPUT processing is changed by OY38471 and the impact of RACF options on its operation.

7.4.2.4 RACF Definitions with TSO OUTPUT Command

The sample exit IKJEFF53 from SYS1.SAMPLIB **must** be installed in SYS1.LINKLIB. Assume the TSO users are defined as USER4 and USER99, and that the JESSPOOL class has been defined with a universal access of none:

```
RDEFINE   JESSPOOL   C2JES2.USER4.**   UACC(NONE)
PERMIT C2JES2.USER4.**   CLASS(JESSPOOL)   ID(USER99)   ACCESS(READ)
PERMIT C2JES2.USER4.**   CLASS(JESSPOOL)   ID(USER99)   ACCESS(ALTER)

SETROPTS   CLASSACT(JESSPOOL)
SETROPTS   RACLIST(JESSPOOL)
```

USER4 submits a job called PAYAAID. The 'USER=' statement on the jobcard is omitted and is propagated from the user's profile. An example of the jobcard is:

```
//PAYAAID JOB (999,POK),¢PAYCLERK¢,
```

Example 1

TSO user USER99 issues the TSO OUTPUT command in the following format:

```
TSO  OUTPUT  PAYAAID(J4375)
```

The TSO output command processing does not allow the job to be displayed and issues the following messages at the TSO terminal:

```
IKJ56339I NO HELD OUTPUT FOR JOB PAYAAID(JOB04375)
```

The following messages appear in the SYSTEM log:

```
IEF196I ICH408I USER(USER99  ) GROUP(USER    ) NAME(ANDRE       )
IEF196I ICH408I   C2JES2.USER4.PAYAAID.JOB04375.D0000002.JESMSGLG
IEF196I ICH408I   CL(JESSPOOL)
IEF196I ICH408I   INSUFFICIENT ACCESS AUTHORITY
IEF196I ICH408I   FROM C2JES2.USER4.** (G)
IEF196I ICH408I   ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

Example 2

In this example, USER99 has READ access and is able to browse the data but not purge it. USER99 then issues the TSO OUTPUT command in the following format to attempt to purge the data set:

```
TSO OUTPUT PAYAAID(J4375) DELETE
```

RACF does not allow the job to be displayed and issues the following messages at the TSO terminal:

```
IKJ56339I NO HELD OUTPUT FOR JOB PAYAAID(JOB04375)

At the SYSTEM log....

IEF196I ICH408I USER(USER99  ) GROUP(USER    ) NAME(ANDRE       )
IEF196I ICH408I   C2JES2.USER4.PAYAAID.JOB04375.D0000002.JESMSGLG
IEF196I ICH408I   CL(JESSPOOL)
IEF196I ICH408I   INSUFFICIENT ACCESS AUTHORITY
IEF196I ICH408I   FROM C2JES2.USER4.** (G)
IEF196I ICH408I   ACCESS INTENT(ALTER  )  ACCESS ALLOWED(READ   )
```

7.4.2.5 SECLABEL Class active

With SECLABEL checking active, it is possible to run a job with a SECLABEL higher than the one the user is logged on with, provided he is authorized to use the higher one. However, during the current TSO/E session, the OUTPUT command to process the output created by this job cannot be used. The user has to log off and log on with a SECLABEL that is higher or equal to the job's SECLABEL:

```
RDEFINE  JESSPOOL  C2JES2.USER1.**  UACC(NONE)
RDEFINE  JESSPOOL  C2JES2.OPER1.REPT*.**  UACC(NONE)

PERMIT  C2JES2.USER1.**  CL(JESSPOOL)  ID(USER2)  ACCESS(ALTER)
PERMIT  C2JES2.OPER1.REPT*.**  CL(JESSPOOL)  ID(USER2)  ACCESS(READ)
```

Job output is created by OPER1, USER1, and USER2. Some SYSOUT data sets have a blank SECLABEL fields and others have valid values in this field.

With SECLABEL checking inactive, OPER1 can access only his own SYSOUT data sets. USER1 can access only his own SYSOUT data sets. USER2 can view and change all SYSOUT data sets of USER1, and can view those SYSOUT data sets of OPER1 that have a jobname starting with the characters REPT. Of course, USER2 also has full access to his own SYSOUT data sets.

With the SECLABEL resource class active, the users OPER1, USER1, and USER2 can access the same SYSOUT data sets as above, except when the SYSOUT data set has a SECLABEL higher than the one

they are logged on with, or when the SYSOUT data set carries a SECLABEL for which they are not authorized.

Example

In this example, USER99 has been permitted ALTER access and can browse or purge the data set. USER99 logged on without a SECLABEL and because the data set has a SECLABEL, USER99 cannot access the data set. USER99 issues the TSO OUTPUT command to browse the data set:

```
TSO OUTPUT PAYAAID(J4376)
```

TSO command processing does not allow the job to be displayed and the following messages are displayed:

```
At the TSO terminal....

IKJ56339I NO HELD OUTPUT FOR JOB PAYAAID(JOB04376)

At the SYSTEM console....

IEF196I ICH408I USER(USER99  ) GROUP(USER    ) NAME(ANDRE       )
IEF196I ICH408I   C2JES2.USER4.PAYAAID.JOB04376.D0000002.JESMSGLG
IEF196I ICH408I   CL(JESSPOOL)
IEF196I ICH408I   INSUFFICIENT SECURITY LABEL AUTHORITY
IEF196I ICH408I   FROM C2JES2.USER4.** (G)
IEF196I ICH408I   ACCESS INTENT(READ   )
```

7.4.2.6 TSO OUTPUT Command Considerations

Note the following:

- The owner is always allowed to access data by using the OUTPUT command without a RACF definition.
- For RACF to check the authorization, the sample IKJEFF53 in SYS1.SAMPLIB **must** be installed.
- If the SECLABEL class is active, ensure that users wanting to access other user's data are logged on with the correct SECLABEL before attempting access to the data.
- Ensure that the data to be browsed with the TSO OUTPUT command is in the held SYSOUT class or is held by issuing HOLD operator command.

For information on JES2 SDSF requests, see Chapter 12, "SDSF Release 3" on page 205.

7.4.3 External Writers

An external writer is a started task used to process output. Because an external writer is a started task, it has a userid associated with it. This userid is specified in the Started Procedure Table when defining the writer to RACF. Output can be processed with an external writer by naming the writer on a DD statement that defines the output as follows:

```
//OUTXWTR  DD  SYSOUT=(Y,XWTR)
```

For the writer to be able to process the output, the writer's userid must be in an access list that permits the writer's userid to the SYSOUT data set. The minimum access required for external writers is ALTER.

The external writer can be defined with the TRUSTED attribute in the Started Procedures Table. However, external writers are installation-written programs and therefore it is strongly recommended that you avoid giving them the TRUSTED attribute.

Also, if the security policy in an installation requires security label checking, the SECLABEL associated with the external writer must be equal or higher than the SECLABEL associated with the SYSOUT data set.

7.4.4 TSO TRANSMIT/RECEIVE Commands

The TSO TRANSMIT (or XMIT) and RECEIVE commands are used to transmit and receive messages and data sets. This is similar to the SEND and LISTBC commands. The SEND command cannot, however, send messages or data sets to users on different nodes. The security implementation is the same as for SECLABELS. A user with a higher SECLABEL cannot send messages or data sets to users with SECLABELS that are defined with a lower access level.

As an example, assume that TSO users USER4 and USER99 are defined with SECLABELs:

```
Define SECLABELs....

  RDEFINE   SECDATA   SECLEVEL   ADDMEM(UNC/10,CON/150)
  RDEFINE   SECLABEL   UNC   SECLEVEL(UNC)
  RDEFINE   SECLABEL   CON   SECLEVEL(CON)

Permit users to SECLABELS....

  PERMIT UNC CLASS(SECLABEL) ID(USER4) ACCESS(READ)
  PERMIT CON CLASS(SECLABEL) ID(USER99) ACCESS(READ)
```

USER4 is defined to SECLABEL 'UNC' and USER99 defined to a higher SECLABEL of 'CON'. SECLABEL and SECDATA classes are active. USER4 transmits a message to USER99. USER99 issues the receive command and receives the message. USER99 transmits a message to USER4. When USER4 issues the RECEIVE command, RACF should deny access to the message. The message should be purged unless the JES2 TSO receive authorization exit (exit 38) is in use, which could divert or place a hold on the message. The format of the command that USER99 uses is:

```
XMIT node-name.userid
XMIT C2JES2.USER4 + message text.....
```

The command should fail with the following message:

```
IEF196I ICH408I USER(USER4    ) GROUP(P0112    ) NAME(ROBYN        )
IEF196I ICH408I   C2JES2.USER99.USER99.TSU05247.D0000115.? CL(JESSPOOL)
IEF196I ICH408I   INSUFFICIENT SECURITY LABEL AUTHORITY
IEF196I ICH408I   ACCESS INTENT(READ    )
ICH408I USER(USER4    ) GROUP(P0112    ) NAME(ROBYN                )
ICH408I   C2JES2.USER99.USER99.TSU05247.D0000115.? CL(JESSPOOL)
ICH408I   INSUFFICIENT SECURITY LABEL AUTHORITY
ICH408I   ACCESS INTENT(READ    )
$HASP321 USER99 OUTGRP=12.1.1 PURGED
         RECEIVER DOES NOT HAVE ACCESS TO DATA
```

When USER4, who has a lower SECLABEL, sends a message to USER99, USER99 receives the following message text:

```
TSO RECEIVE

INMR901I data set ** MESSAGE ** from USER4 on C2JES2
Text.............
INMR144I Sender notified of receipt
INMR900I ------------------------------------------------------------
INMR000I No more files remain for the receive command to access.
```

7.4.5 TSO XMIT and RECEIVE Considerations

The RACF principle of protecting data sets sent to users using the XMIT command is the same as for sending messages.

When transmitting messages and data sets to nodes other than the current node, RACF does not prohibit access.

7.4.6 IKJEFF53 User Exit

The IBM supplied exit IKJEFF53 allows the OUTPUT command only for SYSOUTs of jobs that have job names starting with userid. This exit is invoked before RACF is invoked. TSO/E Release 2.1.1 supplies sample code for another IKJEFF53 exit. In this sample code, for the OUTPUT command, check to see whether the JESSPOOL class is active. If it is, no JOBNAME checking is done. This ensures that the JESSPOOL output ″ownership″ is not restricted by this exit to jobnames that begin with USERID.

For the CANCEL command, check to see whether the JESJOBS class is active. If it is, no jobname checking is done. This ensures that the JESJOBS job ″ownership″ is not restricted by this exit to jobnames that begin with USERID.

If either or both classes are inactive, the exit functions as the normal IBM supplied exit for that command.

The IBM supplied exit IKJEFF53 **must** be replaced by the sample code to allow the installation to control the restriction of jobnames with the JESSPOOL and JESJOBS classes. However, the jobname checking that is currently done by the IBM supplied exit cannot easily be duplicated by using RACF; that is, you have to issue:

```
RDEFINE  JESJOBS  CANCEL.user_profile  UACC(NONE)
PERMIT  CANCEL.user_profile  ID(user)  ACCESS(READ)
```

for each user. This would be impractical for installation with large numbers of TSO users. For such installations, it would be more advisable to keep the old IKJEFF53 logic. The JESJOBS resource class for jobs canceling is discussed in 6.8.5, “JESJOBS Class for Job Canceling” on page 90.

7.5 Destination Control with WRITER Class Profiles

Today, JES processes output to any device that matches the output’s selection criteria. This selection does not provide enough security control because users can specify output attributes at will, and JES does not have user security information to restrict them.

Now you can control the specific devices certain users can access. These devices includes local printers, RJE attached printers, and NJE transmitters (logical devices).

Controls can now be placed on which users are allowed to have access to output devices through the use of the resource class **WRITER**. This control can be used in two ways:

- Is the user allowed to access the writer? Verify whether a user is authorized to access the output device or node. See 7.5.1, “User Access to Output Devices.”
- Is the writer allowed to print the data set? This is determined by verifying whether the output data security level is dominated by the output device security level. This check is valid if the SECLABEL class is active and the output device has a SECLABEL. See 7.5.2, “Data Access to Output Devices.”

7.5.1 User Access to Output Devices

An installation, using profiles in the WRITER class, can determine which writers can process which output. Authorization for a destination is implemented with the use of the WRITER class profiles using the following profile format:

```
jesname.LOCAL.devicename
jesname.RJE.devicename       (JES2)
jesname.RJP.devicename       (JES3)
jesname.NJE.nodename
```

For local and FSS printers and punches, the devicename must be PRTnnnn as used in the initialization deck for JES2 and the JNAME specified on the DEVICE statement in the JES3 initialization deck. This support can also be used to control output to remote terminal writers and NJE nodes.

To activate destination control for PRT10 you enter the following commands:

```
RDEFINE   WRITER   JES%.LOCAL.PRT10   UACC(NONE)
PERMIT   JES%.LOCAL.PRT10   CL(WRITER)   ACC(READ)   ID(SYSPROG)
SETROPTS   CLASSACT(WRITER)   RACLIST(WRITER)
```

The above example does not allow SYSOUT to be processed on PRT10 unless the owner of the SYSOUT data set is in the SYSPROG group. If the owner of a specified SYSOUT data set is not authorized for a particular output destination, the SYSOUT is not selected for printing. The data set remains on the spool.

7.5.2 Data Access to Output Devices

Profiles in the WRITER class can be assigned SECLABELs. This allows control on which devices can print output data. The authorization check is to determine if the output's security level meets the device's security level. Verification is based on whether the output's SECLABEL is dominated by the output device's SECLABEL. See 3.1, “SECLABEL Checking” on page 17 for a discussion of RVRSMAC checking.

Security label authorization checks for the WRITER class are reversed from the normal SECLABEL dominance check. For a printer, the device SECLABEL must dominate any output it is to print. If the output's SECLABEL is not dominated, the data set is not selected, and remains on the spool. This dominance check is reversed, since otherwise it would be possible to de-classify data by printing confidential material on a printer with a low security classification.

Using the profiles that follow, users in group SYSPROG can use PRT10 if the data to be printed has an equal or lower SECLABEL to BRCON.

```
RDEFINE   WRITER   JES%.LOCAL.PRT10   UACC(NONE)   SECLABEL(BRCON)
PERMIT   JES%.LOCAL.PRT10   CL(WRITER)   ACC(READ)   ID(SYSPROG)
```

7.5.3 Output Data Set Auditing

A SAF call for output destination security is done to check a users authorization to access the data. This check is made when the data set is printed and JESSPOOL class is checked. Since the owner of a job is propagated to the SYSOUT data sets produced by the job, this authorization call should not result in a denial of access. This SAF call should always pass and is for auditing purpose only.

The new fields in SMF type 6 record are DDNAME, SECLABEL, PROCESS MODE, USERID, DATA SET NAME, OUTPUT, and GROUPID.

Note: These new fields are available if you are using PSF/MVS controlled printers, but not all the new fields are obtained if you are using JES controlled printers.

There are internal changes in the interface between JES and PSF/MVS. New fields in the FSS macros, IAZFSIP and IAZJSPA are used for passing parameters between JES and PSF/MVS. The new fields are VERSION NUMBER, SECLABEL, USERID, DATA SET NAME, USER TOKEN, RESOURCE TOKEN, OUTPUT GROUP ID, JESNEWS INDICATOR, and FSS REASON CODE.

Chapter 8. NJE Security Control

This chapter discusses the implementation of NJE security in a JES environment. The requirements are to ensure that the security of the system is not compromised by NJE functions or jobs, and to enhance the overall security of the network without placing limitations on functions previously available.

NJE security performs the following functions:

- Controls jobs and SYSOUT entering a node on the basis of origin node, userid, groupid, and security label. This is controlled by a new RACF resource class, **NODES**.

  With RACF 1.9 installed, you can reject or accept jobs or SYSOUT:

  - From certain nodes
  - From certain userids at certain nodes
  - From certain groups at certain nodes
  - With certain SECLABELs from certain nodes

  Also, you can accept jobs without password checking:

  - From certain nodes
  - From certain userids at certain nodes
- Controls who can send jobs and data to another node on the basis of destination node, sending userid, and security label. This is controlled by a new RACF resource class, **WRITER**.
- Propagates user validation across the network, so passwords do not have to be sent with the job. When passwords are required, JES provides a means for encrypting them.
- Permits surrogate job submission across NJE nodes.
- Permits different USERIDs, GROUPIDs, or SECLABELs on different nodes and provides a means for translating them to locally-defined values under installation control. This is controlled by a new parameter, **ADDMEM**, when defining the **NODES** profile.

  Translating submitting userids on jobs not requiring passwords can be based on:

  - Node or userid that submitted the job
  - All userids from a given node to a specific userid

  Translation of submitting SECLABEL from a node can be based on:

  - Node or SECLABEL of submitting job
  - All SECLABELs from a given node to a specific SECLABEL

  Translation of submitting GROUP can be based on:

  - Node or GROUP of submitting job
  - All GROUPs from a given node to a specific GROUP.

These controls are exercised primarily on the origin and destination nodes, not on store-and-forward nodes. It is assumed that all the nodes and links in the network are trusted to the extent that they do not make unauthorized changes to security fields in the NJE headers. Centralized SAF/RACF management allows mixed levels of nodes in a network, including previous levels of JES and RACF as well as VM/RSCS and VSE/POWER.

8.1 NJE Networks

NJE security applies only to JES 3.1.3 and RACF 1.9 environments. Enhancements to the SAF controls the access to other nodes by determining the eligibility of those nodes to receive and send data (jobs, SYSOUT and TSO transmit). Data is prevented from entering a node based on the submittor's userid, groupid, security label or node. RACF also has the capability of encrypting the password in the NJE job header and of translating userids, groupids, and SECLABELs to specified local values. These values are the submittor information for locally destined jobs and the owner information for locally destined SYSOUT.

When an installation considers providing security in an NJE network, the following are key components:

- RACF resource classes that affect jobs and SYSOUT entering and and leaving NJE nodes.
- Software levels of the nodes in the network.
- The level of trust assigned to each node.
- Security tokens and the information in the NJE headers.

8.1.1 Resource Classes that Affect NJE

Security is controlled by activating and deactivating certain RACF classes. These classes for the NJE environment are:

- NODES - Authorize nodes (all inbound data and translation)
- WRITER - Authorize transmitters (all outbound data)
- JESINPUT - Authorize receivers (port of entry)
- JESJOBS - Authorize execution (jobs submission)
- SURROGAT - Authorize surrogacy (use another's RACF profile)
- SECLABEL - Authorize valid data classifications
- RACFVARS - Identify common RACF attributes

In addition, profiles within these classes have to be defined to specifically identify authorized work. These definitions can be at a global or discrete level and may selectively apply to userids, groupids, or SECLABELs. Once these profile definitions are in place and the classes activated, all JES work flowing between nodes - jobs, SYSOUT, NJE messages, and data transmitted from TSO - is checked for compliance.

8.1.2 Node Software Levels

A network may consist of nodes at many different software levels as shown in Figure 37. The software installed characterizes the node level as:

Uplevel Node MVS/JES 3.1.3 and RACF 1.9 installed. Valid tokens are created by RACF and propagated by JES.

RACF 1.8.1 or an earlier release is installed. Valid tokens are created by SAF and propagated by JES..

Default Node MVS/JES 3.1.3 installed; however, the following conditions may exist:

RACF 1.9 is installed, but is inactive or RACF is not the security product installed. Under these conditions, default tokens are created by SAF and propagated by JES.

Downlevel Node Pre-MVS/JES 3.1.3 or VM/RSCS installed. JES does not send tokens at this level.

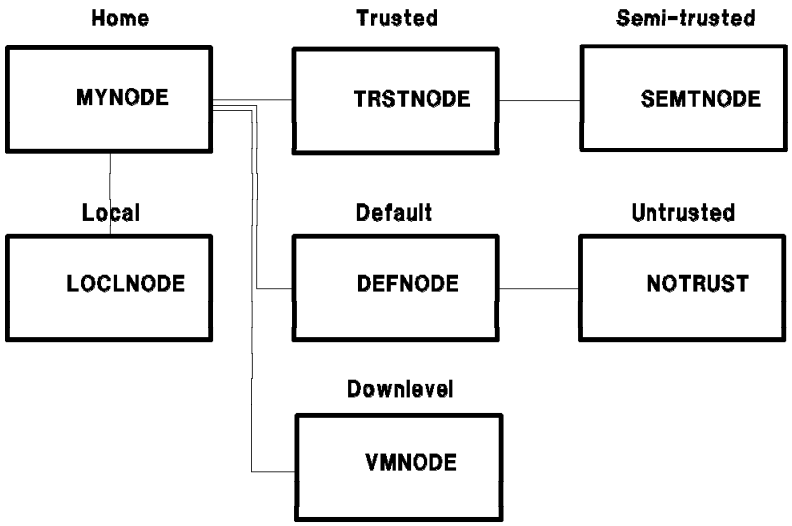

Figure 37. NJE Nodes in a Network

8.1.3 NJE Levels of Trust

The level of RACF verification is determined by the level of trust, as defined in the RACF NODES class profiles. Each access level corresponds to one of the levels of trust (Table 10).

Note: For those NJE nodes that are not defined to RACF through the NODES profiles, the present JES security controls are used.

This level of trust indicates to SAF not only the security catagorization but also the degree of RACF verification which still has to be done for that user.

Table 10. Levels of Trust and ACCESS Level Required

Level of Trust	Verification	NODES Profile Access
Trusted	No password re-verification	UACC(UPDATE)
Trusted (default or down-level)	No password re-verification	UACC(CONTROL)
Semi-trusted	Password required	UACC(READ)
Untrusted	Jobs/SYSOUT purged	UACC(NONE)
Local	Password optional	Define in &RACLNDE profile
Unknown	Need password and userid	Undefined to RACF

Levels of trust are defined as follows:

Trusted The node and userid(s) are accepted as validated without a password. The ″trusted″ attribute is defined by **UACC(UPDATE)** in the NODES profile definition.

For users on default or down-level nodes, the ″trusted″ attribute is defined by **UACC(CONTROL)** in the NODES profile definition. In this case, there is no valid token in the NJE job header, but the user is still considered validated.

Semi-trusted The node is trusted enough to ensure that NJE headers are valid, but the user must supply a password (which can be encrypted). The ″semi-trusted″ attribute is defined by **UACC(READ)** in the NODES profile definition. This is similar to previous releases of JES, except for the encryption.

Note: A submitting userid is not used in the UTOKEN.

Untrusted The node is not trusted, and any jobs received from this node are purged with a message sent to the submitting user. This is defined to RACF as **UACC(NONE)**.

Local Nodes (including the one we are on) that are defined as part of &RACLNDE to RACF. It is assumed that all users and groups (and SECLABELs) are defined identically on all local nodes, and share a compatible RACF database. Do not define a node as ″local″ if this is not the case.

Unknown If the node is not defined to RACF, an unknown user token is explicitly created through a TKNUNKWN session type on VERIFYX. This unknown user token is created and used for NJE work on store-and-forward nodes and for SYSOUT files when the owner is not known.

8.1.4 Tokens in an NJE Environment

Security tokens are associated with all RACF validated work in the system. Each token contains security information about each user and resources associated with that user. Their association continues for the duration of that user. In this way, tokens constructed in certain environments need not be rebuilt each time a user submits a job, but rather the verified tokens are propagated, which is more efficient. This applies to jobs and data transmitted to other nodes as well.

When a job is submitted from one node for execution at another node, a token is part of the security section of the NJE job header. This token is called an **STOKEN** and contains the security information associated with the submittor. At the execution node, this information can be propagated, translated, and overriden to create a **UTOKEN**. This UTOKEN then becomes the life-of-job token while it exists in the execution node.

Tokens are defined as follows:

STOKEN A security token representing the submittor of a job. It is usually transmitted with an NJE job to the execution node before it is transformed into a job token.

UTOKEN A security token representing a user. It is a job token when a job is attempting to access a resource.

(There are two cases where a UTOKEN is carried with a job before execution: (1) If a job is re-routed to a different execution node, and (2) If a job is off-loaded before execution and then re-loaded.)

RTOKEN A security token representing a resource, which may be a job or a SYSIN/SYSOUT data set when a user (or job) is attempting to access it.

Default A skeleton security token created by "default-level" nodes (JES/SP 3.1.3) when RACF is not active. If a default token is created, the skeleton contains all relevant security information that is available either from the job card, submittor tokens, or elsewhere. All the data has been filled in by SAF, but none of it has been verified. Some of it can be trusted because a down-level security product did some verification, but the degree to which it can be trusted depends on the environment.

For example, if RACF 1.8 is installed, in general, a submittor token should have a verified userid and group in it because in order to submit a job, the user must have logged on and therefore was validated. Therefore, the token is not validated until the job begins execution.

Note: If SAF determines that the security product does not understand a VERIFYX type request, it transforms the request into an equivalent VERIFY request.

Unknown User Special security token used at store-and-forward nodes, or created if a user is undefined on the receiving node. See 5.2, "Security Tokens" on page 66 for additional information.

8.1.4.1 NJE Header Security Data

There are new security sections in the NJE headers. This information is in the form of tokens and new fields as shown in Figure 38.

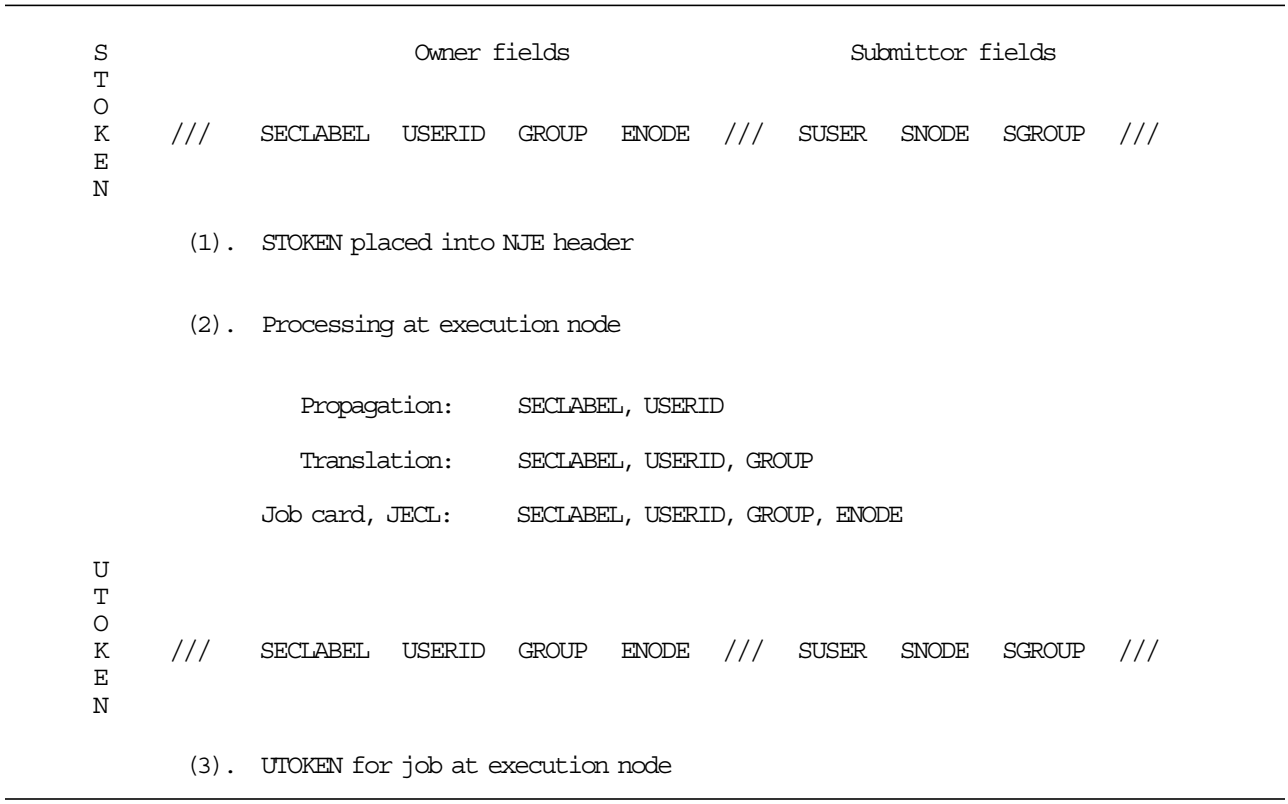

Figure 38. NJE Tokens for Job Submission

The tokens ands fields are:

STOKEN = Submittor's Security Information: The following owner fields are copied from the submittor's UTOKEN into the submittor fields as shown in Figure 38. They are not translated or overridden. This token is passed in the NJE header in field NJHTTOKN.

SUSER Submitting userid (NJHGORGR)

SGROUP Submitting group (not in NJE Header, General Section)

SNODE Submitting/origin node (NJHGORGN)

If the job comes from a default or down-level node, they are propagated from these fields in the NJE job header. (GROUP is copied to SGROUP (submitting group), but not propagated to the Group of the UTOKEN).

UTOKEN = Job's Security Information: The UTOKEN for an NJE job is created during JES input processing for the submitted job at the execution node, as shown in Figure 38 on page 121.

The following fields may be propagated from the STOKEN, translated through NODES profile, or overridden by JCL (or JECL) in the creation of the UTOKEN:

SECLABEL Execution SECLABEL (this may be specified on the job card, translated through the NODES profile, or propagated from the STOKEN).

USERID Execution userid (this may be specified on the job card, translated through the NODES profile, or propagated from the STOKEN).

GROUP Execution group (this is not propagated, but may be specified on the JOB card, or translated through the NODES profile).

ENODE Execution node (from a JES JECL statement, defaulted by JES, or taken from the NJE job header).

8.1.4.2 NJE Fields Passed to SAF

RACF decisions are based on where a job originated and on profiles defined in the NODES class. JES2 and JES3 both pass information available in the NJE header to RACF. Fields that are to be passed are:

NJHTTOKN This field has been added to a new security section of the job header. It contains a mapped version of a SAF security token. If present, this represents the primary source of security data for a job.

NJHTF0JB This flag bit indicates whether the token in NJHTTOKN represents a job (flag on) or the submittor of the job (flag off). It determines whether the token is passed as a STOKEN to SAF or as a UTOKEN.

NJHGORGN This field is used as the submitting node by RACF.

Note: An exposure exists if any MVS node or VM/RSCS node in the network does NOT have the following APAR applied:

- **JES2 - APAR OY17284**
- **JES3 - APAR OY18334**
- **VM/RSCS - APAR VM37404**

Prior to this APAR, this field was used by notify processing as the notification node. Since this is available to be set by the end user, there is no way to guarantee the accuracy of this field. The APAR creates a new field in the general section,

NJHGNTYN, which is now used to contain the notify node. See the APAR documentation for the new JES control block fields that contain the notify node name.

NJHGNTYN The notify or report-to node name. Notify messages are sent to the notify-node.notify-userid.

NJHGORGR In the absence of a security token, this field is used as the submitting userid. This field may be blank (if job came in on a local card reader or from an internal reader on a system that does not have one of the above mentioned APARs installed, a remote name, or a valid userid. These possibilities must be taken into account when defining profiles).

NJHGXEQN Used for JES2 spool reload (not used in JES3). If no security token exists, this field is used in conjunction with a non-blank NJH2USR field in determining if a reverification of a reloaded job is required. There are more details in the spool offload sections.

NJHGPASS Password to use if not specified on the job card.

NJHGNPAS New password to use if not specified on the job card.

NJHGF1PE Flag; if set, NJHGPASS is encrypted.

NJHGF1NE Flag; if set, NJHGNPAS is encrypted.

Valid combinations for the encryption flags are:

- NJHGF1PE and NJHGF1NE both off
- NJHGF1PE and NJHGF1NE both on
- NJHGF1PE on, with NJHGNPAS set to all blanks or all zeros

Any other combination will be considered an error, and be treated as any other protocol errors.

8.2 RACF NODES Class

RACF 1.9 has a new general resource class, NODES. The profiles of the class contain only the names of all controlled network nodes on this system.

Jobs or SYSOUT coming from other nodes are validated during input service processing in the receiving node. The NODES class is used to verify whether the transmitting node and its userid, groupid, or SECLABEL is trusted, semi-trusted, or untrusted. Userids may be translated only from trusted nodes, but groupids and SECLABELs may be translated from trusted or semi-trusted nodes.

The NODES class is used to:

- Control NJE access by node, userid, groupid, and SECLABEL
- Control whether job passwords are required
- Translate userids, groupids, and SECLABELs to local values.

NODES class profiles are defined for security controls when jobs or SYSOUT is received at a node. There are separate profiles for jobs and SYSOUT. For inbound jobs, the NODES profile defines the submittor's userid, groupid, and SECLABEL.

The NODES RACF class uses the profile name:

nodename.keyword.name

Where:

nodename The name of the submitting node (subnode in example below).

keyword The type of work, job, or SYSOUT, together with the attribute types userid, groupid, and SECLABEL you want to be validated. For example, **USERJ** specifies userid for jobs, and **USERS** specifies userid for SYSOUT. You can specify **USER%** to indicate jobs and SYSOUT. This applies also to GROUP% and SECL% for jobs and SYSOUT. The other values are GROUPJ, GROUPS, SECLJ, and SECLS.

name Specifies the userid, groupid, or specific SECLABEL.

NODES Profile Name Examples:

subnode.RUSER.userid	NJE commands
subnode.USERJ.userid	Job userid
subnode.GROUPJ.groupid	Job groupid
subnode.SECLJ.seclabel	Job SECLABEL
xeqnode.USERS.userid	SYSOUT userid
xeqnode.GROUPS.groupid	SYSOUT groupid
xeqnode.SECLS.seclabel	SYSOUT SECLABEL

8.2.1 NODES Profile Access Levels

There are two objectives in NODES class profiles; one defines a level of trust, and the other defines the translation of userid, groupid, and SECLABEL. The level of trust is defined by the access level defined in the (UACC). There are four levels of access available for the NODES class:

NONE Allows no work from the specified node to be entered into this system.

READ Allows work from the specified node to be entered into this system if a userid and password are given.

UPDATE Allows work from the specified node to be entered into this system because the node is trusted and no additional verification is required.

CONTROL Allows work from default or down-level nodes to have the trusted attribute, which allows the user to be validated.

The UACC controls jobs on a node-by-node basis. Any user that requires verification on the current node has to be defined to RACF on this node.

You can also use a generic access entry to cover all work from any nodes entering this system.

NJE security requires the activation of the NODES class for inbound work, and the WRITER class for outbound work. An installation can choose to protect either inbound work, outbound work, or both. For inbound jobs and SYSOUT, you can decide whether to protect jobs, SYSOUT, or both. You can also determine which users or group of users are allowed to enter NJE jobs or SYSOUT and whether SECLABELs are valid for processing.

Note: Permits are never used in the NODES class; only UACCs are used.

8.2.2 Controlling Jobs and SYSOUT Entering a Node

The NODES class controls job and SYSOUT entering a node. Define profiles and activate the NODES class to allow RACF to make the necessary decisions; for example:

- Two nodes, C2JEST and C2JES2, are defined as trusted nodes for jobs and SYSOUT submitted by all userids.

Definition at C2JEST:

```
RDEFINE   NODES    C2JES2.USERJ.*   UACC(UPDATE)
RDEFINE   NODES    C2JES2.USERS.*   UACC(UPDATE)
```

Definition at C2JES2:

```
RDEFINE   NODES    C2JEST.USERJ.*   UACC(UPDATE)
RDEFINE   NODES    C2JEST.USERS.*   UACC(UPDATE)
```

Note: With the use of a **%** , you can define a single profile that includes both jobs and SYSOUT:

```
RDEFINE   NODES    C2JEST.USER%.*   UACC(UPDATE)
```

To allow some users jobs and SYSOUT to be received and to deny a specific user or a group of users, specify:

```
RDEFINE   NODES    C2JEST.USER%.*   UACC(UPDATE)
RDEFINE   NODES    C2JEST.USER%.USER1    UACC(NONE)
RDEFINE   NODES    C2JEST.GROUP%.GROUPA   UACC(NONE)
```

Jobs and SYSOUT for USER1 and GROUPA are purged while all other users are accepted.

Note: A PERMIT command is not used with the NODES class. Any access list created by a PERMIT command is not used in the NODES class. The NODES class profiles have to be RACLISTed.

8.3 Translation between NJE Nodes

For inbound jobs, security information can be translated to local values. RACF can be used to replace inbound userids, groupids, and SECLABELs with locally defined values. This is a very important new concept in NJE security controls and avoids the complexity of defining identical userids, groupids, and SECLABELs in every node in an NJE Network. The USERID, GROUPID, and SECLABELs of jobs or SYSOUT is:

- Defined in a NODES profile.
- Translated at the receiving node. The ADDMEM field of the profile is used for translation to locally defined names.

There is no access list for profiles in the NODES class; authorization is controlled through universal access (**UACC**), which determines the level of trust. The UACC determines the level of RACF translation that takes place:

- **UPDATE** access is necessary for translation for jobs.
- Only **READ** access is necessary for SYSOUT.

The example in Figure 39 shows a job submitted by user A on NODEA, which executes on NODEB under user B, and the output is sent to NODEC under user C, using the NODES profiles shown.

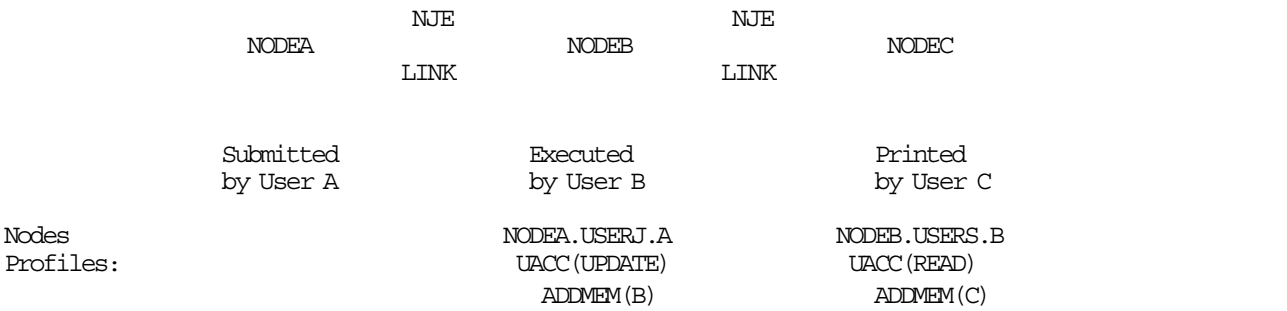

Figure 39. NJE Translation Example

8.3.1 Userid Translation

The example in Figure 40 shows a job submitted by user FRED on NODEA, which executes on NODEB under user MARY, and the output is returned to NODEA. The profile for this translation is:

```
RDEFINE   NODES   NODEA.USERJ.FRED   UACC(UPDATE)   ADDMEM(MARY)    -    at NODEB
```

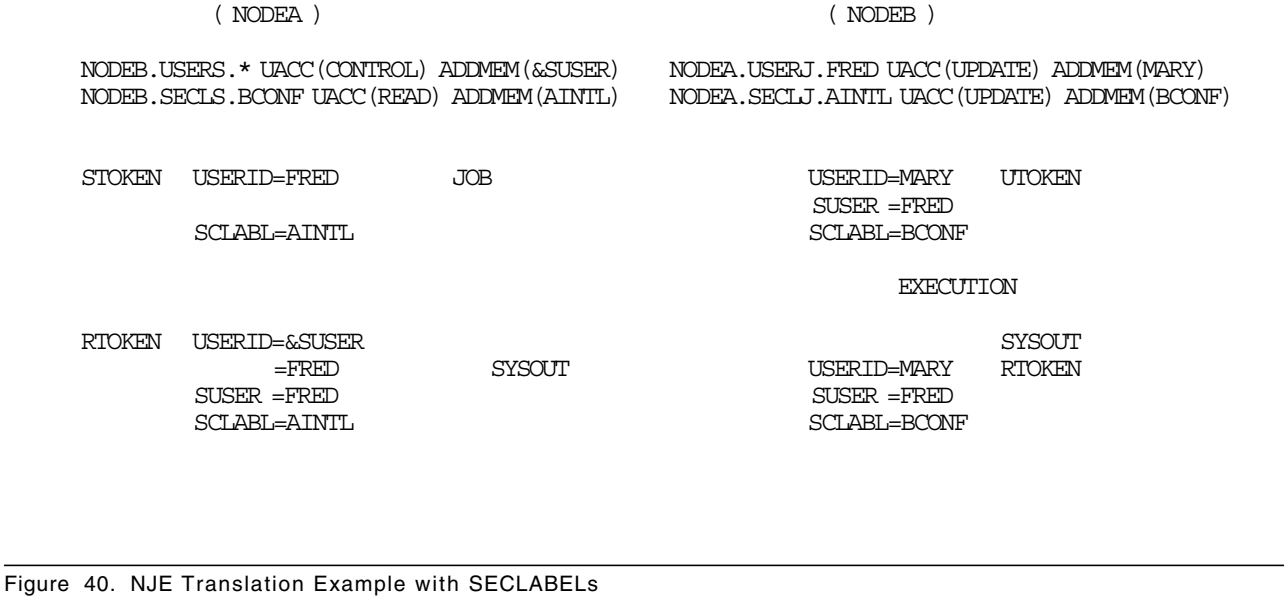

Figure 40. NJE Translation Example with SECLABELs

For the submitting user, FRED, to be able to view the output, the ADDMEM specification in the NODES profile requires a special translation by specifying **&SUSER**. This translates the ownership of the output to the submitting user.

The NODES profile for the translation of the SYSOUT back to the submitting user is:

```
RDEFINE   NODES   NODEB.USERS.*  UACC(CONTROL)   ADDMEM(&SUSER)    -    at NODEA
```

8.3.2 SECLABEL Translation

It is very probable that the SECLABELs at two nodes in a network will be different. Since SECLABELs can also be translated, NODES profiles should exist at each node to translate the SECLABELs. In Figure 40 on page 126, SECLABEL AINTL at NODEA is translated to SECLABEL BCONF at NODEB. The output that is returned to NODEA with a SECLABEL of BCONF is translated to a SECLABEL of AINTL using the following NODES profiles:

```
RDEFINE   NODES   NODEB.SECLS.BCONF   UACC(READ)   ADDMEM(AINTL)       -    at NODEB

RDEFINE   NODES   NODEA.SECLJ.AINTL   UACC(UPDATE)   ADDMEM(BCONF)       -    at NODEA
```

8.3.3 Nodes with and without SECLABELs

It is possible to send NJE jobs or SYSOUT from a node using SECLABELs to a node that does not use SECLABELs. If a user submits a job from a node that has SECLABELs turned on to be executed at a node that has SECLABELs turned off, the output returned to the submittor will have a SECLABEL of RACSLUNK (RACF SECLABEL unknown) assigned to the SYSOUT. The user will not be able to access this SYSOUT unless the RACSLUNK is defined as a SECLABEL and the user is authorized to it or a profile exists to translate the RACSLUNK SECLABEL. The following is an example to translate the RACSLUNK SECLABEL at the receiving node where SECLABELs are active:

```
NODES   NODEB.SECLS.RACSLUNK   UACC(READ)   ADDMEM(SYSLOW)
```

8.4 Defining Local Nodes

With the &RACLNDE profile in the RACF RACFVARS class, you can specify to RACF what nodes are to be considered local.

The following example defines to RACF that nodes NODE1 and NODE2 are to be treated as local nodes. These nodenames also have to be defined in the new RACF class NODES:

```
RDEFINE   RACFVARS   &RACLNDE   ADDMEM(NODE1 NODE2)
```

The use of &RACLNDE allows the installation to protect resources that include the nodename in the profile, even when the local nodename changes or when there is more than one local nodename.

Any node that is contained in &RACLNDE is assumed to be local and can be represented by &RACLNDE. Therefore, the use of &RACLNDE in a profile name allows the installation to refer to all local nodes without actually knowing their names. &RACLNDE should be defined as a profile in the RACF RACFVARS class (Figure 41).

A single asterisk in a profile name refers to all nodes, both external and local, in the absence of more specific profiles.

You should code &RACLNDE in place of the nodename in the profile name when the profile has to cover resources for the local node. RACF internally substitutes the local nodename when matching the entity name to the profile. This is a second way to change the local nodename. By specifying both the old and the new nodename to RACF as a local node, you can easily migrate to the new nodename:

```
RDEFINE   NODES    &RACLNDE.USERS.*   UACC(UPDATE)
```

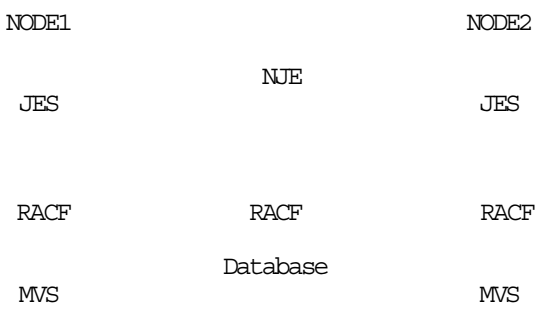

Figure 41. Defining Local Nodes with &RACLNDE

8.5 Receiving SYSOUT Considerations

SYSOUT that is received from other nodes in a network is controlled by the NODES class. The following considerations are necessary for the user who submitted the job to be able to view the output:

- Node level of trust as defined in the UACC of the NODES profile
- Userid assigned to the SYSOUT data set
- JESSPOOL class active
- Translation of SYSOUT using &SUSER in the NODES profile
- Submitting nodes or execution node defined in &RACLNDE

8.5.1 Userid Assignment for SYSOUT

Table 11 indicates the owner userid assigned to the SYSOUT data set when a user submits a job from NODEA to execute at NODEB and the SYSOUT that is created is returned to NODEA. The submitting user can view the SYSOUT in all cases except when the SYSOUT is purged.

Table 11. SYSOUT Received at Submission Node from Execution Node

NODES Profile UACC	Userid Assigned	Comments
None	n/a	SYSOUT purged (Note 1)
READ	????????	(Note 2)
UPDATE	Submitting user	
CONTROL	Submitting user	

Note:

(1). A JES user exit may be used to override the purge of the SYSOUT data set. Use Exit 39 with JES2 and Exit IATUX67 with JES3.

(2). The submitting user can view the output even though the owner userid is ???????? because the JESSPOOL class is not active.

8.5.2 JESSPOOL Class Active

If the JESSPOOL class is active, a submitting user cannot always view his own output. The SYSOUT data set must have the submitting userid as part of the data set name. When a data set is received, the userid assigned to the data set depends on the level of trust defined in the NODES profile as shown in Table 12.

Table 12. SYSOUT Received at Submission Node with JESSPOOL Class Active

NODES Profile UACC	Userid Assigned	Comments
READ	????????	Submittor cannot view output
UPDATE	Submitting user	
CONTROL	Submitting user	

If the receiving node is semi-trusted, the userid assigned to the output is ????????. The data set name has the following format:

```
????????.JOBA.JOB00111.D0000100.JESMSGLG
```

It is possible to create profiles to allow access to data sets with a userid of ????????, for example:

```
RDEFINE  JESSPOOL  NODEA.????????.**  UACC(NONE)

PERMIT  NODEA.????????.**  CL(JESSPOOL)  ID(SECADM)  ACC(UPDA  TE)
```

The security adminstrator, SECADM, can process the output data sets.

8.5.3 Translation and &RACLNDE

For the previous example with UACC(READ) in Table 12, profiles can be created to translate the assigned userid to the submitting userid (Table 13). For nodes that are semi-trusted, UACC(READ), define the following profiles:

```
RDEFINE  RACFVARS  &RACLNDE  ADDMEM(NODEA)

RDEFINE  NODES  NODEA.USERS.*  UACC(READ)  ADDMEM(&SUSER)
```

Note: NODEA in the RACFVARS class profile is the submitting node.

Table 13. Semi-trusted Node Defined as Local Node

NODES Profile UACC	Userid Assigned	Comments
READ	Submitting user	Submittor can view output

Another method to ensure the submitting userid is assigned to the output is:

```
RDEFINE  RACFVARS  &RACLNDE  ADDMEM(NODEB)

RDEFINE  NODES  NODEA.USERS.*  UACC(READ)
```

Note: NODEB in the RACFVARS class profile is the execution node. The ADDMEM for translation is not required with this definition.

8.6 RACF WRITER Class

Security control for jobs or SYSOUT transmitted to other NJE nodes is defined differently than for receiving. The password is not verified and there is no translation.

With the enhancements to SAF and RACF, control is provided using the WRITER class.

The profile name is:

```
subsystem-name.NJE.nodename
```

Where:

subsystem-name Either JES2 or JES3, or the name of your Job Entry Subsystem.

NJE NJE indicates the WRITER class profile is an NJE profile.

nodename Specifies the node name of the sending node.

The WRITER class controls who is allowed to submit jobs or SYSOUT to another node. Unlike the NODES class, there is no distinction between jobs or SYSOUT on outbound NJE transmissions.

The WRITER class in RACF controls where output can be sent. You can restrict or authorize the use of writers for local printers and punches, JES2 RJE devices, JES3 RJP devices, and NJE transmissions. For NJE, RACF verifies the security of jobs and SYSOUT transmissions to ensure that the user is authorized to send data to another node in a network.

8.6.1 Controlling Jobs and SYSOUT Leaving a Node

Two nodes, C2JEST and C2JES2, are defined to allow jobs and SYSOUT to be transmitted from all userids.

Definition at C2JEST:

```
RDEFINE  WRITER  JES%.NJE.C2JES2  UACC(READ)
```

Definition at C2JES2:

```
RDEFINE  WRITER  JES%.NJE.C2JEST  UACC(READ)
```

To deny a specific user from sending jobs or SYSOUT, use the PERMIT command:

```
PERMIT  WRITER  JES%.NJE.C2JES2  CLASS(WRITER)  ID(USER1)  UACC(NONE)
```

8.6.2 SECLABELs with WRITER Class

The destination node can also have a SECLABEL assigned that can be used to control the jobs or SYSOUT being networked. This control requires a SECLABEL check when the SECLABEL and the WRITER classes are both active. If the SECLABEL of the job or SYSOUT that is being transmitted is not equal to or higher than the SECLABEL specified in the RDEFINE, then transmission is denied.

Assign a SECLABEL as follows:

```
RDEFINE  WRITER  JES%.NJE.C2JES2  UACC(READ)  SECLABEL(BRCON)
```

8.7 Store-and-Forward Nodes

The security in store-and-forward nodes protects jobs and SYSOUT at the node (Figure 42).

The store-and-forward node can be at any level. These levels are described in the section Chapter 8, “NJE Security Control” on page 117. At store-and-forward nodes, the following are required:

- An undefined user token is assigned to the jobs or SYSOUT.
- If SECLABELs are active at the node, SYSHIGH is assigned.
- Access to the jobs or SYSOUT is through JESSPOOL or operator commands. No access is granted without permission through profiles.

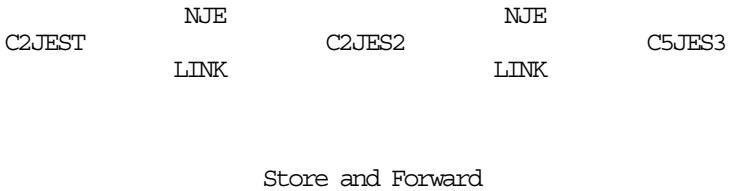

Figure 42. Store-and-Forward Example

No WRITER class check is made when a job or SYSOUT is forwarded (transmitted) from a store-and-forward node.

8.7.1 Up-level Nodes

Jobs or SYSOUT being transmitted (store-and-forward) through an NJE node are assigned an undefined user token. This eliminates the need to translate to a userid at the store-and-forward node, and also protects jobs and SYSOUT without any specific RACF resources being defined.

A token is created for the store-and-forward node, but it is not put into the header when transmitting to the next node.

If SECLABEL is in use at this node, a SECLABEL of SYSHIGH is assigned. The default is SYSLOW, but JES specifies SYSHIGH.

Normally, TSO users cannot affect (CANCEL, OUTPUT, RECEIVE) jobs and SYSOUT at store-and-forward nodes. As needed, profiles can be set up for selected users. For example, the following PERMIT command allows a systems programmer access to all SYSOUT if a more specific profile does not exist:

```
PERMIT  *.**  CL(JESSPOOL)  ACCESS(READ)  ID(SYSPROG)
```

Unknown Origin Node: As in previous releases, no checks are made on the origin node if the job or SYSOUT is not destined for this node.

Unknown Destination Node: As in previous releases, jobs are executed locally if the execution node is undefined. SYSOUT is printed locally if the destination node is not defined. JES continues to process the job locally, but jobs may fail because of RACF checks.

8.7.2 Default Nodes

The processing is the same, except that a default token is created.

8.7.3 Down-level Nodes

No tokens are created.

8.8 Submitting Jobs using NJE

Locally submitted jobs that execute elsewhere must be protected on the submission node. The token used depends on the form of submission and the source of the job.

8.8.1 Two-Job-Card Jobs

When the JES2 /*XMIT, the JES3 XMIT or the JES3 //*ROUTE XEQ cards are used to send a job through an NJE network, two job cards are needed. For the two-job-card jobs:

- The first job card is verified at the submitting node.
- The password, if required at the receiving node, must be on the second job card.

Note: See Table 10 on page 119 for levels of trust.

The token used to protect the transmitted job on this system is the token created by the VERIFYX of the first job card. This token is also placed in the job header field NJHTTOKN as the submittor token.

Note: JES3 always requires two job cards. JES3 has a password control option that determines whether the verification is to be done. See 8.8.4.2, “JES3 Password Encryption” on page 133.

8.8.2 Single-Job-Card Jobs

In the case of a JES2 /*ROUTE XEQ (or /*XEQ) card, there is only one job card associated with the NJE job. This job card is intended for the target node and is not to be verified at the source node. For these jobs as well as the JES3 jobs that do not require verification, if the job was submitted from an internal reader, the token of the job that allocated the reader is used to protect the job on the source system. If the job was submitted from a ″physical″ reader (remote or local), then an unknown user token is assigned to the job. The token is placed in the NJHTTOKN field of the job header as a submittor token, STOKEN.

8.8.3 Tokens for Store-and-Forward Jobs

All store-and-forward jobs are protected on a store-and-forward node with an unknown user token. This limits access to the job to only those explicitly permitted to it. On a store-and-forward node, the token used to protect the job on the node is kept separate from the token in the header. If the job is rerouted locally, the token in the header must be used to verify the job.

8.8.4 Password Encryption

Password encryption by RACF is optional and is specified in the JES initialization parameters. If a password cannot be decrypted, a job routed to a node that does not support encryption, fails.

On the receiving side, JES uses the encryption flags to determine whether the passwords are already encrypted. JES input processing normally encrypts all passwords. This processing is bypassed if the password encryption has already taken place. The method of encryption is unknown to JES. A DES algorithm is used by RACF to encrypt passwords.

Note the following:

- Both USER= and PASSWORD= must be specified on the job card for encryption to take place.
- Encryption is done when a job is sent on the job transmitter at the node of origin.
- At non-origin nodes, the password encryption option has no effect. If a job or SYSOUT is rerouted from a node that supports encryption to one that does not, the job fails.
- The fields NJHGNPAS and NJHGPASS, in the NJE header may be encrypted.

8.8.4.1 JES2 Password Encryption

Password encryption is controlled by the NODE initialization statement:

```
NODE(nnnn)  PENCRYPT=YES|NO
```

PENCRYPT=YES specifies that the password is to be encrypted before the job is sent to the specified node for execution.

Note: For jobs with two job cards, no encryption takes place and no passwords are placed in the NJE header.

The encryption option can also be specified dynamically by:

```
$T  NODE(nnnn),PENCRYPT=YES|NO
```

The operator can display and change a nodes password encryption option:

- **$D NODE(n),PENCRYPT - display option.**
- **$T NODE(n),PENCRYPT=YES|NO** - change option.

8.8.4.2 JES3 Password Encryption

Password encryption is controlled by the NJERMT initialization statement. Passwords are verified at the execution node unless PWCNTL=LOCALCHK is specified:

```
NJERMT  PWCNTL=SENDENC|SENDCLR|LOCALCHK
```

Where:

SENDENC Specifies encryption for the first job card password.

SENDCLR Specifies no encryption of passwords. The password from the first job card is placed in the NJE header in clear text. At the execution node, the password from the second job card is used. If no password is specified, the password from the first job card that is placed in the NJE header is used.

LOCALCHK Specifies that the password is to be verified locally. No passwords are sent to the execution node. The first job card is used for the submittor's security information.

Note: If RACF is inactive or not installed, the password is verified at the execution node.

The encryption option can be changed dynamically by:

```
*F NJE,NAME=nodename,PWCNTL=option
```

8.9 NJE Authorization Flow

JES requests RACF services by issuing the RACROUTE macro. The MVS System Authorization Facility (SAF) processes this request. If RACF 1.9 is installed, SAF passes the security information specified by JES. RACF evaluates this security information and returns the results to JES who then enforces the results of the security check.

Figure 43 shows those RACF classes that exert control over the NJE environment. When a job is transmitted to a node, five resource classes, if active, may be used for decisions on the processing. These processing decisions are:

NODES Is the job or SYSOUT allowed at this node?

SURROGAT Is there a surrogate user submitting the job?

JESINPUT Are there profiles that prohibit this job from entering from the adjacent node?

JESJOBS Is there a profile that controls the job name of the entering job?

WRITER The job has successfully executed and created output. Is the output authorized to be sent back to the origin node?

JESSPOOL If the job output remains at the execution node, check if the user is authorized to view it. Is the USERID permitted to view the output at this node?

NODES	Decision Process
Determine if work is permitted. Is the transmitting node known ? RACF command format: RDEFINE NODE subnode.groupj.userid RDEFINE NODE xeqnode.groups.userid OR RDEFINE NODE subnode.userj.userid RDEFINE NODE xeqnode.users.userid OR RDEFINE NODE subnode.seclj.userid RDEFINE NODE xeqnode.secls.userid	YES Continue processing at SURROGAT. NO Fail job with a JCL error. The following message is displayed on the console of the executing node: ICH408I USER( ) GROUP( ) NAME(???) C2JEST.USERJ.* CL(NODES) NETWORK JOB ENTRY - JOB FROM NODE C2JEST NOT AUTHORIZED Continue processing at WRITER.
SURROGAT	Decision Process
Check for surrogate user submitting job. Does a profile exist? RACF command format: RDEFINE SURROGAT owner-userid.SUBMIT	YES Continue processing at JESINPUT. NO Submittor not owner and no profile exists. Continue processing at WRITER.
JESINPUT	Decision Process
Validate the NJE receiver nodename. Is the port-of-entry authorized for this job ? RACF command format: RDEFINE JESINPUT adjacent nodename	YES Continue processing at JESJOBS. NO Fail the job with a JCL error. The following message is displayed on the console of the executing node: ICH408I USER(USER3) GROUP(P0112) NAME(RON) LOGON/JOB INITIATION - NOT AUTHORIZED TO TERMINAL/CONSOLE Continue processing at WRITER.
JESJOBS	Decision Process
Check for an authorized submittor. Is GROUPID,USERID,SECLABEL permitted to submit work ? RACF command format: RDEFINE JESJOBS SUBMIT.currentnode.jobname.userid	YES Initialize and schedule execution of job. At end-of-job continue at JESSPOOL or WRITER depending on destination of output. NO Fail the job with a JCL error. The following message is displayed on the console of the executing node. ICH408I USER(USER3) GROUP(P0112) NAME(RON) LOGON/JOB INITIATION - USER IS NOT AUTHORIZED TO JOB USER3JX Continue processing at WRITER.
WRITER	Decision Process
Check if the job output is authorized for transmission back to the originating node. Node known for specific or implied routing ? RACF command format: RDEFINE WRITER subsys.NJE.nodename	YES Return the job output to designated node. NO The following messages are displayed on the console of the executing node: ICH408I USER(USER3) GROUP(P0112) NAME(RON) ICH408I JES2.NJE.C2JEST CL(WRITER) ICH408I PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING ICH408I ACCESS INTENT(READ) ACCESS ALLOWED(NONE) $HASP111 USER3JX -- INVALID PRINT/PUNCH ROUTE Purge the output at the execution node.
JESSPOOL	Decision Process
If the job output remains at the execution node, check if the user is authorized to view it. Is the USERID permitted to view the output at this node ? RACF command format: RDEFINE JESSPOOL node.userid.jobname.jobid.Ddsid.dsname	YES Permit access either through TSO OUTPUT or SDSF NO The following message is displayed on the console of the executing node: ICH408I USER(USER3) GROUP(P0112) NAME(RON) ICH408I C2JES2.USER3.USER3JX.JOB03136.D0000002.JESMSGLG ICH408I CL(JESSPOOL) ICH408I PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING ICH408I ACCESS INTENT(READ) ACCESS ALLOWED(NONE)

Figure 43. RACF Resource Classes used by NJE (in order of checking)

8.9.1 NJE Node Scenarios

The main objectives of the nodes class profiles are to define the level of trust and the translation characteristics for userids, groupids and security labels. Consider the implementation of NJE security in the environment where there are two nodes - C2JES2 and C2JEST. The following examples clearly show the RACF definitions for each node and then the progress of a job is traced through the network to the execution node and back again. The messages that would normally be encountered by the submittor and those displayed at the master console are also shown. Messages pertaining to the job executing at another node are returned to the submittor in the JES job log.

The following definitions are used in the NJE scenarios that are described in this section.

Profiles and classes defined in C2JES2

```
RDEFINE   NODES    C2JEST.USERJ.*   UACC(UPDATE)
RDEFINE   NODES    C2JEST.USERS.*   UACC(UPDATE)

RDEFINE   JESINPUT   C2JEST   UACC(READ)

RDEFINE   JESJOBS    SUBMIT.C2JES2.*.*   UACC(READ)

RDEFINE   JESSPOOL   C2JEST.*.**   UACC(READ)

RDEFINE   WRITER     JES2.NJE.C2JEST   UACC(READ)

SETROPTS   CLASSACT   (NODES,JESINPUT,JESJOBS,JESSPOOL,WRITER)
```

Profiles and classes defined in C2JEST

```
RDEFINE   NODES    C2JES2.USERJ.*   UACC(UPDATE)
RDEFINE   NODES    C2JES2.USERS.*   UACC(UPDATE)

RDEFINE   JESINPUT   C2JES2   UACC(READ)

RDEFINE   JESJOBS    SUBMIT.C2JEST.*.*   UACC(READ)

RDEFINE   JESSPOOL   C2JES2.*.**   UACC(READ)

RDEFINE   WRITER     JEST.NJE.C2JES2   UACC(READ)

SETROPTS   CLASSACT   (NODES,JESINPUT,JESJOBS,JESSPOOL,WRITER)
```

Scenario 1 - Submission from an Unknown Node

- At node C2JES2, node C2JEST is not defined to RACF with NODES profiles.
- Job USER3J1 arrives at node C2JES2 and is processed with a userid of USER3 and executes normally. A password is required on the job card.
- USER3 is defined in the execution node.

The job is processed as shown below.

```
Node: C2JES2

RACF Status   - SETROPTS NOCLASSACT(NODES JESINPUT JESJOBS)
                SETROPTS NOCLASSACT(WRITER JESSPOOL)

                NJE related classes inactive.
                C2JEST is unknown to this node.
```

```
Node: C2JEST

RACF Status   - SETROPTS NOCLASSACT(NODES JESINPUT JESJOBS)
                SETROPTS NOCLASSACT(WRITER JESSPOOL)

                NJE related classes inactive.
                C2JES2 is unknown to this node.

Exercise      - USER3 submits a job (USER3J1)
                to execute on node C2JES2
Expected      - Job to execute at C2JES2,
Results         USER3 to be kept informed of
                progress. Output to be
                returned to C2JEST


<===            USER3J1 is submitted

Terminal Response Area

$HASP122 USER3J1 (JOB00051 FROM C2JEST) RECEIVED AT C2JES2 CN(00)
$HASP526 USER3J1 TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 USER3J1 (JOB00051 FROM C2JEST) ENDED AT C2JES2 CN(00)
$HASP546 USER3J1 SYSTEM OUTPUT RECEIVED AT C2JEST CN(00)
```

```
SYSLOG

$HASP100 USER3J1  ON L8.JR1      NORTHRUP
$HASP373 USER3J1  STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 USER3J1  ENDED
$HASP309    INIT  1 INACTIVE ******** C=A
$HASP530 USER3J1  ON L8.ST1            25 RECORDS
$HASP534 L8.ST1   INACTIVE
$HASP250 USER3J1  IS PURGED
```

```
SYSLOG

$HASP100 USER3J1  ON INTRDR      NORTHRUP           FROM TSU00047
USER3
$HASP520 USER3J1  ON L8.JT1
SE ¢JOB00051 $HASP122 USER3J1  (JOB00051 FROM C2JEST)
RECEIVED AT C2JES2¢,LOGON,USER=(USER3)
SE ¢JOB00051 $HASP526 USER3J1  TRANSMITTED FOR EXECUTION AT
C2JES2¢,LOGON,USER=(USER3)
SE ¢JOB00051 $HASP165 USER3J1  (JOB00051 FROM C2JEST) ENDED
AT C2JES2¢,LOGON,USER=(USER3)
$HASP524 L8.JT1   INACTIVE
$HASP250 USER3J1  IS PURGED
SE ¢JOB00051 $HASP546 USER3J1  SYSTEM OUTPUT RECEIVED AT
C2JEST¢,LOGON,USER=(USER3)
$HASP540 USER3J1  ON L8.SR1            25 RECORDS


JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM  SMF2  --  NODE C2JES2

 JOB00051  $HASP373 USER3J1 STARTED - INIT 1 - CLASS A - SYS SMF2
 JOB00051  $HASP395 USER3J1 ENDED
------ JES2 JOB STATISTICS ------
//USER3J1  JOB ,NORTHRUP,CLASS=A,MSGCLASS=X,NOTIFY=USER3
***ROUTE    XEQ C2JES2
```

Scenario 1 - Results and Comments

- Normal NJE processing occured and the output was returned correctly.

Scenario 2 - Submission from an Untrusted Node

- At node C2JES2, node C2JEST is defined to RACF with a NODES profile:

  ```
  RDEFINE   NODES   C2JEST.USERJ.*   UACC(NONE)
  ```

- Job USER3J2 arrives at node C2JES2 and is rejected.

The job is processed as shown below.

```
Node: C2JES2

RACF Status   - RDEFINE NODES (C2JEST.USERJ.*) UACC(NONE)
                RDEFINE NODES (C2JEST.USERS.*) UACC(NONE)
                SETROPTS CLASSACT(NODES)

                These definitions activate NJE
                checks. C2JEST is defined as
                ¢untrusted¢


SYSLOG

$HASP100 USER3J2  ON L8.JR1      NORTHRUP
ICH408I USER(        ) GROUP(        ) NAME(???             )
  C2JEST.USERJ.* CL(NODES   )
  NETWORK JOB ENTRY - JOB FROM NODE C2JEST   NOT AUTHORIZED
$HASP530 USER3J2  ON L8.ST1            11 RECORDS
$HASP534 L8.ST1   INACTIVE
$HASP250 USER3J2  IS PURGED
```

```
Node: C2JEST

RACF Status   - SETROPTS CLASSACT(NODES)

                The NODES class is active but
                C2JES2 is ¢unknown¢.


Exercise      - USER3 submits a job (USER3J2)
                to execute on node C2JES2
Expected      - Job is not permitted
Results         to run at C2JES2 as the
                originating node has been
                defined as ¢untrusted¢

<===            USER3J2 is submitted

Terminal Response Area

$HASP122 USER3J2 (JOB00040 FROM C2JEST) RECEIVED AT C2JES2 CN(00)
$HASP526 USER3J2 TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 USER3J2 (JOB00040 FROM C2JEST) ENDED AT C2JES2 -
                 JCL ERROR CN(00)
$HASP546 USER3J2 SYSTEM OUTPUT RECEIVED AT C2JEST CN(00)

SYSLOG

$HASP100 USER3J2  ON INTRDR      NORTHRUP           FROM TSU00039
USER3
SE ¢JOB00040 $HASP122 USER3J2  (JOB00040 FROM C2JEST)
RECEIVED AT C2JES2¢,LOGON,USER=(USER3)
SE ¢JOB00040 $HASP526 USER3J2  TRANSMITTED FOR EXECUTION AT
C2JES2¢,LOGON,USER=(USER3)
SE ¢JOB00040 $HASP165 USER3J2  (JOB00040 FROM C2JEST) ENDED
AT C2JES2 - JCL ERROR¢,LOGON,USER=(USER3)
$HASP520 USER3J2  ON L8.JT1
$HASP524 L8.JT1   INACTIVE
$HASP250 USER3J2  IS PURGED
SE ¢JOB00040 $HASP546 USER3J2  SYSTEM OUTPUT RECEIVED AT
C2JEST¢,LOGON,USER=(USER3)
$HASP540 USER3J2  ON L8.SR1            11 RECORDS


JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

 JOB00040 ICH408I USER(        ) GROUP(        ) NAME(???       )
          C2JEST.USERJ.* CL(NODES   )
          NETWORK JOB ENTRY - JOB FROM NODE C2JEST NOT AUTHORIZED
------ JES2 JOB STATISTICS ------
//USER3J2  JOB ,NORTHRUP,CLASS=A,MSGCLASS=X,NOTIFY=USER3
***ROUTE    XEQ C2JES2
```

Scenario 2 - Results and Comments

- The job submits and output is returned correctly.
- The job does not fail with a JCL error. The job is failed by the NODES profile check.
- The JES2 job log contains messages describing the failure.

Note: In a JES3 system, the message at the terminal response area is:

```
IAT6108   JOB   USER3J2 (JOB00040)   FAILED BY SECURITY CHECK
```

Scenario 3 - Submission from a Semi-Trusted Node

- At node C2JES2, node C2JEST is defined to RACF with a NODES profile. This requires a password on the job card for jobs transmitted from C2JEST:

  ```
  RDEFINE   NODES   C2JEST.USERJ.*   UACC(READ)
  ```

- Job USER3J2 arrives at node C2JES2 and is accepted because of the password on the job card.

- USER3 is defined in the execution node.

The job is processed as shown below.

```
Node: C2JES2

RACF Status   - RALTER NODES (C2JEST.USERJ.*) UACC(READ)
                RALTER NODES (C2JEST.USERS.*) UACC(READ)
                SETROPTS RACLIST(NODES) REFRESH

                These definitions alter the C2JEST
                node profile to ¢semi-trusted¢
                The option JES(BATCHALLRACF) is active.

SYSLOG

$HASP100 USER3J3  ON L8.JR1      NORTHRUP
ICH70001I USER3    LAST ACCESS AT 16:35:48 ON FRIDAY, FEBRUARY 16,
1990
$HASP373 USER3J3  STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 USER3J3  ENDED
$HASP309    INIT  1 INACTIVE ******** C=A
$HASP530 USER3J3  ON L8.ST1            28 RECORDS
$HASP534 L8.ST1   INACTIVE
```

```
Node: C2JEST

RACF Status   - The NODES class is active but
                C2JES2 is ¢unknown¢.

Exercise      - USER3 submits a job (USER3J3)
                to execute on node C2JES2
Expected      - Job is permitted
Results         to run at C2JES2 provided the
                USERID and PASSWORD is supplied
                on the jobcard.

<===            USER3J3 is submitted

Terminal Response Area

$HASP122 USER3J3 (JOB00061 FROM C2JEST) RECEIVED AT C2JES2 CN(00)
$HASP526 USER3J3 TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 USER3J3 (JOB00061 FROM C2JEST) ENDED AT C2JES2 CN(00)
$HASP546 USER3J3 SYSTEM OUTPUT RECEIVED AT C2JEST CN(00)

SYSLOG

$HASP100 USER3J3  ON INTRDR      NORTHRUP           FROM TSU00055
USER3
SE ¢JOB00061 $HASP122 USER3J3  (JOB00061 FROM C2JEST)
RECEIVED AT C2JES2¢,LOGON,USER=(USER3)
SE ¢JOB00061 $HASP526 USER3J3  TRANSMITTED FOR EXECUTION AT
C2JES2¢,LOGON,USER=(USER3)
$HASP520 USER3J3  ON L8.JT1
$HASP524 L8.JT1   INACTIVE
$HASP250 USER3J3  IS PURGED
SE ¢JOB00061 $HASP165 USER3J3 (JOB00061 FROM C2JEST) ENDED
AT C2JES2¢,LOGON,USER=(USER3)
SE ¢JOB00061 $HASP546 USER3J3 SYSTEM OUTPUT RECEIVED AT
C2JEST¢,LOGON,USER=(USER3)
$HASP540 USER3J3  ON L8.SR1            28 RECORDS

JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

ICH70001I USER3   LAST ACCESS AT 16:35:48 ON FRIDAY,
                  FEBRUARY 16, 1990
$HASP373  USER3J3 STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395  USER3J3 ENDED
------ JES2 JOB STATISTICS ------
//USER3J3  JOB ,NORTHRUP,CLASS=A,MSGCLASS=X,NOTIFY=USER3,
//             USER=USER3,PASSWORD=
***ROUTE    XEQ C2JES2
```

Scenario 3 - Results and Comments

- Normal NJE processing occured and the output was returned correctly.
- The password is not viewable in the SYSOUT.

Scenario 4 - Submission from a Trusted Node

- At node C2JES2, node C2JEST is defined to RACF with a NODES profile.

```
RDEFINE   NODES   C2JEST.USERJ.*   UACC(UPDATE)
```

- Job USER3J4 arrives at node C2JES2 and is accepted without a password on the job card.
- USER3 is defined in the execution node.

The job is processed as shown below.

```
Node: C2JES2

RACF Status   - RALTER NODES (C2JEST.USERJ.*) UACC(UPDATE)
                RALTER NODES (C2JEST.USERS.*) UACC(UPDATE)
                SETROPTS RACLIST(NODES) REFRESH

                These definitions change the C2JEST
                node profile to ¢trusted¢
```

```
Node: C2JEST

RACF Status   - The NODES class is active but
                C2JES2 is ¢unknown¢.

Exercise      - USER3 submits a job (USER3J4)
                to execute on node C2JES2
Expected      - Job is permitted
Results         to run at C2JES2 without providing
                the USERID and PASSWORD on the
                jobcard even if the option
                JES(BATCHALLRACF) is active.

<===            USER3J4 is submitted

Terminal Response Area

$HASP122 USER3J4 (JOB00062 FROM C2JEST) RECEIVED AT C2JES2 CN(00)
$HASP526 USER3J4 TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 USER3J4 (JOB00062 FROM C2JEST) ENDED AT C2JES2 CN(00)
$HASP546 USER3J4 SYSTEM OUTPUT RECEIVED AT C2JEST CN(00)
```

```
SYSLOG

$HASP100 USER3J4  ON L8.JR1      NORTHRUP
ICH70001I USER3    LAST ACCESS AT 21:05:51 ON FRIDAY, FEBRUARY 16,
1990
$HASP373 USER3J4  STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 USER3J4  ENDED
$HASP309    INIT  1 INACTIVE ******** C=A
$HASP530 USER3J4  ON L8.ST1            27 RECORDS
$HASP534 L8.ST1   INACTIVE
```

```
SYSLOG

$HASP100 USER3J4  ON INTRDR      NORTHRUP           FROM TSU00055
USER3
$HASP520 USER3J4  ON L8.JT1
SE ¢JOB00062 $HASP122 USER3J4  (JOB00062 FROM C2JEST  )
RECEIVED AT C2JES2¢,LOGON,USER=(USER3)
SE ¢JOB00062 $HASP526 USER3J4  TRANSMITTED FOR EXECUTION AT
C2JES2¢,LOGON,USER=(USER3)
$HASP524 L8.JT1   INACTIVE
$HASP250 USER3J4  IS PURGED
SE ¢JOB00062 $HASP165 USER3J4  (JOB00062 FROM C2JEST  ) ENDED
AT C2JES2¢,LOGON,USER=(USER3)
$HASP540 USER3J4  ON L8.SR1            27 RECORDS
SE ¢JOB00062 $HASP546 USER3J4  SYSTEM OUTPUT RECEIVED AT
C2JEST¢,LOGON,USER=(USER3)


JES2 JOB LOG

J E S 2  J O B  L O G  --  SYSTEM  SMF2  --  NODE  C2JES2

ICH70001I USER3    LAST ACCESS AT 21:05:51 ON FRIDAY,
                   FEBRUARY 16, 1990
$HASP373  USER3J4  STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395  USER3J4  ENDED
------ JES2 JOB STATISTICS ------
//USER3J4  JOB ,NORTHRUP,CLASS=A,MSGCLASS=X,NOTIFY=USER3
***ROUTE    XEQ C2JES2
```

Scenario 4 - Results and Comments

- Normal NJE processing occured and the output was returned correctly.
- Propagation occured successfully.

Scenario 5 - Submission from a Local Node

- Nodes C2JES2 and C2JEST are defined to RACF as local nodes.
- Job USER3J5 arrives at node C2JES2 and is accepted without a password on the job card.

The job is processed as shown below.

```
Node: C2JES2

RACF Status   - RDEFINE RACFVARS (&RACLNDE) ADDMEM(C2JES2 C2JEST)
                RDEFINE NODES (&RACLNDE.USERJ.*) UACC(UPDATE)
                SETROPTS CLASSACT(RACFVARS)

                These definitions identify C2JES2 and C2JEST
                as local nodes. Both have identically defined
                users, groups and seclabels. C2JES2 is a trusted
                node.
```

```
Node: C2JEST

RACF Status   - Identical to the situation described for
                C2JES2. C2JEST is a trusted node.

Exercise      - USER3 submits a job (USER3J5)
                to execute on node C2JES2
Expected      - Job should be permitted to run
Results         at C2JES2 without providing
                the USERID and PASSWORD on the
                jobcard even if the option
                JES(BATCHALLRACF) is active.

<===            USER3J5 is submitted

Terminal Response Area

$HASP122 USER3J5  (JOB00500 FROM C2JEST  ) RECEIVED AT
                  C2JES2 CN(00)
$HASP526 USER3J5  TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 USER3J5  (JOB00500 FROM C2JEST  ) ENDED AT C2JES2 -
                  JCL ERROR CN(00)
$HASP546 USER3J5  SYSTEM OUTPUT RECEIVED AT C2JEST CN(00)
```

```
SYSLOG

$HASP100 USER3J5  ON L8.JR1      USER3
$HASP373 USER3J5  STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 USER3J5  ENDED
$HASP309    INIT  1 INACTIVE ******** C=A
$HASP530 USER3J5  ON L8.ST1            34 RECORDS
$HASP534 L8.ST1   INACTIVE
$HASP250 USER3J5  IS PURGED
```

```
SYSLOG

$HASP100 USER3J5  ON INTRDR      NORTHRUP             FROM TSU00498
$HASP520 USER3J5  ON L8.JT1
SE ¢09.57.48 JOB00500 $HASP122 USER3J5  (JOB00500 FROM C2JEST  )
RECEIVED AT C2JES2¢,LOGON,USER=(USER3)
SE ¢09.57.48 JOB00500 $HASP526 USER3J5  TRANSMITTED FOR EXECUTION
AT C2JES2¢,LOGON,USER=(USER3)
SE ¢09.57.50 JOB00500 $HASP165 USER3J5  (JOB00500 FROM C2JEST  )
ENDED AT C2JES2¢,LOGON,USER=(USER3)
$HASP524 L8.JT1   INACTIVE
$HASP250 USER3J5  IS PURGED
ICH70001I USER3   LAST ACCESS AT 09:48:17 ON THURSDAY, AUGUST 2
SE ¢09.57.52 JOB00500 $HASP546 USER3J5  SYSTEM OUTPUT RECEIVED AT
C2JEST¢,LOGON,USER=(USER3)
$HASP540 USER3J5  ON L8.SR1            34 RECORDS
$HASP250 USER3J5  IS PURGED


JES2 JOB LOG

J E S 2  J O B  L O G  --  SYSTEM SMF2  --  NODE  C2JES2

JOB00500  ICH70001I USER4    LAST ACCESS AT 09:48:17 ON THURSDAY,
             AUGUST 2, 1990
JOB00500  $HASP373 PAYAAID  STARTED - INIT  1 - CLASS A - SYS SMF2
JOB00500  $HASP395 PAYAAID  ENDED
 ----- JES2 JOB STATISTICS ------
//USER3J5 JOB ,NORTHRUP,CLASS=A,MSGCLASS=X,NOTIFY=USER3
***ROUTE   XEQ C2JES2
```

Scenario 5 - Results and Comments

- Normal NJE processing occured and the output was returned correctly.

8.9.2 NJE Translation Scenarios

Translation enables RACF to authorize the execution of jobs at nodes with different userids, groupids, and SECLABELs than the submitting node. It removes the necessity for identical environments and the duplication associated with them. Translation is activated when a NODES profile is defined that includes the ADDMEM(userid) parameter. Once this environment has been established, all jobs entering the execution node and meeting the translation criteria are translated to the userid specified in the ADDMEM parameter.

In the following scenarios, a user running at node C6JES2 submits a job to execute at node C2JES2. Because of the translation in affect for inbound jobs from C6JES2, the userid is translated and the job is permitted to execute with the translated RACF authority. The following examples clearly show the RACF definitions for each node and then the progress of a job is traced through the network to the execution node and back again. The messages that would normally be encountered by the submittor and those displayed at the master consoles are also shown.

Scenario 1 - Submission, No Translation

```
Node: C2JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C6JES2 has been defined as a ¢trusted¢ node.
                Neither surrogacy or translation is active.
                Userid USER04 is undefined.
```

```
Node: C6JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C2JES2 has been defined as a ¢trusted¢ node.


Exercise      - USER04 submits a job (USER04A)
                to execute on node C2JES2 without
                the USER=USER3 parameter coded
                on the jobcard.
Expected      - Job to execute at node C2JES2 as
Results         if submitted by USER3. Results to
                be returned to C6JES2.

<===            USER04A is submitted

Terminal Response Area

$HASP122 USER04A  (JOB04116 FROM C6JES2  ) RECEIVED AT
                  C2JES2 CN(00)
$HASP526 USER04A  TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 USER04A  (JOB04116 FROM C6JES2  ) ENDED AT C2JES2 -
                  JCL ERROR CN(00)
$HASP546 USER04A  SYSTEM OUTPUT RECEIVED AT C6JES2 CN(00)
```

```
SYSLOG

$HASP100 USER04A  ON L9.JR1      NORTHRUP
IEF196I  IRR012I VERIFICATION FAILED. USER PROFILE NOT FOUND.
ICH408I USER(USER04  ) GROUP(        ) NAME(???                 )
  LOGON/JOB INITIATION - USER AT TERMINAL        NOT RACF-DEFINED
 IRR012I VERIFICATION FAILED. USER PROFILE NOT FOUND.
$HASP530 USER04A  ON L9.ST1            11 RECORDS
$HASP534 L9.ST1   INACTIVE
$HASP250 USER04A  IS PURGED
```

```
SYSLOG

$HASP100 USER04A  ON INTRDR      NORTHRUP              FROM TSU04114
USER04
SE ¢12.48.25 JOB04116 $HASP122 USER04A  (JOB04116 FROM C6JES2  )
RECEIVED AT C2JES2¢,LOGON,USER=(USER04)
SE ¢12.48.25 JOB04116 $HASP526 USER04A  TRANSMITTED FOR EXECUTION AT
C2JES2¢,LOGON,USER=(USER04)
SE ¢12.48.25 JOB04116 $HASP165 USER04A  (JOB04116 FROM C6JES2  ) ENDED
AT C2JES2 - JCL ERROR¢,LOGON,USER=(USER04)
$HASP200 C2JES2   STARTED ON LINE9    SESSION A02JES2  BFSZ=0556
$HASP520 USER04A  ON L9.JT1
$HASP534 L9.ST1   INACTIVE
SE ¢12.48.26 JOB04116 $HASP546 USER04A  SYSTEM OUTPUT RECEIVED AT
C6JES2¢,LOGON,USER=(USER04)
$HASP524 L9.JT1   INACTIVE
$HASP250 USER04A  IS PURGED
$HASP540 USER04A  ON L9.SR1            11 RECORDS


JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

JOB04116 ICH408I USER(USER04  ) GROUP(        ) NAME(???        )
           LOGON/JOB INITIATION - USER AT TERMINAL
                                  NOT RACF-DEFINED
JOB04116 IRR012I VERIFICATION FAILED. USER PROFILE NOT FOUND.
------ JES2 JOB STATISTICS ------
//USER04A  JOB (POK,999),NORTHRUP,CLASS=A,MSGCLASS=X,
               NOTIFY=USER04
***ROUTE   XEQ C2JES2
```

Scenario 1 - Results and Comments

- Normal NJE processing occured.
- The job was not permitted to execute because USER04 was undefined at the execution node.
- The output was returned to the submitting node and could be viewed from userid USER04.

Scenario 2 - Submission Utilizing Translation (JOB)

```
Node: C2JES2

RACF Status   - RDEFINE NODES C6JES2.USERJ.USER04 UACC(UPDATE)
                          ADDMEM(USER3)
                Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C6JES2 has been defined as a ¢trusted¢ node.
                Surrogacy is inactive and userid USER04 is
                undefined. The RDEFINE above activates translation
                at this node for jobs submitted by USER04 at
                node C6JES2.
```

```
Node: C6JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C2JES2 has been defined as a ¢trusted¢ node.


Exercise      - USER04 submits a job (USER04B)
                to execute on node C2JES2 without
                the USER=USER3 parameter coded
                on the jobcard.
Expected      - Job to execute at node C2JES2 as
Results         if submitted by USER3. Results to
                be returned to C6JES2.

<===            USER04B is submitted

Terminal Response Area

$HASP122 USER04B  (JOB04119 FROM C6JES2  ) RECEIVED AT
                  C2JES2 CN(00)
$HASP526 USER04B  TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 USER04B  (JOB04119 FROM C6JES2  ) ENDED AT
                  C2JES2 CN(00)
$HASP546 USER04B  SYSTEM OUTPUT RECEIVED AT C6JES2 CN(00)
```

```
SYSLOG

$HASP100 USER04B  ON L9.JR1      NORTHRUP
ICH70001I USER3   LAST ACCESS AT 20:33:41 ON THURSDAY,
                  FEBRUARY 22,1990
$HASP373 USER04B  STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 USER04B  ENDED
$HASP309    INIT  1 INACTIVE ******** C=A
$HASP530 USER04B  ON L9.ST1            28 RECORDS
$HASP534 L9.ST1   INACTIVE
$HASP250 USER04B  IS PURGED
```

```
SYSLOG

$HASP100 USER04B  ON INTRDR      NORTHRUP           FROM TSU04114
USER04
SE ¢13.08.22 JOB04119 $HASP122 USER04B  (JOB04119 FROM C6JES2  )
RECEIVED AT C2JES2¢,LOGON,USER=(USER04)
SE ¢13.08.22 JOB04119 $HASP526 USER04B  TRANSMITTED FOR EXECUTION
AT C2JES2¢,LOGON,USER=(USER04)
$HASP520 USER04B  ON L9.JT1
SE ¢13.08.24 JOB04119 $HASP165 USER04B  (JOB04119 FROM C6JES2  )
ENDED AT C2JES2¢,LOGON,USER=(USER04)
$HASP524 L9.JT1   INACTIVE
$HASP250 USER04B  IS PURGED
SE ¢13.08.26 JOB04119 $HASP546 USER04B  SYSTEM OUTPUT RECEIVED AT
C6JES2¢,LOGON,USER=(USER04)
$HASP540 USER04B  ON L9.SR1            28 RECORDS


JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

ICH70001I USER3   LAST ACCESS AT 20:33:41 ON THURSDAY,
                  FEBRUARY 22, 1990
$HASP373 USER04B  STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 USER04B  ENDED
------ JES2 JOB STATISTICS ------
//USER04B  JOB (POK,999),NORTHRUP,CLASS=A,MSGCLASS=X,
               NOTIFY=USER04
***ROUTE   XEQ C2JES2
```

Scenario 2 - Results and Comments

- Normal NJE processing occured.
- RACF issued message ICH70001I stating that USER3's authorization profile was being used by the job.
- The job was permitted to execute and ended normally.
- The output was returned to the submitting node and could be viewed from userid USER04.

Scenario 3 - Submission Utilizing Translation (SYSOUT and SECLABEL)

```
Node: C2JES2

RACF Status   - RDEFINE NODES C6JES2.SECLS.RACSLUNK UACC(READ)
                          ADDMEM(SYSLOW)
                Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C6JES2 is undefined.
                Surrogacy is inactive and userid P0112RN is
                defined. The RDEFINE above activates translation
                at this node for SYSOUT returning from C6JES2
                with a SECLABEL of RACSLUNK.


Exercise      - P0112RN submits a job (P0112RNX)
                to execute on node C6JES2 without
                user or password information coded
                on the jobcard.
Expected      - Job fails to execute at C6JES2.
Results         Output is returned and the
                SECLABEL translated from RACSLUNK
                to SYSLOW.

                                    P0112RNX is submitted ===>

Terminal Response Area

$HASP122 P0112RNX (JOB05836 FROM C2JES2  ) RECEIVED AT
                  C6JES2 CN(00)
$HASP526 P0112RNX TRANSMITTED FOR EXECUTION AT C6JES2 CN(00)
$HASP165 P0112RNX (JOB05836 FROM C2JES2  ) ENDED AT C6JES2 -
                  JCL ERROR CN(00)
$HASP546 P0112RNX SYSTEM OUTPUT RECEIVED AT C2JES2 CN(00)

SYSLOG

$HASP100 P0112RNX ON INTRDR      NORTHRUP           FROM TSU05832
P0112RN
$HASP520 P0112RNX ON L8.JT1
SE ¢11.24.05 JOB05836 $HASP122 P0112RNX (JOB05836 FROM C2JES2  )
RECEIVED AT C6JES2¢,LOGON,USER=(P0112RN)
SE ¢11.24.05 JOB05836 $HASP526 P0112RNX TRANSMITTED FOR EXECUTION
AT C6JES2¢,LOGON,USER=(P0112RN)
SE ¢11.24.05 JOB05836 $HASP165 P0112RNX (JOB05836 FROM C2JES2  )
ENDED AT C6JES2 - JCL ERROR¢,LOGON,USER=(P0112RN)
$HASP524 L8.JT1   INACTIVE
$HASP250 P0112RNX IS PURGED
SE ¢11.24.07 JOB05836 $HASP546 P0112RNX SYSTEM OUTPUT RECEIVED AT
C2JES2¢,LOGON,USER=(P0112RN)
$HASP540 P0112RNX ON L8.SR1            11 RECORDS
IRR807I PROFILE C6JES2.SECLS.RACSLUNK      IN THE NODES CLASS WAS
        USED TO TRANSLATE SECLABEL RACSLUNK TO SYSLOW
```

```
Node: C6JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C2JES2 is undefined.

SYSLOG

$HASP100 P0112RNX ON L8.JR1      NORTHRUP
ICH408I USER(P0112RN ) GROUP(P0112   ) NAME(RON NORTHRUP        )
  LOGON/JOB INITIATION - INVALID PASSWORD
 IRR013I VERIFICATION FAILED. INVALID PASSWORD GIVEN.
IEF196I  IRR013I VERIFICATION FAILED. INVALID PASSWORD GIVEN.
$HASP530 P0112RNX ON L8.ST1            11 RECORDS
$HASP534 L8.ST1   INACTIVE
$HASP250 P0112RNX IS PURGED

JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C6JES2

ICH408I USER(P0112RN ) GROUP(P0112   ) NAME(RON NORTHRUP        )
  LOGON/JOB INITIATION - INVALID PASSWORD
 IRR013I VERIFICATION FAILED. INVALID PASSWORD GIVEN.
------ JES2 JOB STATISTICS ------
//P0112RNX JOB (POK,999),NORTHRUP,CLASS=A,MSGCLASS=X,
               NOTIFY=USER04
***ROUTE   XEQ C6JES2
```

If the user submits a job from a node that has SECLABELs turned on to be executed at a node that has SECLABELs turned off, the output returned to the submittor will have a SECLABEL of RACSLUNK (RACF SECLABEL unknown) assigned to the SYSOUT. The user will not be able to access this SYSOUT unless the RACSLUNK is defined as a SECLABEL and the user is authorized to it. To bypass

this problem, translation can be used to translate the SECLABEL of RACSLUNK, in this case to SYSLOW, which allows all users to have access to SYSOUT data sets.

Scenario 3 - Results and Comments

- Normal NJE processing occured.
- When the ADDMEM portion of the RDEFINE is omitted, the output is returned with a SECLABEL of RACSLUNK:

```
$LJ5835,ALL
$HASP688 P0112RNX OUTGRP=1.1.1             S=Y P=144 Q=X
$HASP688 D= LOCAL       11 OF 11 RECORDS
$HASP688 B=N F=STD      O=**** T=**** C=**** W=(NONE)    PRMODE=LINE
$HASP688 SECLABEL=RACSLUNK USERID=????????
```

- When the ADDMEM portion of the RDEFINE is included, the output is returned and the SECLABEL translated to SYSLOW.
- The output was returned to the submitting node and could be viewed from userid P0112RN:

```
$LJ5836,ALL
$HASP688 P0112RNX OUTGRP=1.1.1             S=Y P=144 Q=X
$HASP688 D= LOCAL       11 OF 11 RECORDS
$HASP688 B=N F=STD      O=**** T=**** C=**** W=(NONE)    PRMODE=LINE
$HASP688 SECLABEL=SYSLOW   USERID=????????
```

- The JCL error is due to propagation not taking place between untrusted and semi-trusted nodes.

8.9.3 NJE Surrogation Scenarios

Surrogation allows a user to submit a job using another user′s RACF authorization profile. This is achieved by supplying the owner userid on the job card through the ′USER=′ parameter. No password is required, but the submitting nodes must be trusted and both userids have to be defined at the execution node. RACF permits surrogation when the SURROGAT class is active and the correct profile has been defined. Surrogate processing occurs after translation processing if it has been activated.

In the following scenarios, userid P0112RN running at node C6JES2 submits a job to execute with USER3′s RACF authorization at node C2JES2. Remember in this scenario, both P0112RN and USER3 are defined at node C6JES2. The following examples clearly show the RACF definitions for each node and then the progress of a job is traced through the network to the execution node and back again. The messages that would normally be encountered by the submittor and those displayed at the master consoles are also shown. The job′s output, after execution, still belongs to the owner and the surrogate user has to be authorized with the PERMIT command in the JESSPOOL class to access the SYSOUT data.

Scenario 1 - Submission Utilizing Surrogation-SURROGAT Class not Active

```
Node: C2JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C6JES2 has been defined as a ¢trusted¢ node.
                Userids P0112RN and USER3 are both defined.
```

```
Node: C6JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C2JES2 has been defined as a ¢trusted¢ node.
                Userid P0112RN is defined.

Exercise      - P0112RN submits a job (P0112RNA)
                to execute on node C2JES2 with
                the USER=USER3 parameter coded
                on the jobcard.
Expected      - Job to execute at node C2JES2 as
Results         if submitted by USER3. Results to
                be returned to C6JES2.

<===            P0112RNA is submitted

Terminal Response Area

$HASP122 P0112RNA (JOB04066 FROM C6JES2  ) RECEIVED AT
                  C2JES2 CN(00)
$HASP526 P0112RNA TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 P0112RNA (JOB04066 FROM C6JES2  ) ENDED AT C2JES2 -
                  JCL ERROR CN(00)
$HASP546 P0112RNA SYSTEM OUTPUT RECEIVED AT C6JES2 CN(00)
```

```
SYSLOG

$HASP100 P0112RNA ON L9.JR1      NORTHRUP
ICH408I USER(USER3   ) GROUP(P0112   ) NAME(RON               )
  SUBMITTER(P0112RN )
  LOGON/JOB INITIATION - SURROGAT CLASS IS INACTIVE
$HASP530 P0112RNA ON L9.ST1            11 RECORDS
$HASP534 L9.ST1   INACTIVE
$HASP250 P0112RNA IS PURGED
```

```
SYSLOG

$HASP100 P0112RNA ON INTRDR      NORTHRUP           FROM TSU04055
P0112RN
SE ¢20.46.02 JOB04066 $HASP122 P0112RNA (JOB04066 FROM C6JES2  )
RECEIVED AT C2JES2¢,LOGON,USER=(P0112RN)
SE ¢20.46.02 JOB04066 $HASP526 P0112RNA TRANSMITTED FOR EXECUTION
AT C2JES2¢,LOGON,USER=(P0112RN)
SE ¢20.46.02 JOB04066 $HASP165 P0112RNA (JOB04066 FROM C6JES2  )
ENDED AT C2JES2 - JCL ERROR¢,LOGON,USER=(P0112RN)
$HASP520 P0112RNA ON L9.JT1
$HASP524 L9.JT1   INACTIVE
$HASP250 P0112RNA IS PURGED
SE ¢20.46.04 JOB04066 $HASP546 P0112RNA SYSTEM OUTPUT RECEIVED AT
C6JES2¢,LOGON,USER=(P0112RN)
$HASP540 P0112RNA ON L9.SR1            11 RECORDS

JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

JOB04066 ICH408I USER(USER3   ) GROUP(P0112   ) NAME(RON )
           SUBMITTER(P0112RN )
           LOGON/JOB INITIATION - SURROGAT CLASS IS INACTIVE
------ JES2 JOB STATISTICS ------
```

Scenario 1 - Results and Comments

- Normal NJE processing occured.
- The job was not permitted to execute as the SURROGAT class was inactive.
- The output was returned to the submitting node and could be viewed from userid P0112RN.

Scenario 2 - Submission Utilizing Surrogation

```
Node: C2JES2

RACF Status   - SETROPTS CLASSACT(SURROGAT)

                Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C6JES2 has been defined as a ¢trusted¢ node.
                Userids P0112RN and USER3 are both defined.

                Surrogation is active at this node.
```

```
SYSLOG

$HASP100 P0112RNB ON L9.JR1      NORTHRUP
ICH408I USER(USER3   ) GROUP(P0112   ) NAME(RON                )
  SUBMITTER(P0112RN )
  LOGON/JOB INITIATION - SUBMITTER IS NOT AUTHORIZED BY USER
$HASP530 P0112RNB ON L9.ST1            11 RECORDS
$HASP534 L9.ST1   INACTIVE
$HASP250 P0112RNB IS PURGED
```

```
Node: C6JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C6JES2 has been defined as a ¢trusted¢ node.
                Userid P0112RN is defined.


Exercise      - P0112RN submits a job (P0112RNB)
                to execute on node C2JES2 with
                the USER=USER3 parameter coded
                on the jobcard.
Expected      - Job to execute at node C2JES2 as
Results         if submitted by USER3. Results to
                be returned to C6JES2.

<===            P0112RNB is submitted

Terminal Response Area

$HASP122 P0112RNB (JOB04062 FROM C6JES2  ) RECEIVED AT
                  C2JES2 CN(00)
$HASP526 P0112RNB TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 P0112RNB (JOB04062 FROM C6JES2  ) ENDED AT C2JES2 -
                  JCL ERROR CN(00)
$HASP546 P0112RNB SYSTEM OUTPUT RECEIVED AT C6JES2 CN(00)
```

```
SYSLOG

$HASP100 P0112RNB ON INTRDR      NORTHRUP           FROM TSU04055
P0112RN
SE ¢20.05.19 JOB04062 $HASP122 P0112RNB (JOB04062 FROM C6JES2  )
RECEIVED AT C2JES2¢,LOGON,USER=(P0112RN)
SE ¢20.05.19 JOB04062 $HASP526 P0112RNB TRANSMITTED FOR EXECUTION
AT C2JES2¢,LOGON,USER=(P0112RN)
SE ¢20.05.19 JOB04062 $HASP165 P0112RNB (JOB04062 FROM C6JES2  )
ENDED AT C2JES2 - JCL ERROR¢,LOGON,USER=(P0112RN)
$HASP520 P0112RNB ON L9.JT1
$HASP524 L9.JT1   INACTIVE
$HASP250 P0112RNB IS PURGED
SE ¢20.05.22 JOB04062 $HASP546 P0112RNB SYSTEM OUTPUT RECEIVED AT
C6JES2¢,LOGON,USER=(P0112RN)
$HASP540 P0112RNB ON L9.SR1            11 RECORDS


JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

JOB04062 ICH408I USER(USER3   ) GROUP(P0112   ) NAME(RON
          SUBMITTER(P0112RN )
          LOGON/JOB INITIATION - SUBMITTER IS NOT AUTHORIZED
                                 BY USER
------ JES2 JOB STATISTICS ------
```

Scenario 2 - Results and Comments

- Normal NJE processing occured.
- The job was not permitted to execute as the SURROGAT profile for USER3 had not been defined at the execution node.
- The output was returned to the submitting node and could be viewed from userid P0112RN.

Scenario 3 - Submission Utilizing Surrogation

```
Node: C2JES2

RACF Status   - SETROPTS CLASSACT(SURROGAT)
                RDEFINE  SURROGAT USER3.SUBMIT UACC(NONE)
                PERMIT   USER3.SUBMIT CLASS(SURROGAT)
                         ID(P0112RN) ACCESS(READ)

                Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C6JES2 has been defined as a ¢trusted¢ node.
                Userids P0112RN and USER3 are both defined.

                Surrogation active at this node. Surrogat profile
                defined for USER3 and permitted for P0112RN.
```

```
Node: C6JES2

RACF Status   - Node related profiles have been defined and
                the relavent classes activated:
                (NODES, JESINPUT, JESJOBS, WRITER, JESSPOOL)

                C2JES2 has been defined as a ¢trusted¢ node.
                Userid P0112RN is defined.


Exercise      - P0112RN submits a job (P0112RNC)
                to execute on node C2JES2 with
                the USER=USER3 parameter coded
                on the jobcard.
Expected      - Job to execute at node C2JES2 as
Results         if submitted by USER3. Results to
                be returned to C6JES2.

<===            P0112RNC is submitted

Terminal Response Area

$HASP122 P0112RNC (JOB04065 FROM C6JES2  ) RECEIVED AT
                  C2JES2 CN(00)
$HASP526 P0112RNC TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 P0112RNC (JOB04065 FROM C6JES2  ) ENDED AT C2JES2 CN(00)
$HASP546 P0112RNC SYSTEM OUTPUT RECEIVED AT C6JES2 CN(00)
```

```
SYSLOG

$HASP100 P0112RNC ON L9.JR1      NORTHRUP
ICH70001I USER3    LAST ACCESS AT 15:57:58 ON WEDNESDAY,
FEBRUARY 21,1990
$HASP373 P0112RNC STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 P0112RNC ENDED
$HASP309    INIT  1 INACTIVE ******** C=A
$HASP530 P0112RNC ON L9.ST1            28 RECORDS
$HASP534 L9.ST1   INACTIVE
$HASP250 P0112RNC IS PURGED
```

```
SYSLOG

$HASP100 P0112RNC ON INTRDR      NORTHRUP           FROM TSU04055
P0112RN
SE ¢20.33.41 JOB04065 $HASP122 P0112RNC (JOB04065 FROM C6JES2  )
RECEIVED AT C2JES2¢,LOGON,USER=(P0112RN)
SE ¢20.33.41 JOB04065 $HASP526 P0112RNC TRANSMITTED FOR EXECUTION
AT C2JES2¢,LOGON,USER=(P0112RN)
$HASP520 P0112RNC ON L9.JT1
$HASP524 L9.JT1   INACTIVE
$HASP250 P0112RNC IS PURGED
SE ¢20.33.42 JOB04065 $HASP165 P0112RNC (JOB04065 FROM C6JES2  )
ENDED AT C2JES2¢,LOGON,USER=(P0112RN)
$HASP540 P0112RNC ON L9.SR1            28 RECORDS
SE ¢20.33.45 JOB04065 $HASP546 P0112RNC SYSTEM OUTPUT RECEIVED AT
C6JES2¢,LOGON,USER=(P0112RN)


JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

JOB04065  ICH70001I USER3   LAST ACCESS AT 15:57:58 ON WEDNESDAY,
                            FEBRUARY 21, 1990
JOB04065  $HASP373 P0112RNC STARTED - INIT 1 - CLASS A - SYS SMF2
JOB04065  $HASP395 P0112RNC ENDED
------ JES2 JOB STATISTICS ------
//P0112RNC JOB (POK,999),NORTHRUP,CLASS=A,MSGCLASS=X,
              NOTIFY=P0112RN,USER=USER3
***ROUTE   XEQ C2JES2
```

Scenario 3 - Results and Comments

- Normal NJE processing occured.
- RACF issued message ICH70001I stating that USER3's authorization profile was being used by the job.
- The output was returned to the submitting node and could be viewed from userid P0112RN.
- The owner userid is assigned to the SYSOUT output.

8.9.4 NJE Propagation Scenarios

JES sends the userid and SECLABEL from each already validated RACF user who submits jobs through the INTRDR or the TSO SUBMIT command. These jobs are automatically associated with the user and the user's default RACF group. In the NJE environment, JES sends this information between nodes in the NJE job header. This job header is the first record to be transmitted. The job header contains the token information that was previously constructed by the System Authorization Facility in response to the initial RACROUTE TOKENBLD request for that user. In this way, propagation occurs, but only for the userid and SECLABEL fields. Propagation can occur only between trusted nodes.

When the job arrives at the execution node, SAF requests RACF to assign attributes to the job based on the user's RACF profile at that node. Submitting jobs without the userid and password from unknown or semi-trusted nodes results in a JCL error caused by errors when RACF attempts to verify the default userid assigned to the job. Messages are issued indicating that the password is invalid. When the output is returned to the submitting node the user field contains '+++++++++' and the SECLABEL field contains 'RACSLUNK'. Between trusted nodes, the correct userid and SECLABEL information is propagated.

Scenario - Submission Showing Propagation

```
Node: C2JES2

RACF Status   - C6JES2 has been defined as a ¢trusted¢ node.


SYSLOG

$HASP100 P0112RNB ON L9.JR1      NORTHRUP
ICH70001I P0112RN  LAST ACCESS AT 18:32:10 ON WEDNESDAY,
                   FEBRUARY 28,1990
$HASP373 P0112RNB STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 P0112RNB ENDED
$HASP309    INIT  1 INACTIVE ******** C=AB
$HASP530 P0112RNB ON L9.ST1            27 RECORDS
$HASP534 L9.ST1   INACTIVE
$HASP250 P0112RNB IS PURGED
```

```
Node: C6JES2

RACF Status   - C2JES2 has been defined as a ¢trusted¢ node.


Exercise      - P0112RN submits a job (P0112RNB)
                to execute on node C2JES2 without
                USER= or SECLABEL= parameters coded
                on the jobcard.
Expected      - Job to execute at node C2JES2.
Results         Demonstrate the propagation of user
                and seclabel fields across node
                boundries.

<===            P0112RNB is submitted


Terminal Response Area

$HASP122 P0112RNB (JOB04501 FROM C6JES2  ) RECEIVED at
                                           C2JES2 CN(00)
$HASP526 P0112RNB TRANSMITTED FOR EXECUTION AT C2JES2 CN(00)
$HASP165 P0112RNB (JOB04501 FROM C6JES2  ) ENDED AT C2JES2 CN(00)
$HASP546 P0112RNB SYSTEM OUTPUT RECEIVED AT C6JES2 CN(00)


SYSLOG

$HASP100 P0112RNB ON INTRDR      NORTHRUP           FROM TSU04495
P0112RN
$HASP520 P0112RNB ON L9.JT1
SE ¢18.46.00 JOB05022 $HASP122 P0112RNB (JOB04501 FROM C6JES2  )
RECEIVED AT C2JES2¢,LOGON,USER=(P0112RN)
SE ¢18.46.00 JOB04501 $HASP526 P0112RNB TRANSMITTED FOR EXECUTION
at C2JES2¢,LOGON,USER=(P0112RN)
$HASP524 L9.JT1   INACTIVE
$HASP250 P0112RNB IS PURGED
SE ¢18.46.00 JOB05022 $HASP165 P0112RNB (JOB04501 FROM C6JES2  )
ended AT C2JES2¢,LOGON,USER=(P0112RN)
$HASP540 P0112RNB ON L9.SR1            27 RECORDS
SE ¢18.46.04 JOB04501 $HASP546 P0112RNB SYSTEM OUTPUT RECEIVED AT
C6JES2¢,LOGON,USER=(P0112RN)
$HASP540 P0112RN  ON L9.SR1            56 RECORDS
$HASP250 P0112RN  IS PURGED


JES2 JOB LOG

JES2 JOB LOG  --  SYSTEM SMF2  --  NODE C2JES2

ICH70001I P0112RN  LAST ACCESS AT 18:32:10 ON WEDNESDAY,
                   FEBRUARY 28, 1990
$HASP373 P0112RNB STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395 P0112RNB ENDED
------ JES2 JOB STATISTICS ------
//P0112RNB JOB (POK,999),NORTHRUP,CLASS=A,MSGCLASS=X,
//             NOTIFY=USER04
***ROUTE   XEQ C2JES2
```

Scenario - Results and Comments

- User P0112RN is defined in both nodes.
- Normal NJE processing occured.
- The job was permitted to execute and ended normally.
- If the submitting node is unknown or undefined, the job is executed sucessfully but neither userid nor SECLABEL is propagated. This can be checked after the output has been returned:

```
$LJ4500,ALL
$HASP688 P0112RNA OUTGRP=1.1.1             S=Y P=144 Q=W
$HASP688 D= LOCAL       11 OF 11 RECORDS
$HASP688 B=N F=STD      O=**** T=**** C=**** W=(NONE)    PRMODE=LINE
$HASP688 SECLABEL=RACSLUNK USERID=++++++++
```

- When the submitting node is trusted, propagation occurs to the executing node and back again, as follows:

```
$LJ4501,ALL
$HASP688 P0112RNB OUTGRP=1.1.1              S=Y P=144 Q=W
$HASP688 D= LOCAL       27 OF 27 RECORDS
$HASP688 B=N F=STD      O=**** T=**** C=**** W=(NONE)    PRMODE=LINE
$HASP688 SECLABEL=SYSHIGH  USERID=P0112RN
```

8.9.5 Major RACF Checks for NJE Submission (inbound/outbound)

Node: C2JES2

RACF Status - C6JES2 has been defined as a ¢trusted¢ node.

$HASP100 P0112RNB ON L9.JR1

1. Job enters system with propagated fields.
   Submittor (NODE and POE) authorized ?

SAF/RACF checks and logged results

```
RACROUTE REQUEST=VERIFYX,SESSION=NJEBATCH


JOBID=(JES2 90.058 09:14:44),USERDATA=(),OWNER=P0112DP
AUTH=(NORMAL),REASON=(LOGOPTIONS)
USER SECLABEL=SYSHIGH,SESSION=NJE BATCH JOB,JESINPUT=C6JES2,
EXENODE=C2JES2,SUBMITTING USER=P0112RN,SUBMITTING NODE=C6JES2,
SUBMITTING GROUP=P0112
JESINPUT=C6JES2,LEVEL=00,INTENT=READ,ALLOWED=READ
```

2. Perform local SAF/RACF checks. Allow job to execute if ok.

```
ICH70001I P0112RN  LAST ACCESS AT 19:57:11 ON THURSDAY,
                   MARCH 1, 1990
$HASP373  P0112RNB STARTED - INIT  1 - CLASS A - SYS SMF2
$HASP395  P0112RNB ENDED
```

3. Check for transmit authority to C6JES2.

SAF/RACF checks and logged results

```
RACROUTE REQUEST=AUTH,CLASS=WRITER

JOBID=(JES2 90.058 09:14:44),USERDATA=(),OWNER=P0112RN
AUTH=(NORMAL),REASON=(LOGOPTIONS)
LOGSTR=¢TRANSMIT AUTHORIZATION¢
USER SECLABEL=SYSHIGH
WRITER=JES2.NJE.C6JES2,GENPROF=JES2.NJE.**,LEVEL=00,
INTENT=READ,ALLOWED=ALTER,RESOURCE SECLABEL=SYSHIGH
```

Node: C6JES2

RACF Status - C2JES2 has been defined as a ¢trusted¢ node.

Exercise - P0112RN submits a job (P0112RNB) to execute at node C2JES2 without the USER= or SECLABEL= parameters coded on the jobcard. Trace the sequence of events.

<===

$HASP100 P0112RNB ON INTRDR

1. Check for transmit authority. Writer profile defined ?
   User seclabel equal to or less than writer seclabel.

SAF/RACF checks and logged results

```
RACROUTE REQUEST=AUTH,CLASS=WRITER

JOBID=(JES2 90.058 09:12:33),USERDATA=(),OWNER=P0112RO
AUTH=(NORMAL),REASON=(LOGOPTIONS)
LOGSTR=¢TRANSMIT AUTHORIZATION¢
USER SECLABEL=SYSHIGH
WRITER=JES2.NJE.C2JES2,LEVEL=00,INTENT=READ,ALLOWED=READ,
RESOURCE SECLABEL=SYSHIGH
```

2. Transmit to execution node (C2JES2).

```
$HASP520 P0112RNB ON L9.JT1
$HASP524 L9.JT1   INACTIVE
$HASP250 P0112RNB IS PURGED
```

3. Output enters system - authorized NODE and POE ?

```
$HASP540 P0112RNB ON L9.SR1            27 RECORDS
```

SAF/RACF checks and logged results

```
RACROUTE REQUEST=VERIFYX,SESSION=NJEBATCH


JOBID=(JES2 90.058 09:12:33),USERDATA=(),OWNER=P0112RO
AUTH=(NORMAL),REASON=(LOGOPTIONS)
USER SECLABEL=SYSHIGH,SESSION=NJE SYSOUT,JESINPUT=C2JES2,
EXENODE=C2JES2,SUBMITTING USER=P0112RN,SUBMITTING NODE=C6JES2,
SUBMITTING GROUP=P0112
JESINPUT=C2JES2,GENPROF=*,LEVEL=00,INTENT=READ,ALLOWED=ALTER
```

```
4. Transmit output back to submitting node.

$HASP530  P0112RNB ON L9.ST1           27 RECORDS
$HASP534  L9.ST1   INACTIVE
$HASP250  P0112RNB IS PURGED
```

Results and Comments: The userid in the 'OWNER=' field, specifies the owner of the profile and not the owner of the job. For more information on the contents of the NJE header and tokens, see Appendix C, "NJE Job Header and Token DSECTS" on page 269.

8.10 JES2 Spool Offload and JES3 Dump Job

The JES2 Spool Offload and JES3 Dump Job facilities are often used to ease migration from release to release of the JESes. Jobs and data sets reloaded using these facilities can now be verified before entering the system by using RACF 1.9 and defining profiles to control an offload and reload. The NODES and WRITER classes are used to define the profiles.

For jobs, there is data available at reload time that is not available in the NJE case. This is particularly true if the job is from a down-level system, which can happen on a migration to JES 3.1.3.

8.10.1 JES2 Spool Offload

With the additional security functions provided in JES2 3.1.3 and RACF 1.9, there is enhanced protection for jobs and SYSOUTs transmitted to or received from spool offload data sets. Offloaded information should be protected by defining the offload data set to RACF with universal access authority of NONE. If SECLABEL checking is active, the offload data set must be assigned a SECLABEL of SYSHIGH to prevent unauthorized access.

Offload: When jobs are offloaded, there are no security checks. When SYSOUT is offloaded, destination control is available through the WRITER class profiles.

Reload: Security checking at reload time is similar to the NJE job/SYSOUT reception.

- When jobs are reloaded, verification checks that the user is still defined, and that the security classification is still valid. If a return of "not authorized" is received, the job is not reloaded and messages are written to the console log. Audit records are written for these SAF calls. The records identify all jobs reloaded and all attempts to reload jobs that no longer have valid userids or security classifications.

  JESINPUT class control and POE control is also available while reloading jobs.
- When SYSOUT is reloaded and the user does not exist, an "unknown userid" will be assigned. The JESINPUT class control is available.
- If the NODES class is active, do not forget to specify the necessary profiles to allow reception of SYSOUT and jobs.

During the offload and reload scenarios below, the JESJOBS, JESINPUT, NODES, JESSPOOL, and RACFVARS classes are active, unless specified different for a particular case. It is assumed that the necessary profiles in the JESJOBS, and JESSPOOL class are defined, to allow offload jobs and SYSOUT data sets to be received. The option BATCHALLRACF is also active.

8.10.1.1 Spool Offload

While running the scenarios, very general entries are set up for the Offload Transmitter and Receiver. The following entries are defined in the JES2 initialization parameters:

```
OFFLOAD1  DSN=SYS1.JES2.OFFLOAD
OFF1.ST   DISP=KEEP,
          WS=(/),
          DS=ANY
OFF1.SR   DS=ANY,
          WS=(/)
OFF1.JT   DISP=KEEP,
          WS=(/),
          SYS=ANY
OFF1.JR   SYS=ANY,
          WS=(/)
```

The Spool Offload data set has to be catalogued and can reside on DASD or tape/cartridge. In the DATASET class, a profile for this data set has to be established with **UACC(NONE)** to protect classified data on the spool after the offload has been completed. If security label checking is active, assign the offload data set a SECLABEL of SYSHIGH.

During the offload, RACF checks whether the owner of the SYSOUT has access to the offload device. Jobs are not checked, so before starting the offload, define access to the offload resource, OFFn.ST in the WRITER class:

```
RDEFINE  WRITER  JES2.OFF1.ST  UACC(READ)
```

The operator id, or the console id from which the command to start the Offload transmitters is issued, must be defined with UPDATE access to the profile JES2.START.DEV in the OPERCMDS class.

The above access authorities allow the spool offload to complete successfully. Remember, if SECLABEL checking is active, the offload data set needs a SECLABEL of SYSHIGH to receive those jobs and SYSOUT data sets with a SECLABEL of SYSHIGH.

8.10.1.2 Spool Reload from the Same Node

Before the reload, the offloaded jobs and SYSOUTs has to be authorized to re-enter the system. Define the following in the JESINPUT class profiles:

```
RDEFINE  JESINPUT  OFF1.JR  UACC(READ)
RDEFINE  JESINPUT  OFF1.SR  UACC(READ)
```

In the NODES class, the offload node is defined as a trusted node and job reverification is not done. This prevents all jobs from abending with the following message:

```
ICH408I  USER(USER2   ) GROUP(P0112   ) NAME(DON     )
  LOGON/JOB  INITIATION  -  INVALID  PASSWORD  GIVEN
 IRR013I  VERIFICATION  FAILED.  INVALID  PASSWORD  GIVEN.
```

Because the offload node is the same as the reload node, and therefore shares the same RACF database, the node is specified as ″local″ in the RACFVARS class and is then established as a trusted node by the following definitions:

```
RDEFINE  RACFVARS  &RACLNDE  ADDMEM(C2JES2)
RDEFINE  NODES  &RACLNDE.USERS.*  UACC(UPDATE)  ADDMEM(&SUSER)
RDEFINE  NODES  &RACLNDE.USERJ.*  UACC(UPDATE)
```

All jobs and SYSOUTs reload correctly, except that:

- With SECLABEL checking inactive, jobs and SYSOUTs with a blank SECLABEL are received with a SECLABEL of RACSLUNK.
- With SECLABEL checking active, all jobs reload correctly, but SYSOUTs with a blank SECLABEL do not reload.

8.10.1.3 Spool Reload from Down-level Node

The down-level node used in this scenario is running MVS/ESA 3.1.0e with JES2 3.1.1, and RACF 1.8. To receive the offloaded jobs and SYSOUTs, an entry for the down-level node has to be made in the NODES class. The following profile is established to make the down-level node a trusted node in the receiving node:

```
RDEFINE  NODES  C7JES2.USERJ.*  UACC(CONTROL)
RDEFINE  NODES  C7JES2.USERS.*  UACC(CONTROL)  ADDMEM(&SUSER)
```

To receive the offloaded spool data sets, define the following profiles in the JESINPUT class:

```
RDEFINE  JESINPUT  OFF1.JR  UACC(READ)
RDEFINE  JESINPUT  OFF1.SR  UACC(READ)
```

During reloading of the jobs and SYSOUTs, the SECLABEL field is left blank. The reload of those jobs, which have no userid defined in the RACF database on the receiving system, fails with the message:

```
IRR008I  JOB FAILED.  USER PARAMETER REQUIRED ON JOB STATEMENT.
```

If SECLABEL checking is active on the receiving node, the default SECLABEL defined in the RACF user profile is assigned to the SECLABEL field of the job.

Those SYSOUTs that have no valid userid on the receiving system are reloaded, but the userid field contains ???????? , which is the unknown NJE userid.

8.10.1.4 Spool Reload to a Different Node

The nodes used in this scenario are at the same software levels, and share the same RACF database. Before starting the reload, define the following profile in NODES, RACFVARS, and JESINPUT class:

```
RALTER  RACFVARS  &RACLNDE  ADDMEM(C2JEST)
RDEFINE  NODES  &RACLNDE.USERS.*  UACC(UPDATE)  ADDMEM(&SUSER)
RDEFINE  NODES  &RACLNDE.USERJ.*  UACC(UPDATE)

RDEFINE  JESINPUT  OFF1.JR  UACC(READ)
RDEFINE  JESINPUT  OFF1.SR  UACC(READ)
```

Upon receiving the jobs, it may be necessary to change the execution node to the local node, unless the jobs have to execute specifically on the original node. This can be achieved using the parameter MOD=(ROUTECDE=LOCAL). To hold the jobs for execution later, specify MOD=(HOLD=YES). The following example gives the parameters that can be specified for the job receiver and the SYSOUT receiver:

```
OFF1.JR   SYS=ANY,
          MOD=(ROUTECDE=LOCAL,HOLD=YES),
          WS=(CL/)
OFF1.SR   DS=ANY,
          WS=(/)
```

During the job reload, the following messages are generated on the console log:

```
13.50.20.73 JOB00190  $HASP100 USER1C01 ON OFF1.JR      IEBCOPY
13.50.20.93 JOB04160  $HASP100 USER2C02 ON OFF1.JR      BR14
13.50.21.13 JOB04161  $HASP100 USER1C01 ON ROUT.JR1     IEBCOPY
13.50.21.18 JOB04163  $HASP100 P0112WK3 ON OFF1.JR      ASMLNK1A
13.50.20.84 JOB00190  $HASP520 USER1C01 ON ROUT.JT1
13.50.21.43 JOB04160  $HASP520 USER2C02 ON ROUT.JT1
13.50.21.45 JOB04161  $HASP101 USER1C01 HELD
13.50.22.79 JOB04165  $HASP100 USER2C02 ON ROUT.JR1     BR14
13.50.22.70 JOB00190  $HASP250 USER1C01 IS PURGED
13.50.23.12 JOB04165  $HASP101 USER2C02 HELD
13.50.23.28 JOB04160  $HASP250 USER2C02 IS PURGED
13.50.23.71 JOB04163  $HASP520 P0112WK3 ON ROUT.JT1
13.50.24.99 JOB04170  $HASP100 P0112WK3 ON ROUT.JR1     ASMLNK1A
13.50.25.29 JOB04170  $HASP101 P0112WK3 HELD
13.50.25.46 JOB04163  $HASP250 P0112WK3 IS PURGED
13.50.43.02           IEF196I IEF287I   SYS1.JES2.OFFLOAD
13.50.43.03           IEF196I RECTLGD 2
13.50.43.03           IEF196I IEF287I   VOL SER NOS= JS2TS1.
13.50.42.99           $HASP097 OFF1.JR  IS DRAINED
13.50.43.04           $HASP097 OFFLOAD1 IS DRAINED
```

The security checks for SYSOUT data sets during reload are the same as for NJE SYSOUT reception. As the originating node is defined as trusted in the NODES class, all SYSOUT reloads correctly, except for the following cases:

- With SECLABEL checking inactive, those SYSOUTs with a blank SECLABEL field during the offload are received with RACSLUNK as a SECLABEL.
- With SECLABEL checking active, those SYSOUTs with a blank SECLABEL field during offload are not received.

8.10.1.5 Classes and Profiles used

The following profiles are needed during Spool Offload and Reload. If the associated class is not active in the system, ignore the entries:

Offload -

```
WRITER      JESx.OFFn.ST            READ
OPERCMDS    JESx.START.DEV          UPDATE
```

Reload to the same node -

```
JESINPUT    OFFn.JR                 READ
JESINPUT    OFFn.SR                 READ
RACFVARS    &RACLNDE                ADDMEM(nodeid)
NODES       &RACLNDE.USERJ.*        UPDATE
NODES       &RACLNDE.USERS.*        UPDATE  ADDMEM(&SUSER)
```

Reload from a down-level node -

```
JESINPUT    OFFn.JR                 READ
JESINPUT    OFFn.SR                 READ
NODES       nodeid.USERJ.*          CONTROL
NODES       nodeid.USERS.*          CONTROL  ADDMEM(&SUSER)
```

Reload from a different node -

```
JESINPUT    OFFn.JR                 READ
JESINPUT    OFFn.SR                 READ
NODES       nodeid.USERJ.*          UPDATE
NODES       nodeid.USERS.*          UPDATE  ADDMEM(&SUSER)
```

If the offload node shares the RACF database with the reload node, the following profile definitions can be used when reloading from a different node:

```
JESINPUT    OFFn.JR                 READ
JESINPUT    OFFn.SR                 READ
RACFVARS    &RACLNDE                ADDMEM(nodeid)
NODES       &RACLNDE.USERS.*        UPDATE  ADDMEM(&SUSER)
NODES       &RACLNDE.USERJ.*        UPDATE

JESINPUT    ROUT.JT1                READ
```

8.10.2 JES3 Dump Job Processing

The function of the JES2 Spool Offload and the JES3 Dump Job are identical, but the implementation is very different. In JES3, control blocks are dumped rather than NJE headers. The information necessary for validation must be obtained from fields within control blocks rather than from NJE headers.

A JES3 dump job, type=in, is handled in the same manner as NJE validation. The job structure goes through job validation (NJEBATCH calls) if the job requires execution processing. Otherwise, it goes through SYSOUT processing. All fields needed for the RACF calls are available in the in-storage JES3 control blocks. NJE headers are not used in dump job reload processing.

Chapter 9. RJE/RJP Security Control

RJE and RJP workstation signon/logon security controls are managed by passwords in the signon card or LOGON command (Figure 44). In previous releases of JES2 and JES3, passwords for signon and logon were maintained by JES2 and JES3. Now signon/logon from RJE or RJP is verified using SAF and RACF.

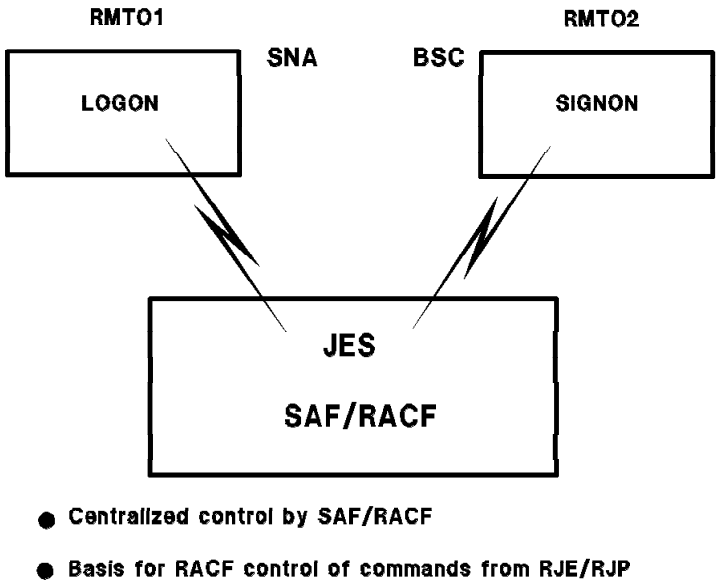

Figure 44. RJE/RJP Signon and Logon Security

9.1 RJE/RJP Signon

The RJE/RJP signon/logon security control using SAF/RACF also provides the basis for RACF checking of commands from RJE/RJP terminals. This change is applied to both SNA RJE/RJP and BSC RJE/RJP.

A remote terminal name must be defined as the profile name in the FACILITY class. The existence of a profile in the FACILITY class for a remote workstation forces the user to enter a password to be checked by RACF rather than JES:

```
RDEFINE   FACILITY   RJE.RMT01
RDEFINE   FACILITY   RJE.RMT*
RDEFINE   FACILITY   RJE.WS001
```

The second example, **RJE.RMT*** shows a generic definition for many remote terminals.

The remote terminal name must be defined as a valid RACF userid and with a password. This password must be changed the first time the remote terminal issues a signon or logon.

```
ADDUSER   RMT01   PASSWORD(xxxxxxxx)
```

The FACILITY class must be active:

```
SETROPTS   CLASSACT(FACILITY)
```

9.2 RJE Signon (JES2)

In the previous release, passwords for signon/logon were checked by JES2. These passwords were specified in JES2 initialization parameters. With JES2 Release 3.1.3 and RACF 1.9, signon processing is changed to use the new security checking:

- The password is checked by SAF/RACF. RACF control for password checking is available only if all the following conditions are met:
  - RACF Release 1.9 is active.
  - FACILITY resource class is active.
  - " RJE.rmt-name " profile is defined in FACILITY class.
  - RJE remote name defined as a user to RACF through ADDUSER command.

  Signon/logon checking uses RACROUTE REQUEST=VERIFYX. The userid that represents the remote name, password, and new password are passed to SAF/RACF. After this check, the remote name is recognized as a userid by RACF. Security checking for commands from remote terminals are also checked by RACF. Instead of the remote terminal password, you can use the line password for SIGNON/LOGON commands. The line passwords are still checked by JES2.
- JES2 checks the password if no decision is made by RACF, or if the FACILITY class profile is not found. This password check is made just as in previous releases.
- The format of the LOGON and SIGNON commands is changed for RACF control, as follows:
  - When passwords are to be checked by RACF - A remote password is always required, and a new password is optional; for example:

    ```
    BSC  /*SIGNON RMTnn     linepass      newpass             pass
    SNA  LOGON APPLID(JES2) LOGMODE(name) DATA(RMTnn,linepass,pass,newpass)
    ```
  - When passwords are to be checked by JES2 - If RACF makes no decision because of the reasons listed above, you have to use a different format. For BSC /*SIGNON, columns 35-45 should be blank. For SNA LOGON, the fourth parameter in the DATA field should be omitted; for example:

    ```
    BSC  /*SIGNON RMTnn     linepass                        pass
    SNA  LOGON APPLID(JES2) LOGMODE(name) DATA(RMTnn,linepass,pass)
    ```
- When dedicated BSC lines are started, RJE userids are checked, but no passwords are used. When sessions are started by a $SRMTn command or autologon, RJE useids are checked at JES2 initiated logon time, but no passwords are used.
- Exit17 (BSC signon/off), and Exit18 (SNA logon/off) can be used to perform additional security checks or selectively bypass security checks.

9.3 RJP Signon (JES3)

The protection of RJP workstations was based on JES3 verification of the workstation passwords. JES3 initialization parameters or operator commands were used to update the passwords.

The protection of RJP workstations can be controlled by RACF. To activate the RACF protection, you must define the RJP workstation to the FACILITY class and also define a RACF userid with the workstation name. Additionally, if you want to control remote printers or punches, you must define them to the WRITER class. A remote reader can be controlled by defining the workstation name to the JESINPUT class. Also, operator commands that are allowed from the workstation can be controlled.

If the workstation is not defined in the RACF FACILITY class, the existing JES3 verification is used.

Security enhancements cover both BSC and SNA RJP workstations. The format of the workstation LOGON or SIGNON command is different if RACF protection is defined.

9.3.1 SNARJP Logon

If the workstation is using formatted system services, the format of the LOGON command is workstation dependent. However, the data field must be as defined.

```
LOGON APPLID(applname)  DATA(wsname,old-password,new-password)
```

9.3.2 BSCRJP Signon

The SIGNON card is still column dependent; *old-password* starts in column 35 and *new-password* in 44. The *new-password* is optional:

```
/*SIGNON  wsname  A R  line-password old-password new-password
```

Chapter 10. Console and Command Security

If you desire to have operator consoles and the commands that are entered on them as part of your security environment, the following options should be considered:

- Operators can be required to log on to consoles. Use of the operator logon increases the security of MCS consoles, as it restricts the use of MCS consoles only to those users defined, identified, and authenticated to RACF. A new RACF resource class, **CONSOLE**, is used to control the use of the consoles.
- Operators can be permitted or restricted to certain commands based on their userid.
- Operator commands can be authorized by RACF through the new **OPERCMDS** resource class. In this chapter, operator commands are discussed separately in the following sections:
  - MVS console commands
  - JES3 console commands
  - JES2 console commands
- Certain operator commands can be authorized based on their input source. The input source may be a terminal, an MCS console, or a JES input device.
- Auditing of some or all commands entered can be specified. SMF type 80 records are created. SMF audit records include time and date, issuer identification, command image, origin of command, and success/failure indicator.

The control of the use of operator commands is based on the identity of the operator issuing the command. As shown in Figure 45, operator commands can be issued from the following sources and command authorization can be provided by SAF and RACF:

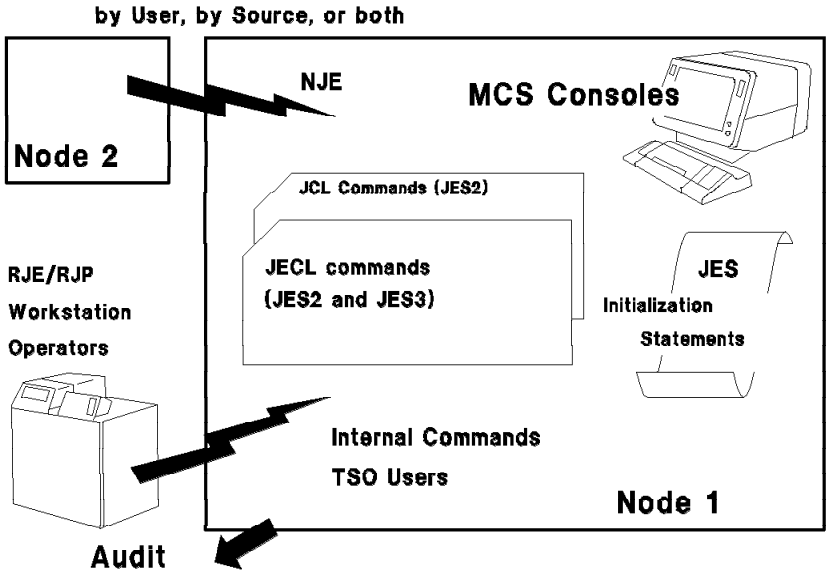

Figure 45. MVS/ESA Operator Commands

The command sources are:

- MCS consoles.
- System commands within JCL (JES2).
- JES2 and JES3 commands with JES JCL (JECL).
- RJE and RJP workstation operators.
- Commands from other NJE nodes.
- JES2 initialization statements.
- Commands issued internally within the system using SVC 34 (MGCR), for example, NetView.

An installation can also control commands based on where the command was issued. Thus, commands can be controlled, for example, by a userid:

- Using a specified console
- Using a specified terminal
- Using a specified JESINPUT device

10.1 Grouping of Operator Functions

For organizational and administrative ease, functional groups should be created for all types of operative functions. In Figure 46, a number of common operator functions are shown and the types of commands required by each function.

The following list describes the functions:

Group Name	Operator Function
MSTCONS	System control operator. Controls JES and MVS. Should be allowed to issue all JES and MVS commands with the exception of commands reserved for system programmer use.
SETUP	Tape setup operator. Mounts tapes according to JES3 or MVS mount messages.
OUTSERV	Output service operator. Controls local and remote JES and FSS driven printers. Should be allowed to issue JES inquiry commands, to start, restart, and cancel writers, and handle printer FSSes.
HLPDESK	User help desk service. Is the focal point for all end user requests for console service functions related to user jobs, TSO sessions, and so on. Should be allowed to issue display commands.
PRODCTL	Production control personnel. Has the responsibility for the installation's production jobs.
NETWORK	VTAM network control operator. Controls the VTAM network. Should be allowed to issue display commands.
SYSPROG	System programmer. Has the responsibility for JES and MVS. Should be allowed to issue all JES and MVS commands.
DEFOPER	Default users for MCS consoles. The userids connected to this group are automatically logged on after IPL. Should be allowed to issue JES inquiry and MVS display commands.

OPER Group for general operator functions. Should be allowed to issue JES inquiry and MVS display commands.

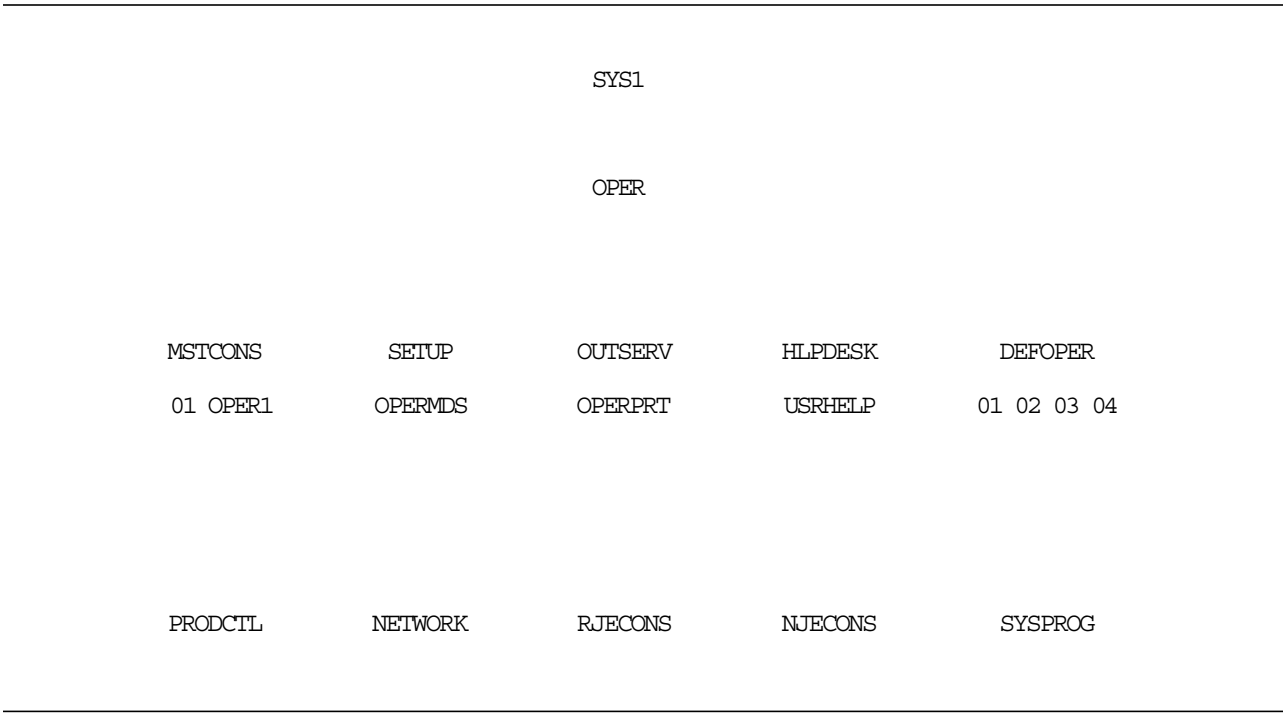

Figure 46. Console Operator RACF Grouping

The following example shows how to define these groups:

```
ADDGROUP   MSTCONS   OWNER(OPER)   SUPGROUP(OPER)
ADDGROUP   SETUP     OWNER(OPER)   SUPGROUP(OPER)
ADDGROUP   OUTSERV   OWNER(OPER)   SUPGROUP(OPER)
ADDGROUP   HLPDESK   OWNER(OPER)   SUPGROUP(OPER)
ADDGROUP   PRODCTL   OWNER(OPER)   SUPGROUP(OPER)
ADDGROUP   NETWORK   OWNER(OPER)   SUPGROUP(OPER)
ADDGROUP   SYSPROG   OWNER(OPER)   SUPGROUP(OPER)
```

The individual operators receive the required authorization by connecting their userids to the appropriate functional groups.

10.1.1 Connecting Users to Groups

Use the CONNECT command to place operator userids into a group, as follows:

```
CONNECT  (01 OPER1)   GROUP(MSTCONS)  OWNER(OPER)
```

10.2 Console Security Definitions

Create a profile for each console to be protected.

```
RDEFINE   CONSOLE   01   UACC(NONE)
RDEFINE   CONSOLE   02   UACC(NONE)
RDEFINE   CONSOLE   03   UACC(NONE)
RDEFINE   CONSOLE   04   UACC(READ)
```

Where:

01-04 The console number corresponding to the MCS console id. An '*' in this field defines all consoles.

NONE Use PERMIT commands to allow users or groups of users to use the consoles defined with UACC(NONE).

READ Any userid can log on to this console.

When the consoles are defined, permit console use as follows:

- User 02 is to be authorized to use console CN(02). READ access is the minimum access required for authorization.

```
PERMIT   02   CLASS(CONSOLE)   ID(02)   ACCESS(READ)
```

- As shown in Figure 46, using group MSTCONS authorizes users 01 and OPER1 to use console CN(01) as defined by the following PERMIT command:

```
PERMIT   01   CLASS(CONSOLE)   ID(MSTCONS)   ACCESS(READ)
```

10.2.1 Defining Default Console Userids

For each MCS console, a default userid must be created in the RACF database. MCS consoles are numbered in the order of the CONSOLE statements in member CONSOLxx in SYS1.PARMLIB, starting with **01**. These console ids must be defined to RACF in **ADDUSER** as valid userids with passwords.

The following examples show how to define default userids for consoles:

```
ADDGROUP   OPER   OWNER(SYS1)   SUPGROUP(SYS1)
ADDGROUP   DEFOPER   OWNER(OPER)   SUPGROUP(OPER)
```

The example shows four MCS consoles defined in SYS1.PARMLIB.

```
ADDUSER (01 02 03 04 ) NAME(¢DEFAULT CONSOLE USER¢)   DFLTGRP(DEFOPER)
```

Note: With LOGON(AUTO), it is necessary to define 01 to 04 as valid userids to ensure that a proper logon occurs after console initialization. If these userids are not defined in the RACF database, at NIP time, when MCS tries to logon using the default userid, the following error messages appear:

```
ICH408I USER(02      ) GROUP(         ) NAME(DEFAULT CONSOLE USER)
  LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
 IRR012I VERIFICATION FAILED. USER PROFILE NOT FOUND.
LOGON 02
IEE186I LOGON    COMMAND NOT PROCESSED. SECURITY INTERFACE DORMANT.
```

Since the security interface is dormant, MCS uses the MCS Authority instead.

The following examples show operator userid definitions:

```
ADDUSER   (OPER1)   NAME(¢JES OPERATOR¢)
ADDUSER   (OPERMDS)   NAME(¢JES TAPE OPERATOR¢)   DFLTGRP(SETUP)
ADDUSER   (OPERPRT)   NAME(¢JES PRINT OPERATOR¢)   DFLTGRP(OUTSERV)
ADDUSER   (USRHELP)   NAME(¢USER HELP DESK¢)   DFLTGRP(HLPDESK)
```

All default userids are members of group DEFOPER, which appear in the access lists of appropriate RACF profiles. Group OPER is the superior group for all operative groups. Figure 46 shows an operator environment as a sample use of userids and groupids for defining a security environment for JES3/MCS consoles.

10.2.2 Activating Consoles

To use RACF for protection of the use of the consoles, the CONSOLE class must be activated. Also, for better performance when access authority checks are made, the class should be activated for RACLIST processing:

```
SETROPTS  CLASSACT(CONSOLE)  RACLIST(CONSOLE)
```

10.2.3 Console LOGON Options

Consoles can be protected by RACF profiles that specify which operators or group of operators attempting to log on have the authority to do so. This is achieved by identifying your consoles to RACF, and indicating that before any console activity is allowed, the operator must log on to the console.

Figure 47 shows the log-on area on the console screen. The concept is based on the fact that an operator is responsible for the console.

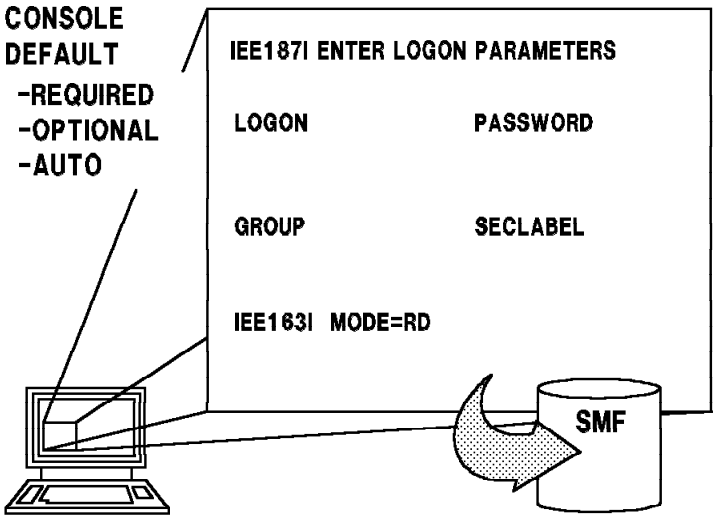

Figure 47. Operator Console Logon

The log-on requirement is optional. Another option is to have consoles automatically logged on. In order to log on to a console, the operator must key in a userid and password in the fields provided in the log-on prompt area. A password may be changed when the operator logs on by specifying the password in an old-password/new-password/new-password format. The password is not displayed. The user can be limited to signing on at specific consoles. Other parameters can be supplied at logon, such as the current connect group. A RACF group and SECLABEL can optionally be specified. Unsuccessful attempts to log on could cause RACF to revoke the userid. The log-on prompt shown in Figure 47 is written to the screen any time that:

- A console is activated during system initialization.
- The status of the console is switched from status-display or message-stream to full-capability.
- The console is varied online by the VARY command.

- A LOGOFF command is issued from the MCS console.
- The LOGON command is issued.

The use of the log-on feature is an installation option, which is set up by the security administrator by a change to the SYS1.PARMLIB definition of the console. There is no control over Write-to-Operator (WTO) traffic. You may choose whether to use the operator logon function by specifying the appropriate option in the DEFAULT section of the CONSOLxx member in SYS1.PARMLIB. The following options are possible:

LOGON(OPTIONAL) Indicates that a user can use the console without logging on. This is the default, and may be useful when training operators. The operators can optionally log on but commands can be issued without a logon. If the operator logs on, and the CONSOLE and OPERCMDS resource classes are active, the operator can access only consoles and resources to which that operator is authorized. If the operator is not logged on, commands may be issued as in pre-MVS Release 3.1.3 systems.

LOGON(AUTO) Indicates that each MCS console is automatically logged on using the 2-character EBCDIC representation of the console id.

LOGON(REQUIRED) Indicates that the operator must log on to the MCS console before being able to issue commands. RACF must be fully initialized. Failure to log on to any console defined in this way means all commands issued are rejected.

During initialization only the master console can issue commands. All commands from the secondary consoles are rejected. However, after the security product (RACF) is fully initialized, all operators must log on.

10.2.4 Console Log-on Processing

The LOGON command is not echoed to the screen, nor is it retrievable with the PA1 key. The LOGON command is hardcopied like a normal command except that the password field is suppressed. No RACROUTE call is performed to authorize the LOGON command.

Regardless of which log-on option is used, only one operator can be logged on to a console at any one time. This means that if you try to log on and another operator is already logged on, the other operator is logged off automatically before your logon is processed. In addition, a single unique userid may be in use only on a single console at any time, and userid reuse is not allowed.

When an operator is logged off an MCS console due to a console switch, the active logon of the failing console is not propagated to its alternate. This means that an operator must log on or be logged on to the alternate console to successfully issue commands from it.

When no master console or its alternate is available, the VARY MSTCONS command is accepted from any secondary console, regardless of authority.

Use of the operator logon with the appropriate RACF command resource profiles increases the granularity of command authorization for MVS operator commands to a command and operator basis. This provides a better degree of control over MVS operator commands than was previously possible. Figure 48 shows an overview of log-on processing.

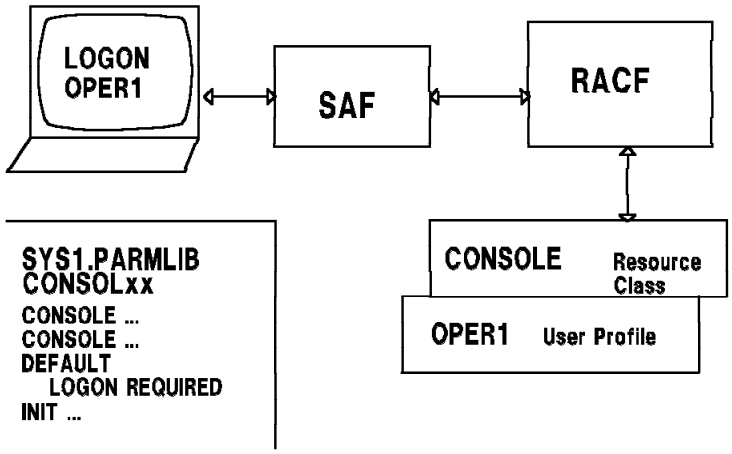

Figure 48. Operator Logon Overview

The difference between the **OPERCMDS** and **CONSOLE** resource classes is that CONSOLE checks whether a user can log on to a console or not, and OPERCMDS determines what commands this user is allowed to issue from the console. For a secure environment, these two classes should be used together.

10.2.5 LOGON Processing at Initialization

Depending on the log-on option you choose, the consoles are initialized as follows:

LOGON(AUTO) At NIP time, MCS issues a LOGON on behalf of the console but the logon is actually done after the security product (RACF) is active. The userid logged on to the console is the same as its console number, for example:

- CN(01) is logged on as 01
- CN(02) is logged on as 02
- CN(03) is logged on as 03
- CN(04) is logged on as 04

LOGON(REQUIRED) At the very start of NIP, right after the nucleus is built, the following message appears:

```
IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE 03.8 ,VER=SP3.1.3
```

At this point, since nothing is active, the system is exposed.

Note: Assuming this option is chosen, high security is required. However, if the operator can override the CONSOL parameter at this point, security can be bypassed. It is advisable that the usage of alternative CONSOLxx parameters be prevented by specifying CON=(xx,OPI=NO) in all IEASYSxx members in SYS1.PARMLIB. After responding to this message, logon proceeds until you get the message:

```
ICH520I RACF 1.9.0 IS ACTIVE
ICH531I RACF DATA SET ALLOCATION/DEALLOCATION INTERFACE IS ACTIVE
```

Now all operators on the master and secondary consoles are required to enter their userids and passwords. If they do not log on and press the PA2 key, the data entry lines disappear and they are not able to enter any commands except LOGON. See 10.2.6, “Logon Auditing and Logging” on page 171.

The new LOGON command allows the operator to identify himself to the system through a userid and a password. When an operator enters the command:

```
logon oper1
```

A logon data entry screen appears at the bottom of the console as shown in Figure 47 on page 167 and displays:

```
IEE1871 ENTER LOGON PARAMETERS                    ENTER    CANCEL
LOGON oper1    PASSWORD
GROUP          SECLABEL
IEE163I MODE=RD
```

If the authorization is successful, the operator may now proceed.

Display the console configuration using the command:

```
d consoles,l
```

Userids are new in the display on the console and are shown by the ----> . This display shows userids 01 and OPER1 as the logged on userids for the two consoles shown:

```
IEE249I 13.56.30 CONSOLE DISPLAY 585
MSG  CURR=1    LIM=1500  RPLY  CURR=0   LIM=20     SUBSYS     PFK=00
 CONSOLE/ALT      ID   --------------- SPECIFICATIONS ----------------
 SYSLOG                COND=H       AUTH=CMDS                NBUF=0
                       ROUTCDE=ALL
      8E0/8E2      01   COND=M       AUTH=ALL                 NBUF=0
---->01                AREA=Z       MFORM=T,J
                       DEL=RD    RTME=1     RNUM=5     SEG=28    CON=N
                       USE=FC    LEVEL=ALL             PFKTAB=MCONPFK0
                       ROUTCDE=ALL
      8E2/8FF      02   COND=A       AUTH=INFO                NBUF=0
---->OPER1             AREA=Z,A     MFORM=T
                       DEL=RD    RTME=1     RNUM=5     SEG=28    CON=N
                       USE=FC    LEVEL=ALL             PFKTAB=SCONPFK0
                       ROUTCDE=ALL
```

Also, the logged userid is shown on the console screen as shown below:

```
IEA152I ENTER  CANCEL D C,K USERID=OPER1      <--------

-
IEA192I MODE=RD
```

In this example of the console screen, the logged on userid is OPER1. The ′ - ′ shown on the next line is the cursor for operator input.

10.2.6 Logon Auditing and Logging

Operator LOGON and LOGOFF commands are written to SYSLOG and are audited. A report can be generated by the RACF Report Writer from the SMF type 80 audit records. A sample of the commands entered is shown in Figure 49 and in a report produced by the Report Writer is shown in Appendix D, “Sample RACF Report Writer Listing” on page 275. In the report:

- Successful logons are shown as resource access records.
- Invalid logons appear as event 1 - TSO logon or logoff.
- Commands entered in an unsecured terminal with LOGON(REQUIRED) are not logged.

```
 1  1748220  SY1=          LOGON OPER1    PASSWORD
    1748220  SY1=          IEE185I LOGON  OPER1    COMPLETE FOR DEVNUM 8E2  CN 02.

    1748556  SY1=          IEE185I LOGOFF OPER1    COMPLETE FOR DEVNUM 8E2  CN 02.
 2  1748556  SY1=          LOGON 02       PASSWORD
    1748556  SY1=          IEE185I LOGON  02       COMPLETE FOR DEVNUM 8E2  CN 02.

 3  1749067 -LOGOFF
    1749067  SY1=          IEE185I LOGOFF 02       COMPLETE FOR DEVNUM 8E2  CN 02.

 4  1749108 -D A
    1749108  SY1=          IEE186I D        COMMAND NOT PROCESSED.   LOGON REQUIRED.

 5  1749206  SY1=          ICH408I USER(02      ) GROUP(ALL     ) NAME(CONSOLE 8E2         )
    1749206  SY1=            LOGON/JOB INITIATION - INVALID PASSWORD
    1749206  SY1=           IRR013I VERIFICATION FAILED. INVALID PASSWORD GIVEN.
    1749206  SY1=          LOGON 02       PASSWORD

 6  1749351  SY1=          LOGON  LASJD   PASSWORD
    1749400  SY1=          ICH408I USER( LASJD  ) GROUP(        ) NAME(                    )
    1749400  SY1=            LOGON/JOB INITIATION - NOT AUTHORIZED TO TERMINAL/CONSOLE
```

Notes:

1. Logged on to the system as OPER1.

2. Logged on operator id 02 without logging off OPER1. This resulted in MCS logging off OPER1 as seen in the line above.

3. Issued LOGOFF command for operator 02.

4. Issued commands without any logged on operator.

5. Logged on as operator 02 using a wrong password.

6. Logged on using an invalid userid and password. MCS does not respond with an invalid userid error.

Figure 49. Sample JES3 SYSLOG of Console LOGON Processing

10.2.7 Console Logon Considerations

For the log-on option for consoles (in the DEFAULT section of the CONSOLxx member in SYS1.PARMLIB), we recommend one of the following alternatives:

LOGON(AUTO) This causes a default userid (the console number) to be automatically logged on at every console. These default userids should be given the authority for display commands only. For commands requiring higher authority, the operator has to re-logon with a valid userid. See 10.2.1, “Defining Default Console Userids” on page 166 for sample definitions.

LOGON(REQUIRED) This makes any console unusable after RACF initialization unless an operator logs on with a valid userid.

While the first option does not require an operator to log on at a console if only display level commands are to be entered, the second option provides the possibility for complete operator accountability for all commands entered at the price of requiring explicit logon at every console. Unless stringent auditing requirements are to be observed, we feel the LOGON(REQUIRED) option to be most appropriate.

Note: LOGON is a system-wide option. Therefore, it is not possible to require a logon for some consoles, autolog others, and make the logon optional for the rest.

The following consoles are **not** supported for audit or control:

- JES3-managed consoles
- Operator interactions with the Disabled Console Communications Facility (DCCF)
- Hardware consoles (consoles attached to the service processor).

All MCS consoles (with the exception of the Master Console) should be defined with AUTH(INFO), the lowest possible MCS console authorization level when you allow RACF to control command processing.

10.2.8 Console Logoff Processing

An operator is logged off an MCS console when:

- The LOGOFF command is entered.
- A LOGON request is issued from a console that already has an operator logged on.
- The status of the console is changed from full capability to message stream or status display.
- The console is varied OFFLINE or ONLINE.
- The console fails, resulting in a console switch.

If you have specified LOGON(AUTO), the LOGOFF command causes the operator to be logged off and a RACINIT to be issued to log on the console automatically.

Note: If the default operator 02 is now logged on and another user logs on, such as OPER1, 02 is automatically logged off.

The following messages are written into the SYSLOG:

```
IEE185I LOGOFF 02       COMPLETE FOR DEVNUM 8E2  CN 02.
LOGON OPER1    PASSWORD
IEE185I LOGON  OPER1    COMPLETE FOR DEVNUM 8E2  CN 02.
```

On a system IPLed with LOGON(REQUIRED), the LOGOFF command can be used to cancel the logon of an operator and enable the console to be left in a secure, unattended state. In this state, no commands are accepted from the console until an operator again logs on to that console. While no operator is logged on to a given console, messages can continue to flow to that device. When LOGON(REQUIRED) is specified, LOGOFF command processing concludes by placing the console in roll mode. This ensures that messages do not back up on the logged off console, causing a potential buffer shortage.

The LOGOFF command has INFO authority and executes in the Communication Task address space. Like other commands, it is echoed to the console; however, it is not retrievable with the PA1 key.

A RACROUTE REQUEST=VERIFY,ENVIR=DELETE call is made as a part of LOGOFF command processing to delete the ACEE established by the previous logon for a given console. The RACROUTE interface is not called to perform command authorization for the LOGOFF command. All operators are implicitly authorized to use this command. However, the command is audited by the RACF SMF record type 80 for ENVIR=DELETE processing.

10.3 Command Security Authorization

Using the OPERCMDS class, installations can control which groups of users (operators amd system programmers) can issue commands. RACF authorizes or restricts users from entering some or all commands through the use of profiles.

10.3.1 Options

Operator command security in this new environment provides two main functions or options:

- To provide SAF/RACF control when entering:
  - MVS commands
  - JES3 commands
  - JES2 commands.
- To provide an audit trail for all commands entered in an MVS/ESA system. RACF generates security audit records that indicate:
  - Who issued the command
  - When the command was issued
  - Whether the issuer was authorized for the command
  - The command text.

Note: If RACF is not active or the RACF options that control operator commands are not active, then the existing JES or MCS controls are used for command authorization.

10.3.2 Command Authorization with SAF/RACF

All commands in an MVS/ESA environment are authorized by passing the following information to SAF, which is then used by RACF to determine the authorization allowed:

- A security token representing the issuer of the command.
- The profile name of the command and the access level required to issue the command.

Command processors that do their own RACROUTEs for command authorization must base their actions on the results of the RACROUTE in the same manner as console services. Therefore, it is strongly recommended that any command processor that takes its command authorization from console services use the new command authorization service, CMDAUTH. This new CMDAUTH service is used by JES2 Release 3.1.3. CMDAUTH is a general service to perform SAF and RACF command authorization. JES2, as the invoker of the service, supplies the necessary information for SAF and RACF to make a security decision.

The CMDAUTH service returns the security decision to JES2, as follows:

RC=0 Operator authorized, command issued

RC=4 RACF unable to make decision, existing JES2 controls used

RC=8 Operator not authorized, command rejected

Any messages returned by CMDAUTH are returned to the issuing console.

JES3 Release 3.1.3 calls SAF and RACF directly for command authorization. JES3 uses the IATXSEC macro for command authorization as described in Figure 28 on page 73. This action is consistent with all security calls to SAF/RACF. The same return codes for commands are received as described above.

10.3.2.1 UTOKEN of the Issuer

The UTOKEN of the command issuer represents the subject of the request (Table 14). For example, if the operator issued the command, the operator's token is used for this UTOKEN.

Table 14. UTOKENs for Processing JES2/JES3 Commands

Command Source	**UTOKEN Associated with Command**
Local console (operator logged on)	Operator logged on to console
Local console (no operator logged on)	+BYPASS+ Undefined user (always RC=4)
Local card reader	Operator who started reader
Internal reader	TSO user or Job allocated to internal reader
JES2 card reader (started from init deck)	JES2 TOKEN
JES2 init deck	JES2 TOKEN

A userid of **+BYPASS+**, with LOGON(OPTIONAL), is used for commands that are issued from a console that does not require an operator to be logged on. RACF issues an RC=4, indicating that command authorization should be done by JES or MCS, depending on who issued the command.

Commands are allowed to execute if the command issuer has an access level either equal to or greater than the command access level. For instance, you are allowed to issue DISPLAY commands if you have UPDATE or CONTROL access level for this command in the OPERCMDS class profile.

Note: If RACF is not active, the existing JES or MCS authority checking is used for command authorization.

10.3.2.2 Command Profile Names for Authorization

The command profile names for JES2 and JES3 commands are shown in Appendix E, "JES3 Command Profile Names" on page 277 and Appendix F, "JES2 Command Profile Names" on page 281. When a command is issued and a security call is made for authorization, JES2 and JES3 provide the authority access level from tables they keep for all commands that they process.

10.4 MVS Command Security

Prior to MVS 3.1.3, commands were authorized by Multiple Console Support or MCS. The commands were grouped into five categories comprising of MASTER, SYS, I/O, CONS and INFO. These categories are hierarchical so that a higher authority also includes those it dominates (Table 15).

Table 15. MCS Authority

MCS AUTHORITY	DESCRIPTION
MASTER	The highest level. Consoles with this authority can issue all MVS commands.
ALL	Ffurther broken down into I/O, CONS, or SYS authorities. You can issue commands to control the installations I/O devices, the MVS Consoles, and other system commands.
INFO	In general, the user can display system activities but cannot alter them.

Authorization is defined for each terminal address and is specified in the active CONSOLxx member in SYS1.PARMLIB.

With MVS security enhancements and in conjunction with the OPERCMDS class in RACF 1.9, MCS now checks RACF for the operator's authority to issue MVS commands. Furthermore, you can audit the commands. The audit trail is written to an SMF type 80 record.

Commands can now be authorized at the userid level. This implies that the user must have a valid userid and password and a new facility to log on to the MCS console. This is described in 10.2.1, "Defining Default Console Userids" on page 166.

10.4.1 SVC 34 Command Processing

Command authorization based on decisions by SAF and RACF is added to SVC 34 processing as shown in Figure 50. The SVC 34 parameter list has been expanded to allow the caller to pass a UTOKEN and SVC 34 processing has been modified to call the CMDAUTH service. If the OPERCMDS class is active and a UTOKEN is not supplied in the parameter list, CMDAUTH creates a UTOKEN. If passed to SVC 34, this UTOKEN should contain the userid whose authority to issue the command is to be verified by RACF; if built by CMDAUTH, this token contains the userid of the issuing address space. CMDAUTH issues a RACROUTE REQUEST=AUTH for the MVS operator command to control the use of operator commands and to provide an audit trail. Table 16 shows the action taken by SVC 34 processing based on the results of the RACROUTE REQUEST=AUTH.

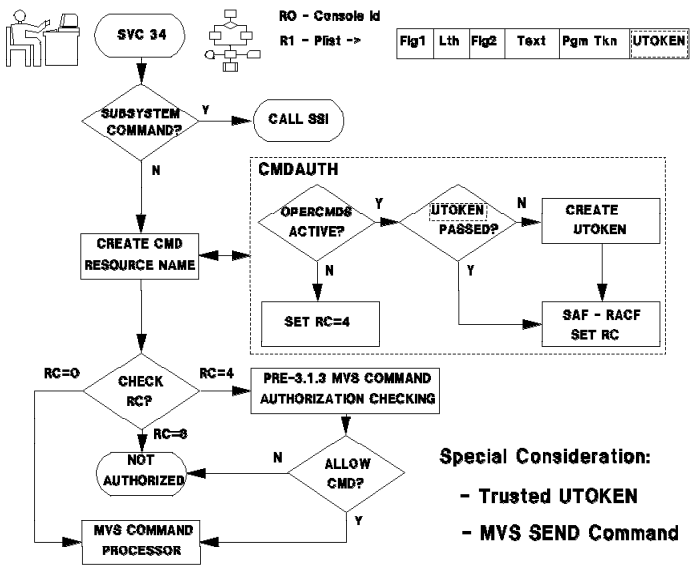

Figure 50. SVC 34 Processing

RACF may not be able to make an authorization decision (RC=4) because:

- RACF is not installed
- RACF is not active
- OPERCMDS resource class is not active
- No profile exists for the command

Table 16. SVC 34 Processing Actions

RACF Decision	SVC 34 Action
RC=0; command is authorized	Command is executed
RC=8; command was not authorized	Command is rejected
RC=4; no decision was made	MCS command authorization is used

10.4.2 OPERCMDS Resource Class

The **OPERCMDS** profile has the following format for MVS commands:

MVS.command.command-qualifier.command-object

Where:

command Usually the full MVS command. In some instances, this may be different in order to give a specific command a different authority. For example, the FORCE command, when used with the ARM parameter, uses FORCEARM in its profile.

command-qualifier Usually a keyword belonging to the command in question. This allows the user to give different access depending on the keyword. For example, 'SET IOS=xx' uses the profile 'MVS.SET.IOS' to check for authority.

command-object Specifies the target of the command, such as a device or job.

In the example below, the MVS force command requires master authority for the console to enter the command. The equivalent RACF authority or access level is CONTROL.

```
RDEFINE  OPERCMDS  MVS.FORCE.TSU.**  UACC(CONTROL)
```

With OPERCMDS activated, MCS issues a RACROUTE REQUEST=AUTH for a RACF resource name 'MVS.FORCE.TSU.userid'. This requires that the user has CONTROL access authority over that resource. A complete list of profiles is listed in *MVS/ESA Operations: System Commands*.

The "old" MCS authority and the RACF access required are mapped in Table 17. MVS, JES2, and JES3 commands all have been assigned RACF ACCESS levels that are equivalent to their "old" authority levels. These access levels are used by RACF for command authorization if you activate the OPERCMDS resource class. The JES3 and JES2 operator command profile names are shown in Appendix E, "JES3 Command Profile Names" on page 277 and Appendix F, "JES2 Command Profile Names" on page 281.

Table 17. MCS Authority with RACF

MCS Authority	ACCESS Required
MASTER	CONTROL
ALL (SYS, IO, CONS)	UPDATE
INFO	READ

MCS authority level ALL includes MCS levels SYS, IO, and CONS. When the RACROUTE call is made for a command requiring MASTER authority, ACCESS=CONTROL will be specified; for a command requiring SYS, IO, or CONS, ACCESS=UPDATE is specified; for any command requiring INFO, ACCESS=READ is specified. This translation is provided in the interest of simplifying security management.

The command authorization provided by MVS Release 3.1.3 is compatible with the previous release of MVS if RACF 1.9 is not installed, or if the security resource class OPERCMDS is not active. To build your security system for consoles, consider the following:

- RACF user profiles for all operators
- Command resource profiles to cover all commands
- All procedure names in the RACF Started Procedure Table
- IPL with a CONSOLxx member that specifies LOGON(REQUIRED)

10.4.3 MVS Set-Up Procedure

This function does not require any parameter during MVS initialization. However, in order to ensure that there is no serious security breach if RACF is down, all secondary consoles should be defined as INFO and can be altered from the master console if necessary.

10.4.4 RACF Set-Up Procedure

The MCS authority is implemented in RACF through the access authority for each command profile that you define. Depending on your security requirements, this could be as simple as three profile definitions, or more if you want better control. An example is shown later in this section.

10.4.5 Create a Started Procedures Table Entry

Some started tasks, such as JES3 and RMF, issue MVS commands through SVC 34. For instance, JES3 issues a SEND command to notify users of job termination. These procedures must be defined as trusted, otherwise an ICH408I message is issued.

To avoid this, a Started Procedures Table (SPT) should be updated before activating OPERCMDS. An IPL is required when the SPT is updated.

10.4.6 Activating OPERCMDS Resource Class

The following commands are needed to activate OPERCMDS:

```
SETROPTS  GENERIC(OPERCMDS)  GENCMD(OPERCMDS)  EGN
SETROPTS  CLASSACT(OPERCMDS)  RACLIST(OPERCMDS)
```

Use this definition of OPERCMDS for generic profiles and commands.

When activating the OPERCMDS class, you should RACLIST the class to:

- Prevent long waits in MVS.
- Continue operations should the RACF database have an error.

For more discussion on OPERCMDS refer to *RACF Security Administrators Guide*.

OPERCMDS Active - No Operator Logged on: The default mode is not active, for the OPERCMDS class.

After activating the OPERCMDS class using the SETROPTS command, define a default profile with UACC(NONE) and FAIL(WARNING) set on. This approach allows unauthorized commands to be executed with an accompanied message in the SYSLOG noting the violation. This should be allowed only during the testing period and the WARNING(FAIL) set off when you are satisfied with the results:

```
RDEFINE  OPERCMDS  MVS.**  UACC(NONE)  FAIL(WARNING)
```

Note: If the OPERCMDS class is RACLISTed, the WARNING(FAIL) option does not allow the user to be authorized to the system command. Do not RACLIST the OPERCMDS class if you want the WARNING(FAIL) option to work. Since there is no user logged on, RACF cannot validate the command and the authority defaults to the MCS. This is the way the system was designed.

10.4.7 RACF Class and MCS Class Authorities

This section describes what happens when there is a conflict between MCS and RACF authority when commands are entered.

The ″Test Matrix″ tables illustrate what happens when a userid P0112ZZ as the operator id, enters the following three commands, one for each type, as shown in Table 18.

Command	Authority
K M	To test MASTER or CONTROL command
F LLA,REFRESH	To test ALL or UPDATE authority
D A	To test INFO or READ authority

Each command represents one of the MCS authority levels and needs the required RACF ACCESS levels to be authorized, as shown in Table 17.

10.4.7.1 RACF Authority Equal to MCS Authority

Enter the following commands to match the profile with the MCS authority:

```
VARY  8FF,CONSOLE,AUTH=INFO
PERMIT  MVS.**  ID(P0112ZZ)  ACCESS(READ)  CLASS(OPERCMDS)
```

Console 8FF now has INFO authority. Operator P0112ZZ has been given READ access to all MVS commands.

The result of the test are shown in Table 18.

Table 18. Test Matrix 1

Command	Command Access	Operators Access	MCS Authority	Comments
K M	Control	Read	INFO	Message ICH408I Insufficient Access (note 1)
F LLA	Update	Read	INFO	Message ICH408I Insufficient Access (note 1)
D A	Read	Read	INFO	Command was executed (note 2)

Note:

(1) Since the command required CONTROL authority and the profile allowed READ, it is rejected and you get an ICH408I message.

(2) Requested Access permitted and the **D A** command is executed.

10.4.7.2 RACF Authority is Greater than MCS Authority

Console 8FF has INFO authority. Operator P0112ZZ is given UPDATE access to all MVS commands:

```
PERMIT  MVS.**  ID(P0112ZZ)  ACCESS(UPDATE)  CLASS(OPERCMDS)
```

The results are shown in Table 19.

Table 19. Test Matrix 2

Command	Command Access	Operators Access	MCS Authority	Comments
K M	Control	Update	INFO	Message ICH408I (note 1)
F LLA	Update	Update	INFO	Command was executed (note 2)
D A	Read	Update	INFO	Command was executed (note 2)

Note:

(1) The command issued required CONTROL and the profile had UPDATE authority, so the command fails.

(2) The RACF authority is greater than the MCS authority and the command is executed. RACF authority dominates MCS authority.

10.4.7.3 RACF Authority is Less than MCS Authority

Console 8FF has ALL authority. Operator P0112ZZ is given READ access to all MVS commands:

```
PERMIT   MVS.**   ID(P0112ZZ)   ACCESS(READ)   CLASS(OPERCMDS)
```

The results show that the operator's authority is the deciding factor. Since the logged on operator, P0112ZZ, had only READ access, the only command allowed in the test was the command with an ACCESS profile of READ.

The results are shown in Table 20.

Table 20. Test Matrix 3

Command	Command Access	Operators Access	MCS Authority	Comments
K M	Control	Read	ALL	Message ICH408I Insufficient Access
F LLA	Update	Read	ALL	Message ICH408I Insufficient Access
D A	Read	Read	ALL	Command was executed

10.4.7.4 Results and Comments

The above examples show that RACF authority overrides MCS authority as long as a user is logged on.

Ensure that security consoles should as a minimum be AUTO logged on.

Another concern is that it is possible that two MSTCONS operators could log on. In previous releases of RACF, there is no protection for consoles, and this function is controlled by having only one MASTER console. To prevent two or more operators from logging on as MSTCONS operators, use the OPERCMDS class in conjunction with the CONSOLE class.

10.4.8 Command Logging and Auditing

If you want to log all MVS commands as they are entered by the operator, enter the RACF command:

```
RDEFINE   MVS.**   AUDIT(ALL)
```

Care should be taken in doing this because it could generate enough SMF activity to impact overall system performance. Since we have more granulariy in defining profiles, we can now do the same for auditing. To audit all commands that changes the date or time, we could do the following:

```
RDEFINE   MVS.SET.TIME*   UACC(NONE)   AUDIT(ALL)
```

For more information see 4.14, “Auditing Enhancements” on page 55.

10.5 Command Security in a JES3 Environment

To control the use of JES3 operator commands using RACF, the RACF class, **OPERCMDS**, must be active. The commands can be protected by RACF profiles that specify which operators can issue commands.

To control the use of operator commands (MVS and JES3) in a JES3 system, it is necessary to use only MCS consoles. If JES3 consoles are used, RACF is not used to control the commands entered from these consoles and SMF logging is not available for them. Commands entered on JES3 consoles have the same authority checking as in all previous releases.

10.5.1 Operator Command Access Levels

JES3 command groups are associated with RACF access levels. Thus, each JES3 command has a RACF defined command access level of CONTROL, UPDATE, or READ. The equivalent MCS, JES3, and RACF command authority levels are shown in Table 21.

Table 21. MCS/JES3 Command Authority Levels and Equivalent RACF Level

MCS Authority Level	JES3 Authority Level	RACF Access Level
MASTER	15 *FREE, *DUMP *FAIL, *RETURN	CONTROL
ALL (SYS, IO, CONS)	5 and 10 *X, *C, *R, *T, *S *DELAY *ENABLE *SWITCH, *V, *F *DISABLE	UPDATE
INFO	0 *ERASE, *I, *MESSAGE	READ

10.5.2 JES3 Consoles and the OPERCMDS Class Inactive

JES3 command authority is based on the console from which the command is issued. Each JES3 command has a JES3 defined authority associated with it. The JES3 ranges are 15, 5 and 10, and 0, as shown in Table 21. For JES3 consoles, the authority level is placed on the CONSOLE initialization statement to define the JES3 command authority level. The AUTH parameter of the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB determines which JES3 commands are allowable from an MCS console. Commands are processed if the authority level of the console is equal to or higher than the command authority.

10.5.3 JES3 Profiles Definitions in OPERCMDS Class

With the OPERCMDS class, RACF can be used to authorize JES3 commands. JES3 passes the following to SAF/RACF for security checking:

- The UTOKEN of the issuer of the command.
- The profile name for the command. These profiles are shown in Appendix E, “JES3 Command Profile Names” on page 277.
- The access level required to issue the command.

Profiles can be defined in the OPERCMDS class for the authorization of JES3 operator commands to allow operators to have authority to issue them. Each command has a profile name associated with its command name in the form:

```
JES3.command-verb.qualifier
```

DSP routines that can be invoked by a //*PROCESS statement have profile names and can be protected by RACF. For example, the profile name of the DR (Disk Reader) DSP, when it is called by a //*PROCESS statement, is **JES3.PROCESS.DR.membername**.

To effect the JES3 command authorization shown in Figure 46 on page 165, the following profiles in the OPERCMDS class have to be defined:

```
RDEFINE   OPERCMDS JES3.**    OWNER(OPER)
PERMIT   JES3.** CLASS(OPERCMDS)   ACC(READ)   ID(OPER DEFOPER SETUP OUTSERV HLPDESK PRODCTL NETWORK)
PERMIT   JES3.** CLASS(OPERCMDS)   ACC(CONTROL)   ID(MSTCONS SYSPROG)
```

The parameter **FROM(JES3.**)** used in the definition of the profiles in Figure 51, causes the information contained in profile **JES3.**** to be copied into the newly created profile. This causes the **OWNER** information and the access list to also be copied into the new profile. For example, the access list of profile **JES3.MODIFY.F** has the following entries:

- MSTCONS and SYSPROG with access authority CONTROL
- OPER, DEFOPER, SETUP, HLPDESK, PRODCTL, and NETWORK with access authority READ
- OUTSERV with access authority of UPDATE.

For this profile, and most of the other profiles defined later, the minimum required access authority is UPDATE. Thus, the groups with READ access authority cannot use the commands, making the entry meaningless. Only MSTCONS, SYSPROG, and OUTSERV are authorized to use the command *F F,... . The parameter **FROM** is used to reduce the administrative work and the possibility of error.

```
RDEFINE   OPERCMDS   JES3.MODIFY.F   FROM(JES3.**)
PERMIT   JES3.MODIFY.F   CLASS(OPERCMDS)   ACC(UPDATE)   ID(OUTSERV)

RDEFINE   OPERCMDS   JES3.MODIFY.G    FROM(JES3.**)
PERMIT   JES3.MODIFY.G   CLASS(OPERCMDS)   ACC(UPDATE)   ID(PRODCTL)

RDEFINE   OPERCMDS   JES3.MODIFY.JOB    FROM(JES3.**)
PERMIT   JES3.MODIFY.JOB   CLASS(OPERCMDS)   ACC(UPDATE)   ID(SETUP OUTSERV PRODCTL HLPDESK)

RDEFINE   OPERCMDS   JES3.MODIFY.JOBP    FROM(JES3.**)
PERMIT   JES3.MODIFY.JOBP   CLASS(OPERCMDS)   ACC(UPDATE)   ID(PRODCTL)

RDEFINE   OPERCMDS   JES3.MODIFY.N    FROM(JES3.**)
PERMIT   JES3.MODIFY.N   CLASS(OPERCMDS)   ACC(UPDATE)   ID(PRODCTL HLPDESK)

RDEFINE   OPERCMDS   JES3.MODIFY.S    FROM(JES3.**)
PERMIT   JES3.MODIFY.S   CLASS(OPERCMDS)   ACC(UPDATE)   ID(SETUP)

RDEFINE   OPERCMDS   JES3.MODIFY.U   FROM(JES3.**)
PERMIT   JES3.MODIFY.U   CLASS(OPERCMDS)   ACC(UPDATE)   ID(OUTSERV PRODCTL HLPDESK)

RDEFINE   OPERCMDS   JES3.VARY.DEV   FROM(JES3.**)
PERMIT   JES3.VARY.DEV   CLASS(OPERCMDS)   ACC(UPDATE)   ID(SETUP OUTSERV)

RDEFINE   OPERCMDS   JES3.PROCESS.DIS*   FROM(JES3.**)

RDEFINE   OPERCMDS   JES3.ROUTE.CMD.*   FROM(JES3.**)
PERMIT   JES3.ROUTE.CMD.*   CLASS(OPERCMDS)   ACC(UPDATE)   ID(PRODCTL HLPDESK)

RDEFINE   OPERCMDS JES3.*.SETUP   FROM(JES3.**)
PERMIT   JES3.*.SETUP   CLASS(OPERCMDS)   ACC(UPDATE)   ID(SETUP)

RDEFINE   OPERCMDS JES3.*.WTR   FROM(JES3.**)
PERMIT   JES3.*.WTR   CLASS(OPERCMDS)   ACC(UPDATE)   ID(OUTSERV)

RDEFINE   OPERCMDS JES3.TRACE   OWNER(OPER)
PERMIT   JES3.TRACE   CLASS(OPERCMDS)   ACC(CONTROL)   ID(SYSPROG)

RDEFINE   OPERCMDS   JES3.PROCESS.**    FROM(JES3.TRACE)

RDEFINE   OPERCMDS   JES3.*.DC    FROM(JES3.TRACE)
```

Figure 51. JES3 OPERCMDS Profile Examples

10.5.4 JES3 Command Processing

We have emphasized commands that are entered from MCS consoles; however, there are additional sources from which JES3 commands may enter the JES3 system, such as:

- SVC 34
- DSPs from internally generated commands via INTERCOM
- Readers, RJP, NJE and BDT.

Table 22 shows where the security check is made for the various command sources and the UTOKEN that is associated with the command issuer.

Command security checking is done in module IATCNIA. However, user exit IATUX18 is always called before IATCNIA and the call to SAF/RACF.

Table 22. Command Sources and JES3 Processing

Command Source	Security Check	UTOKEN
SVC 34	IATCNIA	Command Issuer
DSPs - INTERCOM	IATCNIA	Calling Operator's
INTERCOM CHK=NO	IATCNIA	JES3 token from TVT
Reader	IATCNIA	Operator that called reader
RJP console	IATCNIA	Obtained by VERIFYX at signon/logon
NJE	IATUX35, IATCNIA	Obtained by VERIFYX for node
BDT	IATUX56, IATCNIA	Provided by IATUX56

A TOKEN keyword has been added to the INTERCOM macro. An 80-byte UTOKEN field is added to the IATYS34 and IATYCNS mapping macros. For BDT, a 4-byte field containing the address of the UTOKEN is added to the IATYBDD mapping macro.

The security calls to SAF/RACF from the IATXSEC macro invoke user exits IATUX58 and IATUX59. These exits are discussed in 6.3, "JES Exits for SAF Calls" on page 73. For all commands that are rejected, RC=8 is returned to the caller. A new message is issued:

```
IAT7138 SECURITY CHECK FAILED - JES3 COMMANDS NOT PERMITTED
```

10.5.5 User Considerations for Command Authorization

For installations that have created their own JES3 commands, module IATCNIA contains the command profile names, as shown in Appendix E, "JES3 Command Profile Names" on page 277. It would be necessary to update the tables in this module to provide command authorization with SAF/RACF for user created commands. See also the section 10.3.2, "Command Authorization with SAF/RACF" on page 173 for more details about authorizing commands.

The order in which exits are called for command authorization is:

1. IATUX18 - Installation user exit
2. IATCNIA - Issues IATXSEC security call to SAF/RACF
3. IATUX58 - Pre-SAF exit
4. IATUX59 - Post SAF/RACF exit

The return codes for command authorization are described in 10.3.2, "Command Authorization with SAF/RACF" on page 173.

10.6 Command Security in a JES2 Environment

Prior to JES2 3.1.3 and RACF 1.9, protection of MVS and JES2 commands could be implemented by JES2 initialization statements, and keywords in the CONSOLxx member of PARMLIB. RACF 1.9 and JES2 3.1.3 introduces command security using the SAF interface. Use of PARMLIB keywords and JES2 initialization statements to control operator commands still function as they did previously, if you choose not to implement command security with SAF. You may also choose some combination of SAF command security and MVS/JES2 command security.

Control of consoles is required if your site chooses to implement command authorization using SAF/RACF. This provides a basic secure environment for operator commands see 10.2, “Console Security Definitions” on page 165 for protection of consoles.

If there is no operator logged on to a console, commands are authorized by MVS/JES2 security checking. With SAF/RACF command authorization enabled by activating the OPERCMDS class, a RACROUTE TYPE=VERIFYX call is made to SAF for every command entered. This call passes a token to SAF. Within the token there is a userid used for authorization, for consoles with an operator logged on the userid corresponds to the operator. For a console with no operator logged on, the userid is +BYPASS+. A userid of +BYPASS+ causes SAF to provide a return code of 4, and authorization checking defaults to MVS/JES2 security checks. For more detail, see Table 14 on page 174.

For recommendations on the LOGON options specified in the CONSOLxx member, see 10.2.7, “Console Logon Considerations” on page 172.

In addition to specifying LOGON keywords, you may also wish to protect console access. For use of the CONSOLE class refer to 10.2, “Console Security Definitions” on page 165.

10.6.1 Operator Command Access Levels

You may choose to implement your current MCS and JES authorization structure with RACF. Table 23 relates MCS and JES2 command groups to the RACF access levels.

Table 23. Cross-reference of RACF Access Level with JES2 and MVS Commands

RACF Access Level	JES2 Command Group	MCS Command Group
CONTROL	SYSTEM	MASTER, CONSOLE
UPDATE	JOB, DEVICE	SYS, IO
READ	DISPLAY	INFO

Using this table you should be able to duplicate your existing JES and MCS command authorization using RACF.

10.6.2 JES2 Profile Definitions in OPERCMDS Class

Commands are authorized with SAF/RACF using the OPERCMDS class. Each command has a profile name of the form:

```
JES2.command-verb.qualifier
```

Where:

JES2 Name of the your subsystem.

command-verb Name of the command.

qualifier The type of object to command specifies (for example, JOB or SYS).

For a full list of JES2 commands and corresponding profiles see Appendix F, "JES2 Command Profile Names" on page 281. For a more detailed discussion with MVS commands, and their considerations, see 10.4, "MVS Command Security" on page 175. This section primarily deals with JES2 commands.

Your security policy should indicate which commands should be protected and who should be authorized for various commands. An installation should:

- Define profiles for JES2 commands.
- Permit operators to issue certain commands. This access authority can be granted by userid or groupid.

For instance, you might want to give all operators a DISPLAY capability, while only giving the master console operator access to a $E SYS command:

```
RDEFINE  OPERCMDS  JES2.**  UACC(NONE)  AUDIT(ALL)
PERMIT  JES2.DISPLAY.*  CL(OPERCMDS)  ACC(READ)  ID(OPER1)
PERMIT  JES2.RESTART.SYS  CL(OPERCMDS)  ACC(CONTROL)  ID(MSTCONS)
```

Note: When defining profiles and using them in a JES2 primary and secondary system, consider using **JES%** as the subsystem name. Then, if your primary JES is JES2 and your secondary JES is JESA, the same RACF database can be used with the same profiles:

```
RDEFINE  OPERCMDS  JES%.**  UACC(NONE)  AUDIT(ALL)
PERMIT  JES%.DISPLAY.*  CL(OPERCMDS)  ACC(READ)  ID(OPER1)
PERMIT  JES%.RESTART.SYS  CL(OPERCMDS)  ACC(CONTROL)  ID(MSTCONS)
```

With RACF 1.9 and JES2 3.1.3, you are now able to expand your authorization policy beyond what you had with MCS and JES2 authorization groups. The granularity of authorization has improved with this new release of RACF. Some changes you are able to make can be added easily to your current structure. For instance, you can now have several consoles with master authority. This means you could enter, for example, the CONFIG command from a console other than the master console.

You may choose to re-organize your command authorization policy along the lines of conventional RACF authorization structures. For example, you can define groups for different types of operators, and authorize those groups to use particular commands, such as in section 10.1, "Grouping of Operator Functions" on page 164.

For each of the groups, you would allow required and appropriate commands. This type of structure would probably use the LOGON(REQUIRED) option in the CONSOLxx member. You may also consider command authorization for help desk personnel, system programmers, and application developers.

10.6.2.1 Command Examples

In the following examples, JES2 commands to be controlled by RACF are defined:

```
RDEFINE  OPERCMDS  JES2.DISPLAY.*  UACC(READ)
RDEFINE  OPERCMDS  JES2.START.*  UACC(CONTROL)
PERMIT  JES2.START.*  CL(OPERCMDS)  ACCESS(CONTROL)  ID(OPER1)
```

To activate SAF/RACF command authorization checking, enter the following commands:

```
SETROPTS   CLASSACT(OPERCMDS)
SETROPTS   GENERIC(OPERCMDS)
SETROPTS   RACLIST(OPERCMDS)
```

OPERCMDS must be RACLISTed for command authorization to occur. This is a requirement, not a recommendation. If OPERCMDS is not RACLISTed, MVS/JES2 command authorization is used. OPERCMDS must also be eligible for generic profile checking if you have defined generic profiles.

10.6.2.2 Conditional Access Checking

RACF allows conditional access for consoles, terminals, and batch jobs. The RACF classes that can be used are:

Consoles The CONSOLE class is used. Allows access from appropriate consoles.

Terminals The TERMINAL class is used. Allows access from appropriate terminals

Batch jobs The JESINPUT class is used. Allows access from NJE NODES, internal readers (not for started tasks or TSO LOGON), device readers, and RJE workstations.

With conditional access checking for a particular class, appropriate profiles must be defined in the class, and the class must be active.

To allow JES2 START commands to be allowed, for example, only from console 08, and only by OPER1, you would issue the following commands:

```
RDEFINE   OPERCMDS   JES2.START.*   UACC(NONE)
PERMIT   JES2.START.*   CL(OPERCMDS)   ACC(CONTROL)   ID(OPER1)   WHEN(CONSOLE(08))
SETROPTS   RACLIST(OPERCMDS)   REFRESH
```

For more information see 4.13, “New Forms of Conditional Access” on page 53.

10.6.2.3 Commands from NJE/RJE

JES2 commands may be entered directly from NJE nodes or RJE workstations. For NJE, the $N or $M commands are used. RACF allows you to define an NJE node, and to permit or deny commands from that node. To enable SAF/RACF command authorization for NJE/RJE, you must:

- Define the NJE node/RJE workstation in the FACILITY class. The profile is of the form:

  ```
  NJE.nodename or RJE.RMTnnnn
  ```

- Define the NJE node/RJE workstation as a valid RACF user using the node/remote name.
- Permit the USERID to the appropriate commands.

The following sample commands allow JES2 DISPLAY commands from NJE node C6JES2 to be executed on node C2JES2:

```
ADDUSER   C6JES2   DFLTGRP(NETWORK)
RDEFINE   FACILITY   NJE.C6JES2   UACC(NONE)
PERMIT   JES2.DISPLAY.**   CL(OPERCMDS)   ACC(READ)   ID(C6JES2)
```

10.6.3 Security Label Considerations

If you are using security labels, the SECLABEL associated with the job or session issuing a command must dominate the SECLABEL of the profile that covers the command.

If you have activated SECLABELs for operator commands, it is possible you are using SECLABELs for the CONSOLE class. You should be aware that SECLABEL dominance checks for the CONSOLE class are reversed. You cannot log on to consoles if the user SECLABEL dominates the CONSOLE SECLABEL. The CONSOLE security label must dominate the user logging on. This is appropriate since the only purpose a user can achieve by logging on to a console is to issue commands. You should not be able to issue a command where the SECLABEL of the command dominates the SECLABEL of the console.

10.6.4 Special JES2 Commands

If you activate command security authorization, it is necessary to consider these changes to JES2 commands:

$VS command

The $VS command is no longer valid from an MCS console, although this command is still allowed from internal readers.

$PJES2,ABEND,FORCE

$TCKPTDEF,RECONFIG=YES

These commands should be protected so that only authorized operators can enter them. The above two commands are examples of and each user has to evaluate his own needs. For a complete list of all the command profiles that can be defined in a JES2 environment, see Appendix F, “JES2 Command Profile Names” on page 281.

10.6.5 JES2 Automatic Commands

For the following automatic commands, authority is required for both the automatic command and the subject commands:

$SA Start automatic command processing

$TA Define or change an automatic command

$CA Cancel automatic command processing

$ZA Halt automatic command processing

Table 24 shows how these commands are processed. Also, note the following:

- CONTROL access is required to start or halt automatic command processing with the $SA and $ZA commands.
- When an operator defines or changes an automatic command with the $TA command, the operator must also have the authority to issue the subject command.
- If a different operator modifies or cancels another operator’s automatic commands, CONTROL access is required.

Table 24. JES2 Automatic Command Processing		
Command	**UTOKEN**	**Access Required**
Note: OPER1 starts or halts automatic commands		
$SA	Issuer of $SA command	CONTROL
$ZA	Issuer of $ZA command	CONTROL
Note: OPER1 defines or cancels automatic commands		
$TA	Issuer of $TA command	READ
$CA	Issuer of $CA command	READ
Note: OPER2 now changes or cancels OPER1′s commands		
$TA	Issuer of $TA command	CONTROL
$CA	Issuer of $CA command	CONTROL

10.6.6 JES2 Command Processing

JES2 commands enter the system from the following sources: (Table 25).

- Local consoles
- Local and internal readers
- RJE and NJE
- JES2 initialization parms.

JES2 Exit 5 is invoked after the intial editing of the command. Exit 5 can issue an RC=8 and no further command authorization is done. The $SEAS macro is used to start command authorization. See Figure 28 on page 73 for discussion of this macro. JES2 maintains a command table with the profile names and access levels for command authorization. These are shown in Appendix F, “JES2 Command Profile Names” on page 281. A $RACROUTE routine issues the CMDAUTH macro that passes the text, profile name, access level, and UTOKEN to SAF/RACF. This replaces the RACROUTE VERIFYX call shown in the Figure 28 on page 73.

Table 25. Command Sources and JES2 Processing		
Command Source	**Security Check**	**UTOKEN**
Local Consoles (SVC 34)	CMDAUTH call	Logged on Operator
Internal reader	CMDAUTH call	$DCT has token of caller
Local reader	CMDAUTH call	$DCT has token of caller
Initialization parms	CMDAUTH call	JES2 address space

10.6.6.1 Command Authorization Return Codes

The $SEAS macro call for command authorization has the following return codes:

RC=0 Normal JES2 processing of the command.

RC=4 Use command authorization as is done in all previous releases of JES2.

RC=8 The command is rejected.

When commands are rejected, the $HASP690 message is issued:

```
$HASP690  text of RACF message
$HASP690  COMMAND REJECTED - AUTHORIZATION FAILURE
```

10.6.6.2 Unknown Command Processing

When an operator issues a command that is unknown, the $SEAS call for authorization is made to audit the command. When JES2 checks the profile name table and finds no command, a profile name of **JES2.UNKNOWN** is passed with the $SEAS call.

10.6.7 Audit and Logging

Auditing can be activated and deactivated in the normal way for command authorization. If auditing is enabled, the entire text of the command is recorded in an SMF type 80 record.

Calls are made to SAF even if a command is not valid. A special profile, JES2.UNKNOWN, can be established in the OPERCMDS class to log unknown commands.

10.6.8 User Considerations

With the new security related operator commands in this release of JES2, several initialization changes have been made. The JES2 initialization statements **INTRDR** and **RDR(nn)** have an **AUTH** keyword that specifies a value for the level of commands that are allowed to be issued through the internal reader and local card readers. The default value of 0 is changed to 7. This change now allows only display commands to be issued.

Note: These command levels are used only if RACF is not active or the OPERCMDS class is not active.

Chapter 11. Functional Subsystem Printing Enhancements

All printing or output on a JES Release 3.1.3 and RACF 1.9.0 system undergoes authorization checking for output destination and spool access when corresponding RACF resource classes have been activated. When output is selected, two calls are made to the SAF/RACF interface. These calls check for authorization to a particular device, and for a user's authority to read the particular spool data set.

PSF/MVS Release 3.0 introduces the concept of print labeling. This function allows you to place identification labels on each page of print output (Figure 52). An identification label can be composed of text, graphics, or a combination of text and graphics. The printed page is composed of two areas; the user printable area **(UPA)**, where the user is allowed to print his data, and an area where security information, called identification labels, can be printed. See Figure 53 on page 193.

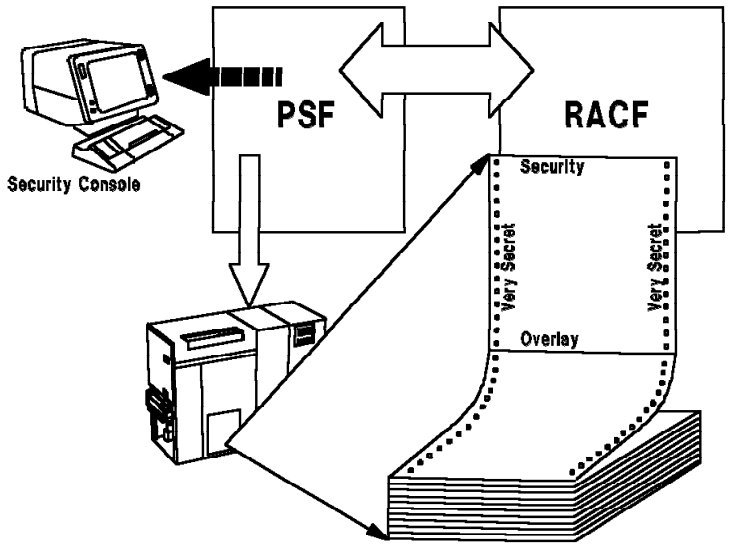

Figure 52. Identification Labeling with PSF/MVS

The following new terms are introduced with this release:

Identification labeling This process places security overlays on print output. These overlays collectively are called an identification label, and relate to the security label of the output data set.

UPA enforcement This allows an installation to define an area of the output page that a user may print on (user printable area). Any attempt to print outside of the UPA fails.

Guaranteed print labeling The integrity of the identification label is ensured by the printer hardware. This integrity is ensured by the positioning of the label on an area of the page where the user data cannot be placed.

Nonguaranteed print labeling The integrity of the identification label is not ensured by the printer hardware. This label can be placed on the page and be overlayed by user data.

Separator page labeling Security number labeling chooses a random number and places it on the job header and trailer pages. This allows header and trailer pages to be associated making it easy for operators to identify a job's output.

Trailer page enforcement Printers that are controlled by PSF/MVS in a secure environment msut always print trailer pages. A dummy data set is used to force a trailer page if the output is marked unprintable by PSF/MVS. With JES controlled printers, a trailer page is not forced.

Identification labels may consist of security overlays, security page segments, and Security Font Library members. Identification labels are stored in the Security Overlay Library. Security related data that can be printed on the pages of the output includes:

- Separator page security numbers that can be printed on the header page and trailer page to ensure that the output belongs to the same job.
- Identification labels that can be printed on each page of the printed output, including:
  - Job header, data-set header, and trailer pages
  - User data pages and PSF/MVS message pages

All messages that are security related can be displayed on a security console or the master console. The security console is a new option for PSF/MVS messages.

The minimum requirement to print identification labels on the output is the activation of the RACF SECLABEL resource class. For the full B1 function, the RACF class PSFMPL must be active. PSF/MVS calls RACF when a user wishes to override the printing of identification labels or to suppress the user printable area definition and print on the total page. A user profile must exist for a user to override security overlays and suppress the UPA definition.

Both SAF/RACF calls are applicable for all types of output on RACF 1.9 and JES 3.1.3 systems. In addition to these authorizations checks, Functional Subsystem (FSS) printers also provide print labeling. Print labeling is enabled with the levels of JES and RACF indicated above, and with PSF/MVS 1.3.0.

As shown in Figure 53, both the UPA and the area where identification labels are printed must be within the valid printable area of the physical page. An identification label is obtained from the Security Overlay Library by its SECLABEL name. The identification label may contain up to eight security overlay names. Figure 53 shows three security overlays.

The valid printable area on one physical page is restricted by the printer's capabilities. All page printers have a printable area that is half an inch smaller on every side than the paper itself.

The UPA is the area where the user is allowed to print his user generated data, text, and graphics. The UPA must be within the printable area of the physical page. The user cannot overwrite any identification labels; that is, cannot print outside of the UPA.

A user must determine the size and the location of the UPA based on his application requirements and identification labeling requirements.

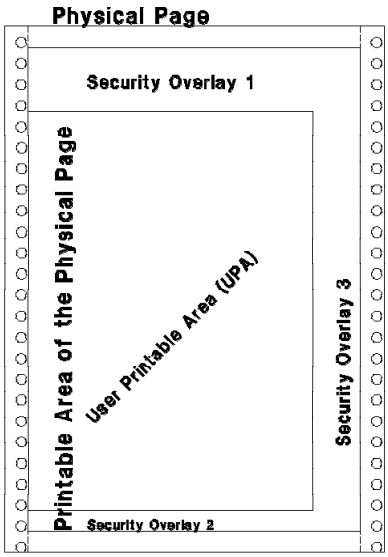

Figure 53. User Printable Area

You can define the size and the location of your UPA in the Security Definitions Library for each SECLABEL you need.

All the space outside of the UPA and within the printable area of the physical page is reserved for identification labels. Identification labels must be printed within the printable area of the physical page and must not be printed in the UPA.

An identification label is placed on the page using a resource called a security overlay. The overlays, fonts, and page segments used by the overlays, reside in special security libraries. The user requests the identification label by specifying a SECLABEL on the job statement and PSF/MVS uses the SECLABEL to find the security overlays identified for that SECLABEL.

This chapter discusses the security implementations for FSS printing environments. Some of the material is applicable to conventional local printing.

11.1 Printer and Software Requirements

There are a number of requirements that must be understood before using print labeling; namely:

- Print labeling is supported only in deferred print mode. Directly attached printers are not supported.
- JES 3.1.3 and RACF 1.9 must be installed.
- The IBM 3825, 3827, and 3835 printers are the only printers that enforce the UPA and ensure integrity of identification labels. These are called 'guaranteed' printers.
- Other PSF/MVS printers (the IBM 3800, 3812, 3816, and 3820 printers) can print identification labels, but do not enforce the UPA, or guarantee the integrity of the identification labels. These printers are 'non-guaranteed' printers.

- Particular levels of microcode are required for print label functions. You should check the program directory for PSF/MVS, and with IBM to obtain the correct microcode for your printer.

The microcode level of your printers is important because of the introduction of two new printer control commands:

OPC - Obtain Printer Characteristics
DUA - Define User Area

If you do not have the correct level of microcode and you attempt to define a UPA, the following message is received when PSF/MVS abends:

```
APS050I APSWPR2 FSS27 *** PRT2 (B10) SYSTEM COMPLETION
(CONT.) CODE ¢024¢X, PSF ABEND REASON CODE ¢0237¢X, HAS BEEN
(CONT.) DETECTED BY APSEFSA
```

11.2 Print Labeling Implementation

The basic implementation of print labeling may be achieved in the following steps, which assume PSF/MVS is already installed and operational at your site:

1. Check printer and software requirements
2. Design pages and overlays for print labeling
3. Allocate security resource and definition libraries
4. Define security definitions
5. Update printer start-up procedures, review JCL changes
6. Assemble the sample module for random numbers on separator pages
7. Enable print labeling.

The discussion of implementation of print labeling is well documented in the PSF/MVS Security Guide.

Before starting to implement print labeling you should ensure you have read and understand 11.3, “Implications for Print Labeling” on page 201.

11.2.1 Designing Pages

You must consider the design of your pages carefully. Some of the factors you may consider are the size of the stationery used, the requirements of print applications, and the SECLABELs that are to be used by your organization.

Your page design should include the size you define for your UPA, and some indication of what your security overlays should be like. Security overlays are combined at print time to create the identification label.

When designing security overlays there is a relationship between the complexity of overlays and printer performance. Complex overlays consume more printer storage. If you plan to use a large number of fonts and page segments in security overlays, you should consider the potential printer performance problems you may be inviting.

When you design your overlays, remember that several overlays can be used to produce an identification label. If you use a base set of overlays to produce multiple identification labels, you save repetitive work if changes to your identification labels are required. For instance, consider a security label PBRCON. This SECLABEL includes the categories Personnel, Business, and Research. The label has a security level of confidential. Consider also the SECLABELs PCON, BCON, and RCON. These labels are for the Personnel, Business, and Research categories respectively. All three SECLABELs

are for the confidential security level. We could create single overlays to be used as the identification labels for the PCON, BCON, and RCON security labels. To save work, and to lessen the administrative burden, we could then combine the three overlays we have created (in the security definitions) to define the identification label for the PBRCON security label.

See 11.3, “Implications for Print Labeling” on page 201 for a discussion of other issues that may affect your page design.

11.2.2 UPA Definition

Print data is positioned within the UPA by the use of appropriate form and page definitions. The page origin, as used by form and page definitions, remains unchanged when UPA enforcement is invoked. This means when you introduce UPA protection, you have to review the FORMDEFs and PAGDEFs you currently use for printing. If your current printing places text outside the UPA area, you have to adjust the form and page definitions used.

If you attempt to print outside the UPA, and UPA protection is in place, the data is not printed. Print error marks are placed on the page boundary where the problems occurred, and error messages are issued. This is the same as trying to print outside the logical page boundary on a system not using print labeling.

The UPA is enforced only for user data pages. Separator, message, and system data set pages do not have UPA protection. For example, consider the job in Figure 54. The output from this job consists of five data sets, plus job header and trailer pages. The five data sets are produced from the following DD cards:

JESMSGLG
JESJCL
JESYSMSG
SYSPRINT
SYSUT2

The UPA is enforced only for the data sets defined by the SYSPRINT and SYSUT2 DD statements. On all the other pages produced when the job is printed, the UPA is not enforced.

```
//P0112DP@ JOB (999,POK),¢SECLABEL BRCON¢,CLASS=A,NOTIFY=P0112DP,
//        MSGLEVEL=(1,1),MSGCLASS=I,SECLABEL=BRCON
/*ROUTE PRINT PR3835
//*********************************************************************
//*
//*  SAMPLE JOB FOR TESTING NEW PRINT SECURITY FUNCTIONS
//*  AVAILABLE WITH RACF 1.9.0 AND PSF/MVS 1.3.0
//*
//*********************************************************************
//STEP1    EXEC PGM=IEBPTPCH
//OUTPUT1  OUTPUT DEFAULT=YES,
//         DPAGELBL=NO,SYSAREA=NO
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=SYS1.PROCLIB,DISP=SHR
//         DD   DSN=SECURE.SECDEFS,DISP=SHR
//SYSUT2   DD   SYSOUT=*
//SYSIN    DD   *
   PRINT  TYPORG=PO,MAXFLDS=2,MAXNAME=2
   MEMBER NAME=APSWPR10
   RECORD FIELD=(72)
   MEMBER NAME=DEFLABEL
   RECORD FIELD=(72)
/*
```

Figure 54. JCL Using DPAGELBL and SYSAREA on the OUTPUT Card

UPA protection is possible only on guaranteed printers. If you try to enforce the UPA on a non-guaranteed printer, PSF/MVS abends. Your requirements determine your page layout. The space for security overlays has to be sufficient for the size of your overlays. Security overlays can be placed on the top, bottom, or sides of all pages within a job. In addition, the space defined for your UPA has to fit your users' requirements.

After you determine the size and location of the UPA, you can define the layout in the Security Definition Library. For each paper size and paper name, you must create a single member in this data set. The definition of the size and location of the UPA is as follows:

- Identify each form size that can be in the printer:
  PAPERNAM paper name | PAPERSIZ width height
- Specify whether the top of the form will be narrow or wide:
  TOP (NARROW|WIDE)
- Specify the starting location of the UPA:
  UPAORG xorg yorg
- Specify the size of the UPA:
  UPADIM xlength ylength
- Identify the Security Overlay Name(s):
  SECOVLY overlay1 overlay2 ... overlay8

11.2.3 System-Defined Paper Names

You can use the PAPERNAM keyword to specify one of the paper names shown in Table 26 on page 197, if it fits your paper size. If not, you must specify the size with the PAPERSIZ keyword.

Table 26. Paper Names and Sizes		
Paper Name	**Size in mm**	**Size in inches**
A4	210 x 297	8.27 x 11.69
B4	257 x 364	10.12 x 14.33
B5	182 x 257	7.17 x 10.12
EXEC1	178 x 267	7.00 x 10.50
EXEC2	184 x 267	7.25 x 10.50
EXEC3	190 x 267	7.50 x 10.50
LEGAL	216 x 356	8.50 x 14.00
LETTER	216 x 279	8.50 x 11.00
LISTING1	378 x 279	14.88 x 11.00
LISTING2	305 x 216	12.00 x 8.50
LISTING3	279 x 216	11.00 x 8.50

11.2.4 Security Resource and Definition Libraries

Another implementation activity you must perform is to allocate security resource libraries, and a library for the security definitions you code. Table 27 lists the libraries, with the attributes you have to allocate for print labeling.

Table 27. Security Libraries					
Library Contents	**Data Set Name**	**DSORG**	**LRECL**	**BLKSIZE**	**RECFM**
Security Definitions	SECURE.SECDEFS	PO	8205	8209	VBM
Security Fonts	SECURE.FONTLIB	PO	8205	8209	VBM
Security Fonts	SECURE.FONTLIBB	PO	8205	8209	VBM
Security Overlays	SECURE.OVERLIBB	PO	8205	8209	VBM
Security Page Segments	SECURE.PSEGLIBB	PO	8205	8209	VBM

Note: The data set names in the table are not compulsory, but they must be consistent with DD cards in the printer start-up procedure.

11.2.5 Security Definitions

To relate security overlays and the size of the UPA to a particular SECLABEL, you must code security definitions. Figure 55 contains a sample definition for the security label of BRCON. The security definition member name must match the SECLABEL name the definitions are related to. For example, the security definitions for the BRCON security label are found in the BRCON member of the security definitions library.

```
*  SECURE.SECDEFS(BRCON)
***  DEFINITION FOR CUT SHEET PAPER
     PAPERNAM LETTER
     TOP      NARROW
     UPAORG   1.00   IN  1.00   IN
     UPADIM   7.25   IN  9.75   IN
     SECOVLY  BUSCON RESCON
***  DEFINITION FOR NARROW CONTINUOUS FORMS
     PAPERSIZ  9.87  IN 11.00   IN
     TOP      NARROW
     UPAORG   1.00   IN  1.00   IN
     UPADIM   7.25   IN  9.75   IN
     SECOVLY  BUSCON RESCON
***  DEFINITION FOR WIDE CONTINUOUS FORMS
     PAPERSIZ 11.00  IN  8.50   IN
     TOP      WIDE
     UPAORG   1.00   IN  0.25   IN
     UPADIM   9.75   IN  7.25   IN
     SECOVLY  BUSCOW RESCOW
*  END SECURE.SECDEFS(BRCON)
```

Figure 55. BRCON Security Label Definition

Security definitions must define the size of paper to be used for printing. A selection of paper sizes can be defined in each member. This allows you to code a single definitions member even though you may have several printers using different paper sizes. The PAPERSIZ parameter is used to define the paper size.

IBM supplies a selection of predefined paper sizes that may be used to define paper size. In Figure 55, the definition for cut sheet paper uses a predefined paper size of LETTER. In contrast, the definition for narrow continuous forms defines paper size using actual dimensions.

If you are using a 3835 printer, your defined paper size must match the paper size that has been defined by using the operator panel on the printer. If your paper size is incorrect, you may see a sequence of messages as in Figure 56.

```
APS237I APSWPR10 FSS35 *** PRT10 (B11) A PAPER SIZE
(CONT.) MATCHING THE LOADED PAPER WAS NOT FOUND IN THE SYSHIGH
(CONT.) MEMBER OF THE SECURITY DEFINITIONS LIBRARY.
APS101I APSWPR10 FSS35 *** PRT10 (B11) SYSTEM ACTION
(CONT.) TAKEN: THE ERRORS LISTED PREVENTED PRINTING OF THE DATA SET
(CONT.) FOR JOB ¢P0112DPT¢, STEP ¢********¢, OUTGROUP ¢1¢.
APS225I APSWPR10 FSS35 *** PRT10 (B11) SYSTEM ACTION
(CONT.) TAKEN: THE ERROR(S) LISTED CAUSED THE DATA SET TO BE MARKED
(CONT.) UNPRINTABLE AND TO BE PLACED ON A HOLD QUEUE FOR RELEASE BY
(CONT.) THE OPERATOR.  CONTACT THE SYSTEM OPERATOR TO REPRINT THE
(CONT.) DATA SET.
APS061I APSWPR10 FSS35 *** PRT10 (B11) SYSTEM ACTION
(CONT.) TAKEN: THE ERRORS LISTED PREVENTED PRINTING OF THE JOB
(CONT.) HEADER.  ALL DATA SETS IN THE JOB WILL BE HELD.
$HASP704 P0112DPT DATA SET UNPRINTABLE - OUTGRP=1.1.1           HELD
- FSS REASON(APS237I )
```

Definitions Member

Figure 56. PSF/MVS Messages Caused by Incorrectly Coded Paper Size

The TOP parameter is used for the 3835 if LANDSCAPE forms are used.

The definitions member also defines the UPA by a UPA origin, and then setting the dimensions of the area. The parameters used to define the UPA origin and UPA dimensions are UPAORIG and UPADIM, respectively.

Security overlays are associated with the definitions member by using the SECOVLY parameter.

Generally, if a problem is detected by PSF/MVS when processing a security definition member, the spool data set being processed is placed on hold, and not purged (Figure 56).

Note: Do not include line numbers in your security definition member as this causes errors when PSF/MVS attempts to use the member.

11.2.6 Update Printer Procedures

Figure 57 contains an example of a start-up procedure with print labeling enabled. Changes that have to be made to enable print labeling are to define the security libraries, and to specify system defaults for identification labeling and UPA enforcement.

In the example in Figure 57, DD statements have been added for the libraries SECURE.FONTLIBB, SECURE.PSEGLIB, SECURE.OVERLIB, and SECURE.SECDEFS. These DD statements have also been identified in the PRINTDEV statement using the SPSEGDD, SOVLYDD, SFONTDD, and SDEFDD parameters.

Printer defaults for identification labeling and UPA enforcement are defined using the SPAGELBL, DPAGELBL, and SYSAREA parameters on the PRINTDEV card. SPAGELBL and DPAGELBL are used respectively to specify whether identification labeling is to be enabled for separator and data pages. SYSAREA indicates whether the UPA is to be enforced. The example has identification labeling for both separator and data pages, and the UPA is being enforced.

You can override system defaults for a print job by specifying DPAGELBL and SYSAREA on the OUTPUT card of your job. This specification overrides settings for the data pages only if you are authorized. Figure 54 on page 196 contains an example in which the data pages for a print job do not have identification labels, and the UPA is not enforced.

```
//APSWPR10 PROC
//**************** THE PSF/MVS WRITER PROCEDURE ******************
//*
//*
//*   SAMPLE PSF/MVS WRITER PROCEDURE USING PRINT LABELING
//*
//*
//*****************************************************************
//*
//STEP01   EXEC PGM=APSPPIEP,REGION=4096K
//JOBHDR OUTPUT PAGEDEF=V06482,     /* JOB SEPARATOR PAGEDEF        */
//       FORMDEF=SECURE,CHARS=GT15  /* JOB SEPARATOR FORMDEF        */
//JOBTLR OUTPUT PAGEDEF=V06482,     /* JOB SEPARATOR PAGEDEF        */
//       FORMDEF=SECURE,CHARS=GT15  /* JOB SEPARATOR FORMDEF        */
//DSHDR  OUTPUT PAGEDEF=V06482,     /* DATA SET SEPARATOR PAGEDEF   */
//       FORMDEF=SECURE,CHARS=GT15  /* DATASET SEPARATOR FORMDEF    */
//MSGDS  OUTPUT PAGEDEF=P2075,      /* MESSAGE DATASET PAGEDEF      */
//       FORMDEF=SECURE             /* MESSAGE DATASET FORMDEF      */
//FONT01   DD   DSN=SYS1.FONTLIBB,  /* SYSTEM FONTS                 */
//         DISP=SHR
//PSEG01   DD   DSN=SYS1.PSEGLIB,   /* SYSTEM PAGE SEGMENTS         */
//         DISP=SHR                 /*                              */
//OLAY01   DD   DSN=SYS1.OVERLIB,   /* SYSTEM OVERLAYS              */
//         DISP=SHR                 /*                              */
//PDEF01   DD   DSN=SYS1.PDEFLIB,   /* SYSTEM PAGEDEFS              */
//         DISP=SHR
//FDEF01   DD   DSN=SYS1.FDEFLIB,   /* SYSTEM FORMDEFS              */
//         DISP=SHR
//SFONT01  DD DSN=SECURE.FONTLIBB,  /* SECURITY FONTLIBS            */
//         DISP=SHR
//SOLAY01  DD DSN=SECURE.OVERLIB,   /* SECURITY OVERLAYS            */
//         DISP=SHR
//SPSEG01  DD DSN=SECURE.PSEGLIB,   /* SECURITY PAGE SEGMENTS       */
//         DISP=SHR
//SDEF01   DD DSN=SECURE.SECDEFS,   /* SECURITY DEFINITIONS         */
//         DISP=SHR
//PRT10    CNTL
//PRT10    PRINTDEV FONTDD=*.FONT01, /* FONT LIBRARY DD             */
//           OVLYDD=*.OLAY01,        /* OVERLAY  LIBRARY DD         */
//           PSEGDD=*.PSEG01,        /* SEGMENT  LIBRARY DD         */
//           PDEFDD=*.PDEF01,        /* PAGEDEF  LIBRARY DD         */
//           FDEFDD=*.FDEF01,        /* FORMDEF  LIBRARY DD         */
//           SFONTDD=*.SFONT01,      /* SECURITY FONT    LIBRARY DD */
//           SOVLYDD=*.SOLAY01,      /* SECURITY OVERLAY LIBRARY DD */
//           SPSEGDD=*.SPSEG01,      /* SECURITY SEGMENT LIBRARY DD */
//           SDEFDD=*.SDEF01,        /* SECURITY DEFS    LIBRARY DD */
```

Figure 57 (Part 1 of 2). Example Printer Start-up Procedure

```
//          JOBHDR=*.JOBHDR,          /* JOB HEADER SEPARATOR OUTPUT  */
//          JOBTRLR=*.JOBTLR,         /* JOB TRAILER SEPARATOR OUTPUT */
//          DSHDR=*.DSHDR,            /* DATA SET HEADER SEPARATOR    */
//          MESSAGE=*.MSGDS,          /* MESSAGE DATA SET OUTPUT      */
//          BUFNO=5,                  /* NUMBER OF WRITE DATA BUFFERS */
//          SPAGELBL=YES,             /* ACTIVATE LABEL IF NO PSFMPL  */
//          DPAGELBL=YES,             /* ACTIVATE LABEL IF NO PSFMPL  */
//          SYSAREA=YES,              /* ACTIVATE UPA IF NO PSFMPL    */
//          PAGEDEF=P2075,            /* DEVICE PAGEDEF DEFAULT       */
//          FORMDEF=SECURE,           /* DEVICE FORMDEF DEFAULT       */
//          CHARS=GT20,               /* DEVICE DEFAULT FONT SET      */
//          PIMSG=YES,                /* ACCUMULATE DATA SET MESSAGES */
//          DATACK=UNBLOCK,           /* UNBLOCK DATA CHECKS          */
//          TRACE=NO                  /* BUILD INTERNAL TRACE         */
//PRT10     ENDCNTL
```

Figure 57 (Part 2 of 2). Example Printer Start-up Procedure

11.2.7 Assembling Sample Module for Separator Pages

If you wish to implement random numbering of separator pages, IBM supplies sample source in SYS1.SAMPLIB. Members APSUX01S and APSUX02S must be assemble and link-edited. These exits must replace the APSUX01 and APSUX02 exits shipped with the product.

11.2.8 Activating Print Labeling

RACF uses the PSFMPL class to activate certain print labeling functions. You may also define two profiles to control who may use the DPAGELBL and SYSAREA keyword in JCL and hence override the printer defaults:

```
PSFMPL.DPAGELBL
PSFMPL.SYSAREA
```

To authorize a user to override identification labeling and UPA enforcement, you allow them READ authority to the DPAGELBL and SYSAREA profiles, as above. The following commands show how print labeling can be activated, and how to permit USER5 to override the printer defaults:

```
RDEFINE   PSFMPL   PSFMPL.DPAGELBL   UACC(NONE)
RDEFINE   PSFMPL   PSFMPL.SYSAREA    UACC(NONE)
PERMIT    PSFMPL.DPAGELBL   CL(PSFMPL)   ACC(READ)   ID(USER5)
PERMIT    PSFMPL.SYSAREA    CL(PSFMPL)   ACC(READ)   ID(USER5)
SETROPTS  CLASSACT(PSFMPL)
```

This example assumes security labels have already been defined, and the SECLABEL class is active. If the SECLABEL class is inactive, identification labeling does not work as expected, since security labels are not propagated. Unless there is an exit in place as described in 11.3.3, “Changing the Propagated SECLABEL” on page 203, the SECLABEL class MUST be active.

11.3 Implications for Print Labeling

This section includes discussions and further considerations for implementing print labeling functions. The discussion is grouped into areas relating to the UPA and identification labels, and the PSFMPL class.

11.3.1 Identification Labels

When considering the UPA and identification labels, note the following:

- Identification labels can be printed anywhere on an output page. The security overlays that make up the identification label may be placed both inside and outside the UPA.
- You should consider carefully putting overlays inside and outside the UPA. Combining an overlay with the print data would prevent easy removal of the identification label by trimming the paper. However, placing a security overlay within the UPA could result in the user overwriting the label. For these reasons you may consider using a combination of overlays, both inside and outside the UPA, to define your identification label.
- Identification labeling can be enabled for user data pages only, or for system pages only, or for both. What elements of your print output are labeled is defined by the use of the SPAGELBL and DPAGELBL parameters. See 11.2.2, “UPA Definition” on page 195 for an example of what constitutes the user pages, and what constitutes the system pages.
- You should consider your existing page design before enabling identification labels. Existing separator pages may have to be re-designed because they impinge on the identification label data. Similarly you have to examine the form and page definitions use for the system pages. For example, you should check the FORMDEFs and PAGEDEFs used for the JESJCL data set so as not to overlay the identification label.
- Trailer pages can be suppressed with an operator command. This can cause a problem with PSF/NVS controlled printers. Resources loaded into a printer are deleted either:

  After the job trailer has printed
  By the PSF/MVS resource deletion exit

  If the trailer pages are being suppressed by an operator command, and resources are not being deleted by the PSF/MVS exit, there may be a potential exposure since resources loaded continue to accumulate. This could result in a printer performance problem due to the printer storage becoming full.
- If a printer with identification labeling enabled selects a spool data set for printing, and the spool data set has no security label associated with it, PSF/MVS issues the following error message:

```
APS532I APSWPR10 FSS35 *** PRT10 (B11) A SECURITY
(CONT.) DEFINITIONS RESOURCE WITH MEMBER NAME(ABC) WAS NOT FOUND -
(CONT.) RETURN CODE ¢0004¢X, REASON CODE ¢00000000¢X RETURNED FROM
(CONT.) SYSTEM (BLDL) FUNCTION.
```

  The message indicates that the security definitions library member (ABC) could not be found, and the spool data set is placed on the held queue. If you are in the process of migrating to an environment with security labels, you may have a mixed environment where some users have SECLABELs and some do not. In this situation, you may have errors when output data sets without SECLABELS are sent to a printer with identification labeling enforced. See 11.3.3, “Changing the Propagated SECLABEL” on page 203 for some exits that assign default security labels to spool data sets.

11.3.2 PSFMPL Resource Class

Note the following:

- For generation of random number labeling of header and trailer pages, the PSFMPL class must be active.
- Auditing of suppression of system defaults for SYSAREA and DPAGELBL is activated with PSFMPL.
- Identification labeling and UPA enforcement do not require PSFMPL to be active.

11.3.3 Changing the Propagated SECLABEL

There are two situations where you may wish to change the security label associated with a spool data set. The first is when output without a SECLABEL may be selected by a printer with identification labeling enabled. This could occur when you are in a transition phase implementing security labels. Here you could have some users with SECLABELs and some without. In this situation, the SECLABEL class would be activated.

The second situation is when you may wish to perform print label functions without activating the SECLABEL class. Normally, to use identification labeling, you would have to implement SECLABELs. SECLABELs have far-reaching effects on a system, and any implementation would generally occur over an extended period. You may wish to implement identification labeling more quickly, or alternatively, you may not wish to implement security labels generally at all.

Regardless of your reasons for not activating the SECLABEL class, if this class is not activated you must assign a SECLABEL to a spool data set. If the class is not active, SECLABELs are not propagated, and since PSF/MVS does not check whether the SECLABEL class is active before attempting identification labeling, errors occur. You can use a modification of the exits below to add a security label to a spool data set. You could perhaps add SECLABELs based on the job name, or perhaps the userid of the spool data set owner.

Note: The security label is not propagated even if you specify the SECLABEL= parameter in a JOBCARD.

11.3.4 JES2 Exit

JES2 Exit 23 can be used to change the security label of an output group. See Appendix G, “JES2 Exit for Assigning a Default SECLABEL to Output” on page 285 for the source code of the exit.

This exit is to modify the Job Separator Page Area (JSPA). The exit is invoked in the FSA address space from the HASPFSSM module. This exit is invoked only once for FSS printers.

The JSPA contains a field for the security label of the output group. Ideally, we would change this field to provide our security label; unfortunately, this field is over-written by HASPFSSM after the exit returns control. The security label field is written from a copy of the Characteristic Job Output Element area (CHAR-JOE) inside the Job Information Block (JIB). In the sample exit, we check for a null SECLABEL in the CHAR-JOE. If we find the field contains X′00′, we insert our default security label. In this case we used the label ′DEFLABEL′.

Note: The label you add must be a defined security label. If you add a label that is not defined to RACF, the job is held. No messages are issued.

11.3.4.1 JES3 Exit

JES3 User Exit 45 can be used to change the propagated security label. See Appendix H, “JES3 Exit to Assign a Default SECLABEL to Output” on page 289 for the source code of this exit.

The security label in the JSPA can be changed without the information being over-written after returning from the exit. This exit tests whether the SECLABEL field in the JSPA is filled with X′00′ characters. If it is, the default security label is assigned.

11.4 Auditing with PSF/MVS

PSF/MVS produces SMF type 6 records for auditing purposes. PSF/MVS 1.3.0 includes additions to the record. An SMF record is produced for each data set in an output group that is printed. New security related fields in the record include:

- Indicator flagging JCL keywords in effect (that is the values of SYSAREA, SPAGELBL, and DPAGELBL in effect for the output being processed, not necessarily the system defaults).
- Indicator of a successful print operation.
- Indicator of a printer hardware error.
- Indicator of job header successfully printed.
- Indicator of job trailer successfully printed.
- Indicator that security label integrity is guaranteed. (indicates a guaranteed printer was used).
- Indicator that DPAGELBL defaults were suppressed.
- Indicator that SYSAREA defaults were suppressed.
- Identifier of the number of security overlays used.
- Identifier of the number of security fonts used.
- Identifier of the number of security page segments used.
- FORMDEF used to print the data set.
- PAGEDEF used to print the data set.

In addition to the above, the record also contains:

- The step name and procedure step name of the job/session that created the data set.
- The DDNAME that was use to create the data set.
- The userid associated with the job/session that created the data set.
- The security label of the created data set.
- The processing mode of the data set.
- The name of the data set being printed.

MVS/ESA SPL: System Management Facilities, GC28-1819 indicates there is a place reserved in the record for the output user security token, but this field is not generally used.

If the PSFMPL class is not active, the SMF records do not indicate whether the system defaults for SYSAREA and DPAGELBL have been suppressed. All other security related fields are updated.

Chapter 12. SDSF Release 3

SDSF Release 3 allows you to use the system authorization facility (SAF) as an alternative to ISFPARMS. If you decide to use SAF security, you need MVS 3.1.3 and JES2 3.1.3 or higher and a security product that is functionally equivalent to RACF 1.9.

Although you can accomplish security through SAF, you still need an ISFPARMS module that contains the ISFPMAC, ISFGRP, and ISFTR macros for nonsecurity functions.

You can use your existing ISFPARMS module for backup security when SAF delivers an indeterminate response (RC=4). In those cases where no appropriate profile is defined in the active resource classes, SDSF will use the ISFPARMS authorization checking for security. Exceptions are the JESSPOOL and WRITER classes. If no profiles exist when they are active, then access to resources is denied regardless of the entries in the ISFPARMS module. SAF is always checked first for security authorization.

Using the SAF interface with SDSF protects the following SDSF resources:

- SDSF panels and authorized commands
- Ability to issue MVS and JES2 commands from the command line
- Overtypeable fields
- Jobs and output affected by action characters and overtypeable fields
- Initiators
- Printers
- Destination names
- Spool data sets for browsing and viewing
- Generated MVS and JES2 commands.

SAF security provides a dynamic means of authorizing SDSF users to issue commands and process job output. Once the user has invoked SDSF, SAF authorization dynamically affects the next user interaction. Only authorization changes to the ″destination operator profiles″ for JESSPOOL resources and destination name resources require an SDSF session to be ended and then restarted.

The following general steps are necessary to provide security through SAF:

- Define profiles to protect the resources in the SDSF, WRITER, OPERCMDS, and JESSPOOL classes. Generic or discrete profile names may be used. It is assumed that the EGN option is active in your system. Define broad resource protection using generics first, and then more restrictive resource profiles can be defined later. UACC authority can also be used to grant access to users.
- Allow users to access recources in the above classes by defining the necessary access levels in the appropriate profiles.
- Activate the OPERCMDS, SDSF, JESSPOOL, and WRITER classes. It is recommended that you use the RACLIST option for these classes and that you have the EGN option active. See the following commands:

```
SETROPTS  CLASSACT(SDSF  WRITER  OPERCMDS  JESSPOOL)
SETROPTS  GENERIC(SDSF  WRITER  OPERCMDS  JESSPOOL)
SETROPTS  GENCMD(SDSF  WRITER  OPERCMDS  JESSPOOL)
SETROPTS  RACLIST(SDSF  WRITER  OPERCMDS  JESSPOOL)
```

For a new SDSF customer, the *System Display and Search Facility Guide and Reference* describes an excellent method for installing SAF security checking. If you are an existing SDSF customer, you should review ISFPARMS and decide what you want to protect with SAF. For more detail on how to migrate to SAF checking, see the section 12.2, “SDSF Migration” on page 213.

12.1 Protecting SDSF Resources

Many SDSF functions require a profile in more than one resource class to protect them. For example, to protect the overtyping of fields, you must define at least three profiles: one for the fields (SDSF class), one for the MVS/JES2 commands that are generated (OPERCMDS class), and others for the object of the overtypeable field (WRITER, JESSPOOL, SDSF, or OPERCMDS class). The following examples of SDSF tasks show the several resources that may be required to protect them:

- Command line commands (/)
  1. SDSF class to protect commands on the command line
  2. OPERCMDS class to protect the MVS/JES2 commands
- Overtypeable fields
  1. SDSF class to protect the overtypeable fields
  2. OPERCMDS class to protect the MVS/JES2 commands that are generated by overtyping the fields
  3. Object of overtyping
     - SDSF class for initiators
     - WRITER class for printers
     - JESSPOOL class for jobs, output groups
- Action characters
  1. OPERCMDS class to protect the MVS/JES2 commands that are generated by overtypeable fields and action characters
  2. Object of action characters
     - SDSF class for initiators
     - WRITER class for printers
     - JESSPOOL class for jobs, output groups, SYSIN/SYSOUT data sets

As an example, consider that a user, DEVL2, wishes to release and print a held SYSOUT data set, created by USER1:

1. Allow user DEVL2 to issue the ″O″ line command on the H panel by defining:

   ```
   PERMIT ISFATTR.** CLASS(SDSF) ID(DEVL2) ACCESS(UPDATE)
   ```

2. The ″O″ line command generates the JES2 command, $O Jnnnnn, which is protected by the profile, JESx.RELEASE.BATOUT in the OPERCMDS class. Allow user DEVL2 to issue the command by defining:

   ```
   PERMIT JES2.RELEASE.BATOUT CLASS(OPERCMDS) ID(DEVL2) ACCESS(UPDATE)
   ```

3. User DEVL2 also needs authority in the JESSPOOL class, to be able to alter the status of a spool data set owned by USER1. In the JESSPOOL class define:

   ```
   PERMIT C2JES2.USER1.** CLASS(JESSPOOL) ID(DEVL2) ACCESS(ALTER)
   ```

4. To print the SYSOUT data set on PRINTER2, user DEVL2 needs access authority to this printer. This is accomplished by defining the following profile in the WRITER class:

```
PERMIT JES2.LOCAL.PRT2 CLASS(WRITER) ID(DEVL2) ACCESS(READ)
```

The *System Display and Search Facility Guide and Reference* gives a complete list of the SDSF tasks and the resources needed to protect them.

12.1.1 SDSF Authorized Commands

SDSF authorized commands are protected by defining resource names in the SDSF class. These commands can be protected by using discrete or generic profiles. The entries on the SDSF main menu depend on the access authority the user has for the SDSF authorized commands. READ access to the resources is required to be able to issue these commands.

The SDSF authorized commands are those that can appear on the AUTH= parameter in ISFPARMS, with the exception of OWNER, which can be protected only through SAF. If no SAF protection exists for the OWNER command, then all users can issue this command.

The DEST command is treated like any other SDSF command, but you can further protect the destination names with the DEST command. See 12.1.4, “Destination Names” on page 209.

The following examples show profiles for protecting SDSF authorized commands:

- To protect all commands and grant access to user SYSPRG1, specify:

```
RDEFINE SDSF ISFCMD.** UACC(NONE)
PERMIT ISFCMD.** CLASS(SDSF) ID(SYSPRG1) ACCESS(READ)
```

- To allow access only to the DA, H, I, O, and ST panels, define the following profile:

```
RDEFINE SDSF ISFCMD.DSP.** UACC(READ)
```

- To allow no access to the DA panel, define the following discrete profile:

```
RDEFINE SDSF ISFCMD.DSP.ACTIVE.JESx UACC(NONE)
```

For a complete list of the SDSF resource class profiles for the SDSF authorized commands see Appendix I, “SDSF Resource Names Tables” on page 295.

12.1.2 Command Line Commands (/)

The ability to issue MVS and JES2 commands on the command line by using the slash (/) is protected by the resource profile ISFOPER.SYSTEM in the SDSF class. A user must have READ authority to ISFOPER.SYSTEM.

SAF checking occurs on the user's authority to issue commands from the command line, but not on the command itself. Command checking on the command is done in the OPERCMDS class after SDSF authorizes use of the command line. For instance, you can specify the following to allow the members of group OPER to issue commands from the command line:

```
RDEFINE SDSF ISFOPER.SYSTEM UACC(NONE)
PERMIT ISFOPER.SYSTEM CLASS(SDSF) ID(OPER) ACCESS(READ)
```

Note: The conditional access checking, WHEN(CONSOLE(SDSF)), does not apply to commands issued from the command line. For more information on the WHEN option, see 12.1.3, "Overtypeable Fields" on page 208.

12.1.3 Overtypeable Fields

There are three parts to protecting most overtypeable fields:

1. The user must be permitted to overtype on a designated overtypeable field. If the user is authorized, the field is conditioned for overtyping. If the user is not authorized, the field is displayed on the panel, but input is not allowed. A user can be granted access by defining the correct level of authority in the resource profile beginning with ISFATTR in the SDSF class.

   UPDATE authority is required to the resource in order to overtype the fields. As an example, after defining a profile to protect all overtypeable fields, permit USER2 to overtype as follows:

   ```
   RDEFINE SDSF ISFATTR.** UACC(NONE)
   PERMIT ISFATTR.** CLASS(SDSF) ID(USER2) ACCESS(UPDATE)
   ```

2. The user must be permitted to use the objects of the overtypeable fields, such as initiators, jobs, output groups, and printers in the SDSF, WRITER, and JESSPOOL classes.

   Levels of authority needed are:

   - CONTROL authority for initiators in the SDSF class.
   - CONTROL authority for printers in the WRITER class. Note that the ″C″ action character requires ALTER authority.
   - ALTER authority for JESSPOOL class resources.

3. Finally, the user must be permitted in the OPERCMDS class to issue the MVS and JES2 commands that are generated by overtyping the field. For the required access authority in the OPERCMDS class, see Appendix I, "SDSF Resource Names Tables" and Table 37 on page 296 and Table 38 on page 299.

Users can be allowed to issue MVS or JES2 commands at all times or they can be restricted to issue these commands only while using SDSF. To permit users to issue all JES2 commands unconditionally, define:

```
RDEFINE OPERCMDS JESx.** UACC(NONE)
PERMIT JESx.** CLASS(OPERCMDS) ID(groupid | userid) ACCESS(CONTROL)
```

To allow users to issue JES2 commands only while running under SDSF, define the following profiles and activate the CONSOLE class:

```
RDEFINE CONSOLE SDSF UACC(NONE)
PERMIT SDSF CLASS(CONSOLE) ID(groupid | userid) ACCESS(READ)
SETROPTS CLASSACT(CONSOLE)

RDEFINE OPERCMDS JESx.** UACC(NONE)
PERMIT JESx.** CLASS(OPERCMDS) ID(*) ACCESS(CONTROL) WHEN(CONSOLE(SDSF))
```

For a complete list of the OPERCMDS resource name associated with the action characters, see Appendix I, "SDSF Resource Names Tables" and Table 39 on page 302 and Table 40 on page 303.

12.1.4 Destination Names

The DEST command is treated like any other SDSF authorized command. The profile, ISFCMD.FILTER.DEST, in the SDSF class can be defined to protect the DEST command. A user requiring authorization to all destination names on the DEST command must be granted READ access to the resource ISFCMD.FILTER.DEST and READ access to the resource ISFOPER.ANYDEST.JESx in the SDSF class.

Those users who are to be restricted in the use of destination names should not be permitted access the ISFOPER.ANYDEST.JESx resource.

Use of the IDEST parameter in ISFPARMS causes SDSF to initialize the panels to only those jobs and output groups having destination names listed in the ISFNTBL macro. For information on how to code the ISFNTBL macro entry refer to *System Display and Search Facility Guide and Reference* . SDSF ignores any names that are invalid (not defined to the active JES2 subsystem) or to which the user does not have SAF authorization.

When a user is denied access to the ISFOPER.ANYDEST.JESx resource:

- A list of initial values must be specified for that user through the IDEST parameter on the ISFGRP macro.
- ISFAUTH.DEST.destname resource profiles must be defined for those destinations.
- That user must be given READ authority in the above mentioned profiles to access these destinations.

Note: If you use SAF security, a user who has no IDEST list in ISFPARMS must be permitted access to the resource ISFOPER.ANYDEST.JESx. Otherwise, the destination filter is set to the character string ???????? , and no jobs or output groups appear on the display panels.

The following examples demonstrate destination names control.

- To allow USER1 access to all destination names, enter:

```
RDEFINE  SDSF  ISFOPER.ANYDEST.JES2  UACC(NONE)
PERMIT  ISFOPER.ANYDEST.JES2  CLASS(SDSF)  ID(USER1)  ACCESS(READ)
```

- RMT1 is specified in the list for the IDEST parameter on the ISFGRP macro for USER2. To give USER2 access to the destination name RMT1, define:

```
RDEFINE  SDSF  ISFAUTH.DEST.RMT1  UACC(NONE)
PERMIT  ISFAUTH.DEST.RMT1  CLASS(SDSF)  ID(USER2)  ACCESS(READ)
```

Note: Changes to authorization for destination names on the DEST command take effect during the current SDSF session. Changes to authorization for destination names affect what is displayed on the SDSF panels only after the user begins a new SDSF session.

For additional information see 12.2.5, “Destination Control with SDSF and SAF” on page 220.

12.1.5 Initiators

The initiators are protected in the SDSF class by the resource name of ISFINIT.Ixx.JESx, where xx is the initiator identifier and JESx is the targeted JES2 subsystem. Authority to access the job will not be checked.

To allow users access to the initiators, the following authority is needed:

- READ access in the initiator profile is required for the user to be able to display information about an initiator.
- All other overtypeable fields and action characters require CONTROL access in the initiator profile.

For example, to allow the initiators to be controlled only by the operators in group OPER:

```
RDEFINE SDSF ISFINIT.** UACC(NONE)
PERMIT ISFINIT.** CLASS(SDSF) ID(OPER) ACCESS(CONTROL)
```

To protect the commands issued against initiators, see Appendix I, “SDSF Resource Names Tables” and Table 38 on page 299 and Table 39 on page 302.

12.1.6 Printers

To protect the printers, define, in the WRITER class, profiles with the resource name of JESx.LOCAL.devicename (for local printers and punches), and JESx.RJE.devicename (for RJE devices). Authority to access the job on the printer is not checked. Also, UPDATE authority to the appropriate profiles starting with ISFATTR in the SDSF class is required.

The following access authority in the WRITER class is required to control the printers:

- READ authority to the printer is required to allow the user to display information about the printer.
- ALTER authority to the printer is required for purging output.
- All other action characters and overtypeable fields require CONTROL authority to the printer.

To protect the commands issued to printers, see Appendix I, “SDSF Resource Names Tables” and Table 38 on page 299 and Table 39 on page 302.

In the following example, all printers are protected and only members of the OPER group can control them:

```
RDEFINE SDSF ISFATTR.** UACC(NONE)
PERMIT ISFATTR.** CLASS(SDSF) ID(OPER) ACCESS(UPDATE)

RDEFINE WRITER JES2.** UACC(NONE)
PERMIT JES2.** CLASS(WRITER) ID(OPER) ACCESS(ALTER)
```

Users can be conditionally permitted to access the WRITER class resources, so that they can access printers only while they are running SDSF. To achieve this, issue the following command:

```
PERMIT JES2.** CLASS(WRITER) ID(OPER4) ACCESS(ALTER) WHEN(CONSOLE(SDSF))
```

OPER4 can now control the printers only from his SDSF session. It is assumed here that the CONSOLE resource class is active and that OPER4 has READ authority to the SDSF profile in this class.

12.1.7 Jobs, Output Groups, and SYSIN/SYSOUT Data Sets

The types of objects that can be protected with the JESSPOOL class are jobs, output groups, and SYSIN/SYSOUT data sets. These objects are on the following panels:

- Jobs are found on the I, ST, and DA panels.
- Output groups are found on the O and H panels.
- SYSIN/SYSOUT data sets are found on the Job Data Set panel or any panel where you browse with the S or V action character.

At the moment, JES2 uses the JESSPOOL class to protect only SYSIN/SYSOUT data sets. SDSF has extended the use of the JESSPOOL class to protect SDSF jobs and output groups as well. Protection for each type of resource can de defined separately.

If you do not want to make a distinction between the types of resources, you can allow access to all for a user, as is demonstrated by the following example:

```
RDEFINE JESSPOOL C2JES2.USER1.** UACC(NONE)
PERMIT C2JES2.USER1.** CLASS(JESSPOOL) ID(DEVL2) ACCESS(ALTER)
```

In this case user DEVL2 has authority to all spool resources from USER1 for all action characters and overtypes. Typically, DEVL1 needs authority in other resource classes as well for action characters and overtypeable fields.

The resource profile names in the JESSPOOL class are:

```
Jobs          :  nodeid.userid.jobname.jobid
Output Groups :  nodeid.userid.jobname.jobid.GROUP.ogroupid
SYSIN/SYSOUT  :  nodeid.userid.jobname.jobid.Ddsid.dsname
```

Where:

nodeid The NJE node ID of the JES2 subsystem.

userid The ″local″ user ID of the job owner.

jobname The name of the job.

jobid The job number JES2 assigned to the job, prefixed by the type of object (TSU, JOB, or STC).

GROUP The character string GROUP.

ogroupid The output group name as specified through the GRPID= keyword on the MVS //OUTPUT statement.

Ddsid Unique data set number JES2 assigned to the spool data set, prefixed by a D.

dsname The user-specified or system-assigned data set name.

Users can permit others to select their jobs, output groups, and SYSIN/SYSOUT data sets using the ″S″ action character. The ″S″ action character, does not automatically authorize the user to access all SYSIN/SYSOUT data sets within a job or output group when the user is authorized to access the job

or output group itself. Only those SYSIN/SYSOUT data sets to which the user has at least READ authority are displayed.

The following examples demonstrate the use of the different JESSPOOL profiles used for authorization while running under SDSF:

- Protect all jobs for userid USER1 on node C6JES2, and permit USER2 to access the resource unconditionally:

```
RDEFINE JESSPOOL C6JES2.USER1.*.* UACC(NONE)
PERMIT C6JES2.USER1.*.* CLASS(JESSPOOL) ID(USER2) ACCESS(ALTER)
```

- Permit only USER99 to access all output groups for userid DEVL1:

```
RDEFINE JESSPOOL C6JES2.DEVL1.*.*.GROUP.* UACC(NONE)
PERMIT C6JES2.DEVL1.*.*.GROUP.* CLASS(JESSPOOL) ID(USER99) ACCESS(ALTER)
```

- Protect all SYSIN/SYSOUT data sets for jobs starting with APD on node C2JES2 and allow USER2 to browse these spool data sets:

```
RDEFINE JESSPOOL C2JES2.*.APD*.*.D*.* UACC(NONE)
PERMIT C2JES2.*.APD*.*.D*.* CLASS(JESSPOOL) ID(USER2) ACCESS(READ)
```

For additional information on protecting spool data sets, see Chapter 7, “SYSIN / SYSOUT - JES Spool” on page 95.

12.1.8 Operator Authorization to Access JESSPOOL Resources

Users may be given operator authority by destination to jobs, output groups, and SYSIN/SYSOUT data sets. This allows the operator to access the spool data sets for a particular destination without explicitly being authorized through JESSPOOL profiles.

This authority is the equivalent of the ISFPARMS CMDAUTH=DEST and DSPAUTH=ADEST authority. For example, to allow OPER1 to control all spool data sets for destinations RMT30 through RMT39, you define the following profiles in the SDSF class:

```
RDEFINE SDSF ISFOPER.DEST.JES% UACC(NONE)
RDEFINE SDSF ISFAUTH.DEST.RMT3% UACC(NONE)
RDEFINE SDSF ISFAUTH.DEST.RMT3%.DATASET.* UACC(NONE)

PERMIT ISFOPER.DEST.JES% CLASS(SDSF) ID(OPER1) ACCESS(READ)
PERMIT ISFAUTH.DEST.RMT3% CLASS(SDSF) ID(OPER1) ACCESS(ALTER)
PERMIT ISFAUTH.DEST.RMT3%.DATASET.* CLASS(SDSF) ID(OPER1) ACCESS(ALTER)
```

If either of the above authorizations is not given, OPER1 needs access to the individual job or data set defined in the JESSPOOL class to be able to control the spool data set.

Specifying the qualifier JES% in the above profiles allows access to the resources for all active JES2 subsystems.

When SECLABEL checking is active, a user must be logged on with the appropriate SECLABEL in order to access the JESSPOOL resources, even if the user has operator authorization.

Note: When the user is authorized to access the SDSF resource ISFOPER.DEST.JESx, but is not authorized to access the particular ISFAUTH.DEST.destname or ISFAUTH.DEST.destname.DATASET.dsname resource, then CDMAUTH=DEST and DSPAUTH=ADEST in ISFPARMS is bypassed for authorization checking when a fallback to ISFPARMS occurs.

If SAF denies operator authority to a user, ISFPARMS does not override that decision.

12.1.9 Auditing SDSF SAF Requests

Accesses to a resource are logged according to the audit setting in the resource profile. All security checks are logged except SAF checks for the following SDSF class resources:

- ISFOPER.DEST.JESx
- ISFAUTH.DEST.destname
- ISFAUTH.DEST.destname.DATASET.dsname
- ISFOPER.ANYDEST.JESx
- All resources beginning with ISFATTR

The user is not specifically trying to gain access to any of the above resources, and therefore logging is not performed for these security calls.

12.2 SDSF Migration

Before starting the migration to SAF, you should review your current ISFPARMS entries, and decide what ISFGRP parameters you want to protect with the corresponding SAF resource. Remember that you still need an ISFPARMS module for the nonsecurity parameters on ISFGRP entries.

When SAF cannot make a security decision, ISFPARMS is used to determine authorization. SAF returns an indeterminate result when the resource class is inactive or no profiles are defined to protect a resource. You should retain your fully coded ISFPARMS module so that it can serve as backup to the SAF security checking until you have completed migration to SAF.

SAF denies access to resources protected by the JESSPOOL and WRITER classes, if no profiles are defined for the resources in these classes. There is no fallback to ISFPARMS for these two classes.

It is recommended that you migrate one group at the time, starting with the group that has the highest authority in the ISFPARMS module. This approach will enable you to limit the changes that have to be made for each conversion step.

The migration in this section is based on the three groups as they appear in the current default ISFPARMS module. The groups are:

- System Programmers, who have ACCT, OPER, and JCL authority.
- Operators, who have OPER and JCL authority
- End users, who have JCL authority

Using this method, you start by defining very broad generic profiles for all resources used by SDSF. More restrictive profiles are then defined for the operator group to limit their authority. And finally, even more restrictive profiles are defined for the end-user group.

Note: System programmers must have access to those RACF resources defined for their own group, the operator group, and the end-user group. The operators need access to those resources defined for the operator group and the end-user group. The end users need access to the resources defined for their group.

The ISFPARMS module is changed each time a group has been migrated successfully.

12.2.1 System Programmer Group

The system programmer group is selected first for migration. This group has access authority to all profiles in order to retain access to all resources used by SDSF. One member of the team performing the migration task must have system-SPECIAL in RACF, so any incorrect definitions in the RACF classes can be changed by this person dynamically. If the security policy in an installation does not allow one member of the migration team to have system-SPECIAL, the CLAUTH authority should be granted for the resource classes OPERCMDS, SDSF, WRITER, and JESSPOOL.

The current ISFGRP entry in the ISFPARMS module for the system programmer group is:

```
 ****************************************************************
 * GROUP1                                                       *
 *         SYSTEM PROGRAMMERS                                   *
 *                                                 Non-SAF      *
 ****************************************************************

        ISFGRP TSOAUTH=(JCL,OPER,ACCT),                          X
              AUTH=(LOG,I,O,H,DA,DEST,PREF,SYSID,ABEND,ACTION,   X
              INPUT,FINDLIM,ST,INIT,PR,TRACE),                   X
              CMDAUTH=(ALL),                                     X
              CMDLEV=7,                                          X
              ILOGCOL=25,                                        X
              DSPAUTH=(ALL),                                     X
              DFIELD2=DAFLD2,                                    X
              GPLEN=2,                                           X
              ACTION=ALL,                                        X
              DADFLT=(IN,OUT,TRANS,STC,TSU,JOB)
```

First the RACF classes OPERCMDS, WRITER, SDSF, and JESSPOOL are activated, as follows:

```
SETROPTS  CLASSACT(SDSF  WRITER  OPERCMDS  JESSPOOL)
SETROPTS  GENERIC(SDSF  WRITER  OPERCMDS  JESSPOOL)
SETROPTS  GENCMD(SDSF  WRITER  OPERCMDS  JESSPOOL)
```

Now that you have activated the JESSPOOL class, and you have not defined any profiles in it yet, users can only process their own spool data sets.

If you have groups of users processing each other's spool data sets, you must at this point define all necessary profiles in the JESSPOOL class, before you activate this class. For more information on JESSPOOL profile definitions, see section Chapter 7, "SYSIN / SYSOUT - JES Spool" on page 95.

Activating the WRITER class causes authorization check failures unless you specify profiles for the print devices in your system. To allow a minimum of interuption, define:

```
RDEFINE  WRITER  JES*.**  UACC(READ)
PERMIT  JES*.**  CLASS(WRITER)  ID(SYSPRG OPER)  ACCESS(ALTER)
```

The following additional profiles that have to be defined in the SDSF class to allow the system programmer group to continue with the same access authority as before:

```
RDEFINE  SDSF  ISF*.**  UACC(NONE)

PERMIT  ISF*.**  CLASS(SDSF)  ID(SYSPRG)  ACCESS(ALTER)
PERMIT  ISF*.**  CLASS(SDSF)  ID(OPER)  ACCESS(CONTROL)
PERMIT  ISF*.**  CLASS(SDSF)  ID(USER)  ACCESS(UPDATE)
```

To grant the same access authority in the OPERCMDS class to the system programmers and the operators as before, define the following profiles:

```
RDEFINE  OPERCMDS MVS.**  UACC(NONE)
RDEFINE  OPERCMDS JES2.**  UACC(NONE)

PERMIT  MVS.**  CLASS(OPERCMDS)  ID(SYSPRG)  ACCESS(ALTER)
PERMIT  JES2.**  CLASS(OPERCMDS)  ID(SYSPRG)  ACCESS(ALTER)
PERMIT  MVS.**  CLASS(OPERCMDS)  ID(OPER)  ACCESS(CONTROL)
PERMIT  JES2.**  CLASS(OPERCMDS)  ID(OPER)  ACCESS(CONTROL)
```

The end-user group generates MVS/JES2 commands by overtyping fields on the I, O, H, DA, and ST panels. To allow these commands to execute, define the following profiles in the OPERCMDS class and grant the necessary access authority:

```
RDEFINE  OPERCMDS  JES2.MO*.*  UACC(NONE)
RDEFINE  OPERCMDS  JES2.CA*.*  UACC(NONE)
RDEFINE  OPERCMDS  JES2.RE*.*  UACC(NONE)

PERMIT  JES2.MO*.*  CL(OPERCMDS)  ID(SYSPRG)  ACCESS(ALTER)
PERMIT  JES2.MO*.*  CL(OPERCMDS)  ID(USER)  ACCESS(UPDATE)
PERMIT  JES2.MO*.*  CL(OPERCMDS)  ID(OPER)  ACCESS(CONTROL)

PERMIT  JES2.CA*.*  CL(OPERCMDS)  ID(SYSPRG) ACCESS(ALTER)
PERMIT  JES2.CA*.*  CL(OPERCMDS)  ID(OPER USER) ACCESS(UPDATE)

PERMIT  JES2.RE*.*  CL(OPERCMDS)  ID(SYSPRG) ACCESS(ALTER)
PERMIT  JES2.RE*.*  CL(OPERCMDS)  ID(USER) ACCESS(UPDATE)
PERMIT  JES2.RE*.*  CL(OPERCMDS)  ID(OPER) ACCESS(CONTROL)
```

Now that you have defined all necessary profiles and activated the relevant classes, you can change the ISFGRP entry for the system programmer group. The following entries are sufficient:

```
*****************************************************************
* GROUP1                                                        *
*         SYSTEM PROGRAMMERS                                    *
*                                                  SAF          *
*****************************************************************

        ISFGRP TSOAUTH=(JCL,OPER,ACCT),                          X
              DFIELD2=DAFLD2,                                    X
              ILOGCOL=25,                                        X
              GPLEN=2,                                           X
              ACTION=ALL,                                        X
              DADFLT=(IN,OUT,TRANS,STC,TSU,JOB)
```

Note: For more information on the ISFGRP parameters, see *System Display and Search Facility Guide and Reference*.

12.2.2 Operator Group

The next group to convert is the Operator group. This approach keeps the migration effort still within the Data Center, and the number of profiles to be defined to RACF is limited for this group.

The current entry for the Operator group in the ISFPARMS module is:

```
*****************************************************************
* GROUP2                                                        *
*          SAMPLE OPERATOR GROUP ENTRY                          *
*                                                 Non-SAF       *
*****************************************************************

        ISFGRP TSOAUTH=(JCL,OPER),                                X
              AUTH=(LOG,I,O,H,DA,PREF,DEST,SYSID,ACTION,          X
              FINDLIM,ST,INIT,PR,INPUT),                          X
              CMDAUTH=(ALL),                                      X
              CMDLEV=7,                                           X
              DSPAUTH=(GROUP,NOTIFY,AMSG),                        X
              ILOGCOL=25,                                         X
              GPLEN=2,                                            X
              ACTION=ALL,                                         X
              DADFLT=(IN,OUT,TRANS,STC,TSU,JOB)
```

The operator group is allowed to perform all authorized commands except the ABEND and TRACE commands. Therefore, the following profiles are defined in the SDSF class:

```
RDEFINE  SDSF  ISFCMD.DSP.**  UACC(READ)
RDEFINE  SDSF  ISFCMD.ODSP.**  UACC(NONE)
RDEFINE  SDSF  ISFCMD.FILTER.ACTION  UACC(NONE)
RDEFINE  SDSF  ISFCMD.FILTER.FINDLIM  UACC(NONE)
RDEFINE  SDSF  ISFCMD.FILTER.DEST  UACC(NONE)
RDEFINE  SDSF  ISFCMD.FILTER.PREFIX  UACC(NONE)
RDEFINE  SDSF  ISFCMD.FILTER.SYSID  UACC(NONE)
RDEFINE  SDSF  ISFCMD.FILTER.OWNER  UACC(NONE)
RDEFINE  SDSF  ISFCMD.FILTER.INPUT  UACC(NONE)
RDEFINE  SDSF  ISFCMD.MAINT.*  UACC(NONE)

PERMIT  ISFCMD.ODSP.**  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.FILTER.ACTION  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.FILTER.FINDLIM  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.FILTER.PREFIX  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.FILTER.DEST  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.FILTER.SYSID  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.FILTER.OWNER  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.FILTER.INPUT  CL(SDSF)  ID(OPER  SYSPRG)  ACCESS(READ)
PERMIT  ISFCMD.MAINT.*  CL(SDSF)  ID(SYSPRG)  ACCESS(READ)
```

The OWNER command is new in SDSF Release 3 and has no equivalent entry in the AUTH= field of the ISFGRP macro. It allows users to limit jobs and SYSOUT, displayed on the SDSF panels, to the owning user IDs of those spool data sets.

To allow the operators to issue command line commands (/) the following profile has to be specified:

```
RDEFINE  SDSF  ISFOPER.SYSTEM  UACC(NONE)
PERMIT  ISFOPER.SYSTEM  CL(SDSF)  ID(OPER)  ACCESS(READ)
PERMIT  ISFOPER.SYSTEM  CL(SDSF)  ID(SYSPRG)  ACCESS(ALTER)
```

In the OPERCMDS class you must give the operator group authority to all MVS and JES2 commands that are generated by action characters and overtypeable fields. Profiles to grant this access authority were defined when the system programmer group was migrated. In your installation you may wish to limit access authority to the MVS and JES2 commands for the operator group by defining more specific profiles and access rules.

You can also permit some operators conditionally to certain commands while they are using SDSF. You must have the CONSOLE class active, the SDSF console defined in this class, and the operator(s) authorized to the SDSF console with READ access. For example, to permit OPER1 and OPER2 to JES2 and MVS commands only while using SDSF:

```
PERMIT JES2.** CLASS(OPERCMDS) ID(OPER1 OPER2) ACCESS(CONTROL) +
       WHEN(CONSOLE(SDSF))
PERMIT MVS.** CLASS(OPERCMDS) ID(OPER1 OPER2) ACCESS(CONTROL) +
       WHEN(CONSOLE(SDSF))
```

Note: Conditional checking is done only for action characters and overtypes. The command line commands (/) go directly through OPERCMDS checking; SDSF does not do any checking or changes to the TOKEN for them.

To allow the operators to control destinations, define the following profile:

```
RDEFINE SDSF ISFOPER.ANYDEST.JES2 UACC(NONE)
PERMIT ISFOPER.ANYDEST.JES2 CL(SDSF) ID(OPER) ACCESS(READ)
PERMIT ISFOPER.ANYDEST.JES2 CL(SDSF) ID(SYSPRG) ACCESS(READ)
```

For access authority to the action characters and the overtypeable fields on the SDSF panels the operator group will need the profiles:

```
RDEFINE SDSF ISFATTR.** UACC(NONE)
PERMIT ISFATTR.** CL(SDSF) ID(OPER SYSPRG USER) ACCESS(UPDATE)

RDEFFINE JESSPOOL *.*.*.* UACC(NONE)
RDEFINE JESSPOOL *.*.*.*.GROUP.* UACC(NONE)
PERMIT *.*.*.* CLASS(JESSPOOL) ID(OPER) ACCESS(ALTER)
PERMIT *.*.*.*.GROUP.* CLASS(JESSPOOL) ID(OPER) ACCESS(ALTER)
```

The first JESSPOOL profile relates to the Input Queue, Status, and DA panels. The second profile relates to the Held Output and Output Queue panels. These JESSPOOL profiles do not allow the operators to view the jobs and output groups.

The DSPAUTH=AMSG authority in the ISFGRP macro can be given with the JESSPOOL profiles:

```
RDEFINE JESSPOOL *.*.*.*.D*.JES* UACC(NONE)

PERMIT *.*.*.*.D*.JES* CL(SDSF) ID(OPER) ACCESS(READ)
```

Note: Instead of defining the above profile in the JESSPOOL class, you may use the destination operator interface for global access to JESSPOOL resources. For additional information, see section 12.1.8, “Operator Authorization to Access JESSPOOL Resources” on page 212.

The initiators are protected in the SDSF class by the resource name **ISFINIT.Ixx.JESx**, where the **xx** is the initiator identifier. Authority to the job will not be checked. To permit the operator group to control the initiators, define the following:

```
RDEFINE SDSF ISFINIT.** UACC(NONE)
PERMIT ISFINIT.** CL(SDSF) ID(OPER SYSPRG) ACCESS(CONTROL)
```

The resource profiles to protect the printers must be defined in the WRITER class. Authority to the job on the printer is not checked. The operator group will need ALTER authority to be able to purge output (”C” line command) on the printer.

To protect all printers:

```
RDEFINE  WRITER  JES2.**  UACC(NONE)
PERMIT  JES2.**  CLASS(WRITER)  ID(SYSPRG  OPER)  ACCESS(ALTER)
```

Note: This profile was defined earlier during the migration of the system programmer group.

The WHEN(CONSOLE(SDSF)) option can be used on resource profiles in the WRITER class to restrict printer control to the SDSF PR panel only.

The end users have to be permitted to the ISFOPER.ANYDEST.JES2 profile with an access of READ until their group is migrated. With authority to all job and output destinations, the end users are able to display their jobs and output on the SDSF panels.

You can now change the ISFPARMS entry for the operator group to the following:

```
*********************************************************************
* GROUP2                                                            *
*          SAMPLE OPERATOR GROUP ENTRY                              *
*                                                 SAF               *
*********************************************************************

         ISFGRP TSOAUTH=(JCL,OPER),                                  X
               ILOGCOL=25,                                           X
               GPLEN=2,                                              X
               ACTION=ALL,                                           X
               DADFLT=(IN,OUT,TRANS,STC,TSU,JOB)
```

12.2.3 End-user Group

The last group to migrate is the end-user group. In your installation, you may have more than one end-user group, but for simplicity, all end users are represented by one group in the ISFPARMS module.

The current entry for the end user group in ISFPARMS is:

```
*********************************************************************
* GROUP3                                                            *
*          SAMPLE ENDUSER GROUP ENTRY                               *
*                                               Non-SAF             *
*********************************************************************

         ISFGRP TSOAUTH=(JCL),                                       X
               AUTH=(I,O,H,DA,ST),                                   X
               CMDAUTH=(GROUP,NOTIFY),                               X
               CMDLEV=2,AUPDT=10,                                    X
               DSPAUTH=(GROUP,NOTIFY),                               X
               ILOGCOL=25,                                           X
               GPLEN=4,                                              X
               ACTION=(11,12,USER),                                  X
               DADFLT=(IN,OUT,TRANS,STC,TSU,JOB)
```

The end user group can display the I, O, H, DA, and ST panels from the SDSF primary menu. The jobs and output groups displayed on the panels belong to the group. Depending on the entries in the JESSPOOL class, each user can view individually owned spool data sets, and some or all of the group's spool data sets.

All overtypeable fields can be altered, spool data sets can be deleted or requeued, but jobs cannot be released for execution. To permit the end-user group the same access, using the SAF security interface, the following profiles are defined:

```
RDEFINE  SDSF  ISFATTR.OUTPUT.PRTY  UACC(NONE)
PERMIT  ISFATTR.OUTPUT.PRTY  CL(SDSF)  ID(SYSPRG  OPER)  ACCESS(ALTER)

RDEFINE  OPERCMDS  JES2.DISPLAY.BA*  UACC(READ)
RDEFINE  OPERCMDS  JES2.DISPLAY.TS*  UACC(READ)
RDEFINE  OPERCMDS  JES2.DISPLAY.ST*  UACC(READ)
RDEFINE  OPERCMDS  JES2.MSEND.CMD  UACC(READ)
RDEFINE  OPERCMDS  JES2.MODIFY.%%%OUT  UACC(CONTROL)      where %%% can be:
RDEFINE  OPERCMDS  JES2.RELEASE.%%%OUT  UACC(UPDATE)       BAT, STC, or TSU
RDEFINE  OPERCMDS  JES2.CANCEL.*  UACC(UPDATE)
RDEFINE  OPERCMDS  JES2.CANCEL.DEV  UACC(NONE)
RDEFINE  OPERCMDS  JES2.RESTART.BAT  UACC(CONTROL)
RDEFINE  OPERCMDS  JES2.MODIFYHOLD.*  UACC(UPDATE)
RDEFINE  OPERCMDS  JES2.ROUTE.JOBOUT  UACC(UPDATE)

PERMIT  JES2.DISPLAY.BA*  CL(OPERCMDS)  ID(SYSPRG  OPER)  ACCESS(ALTER)
PERMIT  JES2.DISPLAY.TS*  CL(OPERCMDS)  ID(SYSPRG  OPER)  ACCESS(ALTER)
PERMIT  JES2.DISPLAY.ST*  CL(OPERCMDS)  ID(SYSPRG  OPER)  ACCESS(ALTER)
PERMIT  JES2.MSEND.CMD  CL(OPERCMDS)  ID(SYSPRG  OPER)  ACCESS(ALTER)
PERMIT  JES2.MODIFY.%%%OUT  CL(OPERCMDS)  ID(OPER  SYSPRG)  ACCESS(ALTER)
PERMIT  JES2.RELEASE.%%%OUT  CL(OPERCMDS)  ID(SYSPRG  OPER) ACESS(ALTER)
PERMIT  JES2.CANCEL.*  CL(OPERCMDS)  ID(SYSPRG  OPER)  ACCESS(ALTER)
PERMIT  JES2.CANCEL.DEV  CL(OPERCMDS)  ID(SYSPRG  OPER) ACCESS(ALTER)
PERMIT  JES2.RESTART.DEV  CL(OPERCMDS)  ID(SYSPRG)  ACCESS(ALTER)
PERMIT  JES2.MODIFYHOLD.*  CL(OPERCMDS)  ID(SYSPRG)  ACCESS(ALTER)
PERMIT  JES2.ROUTE.JOBOUT  CL(OPERCMDS)  ID(SYSPRG OPER)  ACESS(ALTER)
```

Notes:

- The end-user group is not allowed to change the priority of the output on the O panel.
- No PERMITs are defined for the end users, as they need access the resources in the OPERCMDS class only at the universal access level.

To permit the end users to release jobs for execution, the profile ″JESx.MODIFYRELEASE.**type**″ has to be specified in the OPERCMDS class. **Type** can be BAT, STC, or TSU. The end user needs UPDATE access to this profile.

The entry in ISFPARMS is changed to the SAF entry.

```
*****************************************************************
* GROUP3                                                        *
*         SAMPLE ENDUSER GROUP ENTRY                            *
*                                                  SAF          *
*****************************************************************

        ISFGRP TSOAUTH=(JCL),                                    X
              AUPDT=10,                                          X
              ILOGCOL=25,                                        X
              GPLEN=4,                                           X
              ACTION=(11,12,USER),                               X
              DADFLT=(IN,OUT,TRANS,STC,TSU,JOB)
```

The end users, operators, and system programmers are all now able to perform the same SDSF functions as they could with only the ISFPARMS module and no SAF security checking.

12.2.4 Considerations

Before you activate the JESSPOOL or WRITER class, it is important to define all necessary profiles and access rules in the class.

Remember, whenever you define profiles in the SDSF class, you must define the related profiles and access rules in the OPERCMDS, WRITER, or JESSPOOL class at the same time.

Change the ISFGRP macro entries to the SAF entries as soon as you can after migration, if you decided during the migration to keep them. It is easier to look at and debug one set of definitions, in this case the RACF resource class entries, when SDSF returns unpredictable results during processing.

There is no protection for the OWNER command using ISFPARMS. This command can be protected only with SAF. If the command is not protected, all users can use the OWNER command to further restrict the jobs that appear on their displays.

There is no SAF equivalent for CMDAUTH=NOTIFY or DSPAUTH=NOTIFY. To obtain similar functions, a user must have access to the appropriate person's output through the JESSPOOL resource.

When converting command level authorization to SAF, for every CMDLEV you wish to authorize with SAF, you must permit the user to access all corresponding OPERCMDS resources at that CMDLEV and the CMDLEV prior to it. Although you can migrate from ISFPARMS CMDLEV command protection to SAF OPERCMDS command protection in a one-to-one fashion, it is not advised. SAF provides a more flexible means of authorizing users to the various commands.

12.2.5 Destination Control with SDSF and SAF

So far during the migration, no attempt has been made to limit the destinations for which the users can view jobs and output.

During the migration it is important to define the ISFOPER.ANYDEST.JESx profile and give every user READ access to it before defining any ISFAUTH.DEST.destname profiles. Otherwise, unexpected authorization results may occur.

There are two ways in which you can control the display of jobs and output by destination: the SAF security interface and the ISFGRP macro.

12.2.5.1 Using the SAF Security Interface

To limit the viewable destinations available to a user group, the ISFGRP macro must include an IDEST parameter. On the ISFNTBL list connected to the IDEST parameter, you cannot specify more than four destinations. If more than four destinations are specified for the IDEST parameter, you will receive the following message when you access SDSF from TSO or ISPF:

```
ISF005I  INVALID IDEST FOR USER2   TOO MANY DESTS
```

In the following example three destinations are specified in the ISFNTBL macro:

```
DST1   DS    0H
       ISFNTBL LOCAL,1,U9,1,R35,1
```

To control authority to destinations available to a user, define in the SDSF class a profile for these destinations, for instance:

```
RDEFINE  SDSF  ISFAUTH.DEST.LOCAL.**  UACC(NONE)
RDEFINE  SDSF  ISFAUTH.DEST.R35.**  UACC(NONE)
RDEFINE  SDSF  ISFAUTH.DEST.U9.**  UACC(NONE)

PERMIT  ISFAUTH.DEST.LOCAL.**  CL(SDSF)  ID(USER1)  ACCESS(READ)
PERMIT  ISFAUTH.DEST.R35.**  CL(SDSF)  ID(USERB)  ACCESS(READ)
PERMIT  ISFAUTH.DEST.U9.**  CL(SDSF)  ID(USER1 USERB)  ACCESS(READ)
```

This gives USER1 access only to jobs and SYSOUTs with a destination of LOCAL. USERB can view jobs and output only for destination R35. Both users can view output destined for U9, which is PRINTER9 in the JES2 environment.

You may give users operator authority by destination to jobs, output groups, and SYSIN/SYSOUT data sets, through granting READ access to the ISFOPER.DEST.JESx. For more information, see 12.1.8, “Operator Authorization to Access JESSPOOL Resources” on page 212.

The users can specify other destinations outside the scope of their IDEST parameter by overtyping the DEST field on the O or H panel, but when the destination is changed the output cannot be viewed any more. The same applies when a user overtypes the PRTDEST field on the I, ST panel. To prevent this from occurring, the access to the profiles ISFATTR.JOB.PRTDEST and ISFATTR.OUTPUT.DEST in the SDSF class must be changed from UPDATE to READ.

If the installation performs SECLABEL checking, a user must log on with the appropriate SECLABEL, even when the user has the correct access to the ISFAUTH.DEST.** profile, to be able to issue line commands or to change overtypeable fields for jobs and output with a SECLABEL defined.

12.2.5.2 Using the ISFGRP Macro

To allow users in an SDSF group to view jobs and SYSOUT for certain destinations, while not defining profiles in the SDSF class, you must include both the DEST= entry, and the IDEST= entry in the ISFGRP macro. The following shows an example of the ISFGRP macro entries required:

```
       ISFGRP TSOAUTH=(JCL),                          X
             AUPDT=10,                                X
             ILOGCOL=25,                              X
             GPLEN=4,                                 X
             DEST=DST2,                               X
             IDEST=DST2,                              X
             ACTION=(11,12,USER),                     X
             DADFLT=(IN,OUT,TRANS,STC,TSU,JOB)

DST2   DS   0H
       ISFNTBL LOCAL,1,U9,1,R35,1,R36,1
```

The ISFPARMS module is assembled and link-edited. After logon, the user selects the I panel. Jobs for the above destinations are displayed. The H, O, and ST panel show spool data sets only for the above destinations.

Note: A generic profile such as **ISF*.**** in the SDSF class can cause unpredictable results. Any profile starting with **ISFAUTH.DEST** may also cause unexpected results.

To check the setting of the DEST field for your current SDSF session, enter following command:

```
SET DISPLAY ON
```

This is a new command in SDSF Release 3 and you can set it in the ISFGRP macro for a group. The default is set to DISPLAY OFF. If DISPLAY ON is active, the values for the DEST, PREFIX, and OWNER fields are displayed on the I, O, H, PR, INIT, and DA panels.

Chapter 13. Additional Security Implementations

The following security enhancements for MVS/SP 3.1.3, MVS/DFP 3.1.1, ACF/VTAM 3.3, and TSO/E 2.1.1 can be implemented with RACF 1.9:

- Controlling the allocation of devices for graphic, unit record, and teleprocessing devices.
- Protecting LLA commands, access to the PARMLIB members that LLA accesses, and access to LLA-managed data sets.
- Protecting VTAM ACBs from unauthorized OPENs.
- Controlling partner LU 6.2 session establishment between partner LUs.
- Restricting temporary data set access to the owning job and deletion to RACF OPERATIONS users.
- Controlling the use of the Hiperspace* by both the Batch LSR subsystem and the Data Lookaside Facility (DLF).
- Restricting who is allowed to send and receive messages sent with the TSO SEND command.

13.1 Device Allocation Control

With MVS/SP 3.1.3 and RACF 1.9, an installation can control which users are allowed to allocate unit record, teleprocessing, or graphic devices as shown in Figure 58. DASD, tape, and terminal devices are not supported by this function; it applies only to the allocation of unit record, teleprocessing, and graphic devices.

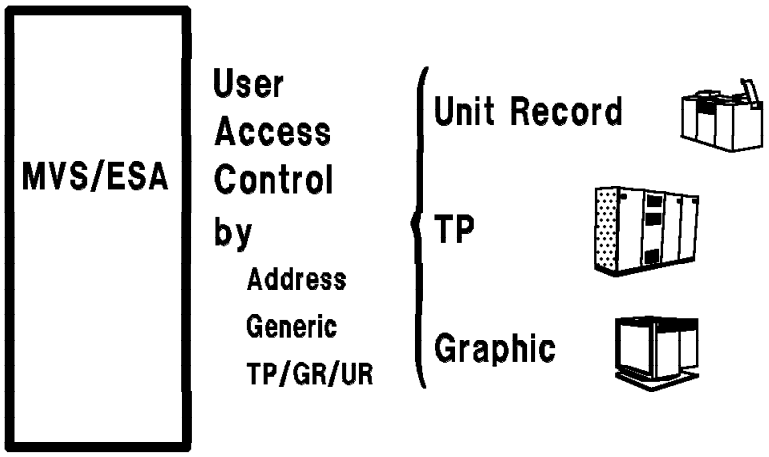

Figure 58. Controlling Devices with RACF 1.9

13.1.1 Implementing Device Allocation Control

Device allocation can be controlled with RACF 1.9 by defining profiles in the DEVICES class. The format of a DEVICES class profile name is:

```
sysid.device-class.unit-name.device-number
```

Where:

sysid The MVS system identifier defined on the SYSNAME keyword in the IEASYSxx member of SYS1.PARMLIB.

device-class There are three classes of devices:

TP	Teleprocessing or communication devices
UR	Unit record devices
GRAPHIC	Graphic devices.

unit-name An esoteric unit name (such as PRINTER1) or a generic name (such as 3800).

device-number A 3-byte field that supplies the address of a specific device. This field is specified on the ADDRESS keyword of the IODEVICE statement.

To control the allocation of devices, the following steps should be taken:

1. Determine which devices to protect. An installation can, for example, protect a specific device with a discrete profile, or several devices with a generic profile. If a device is not protected by a profile, RACF returns a profile-not-found condition to MVS allocation, which allows the user to allocate the device.
2. Get the following information from the MVS system programmer for each device to be protected:
   - Names for the devices to be protected, including the device class, unit name, and device address.
   - RACF-defined users that should be allowed to allocate the device protected by the profile.
3. Define profiles to protect the devices, for example:

```
RDEFINE   DEVICES   SYS5.TP.3705.822          UACC(NONE)

RDEFINE   DEVICES   SYS5.UR.3800.00E          UACC(NONE)

RDEFINE   DEVICES   SYS5.GRAPHIC.3277-2.B1F   UACC(NONE)

RDEFINE   DEVICES   *.UR.**                   UACC(NONE)
```

   Where an access authority of NONE prevents users from allocating the device unless specifically authorized. Generic characters can be used in any qualifier to protect devices in groups rather than individually.

4. Give users the appropriate access to the profile, for example:

```
PERMIT   SYS5.TP.3705.822   ID(USER1)   CLASS(DEVICES)   ACCESS(READ)
```

   Where an access authority of READ allows the user to allocate the device.

5. Activate the DEVICES class. RACLIST processing helps ensure high performance when accessing RACF profiles by loading the profiles into storage. RACLIST is required for the DEVICES class

because the class descriptor table specifies RACLREQ=YES for DEVICES. The following command is used to activate and RACLIST the DEVICES class:

```
SETROPTS   CLASSACT(DEVICES)   RACLIST(DEVICES)
```

Any time a change is made to a DEVICES profile, the in-storage profiles have to be refreshed for the DEVICES class before the changes take effect, for example:

```
SETROPTS   RACLIST(DEVICES)   REFRESH
```

For example, USER1 is given access to device address 822, which is the communications controller on system SYS5, using the following RACF commands:

```
RDEFINE   DEVICES  SYS5.TP.3705.822  UACC(NONE)

PERMIT    SYS5.TP.3705.822  CLASS(DEVICES)  ID(USER1)  ACCESS(READ)
```

USER1 and USER2 both submit the following JCL to allocate the 3705 at device address 822:

```
          //DEVTESTJ JOB (IBM,POK),DEVTEST,MSGCLASS=X
          //*
          //LOAD1   EXEC PGM=IFLOADRN,REGION=512K
          //STEPLIB  DD  DSN=ACFNCP.SSPLIB,DISP=SHR
          //SYSPRINT DD  SYSOUT=*
          //SYSUT1   DD  DSN=ACFNCP.LINK3725,DISP=SHR
          //SYSUT3   DD  DSN=ACFNCP.SSPLIB,DISP=SHR
>>>>>     //U3705    DD  UNIT=822
          //SYSUDUMP DD  SYSOUT=*
          //SYSIN    DD  *
             LOAD         LOADMOD=NCPSRC,3705=U3705,DIAG=NO
          /*
          //*
```

MVS allocation invokes RACF using the RACROUTE REQUEST=AUTH to determine whether the user has the authority to allocate the specified device. If RACF is not installed, the DEVICES class is not active, or a profile is not found, no decision is returned by RACF, and MVS allocation allows the device to be allocated. If the DEVICES class is active, a profile is found for the device, and the user has at least READ access, MVS allows the device to be allocated. If the user does not have at least READ access to the profile protecting the device, MVS does not allow the device to be allocated.

In the example, USER1 receives the following messages indicating that he successfully allocated device 822 to U3705:

```
          IEF236I ALLOC. FOR DEVTESTJ LOAD1
          IEF237I 787  ALLOCATED TO STEPLIB
          IEF237I JES3 ALLOCATED TO SYSPRINT
          IEF237I 787  ALLOCATED TO SYSUT1
          IEF237I 787  ALLOCATED TO SYSUT3
>>>>>     IEF237I 822  ALLOCATED TO U3705
          IEF237I JES3 ALLOCATED TO SYSUDUMP
          IEF237I JES3 ALLOCATED TO SYSIN
          IEF142I DEVTESTJ LOAD1 - STEP WAS EXECUTED - COND CODE 0000

              COMM CTLR SYSTEM SUPPORT UTILITIES --- IFLOADRN
              LOAD         LOADMOD=SC5N822,3705=U3705,DIAG=Y8
              IFL017I LOAD OR DUMP IN PROGRESS ACROSS ANOTHER CHANNEL ADAPTER
              LOADING PROCESS TERMINATED
              IFL001I UTILITY END  04 WAS HIGHEST SEVERITY CODE
```

In contrast, USER2 receives the following messages indicating that he is not allowed to allocate device 822 because the RACF check for the device failed:

```
ICH408I USER(USER2    ) GROUP(P0112    ) NAME(USER TEST ID        )
  SYS5.TP.3705.822 CL(DEVICES )
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
ICH408I USER(USER2    ) GROUP(P0112    ) NAME(USER TEST ID        )
ICH408I   SYS5.TP.3705.822 CL(DEVICES )
ICH408I   INSUFFICIENT ACCESS AUTHORITY
ICH408I   ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )

USER(USER2    ) GROUP(P0112     ) NAME(USER TEST ID        )
  SYS5.TP.3705.822 CL(DEVICES )
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
DEVTESTJ LOAD1 U3705 - DEVICE - 822 CANNOT BE ALLOCATED -
          - IMPROPER AUTHORIZATION
DEVTESTJ LOAD1 - STEP WAS NOT EXECUTED.
```

13.1.2 Recommendations

RACF authorization is done with the unit name that the user specifies at allocation. This requires that profiles be defined for all possible generic and esoteric names to ensure that the resource is protected by RACF. For example, if a 3800 printer is also defined as PRINTER1 and a *.UR.3800.* profile exists, but a *.UR.PRINTER1.* profile does not, a user can be prevented from allocating the device as 3800, but not as PRINTER1.

Trusted procedures such as JES2, JES3, PSF, and VTAM do not need profile authorization to allocate their devices. For those started procedures that are not trusted, the userid specified in the PERMIT command should be the same userid specified in the RACF started procedures table.

13.2 LLA Control

MVS/SP 3.1.3 and RACF 1.9 introduces security checking for LLA. RACF verifies the user's authority to the following resources before processing a START or MODIFY LLA command:

- The START or MODIFY LLA command.
- Each PARMLIB member that LLA accesses.
- Each data set that LLA manages.

When the LLACOPY macro is issued, access to each data set that LLA manages is verified. In all cases, if a profile is found and the user has sufficient authority, LLA processing continues; if the user does not have sufficient authority, LLA processing is terminated. If, however, RACF is not installed or a profile is not found, no decision is returned by RACF, and LLA processing continues. To implement LLA checking, ensure that the appropriate RACF profile information is coded for LLA commands, LLA PARMLIB data sets, and LLA-managed data sets that are to be protected from unauthorized LLA processing.

13.2.1 Implementing LLA Control

The following steps should be taken to implement LLA control:

1. Control those users who can issue the START LLA and MODIFY LLA commands. If the user is authorized to issue the command, LLA is succesfully updated, as indicated by the following message:

   ```
   CSV210I LIBRARY LOOKASIDE UPDATE
   ```

   If the user is not authorized to issue the command, LLA is not updated and a message similar to the following is displayed on the console:

   ```
   ICH408I User(USER2   ) Group(GROUP2 ) Name(SALLY JONES          )
    MVS.MODIFY.JOB.LLA CL(OPERCMDS)
    INSUFFICIENT ACCESS AUTHORITY
    FROM MVS.*.JOB.LLA (G)
    ACCESS INTENT(UPDATE) ACCESS ALLOWED(NONE)
   ```

   The OPERCMDS and CONSOLE classes must be active and profiles must be defined to protect the START and MODIFY LLA commands. Refer to Chapter 10, “Console and Command Security” on page 163 for details.

2. Control access to the LLA PARMLIB data sets with the following steps:

   - Define DATASET class profiles for each LLA PARMLIB data set containing CSVLLAxx members that specify which libraries LLA is to manage and how it is to manage them:

     ```
     ADDSD    data_set_name  UACC(NONE)
     ```

     Where an access authority of NONE prevents users from accessing the CSVLLAxx members in the data set unless specifically authorized.

   - Give users the appropriate access to the profile, for example:

     ```
     PERMIT   data_set_name  ID(userid or group)  ACCESS(READ)
     ```

     Where an access authority of READ allows users to access the CSVLLAxx members in the data set.

   MVS invokes the RACROUTE REQUEST=AUTH macro for each PARMLIB data set whenever a LLA START or MODIFY command is issued. If a profile is found and the user has at least READ access, LLA processing continues; if the user does not have at least READ access, LLA processing is terminated. If, however, RACF is not installed or a profile is not found, no decision is returned by RACF, and LLA processing continues.

3. Control LLA-managed data sets with the following steps:

   - Define DATASET class profiles for all data sets that are defined in all the CSVLLAxx members in SYS1.PARMLIB. For the benefit of the security administrator, all data sets that exist in the active LNKLSTxx libraries should have profiles in RACF:

     ```
     ADDSD   ¢SYS2.**¢    UACC(READ)

     ADDSD   ¢ISF.**¢     UACC(READ)
     ```

   - Define profiles in the FACILITY class to protect each of the LLA managed data sets. These data sets are the libraries specified in the CSVLLAxx and LNKLSTxx members in SYS1.PARMLIB:

     ```
     RDEFINE   FACILITY   CSVLLA.SYS2.**    UACC(NONE)

     RDEFINE   FACILITY   CSVLLA.ISF.**     UACC(NONE)
     ```

Where an access authority of NONE prevents users from accessing the data set for LLA processing unless specifically authorized.

- Give users and groups the appropriate access authority to the FACILITY profile, for example:

```
PERMIT   CSVLLA.SYS2.**   CLASS(FACILITY)   ID(USER1)   ACCESS(UPDATE)
```

  Where an access authority of UPDATE allows users to accessing the data set for LLA processing.

- Activate the FACILITY class if it is inactive or REFRESH the FACILITY class if a RACLIST of the class was already done, using one of the following commands:

```
SETROPTS  CLASSACT(FACILITY)   -or-

SETROPTS  RACLIST(FACILITY)  REFRESH
```

MVS invokes the RACROUTE REQUEST=AUTH macro for the DATASET class profile for each LLA managed data set whenever a LLA START or MODIFY command or LLACOPY macro is issued. If a profile is found and the user has at least UPDATE access, LLA processing continues; if the user does not have at least UPDATE access, LLA invokes the RACROUTE REQUEST=AUTH macro in the FACILITY class. If a profile is found and the user has at least UPDATE access, LLA processing continues; if the user does not have at least UPDATE access, LLA processing is terminated. If, however, RACF is not installed, the FACILITY class is not active, or a profile is not found, no decision is returned by RACF, and LLA processing continues.

13.2.2 Recommendations

The following should be considered when implementing LLA control:

- Ensure that definitions in the DATASET and the FACILITY classes use the same data set name conventions in the profiles. This makes the profiles manageable and also makes it easier to identify resources that are not fully protected. For example, assume the following profiles are defined:

```
ADDSD    ¢SYS2.**¢  UACC(READ)

RDEFINE  FACILITY   CSVLLA.SYS2.*  UACC(NONE)
```

  LLA processing from data set SYS2.TEST.LOADLIB would be allowed because there is no FACILITY class profile to prevent LLA access to the data set.

- Ensure that LLA-managed data sets DATASET class profiles do not allow unauthorized LLA processing. For example, assume the following profiles are defined:

```
ADDSD    ¢SYS2.**¢  UACC(UPDATE)

RDEFINE  FACILITY   CSVLLA.SYS2.**  UACC(NONE)
```

  LLA processing is allowed for any user who has at least UPDATE access to the SYS2.** profile; the FACILITY class profile is never checked. This can also occur with a GLOBAL (DATASET) profile that allows UPDATE or higher access.

13.3 VTAM Controls

With MVS/SP 2.1.0, ACF/VTAM 3.3 and RACF 1.9, VTAM security processing has been enhanced in two areas, shown in Figure 59, and described as follows:

- Control of VTAM applications. When a user opens a VTAM application, if the user is not APF authorized or in a system key, VTAM invokes RACF to determine whether the user is RACF authorized to open the requested VTAM application. The use of VTAM applications by authorized programs is not controlled (or audited) by this mechanism.
- Partner LU 6.2 verification. Partner Logical Units (LU) verification provides a means for controlling session establishment between partner LUs in a VTAM LU 6.2 environment. This support is also referred to as Password-on-Bind or Session-Level security. VTAM performs the authorization to establish the sessions; RACF is used as the repository for the required information and to audit the success or failure of the authorization. A profile is defined for each LU and session keys are assigned to the profiles kept by RACF. For a session to be established between the LUs, both profiles must be defined and have the same key.

  Note: The LU 6.2 documentation uses the term *password*; RACF uses *session key* in order to avoid confusion with the RACF user password. The session key receives none of the support that the user password receives; to RACF, it is like any other field in any general resource profile.

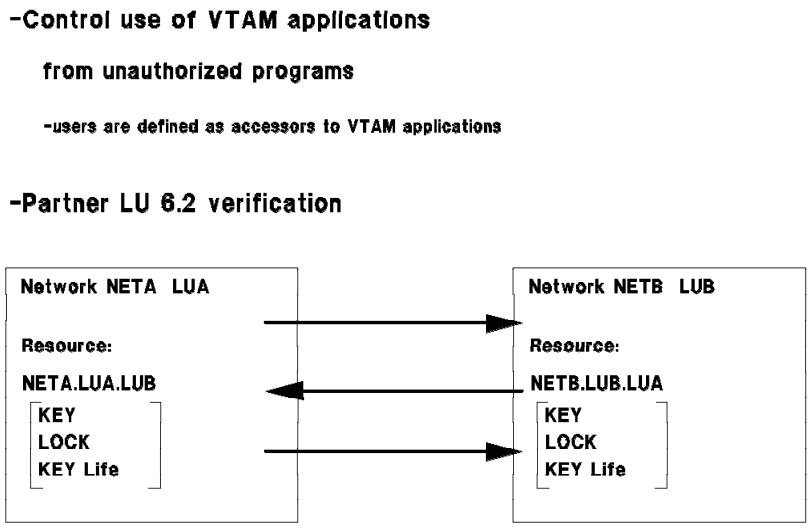

Figure 59. VTAM Controls

13.3.1 VTAM Application Control

Profiles in the VTAMAPPL class allow an installation to specify which users can open an ACB for a VTAM application program from a non-APF authorized program or command processor. The use of VTAM applications by authorized programs is not controlled (or audited) by this mechanism. The VTAMAPPL class profile name is the same name as the ACBNAME parameter on the APPL statement defining the application.

An example of a non-authorized VTAM application program named EC01SRC from IPO1.SAMPLIB can be used to demonstrate RACF 1.9 support. Following are the VTAM and RACF definitions to control access to the EC01 application:

- Define the EC01 application in SYS1.VTAMLST as follows:

```
C5APP VBUILD TYPE=APPL              APPLICATION MAJOR NODE
*
*
A05EC01  APPL  EAS=1,               ESTIMATED CONCURRENT SESSIONS   *
               ACBNAME=EC01,        APPLID FOR ACB                  *
               AUTH=(ACQ)           EC01 CAN ACQUIRE TERMINALS
```

- Define profiles in the VTAMAPPL class to protect the application from all users:

```
RDEFINE   VTAMAPPL   EC01  UACC(NONE)
```

  Where an access authority of NONE prevents users from opening the VTAM application ACB unless specifically authorized.

- Give users or groups the appropriate authority to the application:

```
PERMIT   EC01   CLASS(VTAMAPPL)   ID(USER1)    ACCESS(READ)

PERMIT   EC01   CLASS(VTAMAPPL)   ID(GROUP1)   ACCESS(READ)
```

  Where an access authority of READ allows users to open the VTAM application ACB.

- Activate the VTAMAPPL class. RACLIST processing helps ensure high performance when accessing RACF profiles by loading the profiles into storage. RACLIST is required for the VTAMAPPL class because the class descriptor table specifies RACLREQ=YES for VTAMAPPL. The following command is used to activate and RACLIST the VTAMAPPL class:

```
SETROPTS   CLASSACT(VTAMAPPL)   RACLIST(VTAMAPPL)
```

  Any time a change is made to a VTAMAPPL profile, the in-storage profiles have to be refreshed for the VTAMAPPL class before the changes take effect, for example:

```
SETROPTS   RACLIST(VTAMAPPL)   REFRESH
```

VTAM invokes the RACROUTE REQUEST=AUTH macro for the VTAM application at ACB open time. If the VTAMAPPL class is active, a profile is found for the application, and the user has at least READ access, VTAM allows the user to open the application; if the user does not have at least READ access, VTAM rejects the open request. If, however, RACF is not installed, the VTAMAPPL class is not active, or a profile is not found, no decision is returned by RACF, and VTAM allows the user to open the application.

In the example, unless USER2 belonged to GROUP1, USER2 would receive the following messages indicating that he is not allowed to open the EC01 application:

```
IAT6140 JOB ORIGIN FROM GROUP=ANYLOCAL, DSP=IR , DEVICE=INTRDR  , 000
12:27:46 IAT2000 JOB USER2N   (JOB03912) SELECTED SY1      GRP=A
12:27:46  ICH70001I USER2    LAST ACCESS AT 16:23:58 ON THURSDAY, FEBRUARY 22, 1990
12:27:46  ICH408I USER(USER2   ) GROUP(DEVL    ) NAME(HARRY SMITH         )
12:27:46    EC01 CL(VTAMAPPL)
12:27:46    INSUFFICIENT ACCESS AUTHORITY
12:27:46    ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
12:27:46  IEF450I USER2N   STEP1 RUN1 - ABEND=S000 U1001 REASON=00000000
```

13.3.2 LU 6.2 Partner Verification Control

Profiles in the APPCLU class allow an installation to control type 6.2 session establishment. For more information, see *VTAM Programming for LU 6.2*. LU 6.2 Session-Level LU-LU verification can be used to provide session-level security for distributed LU 6.2 applications. This support is used to verify the identity of an LU to its session partner during activation of LU-LU sessions.

VTAM 3.3 also supports userid verification. With this option, VTAM allows both forms of security defined for LU 6.2 conversations. Based on information negotiated when the session is established, VTAM allows an LU to indicate to its partner LU that the userid and password for a requested conversation have already been verified. Alternatively, VTAM allows an LU to send the userid and password in the request to establish a conversation. This is accomplished by an exchange between two LUs, with each LU using an LU-LU password and the Data Encryption Standard (DES) algorithm. This exchange is called LU-LU verification. LU-LU passwords are established by implementation and installation defined methods outside of SNA. LU-LU passwords are on a partner-LU basis; once the LU-LU password is established between each LU pair, this password is used for all sessions between the pair.

APPCLU profiles are defined for each of the partner LUs on their respective nodes that are allowed to establish a session. In most cases, the nodes of the profiles are different; that is, two different VTAMs running on two different MVS systems. The profiles may even be defined in different RACF databases. The format of an APPCLU profile name is:

```
network-id.local-LU.partner-LU
```

Where:

network-id VTAM network name

local-LU Local logical unit name or *primary LU* on this system.

partner-LU Partner logical unit name or *remote LU* on that system.

In the example in Figure 60, the *primary LU*, LUA, runs under Network-1 as NETA; the *remote-LU*, LUB, runs under Network-2 as NETB.

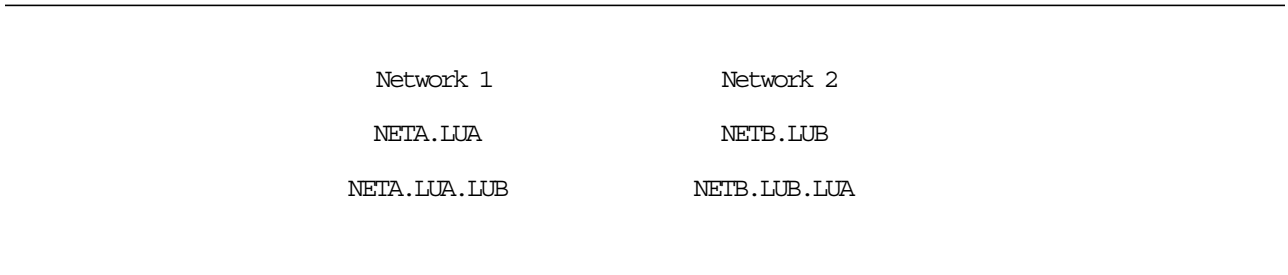

Figure 60. RACF Profiles for LU 6.2 Pairs

Each APPCLU profile contains a SESSION segment with additional profile information that is entered using RACF commands. This information includes:

- The session key, a 16-digit hexadecimal number. VTAM uses this as a password.
- A lock on the use of this session establishment.
- Session interval or a key life; the maximum number of days for which the key is valid. VTAM calls this the password interval.

Note: When VERIFY=REQUIRED is coded in VTAM APPL, as in the example that follows, the session key and session interval are required in the APPCLU profiles. The SETROPTS SESSIONINTERVAL option enables and disables the global limit for the number of days that can elapse before an APPCLU class profile session key must be changed. The range can be from one to 32,767 days.

To implement session establishment control, the following steps should be taken:

- For each LU 6.2 pair, define a profile for each LU. For example:

```
RDEFINE  APPCLU  NETA.LUA.LUB  UACC(NONE)
         SESSION(SESSKEY(X¢0763452A4C331748¢) INTERVAL(30) )

RDEFINE  APPCLU  NETB.LUB.LUA  UACC(NONE)
         SESSION(SESSKEY(X¢0763452A4C331748¢) INTERVAL(30) )
```

- Activate the APPCLU class with the following command:

```
SETROPTS CLASSACT(APPCLU)
```

  Unlike several RACF classes that require the RACLIST option, the CDT entry for the APPCLU class does not allow the RACLIST option to be used. At VTAM initialization, VTAM uses the new FILTER operand on the RACROUTE REQUEST=LIST macro to load only those profiles that are defined for this VTAM node into storage. Since a RACF database can be shared across many systems, this can be an effective way to minimize the use of storage for RACF profiles.

VTAM performs the authorization at BIND time using the information in the SESSION segments of the RACF profiles. The following steps correspond with the steps in Figure 61:

1. Because a session key exists in the NETA.LUA.LUB APPCLU profile on LUA, VTAM on NETA generates some random data, keeps a copy for future use, and sends a copy to NETB in the BIND request.
2. When NETB receives the BIND request with the random data, VTAM on LUB looks for a profile on NETB named NETB.LUB.LUA. If the profile does not exist, NETB fails the request. If the profile does exist, VTAM uses RACF and the session key from the NETB.LUB.LUA profile to encrypt the data received from LUA. VTAM generates some random data of its own, keeps a copy for future use, and sends a copy to NETA in the BIND response along with NETA′s encrypted random data.
3. When NETA receives the BIND response with the encrypted random data, VTAM on NETA looks for a profile on NETA named NETA.LUA.LUB. If the profile does not exist or if the profile does not contain a session key, NETA fails the request. If the profile does exist and it does contain a session key, VTAM uses RACF and the session key from the NETA.LUA.LUB profile to encrypt its original random data and compares it to the encrypted data received in the BIND response. If the data does not match, NETA fails the request. If the data matches, LUA has verified the identity of LUB. NETA then uses RACF and the session key from the NETA.LUA.LUB profile to encrypt NETB′s random data and sends it in a security FM Header(FMH12) to NETB.
4. Upon receiving the FMH12, VTAM on NETB looks for a profile on NETB named NETB.LUB.LUA. If the profile does not exist or if the profile does not contain a session key, NETB fails the request. If the profile does exist and it does contain a session key, VTAM uses RACF and the session key from the NETB.LUB.LUA profile to encrypt its original random data and compares it to the encrypted data received in the FMH12 record. If the data does not match, NETB fails the request. If the data matches, LUB has verified the identity of LUA and partner verification is complete.

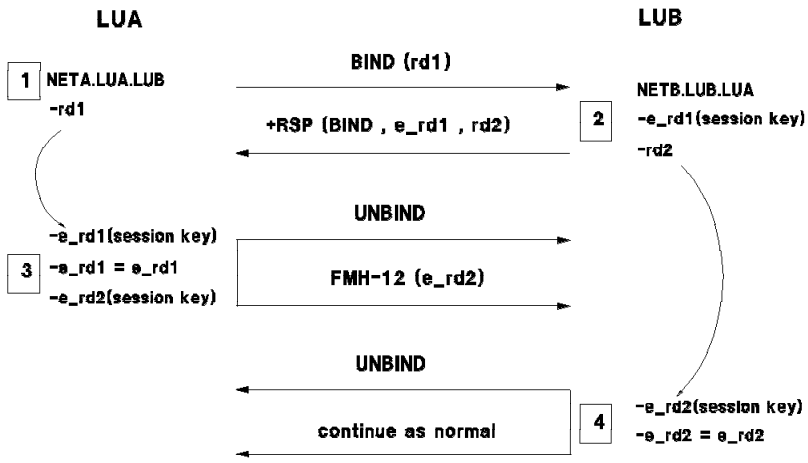

Figure 61. LU 6.2 Session Establishment

After deciding if the session should be established, VTAM audits the authorization for the APPCLU class using the new RACROUTE REQUEST=AUDIT request. For more information, see 4.14.5, “New Audit Controls with RACROUTE Macro” on page 60.

For example, assume a member in SYS1.VTAMLST contains the following application definitions:

```
APPCMJLR VBUILD TYPE=APPL
APPCPGM1 APPL  APPC=YES,                                          *
               AUTH=(ACQ,PASS,SPO),                               *
               .
               .
               .
               LMDENT=19,                                         *
               MODETAB=MTGS3X,                                    *
               PARSESS=YES,                                       *
               VERIFY=REQUIRED   TO PERFORM SESSION LEVEL LU-LU VERF
APPCPGM2 APPL  APPC=YES,                                          *
               AUTH=(ACQ,PASS,SPO),                               *
               .
               .
               .
               LMDENT=19,                                         *
               MODETAB=MTGS3X,                                    *
               PARSESS=YES,                                       *
               VERIFY=REQUIRED   TO PERFORM SESSION LEVEL LU-LU VERF
```

And that the environment for this example is as shown in Figure 62 on page 234.

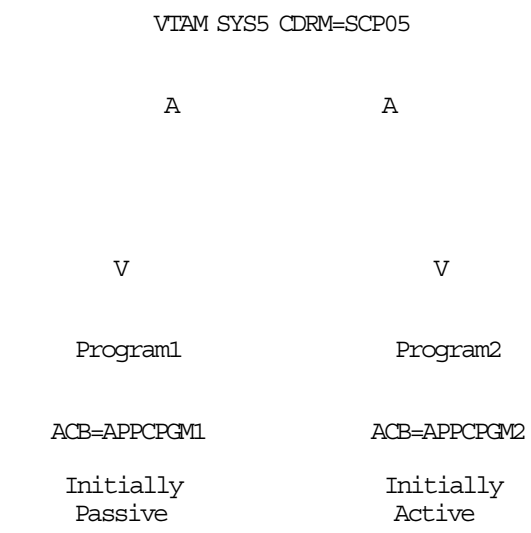

Figure 62. Sample VTAM-to-VTAM Environment

In the following JCL, JOB1 starts the *passive* program and JOB2 starts the program in *passive* mode. The local and partner ACBNAME parameters are passed by the PARM= value. Program1 is started with only one parameter, PARM=APPCPGM1, so it assumes the local ACBNAME and it is *passive*. Program1 assumes all initial contact is by a potential partner and not self initiated. Program2 is started with PARM=(APPCPGM2,APPCPGM1). Program2 assumes that its LU name is APPGPGM2 and it has a partner LU called APPCPGM1 with whom it should attempt a dialog session:

JOB1 to start APPCPGM1:

```
//APPCJOB1 JOB (999,POK),'TEST APPCVTAM',CLASS=A,REGION=4096K,
//    MSGCLASS=T,MSGLEVEL=(1,1),NOTIFY=P0112NZ
//STEP1 EXEC PGM=APPCVTAM,PARM=APPCPGM1
//STEPLIB    DD DSN=P0112NZ.LINKLIB,DISP=SHR
//SYSUDUMP   DD SYSOUT=*
//SYSPRNT    DD SYSOUT=*
//SYSPRNTM   DD SYSOUT=*
//*
```

JOB2 to start APPCPGM2:

```
//APPCJOB2 JOB (999,POK),'TEST APPCVTAM',CLASS=A,REGION=4096K,
//    MSGCLASS=T,MSGLEVEL=(1,1),NOTIFY=P0112NZ
//STEP2 EXEC PGM=APPCVTAM,PARM=(APPCPGM2,APPCPGM1)
//STEPLIB    DD DSN=P0112NZ.LINKLIB,DISP=SHR
//SYSUDUMP   DD SYSOUT=*
//SYSPRNT    DD SYSOUT=*
//SYSPRNTM   DD SYSOUT=*
/*
```

Figure 63 shows the sequence of events as the programs are started. When Program1 is started, it opens its ACB using the LU name passed to it. A SETLOGON OPTCD=START macro is then issued to notify VTAM that the application is ready to accept log-on requests. As this program knows of no other application, it goes to the mainwait state, waiting a list of ECBs. Program2 follows a similar path up to the point when the SETLOGON macro has completed. Because it knows who the partner

application is, it initiates the LU-LU communications. This is started by issuing APPCCMD CONTROL=OPRCNTL,QUALIFY=CNOS. This causes the ATTN exit of program1 to be schedule by VTAM with an event code of CNOS. The output from the example test cases is shown in Appendix J, “Partner LU 6.2 Test Output” on page 305.

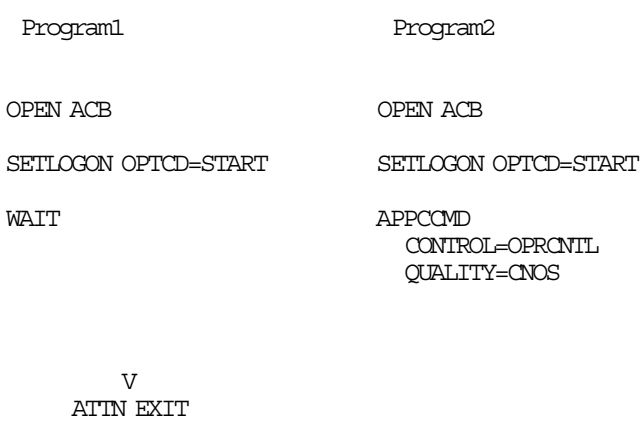

Figure 63. VTAM Macro Flow for Initial Communications

13.3.3 Recommendations

When the APPCLU class is active, ensure that the profiles in the APPCLU class contain the SESSKEY and INTERVAL operands. If these operands are not in the profile, RACF fails the LU-LU session.

13.4 DFP-Managed Temporary Data Set Control

With MVS/SP 3.1.3, MVS/DFP 3.1.1 and RACF 1.9, temporary data set access can be controlled. In an MVS environment, data set names with the following formats are considered DFP-managed temporary data sets:

- Non-VIO data sets:

  SYSyyddd.Tttttttt.RA000.jobname.Rnnnnnnn

- Data sets created by IEHMOVE:

  **SYSUTn.Tnnnnnnn

Temporary data sets have been considered protected from any accesses except by the job or session that created them, and therefore were not protected by RACF. This left no access control for the data set if the job failed, or if the data set was not deleted at the end of the job. It was also not possible to audit accesses to temporary data sets.

MVS/DFP can now use RACF 1.9 to control access to DFP-managed temporary data sets and, depending on the logging options, to generate audit records whenever temporary data sets are created, accessed, or scratched. Only the owning job is allowed to access the data in the temporary data set; users with RACF OPERATIONS can scratch temporary data sets, but cannot access the data.

13.4.1 Implementing Temporary Data Set Control

Temporary data sets are protected by RACF when the TEMPDSN class is activated with the following RACF command:

```
SETROPTS  CLASSACT(TEMPDSN)
```

Profiles cannot be defined for temporary data sets because the class descriptor table entry for TEMPDSN specifies PROFDEF=NO. The class is defined for function enabling and to allow auditing of the authorization checking. Since profiles cannot be defined, the auditor requests logging with a SETROPTS LOGOPTIONS option for the TEMPDSN class. Access failures for the TEMPDSN class are recorded in SMF as DATASET class failures. See 4.14.3, “LOGOPTIONS” on page 57 for details.

Instead of using profiles, access to temporary data sets is controlled by the RACF System Temporary Data Set Table. RACINIT CREATE processing creates the table and chains it off of the address space ACEE. RACDEF builds an entry in the table for each temporary data set that is allocated. When the temporary data set is opened, RACHECK uses the table built by RACDEF to verify that the temporary data set was created within this job before access to the data set is granted. If the data set name is not found in the table, access is not allowed. Since checkpoint restart invokes RACDEF to rebuild the table for restarted jobs, integrity is maintained and access can be granted after a checkpoint/restart operation. RACINIT DELETE processing deletes the table.

A user with the OPERATIONS attribute can scratch any residual DFP-managed temporary data sets remaining on a volume, but cannot access the data sets for any other purpose.

13.4.2 Recommendations

Protect temporary data sets by activating the TEMPDSN class. When the TEMPDSN class is active, the only accesses allowed are as follows:

- The owning job can access its temporary data sets while the job is in execution.
- A user with the OPERATIONS attribute can access temporary data sets for the purpose of scratching them; no other access is allowed.

Do not activate the TEMPDSN class unless all systems sharing the RACF database are at the MVS/DFP 3.1.1 level. When the TEMPDSN class is activated on lower level systems, VIO cannot be used for temporary data sets and the results can be unpredictable.

13.5 Batch Local Shared Resource

With MVS 3.1.0e, the Batch Local Shared Resource (BLSR) function can be used to improve the batch processing environment. Problems currently encountered in an installation may include:

- Running out of batch window time
- Delayed online because of batch reruns
- Virtual storage constraints
- Slow online forward recovery

With the installation of BLSR, programs that process VSAM data sets using Non-Shared Resource (NSR) buffering can be converted automatically to Local Shared Resource (LSR) buffering without modifications; JCL changes are all that is required. This even includes high-level language programs that currently only provide NSR buffering for VSAM data sets. An overview of the BLSR function is shown in Figure 64.

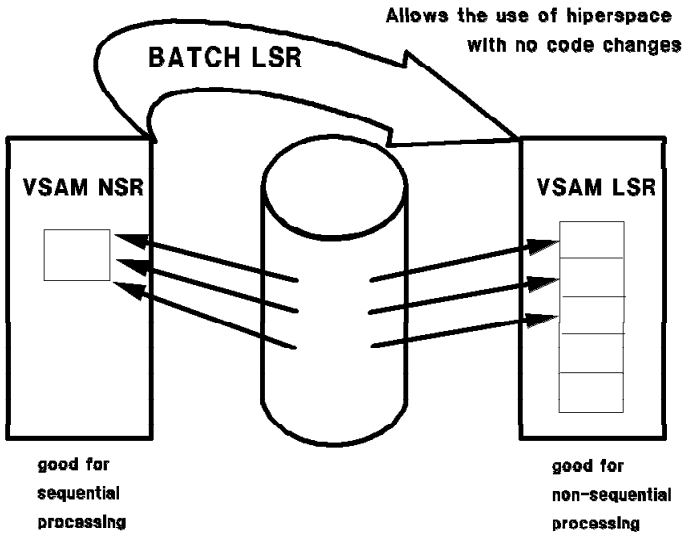

Figure 64. BLSR Function Overview

LSR is an improvement over the NSR buffer management technique. NSR keeps VSAM buffers in storage for as long as the requested records are in the buffer. If a record cannot be found in storage, all the buffers are discarded and a new one obtained, even if the record was in the buffer previous to this. In sequential processing, records are fetched in sequence; NSR is ideally suited for this. In fact, NSR also pre-fetches records in anticipation of this. With LSR, VSAM buffer records stay in storage as long as there are buffers available. If the buffers are full, an LRU algorithm determines which buffer to reuse. However, there is no pre-fetching done, so this technique is ideal for random processing; it should never be used for sequential access. The LSR buffer pool can be built in either central storage or in expanded storage using a Hiperspace. It is also possible with RACF 1.9, to control the use of the Hiperspace as a RACF resource.

13.5.1 Implementing BLSR

To implement BLSR, the following steps should be taken:

1. Define the BLSR subsystem to MVS by adding the following entry to the IEFSSNxx member of SYS1.PARMLIB:

   ```
   ssnm,CSRBISUB
   ```

   Where ssnm is the name of the subsystem. For the remainder of this discussion, the subsystem name is BLSR.

2. Make the following JCL changes to any jobs that are to take advantage of BLSR:

   ```
   Old JCL:       //VSAMMAST DD DSN=VSAM.MASTER.KSDS,DISP=SHR

   Replace with:  //VSAMMAST DD SUBSYS=(BLSR,¢DDNAME=VSAMALT,BUFNI=10,BUFND=10¢,
                  //           ¢HBUFND=100,HBUFNI=100¢
                  //VSAMALT  DD DSN=VSAM.MASTER.KSDS,DISP=SHR
   ```

   Where the SUBSYS= parameters are:

   BLSR The subsystem name specified in IEFSSNxx.

DDNAME Points to the ddname of the real data set to be converted to LSR.

BUFNI The number of buffers in the index buffer pool

BUFND The number of buffers in the data buffer pool

HBUFNI The number of Hiperspace buffers in the index buffer pool. If not specified, no Hiperspace buffer pool is created.

HBUFND The number of Hiperspace buffers in the data buffer pool. If not specified, no Hiperspace buffer pool is created.

This example does not include all possible parameters. For more information and complete list of parameters, refer to *MVS/ESA Batch Local Shared Resources Subsystem*.

3. IPL the system.

At converter time, control is given to the subsystem specified in the DD statement, BLSR. The parameters specified in the JCL are syntax checked and a JCL error occurs if they are specified incorrectly. A Hiperspace request is triggered by the HBUFNI or HBUFND equal to or greater than five. A value of less than five results in a JCL error. If a user does not want to use the Hiperspace, the HUBFNx parameters should be omitted. At OPEN time, BLSR does the following:

- If use of the Hiperspace is requested, calls RACF for authorization. See 13.5.2, "BLSR Hiperspace Control" for details.
- Validates the ACB for applicability; only KSDS and RRDS files are supported by BLSR.
- Changes the ACB from NSR to LSR.
- If authorized, creates a Hiperspace for this data set.
- Modifies the ACB to point to BLSR's I/O interface, so that whenever an I/O is done, BLSR intercepts it and gets the data or moves it into the Hiperspace.

13.5.2 BLSR Hiperspace Control

If the Hiperspace is being used for performance improvement, the use of this resource should be controlled by RACF with the following steps:

1. Define a profile in the FACILITY class to protect the Hiperspace:

```
RDEFINE FACILITY CSR.BLSRHIPR.ssnm UACC(NONE)
```

   Where ssnm is the actual name of the BLSR subsystem and where an access authority of NONE prevents users from using the Hiperspace unless specifically authorized.

2. Give users the appropriate access to the profile, for example:

```
PERMIT CSR.BLSRHIPR.BLSR CLASS(FACILITY) ID(PRODCTL) ACC(READ)
```

   Where an access authority of READ allows the use of the Hiperspace. Now all userids connected to the group PRODCTL can place their buffers in the BLSR Hiperspace.

3. Activate the FACILITY class if it is inactive or refresh the FACILITY class if a RACLIST of the class is already done, using one of the following commands:

```
SETROPTS CLASSACT(FACILITY) -or- SETROPTS RACLIST(FACILITY) REFRESH
```

Figure 65 illustrates BLSR processing with RACF. At converter time, control is given to the subsystem specified in in the DD statement, BLSR.

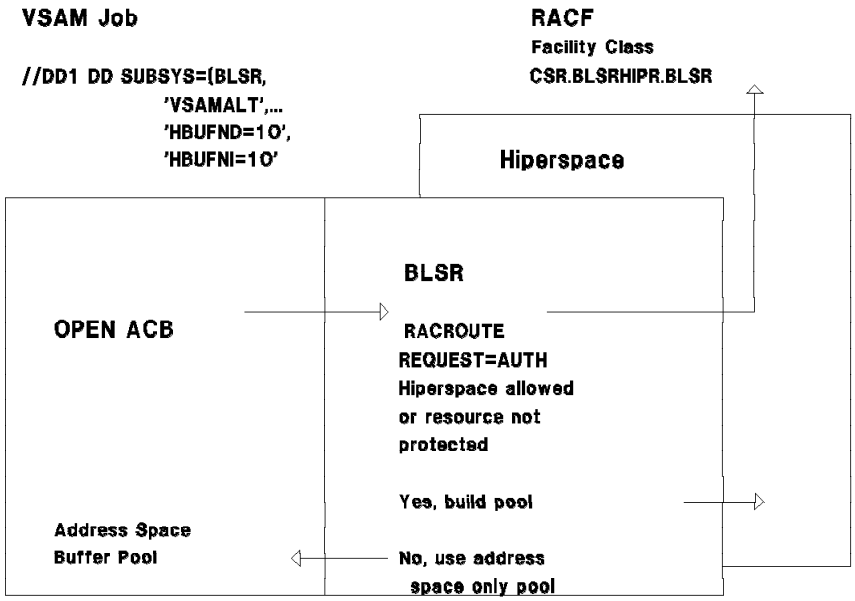

Figure 65. BLSR Hiperspace Control

If either of the two Hiperspace buffer control parameters, HBUFNI or HBUFND, are specified on the DD statement with a minimum value of 5, BLSR issues a RACROUTE REQUEST=AUTH for the profile in the FACILITY class. If the FACILITY class is active, a profile is found for the BLSR subsystem, and the user has at least READ access, BLSR allows VSAM to allocate the LSR buffer pool in the Hiperspace and the following message, indicating a Hiperspace was created, is written to the job log:

```
CSR020I ........ HBUFNI=100, HBUFND=100, ......
```

If the user does not have at least READ access, BLSR does not allow VSAM to allocate the LSR buffer pool in the Hiperspace. In this case, the number of Hiperspace buffers requested are forced to zero and the following messages are written to the job log:

```
CSR006I APPLICATION NOT AUTHORIZED TO USER Hiperspace. DDNAME=VSAMMAST

CSR020I ........ HBUFNI=0, HBUFND=0, .......
```

The job executes, but uses central storage for its LSR buffer pool. If, however, RACF is not installed, the FACILITY class is not active, or a profile is not found, no decision is returned by RACF, and BLSR allows VSAM to allocate the buffer pool in the Hiperspace.

Note: This function controls the use of the Hiperspace, not the use of the BLSR subsystem.

13.5.3 Recommendations

If the buffers are defined in central storage, the region size must be increased to accommodate it. Also, ensure that enough central storage is available, otherwise the additional paging activity may negate the benefits of LSR. If the system has expanded storage, the buffers can be defined in a Hiperspace. If a record is not in central storage, BLSR checks whether it is in expanded storage, and moves it to central storage for VSAM to use. Expanded storage acts as a high-speed DASD cache device. The need for central storage is now significantly reduced. To estimate the size of storage required (expanded storage plus central storage), a Systems Engineering tool, VLBPAA, is available.

Since the default for BLSR is to allow Hiperspace creation, an installation with expanded storage should control BLSR users immediately. Control of BLSR is by userid or groupid only. In order to control its usage, the following points should be considered:

- Only one group is authorized to use the BLSR Hiperspace. All jobs submitted by users connected to this group are eligible to use the BLSR Hiperspace.
- A BLSR Hiperspace can be disabled by a command. It may be necessary to do this to ensure that BLSR Hiperspace is not used during certain shifts. This can be done by releasing a job, by operator intervention, or through COMMNDxx (at IPL time).
- It is a wasted effort to control BLSR expanded storage usage and not control central storage. An installation should limit the region size users can specify by using the IEFUSI exit.
- Users can also be controlled through the ICS/IPS with jobname control and a special PGN for authorized users. The number of users that can run concurrently can be controlled, preventing contention for expanded storage.

13.6 Hiperbatch and the Data Lookaside Facility

Hiperbatch is an extension to the MVS/DFP component that can significantly reduce the execution time of certain batch job streams or multi-step batch jobs that access the same QSAM or VSAM data sets. Two examples of how Hiperbatch can be used are shown in Figure 66. On the left, JOBC reads a data set and causes it to be written into the Hiperspace where JOBA and JOBB can take advantage of the increased performance when they read the data set. On the right, JOB3 creates a data set that is also written into the Hiperspace where JOB1 and JOB2 can take advantage of the increased performance when they read the data set. When a data set is held in the Hiperspace for subsequent jobs, it is referred to in Hiperbatch terms as a *retained* data set.

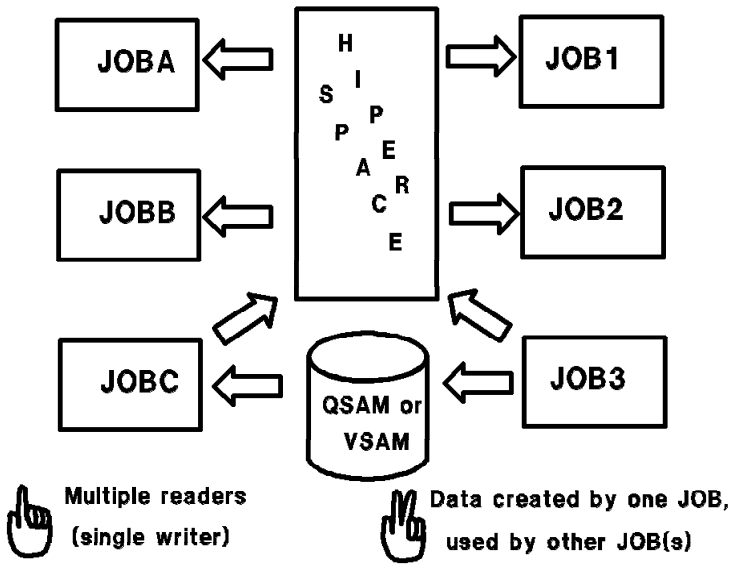

Figure 66. Hiperbatch and the Data Lookaside Facility

The use of Hiperbatch requires MVS/SP 3.1.3, MVS/DFP 3.1.0, expanded storage, a processor with the move-page facility, the data lookaside function (DLF), and a mechanism to determine which QSAM and VSAM data sets are eligible for use by Hiperbatch. DLF is an MVS service that resides in its own address space and is used by Hiperbatch to create and maintain DLF objects, which are a set of Hiperspaces that contain QSAM or VSAM data sets that can be shared by batch jobs. DLF is called by

the QSAM or VSAM access methods when the OPEN or CLOSE macro is issued. Data movement to and from the DLF objects is done by the access method; it is transparent to the user. For more information about Hiperbatch, see *Hiperbatch/DLF Presentation Guide* and *MVS/ESA Application Development Guide: Hiperbatch* .

13.6.1 Implementing Hiperbatch

Implementing Hiperbatch means implementing DLF and identifying which QSAM and VSAM data sets are eligible to be DLF objects. The following steps are required:

1. Choose and implement a mechanism to make the following decisions:

   - Is the data set eligible for DLF?
   - Can the data set be a retained DLF object?
   - Is the user eligible to connect to the DLF object?

   These decisions can be made with profiles defined to RACF 1.9 or with a DLF installation exit routine, or both. Without either an installation exit or RACF 1.9, Hiperbatch cannot decide which QSAM or VSAM data sets are eligible to be DLF objects; the default is that no data sets are eligible to be DLF objects and Hiperbatch is not used. See 13.6.2, “Hiperbatch Hiperspace Control” on page 242 for details.

2. Unless the DLF=nn parameter in IEASYS00 points to some other SYS1.PARMLIB member, modify COFDLF00 to include the following entry:

   ```
   CLASS
     MAXEXPB(0064)
     PCTRETB(25)
     CONEXIT(exitname)
   ```

   Where:

   MAXEXPB The maximum expanded storage allocated to the DLF Hiperspace. In this case, the amount is 64 MB.

   PCTRETB The maximum expanded storage allocated to retained objects. This is expressed as a percentage of MAXEXPB. Here, 25%, or 16 MB, is allocated.

   CONEXIT The name of the DLF installation exit to be called for authorization. If no exit is to be called, the parameter should not be included. APAR OY28154 provides a simple DLF installation exit routine; APAR OY30210 provides a more elaborate implementation.

3. Create a DELOBJ procedure to be used by the operator to delete retained data sets. Create this procedure and place it in a user procedure library:

   ```
   //DELOBJ   PROC  OBJ=¢¢
   //         EXEC  PGM=COFMSTCN,PARM=¢OBJ=&OBJ¢
   ```

   The command to invoke this procedure is:

   ```
   START DELOBJ,OBJ=¢vvvvvvnnnnnn¢
   ```

   Where:

   vvvvvv The DASD volid where the physical data set resides.

   nnnnnn The data set name, with a maximum length of 44 bytes.

4. Since DLF resides in its own address space it must be started either by the operator or by a COMMNDxx member. Start DLF with the following command:

   ```
   START DLF,SUB=MSTR,NN=00
   ```

   Where:

   DLF The name of the DLF procedure distributed with the system in SYS1.PROCLIB.

   SUB SUB=MSTR is required to ensure that DLF continues to run across a JES restart.

   NN The 2-digit suffix of the COFDLFxx member in SYS1.PARMLIB to be used for DLF initialization parameters. In this case, COFDLF00 is used.

5. Verify that DLF is being used with the following commands:

   ```
   D DLF
   F DLF,SB
   ```

   However, for non-retained data, this is helpful only if the program takes some time to finish or goes into a wait since the DLF object is immediately deleted when it is not open for any job.

13.6.2 Hiperbatch Hiperspace Control

With RACF 1.9, the use of Hiperspaces by Hiperbatch can be controlled with profiles defined in the DLFCLASS class. The DLFCLASS profile name is the data set name that represents a data set that is eligible to be processed as a DLF object; the data set itself can also be protected by a profile in the DATASET class. In order to control the use of the Hiperspace by DLF, the following steps should be taken:

- Define profiles for data sets that are eligible to be DLF objects, for example:

  ```
  RDEFINE DLFCLASS dsname UACC(NONE) DLFDATA(RETAIN(YES|NO) JOBNAMES(job1 job2 ...))
  ```

  Where an access authority of NONE prevents users from using the data set as a DLF object unless specifically authorized and operands in the DLFDATA segment are:

  RETAIN Indicates to DLF whether this data set can be a retained DLF object. YES causes the data set to be kept in the Hiperspace until it is specifically deleted; NO causes the data set to be deleted from the Hiperspace as soon as the data set is not OPEN to any job.

  JOBNAMES Specifies the specific jobnames that can use the DLF object. The names can be specified as generic jobnames, such as PAYJOB* to allow PAYJOB1 or PAYJOB2 to connect to the DLF object. Even if JOBNAMES are specified, the user of the job must still have access to the DLF object.

- Give groups or users access to the DLF objects, for example:

  ```
  PERMIT dsname CLASS(DLFCLASS) ID(userid/groupid) ACCESS(READ)
  ```

  Where an access authority of READ allows the user to connect to the DLF object.

- Activate the DLFCLASS class with the following command:

  ```
  SETROPTS CLASSACT(DLFCLASS)
  ```

Whenever a user opens a QSAM or VSAM data set, and DLF is active, DLF attempts to connect the user to the DLF object that corresponds to the data set. On a system with RACF 1.9, as shown in

Figure 67, DLF first calls RACF to verify the user's and the job's authorization to connect to the DLF object.

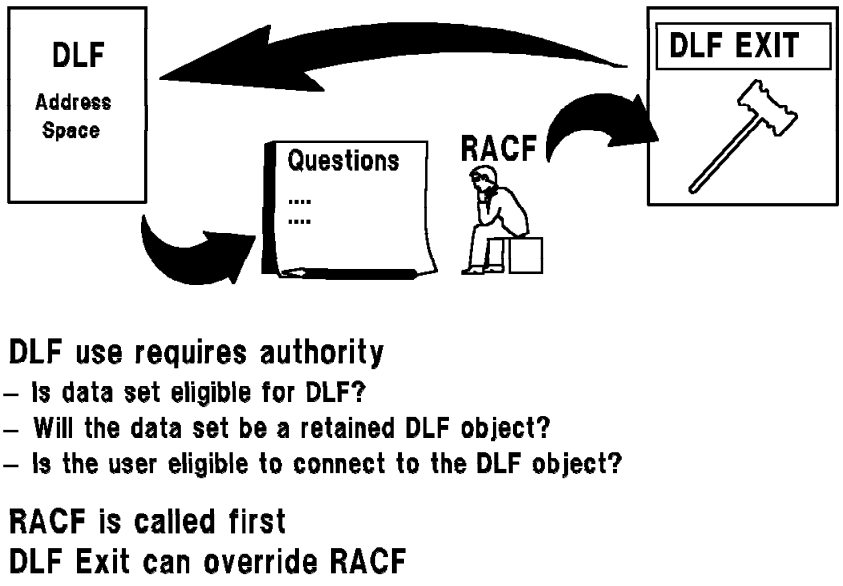

Figure 67. Hiperbatch Control with DLF Exit and RACF 1.9

If there is a DLF exit, DLF uses RACF's decisions to build a parameter list containing the following information to be passed to the DLF installation exit:

- Decisions made by RACF as follows:
  - If a DLFCLASS class is active and a profile is found for this data set name, it indicates that the data set is eligible to be a DLF object. If RACF 1.9 is not installed, the DLFCLASS is not active, or a profile is not found, it indicates that the data set is not eligible to be a DLF object.
  - If the user has at least READ access, it indicates that he is authorized to connect to the DLF object; if the user does not have at least READ access, it indicates that he is not authorized to connect to the DLF object.
  - If there is a JOBNAMES operand in the DLFDATA segment, the jobname is listed in the JOBNAMES operand in the DLFDATA segment, and the user is authorized to connect to the DLF object, it indicates that the jobname is authorized to connect to the DLF object. If there is a JOBNAMES operand in the DLFDATA segment and this jobname is not in the list or the jobname is in the list but the user is not authorized to connect to the DLF object, it indicates that the jobname is not authorized to connect to the DLF object. If there is no JOBNAMES operand in the DLFDATA segment, the jobname is authorized to connect to the DLF object only if the user is authorized to connect to the DLF object; otherwise, it indicates that the jobname is not authorized to connect to the DLF object.
  - If there is a RETAIN operand in the DLFDATA segment, and if the operand is YES, it indicates that the DLF object is to be retained. If there is no RETAIN operand in the DLFDATA segment or if the operand is NO, it indicates that the DLF object is not to be retained.
- The function that is requested. This can be:

Connect A request for DLF to manage this data set into or out of the DLF Hiperspace.

Query Determine the eligibility of a QSAM or VSAM data set to be a DLF object. The DLF exit can allow or disallow the query through the return code.

Disconnect Remove the DLF connection to this file when the user closes the file.

The exit routine can ignore RACF's decision, allow RACF's decision to stand, or use any additional information to override the decision made by RACF. The decision made by the DLF exit is returned to DLF through the exit return code. The two request types for RACF decisions are QUERY and CONNECT. For QUERY requests, return codes are:

0 The data set is eligible for DLF processing.
4 The exit is not making a decision; DLF is to follow RACF authorization.
8 The data set is not eligible for DLF processing.

For CONNECT requests, the return codes are:

0 The user and jobname can connect to the DLF object.
4 The exit is not making a decision; DLF is to follow RACF authorization.
8 DLF is not to permit the user to connect to the DLF object.

When using RACF authorization for access to data sets, always use a return code of 4 for all calls. This tells DLF to use RACFs decision. If there is no DLF installation exit, RACF's decision is final.

If DLF is informed by a non-zero return code from RACF or a return code of 8 from the DLF installation exit, that authorization has failed, Hiperbatch processing is terminated and normal batch processing of the job, including I/Os to the data sets on DASD, takes place. If, however, all the eligibility criteria are met, DLF creates the Hiperspace, creates the DLF object, connects the user, and reads in the data set information as requested by GET and PUT requests. DLF writes statistics to SMF records 14 and 15 for QSAM and record 64 for VSAM.

To demonstrate the DLF control using RACF 1.9 authorizations, consider the following example:

- Assume that the following RACF commands have been issued:

```
RDEFINE   DLFCLASS       *.**.DLFRET.*    UACC(NONE)   DLFDATA(RETAIN(YES))

PERMIT    *.**.DLFRET.*  CLASS(DLFCLASS)  ID(userid/groupid)   ACCESS(READ)

SETROPTS  CLASSACT(DLFCLASS)
```

- Three non-retained objects are created with jobnames JGDLFV2A, JGDLFV2B, and JGDLFV2C.
- Three retained objects are created with jobnames JGDLFV2D, JGDLFV2E, and JGDLFV2F.

The following console display is a result of the DISPLAY DLF command:

```
ISG020I 16.53.21 GRS STATUS 323
S=SYSTEM  SYSVSDO  S00100000 COFGSDO MVSTS3P0112JG.VSAM.DLFRET.COBOL T
SYSNAME       JOBNAME         ASID     TCBADDR   EXC/SHR  OWN/WAIT
SYS5      JGDLFV2D           0018       00AFE168   SHARE     OWN
SYS5      JGDLFV2F           0017       00AFE168   SHARE     OWN
SYS5      JGDLFV2E           001A       00AFE168   SHARE     OWN
S=SYSTEM  SYSVSDO  S00180000 COFGSDO MVSTS3P0112JG.QSAM.DLFNORET.INP T
SYSNAME       JOBNAME         ASID     TCBADDR   EXC/SHR  OWN/WAIT
SYS5      JGDLFV2A           000C       00AFE168   SHARE     OWN
SYS5      JGDLFV2C           0015       00AFE168   SHARE     OWN
SYS5      JGDLFV2B           0019       00AFE168   SHARE     OWN
```

There is no way to distinguish retained from non-retained objects other than by jobname. The data set names as shown are truncated.

The following console display is a result of the MODIFY DLF,STATUS command to display the buffers used for retained and non-retained objects:

```
COF530I DLF STATUS DISPLAY  JG.xx 326
 ESTORE ON-LINE       65536  AVAIL       60344  OK LEVEL         918
        ------------------------------------------------------------
                                    EXIT NAME = COFXRACF
        -------------------------- Maximum ------ Current --- %MAX-
        EXPB (Expanded Buffers)       16384 Blk         24 Blk  <1 %
             (  Non-Retainable)       12288 Blk          0 Blk   0 %
             (  25% Retainable)        4096 Blk         24 Blk  <1 %
        ------------------------------------------------------------
COF536I DLF MODIFY COMMAND PROCESSING COMPLETED.
```

The number of expanded storage buffers used matches that of the COFDLF00 member specified in 13.6.1, “Implementing Hiperbatch” on page 241, (64 MB = 16384 * 4 K, where K=1024). For small usage, a megabyte display does not show usage since the percentage is rounded to zero.

13.6.3 DCB Properties and Retain Options

Whether an object is retained depends not only on the retain option, but also on the type of access specified at open time. For example, an object is

- RETAINED when the user is authorized and the data set is OPEN for sequential output.
- NON-RETAINED when the user is authorized and the data set is OPEN for sequential input.

Table 28 shows what happens if a user violates, or if a program changes, its access mode.

Table 28. Program Access to Data and DLF Exit Calls

PROGRAM ACCESS	DLF PROFILE	RESULT	CONNECT	QUERY
Sequential/Input	Retained	Non-Retained	Yes	No
Sequential/Output	Retained	Retained	Yes	Yes
Sequential/Input	Non-Retained	Non-Retained	Yes	No
Sequential/Output	Non-Retained	Non-Retained	Yes	Yes
Random/Output	N/A	None	No	Yes
Random/Input	N/A	None	??	???

A COBOL program was written to perform this test. The source of that program is listed in Appendix K, “DLF Facility - COBOL Utility Source Listing” on page 311. Results can be summarized as follows:

- Sequential/Input always results in a non-retained data set.
- A QUERY is issued whenever an output is requested irrespective of how the file is accessed.
- For random processing, a DLF object cannot be created, so a DLF profile does not make sense.

13.6.4 Data Integrity

This example demonstrates that synchronization is maintained between the Hiperspace object and the DASD data set. The following environment is established:

- JOB1 reads a VSAM data set whose shareoption is 2.
- A profile exists for the data set with RETAIN(YES).
- JOB2 updates the same data set.
- The DLF exit sets RC=4 for all requests.

When an object is being updated, DLF issues a QUERY to ensure that this data set can be shared by another job, then gets the most recent data. When JOB2 updates the data set, the QUERY request ensures that when the DASD copy is updated, the in-storage copy is also updated, allowing JOB1 access to the most recent data. An integrity exposures exists if a data set is eligible for DLF and then made non-eligible by removing the profile while the data set is being used by JOB1 and JOB2; when the QUERY is done, only the DASD copy is updated. Table 29 shows the steps that cause an integrity exposure.

Table 29. Data Integrity Exposure Steps

STEP	JOB1	JOB2	COMMENTS
1	Reads the data set		A non-retained DLF object is created.
2	WAITs		
3			The DLFCLASS profile for data set is deleted.
4		Updates the data set	Since no profile exists, updates are not reflected in the DLF object.
5		Prints the data set	Data from DASD is printed.
6			Define a DLFCLASS profile for data set.
7		Prints the data set	Data from DLF object is printed.
8		End of job	Data from DLF object is printed.
9	End of job		DLF object is deleted.

The above problem occurs when:

- A VSAM file is defined with SHAREOPTIONS(2,x)
- A DLFCLASS profile is removed or added while the object is in the Hiperspace.
- RACF is used for authorization.

This problem can be avoided if the DLF exit checks for SHAREOPT(2) and allows the QUERY by using RC=0.

13.6.5 Creating DLF Objects with Utilities

Table 30 shows which utilities can be used to create DLF objects.

Table 30. Utilities that Create DLF Objects		
Utility	**DLF Eligible?**	**Access Method Used**
IEBGENER	NO	BSAM
IEBDG	YES	QSAM
REPRO	YES	VSAM; CISIZE must be a multple of 4K in order to use the Hiperspace.
IEBPTPCH	NO	BSAM

13.6.6 Deleting a Retained Data Set

A retained object can be deleted in any of the following ways:

- ISPF menu 3.2 can delete the data set. Deletion can be verified by using the DISPLAY DLF command.
- Submit the following JCL to keep the data set and delete only the retained object:

```
//CLEAN  EXEC  PGM=COFMSTCH,
//             PARM=¢OBJ=vvvvvvnnnnnn¢
```

  Where:

  vvvvvv The volume serial where the data set resides

  nnnnnn The data set name

- Use the DELOBJ started procedure described in 13.6.1, “Implementing Hiperbatch” on page 241 to keep the data set and delete only the retained object. The following example uses MVSTS3 as the volume serial and P0112JG.VSAM.DLFRET.COBOL as the data set name:

```
S DELOBJ,OBJ=¢MVSTS3P0112JG.VSAM.DLFRET.COBOL¢
```

When displaying objects, the names are truncated to a length of 21 bytes, which can make it impossible to obtain the names of DLF objects from the console output; see the sample console output from the DISPLAY DLF command. The batch program COFMSTCH is sensitive to lowercase alphabetic characters, which could result in errors.

13.6.7 Listing DLFCLASS Authorizations

In order to determine what data set names are eligible to be DLF objects, what users and jobname combinations can connect to those objects, and whether the object can be retained, according to RACF, the following RACF command can be used:

```
RLIST DLFCLASS * DLFDADA ALL
```

Assume a profile exists that was created with the following command:

```
RDEFINE DLFCLASS *.**.DLFNORET.* UACC(NONE) DLFDATA(RETAIN(NO) JOBNAMES(PRODJ1A PRODJ1B))
```

The following output would be received from the RLIST command:

```
CLASS      NAME
-----      ----
DLFCLASS   *.**.DLFNORET.* (G)

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    STORADM         NONE              ALTER    NO

USER       ACCESS
--------   ------
STORADM    ALTER
USER1      READ

DLFDATA INFORMATION
-------------------
RETAIN= NO
JOBNAMES= PRODJ1A  PRODJ1B
```

13.6.8 Recommendations

It is strongly recommended that DLF be started before JES. This order ensures that there are no allocated data sets that are eligible before DLF is active.

Only READ access to the DLFCLASS profile is required for a user to be authorized to connect to the DLF object. The user's access to the DATASET class profile should be considered independently.

It is recommended that installations convert any existing DLF exit to at least consider RACF's decision. When coding the exit, it should:

- Be named COFXDLF1 if it should get control for query when DLF is not active.
- Check for SHAREOPTIONS(2,x) if the query function is allowed.

If performance is a concern, start DLF only when TSO is lightly loaded or use VIO to eliminate calls to DLF.

The Storage Administrator should have class authorization (CLAUTH) to the DLFCLASS class since this has more to do with storage than security.

Data set naming conventions should be modified to include DLF. For example, a data set qualifier could be used to indicate that this data set is eligible for DLF and whether or not it can be retained. The advantages are:

- It is easy to administer; data sets that are eligible for DLF can be identified by data set name.
- Only two profiles are required to control all data sets that are eligible for DLF; one for retained objects and one for non-retained objects.
- Profiles cannot be accidentally deleted, minimizing the data integrity exposure mentioned earlier.
- If a data set's retain attribute is to be changed, the data set must be renamed, erasing any copy in the Hiperspace.

13.7 TSO Message Control

In previous releases of TSO, users could send messages to each other using the SEND command without any authority checking. If the message receiver were logged on, the message would be displayed on the terminal when the user pressed the ENTER key because TSO issued a LISTBC command. If the user were not logged on, a message would be displayed at the sender's terminal to indicate that the user was not logged on and the message would be displayed on the receiver's terminal when LOGON issued a LISTBC command.

With TSO/E 1.4.0, the use of the SEND command could be controlled for the installation in the SYS1.PARMLIB member IKJTSOxx; however, the control applied to all users. With TSO/E 2.1.1 and ACF/VTAM 3.3, using IKJTSOxx and RACF 1.9, the security administrator can audit each message that is sent and control the flow of messages between specific users as shown in Figure 68. An administrator can control whether or not a message can be sent and whether or not the message can be received. This control does not prevent a user from transmitting a message or sending a message as a SYSOUT data set; it's primary function is for auditing message traffic.

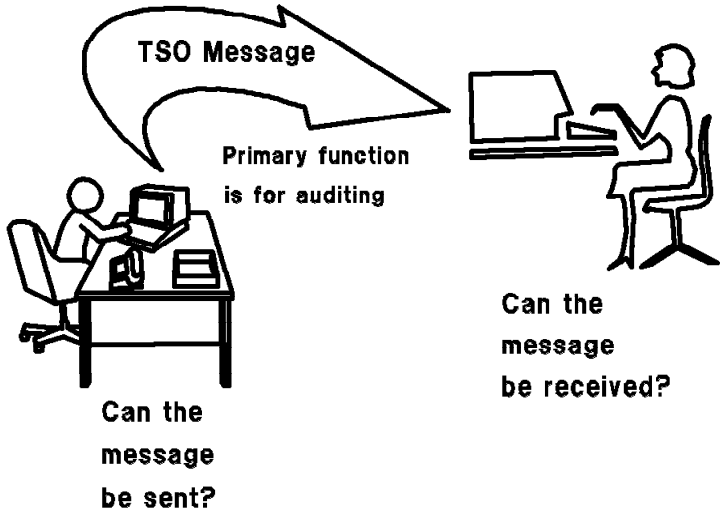

Figure 68. TSO Message Control Overview

13.7.1 Implementing TSO Message Control

The SEND and LISTBC command can now be controlled by RACF 1.9 in conjunction with the following IKJTSOxx parameter settings:

OPERSEND(ON) SEND subcommand of OPERATOR command is allowed.

USERSEND(ON) SEND command is allowed.

SAVE(ON) Messages can be saved in a log data set.

CHKBROD(OFF) LISTBC searches only the user log data set, not SYS1.BRODCAST.

USEBROD(OFF) Messages for users without a user log data set are not saved in SYS1.BRODCAST.

MSGPROTECT(ON) Messages are saved in a user log data set named logname.userid. Access to this data set by the user is allowed using only the LISTBC command in order to prevent the user from browsing the data set. The RTOKEN of the message including the SECLABEL of the message is stored with the message; this requires a new DCB format with LRECL=232 and BLKSIZE=2320. If OFF, the data set name is userid.logname as in previous releases. Because the high level qualifier is the user's own id, RACF allows the user to browse this data set, bypassing any control of the LISTBC command. The old DCB parameters are LRECL=150 and BLKSIZE=1500.

LOGNAME(logname) Logname identifies one or more higher qualifiers of the user log data set. The last qualifier is userid.

13.7.1.1 TSO SEND Control

The ability for a user to send a message to another user can be controlled by defining profiles in the SMESSAGE class. The profile name is the userid of the receiver of the message and no generic characters are allowed in the profile name.

To control the sending of messages, the following steps should be taken:

1. Define profiles for each user who can receive messages, for example:

   ```
   RDEFINE   SMESSAGE  FRED   UACC(READ)

   RDEFINE   SMESSAGE  MARY   UACC(NONE)
   ```

   Where an access authority of NONE prevents any user from sending messages to that user unless specifically authorized; an access authority of READ allows any user to send messages to that user unless specifically prevented. The UACC should be set depending on whether the default should be to allow (READ) or to prevent (NONE) messages from being sent.

2. Give users the appropriate access to the profile, for example:

   ```
   PERMIT   FRED  CLASS(SMESSAGE)  ID(DAVE)  ACCESS(NONE)

   PERMIT   MARY  CLASS(SMESSAGE)  ID(DAVE)  ACCESS(READ)
   ```

   Where an access authority of NONE prevents the user DAVE from sending a message to FRED; an access authority of READ allows the user DAVE to send a message to MARY.

3. Activate the SMESSAGE class with the following command:

   ```
   SETROPTS   CLASSACT(SMESSAGE)
   ```

When the SEND command is issued with the SAVE option, TSO puts the message in the receiver's log data set. When the SEND command is issued with the NOW or LOGON option, VTAM issues a RACROUTE REQUEST=AUTH in the SMESSAGE class to determine if the user is authorized to send the message. If RACF is not installed, the SMESSAGE class is not active, or a profile for the user receiving the message is not found, no decision is returned by RACF, and VTAM allows the message to be sent. If the SMESSAGE class is active but a profile for the user receiving the message is not found, VTAM allows the message to be sent because the default return code (DFTRETC) for the SMESSAGE class in the CDT is 0, indicating that access is allowed when no profile is found. If the SMESSAGE class is active, a profile is found for the user receiving the message, and the sending user has at least READ access, VTAM allows the message to be sent. The only time that VTAM does not allow the message to be sent is when the SMESSAGE class is active, a profile is found for the user

receiving the message, and the sending user does not have at least READ access. In this case, the following message is logged:

```
ICH408I USER(DAVE     ) GROUP(USER     ) NAME(DAVIE JONES          )
ICH408I   DAVE CL(SMESSAGE)
ICH408I   INSUFFICIENT ACCESS AUTHORITY
ICH408I   ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
IKJ55051I NOT ALLOWED TO SEND MESSAGES TO USER(S) FRED,  MESSAGE CANCELLED.
```

Once VTAM decides that the message can be sent, the SEND option, the current logged-on status of the receiving user, and the SECLABEL of the message determines its actual disposition as shown in Table 31. To receive a SECLABEL OK condition, the user's current logged-on SECLABEL must dominate the SECLABEL of the message. For the SEND command, this SECLABEL check only occurs when the receiving user is logged on; if the user is logged-off, the SECLABEL is checked when the LISTBC command is issued.

Table 31. TSO Message Disposition

SEND Option	Logged-on; SECLABEL OK or not active	Logged-on; SECLABEL not OK	Logged-off
SEND NOW	Message immediately displayed	Message cancelled; no notification	User not logged on message
SEND LOGON	Message immediately displayed	Message saved in user's log	Message saved in user's log
SEND SAVE	Message saved in user's log	Message saved in user's log	Message saved in user's log

When the SEND option is NOW and the receiving user is logged on at a SECLABEL that does not dominate the SECLABEL of the message, the message is cancelled without notification to the sending or receiving user.

13.7.1.2 TSO LISTBC Control

The ability for a user to receive a message when the LISTBC command is based on the receiver's current logged-on SECLABEL dominating the SECLABEL of the message. There is no discretionary control; there is only mandatory control, and only if both the SECLABEL and DIRAUTH classes are active. These classes can be activated with the following command:

```
SETROPTS  CLASSACT(SECLABEL DIRAUTH) RACLIST(SECLABEL)
```

For information about the SECLABEL class, see Chapter 3, "Implementing SECLABELs" on page 17. Profiles cannot be defined in the DIRAUTH class because the class descriptor table entry for DIRAUTH specifies PROFDEF=NO. The class is defined for function enabling and to allow auditing of the authorization checking. Since profiles cannot be defined, the auditor requests logging with a SETROPTS LOGOPTIONS option for the DIRAUTH class. Access failures for the DIRAUTH class are recorded in SMF as DIRAUTH class failures. See 4.14.3, "LOGOPTIONS" on page 57 for details.

When a LISTBC command is issued, VTAM issues a RACROUTE REQUEST=AUTH in the DIRAUTH class to determine if the user is authorized to receive the message. If RACF is not installed or the DIRAUTH class or the SECLABEL class is not active, no decision is returned by RACF, and VTAM allows the message to be received. If both the SECLABEL class and the DIRAUTH class are active and the receiving user's SECLABEL dominates the SECLABEL of the message, VTAM allows the message to be received. If the receiving user's current logged-on SECLABEL does not dominate the SECLABEL

of the message, but the user has access to another SECLABEL that does, the message is left in the user′s log and the user receives the following message:

```
IKJ56962I YOUR USER LOG CONTAINS MESSAGES THAT CANNOT
          BE VIEWED AT YOUR CURRENT SECURITY LABEL
```

If the receiving user′s current logged-on SECLABEL does not dominate the SECLABEL of the message, and the user does not have access to a SECLABEL that does, the message is cancelled without notification to either the sending or receiving user, but the following message is written to the log:

```
IKJ577I MESSAGE FROM DAVE HAS BEEN DELETED FROM FRED U LOG
        BECAUSE OF INSUFFICIENT AUTHORITY TO VIEW THE MESSAGE
```

For example, RACF definitions are made for FRED and DAVE as follows:

- FRED is permitted to a SECLABEL of NUNC.
- DAVE is permitted to SECLABELs of NUNC and NCON.
- NCON dominates NUNC.

When both users are logged on with SECLABELs of NUNC, they are authorized to send messages to each other. If, however, DAVE is logged with a SECLABEL of NCON and sends a message to FRED using the save option, the message is discarded and the following message is issued to FRED after the TSO LISTBC command:

```
IKJ56951I NO BROADCAST MESSAGES
```

The following message appears in the SYSLOG:

```
IKJ577I MESSAGE FROM DAVE HAS BEEN DELETED FROM FRED U LOG
        BECAUSE OF INSUFFICIENT AUTHORITY TO VIEW THE MESSAGE
```

FRED is also permitted to SECLABEL NCON but is currently logged on with SECLABEL NUNC. RACF verifies that the user is able to access the message. The user is then prompted to log on with a SECLABEL of NCON before being able to view the message. The format of the message to FRED after the TSO LISTBC command is:

```
IKJ56962I YOUR USER LOG CONTAINS MESSAGES THAT CANNOT BE VIEWED AT
          YOUR CURRENT SECURITY LABEL
IKJ56951I NO BROADCAST MESSAGES
```

13.7.2 Recommendations

When implementing TSO message control, consider the following:

- If strict control to messages is required and users should not be allowed to look at any of their log data sets without proper security authority, then the logname data sets should be protected as follows:

  ```
  ADDSD  ¢logname.*¢  UACC(NONE)  SECLABEL(SYSHIGH)
  ```

  The UACC should be NONE and the SECLABEL should be SYSHIGH since the data sets can contain messages from any user at any security label up to and including SYSHIGH. There are two methods by which this control can be implemented:

  - Define profiles in the RACF class SMESSAGE. This can restrict users or groups from sending messages regardless of what SECLABELS are defined to them.
  - Implement SECLABELS over and above the SMESSAGE class. This prohibits users from sending messages to users with insufficient SECLABEL authority.

- In order to submit message to users when jobs are sent across nodes, JES2 or JES3 must be defined as a trusted procedure in the started task table module ICHRIN03. If this is not done, the DIRAUTH class authorization check fails all messages coming from other nodes.

13.8 TSO RACVAR Function

The use of multiple SECLABELs in an MLS active environment, requires that an ISPPROF data set be allocated for each SECLABEL that a user can specify at logon. In a TSO session when ISPF is invoked, the ISPPROF data set is opened for update and, if MLS is active, the SECLABEL must be equal to the SECLABEL of the user. Following is a REXX exec example that uses the RACVAR function to allocate the current SECLABEL to the user's ISPPROF data set. The SECLABEL used is the one with which the user logs on. If the user logs off and logs on with a different SECLABEL, a new profile is allocated with the new SECLABEL imbedded in the data set name; that is, assuming that no profile data set was allocated before. If no SECLABEL is specified at log-on time, the default SECLABEL assigned to the user is used. If the SECLABEL class is not active, the user still needs a profile data set that can start with userid.ISP.ISPPROF:

```
/* REXX */
if racvar(¢SECLABEL¢)=¢¢ then
  dsn_ispprof=†¢†userid()†.ISPF.ISPPROF¢†
else
  dsn_ispprof=†¢†userid()†.†racvar(¢SECLABEL¢)†.ISPPROF¢†
if sysdsn(dsn_ispprof)=¢OK¢ then
  do
   †alloc ddn(ispprof) dsn(†dsn_ispprof†) old reuse†
  end
else
  do
   †alloc ddn(ispprof) dsn(†dsn_ispprof†) new space(10,10)†,
     †tracks dir(10) lrecl(80) blksize(3120) recfm(f b)†
  end
```

13.8.1 Implementing the RACVAR Function

To use the RACVAR REXX exec, three TSO/E modules have to be updated:

- IRXPARMS
- IRXTSPRM
- IRXISPRM

A sample of the these modules is found in SYS1.PARMLIB. The updates to these modules are found in Appendix L, "TSO/E RACVAR Module Updates" on page 317. More information about installing a REXX function package and specifying directory names in the function package table can be found in *TSO/E Version 2 REXX Reference*. These updates are also documented in *RACF Program Directory for MVS Systems*.

13.8.2 Recommendations

If the ISPF profile data set does not have to be protected, a SECLABEL of SYSNONE can be assigned to the standard profile data set. Then the REXX example for the profile data set allocation is not needed. The standard profile data set does however have to be allocated and the SECLABEL of SYSNONE be assigned before ISPF is used.

13.9 New RACROUTE Macro Parameters

Besides the new RACROUTE parameters for auditing that were discussed in 4.14.5, “New Audit Controls with RACROUTE Macro” on page 60, additional new parameters are provided:

- RACROUTE REQUEST=AUTH STATUS= operand:

  The following new operands are available on the REQUEST=AUTH macro:

  - WRITEONLY is used to verify a write-only SECLABEL check for a class whose RVRSMAC=NO. This operand is used for a user′s access to another user′s mail log data set; the user does not have READ access to the data set, he can only add a message using the TSO SEND command.
  - EVERDOM is used to see whether the user has access to a SECLABEL that could ever dominate that of the current object. This operand is used by the LISTBC command if the user′s current SECLABEL does not dominate the SECLABEL of the message. If the user does have access to a SECLABEL that dominates the message, LISTBC saves the message; otherwise, it is cancelled.
  - ACCESS can be used to return the highest access a user has to a profile.

- RACROUTE REQUEST=DIRAUTH:

  Directed authorization checking (DIRAUTH) is used by the message transmission managers (VTAM, TSO, and session manager) to ensure that the receiver of a message meets MAC requirements; that is, the SECLABEL of the receiver of the message must dominate the SECLABEL of the message. This function is SRB compatible on an ESA system.

- RACROUTE REQUEST=STAT:

  Used to determine whether RACF is active and, optionally, whether a specific class is defined to RACF and active. This function is the same as the RACSTAT macro and was the last function not supported by the RACROUTE macro.

- RACROUTE REQUEST=EXTRACT BRANCH=YES:

  BRANCH=YES can be specified on a RACROUTE REQUEST=EXTRACT, TYPE= EXTRACT or ENCRYPT macro. This form is SRB compatible on ESA systems. It works only for SETROPTS RACLIST profiles that contain a subset of the fields in a profile or if the extract can be done from the ACEE.

Chapter 14. B1 Secure Facility

B1 is a level of data processing security for computer installations. It is defined, along with other possible ratings, by the United States Department of Defense in the *DOD Trusted Computer System Evaluation Criteria, DOD 5200.28-STD*. A system is evaluated by the National Computer Security Center (NCSC) for a specific set of hardware and software called the *trusted computing base* (TCB). The ratings go from D (the least secure) to A (the most secure) with each rating incorporating all the requirements of the less secure ratings. Following is a brief description of the requirements of each rating from least to most secure:

- D - Minimal Protection

  Division D systems, with only one class, have been evaluated but failed to meet the requirements of any higher evaluation class.

- C - Discretionary Protection

  Division C systems, with two classes, provide DAC and accountability of users and the actions they initiate:

  - Class C1 systems, Discretionary Security Protection, satisfy the requirements by providing separation of users and data. They contain controls to allow users access to resources on a selective basis. This environment only supports users and data at a single sensitivity level.
  - Class C2 systems, Controlled Access Protection, enforce a more granular DAC than C1, making users accountable for their actions through logon procedures and auditing.

- B - Mandatory Protection

  Division B systems, with three classes, have a TCB that provides MAC using security labels. Security labels are carried with data in the system. The developer must provide a security policy model on which the TCB is based and furnish a specification of the TCB:

  - Class B1 systems, Labeled Security Protection, have an informal statement of the security policy model that provides MAC over users and resources. They must be capable of attaching a security label to all exported information.
  - Class B2 systems, Structured Protection, have a TCB based on a formal security policy model. In addition, covert channels are addressed; the system is relatively resistant to penetration.
  - Class B3 systems, Security Domains, have a TCB that must mediate all accesses to resources by users, be tamperproof, and exclude code not essential to security policy enforcement. A security administrator is supported, auditing is expanded to signal security related events, and system recovery procedures are required. The system is highly resistent to penetration.

- A - Verified Protection

  Division A systems, with one defined class, provides formal security methods to assure that the mandatory and discretionary security controls can effectively protect sensitive information stored or processed by the system. Documentation is required to demonstrate that the TCB meets the security requirements in all aspects of design, development, and implementation:

  - Class A1 systems, Verified Design, are functionally equivalent to B3 systems. The difference is in the formal design specification and verification that is required and the assurance that the TCB is correctly implemented starting with a formal security policy model and formal specification of the design.
  - Systems beyond class A1 are not formally defined, but may include elements of self-protection and self-verification. These systems may be able to be verified at the source-code level using

formal verification techniques. They may be designed in a trusted facility with only trusted (cleared) personnel.

MVS/XA with RACF 1.8 (and supporting products) was awarded a C2 evaluation on June 15, 1988. After work by IBM development and the NCSC evaluation team, MVS/ESA with RACF 1.9 (and supporting products) was awarded a B1 evaluation on September 27, 1990. The B1 evaluation positions IBM as the premiere security vendor in the large system environment. For those customers wishing to order the software that was evaluated by the NCSC, IBM has created two new packages:

- IBM MVS/ESA JES2 Security System Package (5695-052)
- IBM MVS/ESA JES3 Security System Package (5695-053)

After installing one of these packages, installations planning to implement B1 security must strictly follow the customization guidelines provided in the IBM manual, *MVS/ESA Planning: B1 Security*. This chapter summarizes the information from the planning guide to provide an overview of the elements and steps needed to build a B1 system. In addition, it emphasizes B1 requirements that represent good overall security and integrity practices and should be implemented on all MVS/RACF systems, regardless of the environment and desired security level. The basic security requirements for a B1 system is that it provide a security policy and accountability.

The **security policy** must protect resources of different sensitivity levels. It must prevent unauthorized access to information at higher classification levels and the declassification of information (*-property). The implementation of such policy includes:

- MAC - based on mandatory security labels for subjects and objects in combination with automatic labeling of printed output.
- DAC - access control through user or group authorization in access lists.
- Object Reuse - purge of residual date before reassigning storage objects such as erasing data sets or purging address spaces.

Accountability requires that security-relevant events be associated with the users who caused them. The implementation of accountability requires:

- Identification and Authentication - each user in the system, including operators, must be identified and verified.
- Auditing - recording of security-related events and appropriate selection and reporting capabilities.

A B1 secure facility requires the use of MAC based on SECLABELs, Erase-on-Scratch, operator logon and comprehensive logging and reporting facilities. It must also be implemented with specific, evaluated hardware and software products belonging to the TCB. With the exception of declassification control (*-property), these requirements represent excellent guidelines for installations wanting security but not requiring a B1 rating. Enforcement of the *-property should be reserved for only the most secure facilities.

14.1 Trusted Computing Base

The trusted computing base (TCB) is the security-relevant hardware and software components that meet the security criteria defined by the NCSC. *MVS/ESA Planning: B1 Security* contains a specific list of hardware components that can be included in a B1 MVS system. This list includes processors, DASD controllers and devices, tape controllers and devices, printers, terminals and controllers, and communication controllers. The trusted software consists of the evaluated IBM program products presented in Chapter 1, “Implementing Security in an MVS/ESA Environment” on page 1. These products must be included in the system at the release levels that were evaluated in order to maintain the B1 certification. All software outside the TCB is considered untrusted.

TCB software runs either:

- In supervisor state
- With system protection keys
- APF-authorized

The TCB does not include personal computers because there is no security that prevents the downloading of data at one classification and uploading the same data at a lower classification. It also does not include any databases or applications; these components must be certified locally.

Installations not implementing B1 security should regard these requirements as global recommendations for their hardware and software configuration. DB/DC systems which are usually part of commercial software configurations, should be critically evaluated with regard to their impact on MVS system integrity.

14.2 Security Labeling in a B1 System

In a B1 system, SECLABELs provide enforcement of multi-level security and the identification of hardcopy output. In order to use SECLABELs at a B1 level, the following SETROPTS RACF options must be in effect:

- CLASSACT(SECLABEL) - to activate security label checking.
- MLACTIVE - to require security labels for all users and all resources in certain resource classes.
- MLS - to enforce the confinement property or *-property.
- SECLABELCONTROL - to restrict SECLABEL changes to SPECIAL users.
- MLSTABLE - to require a tranquil state before allowing SECLABEL changes.

All settings, except MLS, are recommended for all RACF/MVS installations.

14.3 Establishing a B1 System

The rule for establishing a B1 system is that the evaluated TCB cannot be modified. This means that no user-written exit routines, SVCs, subsystems, or other system modifications are allowed. The term user-written also includes vendor-supplied programs and modules. Some IBM-supplied exits are allowed or required; they are identified specifically for each product. For those modifications that are required, 14.6, “Modifying a B1 System” on page 262 provides guidelines.

14.3.1 MVS Basic Control Program

MVS/SP 3.1.3 BCP implementation must include:

- Console operator logon - all console operators must log on before issuing commands.
- Operator command auditing - the capability of logging MVS operator commands must be established. Note that access control for operator commands exceeds the B1 requirements.
- No security bypass - all resource access must be able to be monitored; therefore, a complete bypass of security using the PPT NOPASS indicator is not allowed. Trusted procedures must be defined with the SPT TRUSTED attribute instead.
- SMF recording - SMF parameters must be set to halt the system when SMF recording stops.
- Resource protection - the following resources must be protected:
  - UR, communication, and graphics devices must be protected from allocation by application programs.

  - LLA PARMLIB and LLA managed data sets must be protected to be accessible only by operators.
  - System data sets must be protected.
- Appropriate JCL - review JCL for the addiition of new parameters such as:
  - SECLABEL keyword in the JOB card (optional)
  - DPAGELBL and SYSAREA parameters on the OUTPUT card
  - DSNAME keyword for the SYSIN/SYSOUT parameter on the DD card

No user-written exit routines or modifications are allowed except the following, supplied by IBM:

IEALIMIT **IEAVTSEL** **IEFDOIXT** **ISGGREX0**

For installations not requiring B1, consider the following:

- The SMF system halt option should be evaluated in the context of other goals such as system service availability.
- In the interest of usability, JCL changes should be considered only where unavoidable; for example, good defaults might eliminate the need for coding SECLABEL parameters.

14.3.2 Job Entry Subsystems

MVS/SP JES2 3.1.3 or MVS/SP JES3 3.1.3 implementation must include:

- Operator command auditing - the capability of logging JES operator commands must be established. Note that access control for operator commands exceeds the B1 requirements.
- SYSIN/SYSOUT data set protection - limit access to the user who created the data sets unless explicitly authorized.
- JES SPOOL data set protection - for example, JESNEWS and SYSLOG.
- JES system data set protection - checkpoint and offload.

The following restrictions apply:

- NJE and RJE/RJP are not permitted,
- Only PSF/MVS printers are used for secure output.
- Commands requiring higher than READ authority are not permitted in the job input stream.
- JES3-managed consoles, tape utilities, and DSPs are not allowed.
- No user-written exit routines or modifications are allowed except the following, supplied by IBM:

  IATUX20 **IATUX21** **IATUX23**

For installations not requiring B1, consider the following:

- NJE and RJE/RJP are probably important functions that cannot and should not be eliminated in commercial environments.
- Printers without PSF/MVS facilities can be used, but only to print unclassified data or data marked SYSLOW.

14.3.3 MVS/Data Facility Product

MVS/DFP 3.1.1 implementation must include:

- Non-temporary data set protection - use SECLABELs in DATASET and TAPEVOL classes.
- Temporary data set protection - activate TEMPDSN class.
- Catalog protection - use SECLABEL SYSNONE.
- SMS protection - use STORCLAS and MGMTCLAS; use PROGRAM class to protect certain ISMF programs.

The following restrictions apply:

- Do not use the DASDVOL class
- Use only ICF catalogs
- No user-written exit routines or modifications are allowed except the following, supplied by IBM:

  IFG0199I **OMODVOL1** **EMODVOL1**

For installations not requiring B1, the use of the DASDVOL class may be preferable to using the OPERATIONS attribute.

14.3.4 Time Sharing Option

TSO/E 2.1.1 implementation must include:

- User definition - define users to RACF; do not use SYS1.UADS.
- Logon audit - record types 30 and 80 SMF records.
- Message protection - use individual user logs instead of SYS1.BRODCAST to audit all messages.
- Spool access - control SUBMIT, CANCEL and OUTPUT through RACF.

The following restrictions apply:

- Do not give users the OPERATOR attribute.
- Do not activate the Information Center Facility.
- Replace the standard IKJEFF53 exit with the one supplied in SYS1.SAMPLIB.
- No user-written exit routines or modifications are allowed except the following, supplied by IBM:

  IKJEFF10 **IKJEFF53*** **INMXZ01** **INMXZ02**
  INMXZ03 **INMRZ01** **INMRZ02** **INMRZ04**
  INMRZ11 **INMRZ12** **INMRZ13**

  *Replace the IBM-supplied exit with the one in SYS1.SAMPLIB

14.3.5 Print Service Facility

PSF/MVS 3.1.1 implementation must include:

- Security labels - activate mandatory print labeling for security labels on each output page.
- Tamper-free separator pages - replace standard PSF/MVS exits.
- Secure PSF/MVS environment - control libraries and startup procedures.
- Override controls - authorize users to override print labeling and operators to override separator pages.

The following restrictions apply:

- Use only guaranteed printers for secure output.
- Do not use the direct printing subsystem (DPSS).
- No user-written exit routines or modifications are allowed except the following, supplied by IBM:

  APSUX01* APSUX02* APSUX03

  *Replace with APSUX01S and APSUX02S

For installations not requiring B1, consider using output labeling on a more selective basis.

14.3.6 Virtual Telecommunications Access Method

ACF/VTAM 3.3 implementation must include:

- Application authorization - control ACB OPEN requests.
- Message authorization - control TSO/E SEND at the VTAM level.

No user-written exit routines or modifications are allowed except the following, supplied by IBM:

ISTPUCWC

14.3.7 Resource Access Control Facility

RACF 1.9 implementation must include:

- Started Procedures Table - set up an SPT with the following trusted entries (instead of PPT NOPASS):

  - CATALOG - CONSOLE - DUMPSRV - JES
  - LLA - MOUNT - PSF - SMF
  - SMS - VLF - VTAM

- Audit all logons - select type 30 SMF records and modify Report Writer module ICHRSMFI accordingly.
- Security labels - define SECLABELs and assign them to all users and all profiles in the following classes:

  - DATASET - DEVICES - TAPEVOL

- Surrogate job submission - create profiles in class SURROGAT.
- Resource classes - activate the following classes (* and RACLIST):

  - ACCINUM* - DEVICES* - DIRAUTH - FACILITY*
  - JESSPOOL - OPERCMDS* - PSFMPL* - RACFVARS*
  - SECLABEL* - SMESSAGE* - TAPEVOL - TEMPDSN
  - TERMINAL* - TSOAUTH* - TSOPROC* - VTAMAPPL*

- SETROPTS options - set the following RACF options:

  - CATDSN(FAILURES) - ERASE(ALL)
  - GENERICOWNER - JES(BATCHALLRACF XMBAALLRACF)
  - MLACTIVE(FAILURES) - MLS(FAILURES)
  - MLSTABLE - PROTECTALL(FAILURES)
  - SECLABELCONTROL

The following restrictions apply:

- Do not activate DASDVOL.

- Use the global access checking table only for resources not requiring SECLABELs or having a SECLABEL of SYSLOW. Grant only READ access.

For installations not requiring B1, consider the following:

- SETROPTS MLS(...) and ERASE(ALL) should be evaluated in the context of the individual policy and needs; SETROPTS NOMLS and ERASE SECLEVEL(...) are probably sufficient in most cases.
- A &RACUID.* /ALTER global access checking table entry is considered reasonable for non-B1 security.

14.4 Auditing a B1 System

Auditing has been enhanced through system options to support comprehensive logging (such as SPT TRUSTED instead of PPT NOPASS) and RACF options to define the type and extent of logging required. For certain security events, mandatory logging occurs, for others a variety of logging options exist. The following enhancements have been provided for B1:

- Mandatory logging - console operator logon and logoff and any unauthorized attempt to use a SECLABEL.
- Optional logging - the AUDITOR has these additional options available:
  - SETROPTS LOGOPTIONS - sets logging options on a resource class basis; they enhance the logging options specified in profiles. This also provides logging for classes that have no profiles.
  - SETROPTS SECLABELAUDIT - logs all access attempts to resources protected by SECLABELs.

The contents of existing SMF records have been enhanced and a new record type (83) has been added. The RACF Report Writer has been enhanced to support the new security events and status settings.

The changes that provide more comprehensive logging are important to non-B1 installations. Each new option should be evaluated individually to determine whether it should be used permanently or temporarily when needed.

14.5 Operating a B1 System

In addition to establishing the system as a B1 system, it must also be operated as a B1 system as follows:

- Messages between users and operators can always be exchanged.
- Printer operators must check the system-generated random numbers on separator pages before distributing output.
- SMF records in storage at system failures must be recovered through IPCS.
- Tapes returned to a scratch pool must be degaussed.
- An OPERATIONS user must scratch residual temporary data sets.
- The operator must drain the system before MLQUIET is issued.

These are also pertinent recommendations for non-B1 systems.

14.6 Modifying a B1 System

The concept of B1 level security is based on MVS system integrity. When trusted software is carefully implemented and maintained under strict controls, untrusted programs and software components are covered by the MVS integrity statement. *MVS/ESA Planning: B1 Security* reviews the existing MVS integrity and coding guidelines as follows:

- Protection - preventing unauthorized programs from accessing sensitive data and functions.
- Identification - preventing alteration of resources passed between authorized and unauthorized programs.
- Validation - validity-checking data passed between programs.
- Serialization - ensuring that data used by one program cannot be altered by another.

MVS/ESA SPL: Application Development Guide discusses coding guidelines to prevent the following potential exposures for authorized programs:

- User-supplied addresses for user storage areas
- User-supplied addresses for protected control blocks
- Resource Identification
- SVC routines calling SVC routines
- Control program and user data accessibility
- Resource serialization

These guidelines are crucial for system integrity and therefore valid and important for MVS system security. All system-level code should be developed using these guidelines, and any product code executing as authorized should be reviewed accordingly. Where source code is not available, MVS integrity statements for such products should be reviewed. If not readily available, such statements should be requested prior to licensing and implementing products requiring MVS authorization.

Appendix A. Minimum Software Requirements for New Functions

Table 32 provides a list of functions and the minimum software levels required to use that function.

Table 32 (Page 1 of 2). Minimum Software Requirements for New Functions

Function	MVS	DFP	TSO	JESx	VTAM	RACF	PSF
B1	3.1.3	3.1.1	2.1.1	3.1.3	3.3	1.9	1.3.0
Batch LSR Control	3.1.0e					1.9	
CATDSNS	3.1.3					1.9	
Console Signon	3.1.3					1.9	
DBCS/NLS	3.1.3	3.1.1	2.1.1		3.3	1.9	1.3.0
Default Userids	3.1.3			3.1.3		1.9	
Device Control	3.1.3					1.9	
DLF Control	3.1.3	3.1.1				1.9 (*)	
Dynamic Parse			1.4.0			1.9	
Early Verify	3.1.3			3.1.3		1.9	
Enhanced Auditing	3.1.3					1.9	
Enhanced Generics						1.9	
Enhanced Propagation	3.1.3			3.1.3		1.9	
GENERICOWNER						1.9	
Group Tree in Storage	3.1.0 (VLF)					1.9	
Hiperbatch Control	3.1.3	3.1.1				1.9 (*)	
Jobname Control	3.1.3			3.1.3		1.9	
Job Source Control	3.1.3			3.1.3		1.9	
LLA Control	3.1.3					1.9	
LU 6.2 Control	2.1.0				3.3	1.9	
NJE Source Control	3.1.3			3.1.3		1.9	
NJE Translation	3.1.3			3.1.3		1.9	
Operator Command Control	3.1.3			3.1.3		1.9	
Printer Control	3.1.3			3.1.3		1.9	

Table 32 (Page 2 of 2). Minimum Software Requirements for New Functions							
Function	**MVS**	**DFP**	**TSO**	**JESx**	**VTAM**	**RACF**	**PSF**
Restructured DB						1.9	
RACVAR Function			2.1.0			1.9	
RJE/RJP Signon	3.1.3			3.1.3		1.9	
Security Overlays	3.1.3			3.1.3		1.9	1.3.0
SECLABELS	3.1.3		2.1.1	3.1.3	3.3	1.9	1.3.0
Surrogate Support	3.1.3			3.1.3		1.9	
SYSIN/SYSOUT Control	3.1.3		2.1.1	3.1.3		1.9	
Temporary DSN Control	3.1.3	3.1.1				1.9	
TSO LOGON Changes	3.1.3		2.1.1			1.9	
TSO Message Control	3.1.3		2.1.1		3.3	1.9	
VTAM Application Control	2.1.0				3.3	1.9	
Work Unit Identity	3.1.3	3.1.1	2.1.1	3.1.3	3.3	1.9	1.3.0

(*) Can be overridden by DLF exit

Appendix B. RACF Resource Classes

Table 33 provides a list of some RACF resource classes and some of their attributes. Not all RACF resource classes are listed. Some resource classes listed are not new with RACF 1.9, but are included where special information has to be given.

If the value in the RACLREQ column is Y, a RACLIST is required for the class. The value in the DFTRETC column is the default return code when no profile is found. If no value is given, the RACF default return code of 4 applies. If the value in the SLBLREQ column is Y, a SECLABEL is required when MLACTIVE is enabled. If the value in the RVRSMAC column is Y, the dominance check is reversed for that class. Parenthetical numbers refer to notes at the end of the table.

Table 33 (Page 1 of 3). RACF Resource Classes and Attributes

Description	Name	Profile Name	RACLREQ	DFTRETC	SLBLREQ	RVRSMAC
Partner LU verification	**APPCLU**	*nodename.local_lu.partner_lu*	(1)			
Conditional access to operator commands	**CONSOLE**	*console_id*		8		Y
Data set access	**DATASET**	*dsname*			Y	
Allocation control of UR, TP, and GRAPHIC devices	**DEVICES**	*sysid.device_class.unit_name.dev_addr*	Y		Y	
ACF/VTAM and TSO/E session managers	**DIRAUTH**	*** no profiles in this class *****				
Hiperbatch* control (2)	**DLFCLASS**	*dsname*				
Control for (3)	**FACILITY**					
--RJE signon		*RJE.rmtname*				
--NJE signon (4)		*NJE.nodename*				
--CATDSNS exceptions		*ICHUNCAT.dsname*				
--LLA control		*CSVLLA.dsname*				
--BLSR control		*CSR.BLSRHIPR.subsys*				

Table 33 (Page 2 of 3). RACF Resource Classes and Attributes

Description	Name	Profile Name	RACLREQ	DFTRETC	SLBLREQ	RVRSMAC
POE control for	**JESINPUT**			8		
-- Internal reader		*INTRDR*				
-- JES2 remotes		*RnnnnRDm*				
-- JES2 NJE		*adjacent nodename*				
-- JES2 local readers		*RDRnn*				
-- Spool offload		*OFFn.JR*				
-- JES3 remotes		*workstation name*				
-- JES3 NJE (SNA)		*NJERDR*				
-- JES3 NJE(BSC)		*adjacent nodename*				
-- JES3 local readers		*jname*				
-- Dump job		*DUMPJOB*				
-- Disk reader		*JES3DRDS*				
Control of jobnames	**JESJOBS**			8		
--For SUBMIT		*SUBMIT.currentnode.jobname.userid*				
--For CANCEL		*CANCEL.currentnode.userid.jobname*				
Control of SYSIN and SYSOUT	**JESSPOOL**	*node.userid.jobname.jobid.Ddsid.dsname*		8		
--Jobs for SDSF		*node.userid.jobname.jobid*				
Control of JOBs and SYSOUT from other nodes (5)	**NODES**					
--Translate job userid		*subnode.USERJ.userid*				
--Translate job group		*subnode.GROUPJ.groupid*				
--Translate job SECLABEL		*subnode.SECLJ.seclabel*				
--Translate SYSOUT userid		*xeqnode.USERS.userid*				
--Translate SYSOUT group		*xeqnode.GROUPS.groupid*				
--Translate SYSOUT SECLABEL		*xeqnode.SECLS.seclabel*				
	NVASAPDT					
Operator commands (6)	**OPERCMDS**	*subsystem.command.qualifier.object*	Y			

Table 33 (Page 3 of 3). RACF Resource Classes and Attributes

Description	Name	Profile Name	RACLREQ	DFTRETC	SLBLREQ	RVRSMAC
Control who does not use propagation	**PROPCNTL**	*userid*	Y			
Override printer security defaults	**PSFMPL**	*PSF.DPAGELBL*	Y	8		
Define variable names	**RACFVARS**	*&profile_name (7)*				
SDSF for JES2	**SDSF**	*For profiles (8)*				
Defines security labels	**SECLABEL**	*security_label*	Y	8		
TSO SEND message	**SMESSAGE**	*sending-to_userid*		0		
Surrogate job submission	**SURROGAT**	*jobcard_userid.SUBMIT*				
Protecting tape volumes	**TAPEVOL**	*volser*			Y	
Control of temporary non-vio data sets	**TEMPDSN**	*** no profiles in this class ***				
Protecting terminals	**TERMINAL**	*terminal_name*			Y	Y
Protecting VTAM applications	**VTAMAPPL**	*applname*	Y			
Output destination	**WRITER**			8	Y	Y
--Local printer		*subsys.LOCAL.PRTnnnn*				
--Local punch		*subsys.LOCAL.PUNnnnn*				
--JES2		*subsys.RJE.Rnnnn.PRm*				
--JES2		*subsys.RJE.Rnnnn.PUm*				
--JES3		*subsys.RJP.devicename*				
--SYSOUT to other nodes		*subsys.NJE.nodename*				

Note:

(1) This class is not RACLISTable on SETROPTS. VTAM uses the RACLIST macro. Key information is kept in the SESSION segment. No MAC checking is done between the users who are establishing sessions.

(2) Contains a DLFDATA segment.

(3) Not all uses of FACILITY are listed.

(4) Required for NJE operator command control.

(5) Access list is ignored; only UACC is used. UACC determines level of trust for the node.job combination.

(6) Need FACILITY class definitions for RJE and NJE commands.

(7) &RACxxx names reserved for RACF use.

(8) For SDSF profiles, see Appendix I, “SDSF Resource Names Tables” on page 295.

Appendix C. NJE Job Header and Token DSECTS

The NJE job header contains the token that is used to identify the RACF profile for a job. The NJE job header is mapped by the macro $NHD found in SYS1.HASPSRC. The security token is imbedded within the job header and is mapped by the macro ICHRUTKN found in SYS1.MACLIB. The following examples show the actual NJE job header in storage just before it is transmitted to the execution node as well as the DSECTS mentioned above. Only the beginning portion of the job header up to and including the security section is shown. From this information SAF is able to make preliminary checks when the job enters the system and later when the job is scheduled for execution by JES.

NJE Job Header

```
0000  01700000 00D40000 161BC1E7 00050101 | .....M....AX.... |
0010  00000000 00000000 00000000 D7F0F1F1 | ............P011 |
0020  F2D9D5C2 D7F0F1F1 F2D9D540 00000000 | 2RNBP0112RN .... |
0030  00000000 00000000 00000000 A1C44D7A | ............D(: |
0040  A5830320 C3F2D1C5 E2F24040 D7F0F1F1 | vc..C2JES2  P011 |
0050  F2D9D540 C3F2D1C5 E2E34040 00000000 | 2RN C2JEST  .... |
0060  00000000 C3F2D1C5 E2F24040 00000000 | ....C2JES2  .... |
0070  00000000 C3F2D1C5 E2F24040 00000000 | ....C2JES2  .... |
0080  00000000 00000000 00000000 00000005 | ................ |
0090  00000258 000061A8 00000064 D5D6D9E3 | ....../y....NORT |
00A0  C8D9E4D7 40404040 40404040 40404040 | HRUP             |
00B0  00000000 00000000 00000000 00000000 | ................ |
00C0  00000000 00000000 00000000 00000000 | ................ |
00D0  00000000 00000000 00348400 00000000 | .........d.....  |
00E0  00000000 00000000 00000000 00000000 | ................ |
00F0  00000000 00000000                   | ........         |
00F8  00000000 00000000 00000000 00000000 | ................ |
0108  0000000C 8A000000 002805F5 DD180058 | ...........5....  |
0118  8C000004 00005001 02060001 0000E2E8 | .....&.......SY  |
0128  E2C8C9C7 C840C3F2 D1C5E2F2 40400000 | SHIGH C2JES2 ..  |
0138  00000000 00000000 00000000 00000000 | ................ |
0148  00000000 0000C9D5 E3D9C4D9 40400000 | ......INTRDR ..  |
0158  00000000 0000D7F0 F1F1F2D9 D540D7F0 | ......P0112RN P0 |
0168  F1F1F240 40400000                   | 112   .......... |
```

The highlighted section is the area mapped by the ICHRUTKN macro. It contains the propagated userid and seclabel.

Job Header DSECT (Part 1)

```
                                1027+*************************************************************************
                                1028+*                                                                       *
                                1029+*        NETWORK JOB HEADER RECORD DSECT                                *
                                1030+*                                                                       *
                                1031+*************************************************************************
000000                          1033+NJH      DSECT                    NETWORK JOB HEADER RECORD
                                1035+*        BLOCK CONTROL INFORMATION
000000 013C                     1037+NJHLEN   DC    AL2(NJHLLEN)        LENGTH OF ENTIRE BLOCK
000002 00                       1038+NJHFLAGS DC    X¢00¢               FLAGS
000003 00                       1039+NJHSEQ   DC    BL.1¢0¢,AL.7(0)     TRANSMISSION SEQUENCE INDICATOR
                          00004 1040+NJHLBCI  EQU   *-NJH               LENGTH OF BLOCK CONTROL INFORMATION
                                1042+*        GENERAL SECTION
000004                          1044+NJHG     DS    0F                  START OF GENERAL SECTION
000004 00D4                     1045+NJHGLEN  DC    AL2(NJHGLLEN)       LENGTH OF GENERAL SECTION
000006                          1046+NJHGFLGS DS    0BL2                SECTION TYPE FLAGS
000006 00                       1047+NJHGTYPE DC    AL1(NTYPGEN)          ID FOR GENERAL SECTION
000007 00                       1048+NJHGMOD  DC    AL1(NJHG$MOD)         MODIFIER
                          00000 1049+NJHG$MOD EQU   B¢00000000¢           VALUE OF MODIFIER
000008 0000                     1051+NJHGJID  DC    Y(0)                JOB IDENTIFIER
00000A C1                       1052+NJHGJCLS DC    C¢A¢                JOB CLASS
00000B C1                       1053+NJHGMCLS DC    C¢A¢                MESSAGE CLASS
00000C 00                       1054+NJHGFLG1 DC    B¢00000000¢         FLAGS
                          00002 1056+NJHGF1PE EQU   B¢00000010¢         NJHGPASS is encrypted
                          00001 1057+NJHGF1NE EQU   B¢00000001¢         NJHGNPAS is encrypted
00000D 00                       1059+NJHGPRIO DC    AL1(0)              SELECTION PRIORITY
00000E 00                       1060+NJHGORGQ DC    AL1(0)              ORIGIN NODE SYSTEM QUALIFIER
00000F 00                       1061+NJHGJCPY DC    AL1(0)              JOB COPY COUNT
000010 00                       1062+NJHGLNCT DC    AL1(0)               JOB LINE COUNT
000011 000000                   1063+         DC    XL3¢00¢              RESERVED
000014 4040404040404040         1064+NJHGACCT DC    CL8¢ ¢              NETWORKING ACCOUNT NUMBER
00001C 4040404040404040         1065+NJHGJNAM DC    CL8¢ ¢              JOB NAME
000024 4040404040404040         1066+NJHGUSID DC    CL8¢ ¢              USERID (TSO, VM) to NOTIFY
00002C                          1067+NJHGPASS DS    CL8                 PASSWORD
000034                          1068+NJHGNPAS DS    CL8                 NEW PASSWORD
00003C 0000000000000000         1069+NJHGETS  DC    FL8¢0¢              ENTRY TIME/DATE STAMP
000044 4040404040404040         1070+NJHGORGN DC    CL8¢ ¢              ORIGIN NODE NAME
00004C 4040404040404040         1071+NJHGORGR DC    CL8¢ ¢              ORIGIN REMOTE NAME
000054 4040404040404040         1072+NJHGXEQN DC    CL8¢ ¢              EXECUTION NODE NAME
00005C 4040404040404040         1073+NJHGXEQU DC    CL8¢ ¢              EXECUTION USER ID(VM/370)
000064 4040404040404040         1074+NJHGPRTN DC    CL8¢ ¢              DEFAULT PRINT NODE NAME
00006C 4040404040404040         1075+NJHGPRTR DC    CL8¢ ¢              DEFAULT PRINT REMOTE NAME
000074 4040404040404040         1076+NJHGPUNN DC    CL8¢ ¢              DEFAULT PUNCH NODE NAME
00007C 4040404040404040         1077+NJHGPUNR DC    CL8¢ ¢              DEFAULT PUNCH REMOTE NAME
000084 4040404040404040         1078+NJHGFORM DC    CL8¢ ¢              JOB FORMS
00008C 00000000                 1079+NJHGICRD DC    F¢0¢                INPUT CARD COUNT
000090 00000000                 1080+NJHGETIM DC    F¢0¢                ESTIMATED EXECUTION TIME
000094 00000000                 1081+NJHGELIN DC    F¢0¢                ESTIMATED OUTPUT LINES
000098 00000000                 1082+NJHGECRD DC    F¢0¢                ESTIMATED OUTPUT CARDS
00009C 4040404040404040         1083+NJHGPRGN DC    CL20¢ ¢             PROGRAMMER¢S NAME
0000B0 4040404040404040         1084+NJHGROOM DC    CL8¢ ¢              PROGRAMMER¢S ROOM NUMBER
0000B8 4040404040404040         1085+NJHGDEPT DC    CL8¢ ¢              PROGRAMMER¢S DEPARTMENT
0000C0 4040404040404040         1086+NJHGBLDG DC    CL8¢ ¢              PROGRAMMER¢S BUILDING NUMBER
0000C8 00000000                 1087+NJHGNREC DC    F¢0¢                RECORD COUNT ON OUTPUT XMISSION
0000CC 00000000                 1088+         DC    F¢0¢                RESERVED
0000D0 4040404040404040         1089+NJHGNTYN DC    CL8¢ ¢              Node to send NOTIFY message
0000D8                          1090+NJHGEND  DS    0F                  END OF GENERAL SECTION
                          00024 1091+NJHGORGU EQU   NJHGUSID            ORGIN USER ID
                          000D4 1092+NJHGLLEN EQU   *-NJHG              LENGTH OF GENERAL SECTION
```

Job Header DSECT (Part 2)

```
                                1094+*        JES2 SUBSYSTEM SECTION
0000D8                          1096+NJH2     DS    0F                START OF JES2 SECTION
0000D8 0034                     1097+NJH2LEN  DC    AL2(NJH2LLEN)     LENGTH OF JES2 SECTION
0000DA                          1098+NJH2FLGS DS    0BL2              SECTION TYPE FLAGS
0000DA 84                       1099+NJH2TYPE DC    AL1(NTYPJES2)       ID FOR JES2 SECTION
0000DB 00                       1100+NJH2MOD  DC    AL1(NJH2$MOD)       MODIFIER
                         00000  1101+NJH2$MOD EQU   B¢00000000¢         VALUE OF MODIFIER
0000DC 00                       1103+NJH2FLG1 DC    B¢00000000¢       FLAGS
0000DD 000000                   1104+         DC    XL3¢00¢           RESERVED
0000E0 40404040                 1105+NJH2ACCT DC    CL4¢ ¢            ORIGINATOR¢S JES2 ACCOUNT NUMBER
0000E4 4040404040404040         1106+NJH2USID DC    CL8¢ ¢            USER SMF FIELD
0000EC 0000000000000000         1108+NJH2USR  DC    0CL8¢ ¢,XL8¢00¢   JCL USER ID (BEFORE SAF CALL)       X
                                   +                                   VERIFIED USER ID (AFTER)
0000F4 0000000000000000         1109+NJH2GRP  DC    0CL8¢ ¢,XL8¢00¢   JCL GROUP ID (BEFORE SAF CALL)      X
                                   +                                   VERIFIED GROUP ID (AFTER)
0000FC 0000000000000000         1110+NJH2SUSR DC    0CL8¢ ¢,XL8¢00¢   SUBMITTER¢S USER ID
000104 0000000000000000         1111+NJH2SGRP DC    0CL8¢ ¢,XL8¢00¢   SUBMITTER¢S GROUP ID
                         00034  1112+NJH2ACML EQU   *-NJH2             MINIMUM LENGTH FOR FIELDS REQUIRED  X
                                   +                                   FOR AUTH CHECKS IN JES2
00010C                          1114+NJH2END  DS    0F                END OF JES2 SECTION
                         00034  1115+NJH2LLEN EQU   *-NJH2             LENGTH OF JES2 SECTION
                                1117+*        NJH2FLG1 BIT DEFINITIONS
                         00003  1118+NJH2FJOB EQU   B¢00000011¢         JOB IS A BATCH JOB WHEN ZERO
                         00001  1119+NJH2FSTC EQU   B¢00000001¢         JOB IS A STARTED TASK
                         00002  1120+NJH2FTSU EQU   B¢00000010¢         JOB IS TIME-SHARING USER
                         00004  1121+NJH2USE  EQU   B¢00000100¢         JCTUSEID PRESENT IN HEADER
                                1123+*        JOB SCHEDULING SECTION
00010C                          1125+NJHE     DS    0F                START OF JOB SCHEDULING SECTION
00010C                          1126+NJHELEN  DS    AL2(NJHELLEN)     LEN OF JOB SCHEDULING SECTION
00010E                          1127+NJHEFLGS DS    0BL2              JOB SCHEDULING FLAGS
00010E 8A                       1128+NJHETYPE DC    AL1(NTYPGJS)        ID FOR JOB SCHEDULING SECTION
00010F 00                       1129+NJHEMOD  DC    AL1(NJHE$JS)        MODIFIER FOR JOB SCHEDULING
                         00000  1130+NJHE$JS  EQU   B¢00000000¢         VALUE OF MODIFIER
000110 00000000                 1131+NJHEPAGE DC    XL4¢00¢           ESTIMATED BEGIN PAGE COUNT
000114 00000000                 1132+NJHEBYTE DC    XL4¢00¢           ESTIMATED BYTE COUNT
000118                          1133+NJHEEND  DS    0F                END OF JOB SCHEDULING SECTION
                         0000C  1134+NJHELLEN EQU   *-NJHE             LEN OF JOB SCHEDULING SECTION
                                1136+*************************************************************************
                                1137+*                                                                       *
                                1138+*        Job Header Security Section.  The NJHTTOKN is mapped by        *
                                1139+*        the ICHRUTKN macro and represents the external format of       *
                                1140+*        the Security Authorization Facility Token.                     *
                                1141+*                                                                       *
                                1142+*************************************************************************
000118                          1144+NJHT     DS    0F                Start of Security Section
000118 0058                     1145+NJHTLEN  DC    AL2(NJHTLLEN)     Length of Security Section
00011A                          1146+NJHTFLGS DS    0BL2              Section type flags
00011A 8C                       1147+NJHTTYPE DC    AL1(NTYPSAF)        ID for Security Section
00011B 00                       1148+NJHTMOD  DC    AL1(NJHT$MOD)       Modifier
                         00000  1149+NJHT$MOD EQU   B¢00000000¢         Value of Modifier
00011C 0004                     1151+NJHTLENP DC    AL2(NJHTTOKN-NJHTLENP)  Length of prefix sectn
00011E 00                       1152+NJHTFLG0 DC    B¢00000000¢       Security section flags
                         00080  1154+NJHTF0JB EQU   B¢10000000¢         Token represents job
00011F 00                       1156+         DC    AL1(0)            Reserved
000120 4040404040404040         1158+NJHTTOKN DC    CL($TKNLEN)¢ ¢    Mapped SAF token                <=== ICHRUTKN inserts here
000170                          1159+NJHTEND  DS    0F                End of Security Section
                         00058  1160+NJHTLLEN EQU   *-NJHT             Length of Security Section
```

Security Token DSECT (Part 1)

```
                          1775       ICHRUTKN
                          1776+*                                  NOT
000000                    1777+TOKEN    DSECT  ,TOKPTR            UTOKEN / RTOKEN MAPPING
                          1778+*
000000                    1779+TOKLEN   DS    XL1                 UTOKEN / RTOKEN LENGTH
                          1780+*
000001                    1781+TOKVERS  DS    XL1                 UTOKEN / RTOKEN VERSION #
                          1782+*
                    00001 1783+TOKVER01 EQU   1                   RELEASE 1.9 UTOKEN
                          1784+*        02-255                    RESERVED FOR EXPANSION
                    00001 1785+TOKCVER  EQU   1                   CURRENT UTOKEN VERSION
                          1786+*
000002                    1787+TOKFLG1  DS    XL1                 MISCELLANEOUS FLAGS
                          1788+*
                    00080 1789+TOKENCR  EQU   X¢80¢               TOKEN IS ENCRYPTED
                    00040 1790+TOKENEXT EQU   X¢40¢               TOKEN IS IN THE EXTERNAL FORMAT
                    00020 1791+TOKLT19  EQU   X¢20¢               TOKEN CREATED BY PRE RACF 1.9 CALL
                    00010 1792+TOKVXPRP EQU   X¢10¢               VERIFYX PROPAGATION OCCURRED
                    00008 1793+TOKUNUSR EQU   X¢08¢               NJE UNKNOWN USER
                    00004 1794+TOKLOGU  EQU   X¢04¢               LOG USER INDICATOR
                    00002 1795+TOKRSPEC EQU   X¢02¢               RACF SPECIAL INDICATOR
                          1796+*
                          1797+*        EQU   X¢01¢               RESERVED
                          1798+*
000003                    1799+TOKSTYP  DS    XL1                 SESSION TYPE (01 - 255 DEC.)
                          1800+*
                    00001 1801+TOKSAS   EQU   1                   SYSTEM ADDRESS SPACE
                    00002 1802+TOKCMND  EQU   2                   COMMAND
                    00003 1803+TOKCONS  EQU   3                   CONSOLE OPERATOR
                    00004 1804+TOKSTP   EQU   4                   STARTED PROCEDURE
                    00005 1805+TOKMNT   EQU   5                   MOUNT
                    00006 1806+TOKTSO   EQU   6                   TSO LOGON
                    00007 1807+TOKBCH   EQU   7                   INTERNAL READER BATCH JOB
                    00008 1808+TOKXBM   EQU   8                   EXECUTION BATCH MONITOR
                    00009 1809+TOKRJE   EQU   9                   RJE OPERATOR
                    0000A 1810+TOKNJE   EQU   10                  NJE OPERATOR
                    0000B 1811+TOKNJEUS EQU   11                  VERIFYX UNKNOWN USER TOKEN
                    0000C 1812+TOKEBCH  EQU   12                  EXTERNAL READER BATCH JOB
                    0000D 1813+TOKRBCH  EQU   13                  RJE BATCH JOB
                    0000E 1814+TOKNBCH  EQU   14                  NJE BATCH JOB
                    0000F 1815+TOKNSYS  EQU   15                  NJE SYSOUT
                    00010 1816+TOKEXBM  EQU   16                  EXTERNAL XBM
                    00011 1817+TOKRXBM  EQU   17                  RJE XBM
                    00012 1818+TOKNXBM  EQU   18                  NJE XBM
                    00012 1819+TOKLSESS EQU   18                  LAST CURRENTLY DEFINED SESSION
                          1820+*          19-255                  RESERVED FOR EXPANSION
                          1821+*
000004                    1822+TOKFLG2  DS    XL1                 MISCELLANEOUS FLAGS
                          1823+*
                    00080 1824+TOKDFLT  EQU   X¢80¢               DEFAULT TOKEN
                    00040 1825+TOKUDUS  EQU   X¢40¢               UNDEFINED USER
                    00020 1826+TOKML    EQU   X¢20¢               ALL REQUIRED ML OPTIONS ACTIVE
                    00010 1827+TOKERR   EQU   X¢10¢               TOKEN IN ERROR
                    00008 1828+TOKTRST  EQU   X¢08¢               PART OF TRUSTED COMPUTER BASE
                    00004 1829+TOKSUS   EQU   X¢04¢               SURROGATE USERID
                    00002 1830+TOKREMOT EQU   X¢02¢               REMOTE JOB INDICATOR
                    00001 1831+TOKPRIV  EQU   X¢01¢               PRIVILEDGED USER INDICATOR
```

Security Token DSECT (Part 2)

```
                                1832+*
000005                          1833+TOKPOEX  DS    AL1               PORT OF ENTRY CLASS INDEX
                                1834+*
                        00001   1835+TOKTERM  EQU   1                 TERMINAL CLASS
                        00002   1836+TOKCON   EQU   2                 CONSOLE  CLASS
                        00003   1837+TOKJESI  EQU   3                 JESINPUT CLASS
                                1838+*              4-255             RESERVED FOR FUTURE
                                1839+*
000006                          1840+         DS    CL2               RESERVED
                                1841+*
000008                          1842+TOKSCL   DS    CL8               SECLABL
000010                          1843+TOKXNOD  DS    CL8               EXECUTION NODE
000018                          1844+TOKSUSR  DS    CL8               SUBMITTING USERID
000020                          1845+TOKSNOD  DS    CL8               SUBMITTER NODE
000028                          1846+TOKSGRP  DS    CL8               SUBMITTING GROUPID
000030                          1847+TOKPOE   DS    CL8               PORT OF ENTRY(CONS ID,TERM. ID)
000038                          1848+         DS    CL8               RESERVED FOR EXPANSION
000040                          1849+TOKUSER  DS    CL8               SESSION OWNER USERID
000048                          1850+TOKGRUP  DS    CL8               SESSION OWNER GROUPID
                                1851+*
                        00050   1852+TOKCURLN EQU   *-TOKEN           CURRENT VERSION LENGTH
```

Appendix D. Sample RACF Report Writer Listing

```
90.058 17:51:02                                   RACF REPORT                                   PAGE   1

                                TEST OPERCMDS NOAUDIT AND CONSOLE AUDIT(ALL)

COMMAND GROUP ENTERED -
 RACFRW  TITLE ('TEST OPERCMDS NOAUDIT AND CONSOLE AUDIT(ALL)')
 SELECT PROCESS GROUP(OPER MASTER ALL INFO)
 EVENT LOGON CLASS(OPERCMDS CONSOLE)
 EVENT ACCESS
  LIST SORT(DATE TIME)
 END


90.058 17:51:02                       RACF REPORT - LISTING OF PROCESS RECORDS                  PAGE   4

                                TEST OPERCMDS NOAUDIT AND CONSOLE AUDIT(ALL)

                                                          E
                                                          V  Q
                                                          E  U
                          *JOB/USER  *STEP/     --TERMINAL--  N  A
 DATE     TIME    SYSID   NAME      GROUP        ID     LVL  T  L

90.058 17:48:22 SMF5    OPERJ3     OPER                  0   2  0  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=P0112JG
                        JES3 MAIN OPERATOR                          AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                                    SESSION=CONSOLE OPERATOR,CONSOLE=02
                                                                    CONSOLE=02,LEVEL=00,INTENT=READ,ALLOWED=READ

90.058 17:48:55 SMF5    02         ALL                   0   2  0  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=P0112JG
                        CONSOLE 8E2                                 AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                                    SESSION=CONSOLE OPERATOR,CONSOLE=02
                                                                    CONSOLE=02,LEVEL=00,INTENT=READ,ALLOWED=READ

90.058 17:49:20         02         ALL                   0   1  1  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=
                        CONSOLE 8E2                                 AUTH=(NONE),REASON=(RACINIT FAILURE)
                                                                    SESSION=CONSOLE OPERATOR,CONSOLE=02

90.058 17:49:23 SMF5    02         ALL                   0   1  1  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=
                        CONSOLE 8E2                                 AUTH=(NONE),REASON=(RACINIT FAILURE)
                                                                    SESSION=CONSOLE OPERATOR,CONSOLE=02

90.058 17:49:25 SMF5    02         ALL                   0   1  1  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=
                        CONSOLE 8E2                                 AUTH=(NONE),REASON=(RACINIT FAILURE)
                                                                    SESSION=CONSOLE OPERATOR,CONSOLE=02

90.058 17:49:40 SMF5    ;LASJD                        0   1  4  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=
                                                AUTH=(NONE),REASON=(RACINIT FAILURE)
                                                                    SESSION=CONSOLE OPERATOR,TOKEN USER ATTRIBUTES=(UNDEFINED USER),
                                                                    CONSOLE=02

90.058 17:50:19 SMF5    OPERJ3     OPER                  0   2  0  JOBID=( 00.000 00:00:00),USERDATA=(),OWNER=P0112JG
                        JES3 MAIN OPERATOR                          AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                                    SESSION=CONSOLE OPERATOR,CONSOLE=02
                                                                    CONSOLE=02,LEVEL=00,INTENT=READ,ALLOWED=READ
```

EVENT Code Description

EVENT No.	EVENT Qualfier	Description
1		Job Initiation. TSO Logon/Logoff
	0	Successful Initiation
	4	Invalid Terminal/Console
2		Resource Access
	0	Successfull Access

Appendix E. JES3 Command Profile Names

Table 34 shows JES3 commands, authority, and profile names.

Table 34 (Page 1 of 3). JES3 Command Profile Names

Command	Authority	Profile Name
*X,DR,M=	UPDATE	jesx.CALL.DR.member
*X	UPDATE	jesx.CALL.dspname
*C,DEV	UPDATE	jesx.CANCEL.DEV.dev
*C	UPDATE	jesx.CANCEL.name
*S,DR,M=	UPDATE	jesx.START.DR.member
*S,DEV	UPDATE	jesx.START.DEV.dev
*S	UPDATE	jesx.START.name
*R,DEV	UPDATE	jesx.RESTART.DEV.dev
*R	UPDATE	jesx.RESTART.name
*FAIL,DEV	CONTROL	jesx.FAIL.DEV.dev
*FAIL	CONTROL	jesx.FAIL.name
*V	UPDATE	jesx.VARY.DEV
*V RECOVER	CONTROL	jesx.VARYRECOVER.DEV
*DELAY	UPDATE	jesx.DELAY
*DISABLE	UPDATE	jesx.DISABLE
*ENABLE	UPDATE	jesx.ENABLE
*SWITCH	UPDATE	jesx.SWITCH
*FREE	CONTROL	jesx.FREE
*TRACE	UPDATE	jesx.TRACE
*ERASE	READ	jesx.ERASE
*Z	READ	jesx.MESSAGE
*T	UPDATE	jesx.ROUTE.CMD.system
*DUMP	CONTROL	jesx.STOP.DUMP
*RETURN	CONTROL	jesx.STOP.RETURN
*I A	READ	jesx.DISPLAY.A
*I B	READ	jesx.DISPLAY.B
*I C	READ	jesx.DISPLAY.C
*I D	READ	jesx.DISPLAY.D
*I F	READ	jesx.DISPLAY.F
*I G	READ	jesx.DISPLAY.G
*I J	READ	jesx.DISPLAY.JOB

Table 34 (Page 2 of 3). JES3 Command Profile Names		
Command	**Authority**	**Profile Name**
*I J E	READ	jesx.DISPLAY.JOBE
*I K	READ	jesx.DISPLAY.K
*I L	READ	jesx.DISPLAY.L
*I M	READ	jesx.DISPLAY.M
*I MT	READ	jesx.DISPLAY.MT
*I N	READ	jesx.DISPLAY.N
*I NJE	READ	jesx.DISPLAY.NJE
*I O	READ	jesx.DISPLAY.O
*I P	READ	jesx.DISPLAY.P
*I PROCLIB	READ	jesx.DISPLAY.PROCLIB
*I Q	READ	jesx.DISPLAY.Q
*I Q N	READ	jesx.DISPLAY.QN
*I R	READ	jesx.DISPLAY.R
*I S	READ	jesx.DISPLAY.S
*I T	READ	jesx.DISPLAY.T
*I U	READ	jesx.DISPLAY.U
*I X	READ	jesx.DISPLAY.X
*F C	UPDATE	jesx.MODIFY.C
*F E	UPDATE	jesx.MODIFY.E
*F F	UPDATE	jesx.MODIFY.F
*F G	UPDATE	jesx.MODIFY.G
*F J	UPDATE	jesx.MODIFY.JOB
*F J P	UPDATE	jesx.MODIFY.JOBP
*F K	UPDATE	jesx.MODIFY.K
*F L	UPDATE	jesx.MODIFY.L
*F M	UPDATE	jesx.MODIFY.M
*F MT	UPDATE	jesx.MODIFY.MT
*F N	UPDATE	jesx.MODIFY.N
*F NJE	UPDATE	jesx.MODIFY.NJE
*F O	UPDATE	jesx.MODIFY.O
*F Q	UPDATE	jesx.MODIFY.Q
*F S	UPDATE	jesx.MODIFY.S
*F T	UPDATE	jesx.MODIFY.T
*F U	UPDATE	jesx.MODIFY.U

Table 34 (Page 3 of 3). JES3 Command Profile Names		
Command	**Authority**	**Profile Name**
*F V	UPDATE	jesx.MODIFY.V
*F V RECOVER	CONTROL	jesx.MODIFYRECOVER.V
*F W	UPDATE	jesx.MODIFY.W
*F X	UPDATE	jesx.MODIFY.X
//*PROCESS CBPRNT	UPDATE	jesx.PROCESS.CBPRNT
//*PROCESS DISPDJC	UPDATE	jesx.PROCESS.DISPDJC
//*PROCESS DISPLAY	UPDATE	jesx.PROCESS.DISPLAY
//*PROCESS DJCPROC	UPDATE	jesx.PROCESS.DJCPROC
//*PROCESS DR,M=	UPDATE	jesx.PROCESS.DR.member
//*PROCESS ISDRVR	UPDATE	jesx.PROCESS.ISDRVR
//*PROCESS JESNEWS	UPDATE	jesx.PROCESS.JESNEWS
//*PROCESS name	UPDATE	jesx.PROCESS.name

Appendix F. JES2 Command Profile Names

Table 35 shows JES2 commands, authority, and profile names.

Table 35 (Page 1 of 4). JES2 Command Profile Names

Command	Authority	Profile Name
$B device	UPDATE	jesx.BACKSP.DEV
$E SYS	CONTROL	jesx.RESTART.SYS
$E device	UPDATE	jesx.RESTART.DEV
$E J	CONTROL	jesx.RESTART.BAT
$E ′ ′	CONTROL	jesx.RESTART.BAT
$E LINE(x)	CONTROL	jesx.RESTART.LINE
$E Lx.yyy	CONTROL	jesx.RESTART.LINE
$E LOGON(x)	CONTROL	jesx.RESTART.LOGON
$F device	UPDATE	jesx.FORWARD.DEV
$G A	UPDATE	jesx.GMODIFY.JOB
$G C	UPDATE	jesx.GCANCEL.JOB
$G D	READ	jesx.GDISPLAY.JOB
$G H	UPDATE	jesx.GMODIFY.JOB
$G R	UPDATE	jesx.GROUTE.JOB
$I device	UPDATE	jesx.INTERRUPT.DEV
$M	READ	jesx.MSEND.CMD
$N	READ	jesx.NSEND.CMD
$N device	UPDATE	jesx.REPEAT.DEV
$O ′ ′	UPDATE	jesx.RELEASE.JOBOUT
$O Q	UPDATE	jesx.RELEASE.JOBOUT
$O J	UPDATE	jesx.RELEASE.BATOUT
$O S	UPDATE	jesx.RELEASE.STCOUT
$O T	UPDATE	jesx.RELEASE.TSUOUT
$RXEQ	UPDATE	jesx.ROUTE.JOBOUT
$RALL	UPDATE	jesx.ROUTE.JOBOUT
$RPRT	UPDATE	jesx.ROUTE.JOBOUT
$RPUN	UPDATE	jesx.ROUTE.JOBOUT
$VS	CONTROL	jesx.VS
$DU	READ	jesx.DISPLAY.DEV
$Dinit stmt	READ	jesx.DISPLAY.initstmt
$D ′ ′	READ	jesx.DISPLAY.JOB

Table 35 (Page 2 of 4). JES2 Command Profile Names		
Command	**Authority**	**Profile Name**
$DJ	READ	jesx.DISPLAY.BAT
$DS	READ	jesx.DISPLAY.STC
$DT	READ	jesx.DISPLAY.TSU
$DI	READ	jesx.DISPLAY.INITIATOR
$DA	READ	jesx.DISPLAY.JOB
$DF	READ	jesx.DISPLAY.QUE
$DN	READ	jesx.DISPLAY.JOB
$DQ	READ	jesx.DISPLAY.JOB
$LSYS	CONTROL	jesx.DISPLAY.SYS
$L ′ ′	READ	jesx.DISPLAY.JOBOUT
$LJ	READ	jesx.DISPLAY.BATOUT
$LS	READ	jesx.DISPLAY.STCOUT
$LT	READ	jesx.DISPLAY.TSUOUT
$DPCE	READ	jesx.DISPLAY.PCE
$DEXIT	READ	jesx.DISPLAY.EXIT
$DTRACE(x)	READ	jesx.DISPLAY.TRACE
$DSPOOL	READ	jesx.DISPLAY.SPOOL
$P initiator	CONTROL	jesx.STOP.INITIATOR
$P	CONTROL	jesx.STOP.SYS
$PJES2	CONTROL	jesx.STOP.SYS
$P TRACE(x)	CONTROL	jesx.STOP.TRACE
$P SPOOL	CONTROL	jesx.STOP.SPOOL
$P device	UPDATE	jesx.STOP.DEV
$P LINE(x)	CONTROL	jesx.STOP.LINE
$P Lx.yyy	CONTROL	jesx.STOP.LINE
$P LOGON(x)	CONTROL	jesx.STOP.LOGON
$P RMT(x)	CONTROL	jesx.STOP.RMT
$Pjobname	UPDATE	jesx.STOP.JOB
$P J	UPDATE	jesx.STOP.BAT
$P S	UPDATE	jesx.STOP.STC
$P T	UPDATE	jesx.STOP.TSU
$P Q	UPDATE	jesx.STOP.JOBOUT
$S	CONTROL	jesx.START.SYS
$S A	CONTROL	jesx.START.AUTOCMD

Table 35 (Page 3 of 4). JES2 Command Profile Names

Command	Authority	Profile Name
$S initiator	CONTROL	jesx.START.INITIATOR
$S N	CONTROL	jesx.START.NET
$S SPOOL	CONTROL	jesx.START.SPOOL
$S TRACE(x)	CONTROL	jesx.START.TRACE
$S device	UPDATE	jesx.START.DEV
$S LINE(x)	CONTROL	jesx.START.LINE
$S Ln.xxx	CONTROL	jesx.START.LINE
$S LOGON(x)	CONTROL	jesx.START.LOGON
$S RMT(x)	CONTROL	jesx.START.RMT
$T init stmt	CONTROL	jesx.MODIFY.init stmt
$T I	CONTROL	jesx.MODIFY.INITIATOR
$T EXIT	CONTROL	jesx.MODIFY.EXIT
$T ALL	CONTROL	jesx.MODIFY.SYS
$T SYS	CONTROL	jesx.MODIFY.SYS
$T ′ ′	UPDATE	jesx.MODIFY.JOB
$T J	UPDATE	jesx.MODIFY.BAT
$T S	UPDATE	jesx.MODIFY.STC
$T T	UPDATE	jesx.MODIFY.TSU
$TO jobname	UPDATE	jesx.MODIFY.JOBOUT
$TO J	UPDATE	jesx.MODIFY.BATOUT
$TO S	UPDATE	jesx.MODIFY.STCOUT
$TO T	UPDATE	jesx.MODIFY.TSUOUT
$T A	UPDATE	jesx.MODIFY.AUTOCMD
$T M	CONTROL	jesx.MODIFY.MSGROUTE
$T NODE	CONTROL	jesx.MODIFY.NODE
$T NUM	CONTROL	jesx.MODIFY.NUM
$T SSI	CONTROL	jesx.MODIFY.SSI
$A A	UPDATE	jesx.MODIFYRELEASE.JOB
$A ′ ′	UPDATE	jesx.MODIFYRELEASE.JOB
$A J	UPDATE	jesx.MODIFYRELEASE.BAT
$A S	UPDATE	jesx.MODIFYRELEASE.STC
$A T	UPDATE	jesx.MODIFYRELEASE.TSU
$A Q	UPDATE	jesx.MODIFYRELEASE.JOB
$H A	UPDATE	jesx.MODIFYHOLD.JOB

Table 35 (Page 4 of 4). JES2 Command Profile Names		
Command	**Authority**	**Profile Name**
$H ′ ′	UPDATE	jesx.MODIFYHOLD.JOB
$H J	UPDATE	jesx.MODIFYHOLD.BAT
$H S	UPDATE	jesx.MODIFYHOLD.STC
$H T	UPDATE	jesx.MODIFYHOLD.TSU
$H Q	UPDATE	jesx.MODIFYHOLD.JOB
$T device	UPDATE	jesx.VARY.DEV
$T OFFLOADx	CONTROL	jesx.MODIFY.OFF
$T OFFx.yy	CONTROL	jesx.MODIFY.OFF
$T FSS	CONTROL	jesx.VARY.FSS
$T LGN	CONTROL	jesx.VARY.LGN
$T RMT	UPDATE	jesx.VARY.RMT
$ADD APPL	CONTROL	jesx.ADD.APPL
$ADD DESTID	CONTROL	jesx.ADD.DESTID
$ADD FSS	CONTROL	jesx.ADD.FSS
$DM	READ	jesx.SEND.MESSAGE
$Z device	UPDATE	jesx.HALT.DEV
$Z initiator	CONTROL	jesx.HALT.INITIATOR
$Z A	CONTROL	jesx.HALT.AUTOCMD
$Z SPOOL	CONTROL	jesx.HALT.SPOOL
$C A	******	jesx.CANCEL.AUTOCMD
$C ′ ′	UPDATE	jesx.CANCEL.JOB
$C J	UPDATE	jesx.CANCEL.BAT
$C S	UPDATE	jesx.CANCEL.STC
$C T	UPDATE	jesx.CANCEL.TSU
$C device	UPDATE	jesx.CANCEL.DEV

Appendix G. JES2 Exit for Assigning a Default SECLABEL to Output

```
         TITLE ¢JES2 EXIT23 DEFAULT SECLABEL PROCESSOR EXIT -- PROLOG (C
               COMMENT BLOCK)¢
************************************************************************
*                                                                      *
* Module Name = HASPXJ23 CSECT                                         *
*                                                                      *
* Descriptive Name = DEFAULT SECLABEL PROCESSOR EXIT                    *
*                                                                      *
* Status = OS/VS2 - See $MODULE expansion below for FMID, VERSION      *
*                                                                      *
* Function = This exit assigns a default SECLABEL to jobs that        *
*            are selected for printing on a PSF printer with           *
*            guaranteed printing enforced.  Normally if such a         *
*            job were selected, it would be NOSELECTed since           *
*            PSF would be unable to locate the appropriate             *
*            member of the security definitions library.               *
*                                                                      *
*                                                                      *
* Notes = See below                                                    *
*                                                                      *
*    Dependencies = 1) JES2 Exit Effector                              *
*                                                                      *
*    Restrictions = This code is provided as an example of install-    *
*                   ation extensions to JES2.  This code is not to     *
*                   considered type 1 supported code of IBM.  Any      *
*                   problems encountered in the use of this sample     *
*                   code is a user responsibility.  The IBM support    *
*                   center does not support user extensions or         *
*                   sample user exits.                                 *
*                                                                      *
*    Register Conventions = See entry point documentation              *
*                                                                      *
*    Patch Label = none                                                *
*                                                                      *
* Module Type = CSECT                                                  *
*                                                                      *
*    Processor = OS/VS ASSEMBLER H                                     *
*                                                                      *
*    Module Size = See $MODEND macro expansion at end of assembly      *
*                                                                      *
*    Attributes = REENTRANT, SUPERVISOR STATE, PROTECT KEY OF          *
*                 HASP¢S (1), RMODE 24, AMODE 24                       *
*                                                                      *
* Entry Point = EXIT23                                                 *
*                                                                      *
*    PURPOSE = SEE FUNCTION                                            *
*                                                                      *
*    LINKAGE = STANDARD MVS LINKAGE                                    *
```

Figure 69 (Part 1 of 4). JES2 User Exit 23 to Provide a Default Security Label

```
*                                                                     *
* EXIT-NORMAL = RETURN TO CALLER (HASPPRPU)                           *
*                                                                     *
* EXIT-ERROR = NONE                                                   *
*                                                                     *
* EXTERNAL REFERENCES = SEE BELOW                                     *
*                                                                     *
*    ROUTINES = MISCELLANEOUS JES2 SERVICE ROUTINES                   *
*                                                                     *
*    DATA AREAS = SEE $MODULE MACRO EXPANSION                         *
*                                                                     *
*    CONTROL BLOCKS = SEE $MODULE MACRO EXPANSION                     *
*                                                                     *
* TABLES = SEE $MODULE MACRO DEFINITION (BELOW)                       *
*                                                                     *
* MACROS = JES2 - $CALL, $DEST, $ENTRY, $FREEBUF, $GETBUF, $MODEND,   *
*                 $MODULE, $PBLOCK, $PRPUT, $RETURN, $SAVE, $SEPPDIR  *
*                                                                     *
* MACROS = MVS  - TIME                                                *
*                                                                     *
* CHANGE ACTIVITY:                                                    *
*                                                                     *
*        @311    MVS/SP-JES2 VERSION 3 RELEASE 1 LEVEL 1              *
*                (SP3.1.1, HJE3311)                                   *
*                                                                     *
***********************************************************************
         TITLE ¢JES2 EXIT23 DEFAULT SECLABEL PROCESSOR -- PROLOG ($HASPC
               GBL)¢
         COPY  $HASPGBL            COPY HASP GLOBALS
HASPXJ23 $MODULE ENVIRON=FSS,NOTICE=NONE,                             C
               $HASPEQU,           HASP EQUATES                       C
               $HFCT,              HASP FSS COMMUNICATION TABLE       C
               $JIB,               JOE INFORMATION BLOCK              C
               $JOE,               JOB OUTPUT ELEMENT DSECT           C
               JSPA,               JOB SEPARATOR PAGE AREA            C
               $JNEW               JESNEWS CONTROL BLOCK
```

Figure 69 (Part 2 of 4). JES2 User Exit 23 to Provide a Default Security Label

```
        TITLE ¢JES2 EXIT23 DEFAULT SECLABEL PROCESSOR -- EXIT23¢
************************************************************************
*                                                                      *
*       EXIT23 - Installation Exit 23 Routine                          *
*                                                                      *
* Overview:                                                            *
*                                                                      *
*       This exit is invoked via the exit effector in HASPSSSM         *
*       from the HASPFSSM module during GETDS processing.  When-       *
*       ever a new JIB is initialized during GETDS processing,         *
*       exit 23 is invoked in HASPFSSM.  At this time, the assoc-      *
*       iated JCT, IOT, and CHECKPOINT records are read and the        *
*       JSPA is built.                                                 *
*                                                                      *
* Function:                                                            *
*                                                                      *
*       The documented use of this exit is to modify the user-         *
*       dependant fields of the JSPA.  Here we test for a              *
*       security label in the CHAR-JOE information in the JIB.         *
*       If a valid security label is not found, we add a               *
*       default label to the JIB CHAR-JOE area.  We cannot add         *
*       a default SECLABEL to the JSPA directly, as the security       *
*       label fields are over-written by the OPENDS routine of         *
*       HASPFSSM after this exit returns control to HASPFSSM.          *
*                                                                      *
*                                                                      *
* Linkage:                                                             *
*                                                                      *
*       Branch entered from the JES2 exit effector                     *
*                                                                      *
* Environment:                                                         *
*                                                                      *
*       Functional Subsystem (HASPFSSM)                                *
*                                                                      *
* Recovery:                                                            *
*                                                                      *
*       None                                                           *
*                                                                      *
* Register Usage:                                                      *
*                                                                      *
*       Registers on entry to the exit routine -                       *
*                                                                      *
*       R0     -  0                                                    *
*       R1     -  A(PARM LIST)--------------> +0 - A(JSPA)             *
*       R2-R10 -  N/A                          +4 - A(JIB)             *
*       R11    -  A(HFCT)                      +8 - A(FSACB)           *
*       R13    -  A(OS-STYLE SAVE AREA)       +12 - A(FSSCB)          *
*       R14    -  RETURN ADDRESS              +16 - A(GDS PARM LIST)  *
*       R15    -  ENTRY ADDRESS                                        *
```

Figure 69 (Part 3 of 4). JES2 User Exit 23 to Provide a Default Security Label

```
*                                                                     *
*       Valid return codes -                                          *
*                                                                     *
*       0    -    Continue normal processing                         *
*       4    -    Continue normal processing                         *
*       8    -    Do not generate job separator page                 *
*                  (note: JESNEWS not printed if rc=8)                *
*       12   -    Force generation of job separator page             *
*                                                                     *
*                                                                     *
*                                                                     *
***********************************************************************
        SPACE 1
***********************************************************************
*                                                                     *
*       EXIT23 MAIN CALLING ROUTINE                                   *
*                                                                     *
***********************************************************************
        SPACE 1
        USING JIB,R2                    establish JIB addressability
        USING JOE,R3                    establish JIB addressability
        SPACE 1
EXIT023 $ENTRY BASE=R12                 exit routine entry point
       $SAVE                            save registers
        LR    R12,R15                   establish base addressability
        SPACE 1
        L     R2,4(,R1)                 load JIB address
        LA    R3,JIBCJOE                load JIB CHAR-JOE copy address
        CLC   JOESECLB-JOE(8,R3),=XL8'0000000000000000'               C
                                        check for null security label
        BNE   ENDIT                     if non-null label get out
        MVC   JOESECLB-JOE(8,R3),=CL8'DEFLABEL'                       C
                                        assign default label
        SPACE 1
ENDIT   SLR   R0,R0
       $RETURN RC=(R0)                  return to HASPFSSM
        SPACE 1
        DROP  R2,R3
        SPACE 1
       $MODEND
        SPACE 1
        END
```

Figure 69 (Part 4 of 4). JES2 User Exit 23 to Provide a Default Security Label

Appendix H. JES3 Exit to Assign a Default SECLABEL to Output

```
UX45     TITLE ¢FSS WTR GETDS SRL MODIFICATION USER EXIT¢
IATUX45  AMODE 31
IATUX45  RMODE ANY
         IATYASM
*START OF SPECIFICATIONS*************************************
*                                                           *
*   $MOD(IATUX45)         PROD(JES3)                        *
*                                                           *
*   MODULE NAME = IATUX45                                   *
*                                                           *
*   DESCRIPTIVE NAME = FSS WTR GETDS SRL                    *
*                      MODIFICATION USER EXIT               *
*                                                           *
*   COPYRIGHT =                                             *
*                                                           *
*   COPYRIGHT = 5685-002                                    *
*               THIS MODULE IS †RESTRICTED MATERIALS OF IBM† *
*               (C) COPYRIGHT IBM CORP. 1981, 1988,         *
*               LICENSED MATERIALS - PROPERTY OF IBM        *
*               REFER TO COPYRIGHT INSTRUCTIONS             *
*               FORM NUMBER G120-2083                       *
*                                                           *
*   STATUS = OS/VS2 HJS3311                                 *
*                                                           *
*   FUNCTION = THIS EXIT IS CALLED BY THE FSS WTR DSP BEFORE *
*              IT RETURNS THE COMPLETED GETDS SRL BACK TO   *
*              THE FSS FOR THE FSA MAKING THE GETDS REQUEST. *
*              THE USER HAS ACCESS TO THE COMPLETE SRL. THIS *
*              INCLUDES JOB HEADER AND TRAILER INFORMATION  *
*              AND DATA SET HEADER INFORMATION IN THE JSPA  *
*              (JOB SEPARATOR PAGE AREA), THE JMR, AND DATA *
*              SET CHARACTERISTICS.                         *
*                                                           *
*   OPERATION = THE EXIT CONTAINS NO JES3 CODE, EXCEPT TO   *
*               RETURN INDICATING EXIT IS A DUMMY.  THIS    *
*               PREVENTS FUTURE CALLS TO THE EXIT BY THE FSS *
*               WRITER.                                     *
*                                                           *
*   NOTES =                                                 *
*                                                           *
*     DEPENDENCIES = NONE                                   *
*                                                           *
*     RESTRICTIONS = NONE                                   *
*                                                           *
```

Figure 70 (Part 1 of 5). JES3 User Exit 45 to Define Default Security Label

```
*     REGISTER CONVENTIONS =                                    *
*                    R0  -  NULL PARAMETER REGISTER             *
*                    R1  -  NULL PARAMETER REGISTER             *
*                    R2  -  WORK REGISTER                       *
*                    R3  -  WORK REGISTER                       *
*                    R4  -  WORK REGISTER                       *
*                    R5  -  WORK REGISTER                       *
*                    R6  -  JSPA BASE                           *
*                    R7  -  GETDS FSIP BASE                     *
*                    R8  -  SRL BASE                            *
*                    R9  -  WORK REGISTER                       *
*                    R10 -  BASE REGISTER                       *
*                    R11 -  FCT ADDRESS                         *
*                    R12 -  TVT                                 *
*                    R13 -  WRITER DATA AREA (IATODFD)          *
*                    R14 -  RETURN REGISTER FOR ASAVE           *
*                    R15 -  MODULE ENTRY ADDRESS/RETURN CODE    *
*                                                               *
*     PATCH LABEL = PTCHUX45                                    *
*                                                               *
*                                                               *
*   MODULE TYPE = PROCEDURE                                     *
*                                                               *
*     PROCESSOR = ASSEMBLER H                                   *
*                                                               *
*     MODULE SIZE = 1K BYTES                                    *
*                                                               *
*     ATTRIBUTES = REENTRANT                                    *
*                                                               *
*   ENTRY POINT = IATUX45                                       *
*                                                               *
*        PURPOSE = SEE FUNCTION                                 *
*                                                               *
*        LINKAGE = ACALLED BY IATOSFG                           *
*                                                               *
*        INPUT = WRITER DATA AREA IN R13, WHICH ALLOWS          *
*                ACCESS TO THE GETDS SRL TO BE MODIFIED.        *
*                                                               *
*        OUTPUT = NONE                                          *
*                                                               *
*        EXIT-NORMAL = ARETURN RC=0   NORMAL EXIT               *
*                      ARETURN RC=4   RESERVED                  *
*                      ARETURN RC=8   RESERVED                  *
*                      ARETURN RC=12  RESERVED                  *
*                      ARETURN RC=16  DUMMY EXIT                *
*                                                               *
*        EXIT-ERROR = NONE                                      *
*                                                               *
*        USER EXIT =  NONE                                      *
*                                                               *
*   EXTERNAL REFERENCES =                                       *
```

Figure 70 (Part 2 of 5). JES3 User Exit 45 to Define Default Security Label

```
*                                                                     *
*        ROUTINES = NONE                                              *
*                                                                     *
*        DATA AREAS = SEE CONTROL BLOCKS                              *
*                                                                     *
*        CONTROL BLOCKS = IATYCNS  -  R/O                             *
*                         IATYEQU  -  R/O                             *
*                         IATYFCT  -  R/O                             *
*                         IATYFDB  -  R/O                             *
*                         IATYFSA  -  R/O                             *
*                         IATYFSS  -  R/O                             *
*                         IATYOSE  -  R/O                             *
*                         IATYREG  -  R/O                             *
*                         IATYSRL  -  R/O                             *
*                         IATYSTA  -  R/O                             *
*                         IATYTVT  -  R/O                             *
*                         IATYUXL  -  R/O                             *
*                         IATYWTR  -  R/O                             *
*                                                                     *
*    TABLES = SEE CONTROL BLOCKS                                      *
*                                                                     *
*    MACROS-EXECUTABLE =                                              *
*                                                                     *
*        JES3 MACROS = ARETURN                                        *
*                      IATXPTCH                                       *
*                      IATYASM                                        *
*                      IATYMOD                                        *
*                                                                     *
*        SYSTEM MACROS = NONE                                         *
*                                                                     *
*    ENQUEUE RESOURCES = NONE                                         *
*                                                                     *
*    MP LOCKS USED =                                                  *
*                                                                     *
*        JES3 LOCKS = NONE                                            *
*                                                                     *
*        SYSTEM LOCKS = NONE                                          *
*                                                                     *
*    MESSAGES = NONE                                                  *
*                                                                     *
*    ABEND CODES = NONE                                               *
*                                                                     *
*END OF SPECIFICATIONS*****************************************
```

Figure 70 (Part 3 of 5). JES3 User Exit 45 to Define Default Security Label

```
         TITLE ¢IATYCNS - CONSOLE BUFFER MAP¢
         IATYCNS TYPE=(INPUT,FCTQ)
         TITLE ¢IATYEQU - JES3 STANDARD EQUATES¢
         IATYEQU
         TITLE ¢IATYFCT - FUNCTION CONTROL TABLE¢
         IATYFCT
         TITLE ¢IATYFDB - FILE DESCRIPTION BLOCK¢
         IATYFDB
         TITLE ¢IATYFSA - FSS APPLICATION TABLE ENTRY¢
         IATYFSA
         TITLE ¢IATYFSS - FUNCTIONAL SUBSYSTEM TABLE ENTRY¢
         IATYFSS
         TITLE ¢IATYOSE - OUTPUT SCHEDULING ELEMENT¢
         IATYOSE
         TITLE ¢IATYREG - JES3 REGISTER EQUATES¢
         IATYREG
         TITLE ¢IATYSRL - FSI SERVICE REQUEST LIST¢
         IATYSRL TYPE=GETDS
         TITLE ¢IATYSTA - STAGING AREA CONTROL TABLE¢
         IATYSTA
         TITLE ¢IATYTVT - TRANSFER VECTOR TABLE¢
         IATYTVT
         TITLE ¢IATYUXL - USER EXIT ADDRESS LIST¢
         IATYUXL
         TITLE ¢IATYWIR - WRITER DATA AREA¢
         IATYWIR DRVR=FSSWIR
         TITLE ¢FSS WIR GETDS SRL MODIFICATION USER EXIT¢
IATUX45  CSECT
         USING IATUX45,R15         TEMPORARY ADDRESSABILITY
         SPACE 1
         IATYMOD BR=YES            IDENTIFY THE MODULE
         SPACE 1
         DROP  R15                 DROP TEMP. ADDRESSABILITY
         LR    R10,R15             SET UP BASE REGISTER AND
         USING IATUX45,R10          SET MODULE ADDRESSABILITY
         USING WIRSTART,R13        WRITER DSECT ADDRESSABILITY
         SPACE 1
*--------------------------------------------------------------*
*        SET FOR USE OF SRL GETDS AREA                         *
*--------------------------------------------------------------*
         L     R7,WIRFSTAR         POINT AT GETDS STAGING AREA
         USING STADSECT,R7          AND SET ADDRESSABILITY
         LA    R8,STADATA          POINT AT SRL
         USING SRLSTART,R8          AND SET ADDRESSABILITY
         DROP  R7                  STADSECT
         SPACE 2
```

Figure 70 (Part 4 of 5). JES3 User Exit 45 to Define Default Security Label

```
*-------------------------------------------------------------*
*        SET FOR USE OF FSI GETDS AREA                        *
*-------------------------------------------------------------*
         LA    R7,SRLGDFSI         POINT TO THE FSI GETDS AREA
         USING GDSPARM,R7           AND SET ADDRESSABILITY
         LA    R6,SRLGDJSP         POINT TO THE JSPA AREA
         USING JSPA,R6              AND SET ADDRESSABILITY
         LA    R5,JSPASIZE(,R6)
         USING JSPEXT,R5
         CLC   JSPCESEC,=XL8¢0000000000000000¢
         BNE   ENDIT
         MVC   JSPCESEC,=CL8¢DEFLABEL¢
         SPACE 2
         DROP  R6,R7,R8            JSPA,GDSPARM,SRLSTART
*-------------------------------------------------------------*
*     RETURN +16: DUMMY EXIT - FUTURE CALLS ARE SUPPRESSED    *
*                              UNTIL THE NEXT SYSTEM RESTART  *
*-------------------------------------------------------------*
ENDIT    ARETURN RC=0              RETURN TO FSS WIR DSP
         EJECT
         IATXPTCH LT
APARNUM  DC    CL7¢       ¢
PTFNUM   DC    CL7¢SP311  ¢
         END   IATUX45
```

Figure 70 (Part 5 of 5). JES3 User Exit 45 to Define Default Security Label

Appendix I. SDSF Resource Names Tables

Tables 31 through 35 contain a list of all resource names in the SDSF and OPERCMDS class you need to use SAF security. Parenthetical numbers in a table refer to notes at the end pf the table.

Table 36. SDSF Class Resource Names and SDSF Authorized Commands

SDSF Class Resource Name	Class	Command	Required Access
ISFCMD.DSP.ACTIVE.JESx	SDSF	DA	READ
ISFCMD.DSP.HELD.JESx	SDSF	H	READ
ISFCMD.DSP.INPUT.JESx	SDSF	I	READ
ISFCMD.DSP.OUTPUT.JESx	SDSF	O	READ
ISFCMD.DSP.STATUS.JESx	SDSF	ST	READ
ISFCMD.FILTER.ACTION	SDSF	ACTION	READ
ISFCMD.FILTER.DEST	SDSF	DEST	READ
ISFCMD.FILTER.FINDLIM	SDSF	FINDLIM	READ
ISFCMD.FILTER.INPUT	SDSF	INPUT	READ
ISFCMD.FILTER.OWNER	SDSF	OWNER	READ
ISFCMD.FILTER.PREFIX	SDSF	PREFIX	READ
ISFCMD.FILTER.SYSID	SDSF	SYSID	READ
ISFCMD.MAINT.ABEND	SDSF	ABEND	READ
ISFCMD.MAINT.TRACE	SDSF	TRACE	READ
ISFCMD.ODSP.INITIATOR.JESx	SDSF	INIT	READ
ISFCMD.ODSP.SYSLOG.JESx	SDSF	LOG	READ
ISFCMD.ODSP.PRINTER.JESx	SDSF	PR	READ

Table 37 (Page 1 of 3). Overtypeable Fields. For the fields to be overtypeable for the user, UPDATE authority is needed to the SDSF resources that protect the fields. Note that the SDSF Resource Name is composed of three qualifiers, but is broken into three lines. It should be interpreted as one line, such as ISFATTR.OUTPUT.CLASS.

Overtypeable Field	**SDSF Resource Name** (*UPDATE REQUIRED*)	**JES2/MVS Command**	**OPERCMDS Resource Name**	**OPERCMDS Required Access**	**SDSF Panel**
C	ISFATTR. OUTPUT. CLASS	$TO (Set output) SSI	*JESx*.MODIFY.*type*OUT●	CONTROL●	O H●
C	ISFATTR. JOB. CLASS	$T (Set)	*JESx*.MODIFY.*type*●	UPDATE	I
CKPTLINE	ISFATTR. PROPTS. CKPTLINE	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
CKPTMODE	ISFATTR. PROPTS. CKPTMODE	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
CKPTPAGE	ISFATTR. PROPTS. CKPTPAGE	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
CKPTSEC	ISFATTR. PROPTS. CKPTSEC	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
CLASSES	ISFATTR. SELECT. JOBCLASS	$T (Set)	*JESx*.MODIFY.INITIATOR	CONTROL	INIT
CPYMOD	ISFATTR. PROPTS. COPYMOD	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
DEST	ISFATTR. OUTPUT. DEST	$TO (set output)	*JESx*.MODIFY.*type*OUT●	CONTROL●	O
DEST	ISFATTR. OUTPUT. DEST	$O (Set output)	*JESx*.RELEASE.*type*OUT●	UPDATE	H●
DEST	ISFATTR. OUTPUT. DEST	SSI	*JESx*.MODIFY.*type*OUT●	CONTROL●	H●
FCB	ISFATTR. OUTPUT. FCB	$TO (Set output)	*JESx*.MODIFY.*type*OUT●	CONTROL●	O
FLASH	ISFATTR. OUTPUT. FLASH	$TO (Set output)	*JESx*.MODIFY.*type*OUT●	CONTROL●	O
FORM	ISFATTR. OUTPUT. FORMS	$TO (Set output)	*JESx*.MODIFY.*type*OUT●	CONTROL●	O
M	ISFATTR. PROPTS. MARK	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR

Table 37 (Page 2 of 3). Overtypeable Fields. For the fields to be overtypeable for the user, UPDATE authority is needed to the SDSF resources that protect the fields. Note that the SDSF Resource Name is composed of three qualifiers, but is broken into three lines. It should be interpreted as one line, such as ISFATTR.OUTPUT.CLASS.

Overtypeable Field	**SDSF Resource Name** (*UPDATE REQUIRED*)	**JES2/MVS Command**	**OPERCMDS Resource Name**	**OPERCMDS Required Access**	**SDSF Panel**
MODE	ISFATTR. PROPTS. MODE	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
NPRO	ISFATTR. PROPTS. NPRO.	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
PGN	ISFATTR. JOB. PGN	E (MVS reset)	MVS.RESET	UPDATE	DA
PRTDEST	ISFATTR. JOB. PRTDEST	$R (Route)	*JESx*.ROUTE.*JOB*OUT	UPDATE	I ST
PRTY	ISFATTR. JOB. PRTY	$T (Set)	*JESx*.MODIFY.*type*●	UPDATE	I ST
PRTY	ISFATTR. OUTPUT. PRTY	$TO (Set output)	*JESx*.MODIFY.*type*OUT●	CONTROL●	O
SBURST	ISFATTR. SELECT. BURST	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SCLASS	ISFATTR. SELECT. CLASS	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SDEST1	ISFATTR. SELECT. DEST	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SEP	ISFATTR. PROPTS. SEP	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SEPDS	ISFATTR. PROPTS. SEPDS	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SFCB	ISFATTR. SELECT. FCB	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SFORM	ISFATTR. SELECT. FORMS	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SFLH	ISFATTR. SELECT. FLASH	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SJOBNAME	ISFATTR. SELECT. JOBNAME	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
SUCS	ISFATTR. SELECT. UCS	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR

Table 37 (Page 3 of 3). Overtypeable Fields. For the fields to be overtypeable for the user, UPDATE authority is needed to the SDSF resources that protect the fields. Note that the SDSF Resource Name is composed of three qualifiers, but is broken into three lines. It should be interpreted as one line, such as ISFATTR.OUTPUT.CLASS.

Overtypeable Field	**SDSF Resource Name** (*UPDATE REQUIRED*)	**JES2/MVS Command**	**OPERCMDS Resource Name**	**OPERCMDS Required Access**	**SDSF Panel**
SWRITER	ISFATTR. SELECT. WRITER	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
UCS	ISFATTR. OUTPUT. UCS	$TO (Set output)	*JESx*.MODIFY.*type*OUT●	CONTROL●	O
WORK-SELECTION	ISFATTR. PROPTS. WS	$T (Set)	*JESx*.MODIFY.DEV	UPDATE	PR
WTR	ISFATTR. OUTPUT. WRITER	$TO (Set output)	*JESx*.MODIFY.*type*OUT●	CONTROL●	O

Notes:

• On the Held Output Queue panel, SDSF uses the subsystem interface (SSI) when you overtype SYSOUT class (C) or DEST or when you enter an O, C, or P action character. You can change the class or destination without releasing the output. In order to release output when the JESSPOOL class is enabled, the user must have ALTER authority to the JESSPOOL resource. This authority is implied for the JESSPOOL resources created by the user.

• *JESx* should be replaced by the name of the targeted JES2 subsystem, and *type* should be replaced by the name of the corresponding object type (that is, BAT, STC, and TSU for batch jobs, started tasks, or TSO users, respectively). As an example, use *JESx.MODIFY.BATOUT*.

SDSF only references BAT, STC, and TSU resources, although other types exist outside of SDSF that it does not use. By using "%%%" to replace type in the profile name, the installation may be authorizing users to use more resources than intended.

• UPDATE access can be used for BAT type work objects but CONTROL access authority is the highest level of authority required by this resource for other types of work objects.

• This occurs only on a secondary JES system. Otherwise, SDSF uses SSI.

• If you cancel or purge a TSO job on the DA, I, or ST panels, SDSF issues the MVS command, which is issued as C U=userid.

Table 38 (Page 1 of 3). Overtypeable Fields by Resource Name. For the fields to be overtypeable for the user, UPDATE authority is needed to the SDSF resources that protect the fields. Note that the SDSF Resource Name is composed of three qualifiers, but is broken into three lines. It should be interpreted as one line, such as ISFATTR.PROPTS.CKPTLINE.

OPERCMDS Resource Name	**OPERCMDS Required Access**	**JES2/MVS Command**	**Overtypeable Field**	**SDSF Resource Name** (*UPDATE REQUIRED*)	**SDSF Panel**
JESx.MODIFY.DEV	UPDATE	$T (Set)	CKPTLINE	ISFATTR. PROPTS. CKPTLINE	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	CKPTMODE	ISFATTR. PROPTS. CKPTMODE	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	CKPTPAGE	ISFATTR. PROPTS. CKPTPAGE	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	CKPTSEC	ISFATTR. PROPTS. CKPTSEC	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	CPYMOD	ISFATTR. PROPTS. COPYMOD	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SBURST	ISFATTR. SELECT. BURST	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SEP	ISFATTR. PROPTS. SEP	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SEPDS	ISFATTR. PROPTS. SEPDS	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SFCB	ISFATTR. SELECT. FCB	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SFORM	ISFATTR. PROPTS. FORMS	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SCLASS	ISFATTR. SELECT. CLASS	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SDEST1	ISFATTR. SELECT. DEST	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SFLH	ISFATTR. SELECT. FLASH	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SJOBNAME	ISFATTR. SELECT. JOBNAME	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SUCS	ISFATTR. SELECT. UCS	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	SWRITER	ISFATTR. SELECT. WRITER	PR

Table 38 (Page 2 of 3). Overtypeable Fields by Resource Name. For the fields to be overtypeable for the user, UPDATE authority is needed to the SDSF resources that protect the fields. Note that the SDSF Resource Name is composed of three qualifiers, but is broken into three lines. It should be interpreted as one line, such as ISFATTR.PROPTS.CKPTLINE.

OPERCMDS Resource Name	**OPERCMDS Required Access**	**JES2/MVS Command**	**Overtypeable Field**	**SDSF Resource Name** (*UPDATE REQUIRED*)	**SDSF Panel**
JESx.MODIFY.DEV	UPDATE	$T (Set)	M	ISFATTR. SELECT. MARK	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	MODE	ISFATTR. PROPTS. MODE	PR
JESx.MODIFY.DEV	UPDATE	$T (Set)	NPRO	ISFATTR. PROPTS. NPRO	PR
JESx.MODIFY.INITIATOR	CONTROL●	$T (Set)	CLASSES	ISFATTR. SELECT. JOBCLASS	Init
JESx.MODIFY.*type*●	UPDATE	$T (Set)	C	ISFATTR. JOB. CLASS	I INIT
JESx.MODIFY.*type*●	UPDATE	$T (Set)	PRTY	ISFATTR. JOB. PRTY	I ST
JESx.MODIFY.*type*OUT●	CONTROL●	$TO (Set output)	FORM	ISFATTR. OUTPUT. FORMS	O
JESx.MODIFY.*type*OUT● *JESx*.MODIFY.*type*OUT●	CONTROL● CONTROL●	$TO (Set output) SSI	DEST	ISFATTR. OUTPUT. DEST	O H●
JESx.MODIFY.*type*OUT●	CONTROL●	$TO (Set output)	FCB	ISFATTR. OUTPUT. FCB	O
JESx.MODIFY.*type*OUT●	CONTROL●	$TO (Set output)	FLASH	ISFATTR OUTPUT. FLASH	O
JESx.MODIFY.*type*OUT●	CONTROL●	$TO (Set output) SSI	C	ISFATTR. OUTPUT. CLASS	O H●
JESx.MODIFY.*type*OUT●	CONTROL●	$TO (Set output)	PRTY	ISFATTR. OUTPUT. PRTY	O
JESx.MODIFY.*type*OUT●	CONTROL●	$TO (Set output)	UCS	ISFATTR. OUTPUT. UCS	O
JESx.MODIFY.*type*OUT●	CONTROL●	$TO (Set output)	WTR	ISFATTR. OUTPUT. WRITER	O

Table 38 (Page 3 of 3). Overtypeable Fields by Resource Name. For the fields to be overtypeable for the user, UPDATE authority is needed to the SDSF resources that protect the fields. Note that the SDSF Resource Name is composed of three qualifiers, but is broken into three lines. It should be interpreted as one line, such as ISFATTR.PROPTS.CKPTLINE.

OPERCMDS Resource Name	**OPERCMDS Required Access**	**JES2/MVS Command**	**Overtypeable Field**	**SDSF Resource Name** (*UPDATE REQUIRED*)	**SDSF Panel**
JESx.RELEASE.*type*OUT●	UPDATE	$O	DEST	ISFATTR. OUTPUT. DEST	H●
JESx.ROUTE.JOBOUT	UPDATE	$R(route)	PRTDEST	ISFATTR. JOB. PRTDEST	I ST
JESx.MODIFY.DEV	UPDATE	$T (Set)	WORK- SELECTION	ISFATTR. PROPTS. WS	PR
MVS.RESET	UPDATE	E (MVS reset)	PGN	ISFATTR. JOB. PGN.	DA

Notes: Refer to notes for Table 37.

.

Table 39. Action Characters. In this table, many action characters have more than one OPERCMDS resource name associated with them. The names vary according to the panel. Choose the OPERCMDS resource name that is related to the panel for which action character access is being given.

Action Character	JES2/MVS Command	OPERCMDS Resource Name	OPERCMDS Required Access	SDSF Panel
A	$TO	*JESx*.MODIFY.*type*OUT●	CONTROL●	O
A	$A (release)	*JESx*.MODIFYRELEASE.*type*●	UPDATE	DA I ST
Bnnn	$B(backspace)	*JESx*.BACKSP.DEV	UPDATE	PR
C	C U=userid $C $O,cancel SSI	MVS.CANCEL.TSU.* *JESx*.CANCEL.*type*● *JESx*.CANCEL.DEV● *JESx*.RELEASE.*type*OUT● *JESx*.CANCEL.*type*●	UPDATE UPDATE UPDATE UPDATE UPDATE	DA I ST● DA I O ST PR H● H●
CD	C U=,DUMP $Cxxxx,D	MVS.CANCEL.TSU.* *JESx*.CANCEL.*type*●	UPDATE UPDATE	DA I ST● DA I ST
D	$D (Display) $M (send msg.)●	*JESx*.DISPLAY.*type*● *JESx*.DISPLAY.INITIATOR *JESx*.DISPLAY.DEV *JESx*.MSEND.CMD	READ READ READ READ	ST I DA INIT PR ST I
E	$E (restart)	*JESx*.RESTART.DEV *JESx*.RESTART.BAT	UPDATE CONTROL	PR ST I DA
Fnnn	$F(forward space)	*JESx*.FORWARD.DEV	UPDATE	PR
H	$H (hold) $TO (set output)	*JESx*.MODIFYHOLD.*type*● *JESx*.MODIFY.*type*OUT●	UPDATE CONTROL●	DA I ST O
I	$I (interrupt)	*JESx*.INTERRUPT.DEV	UPDATE	PR
L	$L (list)	*JESx*.DISPLAY.*type*OUT●	READ	ST I O DA
N	$N (repeat)	*JESx*.REPEAT.DEV	UPDATE	PR
O	$O (release) SSI●	*JESx*.RELEASE.*type*OUT● *JESx*.MODIFY.*type*OUT●	UPDATE CONTROL●	H● ST H●
P	$P (stop)	*JESx*.STOP.DEV *JESx*.STOP.INITIATOR	UPDATE CONTROL	PR INIT
P	C U=userid $C,purge $O,cancel SSI	MVS.CANCEL.TSU.* *JESx*.CANCEL.*type*● *JESx*.RELEASE.*type*OUT● *JESx*.CANCEL.*type*●	UPDATE UPDATE UPDATE UPDATE	DA I ST● DA I O ST H● H●
S	$S (start)	*JESx*.START.DEV *JESx*.START.INITIATOR	UPDATE CONTROL	PR INIT
S (browse)				DA, H, I, JDS, O, ST
V (view)				JDS
Z	$Z (halt)	*JESx*.HALT.DEV *JESx*.HALT.INITIATOR	UPDATE CONTROL	PR INIT
?				DA, H, I, O, ST

Notes:

- On the Held Output Queue panel, SDSF uses the subsystem interface (SSI) when you overtype SYSOUT class (C) or DEST, or you enter an O, C, or P action character. You can change the class or destination without releasing the output. In order to release output when the JESSPOOL class is enabled, the user must have ALTER authority to the JESSPOOL resource. This authority is implied for the JESSPOOL resources created by the user.

- *JESx* should be replaced by the name of the targeted JES2 subsystem, and *type* should be replaced by the name of the corresponding object type (that is, BAT, STC, and TSU for batch jobs, started tasks, and TSO users, respectively). An example is: *JESx.MODIFY.BATOUT*.

SDSF only references BAT, STC, and TSU resources, although other types exist outside of SDSF that it does not use. By using "%%%" to replace type in the profile name, the installation may be authorizing users to access more resources than intended.

- UPDATE access can be used for BAT type work objects but CONTROL authority is the highest level of authority required by this resource for other types of work objects.

- This occurs only on a secondary JES system. Otherwise, SDSF uses SSI.

- If you cancel or purge a TSU job on the DA, I, or ST panels, SDSF issues the MVS command, which is issued as C U=userid.

- This command is issued only to cancel active users on another CPU using the MVS CANCEL command in a MAS environment.

.

Table 40 (Page 1 of 2). Action Characters by OPERCMDS Resource Name. In this table, many action characters have more than one OPERCMDS resource name associated with them. The names vary according to the panel. Choose the OPERCMDS resource name that is related to the panel for which action character access is being given.

OPERCMDS Resource Name	OPERCMDS Required Access	Action Character	JES2/MVS Command	SDSF Panel
JESx.BACKSP.DEV	UPDATE	Bnnn	$B(backspace)	PR
JESx.CANCEL.DEV	UPDATE	C	$C	PR
JESx.CANCEL.*type*●	UPDATE	C CD P	$C SSI $Cxxxx,D $C,purge	DA I O ST H● DA, I, ST● DA, I, O, ST●
JESx.CANCEL.*type*OUT●	UPDATE	P	SSI	H●
JESx.DISPLAY.*type*OUT●	READ	L	$L (list)	ST I O DA
JESx.DISPLAY.*type*● *JESx*.DISPLAY.INITIATOR *JESx*.DISPLAY.DEV	READ READ READ	D	$D (Display)	ST I DA INIT PR
JESx.FORWARD.DEV	UPDATE	Fnnn	$F(forward space)	PR
JESx.HALT.DEV *JESx*.HALT.INITIATOR	UPDATE CONTROL	Z	$Z (halt)	PR INIT

Table 40 (Page 2 of 2). Action Characters by OPERCMDS Resource Name. In this table, many action characters have more than one OPERCMDS resource name associated with them. The names vary according to the panel. Choose the OPERCMDS resource name that is related to the panel for which action character access is being given.

OPERCMDS Resource Name	OPERCMDS Required Access	Action Character	JES2/MVS Command	SDSF Panel
JESx.INTERRUPT.DEV	UPDATE	I	$I (interrupt)	PR
JESx.MODIFY.*type*OUT●	CONTROL●	A H	$TO $TO (set output)	O O
JESx.RELEASE.*type*OUT●	CONTROL● UPDATE	O	SSI	H●
JESx.MODIFYHOLD.*type*●	UPDATE	H	$H (hold)	DA I ST
JESx.MODIFYRELEASE.*type*●	UPDATE	A	$A (release)	DA I ST
JESx.MSEND.CMD●	READ	(see Note 6)	$M (send msg.)	I ST
MVS.CANCEL.TSU.*	UPDATE	C CD P	C U=userid C U=, DUMP C U=userid	DA I ST● DA I ST DA I ST●
JESx.RELEASE.*type*OUT●	UPDATE	C O P	$O cancel $O (release) $O cancel	H● H● ST H●
JESx.REPEAT.DEV	UPDATE	N	$N (repeat)	PR
JESx.RESTART.DEV *JESx*.RESTART.BAT	UPDATE CONTROL	E	$E (restart)	PR ST I DA
JESx.START.DEV *JESx*.START.INITIATOR	UPDATE CONTROL	S	$S (start)	PR INIT
JESx.STOP.DEV *JESx*.STOP.INITIATOR	UPDATE CONTROL	P	$P (stop)	PR INIT

Note: The subscripts are the same as for Table 39 on page 302.

Appendix J. Partner LU 6.2 Test Output

The following table is an index to the partner LU 6.2 test case output provided in this appendix.

Test Case	Figure	Notes
Test-1	Figure 71 on page 306	The console messages indicate that both JOBs ran successfully.
Test-2	Figure 72 on page 308	Reason code=04 indicates that the session key does not match the partner LU session key.
Test-3	Figure 73 on page 309	Reason code=06 indicates that partner LU verification was requested but there is no session key.
Test-4	Figure 74 on page 309	Reason code=12 indicates that a profile was found with an expired session key.

Refer to RACF messages and codes for a description of ICH415I.

```
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
IAT6101 (JOB03631) JOB APPCTST1 (JOB03695), PRTY=01
IAT6101 (JOB03503) JOB APPCTST2 (JOB03696), PRTY=01
IAT2000 JOB APPCTST1 (JOB03695) SELECTED SY1      GRP=A
IAT2000 JOB APPCTST2 (JOB03696) SELECTED SY1      GRP=A
SY1= APPCTST1 ICH70001I P0112NZ  LAST ACCESS AT 09 46 37 ON WEDNESDAY, ..     Y 28, 1990
SY1= APPCTST2 ICH70001I P0112NZ  LAST ACCESS AT 09 52 53 ON WEDNESDAY, ..     Y 28, 1990
SY1= APPCTST1 +-APPCPGM1M OPENING OUTPUT SYSPRNTM
SY1= APPCTST2 +-APPCPGM2M OPENING OUTPUT SYSPRNTM
SY1= APPCTST2 +OPENING ACB APPCPGM2
SY1= APPCTST1 +OPENING ACB APPCPGM1
SY1= APPCTST2 +OPENED  ACB
SY1= APPCTST1 +OPENED  ACB
SY1= APPCTST2 +--APPCPGM2 ISSUED SETLOGON
SY1= APPCTST1 +--APPCPGM1 ISSUED SETLOGON
SY1= APPCTST2 +--APPCPGM2 ISSUING APPCCMD CNOS
SY1= APPCTST1 + TESTING TO SEE IF WE ARE APPCPGM1
SY1= APPCTST1 +WE ARE SO MOVING THE TEST FILE
SY1= APPCTST1 +GOING TO MAINWAIT
SY1= APPCTST1 +--APPCPGM1 ATTN CNOS APPCPGM2
SY1= APPCTST1 +--APPCPGM1 ATTN EXIT APPCPGM2 ADDED TO TABLE
SY1= APPCTST2 +--APPCPGM2 ISSUED APPCCMD CNOS OK
SY1= APPCTST2 + TESTING TO SEE IF WE ARE APPCPGM1
SY1= APPCTST2 +WE ARE APPCPGM2, SO WE START DAILOGA
SY1= APPCTST2 +NOW FOUND A FREE TASK AM GOING TO START IT
SY1= APPCTST2 +--APPCPGM2 ATTACH SUBTASK1 DIALOGA
SY1= APPCTST2 +DIALOGA NOW IN CONTROL, CREATING FMH5
SY1= APPCTST2 +GOING TO MAINWAIT
SY1= APPCTST2 +FMH5 CREATED
SY1= APPCTST2 +--APPCPGM2 ISSUE ALLOCD APPCPGM1
SY1= APPCTST2 +ABOUT TO SNAP THE RPL AREA
SY1= APPCTST2 +AND NOW THE FMH5 AREA
SY1= APPCTST2 +NOW ISSUING THE ALLOC
SY1= APPCTST2 +--APPCPGM2 ALLOCD OK    APPCPGM1
SY1= APPCTST2 + CONVERSATION ID MOVED
SY1= APPCTST2 +SENDING FIRST MESSAGE
SY1= APPCTST2 +DIALOGA  FIRST SEND COMPLETED
SY1= APPCTST2 +DIALOGA  NOW ISSUING RECEIVE
SY1= APPCTST2 +DIALOGA  ISSUING RECEIVE MACRO
SY1= APPCTST1 +--APPCPGM1 ATTN FMH5 APPCPGM2
SY1= APPCTST1 +--APPCPGM1 RCVD TPN  DIALOGP
SY1= APPCTST1 +MAINLIST POSTED
SY1= APPCTST1 +--APPCPGM1 ATTACH SUBTASK1 DIALOGP
SY1= APPCTST1 +DIALOGP  ISSUING FIRST RECEIVE
SY1= APPCTST1 +DIALOGP  RECEIVE POSTED
SY1= APPCTST1 +MESSAGE1 RECEIVED
SY1= APPCTST1 +DIALOGP  ISSUING SEND FOR MESSAGE 1
SY1= APPCTST1 +DIALOGP  SEND POSTED
SY1= APPCTST2 +DIALOGA RECEIVE POSTED
SY1= APPCTST2 +MESSAGE1 RECEIVED
SY1= APPCTST2 +DIALOGA  NOW SENDING MESSAGE 2
SY1= APPCTST2 +DIALOGA  SEND POSTED
SY1= APPCTST2 +DIALOGA  ISSUING RECEIVE MACRO
SY1= APPCTST1 +DIALOGP  RECEIVE POSTED
SY1= APPCTST1 +MESSAGE2 RECEIVED
SY1= APPCTST1 +DIALOGP  ISSUING MSG2 AND DEALLOCING
SY1= APPCTST2 +DIALOGA RECEIVE POSTED
SY1= APPCTST1 +DIALOGP  DEALLOC POSTED
SY1= APPCTST1 +DIALOGP  FILE TRANSFER WILL NOW BE STARTED
SY1= APPCTST2 +DIALOGA  I HAVE ASSUMED THAT THE LAST MESSAGE WAS MSG2
```

Figure 71 (Part 1 of 2). Console Messages for Test-1

```
SY1= APPCTST2 +TESTING FOR DEALLOC ALREADY ISSUED BY PARTNER
SY1= APPCTST1 +MAINLIST POSTED
SY1= APPCTST2 +MAINLIST POSTED
SY1= APPCTST1 +--APPCPGM1 ATTACH SUBTASK1 FILETRNA
SY1= APPCTST1 +FILETRNA NOW IN CONTROL, CREATING FMH5
SY1= APPCTST1 +FILETRNA  FMH5 CREATED
SY1= APPCTST1 +--APPCPGM1 ISSUE ALLOCD APPCPGM2
SY1= APPCTST1 +ABOUT TO SNAP THE RPL AREA
SY1= APPCTST1 +AND NOW THE FMH5 AREA
SY1= APPCTST1 +NOW ISSUING THE ALLOC
SY1= APPCTST1 +--APPCPGM1 ALLOCD OK    APPCPGM2
SY1= APPCTST1 + CONVERSATION ID MOVED
SY1= APPCTST1 +SENDING FIRST DATA RECORD
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST2 +--APPCPGM2 ATTN FMH5 APPCPGM1
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST2 +--APPCPGM2 RCVD TPN  FILETRNP
SY1= APPCTST2 +MAINLIST POSTED
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST2 +--APPCPGM2 ATTACH SUBTASK1 FILETRNP
SY1= APPCTST2 +FILETRNP HERE JUST CHECKING MY FILE STATUS
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST2 +FILETRNP  ISSUING FIRST RECEIVE FOR PIP
SY1= APPCTST2 +FILETRNP  ISSUING RECEIVE
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST1 +FILETRNA  SEND COMPLETED
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST1 +FILETRNA  ALL DATA SENT
SY1= APPCTST1 +FILETRNA  TRYING TO DEALLOCATE THE CONVERSATION
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +ISSUING RECEIVE
SY1= APPCTST2 +FILETRNP RECEIVE POSTED
SY1= APPCTST2 +FILETRNP  ISSUING CONFIRM
SY1= APPCTST1 +FILETRNA  CONVERSATION DEALLOCATED
SY1= APPCTST2 +FILETRNP  GOOD BYE FROM ME
SY1= APPCTST1 +FILETRNA  RETURNING ALL DONE
SY1= APPCTST2 +MAINLIST POSTED
SY1= APPCTST1 +MAINLIST POSTED
```

Figure 71 (Part 2 of 2). Console Messages for Test-1

```
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
IAT6101 (JOB03631) JOB APPCTST1 (JOB03700), PRTY=01
IAT6101 (JOB03503) JOB APPCTST2 (JOB03701), PRTY=01
IAT2000 JOB APPCTST1 (JOB03700) SELECTED SY1     GRP=A
IAT2000 JOB APPCTST2 (JOB03701) SELECTED SY1     GRP=A
SY1= APPCTST1 ICH70001I P0112NZ  LAST ACCESS AT 10 45 43 ON WEDNESDAY, FEBRUARY 28, 1990
SY1= APPCTST2 ICH70001I P0112NZ  LAST ACCESS AT 10 46 00 ON WEDNESDAY, FEBRUARY 28, 1990
SY1= APPCTST1 +-APPCPGM1M OPENING OUTPUT SYSPRNTM
SY1= APPCTST2 +-APPCPGM2M OPENING OUTPUT SYSPRNTM
SY1= APPCTST1 +OPENING ACB APPCPGM1
SY1= APPCTST1 +OPENED  ACB
SY1= APPCTST2 +OPENING ACB APPCPGM2
SY1= APPCTST1 +--APPCPGM1 ISSUED SETLOGON
SY1= APPCTST1 + TESTING TO SEE IF WE ARE APPCPGM1
SY1= APPCTST1 +WE ARE SO MOVING THE TEST FILE
SY1= APPCTST1 +GOING TO MAINWAIT
SY1= APPCTST2 +OPENED  ACB
SY1= APPCTST2 +--APPCPGM2 ISSUED SETLOGON
SY1= APPCTST2 +--APPCPGM2 ISSUING APPCCMD CNOS
SY1= VTMLCL   IST970I LU-LU VERIFICATION ERROR 04 FOR USIBMSC.APPCPGM2.APPCPGM1
SY1=          ICH415I SESSION ATTEMPT REJECTED. REASON CODE = 04
SY1=          ICH415I ENTITY  USIBMSC.APPCPGM2.APPCPGM1
SY1=          ICH415I PROFILE
SY1=          ICH415I AT  10 46 01  ON  02/28/90
SY1=          ICH415I SESSION ATTEMPT REJECTED. REASON CODE = 05
SY1=          ICH415I ENTITY  USIBMSC.APPCPGM1.APPCPGM2
SY1=          ICH415I PROFILE
SY1=          ICH415I AT  10 46 01  ON  02/28/90
SY1= APPCTST1 +--APPCPGM1 ATTN LOSS APPCPGM2
SY1= APPCTST2 IEA995I SYMPTOM DUMP OUTPUT
SY1= APPCTST2   USER COMPLETION CODE=0025
SY1= APPCTST2  TIME=10.46.01  SEQ=00213  CPU=0000  ASID=0019
SY1= APPCTST2  PSW AT TIME OF ERROR  078D1000   0000703C  ILC 2  INTC 0D
SY1= APPCTST2    ACTIVE LOAD MODULE=APPCVTAM  ADDRESS=00006C60  OFFSET=000003DC
SY1= APPCTST2    DATA AT PSW  00007036 - 00181610  0A0DD207  C3EECCE1
SY1= APPCTST2    GPR  0-3  80000000  80000019  00005FF1  00000008
SY1= APPCTST2    GPR  4-7  00000000  00007A2C  0000848C  0000794C
SY1= APPCTST2    GPR  8-11 000079BC  80AFABB8  00000000  00007C60
SY1= APPCTST2    GPR 12-15 00006C60  000076B0  80006F98  92DC6970
SY1= APPCTST2  END OF SYMPTOM DUMP
SY1= VTMLCL   IST804I CLOSE IN PROGRESS FOR APPCPGM2 OPENED BY APPCTST2
SY1= VTMLCL   IST400I TERMINATION IN PROGRESS FOR APPLID APPCPGM2
SY1= APPCTST2 IEF450I APPCTST2 STEP2 - ABEND=S000 U0025 REASON=00000000
SY1= VTMLCL   IST805I VTAM CLOSE COMPLETE FOR APPCPGM2
SE 'IAT6108 JOB APPCTST2 (JOB03701) ENDED,COMP CD=S000 U0025',USER=(P0112NZ),LOGON
SE 'IAT6108 JOB APPCTST2 STEP=STEP2   ,PROC=NONE    ',USER=(P0112NZ),LOGON
```

Figure 72. Console Messages for Test-2

```
SY1= VTMLCL   IST097I VARY ACCEPTED
SY1= VTMLCL   IST093I APPCMJLR ACTIVE
SY1=          IEA989I SLIP TRAP ID=X13E MATCHED
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
IAT6101 (JOB03805) JOB APPCTST1 (JOB03820), PRTY=01
IAT6101 (JOB03801) JOB APPCTST2 (JOB03821), PRTY=01
IAT2000 JOB APPCTST1 (JOB03820) SELECTED SY1     GRP=A
SY1= APPCTST1 ICH70001I P0112NZ  LAST ACCESS AT 12 30 08 ON THURSDAY, MARCH 1, 1990
IAT2000 JOB APPCTST2 (JOB03821) SELECTED SY1     GRP=A
SY1= APPCTST1 +-APPCPGM1M OPENING OUTPUT SYSPRNTM
SY1= APPCTST2 ICH70001I P0112NZ  LAST ACCESS AT 12 34 09 ON THURSDAY, MARCH 1, 1990
SY1= APPCTST2 +-APPCPGM2M OPENING OUTPUT SYSPRNTM
SY1= APPCTST1 +OPENING ACB APPCPGM1
SY1= APPCTST1 +OPENED  ACB
SY1= APPCTST1 +--APPCPGM1 ISSUED SETLOGON
SY1= APPCTST1 + TESTING TO SEE IF WE ARE APPCPGM1
SY1= APPCTST1 +WE ARE SO MOVING THE TEST FILE
SY1= APPCTST1 +GOING TO MAINWAIT
SY1= APPCTST2 +OPENING ACB APPCPGM2
SY1= APPCTST2 +OPENED  ACB
SY1= APPCTST2 +--APPCPGM2 ISSUED SETLOGON
SY1= APPCTST2 +--APPCPGM2 ISSUING APPCCMD CNOS
SY1=          ICH415I SESSION ATTEMPT REJECTED. REASON CODE = 06
SY1= VTMLCL   IST970I LU-LU VERIFICATION ERROR 06 FOR USIBMSC.APPCPGM2.APPCPGM1
SY1=          ICH415I ENTITY  USIBMSC.APPCPGM2.APPCPGM1
SY1=          ICH415I PROFILE
SY1=          ICH415I AT  12 34 09  ON  03/01/90
SY1= VTMLCL   IST663I CINIT REQUEST FAILED, SENSE=080F6051
```

Figure 73. Console Messages for Test-3

```
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
 IRR010I USERID P0112NZ  IS ASSIGNED TO THIS JOB.
IAT6101 (JOB03805) JOB APPCTST1 (JOB03839), PRTY=01
IAT6101 (JOB03801) JOB APPCTST2 (JOB03840), PRTY=01
IAT2000 JOB APPCTST1 (JOB03839) SELECTED SY1     GRP=A
IAT2000 JOB APPCTST2 (JOB03840) SELECTED SY1     GRP=A
SY1= APPCTST1 ICH70001I P0112NZ  LAST ACCESS AT 18 34 36 ON THURSDAY, MARCH 1, 1990
SY1= APPCTST1 +-APPCPGM1M OPENING OUTPUT SYSPRNTM
SY1= APPCTST2 ICH70001I P0112NZ  LAST ACCESS AT 18 34 45 ON THURSDAY, MARCH 1, 1990
SY1= APPCTST2 +-APPCPGM2M OPENING OUTPUT SYSPRNTM
SY1= APPCTST1 +OPENING ACB APPCPGM1
SY1= APPCTST1 +OPENED  ACB
SY1= APPCTST1 +--APPCPGM1 ISSUED SETLOGON
SY1= APPCTST1 + TESTING TO SEE IF WE ARE APPCPGM1
SY1= APPCTST1 +WE ARE SO MOVING THE TEST FILE
SY1= APPCTST1 +GOING TO MAINWAIT
SY1= APPCTST2 +OPENING ACB APPCPGM2
SY1= APPCTST2 +OPENED  ACB
SY1= APPCTST2 +--APPCPGM2 ISSUED SETLOGON
SY1= APPCTST2 +--APPCPGM2 ISSUING APPCCMD CNOS
SY1=          ICH415I SESSION ATTEMPT REJECTED. REASON CODE = 12
SY1= VTMLCL   IST970I LU-LU VERIFICATION ERROR 0C FOR USIBMSC.APPCPGM2.APPCPGM1
SY1=          ICH415I ENTITY  USIBMSC.APPCPGM2.APPCPGM1
SY1=          ICH415I PROFILE
SY1=          ICH415I AT  18 34 46  ON  03/01/90
SY1= VTMLCL   IST663I CINIT REQUEST FAILED, SENSE=080F6051
```

Figure 74. Console Messages for Test-4

Appendix K. DLF Facility - COBOL Utility Source Listing

```
 IDENTIFICATION DIVISION.                                                00010000
   PROGRAM-ID. COBUTIL.                                                  00020000
   INSTALLATION. IBM - POUGHKEEPSIE.                                     00030000
 ENVIRONMENT DIVISION.                                                   00040000
 CONFIGURATION SECTION.                                                  00050000
   SOURCE-COMPUTER. IBM-370.                                             00060000
   OBJECT-COMPUTER. IBM-370.                                             00070000
   SPECIAL-NAMES.                                                        00080000
     CONSOLE IS OPERATOR.                                                00090000
 INPUT-OUTPUT SECTION.                                                   00100000
   FILE-CONTROL.                                                         00110000
*                                                                        00120000
     SELECT UPDT                                                         00130000
         ASSIGN UPDT                                                     00140000
         ACCESS SEQUENTIAL.                                              00150000
*                                                                        00160000
     SELECT VSAMMAST                                                     00170000
         ASSIGN VSAMMAST                                                 00180000
         ORGANIZATION INDEXED                                            00190000
         ACCESS DYNAMIC                                                  00200000
         RECORD KEY VSAMMAST-KEY                                         00210000
         FILE STATUS VSAMMAST-FILE-STATUS VSAMMAST-VSAM-STATUS.          00220000
*                                                                        00230000
     SELECT PFILE                                                        00240000
         ASSIGN PFILE.                                                   00250000
*                                                                        00260000
 DATA DIVISION.                                                          00270000
 FILE SECTION.                                                           00280000
*                                                                        00290000
 FD  UPDT                                                                00300000
     RECORDING MODE F                                                    00310000
     BLOCK 0 RECORDS                                                     00320000
     RECORD 80 CHARACTERS                                                00330000
     LABEL RECORDS STANDARD.                                             00340000
   01 UPDATE-RECORD.                                                     00350000
       02 UPDATE-KEY            PIC 9(10).                              00370000
       02 UPDATE-IND             PIC X.                                  00371000
       02 UPDATE-DATA            PIC X(69).                              00380000
*                                                                        00390000
 FD  VSAMMAST                                                            00400000
     RECORD 100 CHARACTERS.                                              00410000
   01 VSAMMAST-RECORD.                                                   00420000
      02 VSAMMAST-KEY            PIC X(10).                              00430000
      02 VSAMMAST-DATA           PIC X(90).                              00440000
*                                                                        00450000
 FD  PFILE                                                               00460000
     RECORDING MODE F                                                    00470000
     BLOCK 0 RECORDS                                                     00480000
     RECORD 120 CHARACTERS                                               00490000
     LABEL RECORDS STANDARD.                                             00500000
   01 P-REC                     PIC X(120).                             00510000
*                                                                        00600000
```

```
 WORKING-STORAGE SECTION.                                          00610000
*                                                                  00620000
   01 WORKAREA.                                                    00630000
      02 END-FILE              PIC X VALUE ¢N¢.                   00640000
      02 CONSOLE-INPUT         PIC X VALUE ¢ ¢.                   00641000
      02 INVALID-KEY           PIC X VALUE ¢ ¢.                   00642000
      02 RECORD-LIMIT          PIC 9(8) VALUE 99999999.           00643000
      02 RECORD-COUNT          PIC 9(8) VALUE 0.                  00644000
*                                                                  00650000
   01 VSAMMAST-STATUS.                                            00660000
      02 VSAMMAST-FILE-STATUS  PIC XX.                            00670000
      02 VSAMMAST-VSAM-STATUS.                                    00680000
         03 VSAMMAST-RETURN-CODE PIC 9(2) COMP.                   00690000
         03 VSAMMAST-FUNCTION-CODE PIC 9(1) COMP.                 00700000
         03 VSAMMAST-FEEDBACK-CODE PIC 9(3) COMP.                 00710000
   01 P-RECORD.                                                    00711000
      02 P-STAT1               PIC XX.                            00712000
      02 FILLER                PIC X VALUE ¢ ¢.                   00713000
      02 P-STAT2               PIC 99.                            00714000
      02 FILLER                PIC X VALUE ¢ ¢.                   00715000
      02 P-STAT3               PIC 99.                            00716000
      02 FILLER                PIC X VALUE ¢ ¢.                   00717000
      02 P-STAT4               PIC 99.                            00718000
      02 FILLER                PIC X VALUE ¢ ¢.                   00718100
      02 P-DATA                PIC X(50).                         00719000
      02 P-COMMENTS            PIC X(50).                         00719100
*                                                                  00720000
 LINKAGE SECTION.                                                 00730000
   01 SYSTEM-PARMS.                                               00731000
      02 PARM0                 PIC 9(3) COMP.                     00731100
      02 PARM1                 PIC X.                             00732000
      02 FILLER                PIC X.                             00733000
      02 PARM2                 PIC X.                             00734000
      02 FILLER                PIC X.                             00735000
      02 PARM3                 PIC 9(4).                          00736000
* PARM1:                                                          00737000
*    L - LOAD A VSAM FILE WITH DATA IN UPDT                       00738000
*    U - UPDATE VSAM FILE WITH DATA FROM UPDT                     00739000
*    R - READ THE MASTER FILE AND WRITE TO PFILE                  00739100
*    S - THIS WILL READ THE UPDT FILE                             00739201
* PARM2:                                                          00739300
*    Q - QUICK EXIT. DO NOT ASK FOR ANYTHING.                     00739400
*    W - WAIT (DEFAULT)                                           00739500
*    P - PAUSE                                                    00739600
* PARM3:                                                          00739700
*    NNNN - RECORD COUNT. USED TO LIMIT NUMBER OF RECORDS         00739800
*           READ OR WRITTEN. WHEN USED WITH P, IT PAUSES          00739900
*           AFTER NNNN RECORDS THEN CONTINUE.                     00740000
*                                                                  00741000
 PROCEDURE DIVISION USING SYSTEM-PARMS.                           00750000
 MAIN-ROUTINE.                                                    00770000
     OPEN OUTPUT PFILE.                                           00780000
     OPEN INPUT UPDT.                                             00781000
     PERFORM OPEN-VSAMMAST THRU OPEN-VSAMMAST-EXIT.               00782001
     PERFORM ANALYZE-PARMS THRU ANALYZE-PARMS-EXIT.               00782100
*                                                                  00782600
```

```
 PROCESS-AGAIN.                                                          00783000
     IF PARM1 = ¢R¢                                                      00811000
        PERFORM PRINT-MASTER THRU PRINT-MASTER-EXIT                      00811100
            UNTIL END-FILE = ¢Y¢ OR RECORD-COUNT > RECORD-LIMIT          00812000
     ELSE                                                                00813000
        IF PARM1 = ¢S¢                                                   00814001
           PERFORM READ-INPUT THRU READ-INPUT-EXIT UNTIL                 00815001
             END-FILE = ¢Y¢                                              00816001
        ELSE                                                             00817001
        PERFORM PROCESS-UPDATES THRU PROCESS-UPDATES-EXIT                00820001
            UNTIL END-FILE = ¢Y¢ OR RECORD-COUNT > RECORD-LIMIT.         00830000
     PERFORM WAIT-AWHILE THRU WAIT-EXIT.                                 00831000
     IF PARM2 = ¢P¢ AND END-FILE = ¢N¢                                   00832000
        MOVE 0 TO RECORD-COUNT                                           00833000
        MOVE ¢Q¢ TO PARM2                                                00833100
        GO TO PROCESS-AGAIN.                                             00834000
     PERFORM CLOSE-FILES THRU CLOSE-FILES-EXIT.                          00840000
     STOP RUN.                                                           00850000
*                                                                        00851000
 PROCESS-UPDATES.                                                        00860000
     PERFORM READ-INPUT THRU READ-INPUT-EXIT UNTIL END-FILE = ¢Y¢ 00861001
     IF END-FILE = ¢Y¢ GO TO PROCESS-UPDATES-EXIT.                       00862001
*    READ UPDT AT END                                                    00870001
*        MOVE ¢Y¢ TO END-FILE                                            00880001
*        GO TO PROCESS-UPDATES-EXIT.                                     00890001
     MOVE SPACES TO P-RECORD.                                            00891000
     MOVE UPDATE-RECORD TO P-DATA.                                       00892000
     MOVE UPDATE-KEY TO VSAMMAST-KEY.                                    00893000
     MOVE SPACES TO VSAMMAST-STATUS.                                     00894000
     IF UPDATE-IND = ¢D¢                                                 00900000
         PERFORM DELETE-MASTER THRU DELETE-EXIT.                         00910000
*                                                                        00920000
     IF UPDATE-IND = ¢A¢                                                 00930000
         PERFORM ADD-MASTER THRU ADD-EXIT.                               00940000
*                                                                        00950000
     IF UPDATE-IND = ¢U¢                                                 00960000
         PERFORM UPDATE-MASTER THRU UPDATE-EXIT.                         00970000
     WRITE P-REC FROM P-RECORD AFTER ADVANCING 1.                        00971000
 PROCESS-UPDATES-EXIT. EXIT.                                             00980000
*                                                                        00981000
 CLOSE-FILES.                                                            00990000
     CLOSE UPDT.                                                         01000000
     CLOSE VSAMMAST.                                                     01010000
     CLOSE PFILE.                                                        01020000
 CLOSE-FILES-EXIT. EXIT.                                                 01021000
 WAIT-AWHILE.                                                            01030000
     IF PARM2 = ¢Q¢ GO TO WAIT-EXIT.                                     01031000
     IF PARM2 = ¢P¢                                                      01032000
        DISPLAY ¢PAUSING... PLEASE ENTER TO CONTINUE¢ UPON               01033000
        OPERATOR                                                         01034000
     ELSE                                                                01035000
        DISPLAY ¢ENTER ANYTHING TO CONTINUE¢ UPON OPERATOR.              01036000
     ACCEPT CONSOLE-INPUT FROM OPERATOR.                                 01040000
 WAIT-EXIT. EXIT.                                                        01041000
*                                                                        01050000
```

```
 DELETE-MASTER.                                                          01060000
     PERFORM READ-MASTER THRU READ-MASTER-EXIT.                          01061000
     IF INVALID-KEY = ¢Y¢                                                01070000
          MOVE ¢**ERROR READING MASTER **¢ TO P-COMMENTS                 01080000
          PERFORM MOVE-STATUS THRU MOVE-STATUS-EXIT                      01090000
          GO TO DELETE-EXIT.                                             01100000
     DELETE VSAMMAST INVALID KEY                                         01110000
          MOVE ¢**ERROR DELETING MASTER **¢ TO P-COMMENTS.               01120000
 DELETE-EXIT. EXIT.                                                      01160000
*                                                                        01161000
 UPDATE-MASTER.                                                          01170000
     PERFORM READ-MASTER THRU READ-MASTER-EXIT.                          01170100
     IF INVALID-KEY = ¢Y¢                                                01170200
          MOVE ¢**ERROR READING MASTER **¢ TO P-COMMENTS                 01170300
          PERFORM MOVE-STATUS THRU MOVE-STATUS-EXIT                      01170400
          GO TO UPDATE-EXIT.                                             01170500
     MOVE UPDATE-DATA TO VSAMMAST-DATA.                                  01180000
     REWRITE VSAMMAST-RECORD INVALID KEY                                 01431000
          MOVE ¢**ERROR UPDATING MASTER **¢ TO P-COMMENTS                01431100
          PERFORM MOVE-STATUS THRU MOVE-STATUS-EXIT.                     01432000
 UPDATE-EXIT. EXIT.                                                      01439300
*                                                                        01439500
 ADD-MASTER.                                                             01439600
      MOVE UPDATE-DATA TO VSAMMAST-DATA.                                 01439700
      PERFORM WRITE-MASTER THRU WRITE-MASTER-EXIT.                       01440900
      IF INVALID-KEY = ¢Y¢                                               01441000
          MOVE ¢**ERROR ADDING  MASTER **¢ TO P-COMMENTS                 01441200
          PERFORM MOVE-STATUS THRU MOVE-STATUS-EXIT.                     01441300
 ADD-EXIT. EXIT.                                                         01441400
*                                                                        01441500
 PRINT-MASTER.                                                           01441600
      PERFORM READ-MASTER THRU READ-MASTER-EXIT.                         01441700
      IF END-FILE = ¢Y¢ THEN GO TO PRINT-MASTER-EXIT.                    01441900
      WRITE P-REC FROM VSAMMAST-RECORD AFTER ADVANCING 1.                01442000
 PRINT-MASTER-EXIT. EXIT.                                                01442100
*                                                                        01442200
 MOVE-STATUS.                                                            01442300
      MOVE VSAMMAST-FILE-STATUS TO P-STAT1.                              01442400
      MOVE VSAMMAST-RETURN-CODE TO P-STAT1.                              01442500
      MOVE VSAMMAST-FUNCTION-CODE TO P-STAT3.                            01442600
      MOVE VSAMMAST-FEEDBACK-CODE TO P-STAT3.                            01442700
 MOVE-STATUS-EXIT. EXIT.                                                 01442800
*                                                                        01442900
 READ-MASTER.                                                            01443000
      MOVE ¢ ¢ TO INVALID-KEY.                                           01443100
      IF PARM1 = ¢R¢                                                     01443200
         READ VSAMMAST NEXT RECORD AT END                                01443300
              MOVE ¢Y¢ TO END-FILE                                       01443400
      ELSE                                                               01443500
         READ VSAMMAST INVALID KEY                                       01443600
              MOVE ¢Y¢ TO INVALID-KEY.                                   01444000
      ADD 1 TO RECORD-COUNT.                                             01444100
 READ-MASTER-EXIT. EXIT.                                                 01444200
*                                                                        01444301
 READ-INPUT.                                                             01444401
     READ UPDT AT END                                                    01444501
         MOVE ¢Y¢ TO END-FILE.                                           01444601
 READ-INPUT-EXIT. EXIT.                                                  01444802
*                                                                        01444901
 WRITE-MASTER.                                                           01445000
      MOVE ¢ ¢ TO INVALID-KEY.                                           01445100
      IF PARM1 = ¢L¢ WRITE VSAMMAST-RECORD ELSE                          01445200
         WRITE VSAMMAST-RECORD INVALID KEY                               01445300
         MOVE ¢Y¢ TO INVALID-KEY.                                        01445400
 WRITE-MASTER-EXIT. EXIT.                                                01445500
```

```
OPEN-VSAMMAST.                                                  01445600
    IF PARM0 = 0   THEN MOVE ¢ ¢ TO PARM1.                      01445700
    IF PARM1 = ¢L¢ OPEN OUTPUT VSAMMAST.                        01445800
    IF PARM1 = ¢U¢ OPEN I-O VSAMMAST.                           01445900
    IF PARM1 = ¢ ¢ OPEN I-O VSAMMAST.                           01446000
    IF PARM1 = ¢R¢ OPEN INPUT VSAMMAST.                         01446100
OPEN-VSAMMAST-EXIT. EXIT.                                       01446200
ANALYZE-PARMS.                                                  01446300
    DISPLAY ¢SPECIFIED PARMS: ¢ PARM1 PARM2 PARM3 UPON OPERATOR. 01446400
    IF PARM2 = ¢P¢ AND PARM3 < 0                                01446900
       DISPLAY ¢INVALID RECORD LIMIT SPECIFIED. SHOULD BE > 0¢  01447000
       UPON OPERATOR                                            01447100
       STOP RUN.                                                01447200
    IF PARM2 = ¢P¢                                              01447300
       MOVE PARM3 TO RECORD-LIMIT.                              01447400
ANALYZE-PARMS-EXIT. EXIT.                                       01447500
```

Appendix L. TSO/E RACVAR Module Updates

In order to use the RACVAR REXX exec, three TSO/E modules have to be updated:

- IRXPARMS
- IRXTSPRM
- IRXISPRM

A sample is provided below for reference. For details, refer to *RACF Program Directory for MVS Systems* and *TSO/E Version 2 REXX Reference*. In this sample:

- Changes to fields are marked with a "|"
- New field are marked with a "+"

```
IRXPARMS -- Member TSOREXX1 in SYS1.SAMPLIB

|         RPARMS   DS    CL384

       starting at field PACKTB_SYSTEM_TOTAL update:

|         PACKTB_SYSTEM_TOTAL DC F¢2¢
|         PACKTB_SYSTEM_USED DC F¢2¢
          PACKTB_LENGTH DC F¢8¢
          PACKTB_FFFF DC X¢FFFFFFFFFFFFFFFF¢
                   ORG   PACKTB+48
|         PACKTB_ENTRY DS CL16
                   ORG   PACKTB_ENTRY
          PACKTB_NAME DC CL8¢IRXEFMVS¢
+         PACKTB_NAME_RACF DC CL8¢IRREFPCK¢
|                  ORG   PACKTB+64
          LOCAL_PACKTB_ENTRIES DS CL8
                   ORG   LOCAL_PACKTB_ENTRIES
          PACKTB_NAME_LOCAL DC CL8¢IRXFLOC¢
|                  ORG   PACKTB+72
          USER_PACKTB_ENTRIES DS CL8
                   ORG   USER_PACKTB_ENTRIES
          PACKTB_NAME_USER DC CL8¢IRXFUSER¢
|                  ORG   RPARMS+384
```

```
IRXTSPRM -- Member TSOREXX2 in SYS1.SAMPLIB

|         RPARMS   DS    CL424

       starting at field PACKTB_SYSTEM_TOTAL update:

|         PACKTB_SYSTEM_TOTAL DC F¢3¢
|         PACKTB_SYSTEM_USED DC F¢3¢
          PACKTB_LENGTH DC F¢8¢
          PACKTB_FFFF DC X¢FFFFFFFFFFFFFFFF¢
                   ORG   RPARMS+384
|         PACKTB_ENTRIES DS CL16
                   ORG   PACKTB_ENTRIES
          PACKTB_ENTRY_MVS DS CL8
                   ORG   PACKTB_ENTRY_MVS
          PACKTB_NAME_MVS DC CL8¢IRXEFMVS¢
          PACKTB_NEXT_MVS DS 0C
                   ORG   PACKTB_ENTRIES+8
          PACKTB_ENTRY_TSO DS CL8
                   ORG   PACKTB_ENTRY_TSO
          PACKTB_NAME_TSO DC CL8¢IRXEFPCK¢
          PACKTB_NEXT_TSO DS 0C
+                  ORG   PACKTB_ENTRY+16
+         PACKTB_ENTRY_RACF DS CL8
+                  ORG   PACKTB_ENTRY_RACF
+         PACKTB_NAME_RACF DS CL8¢IRREFPCK¢
+         PACKTB_NEXT_RACF DS 0C
|                  ORG   RPARMS+408
          LOCAL_PACKTB_ENTRIES DS CL8
                   ORG   LOCAL_PACKTB_ENTRIES
          PACKTB_ENTRY_LOCAL DS CL8
                   ORG   PACKTB_ENTRY_LOCAL
          PACKTB_NAME_LOCAL DC CL8¢IRXFLOC¢
          PACKTB_NEXT_LOCAL DS 0C
|                  ORG   RPARMS+416
          USER_PACKTB_ENTRIES DS CL8
                   ORG   USER_PACKTB_ENTRIES
          PACKTB_ENTRY_USER DS CL8
                   ORG   PACKTB_ENTRY_USER
          PACKTB_NAME_USER DC CL8¢IRXFUSER¢
          PACKTB_NEXT_USER DS 0C
|                  ORG   RPARMS+424
```

```
IRXISPRM -- Member TSOREXX3 in SYS1.SAMPLIB

          RPARMS   DS    CL384

       starting at field PACKTB_SYSTEM_TOTAL update:

|         PACKTB_SYSTEM_TOTAL DC F¢3¢
|         PACKTB_SYSTEM_USED DC F¢3¢
          PACKTB_LENGTH DC F¢8¢
          PACKTB_FFFF DC X¢FFFFFFFFFFFFFFFF¢
                   ORG   RPARMS+344
|         PACKTB_ENTRIES DS CL24
                   ORG   PACKTB_ENTRIES
          PACKTB_ENTRY_MVS DS CL8
                   ORG   PACKTB_ENTRY_MVS
          PACKTB_NAME_MVS DC CL8¢IRXEFMVS¢
          PACKTB_NEXT_MVS DS 0C
                   ORG   PACKTB_ENTRIES+8
          PACKTB_ENTRY_TSO DS CL8
                   ORG   PACKTB_ENTRY_TSO
          PACKTB_NAME_TSO DC CL8¢IRXEFPCK¢
          PACKTB_NEXT_TSO DS 0C
+                  ORG   PACKTB_ENTRY+16
+         PACKTB_ENTRY_RACF DS CL8
+                  ORG   PACKTB_ENTRY_RACF
+         PACKTB_NAME_RACF DS CL8¢IRREFPCK¢
+         PACKTB_NEXT_RACF DS 0C
|                  ORG   RPARMS+368
          LOCAL_PACKTB_ENTRIES DS CL8
                   ORG   LOCAL_PACKTB_ENTRIES
          PACKTB_ENTRY_LOCAL DS CL8
                   ORG   PACKTB_ENTRY_LOCAL
          PACKTB_NAME_LOCAL DC CL8¢IRXFLOC¢
          PACKTB_NEXT_LOCAL DS 0C
|                  ORG   RPARMS+376
          USER_PACKTB_ENTRIES DS CL8
                   ORG   USER_PACKTB_ENTRIES
          PACKTB_ENTRY_USER DS CL8
                   ORG   PACKTB_ENTRY_USER
          PACKTB_NAME_USER DC CL8¢IRXFUSER¢
          PACKTB_NEXT_USER DS 0C
|                  ORG   RPARMS+384
```

Index

Special Characters

A

B

C

D

E

F

G

H

R

S

T

U

V

W

MVS/ESA and
RACF Version 1 Release 9
Security Implementation Guide

READER'S COMMENT FORM

International Technical Support Center Bulletin GG24-3585-00

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments:

If you wish a reply, give your name, company, mailing address, and date:

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM International Technical Support Center
Department H52, Building 930
P.O. Box 950
Poughkeepsie, New York 12602
U.S.A.

Fold and Tape | Please Do Not Staple | Fold and Tape

IBM®

MVS/ESA and RACF Version 1 Release 9 Security Implementation Guide

Printed in U.S.A.

GG24-3585-00

MVS/ESA and
RACF Version 1 Release 9
Security Implementation Guide

READER'S COMMENT FORM

International Technical Support Center Bulletin GG24-3585-00

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments:

If you wish a reply, give your name, company, mailing address, and date:

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM International Technical Support Center
Department H52, Building 930
P.O. Box 950
Poughkeepsie, New York 12602
U.S.A.

Fold and Tape | Please Do Not Staple | Fold and Tape

IBM®

MVS/ESA and RACF Version 1 Release 9 Security Implementation Guide

Printed in U.S.A.

GG24-3585-00

MVS/ESA and RACF Version 1 Release 9 Security Implementation Guide Printed in U.S.A. GG24-3585-00

GG24-3585-00

APAFOIL Processing Options

APAFOIL **August 31, 1990** **Release 3.0**

Runtime values:

DEVICE 3820A
BIND (Odd, Even) 1.00i, 1.00i
TWOPASS . . . . . NO
INDEX . . . . . YES

Foil Set: 1

Input File (Current) GG243585

Layout of Heading (FOILHD Tag or Default)
FOILHD . . . . . HEAD NULL PRODUCT
FRAME . . . . . NONE

Layout of Body (LAYOUT Tag or Default)
FRAME . . . . . RULE
FRAMEWT . . . . . BOLD
RULE . . . . . SOLID
BORDER . . . . . NONE
RUBRICWT . . . . . LIGHT

Layout of Footing (FOILFT Tag or Default)
FOILFT . . . . . DATE STATUS NUMBER
FRAME . . . . . NONE

Statistics:
Title Page 0
Contents 0
Parts 0
Foils 0
Notes 0
Overflow 0
Total Pages 0

APAFOIL Messages:
Information 0
Warning 0
Error 0

System Variables:
SYSVAR B . . . . . MID
SYSVAR N . . . . . INCLUDE

Grid Definitions

id	File	Page	References
NJEFLOW	I$3NJE	135	135
NJEXMP	I$3NJE	136	137, 138, 139, 140, 141, 143, 144, 145, 147, 148, 149, 151
NJEDSEC	APPNJE	269	269, 270, 271, 272, 273

Table Definitions

id	File	Page	References
MLSSU1	SECLUSE	29	30
MLSSU2	SECLUSE	29	29
HEAD	APPCDT	265	265
BODY	APPCDT	265	265
AROW	APPSDSF	296	296
WROW	APPSDSF	298	299
XROW	APPSDSF		

Example Definitions

Figures

Headings

Index Entries

			(2) LU 6.2 session control
			235, 235
RCL2	MISC		
		231	(1) RACF classes
			(2) APPCLU
			235
HICON1	MISC		
		238	(1) batch local shared resource
			(2) Hiperspace control
			240
HICON2	MISC		
		242	(1) Hiperbatch
			(2) Hiperspace control
			245
RCL3	MISC		
		242	(1) RACF classes
			(2) DLFCLASS
			248
TSOCON	MISC		
		249	(1) TSO
			(2) message control
			253
RCL4	MISC		
		250	(1) RACF classes
			(2) SMESSAGE
			251
RCL5	MISC		
		251	(1) RACF classes
			(2) DIRAUTH
			252

Tables

id	File	Page	References
COMPAR	SECLINT		
		16	1
			16
MLSSUM	SECLUSE		
		30	2
			29
FUNCLS	RACF		
		33	3
			33
PROSEG	RACF		
		34	4
			34
TRSUM	RACF		
		36	5
			36
CONVER	RACF		
		37	6
			36
MSTAB	I$3JOB		
		71	7
			71
POES	I$3JOB		
		80	8
			80, 80, 85
J3NEW	I$3J2SPL		
		101	9
			100
NJETRL	I$3NJE		
		119	10
			119, 132
SYSA	I$3NJE		
		128	11
			128
SYSB	I$3NJE		
		129	12
			129, 129
SYSC	I$3NJE		
		129	13
			129
COMUTOK	I$3J3CMD		
		174	14
			174, 185
MCSAUTH	I$3J3CMD		
		175	15
			175
ACTION	I$3J3CMD		
		176	16
			175
MCSRACF	I$3J3CMD		

Processing Options

Runtime values:

Document fileid	GG243585 SCRIPT
Document type	USERDOC
Document style	IBMXASV
Profile	EDFPRF30
Service Level	0029
SCRIPT/VS Release	4.0.0
Date	95.08.10
Time	09:54:45
Device	3820A
Number of Passes	3
Index	YES
SYSVAR B	MID
SYSVAR G	INLINE
SYSVAR N	YES
SYSVAR P	FINAL

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
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	YES
Folio-by-chapter	NO
Head 0 body text	(none)
Head 1 body text	Chapter
Head 1 appendix text	Appendix
Hyphenation	NO
Justification	NO
Language	ENGL
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	FINAL
Punctuation move characters	.,
Read cross-reference file	(none)
Running heading/footing rule	NONE
Show index entries	NO
Table of Contents (maximum level)	3
Table list on new page	YES
Title page (draft) alignment	RIGHT
Write cross-reference file	(none)

Imbed Trace

Page 0	APAFOIL
Page 0	$SYMBOL
Page i	$EDTN
Page ii	$ABS
Page xv	$SPEC
Page xvii	$TM
Page xvii	$PREF
Page xix	$PUB
Page xxi	$ABBR
Page xxii	$BODY
Page xxii	INTRO
Page 8	SECLINT
Page 16	SECLUSE
Page 32	RACF
Page 62	SAF
Page 70	I$3JOB
Page 94	I$3J2SPL
Page 115	I$3NJE
Page 157	I$3RJE
Page 161	I$3J3CMD
Page 184	I$3J2CMD
Page 190	PSF
Page 204	SDSF
Page 221	MISC
Page 254	B1
Page 262	$APP
Page 262	APPFUNC
Page 264	APPCDT
Page 267	APPNJE
Page 273	APPRARW
Page 275	APPJ3CMD
Page 279	APPJ2CMD
Page 284	APPPSF
Page 293	APPSDSF
Page 304	APPLU62
Page 309	APPDLF
Page 315	APPTSO
Page 328	$BACK